

Doctoral thesis

Doctoral theses at NTNU, 2022:289

Sander Johannes Simon Roet

# Accelerating the understanding of chemistry

with path-sampling and human understandable machine learning algorithms

**NTNU**  
Norwegian University of Science and Technology  
Thesis for the Degree of  
Philosophiae Doctor  
Faculty of Natural Sciences  
Department of Chemistry



Norwegian University of  
Science and Technology



Sander Johannes Simon Roet

# **Accelerating the understanding of chemistry**

with path-sampling and human understandable  
machine learning algorithms

Thesis for the Degree of Philosophiae Doctor

Trondheim, October 2022

Norwegian University of Science and Technology  
Faculty of Natural Sciences  
Department of Chemistry



Norwegian University of  
Science and Technology

**NTNU**

Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Natural Sciences

Department of Chemistry

© Sander Johannes Simon Roet

ISBN 978-82-326-6856-4 (printed ver.)

ISBN 978-82-326-6406-1 (electronic ver.)

ISSN 1503-8181 (printed ver.)

ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:289

Printed by NTNU Grafisk senter



# Abstract

Understanding chemistry is essential for the optimization of reactions and the development of new reactions. Chemical reactions can be investigated by simulations without wasting any precious materials or be influenced by experimental artifacts and four of the six papers included in this thesis use simulations to investigate a wide range of chemistry. They investigate the effect of breaking assumptions of enzymatic assays for covalent inhibitors, the permeation of ions through a membrane, the effect of an oncogenic mutation on protein movements, and the deprotonation pathways for formic acid in atmospheric water droplets.

One of the most efficient ways of simulating chemical reactions is replica exchange transition interface sampling (RETIS), where we focus the simulation on the reaction without wasting computational resources on simulating either the reactants or the products. In three of the six papers we further developed the RETIS algorithms and software, we interfaced with more molecular dynamics software, introduced more efficient Monte Carlo moves, and parallelized the RETIS algorithm. All of these increase the speed of RETIS simulations by orders of magnitude.

Additionally, one of the papers specifically focuses on enhancing the analysis of RETIS simulations with machine learning (ML) algorithms. For this we introduce a new data representation that is translational, rotational and atom index invariant without any preselection of important variables or losing the ability to regenerate a 3D structure from it. This representation is then used with a human understandable ML algorithm, Decision Trees (DTs). The paper also introduces a way of investigating different initial splits of DTs with the help of random forests. This helps increasing the speed at which we can do analysis of RETIS simulations, while also reducing the risk of hypothesis bias.



# Acknowledgments

First of all, I would like to thank my supervisor, Prof. Titus van Erp, for both giving me interesting research projects and allowing me to pursue my own interests. Our discussions and your insight were invaluable to me. I would also like to thank my co-supervisors, Prof. Helge Langseth and Prof. Bruno Pollet, for pointing me to interesting courses to follow and discussing interesting applications for my project.

Thanks to Anders, Raffaella, and Enrico for the amazing welcoming into the reasearch group and continued support througouht my PhD. You made me feel right at home. I would like to give a special thanks to my longest office mate Rolf, specifically for telling me to go home and taking me on unforgettable trips to his family's cabin, together with Asgeir and Julian. I am also very gratefull to have met Anders, Daniel, Regina, Eirik, Sarai, Rosario, Tor, Inge, Alex, Linda, Marcus, and Eivind. Thanks for being amazing colleagues, the great coffee chats, keeping me fit, organizing fun events, and being my rubber ducky whenever I needed one, specifically the people in office D3-124. Additional thanks to Sarai and Alex for providing this  $\LaTeX$  template. To my roommates, Hanne, Enrico, Sahin, Cyril, Mikael and Kiki, Federico, Davide, and Mado thanks for always making me feel at home while I was in Trondheim

For hosting my research stay in Helsinki I would like to thank Chris. I really enjoyed seeing the city and having the 'traditional' buffet lunches with you. A special thanks as well for making it possible to include our paper in this thesis.

For making my trips back to Amsterdam always fun an eventfull, I would like to thank Marieke, KT, Kelly, Mike, Roos, Chris, Rosa, Joris, Lisette, Pieter, Kune, Sacha, and the rest of the crew. A special thanks for the colleagues of Elma for allowing me to join their 'borrels' whenever I was in the country.

Elma, thank you for being ok that I decided to move abroad for 4 years, for always understanding and supporting me, for coming up with great ideas, making me smile and even for writing a paper with me.

I am gratefull for the support and continued interest in my work from my family, Joke, Janna, and Dennis, and the family of Elma; Harry, Dorine, Dirk-Jan, Ellen, and Sander.

Lastly, I would like to thank Piet, my father, for inspiring me to always ask questions and pushing me to be the best I can be. You got to see the start of my PhD, but unfortunately won't get to see the end. I know you would be proud.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 List of papers included in this thesis	4
<b>2 Theoretical background</b>	<b>7</b>
2.1 Molecular Dynamics	7
2.2 Monte Carlo	8
2.3 Transition Path Sampling	9
2.4 Transition Interface Sampling	13
2.5 Replica Exchange Transition Interface Sampling	14
2.6 Decision Trees and random forests	17
<b>3 Accelerating Replica Exchange Transition Interface Sampling Simulations</b>	<b>19</b>
3.1 Leveraging MD engines	21
3.2 Leveraging specialized MC moves	23
3.3 Doing an infinite amount of swapping	26
3.4 Parallelizing the RETIS algorithm	32
<b>4 Accelerating the Analysis</b>	<b>37</b>
4.1 Translational, rotational, and index invariant data representation	38
4.2 Using human understandable ML algorithms on RETIS data	42
<b>5 Applications to biochemical problems</b>	<b>45</b>
5.1 Conformational changes of KRas	45
5.2 Using kinetic simulations for illustrating Covalent Inhibition in Enzymatic Assays	47

<b>6 Conclusion and outlook</b>	<b>49</b>
<b>Bibliography</b>	<b>51</b>
<b>A PyRETIS 2: An improbability drive for rare events</b>	<b>59</b>
<b>B Exact non-Markovian permeability from rare event simulations</b>	<b>69</b>
<b>C Chemistrees: Data-Driven Identification of Reaction Pathways via Machine Learning</b>	<b>95</b>
<b>D A Comprehensive Guide for Assessing Covalent Inhibition in Enzymatic Assays Illustrated with Kinetic Simulations</b>	<b>119</b>
<b>E Exchanging replicas with unequal cost, infinitely and permanently</b>	<b>207</b>
<b>F Path sampling simulations reveal how the Q61L mutation alters the dynamics of KRas</b>	<b>231</b>



# 1 Introduction

The first thing you would think of chemistry is that it is a very experimental research field where a lot is done inside laboratories. While a lot of theories exist, finding new reactions or molecules very often still consist of trying known reactions with slightly altered reactants to see if you get the expected product. Or, in biochemistry, if you want to test how good a new drug binds to the intended target you might need to do multiple experiments to find the right conditions. This can waste a lot of precious material and can be suspect to experimental artifacts. In this case, simulations can help to investigate what changing certain parameters would do without actually having to do the experiment.

In paper D we used did exactly that. We used numerical simulations to investigate the effect of breaking critical assumptions when using certain analysis formulas for calculating inhibitor potency of covalent inhibitors, without having to do the experiments. We also made these simulation scripts available in running environments for the readers via binder<sup>1</sup> so they can easily understand the limits for their own experiments.

Another type of simulation, Molecular Dynamics (MD), allows us to look at reactions at an atomistic level and femtosecond timescale. In these simulations we simulate what happens during chemical reactions and biochemical processes and can try to understand them. However, most reactions and processes are so-called rare-events. For example, if a reaction takes 100 ns to complete, but only happens once every second. Then we are only spending 0.0001% of the time simulating the interesting part (the reaction), while spending most time looking at either the reac-

tant or product states. Also, simulating a second of a process still takes up to 28 years even on the fastest available supercomputer, which can simulate 100  $\mu$ s per day.<sup>2</sup>

Transition Path Sampling (TPS)<sup>3</sup> simulations solve this inefficiency by focusing the MD to only simulate the reaction paths. This allows us to still have a detailed understanding of the reaction, without wasting simulation time on things we are not interested in.

In paper F we used multiple state TPS to investigate the dynamic behavior of a protein, and the differences compared with an oncogenic mutant. This would not have been possible with just MD.

An important property for reactions that we would like to compute are reaction rates. Reaction rates are used for predicting drug effectiveness as in paper D and which molecules/molecular structures will be produced under certain reaction conditions.<sup>4</sup> TPS allows us to compute rates with an algorithm that is based on umbrella sampling, but it requires long simulations for complex pathways.<sup>5</sup>

Instead we use (Replica Exchange) Transition Interface Sampling ((RE)TIS),<sup>6,7</sup> where we use Monte Carlo (MC) algorithms to generate ensembles of paths that are forced to proceed further and further along the reactions, until reaction completion. We then can compute the rate by computing the fraction of paths in an ensemble that would also be valid for an ensemble that is forced to be further along the reaction and multiplying all these fractions. The TIS algorithm is at least twice as efficient for computing rates than the TPS method.

The Replica Exchange move is the main difference between TIS and RETIS. This move allows us to swap information between ensembles for 'free' (without running more MD).

There are two open-source implementations of RETIS simulations, OpenPathSampling (OPS),<sup>8,9</sup> and PyRETIS.<sup>10,11</sup> Recent implementations in improvements of our software, PyRETIS, are described in paper A. One of the improvements mentioned is that we added an interface with OpenMM.<sup>12</sup> This allows us to run the MD on GPUs, accelerating these simulations a 30-fold.<sup>13</sup>

While increasing the MD speed helps us getting the answer faster in RETIS simulations, you can also increase the efficiency of how we use this



MD. This has been done by several new MC protocols, further focusing the MD to certain regions. These protocols use new MC moves for generating paths such as Shooting from the Top,<sup>14</sup> Stone Skipping,<sup>15</sup> and Wire Fencing.<sup>16</sup>

In paper B we developed and successfully applied 2 new MC moves, the mirror-move and the target-swap-move, that greatly increased the sampling efficiency for investigating permeants traveling through a membrane.

While RETIS uses CPUs more efficient than TIS, it still has two important limitations: First, while TIS could run each ensemble independently and at the same time, the replica exchange moves in RETIS makes that only possible until two ensembles have to swap (exchange their replicas). At that point, the fast ensemble has to wait for the slow one to finish.<sup>17</sup> Even this algorithm is hard to implement and both OPS and PyRETIS implement RETIS as a sequential algorithm.

Secondly, you would like to do an infinite amount of swaps between the ensembles before running MD as this would give the most 'free' information as possible. However, this would take an infinite amount of time where it would only resample data and not generate any new data and is therefore not useful in practice. In both OPS and PyRETIS it is customary to have a 50% probability of doing either a swapping move or a shooting move.

In paper E we break both of these limitations. We solve the parallelization issue by reformulating the detailed-balance equation to allow doing swaps without waiting on any ensemble to finish. This reformulation speeds up our simulations depending on how many computers you have available and even distributes these more efficiently than either the TIS or RETIS algorithm. We also increased the probability of swaps to infinity with an infinite swapping approach,<sup>18</sup> but reformulated it into the computation of permanents. This reformulation circumvents the steep factorial scaling reported before, and solves the second limit of RETIS. We named this new application  $\infty$ RETIS. With this we greatly increase how fast we get an answer from our RETIS simulations.

After RETIS simulations are completed we can compute the rate, but we also would like to understand what environments trigger a chemical reaction. To understand reactions from RETIS simulations, you normally

look at the generated paths. However, with the increase in computing power we now can generate so many reaction paths, that you can find proof for almost any (reasonable) hypothesis in your data. This means that there is a risk of hypothesis bias, where not the most common reason for a reaction is reported, but only the ones you expected to find beforehand.

In paper C we tried to solve this problem. We developed a data representation that is invariant to translation, rotation, and atom-indices and requires minimal user input to prevent hypothesis-bias. This was then used to train a decision tree (DT) on the question “does this lead to the deprotonation of formic acid?”. Decision trees are interpretable machine learning algorithms, thus after training this algorithm we could understand what environments would trigger this reaction.

The thesis starts with a select introduction to molecular dynamics, path sampling and decision trees. The following chapters present the improvements made to the path-sampling simulations, the usage of human understandable machine learning for the analysis of simulations. The applications of simulations are presented in chapter 5 and the last chapter presents an outlook.

## 1.1 List of papers included in this thesis

### **Paper A:**

*PyRETIS 2: An improbability drive for rare events*

Enrico Riccardi, Anders Lervik, Sander Roet, Ola Aarøen,  
and Titus S. van Erp

*J. Comput. Chem.* **2020**, *41*, 370–377,

doi: 10.1002/jcc.26112

### **Paper B:**

*Exact non-Markovian permeability from rare event simulations*

An Ghysels, Sander Roet, Samaneh Davoudi, and Titus S. van Erp

*Phys. Rev. Research* **2021**, *3*, 033068,

doi: 10.1103/PhysRevResearch.3.033068

**Paper C:**

*Chemistrees: Data-Driven Identification of Reaction Pathways via Machine Learning*

Sander Roet, Christopher D. Daub, and Enrico Riccardi

*J. Chem. Theor. Comput.* **2021**, *17*, 6193–6202,

doi: 10.1021/acs.jctc.1c00458

**Paper D:**

*A Comprehensive Guide for Assessing Covalent Inhibition in Enzymatic Assays Illustrated with Kinetic Simulations*

Elma Mons, Sander Roet, Robert Q. Kim, and Monique P.C. Mulder

*Current Protocols* **2022**, *2*, e419,

doi: 10.1002/cpz1.419

**Paper E:**

*Exchanging replicas with unequal cost, infinitely and permanently*

Sander Roet, Daniel T. Zhang, and Titus S. van Erp

*arXiv preprint arXiv:2205.12663v1 [physics.comp-ph]*

doi: 10.48550/arXiv.22505.12663

**Paper F:**

*Path sampling simulations reveal how the Q61L mutation alters the dynamics of KRas*

Sander Roet, Ferry Hooft, Peter G. Bolhuis, David W.H. Swenson, and

Jocelyne Vreede

*Manuscript*



## 2 Theoretical background

This chapter contains a highly selective description of the theories that are required background knowledge for the rest of the thesis. Every section ends with a proposed reference for recent developments or greater understanding.

### 2.1 Molecular Dynamics

Molecular Dynamics (MD) allows us to simulate molecules at an atomistic level. It involves a repeating algorithm that moves the atoms through time by solving Newton's equations of motion. The most simple algorithm is the Euler algorithm, but it is neither time symmetric nor area preserving, so it is not suitable for equilibrium simulations.

The simplest time symmetric algorithm is the so called Leap-Frog algorithm.<sup>19</sup> However, the velocities are offset by half a time step from the positions, so it does not define the velocities and positions at the same time. This can be solved by doing a half step velocity update, which is not practical in path sampling codes. Therefore the standard MD algorithm for path sampling is the velocity-Verlet algorithm,<sup>20</sup> which consists of the following steps:

$$\begin{aligned}
v_i(t + \frac{1}{2}\Delta t) &= v_i(t) + \frac{1}{2}a_i(t)\Delta t \\
x_i(t + \Delta t) &= x_i(t) + v_i(t + \frac{1}{2}\Delta t)\Delta t \\
a_i(t + \Delta t) &= \frac{F_i(x(t + \Delta t))}{m_i} \\
v_i(t + \Delta t) &= v_i(t + \frac{1}{2}\Delta t) + \frac{1}{2}a_i(t + \Delta t)\Delta t
\end{aligned}$$

where  $\Delta t$  is the time step,  $x_i(t)$  is the position of atom  $i$  at time  $t$ ,  $x(t)$  is the position vector of all atoms at time  $t$ ,  $v_i(t)$  is the velocity of atom  $i$  at time  $t$ ,  $a_i(t)$  is the acceleration of atom  $i$  at time  $t$ ,  $F_i(x(t))$  is the force on atom  $i$ , and  $m_i$  is the mass of atom  $i$ .

The forces for the atoms can be computed either via a force field, or from quantum chemical calculations.<sup>21,22</sup> MD allows us to simulate trajectories where we can see atoms move through time.

More information about Molecular Dynamics can be found in ref. 23, and an investigation on the effect of integrator splitting (like done here for the velocity updates) can be found in ref. 24.

## 2.2 Monte Carlo

If we are only interested in equilibrium properties and don't care about the dynamics, we can use a Monte-Carlo algorithms instead.<sup>25</sup> The basic algorithm is as follows:

1. Generate a new state  $S^{(n)}$ , possibly from the old state  $S^{(o)}$ .
2. Calculate the acceptance probability  $P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)})$
3. Draw a random number,  $r$  in  $[0, 1)$
4. If  $r < P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)})$  accept  $S^{(n)}$  and count it. Else throw away  $S^{(n)}$  and recount  $S^{(o)}$ . Go to step 1

If we want to sample an equilibrium distribution with MC then the number of moves out of the old state to the new state should be balanced by the moves to the old state from the new state:

$$\rho(S^{(o)})\pi(S^{(o)} \rightarrow S^{(n)}) = \rho(S^{(n)})\pi(S^{(n)} \rightarrow S^{(o)}) \quad (2.1)$$

where  $\rho(S^{(o)})$  is the equilibrium distribution of  $S^{(o)}$  and  $\pi(S^{(o)} \rightarrow S^{(n)})$  is the probability of going from  $S^{(o)}$  to  $S^{(n)}$ , given a set of MC moves.

This balance is normally enforced by  $P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)})$  and if we use the Metropolis-Hastings<sup>26</sup> algorithm it is defined as

$$P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)}) = \min \left[ 1, \frac{\rho(S^{(n)})P_{\text{gen}}(S^{(n)} \rightarrow S^{(o)})}{\rho(S^{(o)})P_{\text{gen}}(S^{(o)} \rightarrow S^{(n)})} \right] \quad (2.2)$$

where  $P_{\text{gen}}(S^{(o)} \rightarrow S^{(n)})$  is the probability of generating  $S^{(n)}$  from  $S^{(o)}$ . If this is equal to  $P_{\text{gen}}(S^{(n)} \rightarrow S^{(o)})$  one could use the less general metropolis<sup>25</sup> acceptance instead:

$$P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)}) = \min \left[ 1, \frac{\rho(S^{(n)})}{\rho(S^{(o)})} \right] \quad (2.3)$$

More information about Monte Carlo sampling can be found in ref. 23.

## 2.3 Transition Path Sampling

Transition Path Sampling<sup>3</sup> (TPS) is the basis of all path sampling algorithms in this thesis. The main idea is that we focus the MD by applying MC to transition paths.

Firstly, paths are a sequence of frames of phase points obtained from a MD simulation. For a path of length  $L$ :

$$X_L = (x_0, x_1, \dots, x_{L-1})$$

where  $X$  is a path and  $x_i$  is a frame obtained from MD (in order). For any path, the path probability  $\rho(X_L)$  is;

$$\rho(X_L) = \frac{1}{Z} \rho(x_0) \prod_{i=0}^{L-2} P(x_i \rightarrow x_{i+1})$$

where  $P(x_i \rightarrow x_{i+1})$  is the probability of generating frame  $x_{i+1}$  from frame  $x_i$ , and  $Z$  is a normalization constant.

A transition path is a path that starts with a frame in state A (commonly the reactant state) and ends with a frame in state B (commonly the product state). To check this we define the following indicator function:

$$h_A(x_i) = \begin{cases} 1, & \text{if } x_i \text{ in state A} \\ 0, & \text{otherwise} \end{cases}$$

Similarly we can define  $h_B$  for state B and the path probability of a transition path becomes

$$P_{\text{tps}}(X_L) = h_A(x_0)h_B(x_{L-1})\rho(X_L)$$

From here we are going to drop the subscript  $L$  for the paths. The acceptance probability for TPS then becomes:

$$P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_A(x_0^{(n)})h_B(x_{L'-1}^{(n)})\rho(X^{(n)})P_{\text{gen}}(X^{(n)} \rightarrow X^{(o)})}{h_A(x_0^{(o)})h_B(x_{L'-1}^{(o)})\rho(X^{(o)})P_{\text{gen}}(X^{(o)} \rightarrow X^{(n)})} \right]$$

where the  $L'$  is to indicate that the length can be different between  $x_{L'-1}^{(n)}$  and  $x_{L'-1}^{(o)}$ . If we then assume that the old path is a valid transition path

$$h_A(x_0^{(o)}) = h_B(x_{L'-1}^{(o)}) = 1$$

This leaves us with

$$P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_A(x_0^{(n)})h_B(x_{L'-1}^{(n)})\rho(X^{(n)})P_{\text{gen}}(X^{(n)} \rightarrow X^{(o)})}{\rho(X^{(o)})P_{\text{gen}}(X^{(o)} \rightarrow X^{(n)})} \right] \quad (2.4)$$

where  $P_{\text{gen}}$  depends on the way we generate a new path from an old path.

One of the common ways of generating a new path is the so-called shooting move. In this move we:

1. select a random frame from  $X^{(o)}$ ,  $x$
2. change the velocities of  $x$ , to make frame  $x'$
3. integrate from  $x'$  backwards in time until you hit a stable state
4. integrate from  $x'$  forward in time until you hit a stable state and add it to the backward sub-trajectory to form a new trajectory,  $X^{(n)}$

Steps 3 and 4 depend on the MD, so  $P_{\text{gen}}$  can be split into.

$$P_{\text{gen}}(X^{(o)} \rightarrow X^{(n)}) = P_{\text{sel}}(x|X^{(o)})P_{\text{vel}}(x \rightarrow x')P_{\text{MD}}(X^{(n)}|x')$$

where  $P_{\text{sel}}(x|X^{(o)})$  is the probability of selecting frame  $x$  given the old trajectory  $X^{(o)}$ ,  $P_{\text{vel}}(x \rightarrow x')$  is the probability of altering the velocities of  $x$  to the velocities of  $x'$ , and  $P_{\text{MD}}(X^{(n)}|x')$  is the probability of generating  $X^{(n)}$  from  $x'$ . As we assume microscopic reversibility;

$$\rho(x')P_{\text{MD}}(X^{(n)}|x') = \rho(X^{(n)})$$



this leads to the following

$$P_{\text{gen}}(X^{(o)} \rightarrow X^{(n)}) = \frac{P_{\text{sel}}(x|X^{(o)})P_{\text{vel}}(x \rightarrow x')\rho(X^{(n)})}{\rho(x')}$$

Filling this in eq 2.4, becomes

$$P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_A(x_0^{(n)})h_B(x_{L-1}^{(n)})P_{\text{sel}}(x'|X^{(n)})P_{\text{vel}}(x' \rightarrow x)\rho(x')}{P_{\text{sel}}(x|X^{(o)})P_{\text{vel}}(x \rightarrow x')\rho(x)} \right] \quad (2.5)$$

If we select the shooting point as a random frame not in a stable state,

$$P_{\text{sel}}(x|X^{(o)}) = \frac{1}{\text{len}(X^{(o)}) - 2}$$

where  $\text{len}(X^{(o)})$  is the length of trajectory  $X^{(o)}$ , including the endpoints in the stable states. Furthermore, the probability of a frame can be split in the probability of the coordinates,  $\rho(r)$ , and the probability of the velocities,  $\rho(v)$ ;

$$\rho(x) = \rho(r)\rho(v)$$

and

$$\rho(x') = \rho(r')\rho(v')$$

If the new velocities are sampled from a Maxwell-Boltzmann distribution, then;

$$P_{\text{vel}}(x \rightarrow x') = \rho(v')$$

Filling this in 2.5 leads to

$$P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_A(x_0^{(n)})h_B(x_{L-1}^{(n)})(\text{len}(X^{(o)}) - 2)\rho(v)\rho(r)\rho(v')}{(\text{len}(X^{(n)}) - 2)\rho(v')\rho(r)\rho(v)} \right] \quad (2.6)$$

If the coordinates are not altered, so  $r = r'$ , this can be simplified to the standard flexible length shooting acceptance:

$$P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_A(x_0^{(n)})h_B(x_{L-1}^{(n)})(\text{len}(X^{(o)}) - 2)}{\text{len}(X^{(n)}) - 2} \right] \quad (2.7)$$

One small comment,  $\text{len}(X^{(n)}) = 2$  should never happen as at least 1 frame outside of the stable states from the old trajectory should be part of the new trajectory.

If we start by sampling a random number, instead of following the sequence as stated in the start of section 2.2, we can know beforehand what the maximum length of the new path is allowed to be. This can be useful because we can then stop long paths that would have been rejected anyway. A path is accepted when

$$r < P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)}) \quad (2.8)$$

where  $r$  is a random number in  $[0, 1)$ , we can replace  $P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)})$  via eq. 2.7

$$r < \min \left[ 1, \frac{h_A(x_0^{(n)})h_B(x_{L-1}^{(n)})(\text{len}(X^{(o)}) - 2)}{\text{len}(X^{(n)}) - 2} \right] \quad (2.9)$$

as  $r$  is always less than 1, only the right element of the  $\min[\dots]$  can lead to a number smaller than  $r$

$$r < \frac{h_A(x_0^{(n)})h_B(x_{L-1}^{(n)})(\text{len}(X^{(o)}) - 2)}{\text{len}(X^{(n)}) - 2} \quad (2.10)$$

this can be rearranged into

$$\text{len}(X^{(n)}) - 2 < \frac{h_A(x_0^{(n)})h_B(x_{L-1}^{(n)})(\text{len}(X^{(o)}) - 2)}{r} \quad (2.11)$$

if the path is assumed to be a valid transition path,  $h_A(x_0^{(n)}) = h_B(x_{L-1}^{(n)}) = 1$ , this gives the definition of the maximum allowed path length for the new path

$$\text{len}(X^{(n)}) < \frac{\text{len}(X^{(o)}) - 2}{r} + 2 \quad (2.12)$$

The acceptance in eq 2.7 can also be extended to transitions between more than 2 states, as was used for paper F. In this extension any transition path between two different stable states is allowed and the acceptance becomes

$$P_{\text{acc}}(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_{\text{any}}(x_0^{(n)})h_{\text{any other}}(x_{L-1}^{(n)})(\text{len}(X^{(o)}) - 2)}{\text{len}(X^{(n)}) - 2} \right]$$

where  $h_{\text{any}}$  is the indicator function that is 1 if the frame is in any of the stable states, and  $h_{\text{any other}}$  is an indicator function that is 1 if the frame is in any stable state except for the stable state of  $h_{\text{any}}$ , and 0 otherwise.

## 2.4 Transition Interface Sampling

TPS allows us to investigate the mechanism of transition paths and allows us to compute reaction rates based on umbrella sampling, however this is very inefficient.<sup>5</sup> To make the computation of the rate more efficient, Transition Interface Sampling (TIS) was developed.<sup>6</sup>

For TIS we start by defining a collective variable (CV) which is some function that takes the state of the system (positions and velocities) and returns a number that represents the progress between two stable states. We then place a series of interfaces (non intersecting hyper surfaces) on the CV values between the two stable states,  $\lambda_0, \lambda_1, \dots, \lambda_B$ , where  $\lambda_0$  defines state A, and  $\lambda_B$  defines state B. For each of the interfaces we start a simulation with a different definition of valid paths. Every simulation thus samples a different ensemble of paths. For every interface,  $i$ , an ensemble is sampled where a valid path is defined as a path that:

1. starts with a single frame in state A
2. crosses  $\lambda_i$ , while not being inside any stable state.
3. ends with a single frame in state A or state B

These requirements are greedy such that a valid path only has 2 frames inside a stable state, the starting and end frame, all others have to be outside the stable state. The acceptance probability (of a shooting move) for interface  $i$  then becomes

$$P_{\text{acc}}^i(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_A(x_0^{(n)})h_{AB}(x_{L-1}^{(n)})h_i(X^{(n)})(\text{len}(X^{(o)}) - 2)}{\text{len}(X^{(n)}) - 2} \right]$$

where  $h_{AB}(x)$  is the indicator function if frame  $(x)$  is in state A or state B, and  $h_i(X)$  is an indicator function if trajectory  $X$  crosses interface  $i$ . If the velocities are not resampled randomly, an extra term is added for the energy difference.

With these different sampling ensembles we can now define the rate between state A and state B,  $k_{AB}$  as:

$$k_{AB} = \Phi_A P(\lambda_B | \lambda_0)$$

where  $\Phi_A$  is the flux (number of crossings per second) out of state A, which can be obtained from regular MD starting in state A, and  $P(\lambda_B | \lambda_0)$  is the

probability that a path crosses  $\lambda_B$  before  $\lambda_0$  after it crossed  $\lambda_0$  from state A. This probability is normally quite low, but with our ensembles we can reformulate this as:

$$P(\lambda_B|\lambda_0) = \prod_{i=0}^n P(\lambda_{i+1}|\lambda_i)$$

where  $\lambda_{n+1} = \lambda_B$ . So instead of waiting for the MD run to produce the event with small probability that 1 path completes the whole reaction, we multiply a couple of bigger probabilities that a path crosses one more interface than it is forced to do by the ensemble definition. Normally interfaces are placed such that this probability is about 20%.<sup>27</sup>

For example, if we sample 10 ensembles at a perfect placement (meaning that the total crossing probability would be  $0.2^{10} = 10^{-7}$ ) and in addition, suppose we want for each path ensemble  $i$  a 99% certainty that we sample at least 1 path that also crosses the next interface  $i + 1$ . In other words, we want to have less than a 1% chance that no trajectory crosses the next interface. In this case, we would need to sample in each path ensemble a number of paths  $n$  equal to:

$$\begin{aligned} 0.8^n &< 0.01 \\ n &> \log_{0.8}(0.01) \\ n &> \frac{\ln(0.01)}{\ln(0.8)} \\ n &> 20.6 \end{aligned}$$

Hence we would need  $n = 21$  trajectories per ensemble and 210 trajectories for our 10 ensembles. In this case the probability that we obtain a non-zero estimate of the crossing probability would be 90% ( $0.99^{10}$ ). Now we can compare this with the number of trajectories that we would need without using TIS, i. e. shooting off MD trajectories from  $\lambda_0$  and see how many cross  $\lambda_B$ . If we want a 90% chance to observe at least one trajectory crossing  $\lambda_B$  we would require  $\frac{\ln(1-0.90)}{\ln(1-10^{-7})} \approx 24$  million trajectories. Hence, TIS greatly enhances the speed at which we can obtain reasonable estimates of the reaction rate and other properties.

## 2.5 Replica Exchange Transition Interface Sampling

TIS theoretically samples the complete equilibrium distribution if the shooting moves are ergodic. However, as shown in paper E and B,

sampling can get stuck inside a single reaction channel while multiple exists. This is due to the fact that new trajectories are generated from previous ones, and swapping between reaction channels requires large movements in orthogonal reaction coordinates. To alleviate this problem and to further increase the sampling efficiency, replica exchange moves were added to TIS. This improved TIS algorithm is called Replica Exchange Transition Interface Sampling (RETIS).<sup>7</sup> With a replica exchange move it is attempted to swap the trajectories between two ensembles as follows:

1. choose two ensembles,  $i$  and  $j$
2. try to swap trajectories  $X_i$  and  $X_j$
3. accept the move if  $X_i$  is valid in ensemble  $j$  (crosses  $\lambda_j$ ) and  $X_j$  is valid in ensemble  $i$  (crosses  $\lambda_i$ ), else reject

These swapping moves are really cheap to perform, as they do not require any MD simulations.

As the probability that the trajectory from the inner interface crosses the outer interface drops significantly when interfaces are not next to each other ( $j \notin \{i - 1, i + 1\}$ ), default implementations in both OpenPath-Sampling<sup>8,9</sup> and PyRETIS<sup>10,11</sup> are to only attempt nearest neighbor swaps ( $j \in \{i - 1, i + 1\}$ )

To further improve the orthogonal sampling of the ensemble that has to cross  $\lambda_0$  ( $[0^+]$ ) a new ensemble is introduced, the  $[0^-]$  ensemble. The rules for a valid path in the  $[0^-]$  ensemble are:

1. starts with 1 frame outside of state A
2. crosses  $\lambda_0$  into state A at the second frame
3. ends with 1 frame outside of state A

This ensemble allows the simulation to also explore stable state A. Now, a swapping move between the  $[0^+]$  and  $[0^-]$  ensemble consists of the following steps, for the  $[0^+] \rightarrow [0^-]$  swap:

1. take the first two frames of the  $[0^+]$  ensemble (one on each side of  $\lambda_0$ ).

2. extend the path backwards (into state A) until  $\lambda_0$  is crossed again.

And for the  $[0^-] \rightarrow [0^+]$  swap:

1. take the last two frames of the  $[0^-]$  ensemble (one on each side of  $\lambda_0$ )
2. extend the path forward (out of state A) until  $\lambda_0$  is crossed again or  $\lambda_B$  (unlikely) is crossed.

With this selection strategy we have a 100% acceptance as a  $[0^+]$  path can always be integrated backward to produce a  $[0^-]$  path, but not always forward if it ends in state B. The reason why we don't randomly select one of the two possible crossing points for the  $[0^-] \rightarrow [0^+]$  swap and (commonly) two possible points for the  $[0^+] \rightarrow [0^-]$  is to prevent a  $P_{\text{acc}} \neq 1$  term if we start with an AB path in  $[0^+]$  and generate an AA path after the swap. This would waste the MD that is required for this swap. For the standard RETIS, either TIS shooting moves or replica exchange moves are chosen randomly, normally with a 50% probability each.

These swapping moves increase the sampling efficiency significantly as they provide, if accepted, a new sample to two ensembles without the MD cost (except for the  $[0^-] \leftrightarrow [0^+]$  swap). Also, having ensembles with a lower  $\lambda_i$  swap with ensembles with a higher  $\lambda_j$ ,  $j > i$ , gives a similar effect as swapping lower temperature and higher temperature replicas in a parallel tempering.<sup>28</sup>

The introduction of the  $[0^-]$  ensemble also gives an alternative way of computing the flux (in units of  $1/(\text{MD time step})$ ):

$$\Phi_A = \frac{1}{\langle \text{len}([0^+]) \rangle + \langle \text{len}([0^-]) \rangle - 4} \quad (2.13)$$

where  $\langle \text{len}([0^+]) \rangle$  is the average path-length in ensemble  $[0^+]$  and the  $-4$  is to correct for over counting.

There are still two limitations for standard RETIS: Firstly, while for TIS the ensembles can be simulated in parallel, the replica exchange moves of RETIS make this difficult and inefficient and both PyRETIS and OpenPathSampling implement it as a sequential algorithm. Secondly, we would like to do as many cheap swapping moves as possible, but that would still take a lot of time. Both of these limitations are solved in paper E.

This was a very focused introduction of a selection of path sampling algorithms, used in the papers. For a more complete overview of this field we advise to read ref. 17 or 29.

## 2.6 Decision Trees and random forests

In paper C we used machine learning algorithms to analyze the output of a RETIS simulation. The machine learning algorithms that we used were Decision Trees and Random Forests.

Decision trees<sup>30</sup> try make your data 'pure' as quickly as possible by asking a question and splitting the data into sets where that question is true and one set where the answer to that question is false. First we need a measure on how 'pure' our data is, we used information entropy for that were the information entropy is defined as:

$$\text{Entropy}(\mathbf{p}) = S(\mathbf{p}) = - \sum_{i=1}^K p_i \log_2(p_i)$$

where  $p_i = \frac{\text{\#objects in class } i}{\text{\# all objects}}$  for  $i \in 1, \dots, K$  classes. For RETIS we only have two classes, reactive paths (paths that go from state A to state B) and unreactive paths (paths that start and end in state A).

Next, we need a measure of how much information purity we can obtain from asking a certain question (for example 'is the distance between atom X and Y bigger than Z?'). For that we define a gain function  $G(D, Q)$  for the collection of data points,  $D$ , and the question  $Q$ :

$$G(D, Q) = S(D) - \sum_{a \in A(Q)} \frac{\#D_a}{\#D} S(D_a)$$

Where  $A(Q)$  is the set of answers a question can have (in our case it would be {True, False}) and  $\#D_a$  is the size of the subset of the data with that answer.

The Decision Tree then runs through all possible questions and selects the one that maximizes this gain function. Then it splits the data depending on the answer to that question and repeats this process on each subset until a certain depth or purity is reached. One small note: most implementations of Decision Trees don't run through all possible questions, but only a random subset.

One great feature of the Decision Trees is that they don't care if there are highly correlated or even duplicate variables (possible questions) in your data, as it will select 1 and never tries the others (as they will not give more information). We used this feature in paper C. For example in that paper, a single data point was a single frame, labeled if it came from a reactive path or an unreactive path, and the variables were all atom-atom distances. Each atom-atom distance occurred twice as we included both the atom1-atom2 and the atom2-atom1 distance, but only one of the copies was selected in the Decision Trees.

However, there is also one big issue with Decision Trees; it is a greedy algorithm and can thus be very sensitive to the initial split. In order to alleviate that, and get a more robust prediction in exchange for interpretability, you can use the Random Forests<sup>31</sup> (RF) algorithms. RF works by training a 'forest' of Decision Trees, but each of them only is allowed to choose questions from a random subset instead of all possible variables questions. A recent review for recent Decision Trees algorithms and applications can be found in ref. 32.



# 3 Accelerating Replica Exchange Transition Interface Sampling Simulations

While Replica Exchange Transition Interface Sampling (RETIS) is a lot more efficient than Molecular Dynamics (MD), it can still take months to generate good estimates for both the rate or any other property of the path ensembles. This is due to the fact that it is inherently a Monte-Carlo (MC) algorithm and, ideally, needs to sample a reasonable amount of path-space. It would thus be really nice to accelerate RETIS simulations and get our answers faster or get a better answer with the same amount of time.

The first way to get the answer faster is by accelerating the MD, as this is still the aspect on which the RETIS algorithm spends most of its time. This can be done by using more efficient MD libraries, or 'engines', like GROMACS,<sup>33</sup> LAMMPS,<sup>34</sup> and OpenMM.<sup>12</sup> Especially GROMACS and OpenMM can generate huge speedups if GPUs are available and leveraging them can generate 30-fold speedups.<sup>13</sup> Some care has to be taken on how one interacts with these MD engines and section 3.1 discusses how this is done in PyRETIS 2.

Another way to get the answer faster is by using the MD more effi-

ciently. This can be achieved with new MC moves, or by altering existing MC moves. The former was originally done with introducing the swapping moves, going from TIS to RETIS.<sup>7</sup> Since then other algorithms were introduced to enhance the shooting, the way we generate a new path from an old one, such as one-way shooting,<sup>35</sup> Precision Shooting,<sup>36</sup> Shooting from the Top,<sup>14</sup> Biased Shooting,<sup>37</sup> Aimless Shooting,<sup>38</sup> Permutation Shooting,<sup>39</sup> Web Throwing an Stone Skipping,<sup>15</sup> and Wire Fencing.<sup>16</sup> Section 3.2 discusses an idea to restrict the (possibly unbound) simulation time in the  $[0^-]$  ensemble and two new MC moves that were developed to speed up the sampling, one for processes that are symmetric and one for processes with multiple possible reactants. An example of a process in a system that is both symmetric and has multiple reactants is the permeation through a membrane.

A third way to accelerate is by using produced data more efficiently. Also this was done with the original development of RETIS, where the swapping move gives two ensembles each a new sample for a negligible amount of CPU-time compared to doing a shooting move. Of course we want to do as many of these swaps as possible, but if we attempt the same swap more than once we gain less information. Even worse, this information gain is 0 for most shooting algorithms, except for the high-accept moves in Stone Skipping, Web Throwing,<sup>15</sup> and Wire Fencing<sup>16</sup> where we need more than 1 swap to sample the right distribution. In practice a random chance of 50% to either perform a swapping move or a shooting moves is chosen to do as many swaps as possible without wasting CPU-time on moves that do not generate more information. Section 3.3 describes how we can use an infinite swapping approach<sup>18</sup> to do an infinite amount of swapping moves without using an infinite amount of time. We also show that the previous reported  $\mathcal{O}(N!)$  scaling for the general solution can be reduced to  $\mathcal{O}(2^N)$  by a reformulation of the problem into the computing of permanents. This can be further reduced to a  $\mathcal{O}(N^2)$  algorithm by leveraging a structure that occurs for most shooting algorithms.

One final way to accelerate requires a discussion about 'time'. Except the MD engines, all the previous ways to speed up the simulation are done by using the CPUs more efficiently, getting the answer in less CPU-time. This is the same as getting the answer faster in general: using less wall-time (time indicated by a clock on the wall). While CPUs are still get-

ting faster, even bigger speedups can be achieved by using multiple CPUs (or GPUs) at the same time with parallelization. GPUs specifically utilize this exactly method, consisting of many cores that are slower individually than the ones in your CPU, but can do many identical operations in parallel. Going from TIS to RETIS, the path sampling algorithm became more CPU-efficient, gain more information per CPU-time, but also lost the ability to run each ensemble independently in an embarrassingly parallel fashion. In section 3.4 the MC acceptance rules are rederived from an 'ensemble with a maybe interacting environment' view instead of the current 'superstate view'. This allows us to effectively parallelize the RETIS algorithm, and we see a  $\frac{N}{2}$  times speedup of the wall-efficiency with minimal reduction of the CPU-efficiency, where  $N$  is the number of ensembles in the simulation.

### 3.1 Leveraging MD engines

As said before, having MD run faster also means we get the RETIS answer faster. However, some care has to be taken when interacting with an MD engine. For RETIS there are several ways the algorithm has to interact with the MD engine:

- A snapshot to start simulating from needs to be extracted from a trajectory.
- (For certain shooting algorithms) The velocities of that snapshot need to be altered, which is not trivial when constraints are involved.
- The altered snapshot to start simulating from needs to be loaded back into the MD engine.
- The time direction has to be set (either forward or backward).
- Figure out when a simulation has entered a stable state.
- Stop the simulation whenever a simulation entered a stable state or reaches a maximum path length.

PyRETIS handles the time direction the same for each MD engine: it just reverses the velocities to go backward in time. This is the main reason

why symmetric integrators are required. The other interactions are handled differently depending on the MD engine.

For GROMACS most of the interactions are handled through files and the command line. The snapshot is extracted by using the provided 'trjconv' command. The velocities are altered by running a 0 fs timestep MD simulation with the option 'genvel' enabled in the .mdp file, which means only full randomization of velocities is supported. The altered snapshot is loaded back in with the provided 'grompp' command. The output files of GROMACS are read with MDTraj<sup>40</sup> to figure out when a stable state has been reached. The stopping of the simulation depends on the PyRETIS version used. The 'gromacs' engine introduced in PyRETIS 1 runs GROMACS 1 frame at a time, analyses that frame, and start the simulation for the next frame if it is not in the state. This is not an efficient way of running these simulations and in PyRETIS 2 the new 'gromacs2' engine was introduced. This new engine tells GROMACS to run a MD simulation until the maximum allowed path length that is pre-computed with eq 2.12. It then analyses the output trajectory file on-the-fly (while the simulation is still running) and kills the process (or process-group when GROMACS runs with MPI) when a stable state is hit. This method has a probability to 'overshoot' by more than 1 frame in a stable state, if the analysis is expensive to run. The resulting trajectory can be trimmed back to the correct size. While this might waste some MD, the second implementation is still significantly more efficient as the overhead for starting a GROMACS simulation is only encountered once.

For OpenMM the snapshot is extracted directly via the python interface, making sure to grab the positions, velocities, and box vectors. The velocities are altered inside of PyRETIS and a new snapshot is constructed with the altered velocities. The position, velocities and box vectors are loaded back into the OpenMM 'Context', without reconstructing (recompiling) it. This last bit is important as constructing a simulation context when the simulations are run on GPUs (a common use case for OpenMM) is really expensive to do, and should be avoided as much as possible. Then, the simulation is run through the Python interface of OpenMM, getting 1 frame back at a time, which is analyzed before a new computation is started. This is less bad than for the GROMACS simulation as most of the simulation (the Simulation Context) stays loaded. Still, future improvements could be made if the data generation and analysis is done in parallel

as was done for the 'gromacs2' engine.

Development or usage of any of the other engines included in PyRETIS are outside of the scope of this thesis. Examples that use these engines can be found on the PyRETIS website.<sup>41</sup>

## 3.2 Leveraging specialized MC moves

Having fast MD simulations helps with getting the answer faster, but using that MDs output efficiently is important as well. This was of course the main idea behind path-sampling and many further developments.

In the RETIS algorithm, MD in the  $[0^-]$  ensemble can take a long time, especially if there is no real confining barrier to the left of  $\lambda_A$ . This can result in simulations that wander really far from the barrier. To stop simulations from wandering away, a possibility was introduced in PyRETIS to also define a  $\lambda_{-1}$  interface to the left of  $\lambda_A$ . In the initial implementation, the MD simulation is stopped and the MC move is rejected as soon as it hits this 'left' interface in the  $[0^-]$  ensemble. This prevented these long trajectories, however if this happened more than a couple times per RETIS simulation, equation 2.13 can not be used anymore to compute the flux, as  $\langle \text{len}([0^-]) \rangle$  becomes inaccurate due to the negligence of the possible long trajectories. This version of  $\lambda_{-1}$  was included in PyRETIS 2 and can be seen in figure 1 of paper A.

The issue with the flux was partially solved later by altering the way  $[0^-]$  is sampled when  $\lambda_{-1}$  is defined. In the new sampling ensemble, named  $[0^-']$  paths are stopped when  $\lambda_{-1}$  or  $\lambda_A$  is hit. However, unlike the previous implementations, none of the paths are immediately rejected. Instead they are accepted with the probability:

$$P_{acc}^i(X^{(o)} \rightarrow X^{(n)}) = \min \left[ 1, \frac{h_{LR}(x_0^{(n)})h_{LR}(x_{L-1}^{(n)})(\text{len}(X^{(o)}) - 2)}{\text{len}(X^{(n)}) - 2} \right] \quad (3.1)$$

where  $h_{LR}$  indicates that this frame is in either in the Left state, left of  $\lambda_{-1}$ , or in the Right state, right of  $\lambda_A$ , and  $\text{len}(X)$  in the length of path  $X$  including the end-points as in chapter 2. The  $[0^+] \leftrightarrow [0^-]$  swap is automatically rejected if the current path in  $[0^-']$  does not end at the right interface (at  $\lambda_A$ ). Also, in the used flexible length algorithm,  $h_{LR}(x_{L-1}^{(n)})$  is only 0 if the path hits the maximum length criteria before hitting any of the 2 interfaces.

This allows a correction to the flux, which becomes

$$\Phi_A = \frac{\xi}{\xi(\langle \text{len}([0^+] \rangle) - 2) + \langle \text{len}([0^-] \rangle) - 2} \quad (3.2)$$

where  $\xi$  is,

$$\xi = \frac{N_{\rightarrow R, [0^-']}}{N_{[0^-']}} \quad (3.3)$$

with  $N_{\rightarrow R, [0^-']}$  being the number of paths in the new  $[0^-]$  ensemble that end at the right interface, and  $N_{[0^-]}$  is the total number of paths in  $[0^-]$ . The main idea of this  $[0^-]$  ensemble is illustrated in figure 2 of paper B. Note that  $\phi_A$  is a conditional flux: it is the flux of a permeant given this permeant is part of overall state A that excludes the phase space at the left of  $\lambda_{-1}$ . For a truly unbound system, the actual flux of a single particle would be 0. However, for properties that are concentration dependent, like permeation, the flux of *any* particle is not 0 as long as the concentration of the solute is constant and this new  $[0^-]$  ensemble allows for the computation of these concentration dependent properties with the help of a normalization. Also, for periodic systems, like permeation through a membrane, this  $\lambda_{-1}$  prevents sampling of 'reactive paths' that only occur due to particles jumping through the periodic boundary, as seen in figure 3 of paper B.

When investigating the problem of permeants permeating through a membrane, there are some other attributes of the system we can leverage with newly developed MC moves. First, if the system is periodic and statistically symmetric, then sampling should be identical going from right to left or left to right through the membrane. However the bi-layer of the membrane can be slightly different between these two options at a certain time. To sample both of these membrane configurations the 'mirror' move was implemented. It mirrors the collective variable (CV, as introduced in section 2.4), around an point halfway between  $\lambda_A$  and  $\lambda_{-1}$ , effectively swapping these two interfaces and moving all other interfaces. This only results in valid paths in the  $[0^-]$  ensemble and is thus only attempted in that ensemble. However, after that this mirrored CV can be propagated through the other ensembles via replica exchange moves. A different representation is shown in figure 5 of paper B, where the coordinates are mirrored instead of the CV. As mirroring coordinates of a periodic system is non-trivial, the implementation relies on mirroring the CV instead.

Another attribute of the permeant system is that there are multiple identical permeants attempting to permeate through the system at the same time, we leverage this with the 'target swap' move, illustrated in figure 4 in paper B. This move is only attempted in the  $[0^-]$ , as we expect negligible success in any of the other ensembles if the event under investigation is rare. It proceeds as follows:

1. For each frame in the old trajectory, count how many other permeants are between  $\lambda_{-1}$  and  $\lambda_A$ .
2. Sum all these options, which will be named  $Z_t^{(o \rightarrow n)}$
3. Pick any of the options (pick a random integer in  $[1, Z_t^{(o \rightarrow n)}]$ ).
4. With this permeant and time-frame selected, follow the path backwards until it crosses either of the interfaces. Here 'follow' means that we use frames from the old path until we run out, and only then generate more frames (if required). We call the initial time-frame  $j$  and the number of backward frames  $n_b$
5. Starting again at time  $j$  we now follow the path forward until either of the interfaces is crossed. The number of frames forward is called  $n_f$
6. With this new path repeat step 1 and 2 to compute  $Z_t^{(n \rightarrow o)}$
7. Compute the number of time slices that would give the same new trajectory from the old path,  $n_s^{(n)}$  (equation 3.4 below), and the number of time slices from which the old trajectory could be formed from the new trajectory,  $n_s^{(o)}$  (equation 3.5 below).
8. Accept the move with  $P_{\text{acc}} = \min \left( 1, \frac{n_s^{(o)} Z_t^{(o \rightarrow n)}}{n_s^{(n)} Z_t^{(n \rightarrow o)}} \right)$

With,

$$n_s^{(n)} = \min(j, n_b) + \min(L^{(o)} - j, n_f - 1) \quad (3.4)$$

and

$$n_s^{(o)} = \min(n_b + 1, j - 1) + \min(n_f, L^{(o)} - j - 1) \quad (3.5)$$

This move is summarized in figure 4 of paper B and just as with the mirror move this 'swapped' target can be propagated through the other

ensembles with replica exchange moves. These new MC moves, especially when combined, lead to a greatly improved sampling of degrees of freedom orthogonal to the  $\lambda$ -CV, as shown in figures 8 and 9 of the same paper. There it is shown that for a 2D system with 2 slightly different reaction channels, the simulation with the added MC moves converges much quicker to the right sampling ratio between the two channels as opposed to more standard RETIS.

### 3.3 Doing an infinite amount of swapping

In the previous section we accelerated the sampling and convergence of the TIS half of RETIS, this leaves the replica exchange part. Ideally we would like to do an infinite amount of replica exchanges (swaps) to make sure we add the path to as many valid ensembles as possible. While the swapping move is cheap compared to the MD, it is not truly 'free' and doing an infinite amount of them would take an infinite amount of time. Luckily, there are a finite number of path-ensemble combinations that can be sampled and directly sampling all the possible combinations in the right ratios would be sufficient. This is the main idea behind the infinite-swapping, and has been presented before in ref. 42 with the expected  $\mathcal{O}(N!)$  scaling. In this section we will present a fully general  $\mathcal{O}(2^N)$  scaling algorithm, and even faster (up to  $\mathcal{O}(N^2)$ ) algorithms that leverage some special properties in RETIS simulations based on shooting.

First some introductions of the used matrix representations. We start with 4 states ( $s_1, s_2, s_3, s_4$ ) with some weight for each of the 4 ensembles ( $e_1, e_2, e_3, e_4$ ),

$$W = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} & W_{44} \end{pmatrix} \end{matrix}$$

where  $W_{ij}$  is the weight of state  $i$  in ensemble  $j$ . And we want to compute the probability of each state for each ensemble after an infinite amount



of swaps:

$$P = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{pmatrix} \end{matrix}$$

where  $P_{ij}$  is the final probability of state  $i$  in ensemble  $j$ . Going from the  $W$  matrix to the  $P$  matrix is what we want to solve in this section.

The methods are best illustrated with an example. So let's start with defining an example  $W$  matrix,  $W_{ex}$ :

$$W_{ex} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} 2 & 1 & 0 & 0 \\ 5 & 4 & 3 & 0 \\ 8 & 7 & 6 & 0 \\ 12 & 11 & 10 & 9 \end{pmatrix} \end{matrix}$$

and we want to end up with the example  $P$  matrix,  $P_{ex}$ :

$$P_{ex} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} 15 & 9 & 0 & 0 \\ 5 & 8 & 11 & 0 \\ 4 & 7 & 13 & 0 \\ 0 & 0 & 0 & 24 \end{pmatrix} \frac{1}{24} \end{matrix}$$

We will now show how  $P_{ex}$  is obtained from  $W_{ex}$ , by showing that for  $W_{ex}$  we find that  $P_{11}$  is  $\frac{15}{24}$ , after which the method can be repeated for all other values of  $P_{ex}$ . A thing we can see from  $P_{ex}$  is that it is bistochastic: every column and row sum to the same number. This is generally true for all  $P$ -matrices. It is due to the fact that for every combination, every state is in exactly one ensemble and every ensemble contains exactly one state.

Let's start with the naive way of computing a P-matrix. First let's define a permutation set,  $C$ , which contains all permutations of distributing the four states over the four ensembles. We also define the subset,  $C_{s_1=e_1}$ , which contains all permutations in which state  $s_1$  is in ensemble  $e_1$ . For a single permutation, say  $(s_2, s_1, s_3, s_4)$  (meaning  $s_2$  is in  $e_1$ ,  $s_1$  in  $e_2$ ,  $s_3$  in  $e_3$  and  $s_4$  in  $e_4$ ) we also have a weight vector

$$w(s_2, s_1, s_3, s_4) = W_{21} \times W_{12} \times W_{33} \times W_{44}$$

which for our  $W_{ex}$  is:

$$w(s_2, s_1, s_3, s_4) = 5 \times 1 \times 6 \times 9 = 270$$

Then the probability of finding  $s_1$  in  $e_1$ ,  $P_{11}$  is computed as

$$P_{11} = \frac{\sum_{\sigma \in C_{s_1=e_1}} w(\sigma)}{\sum_{\sigma \in C} w(\sigma)}$$

where

$$\begin{aligned} \sum_{\sigma \in C_{s_1=e_1}} w(\sigma) &= w(s_1, s_2, s_3, s_4) + w(s_1, s_2, s_4, s_3) + w(s_1, s_3, s_2, s_4) + \\ &\quad w(s_1, s_3, s_4, s_2) + w(s_1, s_4, s_2, s_3) + w(s_1, s_4, s_3, s_2) \\ &= (2 \times 4 \times 6 \times 9) + (2 \times 4 \times 10 \times 0) + (2 \times 7 \times 3 \times 9) + \\ &\quad (2 \times 7 \times 10 \times 0) + (2 \times 11 \times 3 \times 0) + (2 \times 11 \times 6 \times 0) \\ &= 432 + 0 + 378 + 0 + 0 + 0 \\ &= 810 \end{aligned}$$

and

$$\sum_{\sigma \in C} w(\sigma) = 1296$$

so

$$P_{11} = \frac{810}{1296} = \frac{15}{24}$$

which is the result we expected. This can be repeated for each element in  $P$ , after computing  $w$  for each permutation. This algorithm scales as  $\mathcal{O}(N!)$ .

We can also compute this number in a different fashion, by computing permanents of the  $W$  matrix. A permanent is similar to a determinant, except that all signs are positive. For example,

$$\text{perm} \begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix} = A(E \times I + H \times F) + B(D \times I + F \times G) + C(D \times H + E \times I)$$

Just as with the determinant, it does not matter which row or column you choose for the expansion, as all of them lead to the same result.

Our definition for  $P_{ij}$  in terms of permanents becomes,

$$P_{ij} = \frac{W_{ij} \times \text{perm}(W_{\{ij\}})}{\text{perm}(W)} \quad (3.6)$$

where  $W_{\{ij\}}$  is the  $W$  matrix without row  $i$  and column  $j$ . For our example  $P_{11}$  is

$$\begin{aligned}
 P_{11} &= \frac{W_{11} \times \text{perm}(W_{\{11\}})}{\text{perm}(W)} = \frac{2 \times \text{perm} \begin{pmatrix} 4 & 3 & 0 \\ 7 & 6 & 0 \\ 11 & 10 & 9 \end{pmatrix}}{\text{perm} \begin{pmatrix} 2 & 1 & 0 & 0 \\ 5 & 4 & 3 & 0 \\ 8 & 7 & 6 & 0 \\ 12 & 11 & 10 & 9 \end{pmatrix}} \\
 &= \frac{2 \times 9(6 \times 4 + 7 \times 3)}{2 \times 9(6 \times 4 + 7 \times 3) + 1 \times 9(5 \times 6 + 3 \times 8)} = \frac{810}{1296} = \frac{15}{24}
 \end{aligned}$$

where we skipped all terms that would be 0. This leads to the same result as the naive method. This way of computing the permanent still requires  $\mathcal{O}(N!)$  operations, but the Balasubramanian–Bax–Franklin–Glynn (BBFG) formula<sup>43–46</sup> allows for the computing of the permanent with  $\mathcal{O}(2^N)$  operations, making the scaling of computing the  $P$  matrix scale as  $\mathcal{O}(N^2 2^N)$ . This is still a pretty bad scaling, but it actually makes infinite swapping feasible for RETIS simulations. If we want to compute the permanent with the naive method, we can compute up to  $N = 7$  in about 1 second on a mid-to-high-end laptop. The BBFG permanent method can do up to  $N = 12$  in 1 second. If we assume a typical RETIS simulation to have 20 ensembles and states, it would take 15 million years to compute the  $P$  matrix with the naive method, while with the BBFG permanent method it would only take 711 seconds.

This is the best we can do if we want to compute any  $P$  matrix from any  $W$  matrix, however, the  $W$  matrix that comes from RETIS simulations based on shooting has structures we can leverage to further increase the speed of this computation.

One nice feature is already shown in  $W_{ex}$ , most rows end with some zeros. For RETIS simulations this happens due to the way the ensembles are defined. We have a set of non-intersecting interfaces and every path (a sample for RETIS) that crosses an interface has to cross all inner interfaces. So it should have a non-zero value in the  $W$  matrix for each inner ensemble. If a path does not cross an interface it can't cross any of the ones after it. As soon as a weight in the  $W$ -matrix is 0 all the weights for ensembles with interfaces further out have to be 0 as well.

Permanents have the nice property that you can swap rows, without the permanent changing. This means we can sort the rows by the number of non-zero elements. We can then block-diagonalize the problems and solve each block independently. For  $W_{ex}$  we have two blocks, one  $3 \times 3$  highlighted in red, and a  $1 \times 1$  in blue:

$$W_{ex} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} \color{red}2 & \color{red}1 & \color{red}0 & 0 \\ \color{red}5 & \color{red}4 & \color{red}3 & 0 \\ \color{red}8 & \color{red}7 & \color{red}6 & 0 \\ 12 & 11 & 10 & \color{blue}9 \end{pmatrix} \end{matrix}$$

We then use equation 3.6, but only with the reduced block matrix instead of the full matrix. So  $P_{11}$  becomes:

$$\begin{aligned} P_{11} &= \frac{2 \times \text{perm} \begin{pmatrix} 4 & 3 \\ 7 & 6 \end{pmatrix}}{\text{perm} \begin{pmatrix} 2 & 1 & 0 \\ 5 & 4 & 3 \\ 8 & 7 & 6 \end{pmatrix}} \\ &= \frac{2 \times (6 \times 4 + 7 \times 3)}{2 \times (6 \times 4 + 7 \times 3) + 1 \times (5 \times 6 + 3 \times 8)} \\ &= \frac{90}{144} = \frac{15}{24} \end{aligned}$$

This allows us to split our  $12 \times 4 \times 4$  permutation computations into 8  $3 \times 3$  computations and a  $1 \times 1$  computation, which is much faster to compute. For a RETIS simulation, with a total of 20 interfaces an ideal interface placement would mean that each interface has a 20% probability of also crossing the next one<sup>47</sup>. With this ideal placement the probability of ending up with a block bigger than 12 (which takes 1 second) is  $5.73 \times 10^{-9}$ . For a non-optimal simulation with 20 interfaces, placed so that each interface has a 50% probability of crossing the next interface, the chance of getting a block that is bigger than  $12 \times 12$  is still less than 0.1%.

Another feature we can exploit for infinite swapping of RETIS simulations is that for most shooting algorithms the swapping weight is either 1 or 0. If this is the case we can fill the  $P$  matrix with a fast  $\mathcal{O}(N^2)$  algorithm instead. We go from the top-row to the bottom row to fill the  $P$  matrix

from a sorted  $W$  matrix with the following scheme:

$$P_{ij} = \begin{cases} 0, & \text{if } W_{ij} = 0 \\ \frac{1}{n_i+1-i}, & \text{if } W_{ij} = 1 \text{ and } [W_{(i-1)j} = 0 \text{ or } i = 1] \\ \binom{n_{i-1}-i+1}{n_i-i+1} P_{(i-1)j}, & \text{otherwise} \end{cases} \quad (3.7)$$

where  $i = 1$  is the top row, and  $n_i$  is the number of ones in row  $i$  of the  $W$  matrix.

For a second example,  $W_{ex2}$ , with zeroes and ones:

$$W_{ex2} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ s_1 & \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix} \\ s_2 & \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix} \\ s_3 & \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix} \\ s_4 & \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

In the first row  $n_1 = 2$ . Therefore the first row of our  $P_{ex2}$  becomes:

$$P_{ex2} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ s_1 & \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{pmatrix} \\ s_2 & \begin{pmatrix} ? & ? & ? & ? \end{pmatrix} \\ s_3 & \begin{pmatrix} ? & ? & ? & ? \end{pmatrix} \\ s_4 & \begin{pmatrix} ? & ? & ? & ? \end{pmatrix} \end{matrix}$$

where the non-zero values are computed using  $\frac{1}{n_i+1-i} = \frac{1}{2+1-1} = \frac{1}{2}$ . The following row, with  $n_2 = 3$ , then becomes

$$P_{ex2} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ s_1 & \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{pmatrix} \\ s_2 & \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 \end{pmatrix} \\ s_3 & \begin{pmatrix} ? & ? & ? & ? \end{pmatrix} \\ s_4 & \begin{pmatrix} ? & ? & ? & ? \end{pmatrix} \end{matrix}$$

where the third value is computed as above, while the first two values are computed using  $\binom{n_{i-1}-i+1}{n_i-i+1} P_{(i-1)j} = \binom{2-2+1}{3-2+1} \frac{1}{2} = \left(\frac{1}{2}\right) \frac{1}{2} = \frac{1}{4}$ . This can be continued and the final shape of the  $P_{ex2}$  becomes:

$$P_{ex2} = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ s_1 & \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{pmatrix} \\ s_2 & \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 \end{pmatrix} \\ s_3 & \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 \end{pmatrix} \\ s_4 & \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

This is an  $\mathcal{O}(N^2)$  algorithm and can be run under one second for up to  $N = 3500$ , which is way bigger than any foreseeable RETIS simulation.

All of these algorithms allow us to do an infinite amount of swaps in less than a second for typical RETIS simulations, perfectly spreading every data point over all relevant ensembles. This greatly increases the information gain of the swapping moves for a RETIS simulation and thus accelerates convergence of the simulation in general.

### 3.4 Parallelizing the RETIS algorithm

One last way of accelerating RETIS simulations is not by running the simulations more efficiently, but running them in parallel. For this we first have to discuss about the two different 'times' for simulations: The first one is wall-time, the amount of time that has passed on a clock on the wall during the simulation. The second one is CPU-time, the total amount of time that the CPUs have worked. If a simulation is run for 1 hour of wall-time on 2 CPUs in parallel it takes 2 hours of CPU-time. Up til now all improvements led to a speedup in both CPU-time and wall-time, but in this section we are going to accelerate our simulation in wall-time in exchange for a (possible) slow down in CPU-time. In other words we are going to get the answer faster (the time most PhD candidates care about), even if we potentially use the CPU resources less efficient.

When developing the RETIS algorithm from TIS, the algorithm became more efficient in CPU-time due to the introduction of the swapping move. However, it also requires communication between the ensembles. This is hard to parallelize as the time that each ensemble takes to finish a shooting move is considerably different (outer ensembles need longer paths to be generated) and even worse, this difference is not constant at all.

This means that while the TIS algorithm is embarrassingly parallelizable, the current released versions of OPS<sup>8,9</sup> and PyRETIS<sup>10,11</sup> implement the RETIS algorithm as fully sequential. For OPS a semi-parallel algorithm is proposed in ref. 17, where the faster ensemble has to wait for the slower ones to finish their move.

The following reformulation of the MC equations to allow for parallelization is described in relation to the RETIS algorithm, but is generally useful for any type of Monte-Carlo based replica exchange where each of

the ensembles can take a significant different amount of time to generate new samples, such as for configurational bias MC<sup>23,48,49</sup>, cluster MC algorithms<sup>50</sup>, and event-chain MC<sup>51,52</sup>.

If we start from equation 2.3:

$$P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)}) = \min \left[ 1, \frac{\rho(S^{(n)})}{\rho(S^{(o)})} \right] \quad (3.8)$$

then normally in RETIS  $S = (s_1, s_2, \dots, s_N)$  is a superstate of all ensembles at a certain MC step and the probability of that superstate is

$$\rho(S) = \rho(s_1, s_2, \dots, s_N) = \prod_{i=1}^N \rho_i(s_i) \quad (3.9)$$

where  $p_i(\cdot)$  is the probability density of ensemble  $i$ . For a RETIS swapping move that swaps the first two states, indicating  $S^{(o)} = (s_1, s_2, \dots, s_N)$  and  $S^{(n)} = (s_2, s_1, \dots, s_N)$ , the acceptance probability becomes

$$P_{\text{acc}} = \min \left[ 1, \frac{\rho_1(s_2)\rho_2(s_1)}{\rho_1(s_1)\rho_2(s_2)} \right] \quad (3.10)$$

and for a shooting move,  $S^{(o)} = (s_1^{(o)}, s_2, \dots, s_N)$  and  $S^{(n)} = (s_1^{(n)}, s_2, \dots, s_N)$  the acceptance probability via eq 2.2 becomes

$$P_{\text{acc}}(s_1^{(o)} \rightarrow s_1^{(n)}) = \min \left[ 1, \frac{\rho_1(s_1^{(n)})P_{\text{gen}}(s_1^{(n)} \rightarrow s_1^{(o)})}{\rho_1(s_1^{(o)})P_{\text{gen}}(s_1^{(o)} \rightarrow s_1^{(n)})} \right] \quad (3.11)$$

Due to this superstate view and equations 3.8 and 3.9, all ensembles need to know their state (have finished their previous moves) before these moves can be accepted, leading to the sequential RETIS algorithms.

For the new derivation we instead start from a view from a single ensemble (e.g.  $e_1$ ), and we can view states in all the other ensembles as an 'environment' ( $\mathcal{E} = (s_2, \dots, s_N)$ ). One important point we need to consider is that the environment can change while we are doing a MC move, as would happen when we run this system in parallel.

We can then write the probability of state  $s_1$  in ensemble 1 as an integral of the probability given a certain environment

$$\rho_1(s_1) = \int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})d\mathcal{E}. \quad (3.12)$$

and for a single ensemble move, like shooting from  $s_1$  to  $s'_1$ , detailed balance equation 2.1 becomes a twisted detailed-balance equation

$$\rho(s_1|\mathcal{E})\pi_1(s_1, \mathcal{E} \rightarrow s'_1, {}^a\mathcal{E}) = \rho(s'_1|\mathcal{E})\pi_1(s'_1, \mathcal{E} \rightarrow s_1, {}^a\mathcal{E}) \quad (3.13)$$

where  ${}^a\mathcal{E}$  is *any* environment. We called this a twisted detailed balance equation as there is one term that moves in the same direction between the left and right side of the equation,  $\mathcal{E} \rightarrow {}^a\mathcal{E}$ , and one term that becomes twisted,  $s_1 \rightarrow s'_1$  becomes  $s'_1 \rightarrow s_1$ . The complete derivation of this term is shown in the SI of paper E.

Now, because the ensembles are independent

$$\rho_1(s_1|\mathcal{E}) = \rho_1(s_1), \quad (3.14)$$

and because all ensembles progress independently from each other

$$\pi_1(s_1, \mathcal{E} \rightarrow s'_1, {}^a\mathcal{E}) = \pi_1(s_1 \rightarrow s'_1)\pi_1(\mathcal{E} \rightarrow {}^a\mathcal{E}) \quad (3.15)$$

Substituting eq. 3.15 into eq. 3.13, the second  $\pi(\cdot)$  term on each side cancels as  $\pi_1(\mathcal{E} \rightarrow {}^a\mathcal{E})$  appears on both sides of the equation (and because the probability of going from an environment to *any* environment is 1). If we furthermore apply eq 3.14, we end up with

$$\rho_1(s_1)\pi_1(s_1 \rightarrow s'_1) = \rho_1(s'_1)\pi_1(s'_1 \rightarrow s_1) \quad (3.16)$$

which is identical to eq 2.1 and leads to the same acceptance probability as eq 3.11 without relying on the superstate description. For the case that shooting is the only move being used, we already knew that this was allowed due to TIS being an embarrassingly parallel algorithm, where we can run each ensemble independently. However, the twisted detailed-balance allows us to prove that the shooting is still correct even if it is only one of the possible MC moves that can be selected, like in RETIS with the addition of swapping moves (see the SI of paper E).

We can reformulate the swapping move (e.g.  $1 \leftrightarrow 2$ ) in a similar fashion and end up with a similar twisted detailed-balance equation

$$\begin{aligned} \rho_1(s_1|s_2, \mathcal{E}_{\setminus 2})\rho_2(s_2)\hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_{\setminus 2} \rightarrow s_2, s_1, {}^a\mathcal{E}_{\setminus 2}) = \\ \rho_1(s_2|s_1, \mathcal{E}_{\setminus 2})\rho_2(s_1)\hat{\pi}_{1\leftrightarrow 2}(s_2, s_1, \mathcal{E}_{\setminus 2} \rightarrow s_1, s_2, {}^a\mathcal{E}_{\setminus 2}) \end{aligned} \quad (3.17)$$

where  $\mathcal{E}_{\setminus 2}$  is the environment without ensemble 1 and 2 and we use  $\hat{\pi}$  instead of  $\pi$  to indicate that we only look at possible swaps, and not *any* 2



ensemble move. Then using equations 3.14 and 3.15 and the same cancellation as for the shooting move, we end up with

$$\rho_1(s_1)\rho_2(s_2)\hat{\pi}_{1\leftrightarrow 2}(s_1, s_2 \rightarrow s_2, s_1) = \rho_1(s_2)\rho_2(s_1)\hat{\pi}_{1\leftrightarrow 2}(s_2, s_1 \rightarrow s_1, s_2) \quad (3.18)$$

Furthermore,

$$\hat{\pi}_{1\leftrightarrow 2}(s_1, s_2 \rightarrow s_2, s_1) = P_{\text{acc}}(s_1, s_2 \rightarrow s_2, s_1)P_{\text{gen}}(s_1, s_2 \rightarrow s_2, s_1) \quad (3.19)$$

and

$$\hat{\pi}_{2\leftrightarrow 1}(s_2, s_1 \rightarrow s_1, s_2) = P_{\text{acc}}(s_2, s_1 \rightarrow s_1, s_2)P_{\text{gen}}(s_2, s_1 \rightarrow s_1, s_2) \quad (3.20)$$

If the swap is allowed, converting  $(s_1, s_2)$  to  $(s_2, s_1)$  is the only possible result. Therefore,  $P_{\text{gen}}(s_1, s_2 \rightarrow s_2, s_1)$  and  $P_{\text{gen}}(s_2, s_1 \rightarrow s_1, s_2)$  are both 1, canceling out. We thus end up with

$$\rho_1(s_1)\rho_2(s_2)P_{\text{acc}}(s_1, s_2 \rightarrow s_2, s_1) = \rho_1(s_2)\rho_2(s_1)P_{\text{acc}}(s_2, s_1 \rightarrow s_1, s_2) \quad (3.21)$$

which can be satisfied with equation 3.10 as before. For the completed and more detailed derivation, please have a look at Sec. II in the SI of paper E.

With these new derivations we show that we can use the standard formulas to perform swapping and shooting in parallel, as the environment (all other ensembles) is now allowed to change while we perform these moves. It does however lead to a different amount of samples in each ensemble, which requires attention during analysis. In paper E we applied this together with the infinite swapping, coined  $\infty$ RETIS, on 3 model systems with a changing number of workers, number of ensembles simulated in parallel. If the number of workers is 1, this algorithm performs as the standard RETIS algorithm (with infinite swapping) and if the number of workers is identical to the number of ensembles, this algorithm is identical to TIS as there is just 1 ensemble not doing a MC move at a given time, so no swapping can happen. In figure 1 of that paper we also see that we get the expected linear scaling of MD time, while having a more than linear scaling of amount of MC moves done per 12 hours. It also shows that we accelerated the wall-time speed by a 50 fold for one of our test systems, and up to a 10 fold increase for another. Both with minimal reduction in our CPU-time efficiency.

Keep in mind that this new way of deriving the detailed balance equations is not limited to RETIS simulations and with the continuing trend to

run more and more massively-parallel computing jobs we expect this new algorithm to gain importance and that it can open up new avenues in the field of molecular simulations and maybe even beyond.

## 4 Accelerating the Analysis

With the current speed of path sampling simulations we can generate thousands of reaction paths in a couple months. It is common to investigate these by viewing the trajectories and cleverly plotting some histograms, as in paper F. Both the running of rare-event simulations and the analysis require a lot of time and multiple groups have significant contributions using Machine Learning (ML) algorithms to accelerate this.<sup>53-59</sup>

ML algorithms are normally divided into two categories, unsupervised and supervised algorithms. Unsupervised algorithms try to uncover patterns in the data without user supervision, which means that the data does not have to be identified beforehand. This results into a clustering of your data or a dimensionality reduction, and examples of unsupervised methods are Principle Component Analysis (PCA),<sup>60</sup> hierarchical clustering,<sup>61</sup> k-means clustering,<sup>62</sup> and Isometric Mapping.<sup>63</sup>

The other category is supervised ML algorithms, in which an expert has to label (a subset of) the data with either a class or a value that we want the algorithm learn. Supervised algorithms result into a model that can do predictions after training on the labeled subset of the data. Examples of supervised algorithms are decision trees,<sup>30</sup> random forests,<sup>31</sup> neural networks,<sup>64</sup> and gradient boosting.<sup>65</sup>

Other than the choice of ML algorithm that we use to solve our problem, an important decision has to be made on the structure of the data that we use to train our algorithms on. For the analysis of rare-event simulations multiple different algorithms have been used, but most of

them are only trained on a curated data representation of just a few key variables.<sup>54,57,58</sup> This curating has to be done by the researcher and has the risk of introducing a hypothesis bias, where you only do analysis of the key variables that you expect to be important beforehand, potentially completely missing a more important variable that you did not think of. Section 4.1 describes the newly developed completely generic data representation for molecular configurations that is translational, rotational, and atom-index invariant. It only requires minimal user input and greatly reduces the chance of hypothesis bias in the data representation that is used to train the ML algorithms.

After a ML model is trained we would also like to understand 'why' it does certain predictions. For black-box models, like neural networks, and gradient boosting, you can use Shapley values<sup>66</sup> or counterfactuals<sup>67</sup> to try to learn how the trained model is working. Shapley values show how much each variable contribute to the predicted outcome, while counterfactuals show how much a variable needs to change before we predict a different class or outcome. Another option is to use so-called human understandable algorithms, where after the training of the model we can understand the logic that is used to make decisions. One of such algorithms is decision trees (DTs). DTs has its issues and as mentioned in section 2.6 a big issue with DTs is that they can be very dependent on the initial split. In section 4.2 we show how DTs can help us to understand why a chemical reaction is triggered, together with a way to investigate how important different initial splits are.

## 4.1 Translational, rotational, and index invariant data representation

As mentioned in the introduction of this chapter, in order to start to use ML algorithms on our data we first have to decide the shape of our data. For molecular simulations we normally start with the element, x,y,z coordinates and an index for every atom in your simulation box. However, there are some issues with just feeding this into your ML algorithm. Ideally we would like systems that behave identical to have identical data-representations, but as you can see in figure 4.1, this is not the case for the xyz format.

x	y	z	x	y	z
0.0435	0.2326	0.0471	0.0435	0.0471	-0.2326
0.0619	0.3386	0.0742	0.0619	0.0742	-0.3386
0.0157	0.1488	0.1355	0.0157	0.1355	-0.1488
0.0552	0.2129	-0.0830	0.0552	-0.0830	-0.2129
0.0380	0.1114	-0.1131	0.0380	-0.1131	-0.1114



Figure 4.1: The xyz data and structure for a formic acid molecule, one rotated 90 degrees with respect to the other.

The system is just rotated 90 degrees, so the physics should be identical, but in the data the y and z columns are swapped and the z column is multiplied by  $-1$ . A similar shift in data would occur if we just move the origin of our data, while the physics does not change. One option to let your ML algorithm learn how to deal with all the rotations is by generating all possible rotations for each of your data points, the other is using a translational and rotational invariant representation. In paper C we use a distance-distance matrix, which is identical for any rotation as shown in figure 4.2.

However, for molecular systems the distance-distance matrix still has one issue, shown in figure 1 of the SI of paper C. It is not atom-index invariant. If we swap two atoms of the same element, the physics should not change, but the data representation does; two rows and columns swap places. In order to circumvent this problem we introduced a sorted-distance matrix that is shown in figure 2 in paper C. The general idea is as follows:

1. Start with the distance-distance matrix
2. group all atoms with the same element together

	C0	H0	O0	O1	H1
C0	0.0000	0.1109	0.1250	0.1321	0.2010
H0	0.1109	0.0000	0.2047	0.2013	0.2954
O0	0.1250	0.2047	0.0000	0.2311	0.2524
O1	0.1321	0.2013	0.2311	0.0000	0.1073
H1	0.2010	0.2954	0.2524	0.1073	0.0000



Figure 4.2: The distance distance matrix for both of the formic acid molecules shown.

3. One element is chosen as the anchor element, and is put in the top-left corner of our matrix (carbon in our example system)
4. For every time slice one atom from the anchor element is chosen as the anchor atom, this can be a different atom for each frame.(in our example system there is only one carbon to choose)
5. The rows within each element group are ordered based on the distance from the anchor atom.
6. For each row the columns in each element block are sorted based on their value.

After this we end up with a non-symmetric matrix that is invariant to translation, rotation and atom index. How well this representation performs depends on how stable the representation is, which largely depends on how the anchor atom is selected. For path sampling simulations this should not be too hard to determine as there is always an atom or a group of atoms that is tracked to determine in which stable state the system is and one of those atoms can be used as the anchor. This does not mean a single atom has to be traced. For example, in reference 68 the collective

variable was the maximum OH bond of *all* OH-bonds, and either the O or H of this maximum can be selected as the anchor atom.

One issue with using the sorted distance-distance matrix as a data representation is that it can be hard to determine what each variable (a value in our matrix) actually represents, especially if a different anchor atom is chosen in each frame, possible swapping the specific atoms that represent each row.

Luckily, we can regenerate the xyz format from our sorted-distance matrix, up to an arbitrary index-swap, translation and rotation, and in such a way that we can highlight the atoms that are representing selected atom-atom distance.

For this back-mapping of our sorted distance-distance matrix to the xyz format, we first 'unsort' the matrix from bottom to up back into the symmetric matrix. Next we adapt the procedure of reference 69, to map an arbitrary matrix into a  $N$ -dimensional space.

From the symmetric distance-distance matrix,  $D$ , we can construct

$$M_{ij} = \frac{D_{1j}^2 + D_{i1}^2 - D_{ij}^2}{2} \quad (4.1)$$

where  $D_{1j}$  is the  $j$ -th element of the first row and  $D_{i1}$  is the  $i$ th element of the first column. Then an eigenvalue decomposition on  $M$  is performed:  $M = USU^T$ , where  $U$  is a  $N \times N$  matrix where the columns are the eigenvectors of  $M$  and  $S$  is a diagonal matrix with the eigenvalues of  $M$ . This allows for the computing of the matrix  $X = U\sqrt{S}$ . In general, if the matrix is mappable in  $N$  dimensional space, only the first  $N$  eigenvalues ( $S$ ) are non-zero. We know our matrix was generated from 3D space, and should be able to be mapped into 3 dimensions (every atom position can be uniquely defined with just 3 variables; x, y, and z). Therefore, we only take the first 3 columns of  $X$ , which become our x, y, and z columns. This back-mapping was used in paper C where the molecular movies were viewed, but with the distances highlighted that were selected by the ML algorithm. This helped with an more data driven analysis on what triggers the deprotonation of formic acid in small water droplets.

This ability to back-map is also the main improvement over the commonly used symmetry functions from Behler and Parrinello, which handles the atom-index variance by smearing all atoms of the same element together for every single time slice. This method is very useful for the

development of ML based force fields, but can be hard to interpret if we use it for ML-based analysis.<sup>70</sup>

## 4.2 Using human understandable ML algorithms on RETIS data

If we want to investigate what triggers a chemical reaction, RETIS simulations give a nice data set to train our ML algorithm on. Namely, it is a set of paths that start a reaction, but don't manage to cross the barrier (A-A paths), and paths that complete the reaction (A-B paths). We can then use a supervised ML algorithm, give it a frame that is close to the reactant state and ask it to predict if this frame is from a reactive or unreactive path. This forces the ML algorithm to find differences between the reactive and unreactive paths. If we then can understand what the ML algorithm uses to differentiate between these two options, we can learn what makes reactive paths different, and thus what triggers a reaction.

There are a couple issues that our ML algorithm should be able to handle. First, as we are looking at rare events it should work on heavily skewed data-sets. For the test system in paper C of Formic Acid (FA) in water, the probability of a reactive path is only  $10^{-3.745} = 0.02\%$ , so if our ML algorithm only chooses 'unreactive' it is correct 99.98% of the time. Secondly, as we would like to know how the trained algorithm is differentiating between reactive and unreactive paths it should be human-understandable. Lastly, as we use the data-representation described in the previous section, it should be able to handle multiple identical variables being present, due to the fact that every distance occurs twice in the distance-distance matrix.

Luckily there is a type of ML algorithm that handles all three of these issues, Decision Trees (DTs).<sup>30</sup> As described in section 2.6, DTs don't optimize the accuracy directly, but the purity of the subsets. This handles the first and last issue, as the purity is independent of the skewness of the data set, and no further purity increase can be achieved with the second copy of the variable after the first one has been selected. DTs are also interpretable, as long as the input variables are interpretable and the depth is limited. Our purely distance based input variables are interpretable, especially when combined with the back-mapping and visual inspection. In



paper C we settled on a maximum depth of 3 for the interpretability. Example trees with highlighted important distances can be seen in figure 4 and 6 of paper C.

As previously mentioned in section 2.6 a drawback of DTs is that they are greedy algorithms, and can depend heavily on the initial split. We therefore developed a method to estimate the true probability that the first variable is truly the most important, and which other variables are important. It starts with training a random forest (RF)<sup>31</sup> with trees of depth 1 (as we are only interested in the initial split), together with a block-error calculation from 10 blocks. This gives us importance averages and variances, as can be seen in figure 5 and 7 in paper C.

Then, we assume for each variable that this importance is a Gaussian distribution around the reported average with a  $\sigma$  of the reported variance. From each of these distributions, we take a single random sample and count which variable ended up with the highest importance. This is repeated ten-thousand times to give an estimate of how often each variable was the most important variable to split on.

In paper C these methods were used successfully to investigate the deprotonation of FA inside small water droplets. Our product state was achieved when the proton moved more than 3.0 Å from any FA oxygen, meaning that there was at least 1 water molecule in between the proton and FA. For FA with 4 water molecules we saw that it was important that a water 2 hydrogen bonds away from FA was close enough to the water that was going to accept the proton from FA to prevent immediate reprotonation, which would be classified as an AA-path in our data. For FA in 6 waters we saw a similar effect, but also that multiple pathways were possible and important. The availability of multiple possible deprotonation pathways gave a reasonable explanation for the observed increase of the deprotonation rate for FA in the 6 water system.



# 5 Applications to biochemical problems

In this chapter applications are discussed in which simulations were applied to investigate biochemical problems. The first section shows how multiple state transition path sampling (MSTPS) was used to investigate the configurational movements of the protein KRas and how they change when an oncogenic mutation, Q61L, is introduced. The second section shows how rates, either from literature or computed with a RETIS simulation, can be used to simulate ideal reactions. These were used to investigate the limits of analysis formulas for covalent inhibition assays, and distributed as an independent tool for experimentalists to validate that the rates they find correspond with the signal they observe.

## 5.1 Conformational changes of KRas

KRas is a GTPase which signals for cell growth and division. It has an active state in which two flexible loop regions (S1 and S2) are tightly bound to GTP and an inactive state in which S1 and S2 are not connected and unordered. Both the structure of KRas and a schematic representation of the two states can be seen in figure 1 of paper F. It has multiple possible mutations that alter the transition between this active and inactive state, which can result in KRas being active for longer times. As active KRas triggers cell growth and division, mutations of KRas, and the whole RAS family of proteins, are frequently found inside cancer cells.<sup>71</sup> To under-

stand this transition we applied transition path sampling (TPS) on KRas and an oncogenic mutant, Q61L. This mutation changes a glutamine(Q) into a leucine (L) in the S2 loop.

During an initial simulation, we found that there are actually three stable states for both S1 and S2, which were identical for both the wild-type (WT) and Q61L. For S1 they are: the native bound state, where S1 is bound to GTP as in the crystal structure, the non-native bound state, in which a different part of S1 is attached to GTP, and the unbound state in which S1 is disconnected and solvated. For S2 the found stable states are: the tightly bound state where S2 is attached to GTP, the open state in which S2 is solvated, and a third state in which S2 moved away from GTP, but stays attached to the  $\alpha$ 3-helix. All of these stable states can be seen in figure 2 of paper F.

Two state TPS simulations are very useful for chemical reactions, but like KRas a lot of biochemical reactions involve more than two (semi-)stable states. TPS had previously been extended to sample more than two stable states at the same time, called multiple state TPS (MSTPS)<sup>72</sup> and we used MSTPS to further study the dynamics of KRas.

With this extended method we did not see any differences in the sampling behavior of S1, shown in the top of figure 3 and figure 4 of paper F, and because this mutation is in S2 and therefore has the most influence on the dynamics of S2, we focused the rest of our work on the transitions of S2.

Initially we did not observe any difference in the sampling of the different transitions, as can be seen in the bottom row of figure 3 of paper F. However, one effect of the extension from TPS to MSTPS together with one-way shooting is that the simulation can 'switch' between different transitions by changing either the beginning state, for a backwards shot, or the ending state, with a forward shot. We can even go from an  $A \rightarrow B$  transition to a  $B \rightarrow A$  transition, via  $A \rightarrow C$  and  $B \rightarrow C$ . A schematic overview of the switching transitions are shown in figure 8 in paper F.

We also analyzed this switching behavior for WT and Q61L, shown in figure 4 of paper F, which shows a stark difference. This would indicate that Q61L has a lot more trouble switching between transitions than WT. With the assumption that these switches also sample an equilibrium distribution, we can quantify this difference, as shown in figure 5 of paper F.

From the blue arrows in that figure we see that every single switch happens less frequently for Q61L than for WT.

Further mechanistic investigation of the MSTPS simulations then revealed the difference of this switching behavior, shown in figure 6 of paper F.

For the WT there are two possible transition channels to go to the open state, one by direct solvation and one where S2 slides along a hydrophobic pocket on the  $\alpha$ 3-helix into the open state. For Q61L the direct solvation channel disappears, and it binds closer to the hydrophobic pocket of the  $\alpha$ 3-helix. This is reasonable and was to be expected by swapping the hydrophilic glutamine for a hydrophobic leucine. So while the individual stable states and transitions did not (seem) to change between WT and Q61L, MSTPS still alerted us to a potential with the different switching behavior. This might have been missed with 3 independent (regular, non multiple state) TPS simulations.

## 5.2 Using kinetic simulations for illustrating Covalent Inhibition in Enzymatic Assays

If we have reaction rates, either from a RETIS simulation or from experiments, then we can simulate the kinetics of reactions under certain conditions. In paper D we use these simulations to investigate assay conditions for enzymatic assays with covalent inhibitors.

Covalent inhibitors bind strongly with a covalent bond to their target. However, traditional drug design focused mostly on molecules that bind noncovalently to their target, in a reversible manner. Due to the strong binding of covalent inhibitors, they also have long lasting side effects if they bind to the wrong target. These off-target effects were often not discovered until late-stage clinical trials, and drug discovery programs were moving away from covalent inhibitors.<sup>73</sup> However, successful covalent inhibitors are used widely, well before their covalent mechanism was discovered, such as aspirin and penicillin. More recent successes of targeted covalent inhibitors (TCIs) triggered a resurgence in the use of covalent inhibitors in drug discovery programs.<sup>74</sup>

An overview of experimental methods suitable for kinetic evaluation structure-activity relationship (SAR) of covalent inhibitors with their co-

valent binding mode was missing and paper D fits into that gap.

When preparing for that manuscript we discovered that validating the different experimental methods and analysis methods and their (implicit) assumptions by performing the actual experiments was very costly. Also if we did not obtain a reasonable result, we could not differentiate between experimental artifacts, enzyme degradation or broken assumptions of the used analysis.

To circumvent these issues we instead decided to simulate these reactions by direct numerical integration of the rate equations. This allowed us to purely investigate the effects of breaking certain assumptions, and selectively turn dilution effects and enzyme degradation on or off.

During the development of the manuscript we realized that these simulations would also be useful for readers to check and validate their experiments and that it could work as an educational tool to investigate what would happen by varying parameters. Therefore we decided to make them publicly available, runnable through a browser using mybinder.org,<sup>1</sup> at [tinyurl.com/kineticsimulations](https://tinyurl.com/kineticsimulations).

## 6 Conclusion and outlook

In this thesis, several improvements to all stages of path sampling simulations have been presented, increasing the speed at which we can investigate chemistry with these methods. We started by explaining how to interface path sampling code with external Molecular Dynamics (MD) codes that use the increased computing power of GPUs. This greatly improved the amount of MD we can run per hour. Further development in this direction would make the RETIS algorithm even more widely applicable, for example by interfacing RETIS with other specialist code bases, such as the coupled-cluster code eT.<sup>75</sup>

Next, we introduced new sampling moves for the RETIS algorithm. These steered the MD more efficiently to the region of interest. It showed a great increase of the sampling of orthogonal collective variables in a notorious test system when compared to the regular implementations. Using the MD more efficiently is still an active field of research and will probably stay that way. Further improvements are expected from advanced fast decorrelating MC moves like the wire fencing move.<sup>16</sup>

The last improvement in this thesis for the RETIS algorithm was enabling both parallelization and infinite swapping. The infinite swapping greatly enhances the efficiency of the swapping, while the parallelization allows for great scaling with the number of processes available. The exact scaling of the parallelization depends on the correlation of the sampling. It has perfect linear scaling until one worker per ensemble if there is no correlation between two consecutive samples. If there is some correlation between two consecutive samples, like most real simulations, the

efficiency has an optimum. A safe bet for optimal efficiency is about  $\frac{1}{2}$  a worker per ensemble, as long as the average time per Monte Carlo (MC) move is more than 1 s.

This change in algorithm does break a lot of assumptions in both common open source path sampling codes, OpenPathSampling and PyRETIS. Therefore it is required to either convert one of these softwares or develop a new one depending on these new algorithms in future work. Also, these algorithms should be even better for real systems as opposed to the used test cases as a single MC move in a realistic high-dimensional system is in the order of minutes, not seconds. It would be great to see  $\infty$ RETIS be applied to biological systems.

The analysis of a RETIS simulation was also improved by using human understandable machine learning, together with a data representation that is invariant to translation, rotation and atom-index. This data representation together with the machine learning allowed for a speed up of the analysis, without the issue of a potential hypothesis bias that is common in other works. This was then used to elucidate the difference in mechanisms of deprotonation of Formic Acid with either 4 or 6 water molecules. We used Decision Trees (DTs) for our algorithm with the standard information entropy splitting. However, a different information metric for RETIS simulations, called Predictive Power,<sup>76</sup> has been presented before and making a DT algorithm based on this would be an interesting direction for future work. Another direction for future work would be the extension of the data representation with bond-graph information, by mapping the 2D matrix into a 3D sparse matrix.

We also showed that path sampling simulations can be used for biomolecular systems and either enhancing that work with multiple state transition interface sampling or investigating more mutants would be interesting.

Lastly, we showed that relatively simple simulations can be instrumental for understanding and teaching experimental setups especially when provided in a way that 'just works' in students' browser. The setup via Binder<sup>1</sup> could possibly be used for hands on teaching and it also perfectly fits for non teaching focused manuscripts with the current focus of the academic community of data reproducibility.



# Bibliography

- [1] Project Jupyter; Matthias Bussonnier; Jessica Forde; Jeremy Freeman; Brian Granger; Tim Head; Chris Holdgraf; Kyle Kelley; Gladys Nalvarte; Andrew Osheroﬀ; Pacer, M.; Yuvi Panda; Fernando Perez; Benjamin Ragan Kelley; Carol Willing In *Proceedings of the 17th Python in Science Conference*; pp 113 – 120.
- [2] Shaw, D. E.; et al. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*; Association for Computing Machinery: New York, NY, USA; SC '21. <https://doi.org/10.1145/3458817.3487397>.
- [3] Dellago, C.; Bolhuis, P. G.; Chandler, D. “Efficient transition path sampling: Application to Lennard-Jones cluster rearrangements”, *The Journal of Chemical Physics* **1998**, 108, 9236–9245.
- [4] Arjun; Berendsen, T. A.; Bolhuis, P. G. “Unbiased atomistic insight in the competing nucleation mechanisms of methane hydrates”, *Proceedings of the National Academy of Sciences* **2019**, 116, 19305–19310.
- [5] Dellago, C.; Bolhuis, P. G.; Chandler, D. “On the calculation of reaction rate constants in the transition path ensemble”, *The Journal of Chemical Physics* **1999**, 110, 6617–6625.
- [6] van Erp, T. S.; Moroni, D.; Bolhuis, P. G. “A novel path sampling method for the calculation of rate constants”, *The Journal of Chemical Physics* **2003**, 118, 7762–7774.
- [7] van Erp, T. S. “Reaction Rate Calculation by Parallel Path Swapping”, *Physical Review Letters* **2007**, 98, 268301.

- [8] Swenson, D. W. H.; Prinz, J.-H.; Noe, F.; Chodera, J. D.; Bolhuis, P. G. "OpenPathSampling: A Python Framework for Path Sampling Simulations. 1. Basics", *Journal of Chemical Theory and Computation* **2019**, 15, 813–836.
- [9] Swenson, D. W. H.; Prinz, J.-H.; Noe, F.; Chodera, J. D.; Bolhuis, P. G. "OpenPathSampling: A Python Framework for Path Sampling Simulations. 2. Building and Customizing Path Ensembles and Sample Schemes", *Journal of Chemical Theory and Computation* **2019**, 15, 837–856.
- [10] Lervik, A.; Riccardi, E.; van Erp, T. S. "PyRETIS: A well-done, medium-sized python library for rare events", *Journal of Computational Chemistry* **2017**, 38, 2439–2451.
- [11] Riccardi, E.; Lervik, A.; Roet, S.; Aaroen, O.; van Erp, T. S. "PyRETIS 2: An improbability drive for rare events", *Journal of Computational Chemistry* **2020**, 41, 370–377.
- [12] Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics", *PLOS Computational Biology* **2017**, 13, 1–17.
- [13] *OpenMM benchmarks*; 2022. <https://openmm.org/benchmarks>.
- [14] Jung, H.; Okazaki, K.-i.; Hummer, G. "Transition path sampling of rare events by shooting from the top", *The Journal of Chemical Physics* **2017**, 147, 152716.
- [15] Riccardi, E.; Dahlen, O.; van Erp, T. S. "Fast decorrelating Monte Carlo moves for efficient path sampling", *The Journal of Physical Chemistry Letters* **2017**, 8, 4456–4460.
- [16] Zhang, D. T.; Riccardi, E.; van Erp, T. S.; *Path sampling with sub-trajectory moves*; in preparation.
- [17] Bolhuis, P. G.; Swenson, D. W. H. "Transition Path Sampling as Markov Chain Monte Carlo of Trajectories: Recent Algorithms, Software, Applications, and Future Outlook", *Advanced Theory and Simulations* **2021**, 4, 2000237.

- [18] Plattner, N.; Doll, J. D.; Dupuis, P.; Wang, H.; Liu, Y.; Gubernatis, J. E. "An infinite swapping approach to the rare-event sampling problem", *The Journal of Chemical Physics* **2011**, 135, 134111.
- [19] Hockney, R.; Eastwood, J. *Computer Simulations Using Particles*; Taylor & Francis Group, 1988.
- [20] Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters", *The Journal of Chemical Physics* **1982**, 76, 637–649.
- [21] Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. "Improved side-chain torsion potentials for the Amber ff99SB protein force field", *Proteins: Structure, Function, and Bioinformatics* **2010**, 78, 1950–1958.
- [22] Iftimie, R.; Minary, P.; Tuckerman, M. E. "Ab initio molecular dynamics: Concepts, recent developments, and future trends", *Proceedings of the National Academy of Sciences* **2005**, 102, 6654–6659.
- [23] Frenkel, D.; Smit, B. *Understanding Molecular Simulations*; Academic Press, 2002.
- [24] Leimkuhler, B.; Matthews, C. "Rational Construction of Stochastic Numerical Methods for Molecular Sampling", *Applied Mathematics Research eXpress* **2012**, 2013, 34–56.
- [25] Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. "Equation of State Calculations by Fast Computing Machines", *The Journal of Chemical Physics* **1953**, 21, 1087–1092.
- [26] Hastings, W. K. "Monte Carlo sampling methods using Markov chains and their applications", *Biometrika* **1970**, 57, 97–109.
- [27] van Erp, T. S. "Efficiency analysis of reaction rate calculation methods using analytical models I: The two-dimensional sharp barrier", *The Journal of Chemical Physics* **2006**, 125, 174106.
- [28] Swendsen, R. H.; Wang, J.-S. "Replica Monte Carlo Simulation of Spin-Glasses", *Physical Review Letters* **1986**, 57, 2607–2609.

- [29] van Erp, T. “Dynamical rare event simulation techniques for equilibrium and nonequilibrium systems”, *Advances in Chemical Physics* **2012**, 151, 27.
- [30] Swain, P. H.; Hauska, H. “The decision tree classifier: Design and potential”, *IEEE transactions on geoscience electronics* **1977**, 15, 142–147.
- [31] Breiman, L. “Random Forests”, *Machine Learning* **2001**, 45, 5–32.
- [32] Charbuty, B.; Abdulazeez, A. “Classification based on decision tree algorithm for machine learning”, *Journal of Applied Science and Technology Trends* **2021**, 2, 20–28.
- [33] Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers”, *SoftwareX* **2015**, 1-2, 19–25.
- [34] Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in ’t Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; Shan, R.; Stevens, M. J.; Tranchida, J.; Trott, C.; Plimpton, S. J. “LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales”, *Computer Physics Communications* **2022**, 271, 108171.
- [35] Dellago, C.; Bolhuis, P. G.; Chandler, D. “Efficient transition path sampling: Application to Lennard-Jones cluster rearrangements”, *The Journal of Chemical Physics* **1998**, 108, 9236–9245.
- [36] Grünwald, M.; Dellago, C.; Geissler, P. L. “Precision shooting: Sampling long transition pathways”, *The Journal of Chemical Physics* **2008**, 129, 194101.
- [37] Juraszek, J.; Bolhuis, P. G. “Rate Constant and Reaction Coordinate of Trp-Cage Folding in Explicit Water”, *Biophysical Journal* **2008**, 95, 4246–4257.
- [38] Peters, B.; Trout, B. L. “Obtaining reaction coordinates by likelihood maximization”, *The Journal of Chemical Physics* **2006**, 125, 054108.

- [39] Mullen, R. G.; Shea, J.-E.; Peters, B. "Easy Transition Path Sampling Methods: Flexible-Length Aimless Shooting and Permutation Shooting", *Journal of Chemical Theory and Computation* **2015**, *11*, 2421–2428; PMID: 26575542.
- [40] McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. "MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories", *Biophysical Journal* **2015**, *109*, 1528 – 1532.
- [41] *Getting started with PyRETIS*; 2022. <https://pyretis.org/current/user/getting-started.html>.
- [42] Plattner, N.; Doll, J. D.; Dupuis, P.; Wang, H.; Liu, Y.; Gubernatis, J. E. "An infinite swapping approach to the rare-event sampling problem", *The Journal of Chemical Physics* **2011**, *135*, 134111.
- [43] Balasubramanian, K.; Ph.D. thesis; Loyola College; Madras, India; 1980.
- [44] Bax, E.; Ph.D. thesis; California Institute of Technology; Pasadena, United States of America; 1998.
- [45] Bax, E.; Franklin, J. "A finite-difference sieve to compute the permanent", *CalTech-CS-TR-96-04* **1996**.
- [46] Glynn, D. G. "The permanent of a square matrix", *European Journal of Combinatorics* **2010**, *31*, 1887–1891.
- [47] van Erp, T. S. "Efficiency analysis of reaction rate calculation methods using analytical models I: The two-dimensional sharp barrier", *The Journal of Chemical Physics* **2006**, *125*, 174106.
- [48] Siepmann, J. I.; Frenkel, D. "Configurational Bias Monte-Carlo - a new sampling scheme for flexible chains", *Molecular Physics* **1992**, *75*, 59–70.
- [49] Vlught, T.; Krishna, R.; Smit, B. "Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite", *The Journal of Physical Chemistry B* **1999**, *103*, 1102–1118.

- [50] Swendsen, R. H.; Wang, J.-S. “Nonuniversal critical dynamics in Monte Carlo simulations”, *Physical Review Letters* **1987**, 58, 86–88.
- [51] Peters, E. A. J. F.; de With, G. “Rejection-free Monte Carlo sampling for general potentials”, *Physical Review E* **2012**, 85, 026703.
- [52] Michel, M.; Kapfer, S. C.; Krauth, W. “Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps”, *The Journal of Chemical Physics* **2014**, 140, 054116.
- [53] van Erp, T. S.; Moqadam, M.; Riccardi, E.; Lervik, A. “Analyzing complex reaction mechanisms using path sampling”, *Journal of Chemical Theory and Computation* **2016**, 12, 5398–5410.
- [54] Hooft, F.; Pérez de Alba Ortíz, A.; Ensing, B. “Discovering collective variables of molecular transitions via genetic algorithms and neural networks”, *Journal of chemical theory and computation* **2021**, 17, 2294–2306.
- [55] Chen, W.; Ferguson, A. L. “Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration”, *Journal of computational chemistry* **2018**, 39, 2079–2102.
- [56] Schöberl, M.; Zabarar, N.; Koutsourelakis, P.-S. “Predictive collective variable discovery with deep Bayesian models”, *The Journal of chemical physics* **2019**, 150, 024109.
- [57] Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. “Reweighted autoencoded variational Bayes for enhanced sampling (RAVE)”, *The Journal of chemical physics* **2018**, 149, 072301.
- [58] Jung, H.; Covino, R.; Hummer, G. “Artificial intelligence assists discovery of reaction coordinates and mechanisms from molecular dynamics simulations”, *arXiv preprint arXiv:1901.04595* **2019**.
- [59] Rossi, K.; Jurásková, V.; Wischert, R.; Garel, L.; Corminbœuf, C.; Ceriotti, M. “Simulating solvation and acidity in complex mixtures with first-principles accuracy: the case of CH<sub>3</sub>SO<sub>3</sub>H and H<sub>2</sub>O<sub>2</sub> in phenol”, *Journal of chemical theory and computation* **2020**, 16, 5139–5149.

- [60] F.R.S., K. P. “LIII. On lines and planes of closest fit to systems of points in space”, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **1901**, 2, 559–572.
- [61] Jr., J. H. W. “Hierarchical Grouping to Optimize an Objective Function”, *Journal of the American Statistical Association* **1963**, 58, 236–244.
- [62] MacQueen, J. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*; pp 281–297.
- [63] Tenenbaum, J. B.; de Silva, V.; Langford, J. C. “A Global Geometric Framework for Nonlinear Dimensionality Reduction”, *Science* **2000**, 290, 2319–2323.
- [64] Schmidhuber, J. “Deep learning in neural networks: An overview”, *Neural Networks* **2015**, 61, 85–117.
- [65] Friedman, J. H. “Greedy function approximation: A gradient boosting machine.”, *The Annals of Statistics* **2001**, 29, 1189 – 1232.
- [66] Shapley, L. S. *A value for n-person games*; Roth, A. E., Ed.; Cambridge University Press, 1988; p 31–40.
- [67] Wachter, S.; Mittelstadt, B.; Russell, C. “Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR”, *arXiv preprint arXiv:1711.00399* **2017**.
- [68] Moqadam, M.; Lervik, A.; Riccardi, E.; Venkatraman, V.; Alsberg, B. K.; van Erp, T. S. “Local initiation conditions for water autoionization”, *Proceedings of the National Academy of Sciences* **2018**, 115, E4569–E4576.
- [69] Young, G.; Householder, A. S. “Discussion of a set of points in terms of their mutual distances”, *Psychometrika* **1938**, 3, 19–22.
- [70] Behler, J.; Parrinello, M. “Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces”, *Physical Review Letters* **2007**, 98, 146401.
- [71] Prior, I. A.; Lewis, P. D.; Mattos, C. “A Comprehensive Survey of Ras Mutations in Cancer”, *Cancer Research* **2012**, 72, 2457–2467.

- [72] Rogal, J.; Bolhuis, P. "Multiple State Transition Path Sampling", *The Journal of chemical physics* **2009**, 129, 224107.
- [73] Singh, J.; Petter, R. C.; Baillie, T. A.; Whitty, A. "The resurgence of covalent drugs", *Nature Reviews Drug Discovery* **2011**, 10, 307–317.
- [74] Abdeldayem, A.; Raouf, Y. S.; Constantinescu, S. N.; Moriggl, R.; Gunning, P. T. "Advances in covalent kinase inhibitors", *Chem. Soc. Rev.* **2020**, 49, 2617–2687.
- [75] Folkestad, S. D.; et al.. "eT 1.0: An open source electronic structure program with emphasis on coupled cluster and multilevel methods", *The Journal of Chemical Physics* **2020**, 152, 184103.
- [76] van Erp, T. S.; Moqadam, M.; Riccardi, E.; Lervik, A. "Analyzing complex reaction mechanisms using path sampling", *Journal of Chemical Theory and Computation* **2016**, 12, 5398–5410.



# Paper A

## PyRETIS 2: An improbability drive for rare events

Enrico Riccardi, Anders Lervik, Sander Roet, Ola  
Aarøen, and Titus S. van Erp

*J. Comput. Chem.* **2020**, *41*, 370-377;  
doi: 10.1002/jcc.26112



# PyRETIS 2: An Improbability Drive for Rare Events

Enrico Riccardi <sup>\*,[a]</sup> Anders Lervik <sup>[a]</sup> Sander Roet,<sup>[a]</sup> Ola Aarøen,<sup>[b]</sup> and Titus S. van Erp <sup>[a,c]</sup>

The algorithmic development in the field of path sampling has made tremendous progress in recent years. Although the original transition path sampling method was mostly used as a qualitative tool to sample reaction paths, the more recent family of interface-based path sampling methods has paved the way for more quantitative rate calculation studies. Of the exact methods, the replica exchange transition interface sampling (RETIS) method is the most efficient, but rather difficult to implement. This has been the main motivation to develop the open-source Python-based computer library PyRETIS that was released in 2017. PyRETIS is designed to be easily interfaced

with any molecular dynamics (MD) package using either classical or ab initio MD. In this study, we report on the principles and the software enhancements that are now included in PyRETIS 2, as well as the recent developments on the user interface, improvements of the efficiency via the implementation of new shooting moves, easier initialization procedures, analysis methods, and supported interfaced software. © 2019 The Authors. *Journal of Computational Chemistry* published by Wiley Periodicals, Inc.

DOI: 10.1002/jcc.26112

## Introduction

Simulation of long time scales has been a major challenge in molecular simulations. Although the increase of system scale is straightforwardly parallelizable, extending simulation time is not. This is a severe problem as molecular dynamics (MD) typically requires femtoseconds time steps. Even if the computational evaluation of such a step is usually achieved within a fraction of second, it would take centuries of CPU time to compute 1 s of real time. This makes it nearly impossible to study processes like chemical reactions, phase transitions, and conformational changes with MD. These processes typically occur so infrequently that years of computer time are needed to observe even a single event.

The vast majority of methods, which have been developed for overcoming this problem, either alter the potential energy surface or the dynamics of the system (see, e.g., Refs. [1,2]). The use of Monte Carlo (MC) in path space is an approach that does not disturb the underlying physical dynamics, but generates unlikely molecular events like an *improbability drive*.<sup>[3,4]</sup> This approach is the essence of transition path sampling (TPS)<sup>[5–7]</sup> in which repetitively a new path is being generated from an old path via MC moves that obey detailed balance. The most important MC move is the so-called shooting move in which first a random time slice (comprising the phase point at a certain MD step) of the old path is selected and then stochastically modified. For instance, random disturbances could be applied to the velocities of that point. After that, this point is first propagated backward and then forward in time using a standard symplectic and time-reversible integrator for MD, Langevin, or Brownian dynamics. Moreover, as paths consist of time slices (phase points at discrete time steps), velocity Verlet,<sup>[8]</sup> and other reversible integration schemes should be preferred above leapfrog<sup>[9]</sup> as the latter provides velocities that are shifted in time by half a time step.<sup>[10]</sup> After the completion of these time

integrations, the forward and backward trajectories are glued together resulting in a new continuous path that follows the natural dynamics of the system. This path is finally accepted or rejected based on a Metropolis–Hastings rule.<sup>[11,12]</sup>

Transition interface sampling (TIS)<sup>[13]</sup> introduced several fundamental key elements that made efficient quantitative path sampling possible. Firstly, TIS introduced the statistical path ensemble with flexible path length, reducing the redundant exploration of the stable regions. Secondly, a series of path ensembles were defined based on a set of hyperplanes (interfaces). These hyperplanes are generally defined by a value of the order parameter (reaction coordinate/progress coordinate) which is a function of the coordinates (and possibly velocities) of the system.

The first interface, called  $\lambda_0$  or  $\lambda_A$ , defines the region of the initial state. The last interface, called  $\lambda_n$  or  $\lambda_B$ , defines the region of the final state. The first interface is placed such that a straightforward MD simulation starting from the reactant side would cross this interface sufficiently frequently, which enables the determination of the flux through this plane by straight forward MD. The last interface is placed sufficiently far across the barrier so that each trajectory from  $\lambda_0$  to  $\lambda_n$  will not easily return to the

[a] E. Riccardi, A. Lervik, S. Roet, T. S. van Erp  
Department of Chemistry, Norwegian University of Science and Technology, Høgskoleringen 5, 7491 Trondheim, Norway  
E-mail: enrico.riccardi@ntnu.no

[b] O. Aarøen  
Department of Biotechnology and Food Science, Norwegian University of Science and Technology, Høgskoleringen 5, 7491 Trondheim, Norway

[c] T. S. van Erp  
Center for Molecular Modeling (CMM), Ghent University, Technologiepark 903, 9052 Zwijnaarde, Belgium

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2019 The Authors. *Journal of Computational Chemistry* published by Wiley Periodicals, Inc.

initial state  $A$  and, therefore, can be considered as a successful transition from the initial state to the final state. As long as the positioning of  $\lambda_0$  and  $\lambda_n$  is reasonable and respects the above criteria, the final result will not largely be affected by their exact placement. The other interfaces are defined in between  $\lambda_0$  and  $\lambda_n$  and their positions are solely based on efficiency arguments.

Once the interfaces are defined, the TIS rate equation is then given by

$$k_{AB} = f_A \mathcal{P}_A(\lambda_B|\lambda_A)$$
$$\mathcal{P}_A(\lambda_B|\lambda_A) = \prod_{i=0}^{n-1} \mathcal{P}_A(\lambda_{i+1}|\lambda_i) \quad (1)$$

where  $f_A$  is the flux through  $\lambda_A$  and  $\mathcal{P}_A(\lambda_B|\lambda_A)$  is the very small probability that a crossing with  $\lambda_A$  will lead to a crossing with  $\lambda_B$  without recrossing  $\lambda_A$ . As this probability is generally extremely small, it cannot be calculated directly. However, it can be computed by the exact factorization in the second expression in eq. (1). One should realize that  $\mathcal{P}_A(\lambda_{i+1}|\lambda_i)$  is not just the probability to go from  $\lambda_i$  to  $\lambda_{i+1}$ , but rather a history-dependent conditional probability.

Replica exchange TIS (RETIS) samples all path ensembles in parallel and applies replica exchange moves between those.<sup>[14]</sup> In addition, it replaces the MD simulation for the flux calculation with the ensemble  $[0^-]$ , which consists of paths that start and end at  $\lambda_0$  but explore the reactant well region rather than the reaction barrier. For these features, RETIS is considerably more complex to implement in computer codes. This has been the main driving force to develop the open source PyRETIS<sup>[15]</sup> library.

A year after the official disclosure of PyRETIS, the Open Path Sampling (OPS) library was released.<sup>[16,17]</sup> The aims of PyRETIS and OPS are similar, but the libraries have been written with slightly different user communities in mind. OPS is more generic and allows the expert user to design different path ensembles. PyRETIS, on the other hand, has a stronger focus on the RETIS algorithm and a stronger emphasis on user-friendly accessibility (i.e., toward the nonexpert user). There are presently active collaborations between the two developer groups, which potentially could lead to a partial, or even full, merger of the two libraries in the future.

In this article, we discuss the code developments made in the release of PyRETIS 2. PyRETIS 1 was interfaced with GROMACS<sup>[18]</sup> and CP2K<sup>[19]</sup> for respectively, performing classical and ab initio MD. In PyRETIS 2, we improved the GROMACS interface and added interfaces with OpenMM<sup>[20,21]</sup> and LAMMPS.<sup>[22]</sup> Several structural improvements have been made to improve the readability and reliability of the code. The major ones are shortly described in this study. To improve the efficiency, the new MC moves in path space, Stone Skipping and Web Throwing developed by Riccardi et al.<sup>[23]</sup> have been implemented. An easier initialization procedure has also been introduced such that trajectories, or simple snapshots, can now directly be read by PyRETIS 2 and used to initialize the RETIS simulation. In terms of outputs, a graphical user interface (GUI) to quickly inspect trajectories and density plots as functions of different descriptors (collective variables) has been constructed and added to the library.

## Algorithmic Improvements

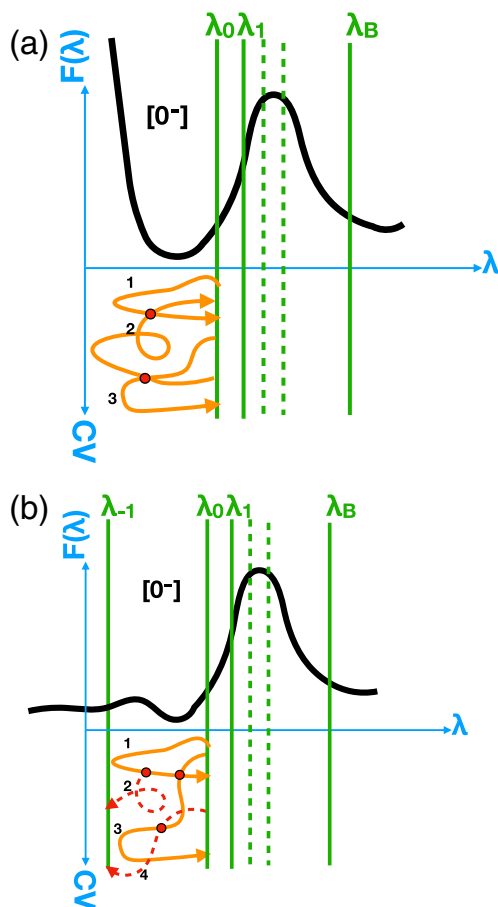
### Advanced path generating MC moves

To optimize the sampling efficiency, PyRETIS 2 allows a direct and intuitive selection of path sampling strategies. Two of the most promising and recent sampling methods, namely, *Stone Skipping and Web Throwing*,<sup>[23]</sup> have now also been included in the code. Stone skipping and web throwing are two advanced MC-based moves that reduce correlations between successively generated paths. This implies that fewer generated trajectories are required to estimate crossing probabilities and rate constants within a desired error range. Therefore, despite that the execution of a single MC step is more costly than standard shooting, the sampling efficiency can increase considerably (for the case study of Ref. [23], the increase in efficiency was more than an order magnitude).

We expect that these new moves will become the default choice as they are intrinsically faster than standard shooting, though it will require some adaptations with how PyRETIS presently handles external engines, before this will be paid off in practice. The essential aspect of the new MC moves is that they launch a sequence of short subpaths via a shooting protocol that shoots from the ensemble-specific interface, that is, from a time slice just before or after the interface. The shooting move for creating a subpath propagates in one time direction only, though requires to cross the interface in one single time step backward in time or forward in time, depending on whether the selected shooting point is a point just beyond or before the interface. If this single-step crossing condition is not fulfilled, new random velocities should be generated until the condition is met. While this single-step crossing condition can be verified in principle for a single MD step without doing an actual force calculation,<sup>[23]</sup> the single-step crossing test has to be carried out explicitly whenever a (PyRETIS) step consist of several MD steps generated by the external engine. This makes the repeated generation of random velocities, followed by the testing of the one-step crossing condition, potentially, a very expensive element of this MC move. If the dynamics is sufficiently stochastic, then this one-step crossing test can be avoided by maintaining the same two time slices before and after the interface and create subpaths via one-way shooting protocol from the point that is after the interface, that is, without changing the velocities. For deterministic dynamics and dynamics that is only moderately stochastic (e.g., underdamped Langevin dynamics), the new MC moves might not yet outperform standard shooting until a new approach for handling the external engines has been implemented.

### The $[0^-]$ ensemble with additional confining interface

The  $[0^-]$  ensemble was introduced by van Erp<sup>[14]</sup> to replace the MD simulation for computing the flux in eq. (1) and to allow for replica exchange moves between all path ensembles. Paths in the  $[0^-]$  ensemble start at  $\lambda_0$ , like all other paths in the other ensembles, but move away from the barrier exploring the reactant well. The path is terminated once it recrosses  $\lambda_0$ . As a result, the time slices of a valid path in the  $[0^-]$  path ensemble



**Figure 1.** Illustration of the difference between standard RETIS interface positioning and RETIS with an extra  $\lambda_{-1}$  interface. Upper parts of a) and b) show the free energy as a function of  $\lambda$  with an example of how interfaces could be positioned. Bottom parts of a) and b) show examples of possible trajectories depicted in the  $(\lambda, CV)$  plane generated via the shooting move in the  $[0^-]$  ensemble, where CV is an additional collective variable different from  $\lambda$ . a) The standard situation of systems for which RETIS was originally designed. The free energy provides a natural boundary at the left side, which implies that the  $\lambda$  coordinate cannot get much smaller than  $\lambda_0$  and respective path generation trials (indicated by the numbers 1, 2, and 3) starting from the shooting points (red circles) will relatively quickly end at  $\lambda_0$  in the backward and forward time direction, yielding new acceptable paths. This situation is, for instance, typical for bond breaking, nucleation, and protein folding. b) The situation for which the additional  $\lambda_{-1}$  interface was introduced and is typical in, for example, permeation studies. In this case, the free energy does not provide a natural boundary at the left. Trajectories can in principle continue to move toward the left, allowing endlessly long trajectories. PyRETIS 2, however, allows the user to define a  $\lambda_{-1}$  interface, such that the shooting moves hitting the  $\lambda_{-1}$  interface are directly rejected. This is the case for trials 2 and 4 which hit the  $\lambda_{-1}$  interface in the backward and forward time direction, respectively. [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

have order parameter values  $< \lambda_0$  except for the start and end points.

This assumes that the reactant well has a natural confinement at the left side of the barrier, either because the potential

energy increases when the order parameter is decreased below the equilibrium position of the reactant state, or because of periodic boundaries. For instance, if the order parameter is (minus) the distance between two molecules that can react if they approach, the periodic boundaries will ensure that their separation is bounded. However, if the box dimensions are large or if another undesired stable state exists at the left side of the reactant well, it might be needed to terminate the path early, as a completed path starting and ending at  $\lambda_0$  could be exceedingly long. This can be done in PyRETIS 2 by setting an extra boundary, a “minus one” interface  $\lambda_{-1}$  with  $\lambda_{-1} < \lambda_0$ , that ensures that the reactant state ensemble is restricted between  $\lambda_{-1}$  and  $\lambda_0$ . In the present implementation, a path is rejected when it hits this left boundary. Please note that the use of a left boundary on this ensemble results in an incorrect flux calculation. A more sophisticated approach will be implemented in a future release in which the flux calculation will account for the presence of the extra boundary. At the present, the usage of the left boundary requires a reevaluation of flux term by means of another simulation without the left boundary. An illustrative scheme of the  $\lambda_{-1}$  is reported in Figure 1.

## Interface with External Engines

The PyRETIS 1 program provided interfaces with GROMACS and CP2K. In PyRETIS 2, we have extended this with LAMMPS and OpenMM. In addition, some fundamental changes related to the GROMACS interface has been established in PyRETIS 2. These new developments are shortly discussed below. Working examples for each external engine can be found in the website (section: *Getting started*). The external engines communicate with PyRETIS at a certain frequency, defined by the *subcycles* number in the PyRETIS input file in the engine section. At each *subcycles* number, the external engine provides the order parameter or the information for its computation. Its magnitude depends on the selected engine and on the system under investigation (trade between speed, accuracy, and storage requirements).

## GROMACS

The GROMACS engine has been updated for PyRETIS 2. The strategy for the GROMACS engine in PyRETIS 1 relied on repeatedly starting and stopping the execution of GROMACS. That is, PyRETIS 1 executes GROMACS for a few MD steps, defined by the number of *subcycles*, obtains the order parameter and uses this to determine if the GROMACS run should be ended or continued. PyRETIS 2 still not only supports this, but also provides a potentially more efficient strategy. It will execute GROMACS and obtain the order parameter while GROMACS is running and will use this to determine when it is time to end the GROMACS run. This increases the efficiency of running GROMACS with PyRETIS as it reduces the number stop/restart calls to the GROMACS engine. To select the old approach, the engine class to select in the input file is *gromacs*, while the latter method is called by the *gromacs2* keyword. The new approach exploits the functionality of the MDTraj<sup>[24]</sup> library for efficiently reading GROMACS trajectory files.

## LAMMPS

PyRETIS 2 also includes an interface for LAMMPS.<sup>[22]</sup> For this engine, PyRETIS 2 will only provide information about the initial configuration and the stopping conditions (i.e., the maximum number of steps to perform and the location of the relevant interfaces). This enables the execution of LAMMPS to be fully handled by LAMMPS itself. This also requires that the order parameter is defined in a separate LAMMPS input file, which LAMMPS can use to calculate it. After the completion of a LAMMPS MD run, PyRETIS 2 will read the calculated order parameter and energies, and store the generated trajectory. Currently, PyRETIS 2 only supports microcanonical (NVE) dynamics when using LAMMPS. Note that temperature effects can still be studied as the temperature is linked to the MC sampling. That is, paths describe dynamics at constant energy, but the energies of different paths will fluctuate due to the shooting move. The approach can be defined as a canonical (NVT) sampling of NVE paths, and it is a popular approach in path sampling since it removes the dynamics from unphysical modifications of the equations of motion by a thermostat, while still sampling the canonical distribution. It reflects a system that is weakly coupled to a heat bath such that its effect is not noticeable on the time scale of the path length.

## OpenMM

PyRETIS 2 has added an interface with OpenMM.<sup>[21]</sup> In this version, OpenMM can only be used with PyRETIS as a library. That means that if an OpenMM Simulation class is initialized, this object can then be used to initiate the OpenMMEngine class inside PyRETIS. This OpenMMEngine class can then be used for all the PyRETIS internals. An automated setup from restructured text input, like the way PyRETIS handles the connection with the other engines, is not yet supported and will be added in a later version of PyRETIS. As OpenMM also supports running on GPUs, special care was taken to not create new OpenMM contexts, but instead update the coordinates and velocities by an *in-place* operation. This minimizes the communication and prevents unnecessary compilation time for running on GPUs. The current implementation is only suitable for simulation in the canonical ensemble. Support for the isothermal–isobaric (NPT) will be added to a later version of PyRETIS.

## Library Structure

### Ensemble structure

The paths generated by Path Sampling are grouped into ensembles. Each of them focuses on a different region of path space. Each one can rely on different MC rules to generate new paths. Dedicated setups, tailored to the region to explore, can thus improve the sampling by enabling the application of the most suitable techniques. These possible techniques are for instance the use of Stone Skipping Web Throwing moves (see “Advanced path generating MC moves” section) or the different ways to disturb the velocities when performing a shooting move. To use this feature, a user should simply declare the

```
Simulation
-----
task = retis
steps = 10000
interfaces = [ -0.9, -0.6, -0.3, 0.0, 1.0]

Ensemble
-----
interface = -0.9
shooting_move = sh

Ensemble
-----
interface = -0.6
shooting_move = ss
n_jumps = 16

Ensemble
-----
interface = -0.3
shooting_move = wt
n_jumps = 4
interface_sour = -0.5
```

**Figure 2.** New input structures to insert specific input for an ensemble. Each ensemble section refers to an interface specified within. In the example, for the three ensembles selected, different shooting moves, with different parameters, have been selected. Note that the default shooting move is “sh,” the first ensemble section is, therefore, not required.

ensemble to modify and specify the dedicated input as shown in the example in Figure 2.

In case that no special treatment for an ensemble is indicated, the main settings will be applied, preserving the same functionality as the previous version of PyRETIS.

### Defining the order parameter and additional collective variables

The input file to PyRETIS has been updated to directly support several collective variables. This implies that a set of additional collective variables can be listed in the input file and those will be calculated along with the main order parameter. These collective variables do not affect the RETIS algorithm but can provide valuable information for the analysis of the path ensembles. The additional collective variables are hence descriptors to be used in postprocessing to elucidate mechanisms occurring in the investigated transition. There are, therefore, no constraints on the number or type of collective variables. The new input scheme to include additional collective variables is illustrated in Figure 3.

### Paths storage and restart reproducibility

The storage of generated paths has been updated for PyRETIS 2. The trajectories can be saved with any frequency in a compressed format. The respective order parameter and energies (as a function of time) are saved in separate files with arbitrary frequencies too. This setup allows independent visual inspections of the generated trajectories and the restart of a new path

```

Orderparameter
-----
class = Position
dim = x
index = 0
periodic = False

Collective-variable
-----
class = Velocity
dim = y
index = 16

Collective-variable
-----
class = Angle
index = 1, 3, 7

Collective-variable
-----
class = Custom
module = orderp.py
index = 11, 13, 17, 22

```

**Figure 3.** New input structures to include multiple collective variables along with the main order parameter. The order of the collective variables will be maintained in the generated output by PyRETIS in the orderp.txt files. In this example, the order parameter is the x position of atom 0, the first collective variable is the y component of the atom 16 velocity. The second collective variable is the angle between the atoms with indexes 1, 3, and 7. While these descriptors are computed with internal functions, the latter order parameter, called Custom, is an example of an externally computed descriptor (located in the module orderp.py).

sampling simulation from a previous successful trajectory (in case that the latest data got corrupted, e.g., by a hardware failure). Furthermore, by selecting a large number of descriptors (order parameter and collective variables), it is now possible to limit the size of stored data, while still having a detailed system description that can be quickly handled by the visualization tool introduced in the present release.

### Random number generators

A new treatment of random number generators has been implemented in PyRETIS 2 to allow an exact reproducibility of simulations even if they have been performed in parallel. The random number generator is called at many places in the RETIS algorithm: In each cycle, a random number is drawn to select (1) the RETIS move (to select between the options swap/time-reversal/shooting or other types of path MC moves), (2) the relative shooting (if applicable), (3) the selection of the frame index, (4) the selection of the new velocities (velocities may be kept unchanged in the case of stochastic dynamics), and (5) the random forces whenever stochastic dynamics is applied.

In PyRETIS 2, each task has a dedicated and unique random number generator. The feature permits a more accessible reproduction of the simulation results and a precise continuation of simulations even in parallel jobs.

### MDTraj

PyRETIS 2 includes MDTraj<sup>[24]</sup> as an interpreter for external files. That is, external trajectories (from GROMACS, CP2K, LAMMPS) can be read with this Python library that was developed to deal efficiently with massive trajectories. The aim is to gain, in later releases of PyRETIS, increased independence from the external engine. As the minimal output that PyRETIS needs to receive from the external engine is only an ordered list of order parameters describing a trajectory, a universal interpreter to read external files in the various formats would simplify and uniform the interface for the various external engines. In PyRETIS 2, MDTraj is used in the load function to extract the desired frames. The package allows the use and development of arbitrary external functions to compute the order parameters and additional collective variables.

## Input/Output Schemes

### Load functionality and initialization

A new functionality, the *load* feature, has been added to PyRETIS 2 that simplifies one of the most user demanding tasks of the path sampling algorithm, the initialization of the simulation. The feature permits a direct initialization of path sampling simulations using configuration frames that could be generated by any type of fast simulation method and software. The new load function reads frames and trajectories supplied to PyRETIS 2 without the need for any further descriptor. PyRETIS 2 will compute the order-parameter, the additional collective variables and eventually the energy of the provided frames and trajectories. The input information will be automatically rearranged to satisfy the various ensemble definitions, when possible. PyRETIS 2 will then start the exploration of the path space from these initial frames indicating, with the “ld” flag in the output “pathensemble.txt” files, that the latest accepted path is a repetition of the initial path loaded. Once the initialization is completed this “ld” flag should no longer be present in the newly produced output lines in the “pathensemble.txt” files.

The strategy allows the inclusions of frames along the transition of interest that can be constructed by for instance constrained dynamics, nudge elastic band<sup>[25]</sup> or metadynamics<sup>[2]</sup> using any type of software. It should be underlined that the load function only simplifies the initialization procedure and will not influence the final converged result. It is, therefore, possible to provide to the load function a hypothetical path that has no physical meaning. The load initialization procedure also has a preprocessing feature to limit the overall simulation memory requirements. That is, for a massive trajectory, that explores the transition region only in a relatively short period, PyRETIS 2 automatically omits the frames that are not part of the ensemble of interest.



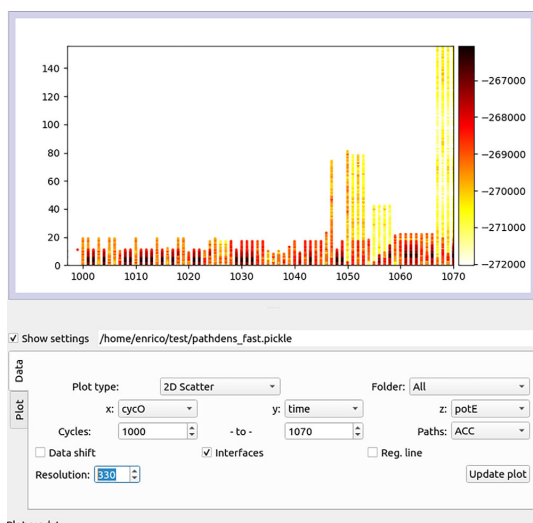
The load function can also be invoked after a restart. The frames from the restart file are then used as frames for the *load* method. The difference compared to a normal restart, which is a continuation, is that the load permits a broad change of the simulation parameters: from the order parameter selection to the interface positioning. It should be reminded here that, unlike a normal restart, the use of load with concomitant changes in the simulation parameters, should not be viewed as a continuation, and the previous sampling should be discarded in the final analysis.

## Visualization

As specified by the user, different output files are created and also the frequency of writing the data can be set. Data sets can be created by optionally writing all-atom coordinates, velocities, and a list of descriptors (collective variables) for each trajectory at each time slice. Writing of all coordinates and velocities gives rise to huge data sets which asks for significant memory requirements for storage, and complicates the interpretation and data analysis. Therefore, we advise the user to reflect on what may be potentially important collective variables and then define a large set of descriptors *prior* to performing production runs. This simplification reduces the dimensionality of the data on which the interpretation of the reaction mechanisms shall be based. Still, even with this reduced dimensionality it can be nontrivial to filter out the essential information.

To facilitate this task, PyRETIS 2 includes a new visualization tool, named PyVisA. The software permits an almost immediate visualization of the initial, partial and final simulation results by allowing the automated generation of plots of various simulation collective variables. A user can promptly plot energies, the order parameter, collective variables, cycle number, and path lengths as a function of each other in different type of plots, for different ensembles, for selected cycle ranges, for accepted or rejected paths. A GUI has been constructed facilitate this visualization and to easily navigate through the PyRETIS 2 outputs.

The visualization of the various descriptors, for example, the collective variable density plots, in the GUI allows the user to interactively inspect different parameter combinations, potentially revealing additional information about the system dynamics. In the initialization stage, a user can better position the various interfaces, select the most appropriate MC move (e.g., standard shooting, Stone Skipping, Web Throwing, etc.), determine metastable states and even evaluate the efficacy of the selected order parameter in comparison with other collective variables. The descriptors include order parameters, collective variables, energy, number of simulation steps for path, RETIS cycles. The user can select the range of cycles to visualize, their type (accepted/rejected) and the plot type (e.g., 2D scatter, 3D scatter, local density). Figures 5 and 4 show two reports that can be obtained by the visualization tool's GUI. To execute the visualization and analysis tool (named "PyVisA"), the flag `-pyvisa` shall be added to the `pyretisanalyse` command. Further details, instructions, and examples can be found in Ref. [27], where the tool structure and features are detailed.



**Figure 4.** Visualization ("Plot type: 2D Scatter") of a set of accepted trajectories ("Paths: ACC") for a given range of cycles ("Cycles: 1000 to 1070") for all ensembles ("Folder: All"), plotted according to the cycle number ("x: cycO"), trajectory length ("y: time") and the potential energy ("z: potE"). With these selections, an user can gain a statistical insight in the progress of the sampling for a restricted range of cycles. From this illustrative plot, it can be noticed that the paths generated in the surrounding of cycle 1070 are longer and with lower potential energy than the previous for the given range. A nonuniform path length distribution can be symptom of a nonconverged sampling. The paths in the latter cycles seems to have identified a region with lower potential energy, a different pathway might have been, therefore, identified. [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

The visualization tool, furthermore, provides a structured computational framework that will permit a general implementation of advanced analysis approaches. Predictor methods<sup>[28]</sup> or machine learning methods<sup>[29]</sup> to evaluate the quality of the selected order parameter in the description of the sampled event are the two most immediate examples.

## User Support

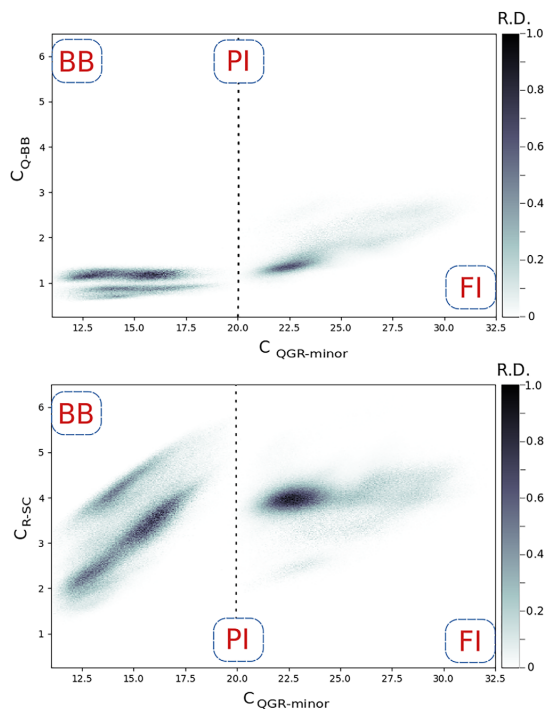
### Compatibility and installation

PyRETIS 2 supports Python 3.6 and 3.7 and is distributed via pip and conda (via the conda-forge channel) for the main releases. The visualization package, PyVisA, can be found in the development versions of PyRETIS (PyRETIS v2.develop and PyRETIS 3. beta) and in the forthcoming releases beyond Version 2.4. The development versions can be installed by downloading the PyRETIS source code from gitlab via the command "python setup.py install" from the main directory. Further details on its installation and usage can be found in Ref. [27].

### Test examples

Along with a long list of unit tests, the PyRETIS development is also tested versus a series of main test simulations that are automatically executed daily to assert the code and its





**Figure 5.** (Figure taken from Ref. [26]) Path frame density plots of RETIS trajectories for the insertion transition of H-NS to DNA.  $C_{\text{QCR-minor}}$  is the main order parameter and it is obtained by a contact map between H-NS (QGR motif) and the DNA minor groove region. The descriptors  $C_{\text{R-SC}}$  and  $C_{\text{Q-BB}}$  are obtained by the contact map of specific H-NS region (R or Q motif) and DNA region (side chain and back bone). A complete description of the descriptors on the axis and of the meta-stable states can be found in the work of Riccardi et al.<sup>[26]</sup> Darker color represents a higher probability. The plots showed that for the Q-BB interaction to happen, the protein has to be in contact with DNA first, while the R-SC interaction seems to anticipate the H-NS-DNA interaction. In essence, these descriptive plots showed part of the mechanisms of H-NS adsorption. [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

dependencies status. These test simulations are available in the development versions of PyRETIS (which can be installed via git as described in the section on code availability). The test simulations have also a pedagogical purpose: users can gain experience and familiarity with the PyRETIS input scheme. These simulations are grouped by the external engine selected and by the functionality under test. Hereby, we are constantly increasing the number of test simulations. At the current stage, test simulations show the usage of different engines, initiation functions (e.g., “kick,” “load,” and “restart”), different simulation schemes (e.g., MD, TIS, and RETIS) and different selection of shooting moves (e.g., Stone Skipping and Web Throwing).

### Web interface

By the release of PyRETIS 1, we created a website ([www.pyretis.org](http://www.pyretis.org)) to give support to PyRETIS users. With PyRETIS 2, new examples and guides have been included in the *Examples*

section and the *User Guide* section to further facilitate and guide the usage of PyRETIS. In particular, the example section contains working examples for each of the external engines supported.

Cases of study performed with PyRETIS are constantly uploaded on the PyRETIS web page under the “Main studies performed by PyRETIS” section, and the files used to initiate and control the simulations are shared to facilitate the reproducibility of our investigations, with respect to FAIR data policies.<sup>[30]</sup>

In particular, our recent investigation of water autoionization<sup>[31]</sup> and the Histone-like Nucleoid Structuring protein (H-NS) binding to DNA<sup>[26]</sup> are listed on the website. They constitute two successful applications of the RETIS algorithm, where the sampling software has been interfaced with CP2K<sup>[19]</sup> in the first work, and GROMACS<sup>[18]</sup> in the latter. The *User Guide* section contains a new set of entries to facilitate the usage of the code. The guide comprises (1) instructions for the installation of PyRETIS in a user and a developer mode, (2) information on how to use and set up external engines and order parameters, (3) help for common errors, and (4) instructions on how to report bugs.

### Future Work

Despite the considerable efforts put in the code development, there are various expansions that would be desirable to increase usability, efficiency, and compatibility with other sampling software. We aim to automatize some of the parameter selections such as the interface positions, the number of jumps in the stone skipping and web throwing moves and the SOUR interface<sup>[23]</sup> in web throwing, the relative shooting weights, and the frequency of selection of the various MC moves.

An interface with VASP has been initiated and partially completed. It will be released after completion and after performing sufficient testing. In parallel, we are considering the implementation of a “translation” platform that might enable a single scheme to deal with the input and output of multiple engines. Besides the already implemented MDTraj,<sup>[24]</sup> Python packages such as MDAnalysis<sup>[32,33]</sup> and Atomic Simulation Environment (ASE)<sup>[34]</sup> can facilitate the realization of such a platform.

In the next PyRETIS release, the visualization tool will become an integral part of the code. It will provide a multidimensional and structured analysis framework. Advanced analysis approaches, such as the predictive power analysis method<sup>[28]</sup> and machine learning based methods to evaluate the quality of different collective variables,<sup>[29]</sup> will be readily implemented. We are therefore interested in direct support and collaboration with potential new developers that are interested to apply and expand PyRETIS.

### Software Availability

PyRETIS 2 is free (released under a LGPLv2.1+ license) and can be obtained as described previously and at <http://www.pyretis.org/user/install.html>. The source code, the visualization tool and the development version are accessible at: <https://gitlab.com/pyretis/pyretis>.

## Acknowledgments

The authors thank the Research Council of Norway for funding (project nos: 250875 and 267669), NOTUR for providing HPC facilities (project no.: NN9254K), the Olav Thon foundation for the support in the development of interactive visualization tools for teaching and the Peder Sather Center for the support in the development of the LAMMPS-PyRETIS interface. The authors also thank the Lorentz Center for supporting our workshop on PyRETIS and Path Sampling in Leiden, March 11–15, 2019. Sudi Jawahery, Anastasia Maslechko, Raffaella Cabriolu, An Ghysels, Jocelyne Vreede, Christopher Daub, and Mahmoud Moqadam are thanked for their feedbacks and useful discussions.

**Keywords:** PyRETIS · rare event · path sampling · python · kinetics

How to cite this article: E. Riccardi, A. Lervik, S. Roet, O. Aarøen, T. S. van Erp. *J. Comput. Chem.* **2020**, *41*, 370–377. DOI: 10.1002/jcc.26112

- [1] F. G. Wang, D. P. Landau, *Phys. Rev. Lett.* **2001**, *86*, 2050.
- [2] A. Laio, F. L. Gervasio, *Rep. Prog. Phys.* **2008**, *71*, 126601.
- [3] D. Castelvechi, *Sci. News* **2007**, *171*, 372.
- [4] D. Adams, *The Hitch Hiker's Guide to the Galaxy: A Trilogy in Five Parts*, Hitchhiker's Guide to the Galaxy Series, Penguin Random House, **1995**.
- [5] C. Dellago, P. G. Bolhuis, F. S. Csajka, D. Chandler, *J. Chem. Phys.* **1998**, *108*, 1964.
- [6] P. G. Bolhuis, C. Dellago, D. Chandler, *Faraday Discuss.* **1998**, *110*, 421.
- [7] C. Dellago, P. G. Bolhuis, D. Chandler, *J. Chem. Phys.* **1999**, *110*, 6617.
- [8] L. Verlet, *Phys. Rev.* **1967**, *159*, 98.
- [9] O. Buneman, *J. Comput. Phys.* **1967**, *1*, 517.
- [10] B. J. Leimkuhler, S. Reich, R. D. Skeel, *Integration Methods for Molecular Dynamics*. In *Mathematical Approaches to Biomolecular Structure and Dynamics*, New York: Springer, **1996**, p. 161.
- [11] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *J. Chem. Phys.* **1953**, *21*, 1087.
- [12] W. K. Hastings, *Biometrika* **1970**, *57*, 97.
- [13] T. S. van Erp, D. Moroni, P. G. Bolhuis, *J. Chem. Phys.* **2003**, *118*, 7762.
- [14] T. S. van Erp, *Phys. Rev. Lett.* **2007**, *98*, 268301.
- [15] A. Lervik, E. Riccardi, T. S. van Erp, *J. Comput. Chem.* **2017**, *38*, 2439.
- [16] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, P. G. Bolhuis, *J. Chem. Theory Comput.* **2019**, *15*, 813.
- [17] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, P. G. Bolhuis, *J. Chem. Theory Comput.* **2019**, *15*, 837.
- [18] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, E. Lindahl, *SoftwareX* **2015**, *1–2*, 19.
- [19] J. Hutter, M. Iannuzzi, F. Schiffmann, J. VandeVondele, *WIREs Comput. Mol. Sci.* **2014**, *4*, 15.
- [20] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, V. S. Pande, *J. Chem. Theory Comput.* **2013**, *9*, 461.
- [21] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, V. S. Pande, *PLoS Comput. Biol.* **2017**, *13*, 1.
- [22] S. Plimpton, *J. Comput. Phys.* **1995**, *117*, 1.
- [23] E. Riccardi, O. Dahlen, T. S. van Erp, *J. Phys. Chem. Lett.* **2017**, *8*, 4456.
- [24] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, V. S. Pande, *Biophys. J.* **2015**, *109*, 1528.
- [25] G. Henkelman, B. P. Uberuaga, H. Jónsson, *J. Chem. Phys.* **2000**, *113*, 9901.
- [26] E. Riccardi, E. C. van Mastbergen, W. W. Navarre, J. Vreede, *PLoS Comput. Biol.* **2019**, *15*, e1006845.
- [27] O. Aarøen and E. Riccardi. [Submitted in October 2019 to PeerJ Physical Chemistry]. Pyvisa: Visualization and Analysis of Path Sampling Trajectories.
- [28] T. S. van Erp, M. Moqadam, E. Riccardi, A. Lervik, *J. Chem. Theory Comput.* **2016**, *12*, 5398.
- [29] M. Moqadam, A. Lervik, E. Riccardi, V. Venkatraman, B. K. Alsberg, T. S. van Erp, *Proc. Natl. Acad. Sci. U. S. A.* **2018**, *115*, E4569.
- [30] E. Riccardi, S. Pantano, R. Potestio, *Interface Focus* **2019**, *9*, 20190005.
- [31] M. Moqadam, E. Riccardi, T. T. Trinh, A. Lervik, T. S. van Erp, *Phys. Chem. Chem. Phys.* **2017**, *19*, 13361.
- [32] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein. MDAnalysis: A Toolkit for the Analysis of Molecular Dynamics Simulations. *J. Comput. Chem.* **2011**, *32*, 2319–2327. <https://doi.org/10.1002/jcc.21787>.
- [33] R. J. Gowers, M. Linke, J. Barnoud, T. J. Reddy, M. N. Melo, S. L. Seyler, J. Domański, D. L. Dotson, S. Buchoux, I. M. Kenney, O. Beckstein, MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. In *Proceedings of the 15th Python in Science Conference*, Vol. 98, SciPy, Austin, TX. **2016**.
- [34] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dulak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K. W. Jacobsen, *J. Phys. Condens. Matter* **2017**, *29*, 273002.

Received: 8 July 2019

Revised: 25 October 2019

Accepted: 29 October 2019

Published online on 19 November 2019




## Paper B

# Exact non-Markovian permeability from rare event simulations

An Ghysels, Sander Roet, Samaneh Davoudi, and Titus  
S. van Erp

*Phys. Rev. Research* **2021**, 3, 033068;  
doi: 10.1103/PhysRevResearch.3.033068


## Exact non-Markovian permeability from rare event simulations

An Ghysels <sup>1</sup>, Sander Roet <sup>2</sup>, Samaneh Davoudi,<sup>1</sup> and Titus S. van Erp <sup>2,3</sup>

<sup>1</sup>*IBiTech - bioMMeda, Faculty of Engineering and Architecture, Ghent University, 9000 Gent, Belgium*

<sup>2</sup>*Department of Chemistry, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway*

<sup>3</sup>*Center for Molecular Modeling (CMM), Ghent University, Technologiepark, 9052 Zwijnaarde, Belgium*

 (Received 10 October 2020; revised 29 January 2021; accepted 24 June 2021; published 19 July 2021)

Permeation of compounds through membranes is important in biological and engineering processes, e.g., drug delivery through lipid bilayers, anesthetics, or chemical reactor design. Simulations at the atomic scale can provide insight in the diffusive pathways and they give estimates of the membrane permeability based on counting membrane transitions or on the inhomogeneous solubility-diffusivity model described by the Smoluchowski equation. For many permeants, permeation through a membrane is too slow to gather sufficient statistics with conventional molecular dynamics simulations, i.e., permeation is a rare event. Recent attempts to improve the description of the dynamics of such rare permeation events have been based on milestoneing, which allows the study of processes at timescales beyond those achievable by straightforward molecular dynamics. The approach is not relying on an overdamped description, but, still, it uses a Markovian approximation which is only valid for small permeants that are not disruptive to the membrane structure. To overcome this fundamental limitation, we show here how replica exchange transition interface sampling (RETIS) can effectively be used on this problem by deriving an effective set of equations that relate the outcome of RETIS simulations and the permeability coefficient. In addition, we introduce two new path Monte Carlo (MC) moves specifically for permeation dynamics, that are used in combination with the ordinary path generating moves, which considerably increase the efficiency. The advantage of our method is that it gives exact results, identical to brute force molecular dynamics, but orders of magnitude faster.

DOI: [10.1103/PhysRevResearch.3.033068](https://doi.org/10.1103/PhysRevResearch.3.033068)

### I. INTRODUCTION

Permeation of compounds through another medium is essential in both biological and engineering processes. In biology, the permeation of molecules, nutrients or nanoparticles through membranes is an integral part of a functioning cell, and understanding the drug delivery process can aid the design of new cancer drugs or anesthetics [1–4]. In chemical engineering, the transport of molecules can play a role in the selectivity of the molecules [5,6]. In highly complex chemical systems, it is valuable to unravel the various diffusion pathways, which can aid the optimization of a chemical reactor setup. Despite the existence of spectroscopic methods, such as fluorescence spectroscopy [7,8] or EPR experiments [9], the insight in diffusive transport at the molecular scale is difficult to obtain experimentally. Especially in inhomogeneous media where the diffusion of permeants is a function of the location, experiments usually only provide a global effective diffusion constant, without discerning local differences at the molecular scale. It is in this respect that molecular dynamics (MD) simulations can play a major role. MD creates molecular

trajectories and transport properties are directly observable at the molecular scale, such as the permeability  $P$ .

The permeability of a membrane is the flux through the membrane as a response to a concentration gradient over the membrane. A first standard approach to derive the permeability  $P$  from MD simulations is the counting method, which is based on measuring the rate of membrane transitions per unit of time and area [10–12]. Another standard approach is Bayesian analysis (BA) using the Smoluchowski equation, which assumes a position-dependent concentration profile as well as a position-dependent diffusion profile across the membrane [12–17]. The Smoluchowski equation is a pure diffusion model, where memory effects are not modeled. The serious limitation of these approaches is that, when the permeability is low, the statistics provided by MD might be insufficient, even when the trajectories are extended up to 1 microsecond of simulation time.

A more recent approach is the extraction of  $P$  from milestoneing. Milestoneing is based on the sampling of many short trajectories released from equilibrium distributions at hypersurfaces (milestones) [18]. Typically, these hypersurfaces are defined as subsets of configuration space with fixed values of the reaction coordinate (RC). The milestoneing method then counts from each originating hypersurface how often the left or right hypersurface is hit first and the time that it takes to let this happen. These two properties for the different milestones are then combined such that the dynamics of the system can be described by a Markovian hopping sequence

*Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.*

from one surface to the other. Rates, mean-first-passage times and other relevant dynamic and thermodynamic data can then be obtained. The central assumption underlying milestoneing is that the set of first hitting points, of the trajectories originating from one surface with a hypersurface left or right from it, is again distributed according to the equilibrium distribution. This leads to conflicting requirements. On the one hand, the milestones need to be set closely as this guarantees the highest efficiency boost compared to brute force MD. On the other hand, they need to be well separated for the central assumption to be sufficiently satisfied. In addition, for the assumption to hold, the choice of the RC is crucial and should ideally coincide with the committor [19]. The committor is generally an unknown coordinate that not only describes the position of the permeant relative to the center of mass of the membrane, but also should include nontrivial membrane deformations. Obtaining approximate forms of the committor is generally an immense task [20].

As a way to overcome the limitation of the Markovian assumption, this work will make use of the transition interface sampling (TIS) framework and of its extension, the replica exchange TIS (RETIS) formalism [21–23]. (RE)TIS gives a completely non-Markovian treatment of the interface hopping probabilities, while still being orders of magnitude faster than brute force MD. The overall rate is obtained in a divide-and-conquer mindset, by constructing a series of conditional probabilities to reach the next interface. Here we will show how the permeability, as defined in steady-state nonequilibrium conditions, can be transformed into functional form that depends on the equilibrium path ensemble properties which can be obtained from RETIS (instead of the typical canonical, NVT, or isobaric, NPT, phase space ensembles).

In an extension, we also show how the rates of transition can be defined in a meaningful way for an infinite system or a system with periodic boundaries such that it can be linked to the permeability. When the phase neighboring the membrane is unbounded, particles have a probability to indefinitely diffuse in the opposite direction of the membrane rather than through the membrane. Also when a system has periodic boundary conditions, it is difficult to detect whether particles reach the other side of the membrane through the periodic boundary or through the membrane itself. The permeability formula will be adapted to treat those cases. In addition, two additional moves will be introduced in the MC sampling of the interface ensembles. They will improve the efficiency when the simulation box contains multiple permeant molecules or when the membrane is symmetric with respect to the membrane center.

This article is organized as follows. In Sec. II, we review the existing approaches for obtaining  $P$  from MD simulations. In Sec. III, the (RE)TIS formalism is revised and the path ensembles are defined. In Sec. IV, the theoretical derivation of the permeability from (RE)TIS is presented. In Sec. V we derive the needed adaptation to treat systems with periodic boundary conditions. In Sec. VI, the two new MC moves are presented. In Sec. VII, we illustrate the accuracy and effectiveness of our approach by computing the permeability of a few basic model test systems. We end with concluding remarks in Sec. VIII.

## II. EXISTING APPROACHES FOR PERMEABILITY CALCULATIONS

### A. Direct counting

The permeability is defined as the ratio of the net flux  $J$  of particles transiting a membrane in the steady-state regime when imposing a small concentration difference  $\Delta c$  over the membrane,

$$P = \frac{J}{\Delta c}. \quad (1)$$

To compute the permeability, one could consider imposing the concentration difference at the membrane boundaries, as suggested by the definition in Eq. (1). However, at the molecular scale it is troublesome to impose a nonequilibrium steady-state concentration gradient, especially when periodic boundary conditions are applied.

A more common approach is therefore to count transitions in both directions through the membrane in an equilibrium simulation [10]. At first sight, Eq. (1) is no longer adequate for the computation of  $P$ . Indeed, note that  $J$  in Eq. (1) is the result of both crossings from left to right and from right to left, corresponding to a positive flux  $J^+$  and a negative flux  $J^-$ . These fluxes are proportional to the concentrations at the left and right hand side of the membrane, respectively, and bear opposite signs. The net flux is, hence, the result that is left after a partial cancellation of  $J^+$  and  $J^-$ , and in an equilibrium situation, where  $\Delta c = 0$ , the cancellation is complete and the net flux is zero. Whereas the permeability in this case is no longer measurable experimentally, in simulations it is still possible since it is relatively easy to trace the  $J^+$  and  $J^-$  fluxes individually from the MD trajectories and one can write

$$P = \frac{|J^+|}{c(-h/2)} = \frac{|J^-|}{c(h/2)} = \frac{|J^+| + |J^-|}{2c_{\text{ref}}}. \quad (2)$$

Here,  $z = 0$  is considered to be the center of the membrane of thickness  $h$  with borders at  $z = \pm h/2$ , and  $c(z)$  is the concentration profile across the membrane. The reference concentration is the concentration outside the membrane  $c_{\text{ref}} = c(-h/2) = c(h/2)$  and corresponds to the concentration of permeants in the bulk liquid.

In an equilibrium run, the sum of fluxes  $|J^+| + |J^-|$  is measured by counting full transitions from  $-h/2$  to  $h/2$  and vice versa. This approach has the advantage that it is nearly model-free. However, it is in practice a challenge to measure the flux at the atomic scale for all but the fastest permeation events [9,10,16,24–33]

### B. Smoluchowski equation

A second common approach is to run equilibrium MD simulations and to analyze these assuming the validity of the inhomogeneous solution-diffusivity (ISD) model, where transport is modeled by position-dependent Brownian diffusion (diffusion profile  $D(z)$ ) on a free energy landscape (profile  $F(z)$ ), as governed by the Smoluchowski equation [15]. The free energy is related to the permeant concentration in equilibrium through the Boltzmann probability,  $c(z) \sim \exp(-\beta F)$ , where  $\beta = 1/(k_B T)$  is the inverse temperature,  $k_B$  is Boltzmann constant. Given the two profiles  $F$  and  $D$ , the

permeability follows directly from solving the Smoluchowski equation for its steady-state solution in the presence of a fixed concentration difference  $\Delta c$  over the membrane [13,16],

$$\frac{1}{P} = e^{-\beta F_{\text{ref}}} \int_{-h/2}^{h/2} \frac{1}{e^{-\beta F(z)} D(z)} dz, \quad (3)$$

where  $F_{\text{ref}}$  is the free energy at the reference location, usually at  $z = -\frac{h}{2}$ .

The structure of Eq. (3) shows that the free energy profile is the dominant contribution to the permeability. Several cases can now be thought of for  $F(z)$ :  $F(z)$  has a barrier, is flat, or has a well. The last two cases are rather academic, as realistic membranes often form a combination of free energy barriers and wells, e.g., for  $O_2$  permeation through phospholipid bilayers [16]. First, when the permeation implies crossing a high free energy barrier, the integration range in Eq. (3) can readily be reduced from  $[-h/2, h/2]$  to  $[-h'/2, h''/2]$  with  $h', h'' < h$ ,

$$\frac{1}{P} \approx e^{-\beta F_{\text{ref}}} \int_{-h'/2}^{h''/2} \frac{1}{e^{-\beta F(z)} D(z)} dz \quad (4)$$

if the integrand can be neglected for the outer regions  $-h/2 < z < -h'/2$  and  $h''/2 < z < h/2$ . Given the exponential dependence on the free energy barrier, the integration boundaries  $-h'/2$  and  $h''/2$  can often be chosen rather close to the maximum of the free energy barrier as long as  $F_{\text{ref}}$  is taken in the bulk region. In the direct counting method, the neglect of the outer regions in the integration relates to the fact that nearly all transitions from  $-h/2$  to  $h/2$  contain one and only one single  $-h'/2$  to  $h''/2$  transition.

Second, the permeability in a homogeneous medium where  $F(z)$  is a flat profile, equals  $P = D/h$  with  $D$  the diffusion constant. The permeability halves when doubling the thickness  $h$ . This shows that the integration boundaries and reference region need to be chosen with some care whenever the free energy barrier is relatively small. Third, when  $F$  shows a free energy well, the permeants may be trapped in the membrane, and the integration boundaries should also be chosen with care.

Besides these conceptual fundamental issues related to the definition of the permeability, there are also practical issues. The danger exists that not all regions inside and outside of the membrane are accurately sampled in the equilibrium MD simulations. Lastly, slow sampling can also originate from a low diffusivity  $[D(z)]$  in Eq. (3), e.g., for permeants that are bulky. The challenge of the Smoluchowski approach is to determine the model parameters, i.e., the  $F(z)$  and  $D(z)$  profiles, which can be extracted *a posteriori* from the MD trajectories in various ways [10,12,15,16,28,29,34]. When MD is inadequate because of its limited timescale, the approach can be combined with rare events simulation techniques like umbrella sampling [35], adaptive bias force [36–38], and biased diffusion [39,40]. All these free energy methods, with the aim to compute the permeability via Eq. (3), have however as a shared limitation that they build on the validity of the Smoluchowski equation. The validity of this equation is questionable for many complex systems in which the transfer of permeants over the barrier involves other types of motion,

like a rotation of the permeant or a local stretch of a membrane opening [38,41].

### C. Path sampling approaches

Path sampling methods seem to provide a natural solution to the permeation problem since they are designed to maintain the natural dynamics of the process as much as possible, while still allowing the sampling of events that happen on long timescales. Among the different path sampling methods, applications on the permeation problem have so far mostly adopted the milestoning method [18]. In this technique, phase space is divided in domains that are separated by interfaces, called milestones. Trajectories are initiated at each milestone and run until they cross another milestone. The statistics over this set of short trajectories give the mean first passage times between pairs of milestones, which are then incorporated in a Markovian rate network model to extract the overall rate. Cardenas and Elber [42] proposed the formula for the permeability

$$P = \frac{J_1 q_f}{c_{\text{ref}} q_1}, \quad (5)$$

where  $J_1$  is the flux of particles hitting the first milestone per area and per time in equilibrium,  $q$  is the absolute flux vector of trajectories crossing the milestones, when solving the Markovian rate model for its steady-state solution with specific boundary conditions. Cardenas and Elber applied this to the permeation of a small peptide [42] or water molecule [43] through a 1,2-dioleoyl-sn-glycero-3-phosphocholine (DOPC) phospholipid bilayer, while Fathizadeh and Elber simulated potassium permeation through this DOPC membrane [44]. Recently, Votapka *et al.* derived an alternative formulation for the permeability based on milestoning [45].

The advantage of milestoning is that the MD trajectories that need to be generated are very short. However, the milestoning relies on the Markovian assumption that the system loses memory when it hits the next interface/milestone [18,46]. This assumption is only correct if the milestones are chosen along the isocommittor lines [19]. Hence, since the milestones are defined by fixed values of the RC, the RC should be the committor [47–50]. The committor is therefore often considered as the ideal RC since the dynamics projected on this one-dimensional coordinate becomes Markovian and models relying on the Markovian assumption, like Smoluchowski and milestoning, become exact. Note that the committor is in principle defined in phase space though by assuming that the dynamics is overdamped the committor can be defined in configuration space alone.

Hence, if milestoning is applied with this *ideal* RC (the committor) then trajectories released from the same milestone have the same probability of reaching  $h/2$  before  $-h/2$  regardless the point of origin within that milestone. This RC should account for all relevant rotations of the permeant, collective motions, and deformations of the membrane that could be vital for the permeation process. Including these motions within a single coordinate is highly nontrivial even if the committor is assumed to be only configuration space dependent based on the overdamped approximation. In practice, this is generally not even attempted. Rather, a simple intuitive



RC is chosen such as the  $z$  coordinate of the permeant that is followed. This pragmatic choice will generally invalidate the Markovian assumption leading to a systematic error that can be mild or severe depending on the system. Another path sampling approach that has similarities with milestoning is the partial path transition interface sampling (PPTIS) [51]. The PPTIS method is a Markovian variant of the TIS method. Still, it uses a less stringent assumption than milestoning by including more memory in the dynamical description. In PPTIS, if the system hits an interface, the chance to move to either its left or right interface depends on the history on the path i.e., from which interface (left or right) it came from. Still, no memory is retained before that point. The systematic error due to the nonideal RC is therefore presumably lower in PPTIS than in milestoning.

Path sampling methods that include the complete history dependence of the dynamics, such as transition path sampling (TPS), transition interface sampling (TIS), forward flux sampling (FFS) [52], adaptive multilevel splitting (AMS) [53], and replica exchange TIS (RETIS), are exact and independent of the RC, which is a big advantage in complex systems. The sampling of complete trajectories will make these methods generally more computationally intensive than milestoning or PPTIS, though a quantitative comparison relies on the trade-off between systematic and statistical error. FFS and AMS are based on a splitting approach which makes it applicable for nonequilibrium dynamics as well. On the other hand, the lack of backward-in-time integration limits the applicability of splitting methods to stochastic dynamics and leads to a relatively high risk of producing nonrepresentative transition trajectories [54]. The other three methods, TPS, TIS, and RETIS, are all based on a MC sampling procedure in path space. The original TPS approach for rate calculations is no longer being used in practice as TIS is both faster and more accurate than TPS. RETIS is even more efficient than TIS, but requires a more complex implementation. By the emergence of open source path sampling libraries like OPS [55,56] and PyRETIS [57,58] the latter aspect has become less of an issue.

In this paper, we show how RETIS can effectively be used to compute permeability coefficients equally exact as the direct counting approach, but orders of magnitude faster. In the next section, we shortly introduce the RETIS approach, then show how it can be amended for permeability calculations in Sec. IV, and show in Sec. VI two new path MC moves that can further enhance ergodic sampling.

### III. REPLICA EXCHANGE TRANSITION INTERFACE SAMPLING

TIS and RETIS are rare event techniques that allow to compute rate constants  $k$  when transitions have to overcome a high free energy barrier, and thus transitions are unlikely to be observed in a standard MD simulation. Intuitively the rate constant  $k$  can be expressed as a number of transitions, from reactant state to product state, per unit time and per amount of reactants. However, the translation of this phenomenological rate constant into a computational measurable property is far from trivial since it requires a microscopic definition of the reactant state and product state. If a single dividing surface is used to assign whether a molecular system is in the reactant

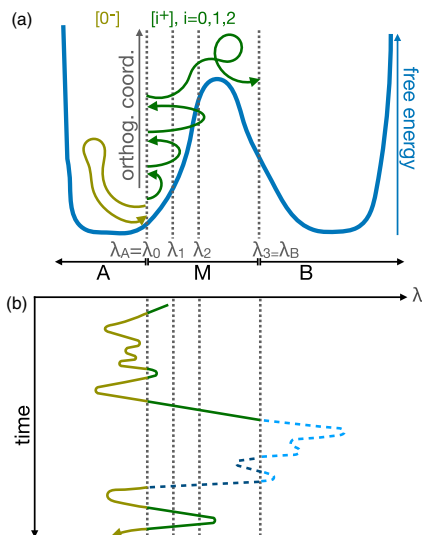


FIG. 1. Path ensembles in RETIS. (a) Paths shown along the RC  $\lambda$  and an arbitrary orthogonal coordinate. The free energy as a function of  $\lambda$  is also shown. All top four paths are part of the  $[0^+]$  ensemble containing paths that start at  $\lambda_A = \lambda_0$ , move in the positive direction and end at  $\lambda_A$  or  $\lambda_B$ . The top three path are part of the  $[1^+]$  ensemble. These are like the  $[0^+]$  paths but should cross  $\lambda_1$  in addition. The top two paths are part of the  $[2^+]$  path ensemble. The bottom path is a  $[0^-]$  path that starts at  $\lambda_A = \lambda_0$ , moves in the negative direction and ends at  $\lambda_A$ . The stable state regions A and B, and the middle region M (no man's land) are shown. Points in the region M can be part of the larger overall states A or B. (b) Reaction coordinate as a function of time for a hypothetical long equilibrium MD run. The line is solid green (light: stable state A, dark: region M) when the system belongs to overall state A. It is dashed blue (bright: stable state B, dark: region M) when part of overall state B.

state or product state, e.g., based on a geometric observable being lower or higher than a specific value, it is expected to observe many correlated recrossings of the dividing surface. Only when the system has moved far enough beyond the dividing surface, it can be considered as stabilized to the other state. To avoid the issue of correlated recrossings, it can be generally better to consider two separate dividing surfaces left and right of the barrier,  $\lambda_A$  and  $\lambda_B$ , respectively [see Fig. 1(a)]. At the left of  $\lambda_A$  and at the right of  $\lambda_B$ , the system is considered committed to the reactant state and product state respectively. The disadvantage is that the barrier region between the dividing surfaces is not assigned to either state. A transition from reactant state to product state needs to cross the *no man's land* between A and B, which makes it difficult to assign for each transition a specific point in time at which it takes place, which is essential to avoid overcounting.

The TIS and RETIS approaches circumvent this problem by introducing *overall states* which are history-dependent state definitions. Using the two dividing surfaces,  $\lambda_A$  and  $\lambda_B$ , the system is part of *stable state A* if the value of the chosen RC is below  $\lambda_A$  and it is part of *stable state B* if it is higher

than  $\lambda_B$ . The overall states are denoted by the curly letters  $\mathcal{A}$  and  $\mathcal{B}$ , and they include the stable states  $A$  and  $B$ , respectively. Further, any instance of the system traveling in no man's land is assigned to *overall state*  $\mathcal{A}$  or *overall state*  $\mathcal{B}$  based on the stable state that was most recently visited.

The advantage is that the system is always assigned to either overall state  $\mathcal{A}$  or  $\mathcal{B}$ , and the overall state regions are rather insensitive to the placement of the dividing surfaces  $\lambda_A$  and  $\lambda_B$  as long as these are reasonable [23]. In addition, a transition from  $\mathcal{A}$  to  $\mathcal{B}$  is well defined without hindrance of recrossings and leads to a microscopically measurable net flux without perturbing the equilibrium conditions. As such, the phenomenological rate constant can be expressed as

$$k = \lim_{\Delta t \rightarrow 0} \frac{\langle h_{\mathcal{A}}(0)h_{\mathcal{B}}(\Delta t) \rangle}{\langle h_{\mathcal{A}} \rangle \Delta t}, \quad (6)$$

where  $h_X(t)$  equals 1 when the system is in state  $X$  at time  $t$  and 0 when the system is not in state  $X$  at time  $t$ . The interpretation of the ensemble average that involves history dependent functions is given in Appendix A.

The above equation counts the rare crossings from left to right through the surface  $\lambda_B$  of points that actually come from  $\lambda_A$  (when the equations of motion are followed backward in time,  $\lambda_A$  is reached without crossing  $\lambda_B$ ). Since we can assume equilibrium, the same number of counts per second will be obtained by considering crossings with  $\lambda_A$  that after crossing that interface reach  $\lambda_B$  without recrossing  $\lambda_A$ . This allows us to write the rate  $k$  as

$$k = f_A P_A(\lambda_B|\lambda_A), \quad (7)$$

where  $f_A$  is the conditional flux (crossings per time) through  $\lambda_A$  counting all crossings in the positive direction per time spent in state  $\mathcal{A}$ , and  $P_A(\lambda_B|\lambda_A)$  is the crossing probability, i.e., the chance that a crossing with  $\lambda_A$  is followed by a crossing with  $\lambda_B$  before a recrossing with  $\lambda_A$  occurs.

$f_A$  can easily be computed with MD, as is done in TIS, since crossing  $\lambda_A$  is not a rare event. In RETIS, the flux is computed from the average path lengths of two path ensembles. The crossing probability  $P_A(\lambda_B|\lambda_A)$  is generally too low to be computed by straightforward MD, but can be recast into the following factorization by defining a set of nonintersecting interfaces:

$$P_A(\lambda_B|\lambda_A) = P_A(\lambda_n|\lambda_0) = \prod_{i=0}^{n-1} P_A(\lambda_{i+1}|\lambda_i). \quad (8)$$

Here,  $P_A(\lambda_{i+1}|\lambda_i)$  is the history dependent conditional probability that, given there is a crossing with interface  $\lambda_i$  for the first time since last  $\lambda_A = \lambda_0$  crossing, interface  $\lambda_{i+1}$  will be crossed before  $\lambda_A$ . These conditional probabilities are computed in  $n - 1$  different path sampling simulations. Each simulation samples a different path ensemble using a set of different MC moves to generate new paths in the ensembles. The ensemble  $[i^+]$  consists of all possible paths that start at  $\lambda_A$  and end at  $\lambda_A$  or  $\lambda_B$  and have at least one crossing with  $\lambda_i$ . The MC approach is tuned such that the same statistical distribution of paths is generated as the distribution of paths that would result if these are cut out from a hypothetical extremely long MD simulation. The fraction of paths in the  $[i^+]$  path ensemble that cross  $\lambda_{i+1}$  equals  $P_A(\lambda_{i+1}|\lambda_i)$ . Performance of

TIS is optimal when the number of interfaces and the spacing between the interfaces are tuned such that each conditional probability is around 0.2 [59]. The most important MC move is the so-called shooting move in which a random point of the previous path is picked, its velocities are randomly modified to generate a new phase point, and from this phase point the equations of motion are followed backward and forward in time to generate a new path. The new path will be accepted if it fulfills the requirements for the specific ensemble (like crossing  $\lambda_i$  in the  $[i^+]$  ensemble) and, depending on the type of velocity randomization procedure, an additional Metropolis acceptance/rejection step will be invoked. In this work, we applied the aimless velocity modification in which velocities are regenerated, independent from the old velocities, from a Maxwell-Boltzmann distribution [60].

Compared to TIS, RETIS has one additional path ensemble called  $[0^-]$ . Like the other ensembles, this ensemble contains paths starting at  $\lambda_0$ , but from there the paths move in the negative direction away from the barrier. The paths are terminated when they cross  $\lambda_0$  again. While the flux  $f_A$  in Eq. (7) is computed with straightforward MD in TIS, in RETIS it is computed from the average path lengths in the  $[0^-]$  and  $[0^+]$  ensembles (see Sec. IV A).

Another difference between RETIS and TIS, is that RETIS employs additional MC moves. Since RETIS is purely based on path sampling simulations, instead of MD and path sampling simulations, replica exchange moves between the different path ensembles (parallel path swapping [22]) can be applied throughout the complete RETIS simulation. The swapping moves enhance the sampling in a similar way as parallel tempering [61], since the  $[i^+]$  path ensemble for a given  $i$  tends to contain trajectories moving on a higher energy surface than the trajectories of the  $[(i-1)^+]$  ensemble. The higher energy trajectories are less likely to get trapped in specific reaction channels that are separated by free energy barriers orthogonal to the RC  $\lambda$  [62]. As paths between  $[(i-1)^+]$  and  $[i^+]$  are sometimes swapped, also the  $[(i-1)^+]$  path ensemble will sample these different reaction channels more easily. The combination of TPS and standard parallel tempering can provide a similar effect [63], though in RETIS there are no additional simulations at alleviated temperatures needed. In contrast, the  $[(i-1)^+] \leftrightarrow [i^+]$  swapping move is very inexpensive as it does not require any force evaluations. If the move is accepted, it provides a new path for both the  $[(i-1)^+]$  and the  $[i^+]$  path ensemble. In addition, the accepted swapping moves provide generally paths that are more decorrelated from the previous path than a shooting move.

The swapping moves between the  $[0^-]$  and  $[0^+]$  path ensembles are done by exchanging the end and start points of the paths and extending those forward and backward in time respectively. Despite not being a free move, the barrierless diffusion within the reactant well of the  $[0^-]$  paths, followed by exchange moves, will basically feed the  $[0^+]$  ensemble with fresh initializations. This facilitates the decorrelation of the MC sampling even further and orthogonal barriers can be avoided without having to cross them in any of the path ensembles. An additional advantage is that this method works in case where parallel tempering is not effective, that is when barriers are mainly entropic in nature.



#### IV. PERMEABILITY FROM RETIS SIMULATIONS

The rate  $k$  in Eq. (6) (with unit 1/time) describes the kinetics of a process, while the permeability  $P$  of a membrane (with unit length/time) describes the transport kinetics through the layer, and clearly a link between  $k$  and  $P$  is expected. However not only the units, but also the formalism for permeability and rates are somewhat different. This paper will derive the correct connection between the TIS framework for rare events and the permeability.

The progression of permeation is most easily measured by the  $z$  coordinate orthogonal to the membrane as the RC. Despite this simple order parameter choice, some details ask for attention. In case of periodic boundary conditions, common in MD simulations of membranes or porous catalytic crystals, the question arises how it may be detected whether a molecule crossed the membrane, or whether it simply circled back to the other side of the membrane through the water phase because of the periodic boundary condition in the  $z$ -direction. Therefore we first derive the connection between  $P$  and  $k$  when the solvent phase is bounded on both sides (e.g., by hard walls as in Fig. 1) in Sec. IV A, and second we derive a relation for the unbounded system when either periodic boundary conditions or an infinite particle bath [64] are applied in Sec. IV B.

##### A. Connecting permeability and rate

Figure 1(b) shows a hypothetical long MD equilibrium run. The timescales are not very realistic as we would in practice expect that thousands of crossings with  $\lambda_A$  would proceed a transition to state  $B$ . Yet, it will be used to show how we can subdivide an ensemble average into different regions  $A$ ,  $B$  or  $A, B$  as follows.

The long trajectory can be seen as a series of visited phase points. Here, we assume that the MD equilibrium run is in fact sampling the distribution of interest. The MD integrator is hence coupled to some kind of thermostat or barostat, whenever this distribution is different from the NVE ensemble. The MD integrator gives a trajectory, which is effectively a series of phase points (time slices) because of the discrete time step  $\Delta t$  in the numerical integration, such as the velocity Verlet integration algorithm [65]. Given ergodicity, the phase points will be visited with the correct relative probability when the trajectory length  $T$  goes to infinity. The trajectory phase points can be divided into subsets corresponding to the regions  $A$ ,  $M$ ,  $B$  in Fig. 1(b) based on the value of the order parameter in each phase point. The ensemble average becomes

$$\langle \dots \rangle = p_A \langle \dots \rangle_A + p_M \langle \dots \rangle_M + p_B \langle \dots \rangle_B, \quad (9)$$

where  $M$  is the membrane region, previously referred to as no man's land in the context of TIS and RETIS. Here,  $p_X$  is the probability that the system is in state  $X$ . Given that the ergodicity makes the time averages respect the relative probabilities, it follows that  $p_X = T_X/T$ , where  $T_X$  is the total time spent in state  $X$ , and  $T = \sum_X T_X$ . In Eq. (9), the notation  $\langle \dots \rangle_A$  refers to the ensemble average over all trajectory phase points associated to  $A$ .

As mentioned earlier in Sec. III, the trajectory phase points can also be assigned to either overall state  $\mathcal{A}$  or  $\mathcal{B}$ . The assignment to  $\mathcal{A}$  or  $\mathcal{B}$  is generally not based on the evaluation of

the order parameter in a single phase point but rather based on the series of phase points in the trajectory (history dependent). The ensemble average can be divided in two contributions,

$$\langle \dots \rangle = p_{\mathcal{A}} \langle \dots \rangle_{\mathcal{A}} + p_{\mathcal{B}} \langle \dots \rangle_{\mathcal{B}}. \quad (10)$$

The notation  $\langle \dots \rangle_{\mathcal{A}}$  refers to the ensemble average over all trajectory phase points associated to  $\mathcal{A}$  (see Appendix A).

Central to the RETIS methodology are the  $[i^+]$  path ensembles connected with interfaces at locations  $\lambda_i$ ,  $i = 0, 1, 2, \dots$  and the  $[0^-]$  path ensemble [see Fig. 1(a)]. Note that any path in the  $[i^+]$  ensemble for  $i > 0$  is automatically a valid path in the  $[0^+]$  ensemble. The  $[0^-]$  paths completely lie in state  $A$  except for the first and last points of the trajectories. In addition, all phase points inside  $A$  must lie on a  $[0^-]$  path. We can therefore say that state  $A$  is equivalent to  $[0^-]$ . That is, we can view each path ensemble also as a phase space ensemble by simply collecting all phase space points that lie on corresponding paths except for the terminating points. The  $[0^-]$  and  $[0^+]$  ensemble (excluding the end points) are disjunct and combined they represent all trajectories that pass through the  $\lambda_0$  interface while not having reached state  $B$  yet. The  $\mathcal{A}$  ensemble is therefore equal to the combined  $[0^-]$  and  $[0^+]$  ensembles.

As is obvious from Fig. 1(b), an equal number of  $[0^-]$  as  $[0^+]$  trajectories can be cut out from the equilibrium run; each time the end point of a  $[0^-]$  path comprises a start point of  $[0^+]$  path and, vice versa, each starting point of a  $[0^+]$  path relates to an end point of a  $[0^-]$  path. When  $N_X$  denotes the number of paths in the  $X$  ensemble that can be cut out from a long equilibrium MD run, this observation can be summarized as  $N_{[0^-]} = N_{[0^+]}$  in thermodynamic equilibrium.

Let us compare the three different flux terms:  $k$  in Eq. (6),  $f_A$  in Eq. (7) and  $J^+$  in Eq. (2). Here,  $f_A$  provides the frequency of state changes from  $A$  to  $M$  within the overall state ensemble  $\mathcal{A}$ .  $J^+$  on the other hand measures all crosses from left to right along the full region  $M$  of all permeants, per time and per membrane surface area  $\sigma$ . The rate constant  $k$  finally also measures the frequency of full crossings like  $J^+$  but with the same time normalization  $T_A$  as for  $f_A$ . In summary,

$$\begin{aligned} f_A &= \frac{\#(A \rightarrow M)_{\text{target}}}{T_A}, \\ k &= \frac{\#(A \rightarrow M \rightarrow B)_{\text{target}}}{T_A}, \\ J^+ &= \frac{\#(A \rightarrow M \rightarrow B)_{\text{all perm}}}{T\sigma}. \end{aligned} \quad (11)$$

One could imagine another flux definition, similar to  $f_A$ , but where the denominator is the total time  $T$  of the long equilibrium run rather than the time spent in  $\mathcal{A}$ ,

$$f = \frac{\#(A \rightarrow M)_{\text{target}}}{T} = p_A f_A. \quad (12)$$

The appearance of the factor  $p_A$  gives  $f_A$  the interpretation of a "conditional" flux compared to  $f$ . The flux  $f$  is however not accessible in a RETIS simulation and will not be further discussed.

If all of the  $N_p$  permeating particles are identical,

$$\#(A \rightarrow M \rightarrow B)_{\text{all perm}} = N_p \#(A \rightarrow M \rightarrow B)_{\text{target}} \quad (13)$$

and we can relate  $J^+$  with  $f_A$  and  $k$  as follows:

$$\begin{aligned} J^+ &= N_p \frac{\#(A \rightarrow M \rightarrow B)_{\text{target}} T_A}{T_A \sigma} \frac{T_A}{T} \\ &= \frac{N_p k p_A}{\sigma} = \frac{N_p f_A P_A(\lambda_B|\lambda_A) p_A}{\sigma}, \end{aligned} \quad (14)$$

where  $P_A(\lambda_B|\lambda_A)$  is the previously introduced crossing probability. All terms in Eq. (14) are measurable in a RETIS simulation except  $p_A$ , the probability of overall state  $\mathcal{A}$ . RETIS simulates the  $\mathcal{A}$  ensemble (through  $[0^-]$  and  $[0^+]$ ) but not the  $\mathcal{B}$  ensemble; hence  $T_B$ ,  $T = T_A + T_B$  and ultimately  $p_A = T_A/T$  are not known. Luckily, the factor  $p_A$  cancels when we include the  $c_{\text{ref}}$  concentration in order to compute the permeability based on Eq. (2), as we will show now.

First, let  $\rho_{\text{ref}}$  be the one-dimensional probability density to find a specific permeant (for instance the targeted permeant) at the reference location  $z_{\text{ref}}$ . We can relate  $\rho_{\text{ref}}$  to  $c_{\text{ref}}$  by taking into account the number of permeants  $N_p$  in the simulation box and the cross section area  $\sigma$ ,

$$\rho_{\text{ref}} = \frac{\sigma c_{\text{ref}}}{N_p}. \quad (15)$$

Further, we can use the subdivision in global states in Eq. (10), giving

$$\begin{aligned} \rho_{\text{ref}} &= \rho(z_{\text{ref}}) = \langle \delta(z_{\text{ref}} - z_t) \rangle \\ &= p_A \langle \delta(z_{\text{ref}} - z_t) \rangle_{\mathcal{A}} + p_B \langle \delta(z_{\text{ref}} - z_t) \rangle_{\mathcal{B}} \\ &= p_A \langle \delta(z_{\text{ref}} - z_t) \rangle_{\mathcal{A}} = p_A (\rho_{\text{ref}})_{\mathcal{A}}, \end{aligned} \quad (16)$$

where  $z_t$  is the  $z$  coordinate of the target permeant, and  $(\rho_{\text{ref}})_{\mathcal{A}}$  refers to the probability density within the  $\mathcal{A}$  ensemble. The factor  $\langle \delta(z_{\text{ref}} - z_t) \rangle_{\mathcal{B}}$  is zero, since  $z_{\text{ref}}$  is located in stable state  $A$  and  $z_t$  does not lie in  $A$  for any trajectory phase point that is assigned to  $\mathcal{B}$  (see Appendix A). Consequently, we can write

$$c_{\text{ref}} = \frac{N_p \rho_{\text{ref}}}{\sigma} = \frac{N_p (\rho_{\text{ref}})_{\mathcal{A}} p_A}{\sigma}. \quad (17)$$

Substitution of Eqs. (17) and (14) in Eq. (2) gives

$$P = \frac{k}{(\rho_{\text{ref}})_{\mathcal{A}}} = \frac{f_A P_A(\lambda_B|\lambda_A)}{(\rho_{\text{ref}})_{\mathcal{A}}}. \quad (18)$$

This is the first theoretical result connecting  $P$  and  $k$ . Equation (18) shows that the  $\mathcal{B}$  path ensemble needs not be simulated to find the permeability. In contrast, the direct counting method of Eq. (2) is based on a long MD run, which includes paths in both the  $\mathcal{A}$  and  $\mathcal{B}$  ensembles. The paths in the  $\mathcal{A}$  ensemble are however sufficient for the computation of the permeability with Eq. (18).

Still, the denominator  $(\rho_{\text{ref}})_{\mathcal{A}}$  in Eq. (18) cannot be computed in a single path ensemble since overall state  $\mathcal{A}$  comprises both  $[0^-]$  and  $[0^+]$ . Therefore let us first consider how the probability density at the reference location would be computed from a long equilibrium MD simulation. We would need to define a certain interval around  $z_{\text{ref}}$  with a width  $\Delta z$ ,  $[z_{\text{ref}} - \Delta z/2, z_{\text{ref}} + \Delta z/2]$ , and  $(\rho_{\text{ref}})_{\mathcal{A}}$  is the ratio of the average time  $T_{\text{ref}}$  spent in the reference interval region versus the total time spent in  $\mathcal{A}$ , divided by  $\Delta z$ ,

$$(\rho_{\text{ref}})_{\mathcal{A}} = \frac{T_{\text{ref}}}{T_A} \frac{1}{\Delta z}. \quad (19)$$

In practice, one would count the number of steps that the  $z$  coordinate is inside the interval, and divide this by  $\Delta z$  and by the total number of steps that the system is part of  $\mathcal{A}$  paths, assuming a constant MD integration time step  $\Delta t$ . If the reference location is in a region where the free energy profile  $F(z)$  is flat, then  $T_{\text{ref}}$  scales linearly with  $\Delta z$ , and  $(\rho_{\text{ref}})_{\mathcal{A}}$  will in principle not be affected by the chosen interval length  $\Delta z$ .

On the other hand, the conditional flux is the number of crossings through  $\lambda_A$  in the positive direction divided by the time spent in  $T_A$ ,

$$f_A = \frac{N_{[0^+]}}{T_A}, \quad (20)$$

where, as said earlier,  $N_{[0^+]}$  is the number of  $[0^+]$  trajectories that can be cut out from the long equilibrium trajectory. Substitution of the two previous equations into the permeability in Eq. (18) makes the  $T_A$  drop out, and gives a practical expression for  $P$ ,

$$P = \frac{N_{[0^+]}}{T_{\text{ref}}} P_A(\lambda_B|\lambda_A) \Delta z. \quad (21)$$

This is the second expression linking  $P$  with RETIS quantities. It gives the insight that  $P$  depends on time spent in the reference region, but not explicitly on time spent in the  $[0^+]$  nor  $[0^-]$  ensemble.

However, Eq. (21) still refers to the quantities  $N_{[0^+]}$  and  $T_{\text{ref}}$ , which are obtained from a long equilibrium simulation. In the next last step, the conversion to path ensemble averages is made. With  $T_X$  the time spent in a path ensemble  $X$  and  $N_X$  the number of trajectories in the path ensemble  $X$ , the average path length is given by  $\tau_X = T_X/N_X$ . An advantage of  $\tau_X$  is that it is in principle independent of the simulation computer time, i.e., if the number of simulated paths in the ensemble is doubled, the average  $\tau_X$  will not change, which gives  $\tau_X$  an intrinsic meaning. We would like to stress that a path ensemble average like  $\tau_X$  is a property that is averaged over all paths in the path ensemble, and it differs from the common phase space average, denoted as  $\langle \dots \rangle_Y$ , where the average is taken over all phase points within the ensemble  $Y$ .

The conversion to path averages is now simply executed by introducing factors  $N_{[0^-]}$ , using  $T_A = T_{[0^-]} + T_{[0^+]}$ , and exploiting the fact that  $N_{[0^-]} = N_{[0^+]}$  in an equilibrium run. For the probability density, this gives

$$\begin{aligned} (\rho_{\text{ref}})_{\mathcal{A}} &= \frac{T_{\text{ref}}}{T_{[0^-]} + T_{[0^+]}} \frac{1}{\Delta z} \\ &= \frac{T_{\text{ref}}/N_{[0^-]}}{T_{[0^-]}/N_{[0^-]} + T_{[0^+]}/N_{[0^+]}} \frac{1}{\Delta z} \\ &= \frac{\tau_{\text{ref},[0^-]}}{\tau_{[0^-]} + \tau_{[0^+]}} \frac{1}{\Delta z}, \end{aligned} \quad (22)$$

while the conditional flux is converted to (see Ref. [22])

$$\begin{aligned} f_A &= \frac{N_{[0^+]}}{T_{[0^-]} + T_{[0^+]}} \\ &= \frac{1}{T_{[0^-]}/N_{[0^-]} + T_{[0^+]}/N_{[0^+]}} \\ &= \frac{1}{\tau_{[0^-]} + \tau_{[0^+]}}. \end{aligned} \quad (23)$$

Here,  $\tau_{[0^-]}$  and  $\tau_{[0^+]}$  are the average lengths of paths in the  $[0^-]$  path ensemble and  $[0^+]$  path ensemble, respectively, while  $\tau_{\text{ref},[0^-]}$  is the average time spent in the  $[z_{\text{ref}} - \Delta z/2 : z_{\text{ref}} + \Delta z/2]$  interval per path in the  $[0^-]$  ensemble.

Again, substituting the previous equations into the permeability in Eq. (18) gives another expression for  $P$ ,

$$P = \frac{k\Delta z}{\tau_{\text{ref},[0^-]}f_A} = \frac{P_A(\lambda_B|\lambda_A)\Delta z}{\tau_{\text{ref},[0^-]}}. \quad (24)$$

This is the third theoretical result expressing  $P$  in terms of intrinsic RETIS quantities. Equation (24) is an expression that can be fully computed in a RETIS simulation since  $P_A(\lambda_B|\lambda_A)$  is standard output of this approach, while  $\tau_{\text{ref},[0^-]}$  can easily be obtained from the paths generated in the  $[0^-]$  ensemble by histogramming the  $z$  coordinate.

Figure 1 is however not typical for an actual membrane system that has no natural confining energy barrier that prohibits the permeants to drift far away from the membrane. To deal with the situation of an unconfined system, we derive in Sec. IV B how a  $\lambda_{-1}$  interface can be used to constrain the system in a way that the permeability coefficient can still be determined exactly, as if the system were unrestrained.

### B. The $\lambda_{-1}$ interface

The previous section linked the rate constant to a permeability, though the system depicted in Fig. 1 is not exemplary for a permeation system where we imagine a membrane in a near-infinite solvent. The rate constant  $k$  that follows a single particle will naturally decrease with the amount of solvent added to the model system since the target particle will spend more time away from the membrane. The permeability  $P$  is insensitive to this since it does not depend on the size of the simulation box. Increasing the solvent while maintaining constant concentration automatically implies an increase in the number of permeants. This increase cancels the decrease of the rate for permeation of the individual particles. Still, as RETIS computes rates rather than permeation directly, some confinement is required in practice.

This confinement is achieved by introducing an extra interface  $\lambda_{-1}$ . With the introduction of this interface the overall state  $\mathcal{A}$  is reduced to the  $\lambda > \lambda_{-1}$  region, and is denoted  $\mathcal{A}'$ . The same can be done at the product region side, but this is not essential for the  $A \rightarrow B$  rate calculation. The new state division is depicted in Fig. 2, where the outer regions are here called *freeze-time zones* which will not be accessed in this adaptation of the RETIS algorithm. A prime is added to show that the states  $\mathcal{A}'$ ,  $\mathcal{B}'$ ,  $\mathcal{A}'$ , and  $\mathcal{B}'$  contain fewer phase points than  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $\mathcal{A}$ , and  $\mathcal{B}$ , respectively. The ensemble average in Eq. (10) is updated to reflect the additional *freeze-time zones*,

$$\langle \dots \rangle = p_{\text{fr-ti}} \langle \dots \rangle_{\text{fr-ti}} + p_{\mathcal{A}'} \langle \dots \rangle_{\mathcal{A}'} + p_{\mathcal{B}'} \langle \dots \rangle_{\mathcal{B}'}. \quad (25)$$

The path ensemble  $[0^-]$  resembles the  $[0^-]$  ensemble, but contains paths that can start and end at  $\lambda_{-1}$  in addition to  $\lambda_0$ . The time slices of paths in path ensembles  $[0^-]$  and  $[0^+]$  fully build up the overall state  $\mathcal{A}'$ . The freeze-time zone on the product side does not affect the  $[0^+]$  ensemble nor the other  $[i^+]$  ensembles. The (RE)TIS crossing probability  $P_A(\lambda_B|\lambda_A)$  remains unaffected by the  $\lambda_{-1}$  interface.

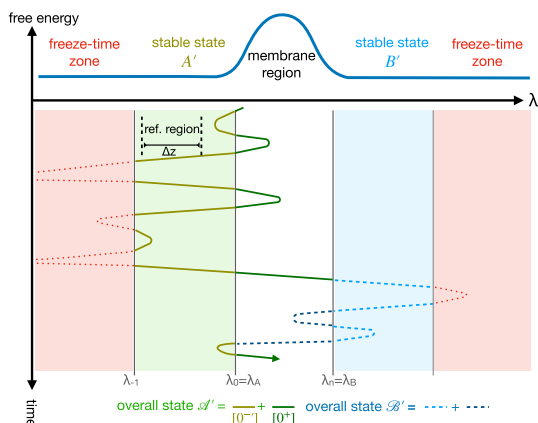


FIG. 2. Illustration of how the *freeze-time zones* in red are introduced by extra interfaces left of  $\lambda_A$  (called  $\lambda_{-1}$ ) and right of  $\lambda_B$  (unnamed). The green and blue regions refer to states  $\mathcal{A}'$  and  $\mathcal{B}'$ , respectively. The line represents a long equilibrium run (RC  $\lambda$  versus time  $T$ ). Note that in the membrane region (previously called *no man's land*) the color is given by the stable states last visited [like in Fig. 1(b)]. Whenever the system enters a freeze-time zone, the time  $T'$  is stopped and continued whenever it exits this zone. The trajectories that can be cut out from the  $[\lambda_{-1}, \lambda_0]$  interval constitute the  $[0^-]$  path ensemble. This ensemble is different from the  $[0^-]$  ensemble since its paths can start and end at  $\lambda_{-1}$  in addition to  $\lambda_0$ .

Figure 2 shows again a hypothetical unrestrained equilibrium MD run just like in Fig. 1(b) with a free energy surface that is flat at either side of the membrane. Conceptually, the existence of an equilibrium in such an infinite system is not obvious as the ergodicity hypothesis implies that for instance ensemble averages are identical to time averages of an infinite equilibrium run. However, if the partition function diverges, even an infinitely long equilibrium run might not visit all the relevant phase space regions. This issue can be solved conceptually by taking the infinite limits for space and time in a controlled way. That is, we consider the potential of Fig. 2 as a special case of the potential shown in Fig. 1, but with the vertical-like increase of the free energy occurring at  $z = -W$  and  $+W$ . By letting  $T \rightarrow \infty$  and  $W \rightarrow \infty$  such that  $W^2/(TD) \rightarrow 0$  with  $D$  the average diffusion constant, it can be shown that the ergodicity hypothesis holds. While this solves the conceptual problem whether we can assume equilibrium statistics, it does not solve the practical problem that the rate  $k$  is still zero in this limit.

The additional  $\lambda_{-1}$  interface, however, ensures that the overall state  $\mathcal{A}'$  becomes finite and that anything that happens in the freeze-time zone can be ignored as if the stopwatch, measuring  $T'$ , is paused and statistics are not updated each time that the trajectory enters that region.

Now that the rate  $k'$  and the flux  $f_{\mathcal{A}'}$  can be computed from this long equilibrium run using the new definition for the overall state  $\mathcal{A}'$ , the same counting strategy applies: it is a number of crossings/transitions divided by the time spent in overall state  $\mathcal{A}'$ . The only difference is that, besides time spent in overall state  $\mathcal{B}'$ , also the freeze-time zone is ignored in the

normalization. The path ensemble  $[0^-]$  is however changed to  $[0^-]$ , which statistically corresponds to the same ensemble one would obtain by cutting out the trajectory segments between  $\lambda_{-1}$  and  $\lambda_0$  from the equilibrium run. Specifically, it is no longer valid that  $N_{[0^-]}$  and  $N_{[0^+]}$  are equal. We will therefore introduce the parameter  $\xi$  to measure the mismatch between number of paths in these two path ensembles,

$$\xi = \frac{N_{[0^+]}}{N_{[0^-]}} = \frac{N_{\rightarrow R, [0^-]}}{N_{[0^-]}} = \overline{h_{\rightarrow R} [0^-]}. \quad (26)$$

Here  $N_{\rightarrow R, [0^-]}$  is the number of  $[0^-]$  paths ending at the right side (at  $\lambda_0$ ) that can be cut out of the long equilibrium run. It is obvious that  $N_{\rightarrow R, [0^-]} = N_{[0^+]}$  from observing the  $\lambda_A$  interface in Fig. 2. It follows that  $\xi < 1$ .

Similarly,  $h_{\rightarrow R}$  is the characteristic function that for each path provides the output 1 if the path ends at the right and 0 if the path ends at the left, irrespective of the starting point. Finally,  $\overline{h_{\rightarrow R} [0^-]}$  is the average of this function over all paths in the  $[0^-]$  ensemble. This parameter  $\xi$  can hence be obtained from a RETIS simulation by the analysis of the  $[0^-]$  path ensemble.

In order to reformulate the expressions for  $f_{A'}$  and  $P$ , we first update the expressions for  $T_{A'}$  and  $T_{\text{ref}}'$  to link them to average path lengths  $\tau_X$ , which are intrinsic quantities of a given path ensemble.

The time  $T_{A'}$  can be written to be proportional to  $N_{[0^+]}$  as

$$\begin{aligned} T_{A'} &= T_{[0^-]} + T_{[0^+]} = N_{[0^-]} \tau_{[0^-]} + N_{[0^+]} \tau_{[0^+]} \\ &= N_{[0^+]} (\xi^{-1} \tau_{[0^-]} + \tau_{[0^+]}) \end{aligned} \quad (27)$$

The time  $T_{A'}$  is smaller than  $T_A$  since  $A'$  comprises fewer phase points than  $A$  (the clock is stopped). The time  $T_{\text{ref}}$  spent in the reference interval is not affected by the presence of the  $\lambda_{-1}$  as the reference interval is located between  $\lambda_{-1}$  and  $\lambda_A$ . The associated intrinsic quantity  $\tau_{\text{ref}, [0^-]}$  can nevertheless change as  $N_{[0^-]}$  might differ from  $N_{[0^+]}$  (e.g., seven paths versus four paths in Fig. 2),

$$\begin{aligned} T_{\text{ref}} &= N_{[0^-]} \tau_{\text{ref}, [0^-]} \\ &= N_{[0^+]} \xi^{-1} \tau_{\text{ref}, [0^-]} \end{aligned} \quad (28)$$

These two equations have implications for  $f_{A'}$  and  $P$ . For  $f_{A'}$  in Eq. (20), the number of paths  $N_{[0^+]}$  remains unaltered according to Fig. 2, but the time  $T_A$  needs to be updated with  $T_{A'}$ , leading to

$$f_{A'} = \frac{N_{[0^+]}}{T_{A'}} = \frac{\xi}{\tau_{[0^-]} + \xi \tau_{[0^+]}}. \quad (29)$$

For  $P$  in Eq. (21), the number of paths  $N_{[0^+]}$  and the crossing probability  $P_A(\lambda_B|\lambda_A)$  remain unaltered, and the time  $T_{\text{ref}}$  is updated with Eq. (28), leading to the generalized version of Eq. (24),

$$P = \frac{\xi \Delta z}{\tau_{\text{ref}, [0^-]}} P_A(\lambda_B|\lambda_A). \quad (30)$$

The last expression is the central expression that links the permeability with thermodynamic averages that can be computed in a RETIS path sampling simulation. The parameter  $\xi$  is obtained from analyzing the end points of the  $[0^-]$  paths, the crossing probability  $P_A(\lambda_B|\lambda_A)$  is a direct output of a (RE)TIS simulation,  $\Delta z$  is a chosen bin width of the reference region

in the flat free energy region to the left of  $\lambda_0$ , and  $\tau_{\text{ref}, [0^-]}$  is the corresponding time spent per path in the  $[0^-]$  ensemble in this reference bin. The latter can be rewritten as

$$\tau_{\text{ref}, [0^-]} = \frac{T_{\text{ref}}}{T_{A'}} = (\rho_{\text{ref}})_{A'} \Delta z \tau_{[0^-]}, \quad (31)$$

which leads to an alternative expression for the permeability

$$P = \frac{\xi P_A(\lambda_B|\lambda_A)}{(\rho_{\text{ref}})_{A'} \tau_{[0^-]}}. \quad (32)$$

Equation (32) combines path ensemble averages ( $\xi$ ,  $P_A(\lambda_B|\lambda_A)$ ,  $\tau_{[0^-]}$ ) with a phase space average ( $(\rho_{\text{ref}})_{A'}$ ), while Eq. (30) is only based on path ensemble averages.

All quantities in Eqs. (30) and (32) are intrinsic, meaning they are expressed as averages over the paths. This means that the same expression is applicable when doubling the number of paths, i.e., running the MC steps in the path ensembles longer. This will not affect the absolute value of any of the terms provided in Eq. (30), but will naturally increase the accuracy of their numerical estimates.

## V. RC FOR PERMEATION WITH PERIODIC BOUNDARY CONDITIONS

In the previous section, we solved the problem to link the permeability with the rate constant in an infinite system by introducing the interface  $\lambda_{-1}$  and the rate constant  $k'$ , which is nonzero unlike  $k$ . In most practical simulations the infinite system is represented by a system with periodic boundary conditions (PBC) which poses the need to properly determine the relative position of the permeant with respect to the center of the membrane, which defines the RC,  $\lambda$ . The simple minimum-image convention will generally not work since the RETIS trajectories can span the full  $[\lambda_{-1}, \lambda_B]$  region and a permeant located at  $\lambda_{-1}$  might actually be closer to the left periodic image of the membrane than to the membrane in the central image.

In order to properly deal with PBC, we first map the  $z_i$  coordinates of all particles  $i$  in the system within the  $[-L_z/2, L_z/2]$  interval, where  $z = 0$  is matched by convention at the center of mass of the membrane and  $L_z$  is the box length in the  $z$  dimension,

$$z_i = (z'_i - z'_{\text{mem.}}) - \text{round}\left(\frac{z'_i - z'_{\text{mem.}}}{L_z}\right) L_z, \quad \forall i. \quad (33)$$

Here,  $z'_i$  is the  $z$  coordinate of particle  $i$  provided by the molecular simulation program. By this operation, the center of mass of the membrane is set at 0, while the center of the solvent slab is at  $\pm L_z/2$ . Here we assume that original coordinates  $z'_j$  of the membrane particles  $j$  are constructed such that the center of mass at the  $z'$  axis can be computed without the need to add or subtract  $L_z$  or multiples of it to any of the membrane particles. As the membrane is stable, we can assume that the above remains valid during the full simulation. That is, membrane particles might move over large distances only if all membrane particles move in cohort.

The RC is given by the relative position  $z_i$  of the tagged permeant  $i$  (target) with respect to the membrane. This implies that if we want to compute the permeability of oxygen through a membrane and our model system contains  $N_p$

oxygen molecules, one of those will be selected and considered as our target permeant.

In some cases, it can be advantageous to select a collective RC such as the maximum value of the  $z$  coordinates of all permeants. This has the advantage that the rate increases which makes it less of a rare event and is therefore easier to compute. In addition, the collective RC facilitates the decorrelation of the sampling since the target permeant defining the RC can switch during the simulation. This strategy was for instance applied to study water dissociation where the RC was defined as the largest OH bond in the system [66]. Also in a recent paper on permeation by some of us [67] such an approach was applied to compute escape rates of permeants being trapped in a membrane.

The reason that we nevertheless choose here a RC based on a single target permeant is because the permeation problem is in some applications less of a rare event than for example water dissociation. This implies that for a system with many permeants there is almost always one of them in the membrane region. In fact, the membrane often does not correspond to a single peaked free energy barrier, but may have a well in the middle where permeants get temporary trapped. This makes a collective RC impracticable for describing stable state A. In addition, there are technical and theoretical problems associated to such a collective RC for the calculation of  $\tau_{\text{ref},[0^-]}$  that is needed for Eq. (30). The implementation of the collective RC would also be more cumbersome with periodic boundaries than when the RC just depends on a single target permeant. Instead, we can recover the advantages of the collective RC by the introduction of the new MC moves discussed in Sec. VI.

Still, also in the target permeant approach, care has to be taken with periodic boundaries when the relative position of the target permeant with respect to the membrane has to be determined. As a start, we obey the convention  $\lambda_{-1} < \lambda_0 < \lambda_n$ .

In addition, the RC should change continuously along the sequence of time slices of a complete path, i.e., it should not suddenly jump by a value equal to  $L_z$  which could lead to untrue transitions between states. Trajectories end and start with a time slice outside the  $[\lambda_{-1}, \lambda_n]$  region and the RC of these points define the states at which they start and end, so the ‘jump-free’ interval needs to be extended slightly beyond the  $[\lambda_{-1}, \lambda_n]$  interval. Based on the above, the safest option is to allow the jump in the RC to occur in the mid-point between  $\lambda_n$  and the periodic image of  $\lambda_{-1}$  (at  $\lambda_{-1} + L_z$ ). This yields for the final RC

$$\lambda = \begin{cases} z & \text{if } -\frac{L_z}{2} < z < \frac{\lambda_B + \lambda_{-1} + L_z}{2} \\ z - L_z & \text{if } \frac{\lambda_B + \lambda_{-1} + L_z}{2} \leq z \leq \frac{L_z}{2} \end{cases} \quad (34)$$

where  $z \in [-L_z/2, L_z/2]$  is the  $z$  coordinate of the target permeant following the convention of Eq. (33).

The RC as a function of the target permeant’s position within a periodic system is shown in Fig. 3. For NPT simulations with fluctuating box dimensions it might be convenient to define the RC relative to the box size along  $z$ :  $\lambda^{\text{NPT}} = \lambda/L_z$  where  $\lambda$  is still defined by Eq. (34). The only difference is that  $L_z$  is the instantaneous box length, which is a variable instead of a constant.

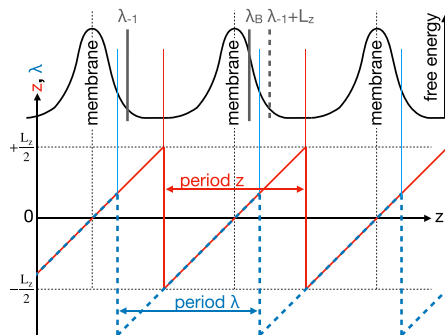


FIG. 3. Definition  $z$  and  $\lambda$ . Here  $z'$  is the unbounded coordinate of the target permeant.  $z$  is the same coordinate adjusted for the PBC such that it is 0 at the membrane center and restricted to  $[-L_z/2, L_z/2]$ . The grey dashed line at  $\lambda_{-1} + L_z$  is the mirror image of  $\lambda_{-1}$ . The mid-point between the grey dashed line and  $\lambda_B$  is located at  $(\lambda_B + \lambda_{-1} + L_z)/2$  (thin blue line) and sets the switch for the  $\lambda$  definition such that  $\lambda$  can have values  $\in [(\lambda_B + \lambda_{-1} - L_z)/2, (\lambda_B + \lambda_{-1} + L_z)/2]$ .

## VI. NEW MC MOVES IN PATH SPACE

The choice to select a single target permeant instead of a more collective RC and allowing the target permeant to cross the membrane in just one direction (left to right) can lead to a somewhat restrictive sampling speed in comparison with a more collective RC. In the next two sections, we show how to remove these restrictions by introducing two new MC moves for the  $[0^-]$  ensemble without the need to alter the definition of the RC or the setup of interfaces.

### A. Target swap move

As discussed above, the RC is determined by the  $z$  coordinate of a single target permeant. The other permeants basically contribute to the environment around the target permeant like any of the other nonpermeating particles in the system. The occurrences of these particles crossing the membrane do not have to be counted as they are part of the natural fluctuations in the environment.

Especially when the membrane is not uniform containing different channels through which the permeants could transfer, it would still be advantageous to utilize the contributions of all the permeants. Some regions in the membrane that are easier to penetrate could be blocked by a nontagged permeant. Waiting for a swap through diffusion of both permeants within the bulk might take a long time. In order to speed up this process, we design a MC move in path space that allows for a swap without diffusion, but by simply reassigning the target.

The target swap move is explained in Fig. 4 and a stepwise description of the algorithm is given below.

(1) Assume the old path (upper panel in Fig. 4) has length  $L^{(o)}$  (including start and end points) and is represented by time frames numbered from 1 to  $L^{(o)}$ .

(2) For each frame (1 to  $L^{(o)}$ ), count the number of non-target permeants inside the  $[\lambda_{-1}, \lambda_0]$  interval. The sum over



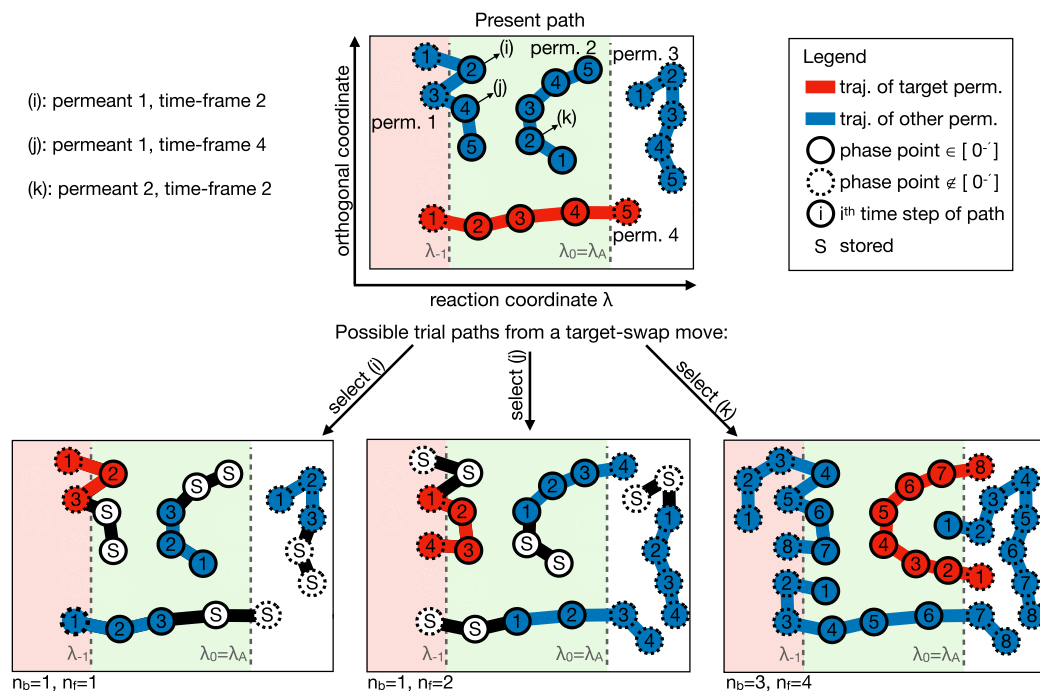


FIG. 4. Illustration of target swap move from an old path (shown in top panel) to a new path. Here, three possible new paths that could have been generated via the target swap move are shown in the lower panels. The numbers inside the circles indicate the frame numbers (time slices) of the path. Red circles represent the target permeant's positions at the different time frames, while blue circles represent the nontargeted permeants. The top panel shows a path of 5 time slices (including start and end points). The position of the target permeant at the first and last frame lies outside the  $[\lambda_{-1}, \lambda_0]$  interval while the other time frames lie inside, in agreement with the path ensemble's criteria. All permeant/time-frames outside the interval are shown by dashed contour lines and cannot be selected in the target swap moves. Possible selectable permeant/time-frames are: time frames 2, 4, 5 of permeant 1 and time frames 1-5 of permeant 2. Permeant 3 is outside the interval at all time frames. The present target, permeant 4, cannot be selected. Each of these permeant/time-frames have an equal probability to be selected, i.e., with a chance  $1/8$  in this example. Three possible selections are indicated by (i), (j), and (k) in the top figure and the resulting new trial paths are shown in the lower panels. After a permeant/time frame is selected, the corresponding permeant is the new target and the path is either lengthened or shortened by going backward and forward in time starting from the selected time frame until the new target has a frame outside the interval. Deleted time frames can be temporarily stored (indicated by "S") such that they could be reused if the next MC move requires extending the path.

all frames will be called  $Z_t^{(o \rightarrow n)}$ . For example, in Fig. 4 top, we have 5 frames to sum over, of which permeant 1 is 3 times and permeant 2 is 5 times inside the interval, leading to  $Z_t^{(o \rightarrow n)} = 8$ .

(3) Pick a random integer  $i$  from 1 to  $Z_t^{(o \rightarrow n)}$ .

(4) Select the permeant/time-frame combination corresponding to the  $i$ -th count at step 2. We will call this the new target permeant and the frame index at which this count was registered is from now on called  $j$ .

(5) Starting from time slice  $j$ , the path is followed (as detailed below) backward until we detect a frame in which the new target has a position outside the  $[\lambda_{-1}, \lambda_0]$  interval. We call the number of backward steps  $n_b$ .

(6) Starting from time slice  $j$ , the path is followed (as detailed below) forward until we detect a frame in which the new target has a position outside the  $[\lambda_{-1}, \lambda_0]$  interval. We call the number of forward steps  $n_f$ .

(7) The new path length is  $n_b + n_f + 1$  which starts with frame index  $j - n_b$  (which can be negative) and ends with frame index  $j + n_f$ . Renumber all frame indices by adding  $-j + n_b + 1$  to each frame index. The new indices now run from 1 to  $n_b + n_f + 1$ .

(8) Compute  $Z_t^{(n \rightarrow o)}$  just as in step 2, but using the new time region and new target permeant. For example, for the three bottom panels in Fig. 4, it is 5, 5, and 8 (from left to right).

(9) Compute the number of selectable permeant/time-frames  $n_s^{(n)}$  [see Eq. (35) below] from which the same new trajectory can be obtained from the old path, and the number of selectable permeant/time frames  $n_s^{(o)}$  at the new path from which the old path could be reobtained [see Eq. (36) below].

(10) Accept the move with a probability [see Eq. (37)

$$\text{below]: } P_{\text{acc}} = \min\left(1, \frac{n_s^{(o)} Z_t^{(o \rightarrow n)}}{n_s^{(n)} Z_t^{(n \rightarrow o)}}\right).$$

At steps 4 and 5, the path is followed backward and forward in time starting from the selected time frame. In order to minimize the number of expensive force evaluations, this is done via one of the three possibilities that are listed here in order of preference. (1) Take the previous/next time slice from the old path, (2) if all time slices of the old path along a time direction are used, check for possible stored time slices in that time direction, and (3) create a new time slice by an actual MD step if no reusable time slice is available at (1) or (2).

At step 9, the calculation of  $n_s^{(n)}$  and  $n_s^{(o)}$  proceeds as follows.  $n_s^{(n)}$  equals the number of selectable permeant/time frames combinations by which the same new path can be obtained. The way how it was actually generated, by selecting the new target permeant and time frame  $j$ , is one of the possible realisations. However, the exact same path might be generated by selecting time frames earlier and/or later. To clarify this, we discuss  $n_s^{(n)}$  for the three cases shown in the bottom panels of Fig. 4. In the bottom-left panel, the new path can only be obtained by selecting time slice number  $j = 2$  and consequently  $n_s^{(n)} = 1$ . The path in the bottom-middle panel can be obtained in two ways: selecting permeant 1 and either time slice  $j = 4$  or  $j = 5$  (renumbered as 2 and 3). Hence  $n_s^{(n)} = 2$  in this case. Finally, the path in the bottom-right panel could be obtained by any of the 5 time slices when permeant 2 is selected ( $n_s^{(n)} = 5$ ).

An expression for  $n_s^{(n)}$  and  $n_s^{(o)}$  is now derived. The number of time slices earlier than  $j$  that, if selected, would result into the same path, is restricted by either the old path,  $j - 1$  time slices, or the new path,  $n_b - 1$  backward steps (the last backward step is outside the  $[\lambda_{-1}, \lambda_n]$  interval and cannot be selected). So this gives a contribution  $\min(j - 1, n_b - 1)$  to  $n_s^{(n)}$ . Forward in time these numbers are  $L^{(o)} - j$  and  $n_f - 1$  for restriction by either the old path or the new path, respectively. This yields a contribution of  $\min(L^{(o)} - j, n_f - 1)$ . Including the time slice  $j$  itself, this gives

$$\begin{aligned} n_s^{(n)} &= \min(j - 1, n_b - 1) + \min(L^{(o)} - j, n_f - 1) + 1 \\ &= \min(j, n_b) + \min(L^{(o)} - j, n_f - 1). \end{aligned} \quad (35)$$

For the reverse move, i.e., selecting the old path from the new one, the same reasoning applies with the roles of the new and old paths switched. Hence,  $j - 1$  and  $n_b - 1$  are replaced by  $n_b$  and  $j - 2$ , respectively, when computing the selectable time slices before  $j$ . Further,  $L^{(o)} - j$  and  $n_f - 1$  are replaced by, respectively,  $n_f$  and  $L^{(o)} - j - 1$ . This gives for  $n_s^{(o)}$

$$\begin{aligned} n_s^{(o)} &= \min(n_b, j - 2) + \min(n_f, L^{(o)} - j - 1) + 1 \\ &= \min(n_b + 1, j - 1) + \min(n_f, L^{(o)} - j - 1). \end{aligned} \quad (36)$$

Then, using Metropolis-Hastings rule [68,69], the acceptance probability can be written as

$$\begin{aligned} P_{acc}(o \rightarrow n) &= \min\left(1, \frac{p(n)P_{gen}(n \rightarrow o)}{p(o)P_{gen}(o \rightarrow n)}\right) \\ &= \min\left(1, \frac{n_s^{(o)}/Z_t^{(n \rightarrow o)}}{n_s^{(n)}/Z_t^{(o \rightarrow n)}}\right), \end{aligned} \quad (37)$$

where  $p(o)$  and  $p(n)$  are the probabilities of the old and new path, respectively, and  $P_{gen}(X \rightarrow X')$  is the generation

probability to generate path  $X'$  from  $X$ . As the identity of the target has no effect on the path probabilities, the probabilities  $p(o)$  and  $p(n)$  are essentially the same except for stochastic force terms related to extending or shortening of the path. However, these terms cancel in Eq. (37) as they are also part of the generation probabilities [54]. The only remaining terms that do not cancel are, hence, the selection probabilities for selecting the permeant/time slice.

## B. Mirror move

In the case that the membrane is symmetric, transitions through the membrane from left to right and from right to left are statistically indistinguishable within an equilibrium sampling. It is then favorable to count transitions in both directions [10,12]. For instance, the direct counting method described in Sec. II uses this strategy to improve statistics.

One obvious way to include two-directional transitions could be achieved by defining the RC as the absolute distance  $|z|$  between the target permeant and the center of the solvent slab (see also Ref. [67]). However, as explained in Sec. V, this can lead to an overlap in the  $z$ -coordinate space. While this could still be solved by letting the RC value depend on the history of the path, i.e., the solvent slab's periodic image to be considered is determined by the minimum distance image at the start of the path, this becomes problematic when the path's history is not yet fully determined. For instance, this is the case when a shooting move is carried out or when the target swap move implies that some backward in time integration is required.

The mirror move in path space (see Fig. 5) achieves the same versatility of the two-directional approach in the counting method without having the problems discussed above. The mirror move in  $[0^-]$ , which is always accepted, mirrors the whole system with respect to the membrane center. The  $z$  coordinates of every particle are mirrored and the  $z$ -component of the velocities are multiplied with  $-1$ . Because of the periodicity, this mirror move is equivalent to mirroring the whole system with respect to the midpoint between the membrane and its periodic image, which is, loosely speaking, the midpoint of the solvent slab. By construction, this midpoint of the solvent slab lies in the middle of the  $[\lambda_{-1}, \lambda_0]$  interval related to  $[0^-]$ .

The mirror move swaps the roles of the  $\lambda_0$  and  $\lambda_{-1}$  interfaces. This requires that  $\lambda_0$  and  $\lambda_{-1}$  are placed at the same distance from the mirror plane. To achieve this, consider the membrane's center-of-mass position at  $z = z_{mem}$  and its left periodic image at  $z = z_{mem} - L_z$ . The distance between  $\lambda_0$  and membrane should equal the distance between  $\lambda_{-1}$  and this left periodic image membrane. In other words,  $\lambda_{-1}$  should be placed such that

$$z_{mem} - \lambda_0 = \lambda_{-1} - (z_{mem} - L_z). \quad (38)$$

Equivalently, the interfaces  $\lambda_{-1}$  and  $\lambda_0$  should have the same distance to the midpoint of the solvent slab at  $z = z_{mem} - L_z/2$ . Since we applied the periodic coordinates  $z$  as defined in Eq. (33), we have  $z_{mem} = 0$  and therefore  $\lambda_{-1} = -(\lambda_0 + L_z)$ .

Given the positioning of  $\lambda_{-1}$  and  $\lambda_0$  according to Eq. (38), the mirror move simply mirrors the  $z$  coordinates of every particle in the system at every time slice with respect to the

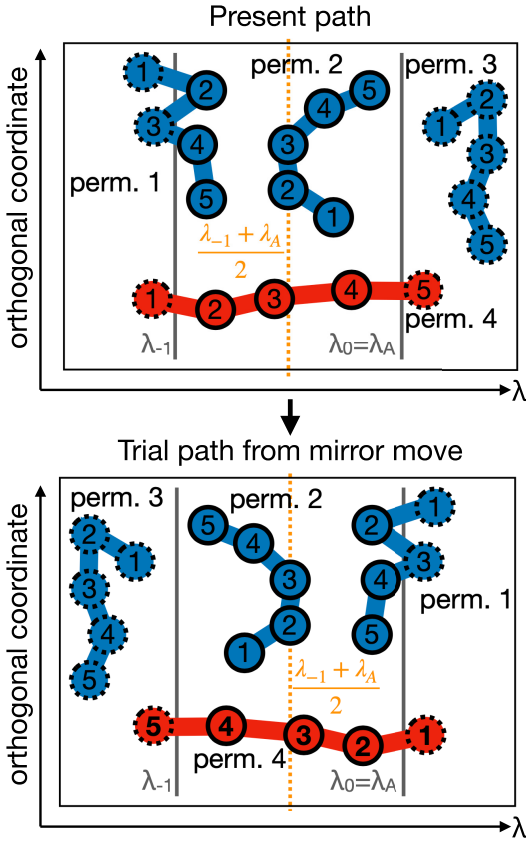


FIG. 5. Illustration of the mirror move. Numbers, line style and color scheme are the same as in Fig. 4. The move requires a specific definition of the  $\lambda_{-1}$  interface that must be symmetrically positioned with respect to  $\lambda_A$ , implying that these are equidistant to the center of the solvent slab. The top panel shows the old path. The bottom panel shows the new path in which the positions for each time slice are inverted in the mirror plane centered at the solvent slab (orange dashed line at  $(\lambda_{-1} + \lambda_A)/2$ ). The mirroring is applied to all particles in the system, not only the permeants.

plane  $z = -L_z/2$ . Hence, it proceeds according to the following steps.

(1) Let  $k$  run over all time slices of the path. This includes the start and end point of the path and might include stored time slices (see Sec. VI A).

(2) For the given time slice, let  $j$  run over all point-particles in the system (atoms/coarse-grained particles describing the permeants, nonpermeating solvent particles, membrane particles, ...).

(3) Consider the  $z'$  coordinate of permeant  $j$  and its velocity component at discrete time step  $k$ :  $z'_j(k)$  and  $v_{z,j}(k)$ .

(4) Mirror operation: replace  $z'_j(k)$  with  $2z'_{\text{mem.}} - z'_j(k) - L_z$  and replace  $v_{z,j}(k)$  with  $-v_{z,j}(k)$ .

(5) Accept the new path. If the old path started/ended at  $\lambda_{-1}$  then the new path will start/end at  $\lambda_0$  and vice versa.

For code-technical reasons we implemented a slightly different approach where we did not alter the coordinates or velocities, but instead the definition of the RC function [Eq. (34) with  $z$  replaced by  $-z$ ]. The system was hence assigned an additional flag which indicates whether Eq. (34) has to be used with the plain  $z$  coordinate of the target permeant or with  $-z$ . This pragmatic choice made it easier to use PYRETIS [57,58] with external MD engines, as these might have very different ways of altering the coordinates and velocities. The flag is also exchanged in the replica exchange moves of the RETIS algorithm.

The new moves are only implemented for the  $[0^-]$  ensemble. For the mirror move, this is because this is the only ensemble where paths can start at both the left or the right hand side. In addition, the target swap move is not expected to give a high acceptance for the  $[i^+]$  ensemble when crossing  $\lambda_i$  is a rare event.

## VII. NUMERICAL RESULTS

The theoretical derivation for the permeability calculation from RETIS has been implemented in the python based open-source code PYRETIS [57,58]. First, a one-dimensional toy system was constructed where a Langevin, Brownian, or deterministic Newtonian particle permeates through a medium with or without barrier. For some limiting cases, an analytical expression for the permeability is available, which can serve as a validation of the new RETIS permeability formula. Second, a two-dimensional membrane was simulated with periodic boundary conditions, where permeants can pass the membrane through two different permeation channels. This last system is used to illustrate efficiency of the new Monte Carlo moves.

### A. One-dimensional system setup

For simplicity, the membrane is located symmetrically around  $z = 0$ , in the region  $|z| < a$ , with  $h = 2a$  as the membrane height. The effect of the membrane is modeled by an external cosine-shaped potential that acts on a single permeant particle,

$$V(z) = \begin{cases} \frac{1}{2}V_0 \left( \cos \frac{\pi z}{a} + 1 \right), & |z| \leq a \\ 0, & a < |z| \leq b \\ \frac{1}{2}k_{\text{harm}}(|z| - b)^2, & |z| > b \end{cases} \quad (39)$$

Here,  $V_0$  is barrier height. This membrane model ensures that the force on the particle is continuous at the membrane borders  $z = \pm a$ . The harmonic potential  $\frac{1}{2}k_{\text{harm}}(|z| - b)^2$  is added only to allow the system also to be studied by a reference simulation without  $\lambda_{-1}$  interface. In most simulation setups, the  $k_{\text{harm}}$  parameter is set to zero, which reflects the real physical situation for a permeation system that is unbounded at either side of the membrane. The dynamics of the permeant is governed by either Langevin dynamics, Brownian motion, or deterministic dynamics.

In the Langevin dynamics, the permeant experiences both friction, inertia effects, and random collisions with a degree that is controlled by the friction parameter  $\gamma$ . The friction constant  $\gamma$  (unit 1/time) of the particle relates to the particle's diffusion constant as  $D = k_B T / (m\gamma)$ . Note that some other



textbooks use a friction coefficient with unit mass/time. This refers to an alternative definition of the friction coefficient  $\tilde{\gamma} = m\gamma$ . The two other types of dynamics can be viewed as limiting cases of the Langevin dynamics.

A Brownian particle propagates in discrete steps without memory under influence of a random Gaussian force and the force  $-dV/dz$  exerted by the potential. It can be considered as the overdamped limit of the Langevin particle, when  $\gamma \rightarrow \infty$ .

A Newtonian particle on the other hand propagates deterministically in time according to the equations of Newton. It has inertia but undergoes no friction nor random collisions, and energy is conserved. In the RETIS algorithm, the effect of temperature is then only present in the MC moves when a new constant-energy path is created from an old constant-energy path. The detailed balance MC procedure allows the energy to change between paths such that the overall path ensembles are canonically distributed. This simulation set up reflects a system that is so weakly coupled to a thermostat that the dynamics of a single crossing basically occurs at a constant energy (NVE, microcanonical). However, at the much longer timescale between crossing events, the energy could change. The Newtonian particle can be seen as the low friction limit of the Langevin particle, when  $\gamma \rightarrow 0$ .

The driver for solving the equations of motion was the internal MD-engine of PYRETIS in our toy system [57]. It is also possible to let PYRETIS manage the path ensembles book keeping, while it calls simulation programs, such as GROMACS, OPENMM, or LAMMPS, to drive the molecular dynamics [58]. Throughout Sec. VII, reduced units are used in which mass  $m$ , the Boltzmann constant  $k_B$ , and temperature  $T$  are equal to unity ( $m = k_B = T = 1$  in reduced dimensionless units). Potentially, several physically realistic systems could be mapped on the model presented in Eq. (39) by tuning appropriate units of energy, length and mass.

We examined the model system Eq. (39) using the following parameters:  $a = 0.1$  ( $h = 0.2$ ),  $k_{\text{harm}} = 100$  or  $0$ ,  $m = 1$ ,  $T = 1$ , and integration time step  $\Delta t = 0.002$ . The friction coefficient  $\gamma$  had values  $0.1, 5, 10, 20, 40, 60, 80$ , or  $100$ . The Brownian dynamics propagates through configuration space at discrete steps with a displacement that is governed by a step-size parameter. This step-size parameter can be associated to a  $\Delta t$  in an equivalent Langevin simulation for a given mass and friction. In our case, the step-size parameter was set such that the time between steps was the same as  $\Delta t$  of the corresponding Langevin simulation with  $\gamma = 100$ .

In the RETIS simulations, the number of cycles was set to 20 000 (this is the number of MC moves in the path ensembles), the RETIS swapping move frequency was  $0.1$ , the shooting frequency  $0.45$ , and the time reversal move frequency  $0.45$ . The position  $z$  was used as the order parameter  $\lambda$ . Three interfaces were used,  $\lambda_0, \lambda_1, \lambda_2$ , which were chosen at  $\lambda = -0.1, 0$ , and  $0.1$ . The additional interface  $\lambda_{-1}$  was chosen at  $\lambda = -0.2$  and varied to a few other locations in the calculations of Table I. The chosen reference region is  $[-0.12, -0.1]$ .

## B. Analysis of permeability

For the flat potential membrane [ $V_0 = 0$  in Eq. (39)], we examined the effect of the  $\lambda_{-1}$  interface versus a system that

TABLE I. Numerical results (reduced units) for permeability  $P$  and the three contributing factors  $\xi$ ,  $\tau_{\text{ref}}/\Delta z$ , and  $P_A(\lambda_B|\lambda_A)$ . RETIS simulation of Langevin particle with  $\gamma = 5$  and flat potential membrane ( $V_0 = 0$ ). On first line,  $\tau_{\text{ref}}$  refers to  $[0^-]$ , on other lines to  $[0^-]$ . The reported error is based on block averaging and error propagation rules assuming independence of the different path ensemble simulations. As the latter assumption is not fully valid due to the replica exchange moves (Ref. [22]), we also estimated the error on  $P$  via 10 independent realizations, shown in the last column.

$\lambda_{-1}$	$\xi$	$\tau_{\text{ref}}/\Delta z$	$P_A(\lambda_B \lambda_A)$	$P$
-	1.000 (0%)	2.49 (1%)	0.662 (2%)	0.266 (2.2,2.0%)
-0.2	0.493 (0.3%)	1.22 (1%)	0.674 (2%)	0.274 (2.4,2.9%)
-0.15	0.504 (0.2%)	1.26 (1%)	0.641 (2%)	0.256 (2.1,2.4%)
-0.3	0.507 (1%)	1.28 (1%)	0.661 (2%)	0.261 (2.3,2.9%)

is bounded by a harmonic potential [ $k_{\text{harm}} = 100$  in Eq. (39)] for Langevin dynamics with  $\gamma = 5$ . Table I shows that the change in  $\tau_{\text{ref}}/\Delta z$  is compensated by the  $\xi$  factor. The permeability further remains fairly unaffected when changing the  $\lambda_{-1}$  position. The time spent per path per length is about  $1.22$  to  $1.28$  in  $[0^-]$ , which lies close to the analytical value  $1.253$  of the deterministic particle (see Appendix B). The time spent per path per length would become larger when the friction increases. The crossing probability is lower than for the deterministic particle, which makes the Langevin permeability lower than the deterministic value  $0.399$  (see Appendix B).

Figure 6 compares the computed permeability for the potential Eq. (39) with  $V_0 = 0$  (flat),  $0.5$ , and  $1$  (cosine barrier membrane) for the three types of particle dynamics. In the Appendix we derived analytical expressions for  $P$  based on the Smoluchowski equation and Kramers' expression for Brownian dynamics and Langevin dynamics, respectively. These theoretical curves are shown in the same graphs. The analytical result for deterministic dynamics can be obtained by taking the limit of Kramers' expression for  $\gamma \rightarrow 0$ . The validity of these theoretical results relies on different kind of approximations. The Smoluchowski expression [Eq. (B1)] is reliable for high friction and low to high barriers, while Kramers' theoretical result [Eq. (B6)] is the reliable reference for high barriers and low to high friction. There is henceforth a blind spot in system parameter space: dynamics with low friction and low barriers is poorly described by both theories.

Indeed, consider the flat potential membrane ( $V_0 = 0$ ) in Fig. 6(a). It shows good agreement between the theoretical Smoluchowski curve,  $P = D/h$ , and the simulated Langevin results for the large  $\gamma$  values. Also the computed values for Brownian and Langevin at  $\gamma = 100$  agree, as Brownian dynamics can be seen as the high friction limit of Langevin dynamics. However, for low friction, the Smoluchowski curve and the numerical Langevin results deviate. The Kramers prediction of the permeability is zero for any  $\gamma > 0$  and thus also fails to approximate the Langevin numerical results. Only the limiting case  $\gamma \rightarrow 0$  with  $\gamma^2/V_0 \rightarrow 0$  has a nonzero solution equal to  $\sqrt{k_B T}/(2\pi m)$ . The Kramers curve in Fig. 6(a) actually shows the theoretical results for  $V_0 = 10^{-4}$  to visualize this limiting case. The deterministic  $\sqrt{k_B T}/(2\pi m)$  limit to the permeability is indicated by the dashed horizontal line. Our numerical data on deterministic dynamics and the Langevin result with  $\gamma = 0.1$  agree with this limit.

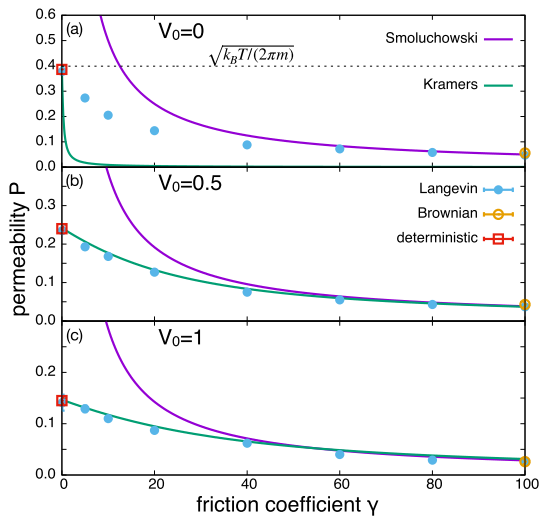


FIG. 6. Permeability  $P$  versus friction coefficient  $\gamma$  for  $V_0 = 0$  (a), 0.5 (b), and 1 (c). The purple and green solid lines refer to the theoretical Smoluchowski expression Eq. (B1) and Kramers expression Eq. (B6), respectively. The Kramers curve in (a) actually shows the function for  $V_0 = 10^{-4}$  instead of  $V_0 = 0$  to visualize a limiting case of this expression in which  $V_0 \rightarrow 0$ ,  $\gamma \rightarrow 0$  and with  $\gamma^2/V_0 \rightarrow 0$ . The dashed horizontal line in the top panel shows the deterministic limit for the permeability in the flat potential. The filled blue circles refer to the numerical Langevin results. The open red square and open gold circle at  $\gamma = 0$  and 100 show the results for deterministic dynamics and Brownian dynamics, respectively. Error bars based on a single standard deviation are mostly within symbol size.

Figure 6(b) and 6(c) show the numerical and theoretical curves for systems with a membrane barrier,  $V_0 = 0.5$  and 1.0, respectively. For these membranes, the two theories agree in the large friction regime. In the low friction regime, the Smoluchowski expression is not a good approximation of the Langevin dynamics. The numerical Langevin simulations agree with the Kramers' curve for all values of  $\gamma$ . The deterministic and Brownian dynamics simulations also agree with Kramers' expression in the limiting  $\gamma = 0$  and  $\gamma = 100$ , respectively.

### C. Two-channel membrane system setup

A two-dimensional system is constructed that mimics a membrane barrier through which particles can permeate through two competing pathways. It could for instance represent a membrane with two transmembrane protein channels. Three noninteracting Langevin particles are subjected to the potential

$$V(y, z) = e^{-cz^2} \left( V_1 + A + A \sin \frac{2\pi y}{L_y} + B + B \cos \frac{4\pi y}{L_y} \right), \quad (40)$$

$$A = (V_2 - V_1)/2,$$

$$B = V_{\max}/2 - V_1/4 - V_2/4.$$

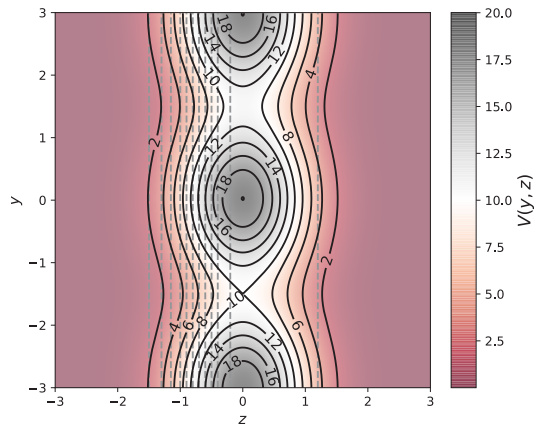


FIG. 7. The potential energy  $V(y, z)$  represents a two-channel membrane. Interfaces  $\lambda_0, \dots, \lambda_{11}$  indicated with vertical dashed lines. Reduced units.

The membrane is located in the center of the unit cell around  $z = 0$ , while  $V(y, z)$  is approximately zero far away from the center due to the factor  $e^{-cz^2}$  (see Fig. 7). Periodic boundary conditions are applied, where the system is made periodic in the  $z$  direction with a period  $[-L_z/2, L_z/2]$  and in the  $y$  direction the period is  $L_y$ . Particles can permeate the membrane through two channels: one channel at about  $y = -0.25L_y$  with barrier height  $V_1$  and another channel at about  $y = 0.25L_y$  with barrier height  $V_2$ . The maximum barrier height is  $V_{\max}$ .

Reduced units are used as in the one-dimensional case. The parameters in our simulations were  $V_1 = 10$ ,  $V_2 = 11$ ,  $V_{\max} = 20$ ,  $c = 1$ , and  $L_z = L_y = 6$ . The PYRETIS simulations were run with three Langevin particles with settings  $\Delta t = 0.02$ ,  $\gamma = 5$ ,  $T = 1$ , and  $m = 1$ .

The order parameter is the reduced  $z$  coordinate of the target permeant:  $\lambda = z_j$  if permeant  $j$  is tagged ( $j = 1, 2, 3$ ). While the  $z$  coordinates lie in the interval  $[-3, 3]$ , the periodicity of  $\lambda$  is shifted to the interval  $[-4.65, 1.35]$  (see Fig. 3). Twelve interfaces are located at  $\lambda = -1.5, -1.3, -1.15, -1, -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.2$ , and 1.2. The  $\lambda_{-1}$  interface is located at  $\lambda = -4.5$ . The reference region for  $\tau_{\text{ref}, [0^-]}$  is chosen as  $[-3.2, -2.8]$ . Three simulations are performed.

(1) TIS: In TIS, there are no swapping moves between the path ensembles. The MC moves are the shooting move and time reversal move, with equal frequency 0.5. Differently to standard TIS, the sampling of state A was done using the  $[0^-]$  path ensemble simulation and not via MD.

(2) RETIS: In standard RETIS, the swapping move of paths between the path ensembles is also allowed as an MC move. The swapping move frequency was 0.5, shooting move frequency 0.25, and time reversal move frequency 0.25.

(3) RETIS\*: In the last simulation, RETIS is performed with swapping moves and the newly implemented MC moves in the  $[0^-]$  ensemble. The mirror plane was located at  $\lambda = -3$ , which is indeed midway between  $\lambda_{-1} = -4.5$  and  $\lambda_0 = -1.5$ .

TABLE II. Numerical results (reduced units) for permeability  $P$  of 3 Langevin particles permeating through a two-channel membrane. Standard error from block averaging and error propagation between brackets.

	$\xi$	$\frac{\tau_{\text{ref},[0^-]}}{\Delta z}$	$P_A(\lambda_B \lambda_A)$ $\times 10^{-5}$	$P$ $\times 10^{-6}$
TIS	0.498 (1%)	5.66 (1%)	1.10 (12%)	0.97 (12%)
RETIS	0.540 (1%)	6.19 (1%)	1.20 (14%)	1.05 (14%)
RETIS*	0.507 (1%)	5.93 (1%)	1.23 (13%)	1.06 (13%)

Each of the three particles could be selected as the target permeant when performing a target swap move. The swapping move frequency was 0.5, time reversal move frequency 0.25, mirror move frequency 0.05, target swap move frequency 0.05, and shooting move frequency 0.15.

After an equilibration run of about 1600 MC moves, the analysis was performed based on a production run of 35 000 MC moves.

### D. Two-channel membrane: analysis

Table II shows the permeability together with the calculated variables that enter in Eq. (30),  $\xi$ ,  $\tau_{\text{ref},[0^-]}/\Delta z$ , and  $P_A(\lambda_B|\lambda_A)$ . Note that the RETIS results on  $\xi$  and  $\tau_{\text{ref},[0^-]}/\Delta z$  are somewhat off compared to TIS and RETIS\*. Naturally, for this symmetric system  $\xi = 0.5$  is the exact result which agrees with TIS and RETIS\* while RETIS is 8% too high. This is due to the relatively wide region between  $\lambda_0$  and  $\lambda_{-1}$  and the lower frequency of shooting moves in the RETIS simulation compared to TIS. In RETIS, 50% of the moves are swapping moves (replica exchange moves between path ensembles), which are very useful to improve the sampling of the barrier region, but not necessarily help the exploration of the water phase. Both the swapping and time-reversal moves are unable to generate a  $\lambda_{-1} \rightarrow \lambda_{-1}$  path from a  $\lambda_0 \rightarrow \lambda_0$  path and vice versa. In RETIS\*, the target swap move and the mirror move repair this weakness even if these moves only represent 10% of the executed MC cycles.

The crossing probabilities  $P_A(\lambda_B|\lambda_A)$  and the permeabilities in Table II give quantitative good agreement within about 10%. Given a barrier of at least 10  $k_B T$  this is a notable result. However, even more difficult challenges for a simulation method in this system, are (i) the ability to sample transitions through both channels, and (ii) to achieve this with the correct ratio. Since the channels' barriers only differ by 1  $k_B T$ , both permeation routes are competing, but successful permeation transitions are expected to proceed via the lowest barrier channel in about 73% of the cases. Getting this ratio right is extremely challenging for any rare event method.

Figure 8 shows the distribution  $P(y^*)$  of the orthogonal coordinate at the first crossing with  $\lambda_i$ ,  $y^*$ , for different path ensembles,  $[i^+]$ ,  $i = 0, 1, \dots, 10$ . The crossing point  $y^*$  of a path is indicative of the channel visited by that path. The distributions show that for TIS, all  $y^*$  crossing points in the ensembles  $[6^+]$  and higher are in the  $V_2$  channel, while for RETIS and RETIS\* the other channel is visited as well in all ensembles. This clearly demonstrates the deficit of the shooting move. The chance for this move to generate an acceptable

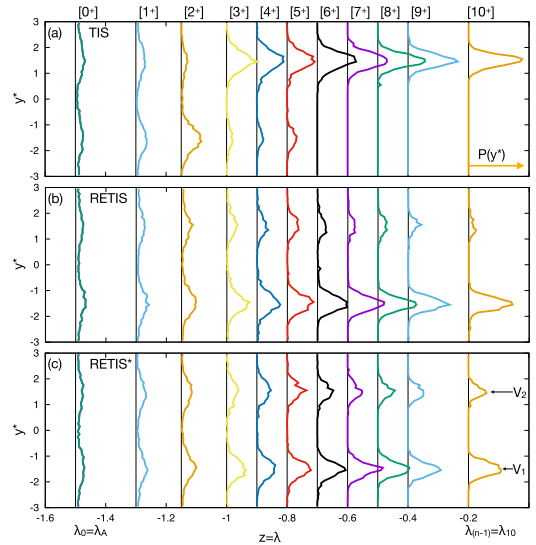


FIG. 8. Distributions  $P(y^*)$  of first crossing point with  $\lambda_i$  along the  $y$ -direction for the different  $[i^+]$  ensembles. Results are shown for (a) TIS, (b) RETIS, and (c) RETIS\*. Note that the TIS simulations only sample the high-energy barrier  $V_2$  for ensembles  $[6^+]$  and higher.

path is highest when the shooting point is chosen on the barrier region, close to  $\lambda_i$  for ensemble  $[i^+]$ . However, switching between channels can practically only occur if the shooting is initiated from the well region. The fact that TIS got stuck in the high-energy channel, rather than the low-energy channel is purely accidental reflecting the memory of the initial path that was used to bootstrap the simulation. The TIS result (Table II) is lower than the other values, as one might expect based on its bias towards the high-barrier channel. Yet, since the crossing probability up to  $\lambda_6$  is based on the progress through both channels, the TIS permeability is still rather close to the RETIS and RETIS\* results.

Provided ergodic sampling, TIS and RETIS should be capable to sample nontrivial multiple-channel systems where splitting based methods, like FFS and AMS, would fail. An example of such a case is a system with two channels in which the lowest-barrier channel goes initially much steeper uphill than the channel with the higher barrier [62]. However, as is clear from Fig. 8, the TIS simulation is not ergodic since it is not able to switch between channels for ensembles  $[6^+]$  and higher with just the shooting move.

For this academic model, this aspect could be repaired using nonlocal shooting moves in which not only the velocities, but also the configuration point is changed by a nonlocal displacement. Such a move, however, would have vanishingly low acceptance in a realistic condensed matter system as nearly every attempt will lead to a molecular overlap.

The RETIS and RETIS\* simulations, however, are able to sample both channels and get the ratio between low- and high-barrier pathways at least qualitatively correct. The replica exchange swapping moves allow the exchange of paths be-

tween the different path ensembles. Consequently, sampling in  $[0^-]$  and  $[i^+]$  with low  $i$  can facilitate the sampling in  $[i^+]$  with high  $i$ . Especially the  $[0^-]$  and  $[0^+]$  path ensembles are very effective to sample the direction that is orthogonal to the RC enabling the entrance of both channels. Figuratively speaking, this improved sampling of the orthogonal coordinate can then trickle down into the other ensembles by means of the swapping moves.

If we examine the height of the distributions in Fig. 8, we see that both RETIS and RETIS\* predict that the majority of transitions will pass via the low-barrier channel. The RETIS simulation, however, seems to overestimate the preference of the  $V_1$  channel, especially when the  $[10^+]$  ensemble is considered. Integration of  $\exp(-\beta V(y, \lambda_{10}))$  over  $y$  along positive and negative values indicates a 2.54 higher probability to be in the negative  $y$ -range. Even if these relative probabilities deviate a bit from the distribution of first crossing points  $y^*$ , this deviation is expected to be marginal for this model system.

To further analyze the effectiveness of the MC schemes we analyzed the number of channels switches observed in the path ensemble simulations. For path ensemble  $[i^+]$  each path is assigned to channel  $V_1$ , channel  $V_2$ , or neither of the two, based on the first crossing point  $y^*$  with  $\lambda_i$ . It is assigned to belong to the  $V_1$ -channel if  $-2.5 < y^* < -0.5$  and to the  $V_2$  channel if  $0.5 < y^* < 2.5$ . A channel switch is counted when the MC move produces a  $V_2$  channel path while the  $V_1$  channel was most recently visited, and vice versa.

Figure 9(a) shows the number of channel switches that are calculated via this approach for different path ensembles. The TIS results are magnified by a factor 20 for visualization as this approach shows dramatically less channel switches than RETIS and RETIS\*. This shows that replica exchange (swaps between path ensembles) is absolutely necessary for efficient sampling. From  $[0^+]$  to  $[5^+]$  ( $\lambda_0 = -1.5$  to  $\lambda_5 = -0.8$ ), the number of channel switches drops from 144 to 1. From  $[6^+]$  to  $[10^+]$  ( $\lambda_6 = -0.7$ ,  $\lambda_{10} = -0.2$ ), there is not a single channel switch observed.

The difference in channel switches seems negligible between RETIS and RETIS\* up to  $\lambda_i = -1$ . After that point, RETIS\* seems to produce significantly more switches. This is remarkable since the extra moves, the mirror and target swap move, are only executed in the  $[0^-]$  ensemble and its effect on the  $[10^+]$  path ensemble is only indirect via the replica exchange moves. It requires at least 10 path ensemble swaps to process any information from  $[0^-]$  up to  $[10^+]$ . Still, the effect is most noticeable for the last seven path ensembles, but hardly before. Our conjecture is that the channel switches due to the mirror move and certainly due to the target swap move are more effective in decorrelating the ensemble. A channel switch is likely more effective if it enters the new channel along its central line and indeed this happens more often with the target swap move than with the shooting move. Also, the number of channel switches does not tell the full story. If two path ensemble  $[i^+]$  and  $[(i+1)^+]$  are at different channels and then make a lot of successful replica exchange moves solely between each other, this will yield a lot of channel switches. However, the effectiveness in decorrelating the sampling will be modest.

To examine this decorrelation, Fig. 9(b) plots the ratio of the number of paths in the  $V_1$  and  $V_2$  channels as a function

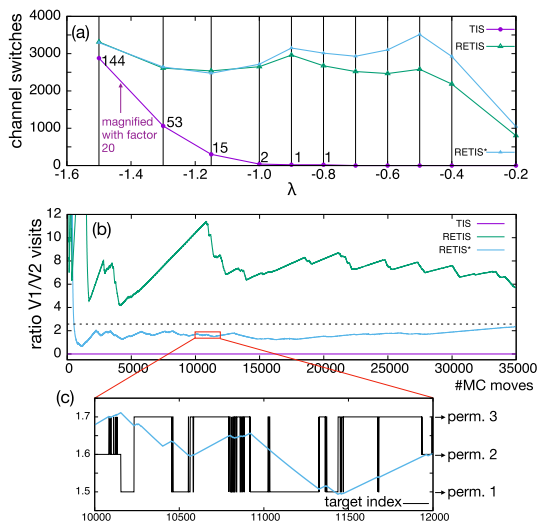


FIG. 9. (a) Number of channel switches observed in  $[i^+]$  as a function of  $\lambda_i$  for TIS, RETIS, and RETIS\*. The TIS results are magnified for visibility and shown with explicit numbers for the nonzero values. (b) Running average of ratio between generated  $V_1$  and  $V_2$  trajectories for the  $[10^+]$  ensemble. Dashed line is the predicted value based on the numerical integration of  $\exp(-\beta V(y, \lambda_{10}))$ . (c) A zoom of the upper curve together with target index on the right vertical axis. It shows that channel switches (indicated by a change in direction of the blue line) occur often when the target permeant is reassigned.

of the number of MC moves in the  $[10^+]$  ensemble. The TIS curve is a flat zero line as it is stuck in the  $V_2$  channel for the full simulation. Comparing RETIS and RETIS\*, we see that RETIS\* converges much faster and approaches the predicted value of 2.54 that was obtained from the numerical integration. Figure 9(c) shows a zoom of the curve together with the index for the target permeant. These running averages for RETIS and RETIS\* have a typical sawtooth shape though the latter is able to flip much more frequently the slope of the curve. The zoom in Fig. 9(c) shows that such a flip often coincides with a change of the index for the target permeant. This indicates that the target swap move has a strong influence in the overall sampling even if it is only applied in the  $[0^-]$  ensemble.

## VIII. CONCLUSION

In this work, we derived a formula for the permeability based on path sampling quantities that can be determined in a RETIS simulation. As the idealized permeation model represents a membrane inside an infinite solution, the RETIS path ensembles require an adaptation via the introduction of an additional interface prior to  $\lambda_0$ , called  $\lambda_{-1}$ , and a newly defined path ensemble  $[0^-]$  that replaces the  $[0^-]$  path ensemble. The resulting approach is exact and does not depend on the positions of the interfaces including  $\lambda_{-1}$ . Their positions are therefore set to optimize efficiency.

In addition to this theoretical derivation, we introduce a few algorithmic developments such as a consistent way to define the reaction coordinate for permeability whenever periodic boundary conditions are applied and two additional MC moves that mainly operate in the new  $[0^-]$  path ensemble. One of these new MC moves is the mirror move which can be applied whenever the membrane is symmetric. The other MC move is the target swap move and can be used when more than one permeant is present in the simulation model system.

Our new theoretical formulation and algorithmic developments have been implemented in the open-source PYRETIS code [57,58], and it was successfully tested on a one-dimensional Langevin system for which analytical results exist. After this, a challenging two-dimensional model membrane with two competing permeation channels was simulated to test the effectiveness of the new MC moves. These simulations show that the replica exchange moves are essential to simulate this system as the plain TIS method gets trapped inside a single channel. The inclusion of the two new MC moves considerably improves the sampling efficiency even further, as is clear when inspecting the relative transmission through the two channels. This noticeable difference is surprising given the fact that the new moves only operate in the  $[0^-]$  ensemble and it takes at least 10 replica exchange moves to transfer the effect of these moves up to the last path ensemble  $[10^+]$ . Still, the direct relation between the improved efficiency and the new MC moves was demonstrated by a correlation between the channel-switches and changes of the target permeant's identity.

The theoretical derivation in this paper is valid for all kinds of microscopically reversible dynamics (e.g., deterministic Newtonian dynamics, Langevin, Brownian, Nosé-Hoover, etc.). Besides the standard ergodicity hypothesis, it does not rely on any further assumption nor approximations. This implies that our approach will in principle give the same value for the permeability as the direct counting method based on brute force simulations, but orders of magnitude faster.

Our approach has the great advantage that the Markovian assumption of memoryless hopping between interfaces (see Sec. I) is not needed like the approaches based on milestone [42–45]. Especially for large molecules the permeation process is often driven by nontrivial membrane fluctuations such that the projected dynamics on a one-dimensional coordinate gets a memory dependent character. Since RETIS is inherently non-Markovian in its description, it allows a much broader range of applications. An interesting route of thought could be the combination of our facilitating RETIS framework with the high-throughput methods that efficiently scan chemical space [70]

On the other hand, a milestone type approach avoids the creation of full transition trajectories which can be computationally demanding when the transit time through the membrane is long. In this case, PPTIS [51] could be an interesting alternative. PPTIS avoids the sampling of complete transition paths like milestone, but still maintains some of the history dependence. Alternatively, the exact non-Markovian character could be kept by alternating between short and long paths by means of stone skipping/web throwing [71]. Both PPTIS and stone skipping/web throwing can straightforwardly be implemented in our theoretical framework.

In conclusion, our permeability method presents a model-free approach for the computation of permeability and it is expected to become a valid standard method when membrane crossings are rare events.

## APPENDIX A: ENSEMBLE AVERAGES IN TRAJECTORY SPACE

At several instances in our article [e.g., Eqs. (6), (10), and (16)], we refer to phase space ensemble averages with the remark that these should actually be viewed as an average over “trajectory phase points.” Even if this point has mainly conceptual importance, we will outline here its mathematical interpretation since it is yet underreported in literature. One convenient way is to refer to path space ensemble averages instead of phase space ensemble averages [72]. Here, the path  $X = \{x_0, x_1, \dots, x_L\}$  can be viewed as a “chain of states” [73] with  $x_i$  the phase point that is visited after  $i$  MD steps, at time  $t = i\Delta t$  with  $\Delta t$  the time step. From this, the path probability follows as

$$P[X] = \rho(x_0) \prod_{i=0}^{L-1} p(x_i \rightarrow x_{i+1}), \quad (\text{A1})$$

where  $\rho(x_0)$  is the probability density of the initial state ( $x_0$  at  $t = 0$ ) of the path, and  $p(x_i \rightarrow x_{i+1})$  are the single time step transition probabilities. The latter are dependent on the type of dynamics. Mostly, we assume that  $\rho(\cdot)$  is the equilibrium phase space density given by the Boltzmann distribution:  $\rho(x) \propto \exp(-\beta E(x))$  with  $E(x)$  the total energy of phase point  $x$ . Actually, while  $P[X]$  and  $p(x_i \rightarrow x_{i+1})$  are commonly referred to as a type of probabilities, it would have been more accurate to call these probability densities as well.

Now, by expressing an observable  $f$  as a functional of  $X$ , the path ensemble average can be formally written as

$$\langle f \rangle = \int dX P[X] f[X] \text{ with } dX = \prod_{i=0}^L dx_i \quad (\text{A2})$$

As we assume microscopically time-reversible dynamics, we can write [72]

$$\rho(x_i)p(x_i \rightarrow x_{i+1}) = p(\bar{x}_{i+1} \rightarrow \bar{x}_i)\rho(x_{i+1}), \quad (\text{A3})$$

where  $\bar{x}$  refers to the momenta-reversed phase point: if  $x = (r, v)$  with  $r$  the configuration and  $v$  the particles' velocities, then  $\bar{x} = (r, -v)$ . Here, it is also assumed that  $\rho(x) = \rho(\bar{x})$  for any phase point  $x$ . Applying Eq. (A3) multiple times on Eq. (A1), allows us to write alternative expressions for the path probability [59]:

$$\begin{aligned} P[X] &= \rho(x_0)p(x_0 \rightarrow x_1)p(x_1 \rightarrow x_2)p(x_2 \rightarrow x_3)\dots \\ &= p(\bar{x}_1 \rightarrow \bar{x}_0)\rho(x_1)p(x_1 \rightarrow x_2)p(x_2 \rightarrow x_3)\dots \\ &= p(\bar{x}_1 \rightarrow \bar{x}_0)p(\bar{x}_2 \rightarrow \bar{x}_1)\rho(x_2)p(x_2 \rightarrow x_3)\dots \\ &= p(\bar{x}_1 \rightarrow \bar{x}_0)p(\bar{x}_2 \rightarrow \bar{x}_1)p(\bar{x}_3 \rightarrow \bar{x}_2)\rho(x_3)\dots \end{aligned}$$

In the TIS and RETIS theoretical framework, the path concept is extended by including time slices *before*  $x_0$ . In this view,  $x_0$  is considered the present state, the principle phase point, while  $x_i$  is a state in the future or in the past whenever  $i$  is, respectively, positive or negative. Hence, for a path  $X = \{x_{-M}, x_{-M+1}, \dots, x_{-1}, x_0, x_1, \dots, x_{L-1}, x_L\}$ , we can



write

$$P[X] = \rho(x_0) \left( \prod_{i=0}^{L-1} p(x_i \rightarrow x_{i+1}) \right) \left( \prod_{i=0}^{M-1} p(\bar{x}_{-i} \rightarrow \bar{x}_{-i-1}) \right). \tag{A4}$$

Using Eq. (A4), we can in principle redefine all phase space ensemble averages, in which one integrates over  $x$ , as path ensemble averages in which one integrates over  $x_0$  and additional phase points  $x_{i \neq 0}$  with both positive and negative index via Eq. (A2) with  $dX = \prod_{i=-M}^L dx_i$ . Whenever the value of  $f$  is instantaneously available from the present phase point  $x_0$ , the integrals over the additional phase points can be ignored, since they are unity.

Another way to generalize the ensemble average, which in some cases could be arguably more intuitive, can be derived from another perspective on the path object as stated by Crooks and Chandler [73]: “A stochastic trajectory can be defined by the chain of states that the system visits, but it can also be represented by the initial state and the set of random numbers, the noise history, that was used to generate the trajectory.” [73]. As an example they show that the probability density of a one-dimensional Brownian dynamics path  $X$  consisting of  $L$  time slices can be written as

$$P[X] = P\{x_0, x_1, \dots, x_L\} = \rho(x_0) \prod_{i=1}^L \frac{1}{\sqrt{2\pi\epsilon}} \exp(-\xi_i^2/2\epsilon),$$

where each  $\xi_i$  is a Gaussian random number of zero mean and  $\epsilon$  variance, the stochastic force acting at time between  $t = (i - 1)\Delta t$  to  $t = i\Delta t$ . Here, we deliberately shifted the indexing of the noise terms from 1 to  $L$  instead of the original [73] indexing from 0 to  $L - 1$ . The reason becomes clear when we introduce Eq. (A8).

Since the phase point of the system at  $t = \Delta t$ ,  $x_1$ , is fully determined by the first phase point and first stochastic noise term, we can write  $x_1 = \phi(x_0, \xi_1)$  with  $\phi$  being the MD time-step integrator. Likewise,  $x_2 = \phi(x_1, \xi_2) = \phi(\phi(x_0, \xi_1), \xi_2)$  etc. It is thus apparent that, when we add to  $x_0$  all the information of the random noise sequence  $\xi_1, \xi_2, \dots, \xi_L$  to make an “extended phase point” or “trajectory phase point,”  $\bar{x} = \{x_0, \xi_1, \dots, \xi_L\} = \{x_0, \xi^L\}$ , basically every property of the system between  $t = 0$  and  $L\Delta t$  becomes a function of  $\bar{x}$ , as if the dynamics would be deterministic.

Henceforth, for a general type of stochastic dynamics that proceeds via random noises that are drawn from a distribution  $p_\xi(\cdot)$ , we can define phase space density of an extended phase point  $\bar{x}$  as

$$\rho(\bar{x}) = P[X(\bar{x})] = \rho(x_0) \prod_{i=1}^L p_\xi(\xi_i). \tag{A5}$$

So equivalently to Eq. (A2), by expressing an observable  $f$  as a function of an extended phasepoint  $\bar{x}$ , we write its ensemble average as

$$\begin{aligned} \langle f \rangle &= \int d\bar{x} \rho(\bar{x}) f(\bar{x}) \text{ with} \\ d\bar{x} &= dx_0 \prod_{i=1}^L d\xi_i = dx_0 d\xi^L, \\ \rho(\bar{x}) &= \rho(x_0) \prod_{i=1}^L p_\xi(\xi_i) = \rho(x_0) p_\xi(\xi^L). \end{aligned} \tag{A6}$$

This automatically becomes a standard phase space ensemble average with  $x$  instead of  $\bar{x}$  when  $f$  is not noise-dependent since all integrals over  $d\xi_i$  become 1.

While the concept of an extended phase point is generally not explicitly referred to, it is often implicitly used. For instance, time-correlation functions are often casually introduced as  $C(t) = \langle a(0)b(t) \rangle$  without being specific about the noise dependence. Based on Eqs. (A5) and (A6), we can rigorously define the ensemble average as an integral over extended phase space:

$$\begin{aligned} C(t) &= \int \rho(\bar{x}) a(x_0) b(x_L) d\bar{x} \\ &= \int \rho(x_0) p_\xi(\xi^L) a(x_0) b(x_L(x_0, \xi^L)) dx_0 d\xi^L \end{aligned} \tag{A7}$$

with  $L = t/\Delta t$ ,

where  $a$  and  $b$  are functions of the phase point of the system at the time under consideration, at  $t = 0$  and  $L\Delta t$ , respectively. The absolute timescale is irrelevant here since we generally assume we are at an equilibrium distribution at  $t = 0$  and the dynamics conserves this distribution, i.e.  $\langle a(0)b(t) \rangle = \langle a(t')b(t + t') \rangle$  or any arbitrary  $t'$ . Hence, the correlation function  $C(t)$  becomes an ensemble average  $\langle a(\bar{x})b(\bar{x}; t) \rangle$  where one just integrates over  $\bar{x}$  and  $b$  is parametrically dependent on  $t$  in addition to its dependence on  $\bar{x}$ .

Comparing Eqs. (A1) and (A5), it is apparent that  $p(x_i \rightarrow x_{i+1}) = p_\xi(\xi_{i+1})$  for stochastic dynamics with  $\xi_{i+1}$  being the noise that forces the dynamics to produce  $x_{i+1}$  from  $x_i$ ;  $x_{i+1} = \phi(x_i, \xi_{i+1})$ . For deterministic dynamics, we can write  $p(x_i \rightarrow x_{i+1}) = \delta(x_{i+1} - \phi(x_i))$ . In addition, it is clear that the path interpretation and the extended phase point interpretation are equivalent; if one knows the initial phase point and the noise sequence, one knows the path  $X$  and vice versa.

As stated before, the TIS and RETIS theoretical framework requires the description of phase points before  $x_0$ . This means that the “noise history” term by Crooks and Chandler to denote  $\xi_+^L = \{\xi_1, \xi_2, \dots, \xi_L\}$  is now recoined as noise future while  $\xi_-^M = \{\xi_{-1}, \xi_{-2}, \dots, \xi_{-M}\}$  is the actual noise history or noise past.

Then, equivalent to Eq. (A4), we can define the phase space density of  $\bar{x} = \{\xi_{-M}, \dots, \xi_{-1}, x_0, \xi_1, \dots, \xi_L\}$  as

$$\begin{aligned} \rho(\bar{x}) &= P[X(\bar{x})] = \rho(x_0) \left( \prod_{i=1}^L p_\xi(\xi_i) \right) \left( \prod_{i=1}^M p_\xi(\xi_{-i}) \right) \\ &= \rho(x_0) p_\xi(\xi_+^L) p_\xi(\xi_-^M), \end{aligned} \tag{A8}$$

where the noise terms have a slightly different interpretations depending on the index being positive or negative. For  $i > 0$ ,  $\xi_i$  is the noise needed for  $\phi$  to produce  $x_i$  given  $x_{i-1}$ , while  $\xi_{-i}$  is the noise needed for  $\phi$  to produce  $\bar{x}_i$  given  $\bar{x}_{-i+1}$ :  $\phi(x_{i-1}, \xi_i) = x_i$  and  $\phi(\bar{x}_{i-1}, \xi_i) = \bar{x}_i$ . Hence, the history of the path  $X$  follows from the negative noise terms as:  $x_{-1} = \phi(\bar{x}_0, \xi_{-1})$ ,  $x_{-2} = \phi(\bar{x}_{-1}, \xi_{-2})$ , etc, again showing that there is a one-to-one relation between  $X$  and  $\bar{x}$ .

Based on Eq. (A8), it is now possible to define the probability of overall state  $\mathcal{A}$  as:  $p_{\mathcal{A}} = \langle h_{\mathcal{A}} \rangle$ . Here  $h_{\mathcal{A}} = h_{\mathcal{A}}(\bar{x}) =$

$h_A(X)$  equals

$$h_A(\bar{x}) = \sum_{n=0}^{\infty} h_A^{-n}(\bar{x}) \quad (\text{A9})$$

with

$$h_A^{-n}(\bar{x}) = h_A(x_{-n}) \prod_{i=0}^{n-1} (1 - h_A(x_{-i}) - h_B(x_{-i})) \quad (\text{A10})$$

such that  $h_A^{-n}$  is a function of  $(x_0, \xi_{-1}, \xi_{-2}, \dots, \xi_{-n})$  which is a part of  $\bar{x}$ . The product term is simply 1 if none of the points  $x_i$  with index  $-n < i \leq 0$  is inside  $A$  or  $B$ . Otherwise it is 0. Further,  $h_A^0(\bar{x})$  is simply  $h_A(x_0)$ . Likewise, we can define

$$h_B(\bar{x}) = \sum_{n=0}^{\infty} h_B^{-n}(\bar{x}) \quad \text{with} \quad (\text{A11})$$

$$h_B^{-n}(\bar{x}) = h_B(x_{-n}) \prod_{i=0}^{n-1} (1 - h_A(x_{-i}) - h_B(x_{-i})).$$

In principle, we consider the path or random noise sequence to extend to infinite in both time directions [ $L \rightarrow \infty$ ,  $M \rightarrow \infty$  in Eq. (A8)]. However, as  $h_A^{-n}(\bar{x})$  does not depend on  $\xi_i$  with  $i > -1$  nor  $i < n$ , many noise integrals are simply 1 and therefore

$$\begin{aligned} p_A \langle h_A \rangle &= \int \rho(\bar{x}) h_A(\bar{x}) d\bar{x} \quad (\text{A12}) \\ &= \int dx_0 \rho(x_0) \sum_{n=0}^{\infty} \int d\xi_{-1}^n p_{\xi}(\xi_{-1}^n) h_A^{-n}(x_0, \xi_{-1}^n) \end{aligned}$$

with  $d\xi_{-1}^n = d\xi_{-1} d\xi_{-2} \dots d\xi_{-n}$ .

Now, suppose  $f$  is a function of phase space:  $f = f(x)$ . Then, the ensemble average of  $f$  does not require the integration of any noise terms

$$\langle f \rangle = \int dx f(x) \rho(x) \quad (\text{A13})$$

though the conditional ensemble average  $\langle f \rangle_A$  does as

$$\begin{aligned} \langle f \rangle_A &= \frac{\langle f h_A \rangle}{\langle h_A \rangle} \quad (\text{A14}) \\ &= \frac{\int dx_0 f(x_0) \rho(x_0) \sum_{n=0}^{\infty} \int d\xi_{-1}^n p_{\xi}(\xi_{-1}^n) h_A^{-n}(x_0, \xi_{-1}^n)}{\langle h_A \rangle}. \end{aligned}$$

The division used in Eq. (10) can now be understood with this extended phase point picture in mind as

$$\begin{aligned} p_A \langle f \rangle_A + p_B \langle f \rangle_B &= \langle h_A \rangle \langle f \rangle_A + \langle h_B \rangle \langle f \rangle_B \\ &= \int dx_0 f(x_0) \rho(x_0) \sum_{n=0}^{\infty} \\ &\quad \times \int d\xi_{-1}^n p_{\xi}(\xi_{-1}^n) [h_A^{-n}(x_0, \xi_{-1}^n) + h_B^{-n}(x_0, \xi_{-1}^n)]. \quad (\text{A15}) \end{aligned}$$

If we consider the third line of Eq. (A15) separately, we can identify this, for a given phase point  $x_0$ , as the chance that the stochastic dynamics needed exactly  $n$  steps backward in time to move outside *no man's land*, i.e., enter either stable state  $A$  or  $B$ . Since we assume that no point  $x_0$  can be trapped into

man's land forever, the sum over  $n$  of this probability equals 1. Hence,

$$p_A \langle f \rangle_A + p_B \langle f \rangle_B = \int dx_0 f(x_0) \rho(x_0) = \langle f \rangle. \quad (\text{A16})$$

Note that the we can use integration over  $x$  [Eq. (A13)] or  $x_0$  [Eq. (A16)] interchangeably since  $\rho(\cdot)$  refers to the equilibrium phase density that is time-invariant. So indeed,  $\langle \dots \rangle = p_A \langle \dots \rangle_A + p_B \langle \dots \rangle_B$  like stated in Eq. (10).

As a special case, we can take  $f(x; z) = \delta(z - z_t)$  with  $z_t$  being the  $z$  coordinate of a specific particle (the target permeant). Here,  $z_t$  is a part of the system's phase point  $x$  that on its turn can be viewed as  $x_0$  which a part of  $\bar{x}$  (recall that the phase space density is time-invariant). In addition,  $z$  is a parameter that specifies a reference region in configuration space. Note that the parametric dependence of  $f$  on  $z$  is not vanishing when taking the ensemble average since it is not a part of  $\bar{x}$  and therefore not integrated out. Therefore, we can write

$$\begin{aligned} \rho(z) &= \langle \delta(z - z_t) \rangle = \langle f(x; z) \rangle \quad (\text{A17}) \\ &= p_A \langle f(\bar{x}; z) \rangle_A + p_B \langle f(\bar{x}; z) \rangle_B \\ &= p_A \frac{\langle f(x_0; z) h_A(\bar{x}) \rangle}{\langle h_A \rangle} + p_B \frac{\langle f(x_0; z) h_B(\bar{x}) \rangle}{\langle h_B \rangle} \\ &= p_A \frac{\langle \delta(z - z_t(x_0)) h_A(\bar{x}) \rangle}{\langle h_A \rangle} + p_B \frac{\langle \delta(z - z_t(x_0)) h_B(\bar{x}) \rangle}{\langle h_B \rangle}. \end{aligned}$$

All ensemble averages in Eq. (A17) are in principle integrals over  $\bar{x}$ , though in the first line an integral over configuration space would be sufficient since the integrals over momenta and noise terms are unity. In the last line of Eq. (A17), the integrals need to be carried out on the principle phase point  $x_0$  and the backward noise terms  $\xi_{-1}, \xi_{-2}, \dots$ . The integrals over the forward noises  $\xi_1, \xi_2, \dots$  are still unity.

The delta-function is only nonzero whenever  $z_t(x_0)$  equals  $z$ . This implies that if  $z$  is inside stable state  $A$ , then the product  $\delta(z - z_t(x_0)) h_B(\bar{x})$  is by definition zero; if  $z_t(x_0) = z \in A$  then  $x_0 \in A$  and, therefore,  $h_B(\bar{x}) = 0$ . This explains the last equality of Eq. (16) where  $z = z_{\text{ref}} \in A$ .

Finally, for the rate in Eq. (6), the product of  $h_A(0)$  and  $h_B(\Delta t)$  should be evaluated. By defining the principle phase point to be  $x_0$ , we have  $\bar{x} = (\dots, \xi_{-2}, \xi_{-1}, x_0, \xi_1, \dots)$  and  $x_1 = \phi(x_0, \xi_1)$  with  $\phi$  the  $\Delta t$  time step integrator. Further, the product can only be nonzero if both  $h_A(\bar{x})$  and  $h_B(x_1)$  are equal to 1, and  $h_B(\Delta t)$  can be replaced by  $h_B(\phi(x_0, \xi_1))$  in the product,

$$\begin{aligned} h_A(0) h_B(\Delta t) &= h_A(\dots, \xi_{-1}, x_0, \xi_1, \dots) h_B(\phi(x_0, \xi_1)) \\ &= h_A(\bar{x}) h_B(\phi(x_0, \xi_1)). \end{aligned}$$

The ensemble average in Eq. (6) can be written as

$$\begin{aligned} k &= \lim_{\Delta t \rightarrow 0} \frac{\langle h_A(0) h_B(\Delta t) \rangle}{\langle h_A \rangle \Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{1}{\langle h_A \rangle \Delta t} \int dx_0 \rho(x_0) \int d\xi_1 p_{\xi}(\xi_1) h_B(\phi(x_0, \xi_1)) \\ &\quad \times \sum_{n=0}^{\infty} \int d\xi_{-1}^n p_{\xi}(\xi_{-1}^n) h_A^{-n}(x_0, \xi_{-1}^n) \quad (\text{A18}) \end{aligned}$$

where noise history and noise future appear within one equation. The limit  $\Delta t \rightarrow 0$  only exists formally since  $\Delta t$  will be taken equal to the typical MD step in any practical case.

## APPENDIX B: THEORETICAL PERMEABILITY FOR 1D TOY SYSTEM

In the 1D toy system, we can express the permeability in an analytical shape for the following three situations: for a Brownian particle (based on the Smoluchowski equation), for a Langevin particle crossing a high barrier (based on the Kramers equation), and for a deterministic Newtonian particle.

The Smoluchowski equation for a Brownian particle leads to the permeability expression in Eq. (3). In the one-dimensional case, the absence of any other degrees of freedom implies that the free energy  $F(z)$  and the potential energy  $V(z)$  are the same. Hence, for overdamped dynamics, we can derive a theoretical expression for  $P$  by inserting  $V(z)$  of Eq. (39) as  $F(z)$  into Eq. (3),

$$P = \frac{D}{h} \frac{e^{-\beta V_0/2}}{I_0(\beta V_0/2)}. \quad (\text{B1})$$

Here,  $h = 2a$  and  $I_0(x) = (1/\pi) \int_0^\pi \exp(\cos\theta) d\theta$  is the 0th order modified Bessel function of the first kind. When  $V_0 = 0$ , then  $I_0(0) = 1$ , and the resulting flat potential yields  $P = D/h$ . For large  $V_0$ , the cosine barrier can be approximated by a second order Taylor expansion about  $z = 0$ ,  $F(z) = V(z) \approx V_0(1 - (\frac{\pi z}{h})^2)$  and it can be assumed that  $\exp(-\beta V(z))$  rapidly decays when moving away from the membrane. This can be inserted into Eq. (3) and the integration boundaries can be moved from  $\pm h/2$  to  $\pm\infty$ . Solving the resulting Gaussian integral yields an approximation of  $P$  for large  $V_0$ ,

$$P = \frac{D}{h} \sqrt{\frac{\pi V_0}{k_B T}} e^{-\beta V_0}. \quad (\text{B2})$$

An alternative approach to the Smoluchowski approach is to use Kramer's relation for the rate constant  $k$  instead. The permeability  $P$  is then obtained via Eq. (18) by first computing the rate  $k$  while assuming a hard wall at  $z = -W$  that can be taken to infinite. Using a harmonic approximation and a high barrier assumption, this rate constant  $k$  can be written as [74]

$$k = \kappa \sqrt{\frac{k_B T}{2\pi m}} \frac{\exp(-\beta V(0))}{\int_{-W}^0 \exp(-\beta V(z)) dz}, \quad (\text{B3})$$

where  $\kappa$  is the transmission coefficient that can be approximated using Kramers' relation

$$\kappa = \frac{1}{\omega_+} \left( -\frac{\gamma}{2} + \sqrt{\frac{\gamma^2}{4} + \omega_+^2} \right). \quad (\text{B4})$$

Here,  $\omega_+$  is the frequency associated to the curvature at the top of the barrier:  $\omega_+ = \sqrt{k_+/m}$  with  $V(z) \approx V_0 - \frac{1}{2}k_+z^2$ . From the above Taylor expansion, we have  $k_+ = 2V_0\pi^2/h^2$  and  $\omega_+ = (\pi/h)\sqrt{2V_0/m}$ .

The conditional probability appearing in Eq. (18) is expressed as

$$(\rho_{\text{ref}})_A = \frac{1}{\int_{-W}^0 \exp(-\beta V(z)) dz} \quad (\text{B5})$$

where we assumed that overall state  $\mathcal{A}$  condition is statistically equivalent to the condition  $z < 0$  for this case, which is a valid assumption for a high barrier.

Inserting Eq. (B4) in Eq. (B3) and inserting Eqs. (B3) and (B5) in Eq. (18) gives the permeability for a high barrier,

$$P = \kappa \sqrt{\frac{k_B T}{2\pi m}} e^{-\beta V_0} \quad (\text{B6})$$

$$= \frac{h}{\pi} \sqrt{\frac{m}{2V_0}} \left( -\frac{\gamma}{2} + \sqrt{\frac{\gamma^2}{4} + \frac{2V_0\pi^2}{mh^2}} \right) \sqrt{\frac{k_B T}{2\pi m}} e^{-\beta V_0}.$$

In the high friction limit where  $\gamma \gg \omega_+$ , Eq. (B4) reduces to  $\kappa = \omega_+/\gamma$ , and  $P$  for large  $V_0$  becomes

$$P = \frac{\pi}{h\gamma} \sqrt{\frac{2V_0}{m}} \sqrt{\frac{k_B T}{2\pi m}} e^{-\beta V_0}$$

$$= \frac{k_B T}{h\gamma m} \sqrt{\frac{\pi V_0}{k_B T}} e^{-\beta V_0}. \quad (\text{B7})$$

Since  $D = k_B T/(m\gamma)$ , this equation is equal to the Smoluchowski equation Eq. (B2) within the harmonic approximation for the high barrier. A Langevin particle with high friction is indeed well described by the overdamped dynamics of a Brownian particle.

In the low friction limit  $\gamma \ll \omega_+$ , Eq. (B4) reduces to  $\kappa = 1$ , and  $P$  in Eq. (B6) becomes, for any  $V_0$ ,

$$P = \sqrt{\frac{k_B T}{2\pi m}} e^{-\beta V_0}. \quad (\text{B8})$$

This friction-less limit is exactly the permeability of the deterministic particle. It can also be obtained from Eq. (24). Here,  $\xi = 1/2$ , since a deterministic particle in a flat free energy region either moves to the right (velocity positive), either to the left (velocity negative), which have equal Boltzmann probability. A particle moving to the right will reach the barrier top with a probability  $\exp(-\beta V_0)$  and it will not recross, and therefore  $P_A(\lambda_B|\lambda_A) = \exp(-\beta V_0)$ . The time spent per path in  $[0^-]$  in a reference region of size  $\Delta z$  can be computed from the flux-weighted velocity distribution as

$$\tau_{\text{ref},[0^-]} = \Delta z \sqrt{\pi \beta m/2}. \quad (\text{B9})$$

Inserting these three factors into Eq. (24) gives Eq. (B8).

Let us recap the case of a flat potential membrane ( $V_0 = 0$ ). For the Brownian particle, the permeability is  $P = D/h$ . If the particle has low friction  $\gamma \rightarrow 0$ , then  $D \rightarrow \infty$ , and the permeability diverges. For the Langevin particle, the high friction limit of  $P$  in Eq. (B7) based on Kramer's relation vanishes when  $V_0 = 0$ , which is not an adequate approximation of a flat potential's permeability. Nevertheless, again considering a Langevin particle and Kramer's equation, the low friction limit in Eq. (B8) converges to  $P = \sqrt{k_B T/(2\pi m)}$ , which is finite.

In conclusion, the two theoretical expressions for the one-dimensional case, Eq. (B1) based Smoluchowski and



Eq. (B6) based on Kramers, use respectively an overdamped assumption or a harmonic approximation to describe the top of the barrier. For high friction and low barriers, Eq. (B1) will be more accurate than Eq. (B6). For high barriers and low friction Eq. (B6) will prevail over

Eq. (B1). In the case that both the friction and the barrier is high, both converge to the same value. In the case that both the friction and the barrier is low, neither Eq. (B1) nor Eq. (B6) will be accurate descriptions of a Langevin particle.

- 
- [1] E. Awoonor-Williams and C. N. Rowley, Molecular Simulation of Nonfacilitated Membrane Permeation, *Biochim. Biophys. Acta, Biomembr.* **1858**, 1627 (2016).
- [2] P. Berben, A. Bauer-Brandl, M. Brandl, B. Faller, G. E. Flaten, A. C. Jacobsen, J. Brouwers, and P. Augustijns, Drug permeability profiling using cell-free permeation tools: Overview and applications, *Eur. J. Pharm. Sciences* **119**, 219 (2018).
- [3] B. J. Bennion, N. A. Be, M. W. McNerney, V. Lao, E. M. Carlson, C. A. Valdez, M. A. Malfatti, H. A. Enright, T. H. Nguyen, F. C. Lightstone, and T. S. Carpenter, Predicting a drugs membrane permeability: A computational model validated with in vitro permeability assay data, *J. Phys. Chem. B* **121**, 5228 (2017).
- [4] E. N. Petersen, H. W. Chung, A. Nayebosadri, and S. B. Hansen, Kinetic disruption of lipid rafts is a mechanosensor for phospholipase D, *Nat. Commun.* **7**, 13873 (2016).
- [5] B. Smit and T. L. M. Maesen, Molecular simulations of zeolites: Adsorption, diffusion, and shape selectivity, *Chem. Rev.* **108**, 4125 (2008).
- [6] A. Ghysels, S. L. C. Moors, K. Hemelsoet, K. De Wispelaere, M. Waroquier, G. Sastre, and V. Van Speybroeck, Shape-selective diffusion of olefins in 8-ring solid acid microporous zeolites, *J. Phys. Chem. C* **119**, 23721 (2015).
- [7] E. L. Elson, Fluorescence correlation spectroscopy: Past, present, future, *Biophys. J.* **101**, 2855 (2011).
- [8] M. Przybylo, A. Olzyska, S. Han, A. Ozyhar, and M. Langner, A fluorescence method for determining transport of charged compounds across lipid bilayer, *Biophys. Chem.* **129**, 120 (2007).
- [9] W. K. Subczynski, M. Pasenkiewicz-Gierula, R. N. McElhaney, J. S. Hyde, and A. Kusumi, Molecular dynamics of 1-palmitoyl-2-oleoylphosphatidylcholine membranes containing transmembrane  $\alpha$ -helical peptides with alternating leucine and alanine residues, *Biochemistry* **42**, 3939 (2003).
- [10] R. M. Venable, A. Krämer, and R. W. Pastor, Molecular dynamics simulations of membrane permeability, *Chem. Rev.* **119**, 5954 (2019).
- [11] R. J. Dotson, K. Smith, Bueche, G. Angles, and S. C. Pias, Influence of cholesterol on the oxygen permeability of membranes: Insight from atomistic simulations, *Biophys. J.* **112**, 2336 (2017).
- [12] A. Krämer, A. Ghysels, E. Z. Wang, R. M. Venable, J. B. Klauda, B. Brooks, and R. W. Pastor, Membrane permeability of small molecules from unbiased molecular dynamics simulations, *J. Chem. Phys.* **153**, 124107 (2020).
- [13] S. Marrink and H. J. C. Berendsen, Simulation of water transport through a lipid membrane, *J. Phys. Chem.* **98**, 4155 (1994).
- [14] D. J. Bicout and A. Szabo, Electron transfer reaction dynamics in non-Debye solvents, *J. Chem. Phys.* **109**, 2325 (1998).
- [15] G. Hummer, Position-dependent diffusion coefficients and free energies from Bayesian analysis of equilibrium and replica molecular dynamics simulations, *New J. Phys.* **7**, 34 (2005).
- [16] A. Ghysels, R. M. Venable, R. W. Pastor, and G. Hummer, Position-dependent diffusion tensors in anisotropic media from simulation: Oxygen transport in and through membranes, *J. Chem. Theory Comput.* **13**, 2962 (2017).
- [17] Z. Yue, C. Li, G. A. Voth, and J. M. J. Swanson, Dynamic protonation dramatically affects the membrane permeability of drug-like molecules, *J. Am. Chem. Soc.* **141**, 13421 (2019).
- [18] A. K. Faradjian and R. Elber, Computing time scales from reaction coordinates by milestoning, *J. Chem. Phys.* **120**, 10880 (2004).
- [19] E. Vanden-Eijnden, M. Venturoli, G. Ciccotti, and R. Elber, On the assumptions underlying milestoning, *J. Chem. Phys.* **129**, 174102 (2008).
- [20] B. Peters, *Reaction Rate Theory and Rare Events* (Elsevier, Amsterdam, Netherlands, 2017).
- [21] T. van Erp, D. Moroni, and P. Bolhuis, A novel path sampling method for the calculation of rate constants, *J. Chem. Phys.* **118**, 7762 (2003).
- [22] T. S. van Erp, Reaction Rate Calculation by Parallel Path Swapping, *Phys. Rev. Lett.* **98**, 268301 (2007).
- [23] R. Cabriolu, K. M. S. Refsnes, P. G. Bolhuis, and T. S. van Erp, Foundations and latest advances in replica exchange transition interface sampling, *J. Chem. Phys.* **147**, 152722 (2017).
- [24] I. I. Ivanov, G. E. Fedorov, R. A. Guskova, K. I. Ivanov, and A. B. Rubin, Permeability of lipid membranes to dioxygen, *Biochem. Biophys. Res. Commun.* **322**, 746 (2004).
- [25] J. Widomska, M. Raguz, and W. K. Subczynski, Oxygen permeability of the lipid bilayer membrane made of calf lens lipids, *BBA - Biomembranes* **1768**, 2635 (2007).
- [26] M. Möller, J. Lancaster, and A. Denicola, The interaction of reactive oxygen and nitrogen species with membranes, *Curr. Top. Membr.* **61**, 23 (2007).
- [27] M. N. Möller, Q. Li, C. Chinnaraj, H. C. Cheung, J. R. Lancaster Jr., and A. Denicola, Solubility and diffusion of oxygen in phospholipid membranes, *Biochim. Biophys. Acta* **1858**, 2923 (2016).
- [28] O. De Vos, T. Van Hecke, and A. Ghysels, Effect of chain unsaturation and temperature on oxygen diffusion through lipid membranes from simulations, *Oxygen Transport to Tissue XL, Advances in Experimental Medicine and Biology* **1072**, 399 (2018).
- [29] A. Ghysels, A. Krämer, R. Venable, W. Teague, E. Lyman, K. Gawrisch, and R. W. Pastor, Permeability of membranes in the liquid ordered and liquid disordered phases, *Nat. Commun.* **10**, 5616 (2019).
- [30] S. Davoudi and A. Ghysels, Sampling efficiency of the counting method for permeability calculations estimated with the inhomogeneous solubility-diffusion model, *J. Chem. Phys.* **154**, 054106 (2021).
- [31] E. Sezgin, I. Levental, S. Mayor, and C. Eggeling, The mystery of membrane organization: composition, regulation and roles of lipid rafts, *Nat. Rev. Mol. Cell Biol.* **18**, 361 (2017).

- [32] C. Dietrich, L. A. Bagatolli, Z. N. Volovyk, N. L. Thompson, M. Levi, K. Jacobson, and E. Gratton, Lipid rafts reconstituted in model membranes, *Biophys. J.* **80**, 1417 (2001).
- [33] A. V. Samsonov, I. Mihalyov, and F. S. Cohen, Characterization of cholesterol-sphingomyelin domains and their dynamics in bilayer membranes, *Biophys. J.* **81**, 1486 (2001).
- [34] O. De Vos, R. M. Venable, T. Van Hecke, G. Hummer, R. W. Pastor, and A. Ghysels, Membrane permeability: Characteristic times and lengths for oxygen and a simulation-based test of the inhomogeneous solubility-diffusion model, *J. Chem. Theory Comput.* **14**, 3811 (2018).
- [35] G. Torrie and J. Valleau, Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling, *J. Comput. Phys.* **23**, 187 (1977).
- [36] E. Darve and A. Pohorille, Calculating free energies using average force, *J. Chem. Phys.* **115**, 9169 (2001).
- [37] J. Comer, C. Chipot, and F. D. González-Nilo, Calculating position-dependent diffusivity in biased molecular dynamics simulations, *J. Chem. Theory Comput.* **9**, 876 (2013).
- [38] C. T. Lee, J. Comer, C. N. Herndon, N. Leung, A. Pavlova, R. V. Swift, C. Tung, C. N. Rowley, R. E. Amaro, C. Chipot, Y. Wang, and J. C. Gumbart, Simulation-Based Approaches for Determining Membrane Permeability of Small Compounds, *J. Chem. Inf. Model.* **56**, 721 (2016).
- [39] M. Badaoui, A. Kells, C. Molteni, C. J. Dickson, V. Hornak, and E. Rosta, Calculating kinetic rates and membrane permeability from biased simulations, *J. Phys. Chem. B* **122**, 11571 (2018).
- [40] A. Krämer, R. W. Pastor, and A. Ghysels, Membrane permeability of small molecules from biased molecular dynamics simulations (unpublished).
- [41] J. Comer, K. Schulten, and C. Chipot, Calculation of lipid-bilayer permeabilities using an average force, *J. Chem. Theory Comput.* **10**, 554 (2014).
- [42] A. E. Cardenas and R. Elber, Computational study of peptide permeation through membrane: Searching for hidden slow variables, *Mol. Phys.* **111**, 3565 (2013).
- [43] A. E. Cardenas and R. Elber, Modeling kinetics and equilibrium of membranes with fields: Milestoning analysis and implication to permeation, *J. Chem. Phys.* **141**, 054101 (2014).
- [44] A. Fathizadeh and R. Elber, Ion permeation through a phospholipid membrane: Transition state, path splitting, and calculation of permeability, *J. Chem. Theory Comput.* **1**, 720 (2019).
- [45] L. W. Votapka, C. T. Lee, and R. E. Amaro, Two relations to estimate membrane permeability using milestoning, *J. Phys. Chem. B* **120**, 8606 (2016).
- [46] A. M. Berezhkovskii and A. Szabo, Committers, first-passage times, fluxes, markov states, milestones, and all that, *J. Chem. Phys.* **150**, 054106 (2019).
- [47] P. L. Geissler, C. Dellago, and D. Chandler, Kinetic pathways of ion pair dissociation in water, *J. Phys. Chem. B* **103**, 3706 (1999).
- [48] Weinan E, W. Ren, and E. Vanden-Eijnden, Transition pathways in complex systems: Reaction coordinates, isocommittor surfaces, and transition tubes, *Chem. Phys. Lett.* **413**, 242 (2005).
- [49] B. Peters, G. T. Beckham, and B. L. Trout, Extensions to the likelihood maximization approach for finding reaction coordinates, *J. Chem. Phys.* **127**, 034109 (2007).
- [50] R. Best and G. Hummer, Reaction coordinates and rates from transition paths, *Proc. Natl. Acad. Sci. U.S.A.* **102**, 6732 (2005).
- [51] D. Moroni, P. G. Bolhuis, and T. S. van Erp, Rate constant for diffusive processes by partial path sampling, *J. Chem. Phys.* **120**, 4055 (2004).
- [52] R. J. Allen, P. B. Warren, and P. R. ten Wolde, Sampling Rare Switching Events in Biochemical Networks, *Phys. Rev. Lett.* **94**, 018104 (2005).
- [53] D. Aristoff, T. Lelièvre, C. G. Mayne, and I. Teo, Adaptive multilevel splitting in molecular dynamics simulations, *ESAIM: Proc.* **48**, 215 (2015).
- [54] T. S. van Erp, Dynamical rare event simulation techniques for equilibrium and nonequilibrium systems, *Kinetics and Thermodynamics of Multistep Nucleation and Self-Assembly in Nanoscale Materials* (John Wiley & Sons, Ltd, 2012), Chap. 2, pp. 27–60.
- [55] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, OpenPathSampling: A Python Framework for Path Sampling Simulations. 1. Basics, *J. Chem. Theory Comput.* **15**, 813 (2019).
- [56] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, OpenPathSampling: A python framework for path sampling simulations. 2. Building and customizing path ensembles and sample schemes, *J. Chem. Theory Comput.* **15**, 837 (2019).
- [57] A. Lervik, E. Riccardi, and T. S. van Erp, PyRETIS: A well-done, medium-sized python library for rare events, *J. Comput. Chem.* **38**, 2439 (2017).
- [58] E. Riccardi, A. Lervik, S. Roet, O. A. en, and T. S. van Erp, PyRETIS 2: An improbability drive for rare events, *J. Comput. Chem.* **41**, 370 (2020).
- [59] T. S. van Erp and P. Bolhuis, Elaborating transition interface sampling methods, *J. Comput. Phys.* **205**, 157 (2005).
- [60] B. Peters and B. L. Trout, Obtaining reaction coordinates by likelihood maximization, *J. Chem. Phys.* **125**, 054108 (2006).
- [61] D. Earl and M. Deem, Parallel tempering: Theory, applications, and new perspectives, *Phys. Chem. Chem. Phys.* **7**, 3910 (2005).
- [62] T. S. van Erp, Efficient path sampling on multiple reaction channels, *Comput. Phys. Commun.* **179**, 34 (2008).
- [63] T. J. H. Vlugt and B. Smit, On the efficient sampling of pathways in the transition path ensemble, *Phys. Chem. Comm.* **4**, 11 (2001).
- [64] M. Grunwald, E. Rabani, and C. Dellago, Mechanisms of the Wurtzite to Rocksalt Transformation in CdSe Nanocrystals, *Phys. Rev. Lett.* **96**, 255701 (2006).
- [65] L. Verlet, Computer experiments on classical fluids. i. thermodynamical properties of lennard-jones molecules, *Phys. Rev.* **159**, 98 (1967).
- [66] M. Moqadam, A. Lervik, E. Riccardi, V. Venkatraman, B. K. Alsberg, and T. S. van Erp, Local initiation conditions for water autoionization, *Proc. Natl. Acad. Sci. U.S.A.* **115**, E4569 (2018).
- [67] E. Riccardi, A. Krämer, T. van Erp, and A. Ghysels, Permeation rates of oxygen transport through POPC membrane using replica exchange transition interface sampling, *J. Phys. Chem. B* **125**, 193 (2021).
- [68] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087 (1953).
- [69] W. Hastings, Monte-Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**, 97 (1970).

- [70] R. Roberto Menichetti, K. H. Kanekal, and T. Tristan Bereau, Drug-membrane permeability across chemical space, *ACS Cent. Sci.* **5**, 290 (2019).
- [71] E. Riccarfdi, O. Dahlen, and T. S. van Erp, Fast decorrelating monte carlo moves for efficient path sampling, *J. Phys. Chem. Lett.* **8**, 4456 (2017).
- [72] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Transition path sampling, *Advances in Chemical Physics* (John Wiley & Sons, Ltd., 2002), Chap. 1, pp. 1–78 .
- [73] G. E. Crooks and D. Chandler, Efficient transition path sampling for nonequilibrium stochastic dynamics, *Phys. Rev. E* **64**, 026109 (2001).
- [74] D. Frenkel and B. Smit, *Understanding Molecular Simulations from Algorithms to Applications* (Academic, San Diego, California, U.S.A., 2002).



## Paper C

# Chemistrees: Data-Driven Identification of Reaction Pathways via Machine Learning

Sander Roet, Christopher D. Daub, and Enrico Riccardi

*J. Chem. Theor. Comput.* **2021**, *17*, 6193–6202;

doi: 10.1021/acs.jctc.1c00458



# Chemistrees: Data-Driven Identification of Reaction Pathways via Machine Learning

Sander Roet,\* Christopher D. Daub, and Enrico Riccardi

Cite This: *J. Chem. Theory Comput.* 2021, 17, 6193–6202

Read Online

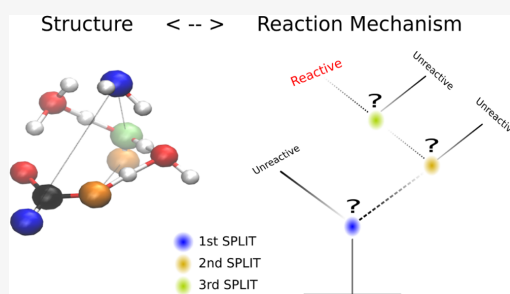
ACCESS |

Metrics & More

Article Recommendations

Supporting Information

**ABSTRACT:** We propose to analyze molecular dynamics (MD) output via a supervised machine learning (ML) algorithm, the decision tree. The approach aims to identify the predominant geometric features which correlate with trajectories that transition between two arbitrarily defined states. The data-driven algorithm aims to identify these features without the bias of human “chemical intuition”. We demonstrate the method by analyzing the proton exchange reactions in formic acid solvated in small water clusters. The simulations were performed with *ab initio* MD combined with a method to efficiently sample the rare event, path sampling. Our ML analysis identified relevant geometric variables involved in the proton transfer reaction and how they may change as the number of solvating water molecules changes.



## 1. INTRODUCTION

In regions far from urban areas, formic acid (FA) has been recognized as one of the main factors which reduces the pH of rainwater, causing acid rain.<sup>1</sup> It has relatively high atmospheric concentrations<sup>2,3</sup> and contributes to the formation of sulfuric acid in the atmosphere.<sup>4,5</sup> Enhanced description of proton exchange reactions involving solvated FA can improve the current atmospheric models. Theoretical studies of proton transport in bulk aqueous media have a long history going back to the elucidation of the Grothuss mechanism.<sup>6</sup> The current view of the solvated proton in water focuses on the formation of Zundel ( $\text{H}_5\text{O}_2^+$ ) and Eigen ( $\text{H}_9\text{O}_4^+$ ) cations and the mechanisms describing transformations between these states.<sup>7–13</sup>

A related area with significant theoretical and computational contributions in the last decade is the study of acid ionization in bulk water<sup>14–18</sup> or at the water–air interface.<sup>9,19–23</sup> By contrast, there are only a few papers which focus on the nature of acidic proton transport in small water clusters.<sup>24–30</sup> In these small systems, thermodynamic approaches appropriate for the bulk system are no longer valid. Instead, these studies have been forced to approach each specific chemical example as a separate problem. As such, the use of a generalizable approach such as the one we present in this study should be of considerable interest.

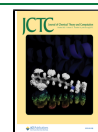
*Ab initio* molecular dynamics (MD) simulations have recently been used to examine FA deprotonation in aqueous solution,<sup>18,22</sup> successfully describing the proton exchange reaction between water and FA. While these studies led to valuable new insights, the limitations of the adopted methods (e.g., usage of a bias potential and continuous collective

variables) could be overcome, thanks to relatively novel methodologies such as replica exchange transition interface sampling (RETIS).<sup>31,32</sup> Respecting the natural dynamics of the system, it allows the study of transitions even with a significant diffusive contribution<sup>33,34</sup> (i.e., a small reaction barrier) and enables the direct investigation of reaction mechanisms.

RETIS is a rare event method developed to investigate transitions. Its main advantages are as follows: (a) it does not alter the natural dynamics of the system, (b) it does not require a particularly accurate order parameter, (c) its results are in principle identical to what would be obtained by an infinitely long unbiased MD simulation. With RETIS, the transition region is explored by continuously generating new paths which start from a stable state and end up either back in such a state (an *unreactive* path), or reach a different state (a *reactive* path). The approach has been successfully employed to study transitions that would, otherwise, require prohibitively long simulation times. The results generated have been used to describe the dynamics of chemical processes (e.g., reaction rates) while considering the entropic contribution in the analysis.<sup>5,33–36</sup> Since significant amounts of data are often generated by the sampling procedure, approaches to pragmatically decode reaction mechanisms are greatly beneficial.

Received: May 10, 2021

Published: September 24, 2021



Our aim is to establish a heuristic approach to describe transitions regardless of whether they involve crossing an entropic barrier. Data-driven, physically consistent, and measurable system descriptors might be generated and their correlations with the system dynamics asserted. It is a classification problem, which a machine learning (ML) algorithm can be trained to solve. The algorithm might then predict if a certain molecular structure (frame) is part of a reactive or a non-reactive trajectory. Connecting the descriptors to measurable quantities provides a data-driven “unbiased” description of a transition that might support, and eventually surpass, human-biased “chemical intuition”.

Data-driven algorithms for enhanced sampling or the analysis of chemical simulations have significant recent contributions.<sup>37–43</sup> Most of these approaches are based on neural networks, which lack physically consistent interpretability, which is, instead, a characteristic of decision trees (DTs).<sup>39,40</sup> Furthermore, in most of these studies implementing neural networks, a pre-selection of trial collective variables<sup>38,41,42</sup> is required, which could lead to a hypothesis-bias. DT<sup>44</sup> classifiers have a unique solution and are not sensitive to highly correlated variables. The results can be readily interpreted if the source variables are also interpretable. The approach was previously adopted to select optimal collective variables with DTs, with reasonable success.<sup>36</sup>

We here propose a method based on DTs, which is both interpretable and hypothesis-bias-free *via* an appropriate system representation invariant to system translation, rotation, and changes in atomic indices. Our aim is to gain insights into reaction mechanisms with a systematic and objective representation of the system.

The approach has been developed with sufficient versatility to be applied to different types of molecular simulations, from conventional MD to rare event methods. It should be noted that conventional MD would require a *a priori* classification of the data, that is, dividing the source trajectory into reactive and unreactive segments. The sampling strategy of rare event methods, instead, generates a data structure which inherently classifies the trajectories. Regardless of the adopted molecular simulation approach, limiting the correlation between samples is a primary task for a quantitative data-driven method to identify reaction paths and the probability of their occurrence.

We demonstrate our data-driven method in this study on small clusters of FA solvated by water,  $\text{HCOOH} + (\text{H}_2\text{O})_n$ ,  $n = 4$  and 6. The system is relatively small and well understood and hence provides an ideal test case for training an ML method. Our analysis provides new quantitative and qualitative insights into the acid–water proton transfer reaction in aqueous clusters.

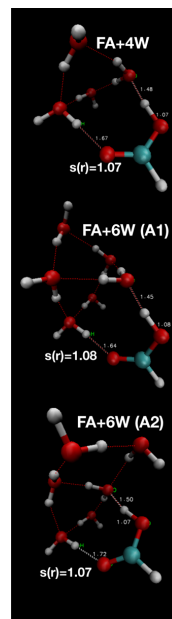
## 2. COMPUTATIONAL MODELS AND METHODS

Since the main focus of the present paper is an ML methodology, we provide only a brief introduction to the simulation methodology. Please consider our previous studies<sup>5,18</sup> for further details.

**2.1. System Description.** For studying proton transport, molecular simulations able to consider bond formation and bond breaking are required. Born–Oppenheimer MD has been shown to be a suitable approximation in previous studies of atmospheric reactions<sup>5,5</sup> and of aqueous FA.<sup>18,22</sup> The density functional theory BLYP, implemented in the Quickstep module of CP2K,<sup>45</sup> has been adopted with a double-zeta

basis set supplemented by the use of Grimme’s D2 dispersion correction.<sup>46</sup>

A set of systems with an increasing number of water molecules around FA were studied. Initial configurations were obtained from minimum energy configurations, of which snapshots are reported in Figure 1. As more water molecules



**Figure 1.** Minimum energy configurations for systems with FA associated with four and six water molecules. In the figures,  $s(r)$  is equal to the  $r_{\text{OH}}$  of the initially protonated FA molecule. These configurations are the initial states used to initiate PyRETIS simulations.

were added, the probability of generating reactive trajectories increased. However, at least four added water molecules were required to allow generation of trajectories with a significant charge separation between the deprotonated FA and the solvated proton. With two additional water molecules, a significantly higher proton transfer rate was measured. The two systems composed by FA surrounded by four and six water molecules have been thus selected and discussed here.

**2.2. Definition of the Collective Variable.** Path sampling simulation requires the definition of a collective variable,  $s(r)$ , to quantify the progress of a transition ( $r$  contains the positions and velocities of all atoms in the system). The method is not limited to continuous collective variables, allowing the consideration of relatively complex functions to describe proton transport.

The collective variable adopted in the present work is inspired by the study of water ionization,<sup>36</sup> with modifications introduced to consider acid deprotonation. As a first step, it locates the smallest distance between any FA oxygen and any reactive hydrogen in the system (excluding the methyl hydrogen in FA). This distance is denoted as  $r_{\text{O}_{\text{FA}},\text{H}_{\text{min}}}$ .



For  $r_{\text{O}_{\text{FA}},\text{H},\text{min}} < 1.4 \text{ \AA}$ , FA is considered protonated, so  $s(r) = r_{\text{O}_{\text{FA}},\text{H},\text{min}}$ . For  $r_{\text{O}_{\text{FA}},\text{H},\text{min}} > 1.4 \text{ \AA}$ , charge separation between the solvated proton and FA becomes significant. To quantify it and thus compute  $s(r)$ , all the distances between reactive hydrogens and oxygens are first calculated. Hydrogens are then assigned to the closest oxygen, either water or FA. Any water oxygen found to be associated with three hydrogens is then indexed. All distances between FA oxygens and hydrogens associated with triply coordinated water oxygens are finally sorted.  $s(r)$  is the minimum value of these distances.

Conceptually, we aimed to describe the formation of complexes resembling Eigen or Zundel cations. A discontinuous jump of  $s(r)$  from  $\sim 1.8$  up to  $\sim 3 \text{ \AA}$  is associated with a change in the identity of the triply coordinated water oxygen and the formation of structures resembling Zundel cations  $\text{H}_5\text{O}_2^+$ . The formation of the Zundel cation with  $s(r) > 2.9 \text{ \AA}$  is here labeled as the product state B.

**2.3. RETIS.** The PyRETIS<sup>47,48</sup> library has been used to perform RETIS<sup>49</sup> simulations coupled with the *ab initio* MD external engine CP2K. In the four-water simulations, the first interface was placed at  $s(r) = 1.05 \text{ \AA}$  and the last interface at  $s(r) = 3.0 \text{ \AA}$ , thus defining the initial and the final states of the transition. Seventeen interfaces were positioned along the interval. Similarly, the six-water simulations had the first interface at  $s(r) = 1.07 \text{ \AA}$  and the last interface at  $s(r) = 3.0 \text{ \AA}$ . Thirteen interfaces were positioned along the interval.

The initial paths describing the transition from protonated to deprotonated FA along  $s(r)$  were generated by using the *kick* method available in the software, starting from the initial configurations shown in Figure 1. The “kick” approach uses a mixture of stochastic and deterministic dynamics to generate a set of initial paths. From the results, the paths that correlated with the initial generated ones were discarded. Finally, the remaining trajectories from a set of multiple independent simulations were merged together for both the four- and six-water-molecule cases.

**2.4. Selection Window.** In a trajectory, each frame can be considered as an instance in a data-representation suitable for the DT. Depending on the simulation setup, a large number of frames would generate a long list of instances with a very high correlation. Furthermore, different trajectories can be highly correlated with one another, depending on the sampling algorithm. Since generating a sufficient number of uncorrelated trajectories often requires excessive computational requirements, an approach to provide a sufficient sampling with a limited correlation is proposed here.

Frames contained in a rather restricted region in the path space can be identified *via* a selection window. By randomly picking a certain number of frames for each trajectory, within the selection window, the correlation between instances is minimized. By placing the selection window in proximity to the initial state, as in the current study, the system configurations which are correlated with the transitions can be identified prior to the transition actually occurring. The selection window location and dimension and the number of frames per trajectory to consider constitute the three hyper-parameters of our approach. In the present work, the ML algorithm has been fed with one frame per trajectory within a selection window defined by values of the order parameter  $1.1 < s(r) < 1.25 \text{ \AA}$ . The range is sufficiently narrow to consider only a few frames for each trajectory, each with a similar order parameter. The ML algorithm should, therefore, be able to determine the

most relevant feature(s) associated with the transition happening without hypothesis-bias on the main descriptor of the transition itself. This limits the correlation of the detected features with the classification of the trajectory.

**2.5. Training the DTs, Labels.** The ML problem we are posing is as follows: “what are the main features that a simulation frame has to have in order to be part of a trajectory that connects an initial state to a final state (reactive)?” and “with which probability?” The information gain (entropy) DT is a viable method for a problem with highly correlated features.<sup>50</sup>

DTs report the most important features that differentiate between reactive and unreactive paths without imposing any prior hypothesis.

Given a set of trajectories, a classification between reactive and unreactive paths is first needed. A numerical descriptor, conventionally defined as the order parameter, can quantify the progress of a given transition. If its value for a given system is within certain arbitrarily defined ranges, the system can be considered to be located in the initial or the final state. A reactive path is defined as a path starting from an initial state and ending at a product state. A non-reactive path, instead, ends at the initial state.

If the input generated by molecular simulation is composed of a single long trajectory, sub-segments will have to be fed to the ML task. In such a case, a segment starting at one state and ending in another state will be considered reactive, whereas a segment starting and ending at the same state without having previously entered another state will be unreactive. When using the input generated by path sampling, paths contained in a single ensemble should be considered (please consider refs 31 and 51 for the definition of an ensemble and further details of the path sampling methods).

**2.6. Training the DTs, Data Matrix.** Generally, all trajectory segments or trajectories for path sampling can be considered in the present analysis approach. When using path sampling, a re-weighting algorithm is adopted to consider all the generated paths. Due to the statistical weights of the different ensembles and for simplicity, we opted to consider only the trajectories included in the outermost ensemble in the path space (for the definition of an ensemble, please consider the RETIS formalism<sup>49</sup>).

From molecular simulations, an ordered data array for the positions and velocities for each atom is written for each selected time frame. While the convention facilitates post-processing and visualization procedures, it includes a bias in the data representation. Small deviations in the observation angle or on the choice of coordinate system (*e.g.*, exchanging  $x$  with  $y$  coordinates) lead to significantly different data sets while corresponding to nearly identical systems. For our work, the data thus have to be pre-processed to become invariant with respect to translations and rotations. Furthermore, the ML problem also has to be atom-index-invariant, and the sorting method also must be reversible to allow back-mapping of the features indicated by the ML to the relevant atom (or atom pairs).

In the present work, we considered atomic distances and velocities as possible features. Since the atomic velocities did not provide a significant contribution in our results, the forthcoming analysis has been based on atomic distances only.

The translation and rotation-independent requisites might be met with an atom–atom distance matrix. The atom-index-invariant approach requires, on the other hand, a more

elaborate representation. First, a reference atom, which can differ in different frames if the atoms are indistinguishable (*i.e.*, the same element in an atomistic simulation), shall be selected. Thereafter, the rows in the atom–atom distance matrix are grouped per element and sorted within each element-group based on the distance from the reference atom. For each row in the matrix, the columns are grouped and sorted following an analogous procedure. The sorting is thus based on the distance from the atom indicated by the row. The column indices can therefore indicate a different atom for each row. A “'” denotes the secondary index.

The resulting matrix reports the distance from a selected reference atom (rows) to its next neighbor (columns). A scheme of the algorithm to generate both the distance matrix and the index-invariant distance matrix is provided in the **Supporting Information**. In **Figure 2**, the two matrix

	C0	H0	O0	O1	H1
C0	0.0000	0.1109	0.1250	0.1321	0.2010
H0	0.1109	0.0000	0.2047	0.2013	0.2954
O0	0.1250	0.2047	0.0000	0.2311	0.2524
O1	0.1321	0.2013	0.2311	0.0000	0.1073
H1	0.2010	0.2954	0.2524	0.1073	0.0000

	C0'	H0'	H1'	O0'	O1'
C0	0.0000	0.1109	0.2010	0.1250	0.1321
H0	0.1109	0.0000	0.2954	0.2013	0.2047
H1	0.2010	0.0000	0.2954	0.1073	0.2524
O0	0.1250	0.2047	0.2524	0.0000	0.2311
O1	0.1321	0.1073	0.2013	0.0000	0.2311



**Figure 2.** Distance matrix (top) for a structure of FA (right). The index-invariant distance matrix (bottom) corresponds to the first distance matrix. The prime on the column atom index indicates that it depends on the row atom index.

representations are provided, as an example, for an isolated FA molecule, where we used the carbon atom (C0) as the trivial identifiable reference atom. The resulting internal coordinate representation allows an independent analysis of each entry and, thus, a suitable data structure for the ML task.

We would like to note here that a common translational- and rotational-invariant representation, the Z-matrix,<sup>52</sup> also provides an appealing internal representation of molecular structures as it scales better with the number of atoms compared to the distance matrix. However, the values of its variables (*i.e.*, distances, angles, and dihedrals) are dependent on each entry and on the atom sequence. In contrast, in our distance matrix representation, each entry is independent. Also, distances are unique, with a lower bound (0) and an upper bound (system size). These three characteristics allow for a suitable split of sample space by the DTs. Furthermore, our representation is index-invariant.

We here report the results obtained by the index-invariant distance matrix, which is the most general approach, even if more computationally demanding. It is worth noting that the index-variant distance matrix can be advantageous for its simplicity and symmetry in certain applications, for example, in the presence of atoms that do not swap order during a transition. The results for the index-variant distance matrix are presented in the **Supporting Information**.

Computationally, a DecisionTree Classifier from scikit-learn<sup>53</sup> has been fed with the index-invariant matrix, flattened

to a feature vector, using the “entropy” splitting criterion and a maximum depth of three.

**2.7. Data Matrix Notation and DT Visualization.** The atom labeling system we use identifies each atom with a character and a digit. The character corresponds to the atom type, while the digit corresponds to the position of the sorted distance list per element with respect to a reference atom, with the indexing starting at 0. The digit of the first entry in the atom–atom distance label refers to the sorted distance list with respect to the reference atom (C0). The digit of the second entry refers to the sorted distance list with respect to the first atom of the atom–atom pair. To highlight it, a prime (') has been added to the second index. As two examples, (a) O2–H5' corresponds to the distance from the third closest oxygen (O2) to the C atom to the hydrogen atom, which is 6th closest to O2. (b) H0–O0' is the distance from the H closest to the C (H0) to the oxygen closest to H0.

A symmetric distance matrix can be back-mapped to *xyz* coordinates (up to a translation and rotation) as described by Young and Householder<sup>54</sup> (and further detailed in the **Supporting Information**). The index-invariant distance matrix can be unsorted into the symmetric matrix up to an atom index difference. The approach permits the addition of dummy atoms according to the splits given by the DT, allowing a direct visualization of the analysis output (*e.g.*, via VMD<sup>55</sup>). For a convenient visualization, only the dummy atoms corresponding to the nodes along each decision path in the tree might be selected. A main decision path is chosen such that a leaf node would have the highest number of pertinent reactive paths weighted by the percentage of pertinent reactive paths:  $n_r \cdot n_r / (n_r + n_u)$ , where  $n_r$  is the number of reactive paths in that node and  $n_u$  the number of unreactive paths.

**2.8. Random Forest Decision Error Estimate.** The prediction error is simulation time-dependent and the true answer is unknown. Furthermore, due to time evolution, the distribution is not Gaussian and the noise is heteroscedastic with respect to the true value. Our implementation of the DT algorithm is not designed to make statistical predictions; instead, it focuses on identifying the most important features (regularization). To provide an estimate of the method's reliability in the feature selection, an error-estimate procedure has been thus developed.

With highly correlated data, significantly different trees can be originated depending upon the first split from minor variations of the input. It is a constitutive limitation of the approach. The relative importance of the first split can be asserted by using random forests<sup>56</sup> with a unit depth. The random forest reports the importance of all features by sampling several DTs, each one generated from a subset of features. By limiting the depth of each tree to 1, the feature importance of such random forests becomes equal to the importance of the first split only. It shall be noted that the feature that has the highest importance in the random forest plot does not necessarily represent the main split for all possible DTs.

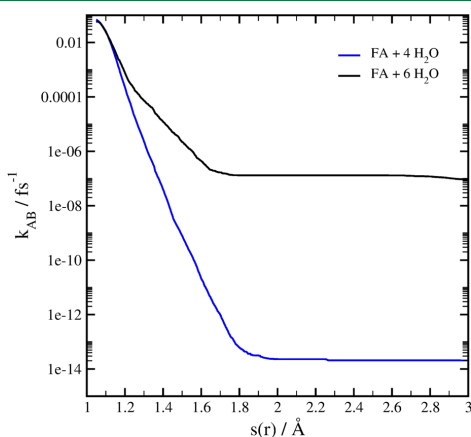
A sequence of forests of DTs has been generated with respect to the time sequence during which sampling output has been generated. Source data have been split into 10 sub-blocks and randomized within each. A random forest has then been computed for each of these subsets, generating a sort of time-dependent profile for the main splits, which allows the computation of a variance  $\sigma$  for each of the main features. The average value for each feature can then be computed by

considering the whole data set. By assuming a Gaussian distribution for each feature and by using the previously obtained variance and mean value for each feature, the relative probability of a feature importance is estimated. By comparing the probability distribution for each feature, the most relevant can be identified even if the data are highly correlated.

Computationally, a RandomForest Classifier from scikit-learn<sup>53</sup> has been fed with the index-invariant matrix, flattened to a feature vector, using the “entropy” splitting criterion and a maximum depth of one.

### 3. RESULTS AND DISCUSSION

The rate of proton transfer from FA to the water molecules has been computed *via* RETIS simulation and *ab initio* MD simulations. Figure 3 reports the rate of reaction for two



**Figure 3.** Effective rate constant  $k_{AB}$  computed using RETIS for FA clustered with four or six water molecules to reach the deprotonated state. Results are obtained from an average of several RETIS simulations from different initial conditions weighted by the respective number of RETIS cycles.

systems, where four and six water molecules surrounded FA. State A (protonated state) is defined as configurations with  $s(r) < 1.05$  Å (four waters) or with  $s(r) < 1.07$  Å (six waters). State B includes configurations with  $s(r) > 3.0$  Å for both systems.

The rate of proton transfer for the four-water-molecule case is  $\sim 2.10 \times 10^{-14}$  and  $\sim 1.01 \times 10^{-7}$   $\text{fs}^{-1}$  for six water molecules around FA ( $10^7$  times difference).

We here investigate the mechanism of reactions *via* DTs to identify the feature(s) that better correlate for each case with pathways that lead to proton transfer. The analysis might provide qualitative and quantitative descriptions of the different system features responsible for the significant difference in the reported rates.

**3.1. FA with Four Water Molecules.** The DT generated for the system with four water molecules clustered around FA is reported in Figure 4. To simplify the visualization of the main splits that lead to the highest reactive trajectories of the DT, in Figure 4, a Cartesian/*xyz* representation has been included. The atoms in blue, yellow, and green are involved in the first, second, and third splits, respectively.

The deprotonation reaction of FA appears to primarily require that the distance between O5 and H9' be smaller than 5.25 Å. The split implies that the distance between the oxygen furthest from the FA carbon (O5) and the furthest hydrogen from O5 should be within a given threshold. As H9' is the hydrogen of FA, it also implies that a certain orientation of the molecule, with respect to the water cluster, is also required. Under these conditions, the probability for the path to be reactive is 38%.

The next split along the branch with the highest probability to be reactive is the distance between O1 and H8' being smaller than 4.25 Å. The distance between one of the FA oxygens and one of the furthest hydrogen atoms should be sufficiently small. This implies that the oxygen of FA should be located around the center of the cluster and that a sort of ordered disposition of the water molecules in the cluster is required. When this condition is satisfied, the probability for a path to be reactive reaches 63%.

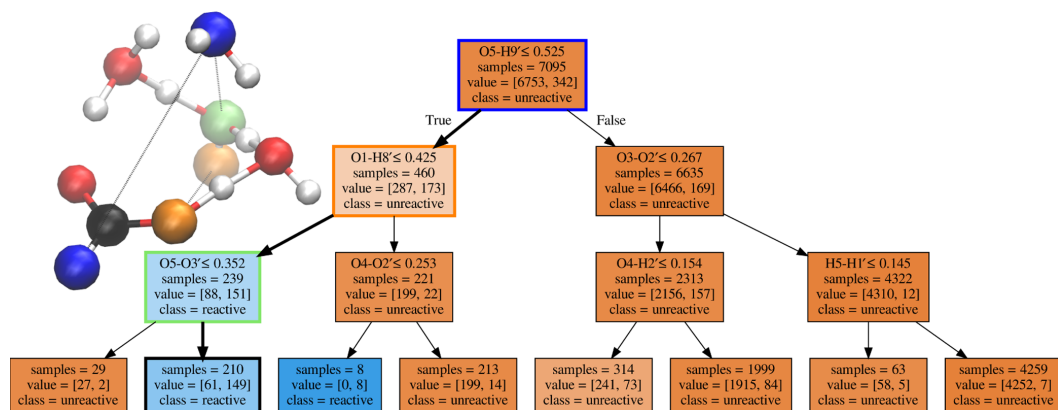
Continuing along the branch with the highest reactive probability, the distance between O5 and O3' being bigger than 3.52 Å represents the last split here considered. This corresponds to the relative position of two water molecules being two hydrogen bonds apart. We interpret the requisite as the suitable distance to establish hydrogen bonding between the atoms.

When all three of these requirements are met, the probability of a path being reactive is 71%. By comparing the number of reactive paths versus the number of unreactive paths in the final splits of the DT, it can be concluded that the indicated reactive path is clearly predominant. A similar conclusion can be reached by observing the first splits reported in Figure 5. The figure that reports the results obtained from a random forest of DTs of depth 1 indicates that the relative probability for the first split to be the most important feature is 39%. The subsequent distances reported by the random forest analysis have a constantly decaying relevance. The first five main splits reported by Figure 5 are correlated and taken together indicate that the water cluster has to be sufficiently compact and FA has to be oriented such that its oxygen molecules are in close contact with the surrounding water molecules.

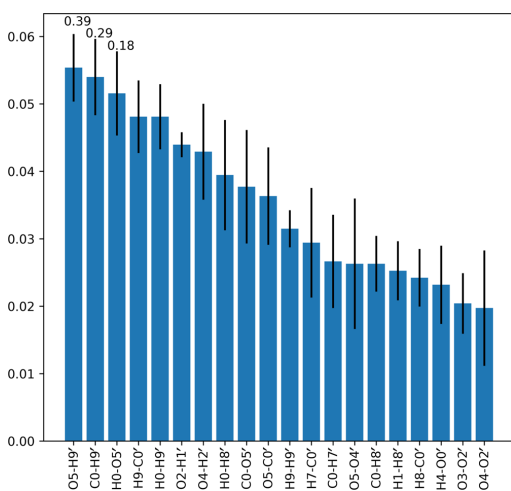
**3.2. FA with Six Water Molecules.** For the clusters with six water molecules around FA, in Figure 6, we report the generated DT. As we did with the four-water-molecule case, a visualization of the main splits of the DT that led to the highest reactive trajectories is also included in Figure 6. The atoms in blue, yellow, and green are involved in the first, second, and third splits, respectively.

The deprotonation reaction of FA in the six-water-molecule cluster requires the distance between O6 and H8' to be smaller than 3.55 Å. This split involves two water molecules in the proximity of FA that need to be within a certain distance. By inspecting the frame reported in Figure 6, the requirement seems to indicate a certain orientation of one of the water molecules associated with another water molecule in proximity to the FA oxygen. Under such conditions, the probability for the path to be reactive is 32%.

The next split, along the branch with the highest probability to be reactive, is the distance between H9 and H11' being smaller than 4.27 Å. The distance between these two atoms can also be interpreted as a combination of molecular orientation of the water molecules in the surroundings of FA



**Figure 4.** DT for the system with four water molecules around the FA molecule based on the index-invariant distance matrix. Each text box represents one node and reports (1) the inequality which splits the data going out of the node, (2) the number of samples entering the node, (3) the number of (unreactive, reactive) samples entering the node, and (4) the majority class of the node (*i.e.*, whether most of the data entering represent unreactive or reactive trajectories). At each split, the “True” branch is on the left and the “False” branch is on the right. The color indicates the ratio between unreactive (brown) and reactive (blue) samples included in a node. Wider arrows have been used to link the sequence of data splittings determined to be the most important in the analysis. In the top left corner, a 3D representation of the system is provided. The atoms highlighted in blue, yellow, and green correspond to the atoms involved in the first, second, and third split of the most important decision branch, respectively. In red, white, and black are the oxygen, hydrogen, and carbon atoms if not already highlighted as the most important decision branch.



**Figure 5.** Importance approximations of the possible first split inequalities generated from a random forest with a depth of one for the four-water system. The bars represent the feature importance of a random forest, with the error bar calculated with a block-error average based on the generated trajectories. The probability that a split is truly the most important split is shown above the bars for the three most probable first splits.

and the water cluster size. The probability of a reactive path reaches 53% when both these conditions occur.

Still along the branch with the highest probability to be reactive, the distance between O2 and O2' being bigger than 2.57 Å represents the last split here considered. This indicates that the closest water oxygen to FA (O2) should be close enough to its second closest oxygen atom to promote the

formation of a hydrogen bond network. When all three requirements are met, the probability for a path to be reactive is 72%.

By comparing the number of reactive paths versus the number of unreactive paths in the final splits of the DT, it can be concluded that the indicated reactive path is clearly favorable, but that other significant paths also exist. The conclusion is also supported by the random forest of DTs with a single split. Before proposing an interpretation, it is worth the reminder here that the random forest reports unconditional entries, while the DT splits depend on the first split. Figure 7 indicates the probability that the first split is the most important feature is 34%, but the second split has a comparable relevance: O2–O2' (28%). It confirms that while a predominant pathway for the reaction has been sampled, different main pathways can co-exist.

As reported in Figure 3, the number of water molecules in the cluster has a significant effect on the rate of the proton transfer reaction. From the comparison of the previously discussed Figures 4 and 6, we note that the distance between a FA oxygen and one of the furthest water hydrogens being below some distance is the predominant characteristic for a trajectory to be reactive. In other words, both clusters have to be sufficiently compact in order to promote the reaction. In the four-water-molecule case, the orientation of FA with respect to the water cluster is the most important feature, while for the six-water-molecule case, the water structure around FA appears to be the predominant feature.

A second main difference between the four- and six-water cases is the possible pathways for the reaction to occur. The smaller system has only one predominant reactive path, while for the six-water-molecule cluster, multiple paths appear to co-exist, contributing to the final reaction rate. Physically, if the system is sufficiently large, different configurations can lead to the proton transfer reaction, consistent with the observation that the overall rate is much higher.

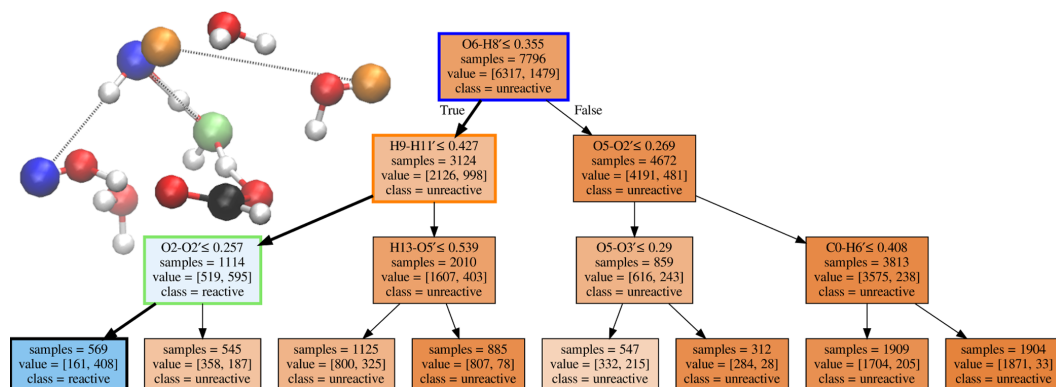


Figure 6. DT for the system with six water molecules around the FA molecule. For details, see the caption for Figure 4.

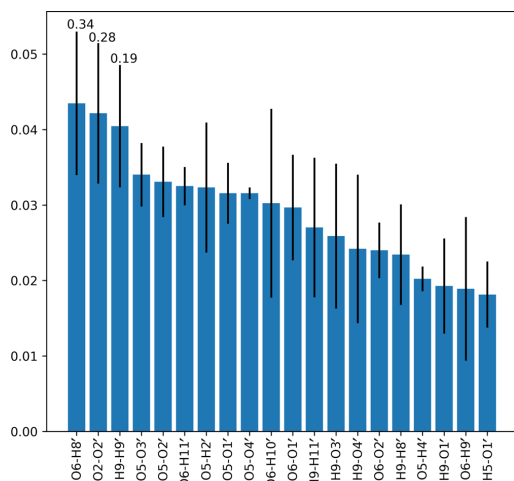


Figure 7. Importance approximations of the first split question from a random forest with a depth of one for the six-water system. For details, see the caption for Figure 5.

**3.3. Computational Cost, Scaling, Method Transferability, and Limitations.** The computational cost of our method is negligible in comparison to the cost of generating pathways *via* MD. It is worth stressing here that our approach does not aim to replace the generation of trajectories but to improve the description of their characteristics.

The time required to train the DT scales as a function of the number of frames and features. The required time scales as  $O(N \log N)$ , where  $N$  is the number of frames, and linearly with the number of features. For the six-water-molecule system, with 529 features and 11,418 frames, the training time required was 1.7 s on a laptop (Dell XPS-15 with an Intel i7-8750H, 6 cores, and 12 threads). The number of features in the proposed representation scales as  $O(M^2)$ , with  $M$  being the number of atoms, which could be an issue in both memory and computational time for relatively large systems. However, the training of the DT can be efficiently parallelized over the

number of features, with only one communication step per split of the DT.

The trained DTs are generally not transferable to other systems for predictions. However, the training of DTs is efficient and the training input of the DT is a feature vector that can be generated directly for any atomistic system as long as the positions and elements of the atoms in a frame and the classification of the trajectory are known. The feature vector can also be extended with user-defined features. Therefore, our described data representation and training/analysis approach can be directly applied to other atomistic simulations.

One main limitation of the presented analysis method (as with any ML/data-driven method) is the effect of “garbage in, garbage out”. We aim to identify the most relevant features for a transition in a simulation. When the configuration data (the proposed feature space) do not properly correlate with the system dynamics (in the presence of underlying potential energy bias as in meta-dynamics simulation<sup>57</sup>) or when frames are more correlated to a source sub-set (e.g., forward-flux-sampling<sup>58</sup>), the DTs still identify the most important feature for the classification, although the feature may be different from unbiased simulation.

## 4. CONCLUSIONS

A data-driven method to systematically compute reaction pathways has been presented. The conventional Cartesian/*xyz* data representation employed in molecular simulations is converted into an index-invariant distance matrix representation, which is also translation- and rotation-invariant. Thereafter, an approach which limits the correlation between elements in the source data (MD trajectories) has been proposed in conjunction with a rare event simulation framework. The data have then been fed to a supervised classifier method, the DT.

To simplify the interpretation of the classifier, a back mapping procedure from the index-invariant matrix has been adopted to emphasize the atoms involved, with each split identified by the DT. Generation of a random forest of DTs, in combination with block averaging, provided an error range for the first split of the DT.

We thus presented a data-driven approach to gain insight into a chemical reaction. The method has been designed such that it is readily applicable to other simulation strategies and



types of transitions. The strength of the present approach is that it allows the use of complex collective variables which may be discontinuous and the estimation of the probability of their occurrence in a transition path. The descriptors to elucidate transition mechanisms might be directly implemented in a prediction method.<sup>37</sup>

The method adopted an index-invariant distance matrix providing a data-driven insight into the reaction pathways. The data-driven identification aims to identify interpretable pathways in a system composed of indistinguishable molecules. Applications to more inhomogeneous systems would be straightforward, especially if only a portion of the system atoms are of interest. The latter case would combine human intuition with a data-driven approach, which would, possibly, provide a better insight into the reaction if, and only if, the introduced bias is correct. Our method can be further expanded by considering a higher number of descriptors alongside the distance matrix. Velocities, angles between molecules, coarse-graining procedures, or a mix of user-defined functions<sup>59</sup> could be fed into the DT and subsequent analysis.

To demonstrate the capabilities of the developed method, a mechanistic description of the proton transfer reaction in small aqueous clusters of FA has been provided. The reaction has been simulated *via* rare event simulation (replica exchange transition interface sampling<sup>31</sup>) and its rate quantified for two water clusters, one composed of four and one of six water molecules surrounding an FA molecule.

The reaction rate we computed is strongly influenced by the number of water molecules present. Mechanistically, the four- and six-water proton transfer reaction requires the water cluster to be sufficiently compact. The four-water-molecule system requires a certain orientation of the FA molecules and of the water molecules in its proximity. For the six-water-molecule case, a certain orientation of the outer water molecules appears to be more significant in describing the reaction path. Furthermore, the four-water cluster system indicated only one predominant pathway for the reaction to occur, while in the six-water-molecule cluster, several pathways have been identified, contributing to the higher reaction rate in this system.

## ■ ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jctc.1c00458>.

It contains the algorithm to generate the atom index-invariant data representation, the theoretical background for mapping the symmetric distance matrix back to XYZ coordinates, and the analysis based on the index-variant distance matrix for FA with four and six water molecules (PDF)

## ■ AUTHOR INFORMATION

### Corresponding Author

Sander Roet – Department of Chemistry, Norwegian University of Science and Technology, 7491 Trondheim, Norway; [orcid.org/0000-0003-0732-545X](https://orcid.org/0000-0003-0732-545X);  
Email: [sander.roet@ntnu.no](mailto:sander.roet@ntnu.no)

## Authors

Christopher D. Daub – Department of Chemistry, University of Helsinki, FI-00014 Helsinki, Finland; [orcid.org/0000-0002-4290-9058](https://orcid.org/0000-0002-4290-9058)

Enrico Riccardi – Department of Informatics, UiO, 0373 Oslo, Norway; [orcid.org/0000-0003-1890-7113](https://orcid.org/0000-0003-1890-7113)

Complete contact information is available at:  
<https://pubs.acs.org/10.1021/acs.jctc.1c00458>

## Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

Part of the work has been performed under the Project HPC-EUROPA3 (INFRAIA-2016-1-730897) with the support of the EC Research Innovation Action under the H2020 Programme. In particular, the authors gratefully acknowledge the support of the Department of Chemistry at the University of Helsinki and the computer resources and technical support provided by CSC. C.D.D. acknowledges funding by the Academy of Finland (grant number 294752) and by the Jane and Aatos Erkko Foundation. The authors also thank Prof. Geir Kjetil Ferkingstad Sandve for his preliminary review of their work.

## ■ REFERENCES

- (1) Finlayson-Pitts, B. J.; Pitts, J. N. *Chemistry of the Upper and Lower Atmosphere: Theory, Experiments, and Applications*; Academic Press: San Diego, CA, USA, 2000.
- (2) Millet, D. B.; Baasandorj, M.; Farmer, D. K.; Thornton, J. A.; Baumann, K.; Brophy, P.; Chaliyakunnel, S.; de Gouw, J. A.; Graus, M.; Hu, L.; Koss, A.; Lee, B. H.; Lopez-Hilfiker, F. D.; Neuman, J. A.; Paulot, F.; Peischl, J.; Pollack, I. B.; Ryerson, T. B.; Warneke, C.; Williams, B. J.; Xu, J. A large and ubiquitous source of atmospheric formic acid. *Atmos. Chem. Phys.* **2015**, *15*, 6283–6304.
- (3) Chaliyakunnel, S.; Millet, D. B.; Wells, K. C.; Cady-Pereira, K. E.; Shephard, M. W. A Large Underestimate of Formic Acid from Tropical Fires: Constraints from Space-Borne Measurements. *Environ. Sci. Technol.* **2016**, *50*, 5631–5640.
- (4) Kangas, P.; Hänninen, V.; Halonen, L. An Ab Initio Molecular Dynamics Study of the Hydrolysis Reaction of Sulfur Trioxide Catalyzed by a Formic Acid or Water Molecule. *J. Phys. Chem. A* **2020**, *124*, 1922–1928.
- (5) Daub, C. D.; Riccardi, E.; Hänninen, V.; Halonen, L. Path sampling for atmospheric reactions: formic acid catalysed conversion of SO<sub>3</sub> + H<sub>2</sub>O to H<sub>2</sub>SO<sub>4</sub>. *PeerJ Phys. Chem.* **2020**, *2*, No. e7.
- (6) von Grothuss, C. J. D. Sur la décomposition de l'eau et des corps qu'elle tient en dissolution à l'aide de l'électricité galvanique. *Ann. Chim.* **1806**, *58*, 54–73.
- (7) Agmon, N. The Grothuss mechanism. *Chem. Phys. Lett.* **1995**, *244*, 456–462.
- (8) Marx, D. Proton transfer 200 years after von Grothuss: Insights from ab initio simulations. *ChemPhysChem* **2006**, *7*, 1848–1870.
- (9) Agmon, N.; Bakker, H. J.; Campen, R. K.; Henschman, R. H.; Pohl, P.; Roke, S.; Thämer, M.; Hassanal, A. Protons and Hydroxide Ions in Aqueous Systems. *Chem. Rev.* **2016**, *116*, 7642–7672.
- (10) Knight, C.; Voth, G. A. The Curious Case of the Hydrated Proton. *Acc. Chem. Res.* **2012**, *45*, 101–109.
- (11) Tse, Y.-L. S.; Knight, C.; Voth, G. A. An analysis of hydrated proton diffusion in ab initio molecular dynamics. *J. Chem. Phys.* **2015**, *142*, 014104.
- (12) Li, C.; Swanson, J. M. J. Understanding and Tracking the Excess Proton in Ab initio Simulations; Insights from IR Spectra. *J. Phys. Chem. B* **2020**, *124*, 5696–5708.
- (13) Li, C.; Voth, G. ChemRxiv. **2021**, Chemrxiv-2021-qkzn7.

- (14) Lee, J.-G.; Ascuitto, E.; Babin, V.; Sagui, C.; Darden, T.; Roland, C. Deprotonation of Solvated Formic Acid: Car-Parrinello and Metadynamics Simulations. *J. Phys. Chem. B* **2006**, *110*, 2325–2331.
- (15) Galib, M.; Hanna, G. Mechanistic Insights into the Dissociation and Decomposition of Carbonic Acid in Water via the Hydroxide Route: An Ab Initio Metadynamics Study. *J. Phys. Chem. B* **2011**, *115*, 15024–15035.
- (16) Tummanapelli, A. K.; Vasudevan, S. Estimating successive  $pK_a$  values of polyprotic acids from ab initio molecular dynamics using metadynamics: the dissociation of phthalic acid and its isomers. *Phys. Chem. Chem. Phys.* **2015**, *17*, 6383–6388.
- (17) Grifoni, E.; Piccini, G.; Parrinello, M. Microscopic description of acid–base equilibrium. *Proc. Natl. Acad. Sci. U.S.A.* **2019**, *116*, 4054–4057.
- (18) Daub, C. D.; Halonen, L. Ab Initio Molecular Dynamics Simulations of the Influence of Lithium Bromide Salt on the Deprotonation of Formic Acid in Aqueous Solution. *J. Phys. Chem. B* **2019**, *123*, 6823–6829.
- (19) Hammerich, A. D.; Buch, V. Ab Initio Molecular Dynamics Simulations of the Liquid/Vapor Interface of Sulfuric Acid Solutions. *J. Phys. Chem. A* **2012**, *116*, 5637–5652.
- (20) Galib, M.; Hanna, G. Molecular dynamics simulations predict an accelerated dissociation of  $H_2CO_3$  at the air–water interface. *Phys. Chem. Chem. Phys.* **2014**, *16*, 25573–25582.
- (21) Gerber, R. B.; Varner, M. E.; Hammerich, A. D.; Riikonen, S.; Murdachaew, G.; Shemesh, D.; Finlayson-Pitts, B. J. Computational Studies of Atmospherically-Relevant Chemical Reactions in Water Clusters and on Liquid Water and Ice Surfaces. *Acc. Chem. Res.* **2015**, *48*, 399–406.
- (22) Murdachaew, G.; Nathanson, G. M.; Gerber, R.B.; Halonen, L. Deprotonation of formic acid in collisions with a liquid water surface studied by molecular dynamics and metadynamics simulations. *Phys. Chem. Chem. Phys.* **2016**, *18*, 29756–29770.
- (23) Partanen, L.; Murdachaew, G.; Gerber, R. B.; Halonen, L. Temperature and collision energy effects on dissociation of hydrochloric acid on water surfaces. *Phys. Chem. Chem. Phys.* **2016**, *18*, 13432–13442.
- (24) Leopold, K. R. Hydrated Acid Clusters. *Annu. Rev. Phys. Chem.* **2011**, *62*, 327–349.
- (25) Forbert, H.; Masia, M.; Kaczmarek-Kedziera, A.; Nair, N. N.; Marx, D. Aggregation-Induced Chemical Reactions: Acid Dissociation in Growing Water Clusters. *J. Am. Chem. Soc.* **2011**, *133*, 4062–4072.
- (26) Chung, Y. K.; Kim, S. K. Dissociation of sulfur oxoacids by two water molecules studied using ab initio and density functional theory calculations. *Int. J. Quantum Chem.* **2017**, *117*, No. e25419.
- (27) Lengyel, J.; Pysanenko, A.; Fárník, M. Electron-induced chemistry in microhydrated sulfuric acid clusters. *Atmos. Chem. Phys.* **2017**, *17*, 14171–14180.
- (28) Gutberlet, A.; Schwaab, G.; Birer, O.; Masia, M.; Kaczmarek, A.; Forbert, H.; Havenith, M.; Marx, D. Aggregation-Induced Dissociation of  $HCl(H_2O)_4$  Below 1 K: The Smallest Droplet of Acid. *Science* **2009**, *324*, 1545.
- (29) Maity, D. K. How Much Water Is Needed To Ionize Formic Acid? *J. Phys. Chem. A* **2013**, *117*, 8660–8670.
- (30) Elena, A. M.; Meloni, S.; Ciccotti, G. Equilibrium and Rate Constants, and Reaction Mechanism of the HF Dissociation in the  $HF(H_2O)_7$  Cluster by ab Initio Rare Event Simulations. *J. Phys. Chem. A* **2013**, *117*, 13039–13050.
- (31) van Erp, T. S. Reaction Rate Calculation by Parallel Path Swapping. *Phys. Rev. Lett.* **2007**, *98*, 268301.
- (32) Riccardi, E.; Dahlen, O.; van Erp, T. S. Fast Decorrelating Monte Carlo Moves for Efficient Path Sampling. *J. Phys. Chem. Lett.* **2017**, *8*, 4456–4460.
- (33) Riccardi, E.; Van Mastbergen, E. C.; Navarre, W. W.; Vreede, J. Predicting the mechanism and rate of H-NS binding to AT-rich DNA. *PLoS Comput. Biol.* **2019**, *15*, No. e1006845.
- (34) Riccardi, E.; Krämer, A.; van Erp, T. S.; Ghysels, A. Permeation Rates of Oxygen through a Lipid Bilayer Using Replica Exchange Transition Interface Sampling. *J. Phys. Chem. B* **2020**, *125*, 193–201.
- (35) Moqadam, M.; Riccardi, E.; Trinh, T. T.; Lervik, A.; van Erp, T. S. Rare event simulations reveal subtle key steps in aqueous silicate condensation. *Phys. Chem. Chem. Phys.* **2017**, *19*, 13361–13371.
- (36) Moqadam, M.; Lervik, A.; Riccardi, E.; Venkatraman, V.; Alsborg, B. K.; van Erp, T. S. Local initiation conditions for water autoionization. *Proc. Natl. Acad. Sci. U.S.A.* **2018**, *115*, E4569–E4576.
- (37) van Erp, T. S.; Moqadam, M.; Riccardi, E.; Lervik, A. Analyzing complex reaction mechanisms using path sampling. *J. Chem. Theory Comput.* **2016**, *12*, 5398–5410.
- (38) Hooft, F.; Pérez de Alba Ortiz, A.; Ensing, B. Discovering Collective Variables of Molecular Transitions via Genetic Algorithms and Neural Networks. *J. Chem. Theory Comput.* **2021**, *17*, 2294–2306.
- (39) Chen, W.; Ferguson, A. L. Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration. *J. Comput. Chem.* **2018**, *39*, 2079–2102.
- (40) Schöberl, M.; Zabarás, N.; Koutsourelakis, P.-S. Predictive collective variable discovery with deep Bayesian models. *J. Chem. Phys.* **2019**, *150*, 024109.
- (41) Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *J. Chem. Phys.* **2018**, *149*, 072301.
- (42) Jung, H.; Covino, R.; Hummer, G. Artificial intelligence assists discovery of reaction coordinates and mechanisms from molecular dynamics simulations. **2019**, arXiv preprint arXiv:1901.04595.
- (43) Rossi, K.; Jurásková, V.; Wischert, R.; Garel, L.; Corminboeuf, C.; Ceriotti, M. Simulating solvation and acidity in complex mixtures with first-principles accuracy: the case of  $CH_3SO_3H$  and  $H_2O_2$  in phenol. *J. Chem. Theory Comput.* **2020**, *16*, 5139–5149.
- (44) Swain, P. H.; Hauska, H. The decision tree classifier: Design and potential. *IEEE Trans. Geosci. Electron.* **1977**, *15*, 142–147.
- (45) VandeVondele, J.; Krack, M.; Mohamed, F.; Parrinello, M.; Chassaing, T.; Hutter, J. QUICKSTEP: Fast and accurate density functional calculations using a mixed Gaussian and plane waves approach. *Comput. Phys. Commun.* **2005**, *167*, 103–128.
- (46) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *J. Chem. Phys.* **2010**, *132*, 154104.
- (47) Lervik, A.; Riccardi, E.; van Erp, T. S. PyRETIS: A well-done, medium-sized python library for rare events. *J. Comput. Chem.* **2017**, *38*, 2439–2451.
- (48) Riccardi, E.; Lervik, A.; Roet, S.; Aaroen, O.; Erp, T. S. PyRETIS 2: an improbability drive for rare events. *J. Comput. Chem.* **2020**, *41*, 370–377.
- (49) van Erp, T. S.; Moroni, D.; Bolhuis, P. G. A novel path sampling method for the calculation of rate constants. *J. Chem. Phys.* **2003**, *118*, 7762–7774.
- (50) Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. *Classification and Regression Trees*; Routledge, 1984.
- (51) Cabriolu, R.; Skjelbred Refsnes, K. M.; Bolhuis, P. G.; van Erp, T. S. Foundations and latest advances in replica exchange transition interface sampling. *J. Chem. Phys.* **2017**, *147*, 152722.
- (52) Baker, J.; Hehre, W. J. Geometry optimization in cartesian coordinates: The end of the Z-matrix? *J. Comput. Chem.* **1991**, *12*, 606–610.
- (53) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
- (54) Young, G.; Householder, A. S. Discussion of a set of points in terms of their mutual distances. *Psychometrika* **1938**, *3*, 19–22.
- (55) Humphrey, W.; Dalke, A.; Schulten, K. VMD – Visual Molecular Dynamics. *J. Mol. Graph.* **1996**, *14*, 33–38.
- (56) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32.

(57) Sutto, L.; Marsili, S.; Gervasio, F. L. New advances in metadynamics. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2012**, *2*, 771–779.

(58) Allen, R. J.; Valeriani, C.; Rein ten Wolde, P. Forward flux sampling for rare event simulations. *J. Phys.: Condens. Matter* **2009**, *21*, 463102.

(59) Aarøen, O.; Kiær, H.; Riccardi, E. PyVisA: Visualization and Analysis of path sampling trajectories. *J. Comput. Chem.* **2021**, *42*, 435–446.



# SI

## Chemistrees:

### data driven identification of reaction pathways via machine learning

Sander Roet,<sup>\*,†</sup> Christopher D. Daub,<sup>‡</sup> and Enrico Riccardi<sup>¶</sup>

<sup>†</sup>*Department of Chemistry, Norwegian University of Science and Technology, Trondheim,  
Norway*

<sup>‡</sup>*Department of Chemistry, University of Helsinki, P.O. Box 55, FI-00014, Helsinki,  
Finland*

<sup>¶</sup>*Department of Informatics, UiO, Gaustadalléen 23B, 0373 Oslo, Norway*

E-mail: sander.roet@ntnu.no

#### Making the data atom-index invariant

In a condensed system, the atoms might swap order during a transition. An index invariant data representation is, therefore, not clearly advantageous since it requires extra processing and it cannot use a symmetric representation. Yet, as it constitutes the most general case, it has been considered in addition to the index-variant representation we use in the main text.

As shown in Figure 1, by choosing a "anchor" atom, which may be different for each frame, the data representation can become invariant with respect to translation, rotation, and changes in the atomic indices. For the FA in the water system the carbon atom is the trivial identifiable anchor. The rest of the atoms are sorted based on the atom type and the distance

from the anchor, as illustrated in Figure 2. The resulting data representation is atom-index invariant. Due to the statistical fluctuations in the atom positions, this procedure requires more data to achieve convergence of a ML algorithm, and imposes further requirements on the interpretation of the resulting random tree.

In our simulation, atoms of different indices do not swap places during transitions (They do, eventually, in the stable states). Therefore, we also reproduced our analysis with the relatively simple distance matrix.

Figure 2 reports a simplified algorithm to generate the distance matrix, while Figure 4 presents the algorithm to generate the index invariant distance matrix.

	C0	H0	O0	O1	H1
C0	0.0000	0.1109	0.1250	0.1321	0.2010
H0	0.1109	0.0000	0.2047	0.2013	0.2954
O0	0.1250	0.2047	0.0000	0.2311	0.2524
O1	0.1321	0.2013	0.2311	0.0000	0.1073
H1	0.2010	0.2954	0.2524	0.1073	0.0000

	C0	H0	O0	O1	H1
C0	0.0000	0.2010	0.1250	0.1321	0.1109
H0	0.2010	0.0000	0.2524	0.1073	0.2954
O0	0.1250	0.2524	0.0000	0.2311	0.2047
O1	0.1321	0.1073	0.2311	0.0000	0.2013
H1	0.1109	0.2954	0.2047	0.2013	0.0000

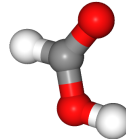


Figure 1: Two distance matrices (left) for an equivalent structure of formic acid (right). The difference between the two matrices is that H0 and H1 swapped places, or indices. This leads to a structure that has identical physics, but not an identical data representation. The distance matrix is therefore not a good representation to train our algorithms on during simulations where atom indices might change over time.

## Back mapping the symmetric distance matrix to *xyz*

We adopted the procedure first suggested by Young and Householder<sup>1</sup>. If our distance matrix for a single frame is  $D_{ij}$ , we can construct the following mapping  $M_{ij} = \frac{D_{1j}^2 + D_{i1}^2 - D_{ij}^2}{2}$ , where

	C0	H0	H1	O0	O1
C0	0.0000	0.1109	0.2010	0.1250	0.1321
H0	0.1109	0.0000	0.2954	0.2013	0.2047
H1	0.2010	0.0000	0.2954	0.1073	0.2524
O0	0.1250	0.2047	0.2524	0.0000	0.2311
O1	0.1321	0.1073	0.2013	0.0000	0.2311

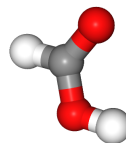


Figure 2: The index invariant distance matrix. It is created by first choosing an anchor point, in the present case, C0. Then the rows are grouped per element and sorted based on the distance from the anchor atom for each element. This representation is translationally, rotationally and atom-index invariant, making it a suitable general data representation to train ML algorithms on, including systems where atom indices might change.

---

**Algorithm 2** Algorithm to group the symmetric distance matrix based on elements. Inputs: 'mat'=a symmetric distance matrix, 'Elements'=a list of all elements in the system. Outputs: 'out'=a symmetric distance matrix where all atoms of the same element are grouped together.

---

```

1: mat = distance_matrix
2: out = output_matrix
3: indices = List()
4: for element in Elements do                                     ▷ group indices per element
5:     for atom in atoms do
6:         if atom.element==element then
7:             indices.append(atom.index)
8:         end if
9:     end for
10: end for
11: row_out, col_out= 0,0
12: for row_i in indices do                                       ▷ Group elements together
13:     for col_i in indices do
14:         out[row_out][col_out] = mat[row_i][col_i]
15:         col_out += 1
16:     end for
17:     row_out += 1
18: end for

```

---

---

**Algorithm 4** Algorithm to make the element grouped distance matrix atom-index invariant. Inputs: 'mat'=a symmetric distance matrix that has been grouped per element, 'Elements'=a list of all elements in the system, 'anchor\_idx'=the row index which is the basis for the order of the rows in the output matrix. Outputs: 'out'=a sorted index-invariant distance matrix.

---

```
1: mat = grouped_distance_matrix           ▷ Assume mat is grouped per element
2: out = output_matrix
3: elem_length = List(count(atoms of element  $e$ ) for  $e \in$  Elements)
4: out_order = List()
5: anchor_idx = 0                           ▷ Set anchor_atom to row 0
6: anchor_row = mat[anchor_idx]
7: i = 0
8: for j in elem_length do                 ▷ Figure out the output row-order.
9:   out_order.append(argsort(anchor_row[i:i+j]))
10:  i += j
11: end for
12: row_out = 0
13: for row_idx in out_order do
14:   row = mat[row_idx]
15:   i = 0
16:   for j in elem_length do
17:     out[row_out][i:i+j] = sort(row[i:i+j])
18:     i += j
19:   end for
20:   row_out += 1
21: end for
```

---

$D_{1j}$  is the  $j$ -th element of the first row of the distance matrix, and  $D_{i1}$  is the  $i$ -th element of the first column.

The eigenvalue decomposition on  $M$ :  $M = USU^T$  allows the calculation of the matrix  $X = U\sqrt{S}$ . Only  $N$  of the eigenvalues ( $S$ ) are non-zero for a system that can be embedded in  $N$  dimensional space, and distances are generated from a 3-dimensional space. Thus, the first 3 columns of  $X$  corresponds to the  $x$ ,  $y$ , and  $z$  coordinates for each row or column in the original distance matrix, up to a translation or rotation.

## Analysis based on the index variant distance-matrix

As the result obtained by the index invariant data matrix has been included in the main paper, only the results obtained from the index variant case have been included here.

### Formic Acid with 4 water molecules

The frames identified by the selection windows have been transformed into the distance matrix representation (translational and rotational invariant) and fed to the machine learning algorithm to generate the forthcoming analysis. The only hyper-parameters are the location and size of the selection window.

Our analysis generated the decision tree reported in Figure 3A. To simplify the visualization of the main splits that lead to the highest reactive trajectories of the decision tree, in Figure 3B we report an  $xyz$  representation. The atoms in blue, yellow, and green are involved in the first, second, and third split, respectively.

The deprotonation reaction of FA appears to primarily require the distance between O0 and O3 to be smaller than 2.56 Å. In other words, the first water molecule that accepts the proton from FA has to be sufficiently close. When this condition occurs, the probability to obtain a reactive path rises from 5% to 12%.

From Figure 4 (bottom) the probability that this is the most important feature is 3%.

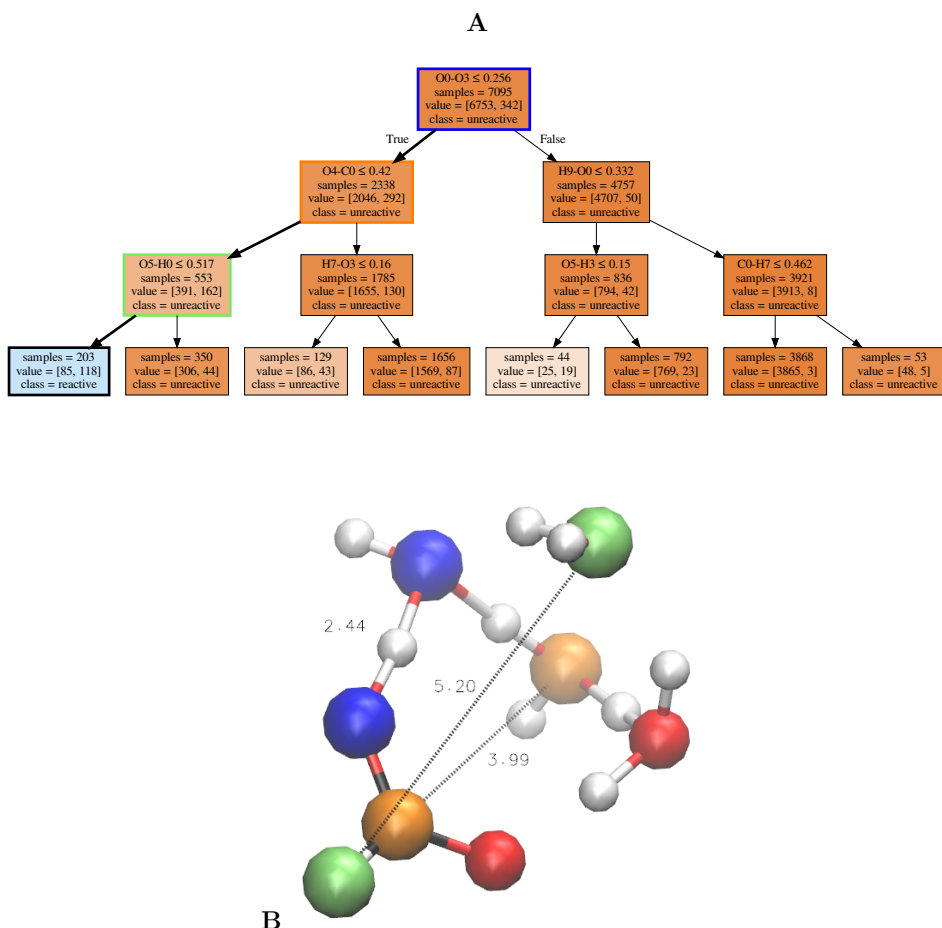


Figure 3: Decision tree for the system with 4 water molecules around the formic acid molecule based on the index-invariant distance matrix. Three splits divide the input, each square reports 1) the question to split the data, 2) the number of samples going into the node, 3) the number of [unreactive, reactive] samples going into the node and 4) the majority class of the node. At each split, the True branch is on the left, False on the right. The color indicates the ratio between unreactive (brown) and reactive (blue) samples included in a node. Wider arrows have been used to link the decision three split with the atoms involved. In panel B, a 3D representation of the system is provided. The atoms highlighted in blue, yellow, and green correspond to the atoms involved in the first, second, and third split of the decision tree, respectively. In red are the oxygen and in white the hydrogen atoms not indicated by the decision tree reported in the panel B.

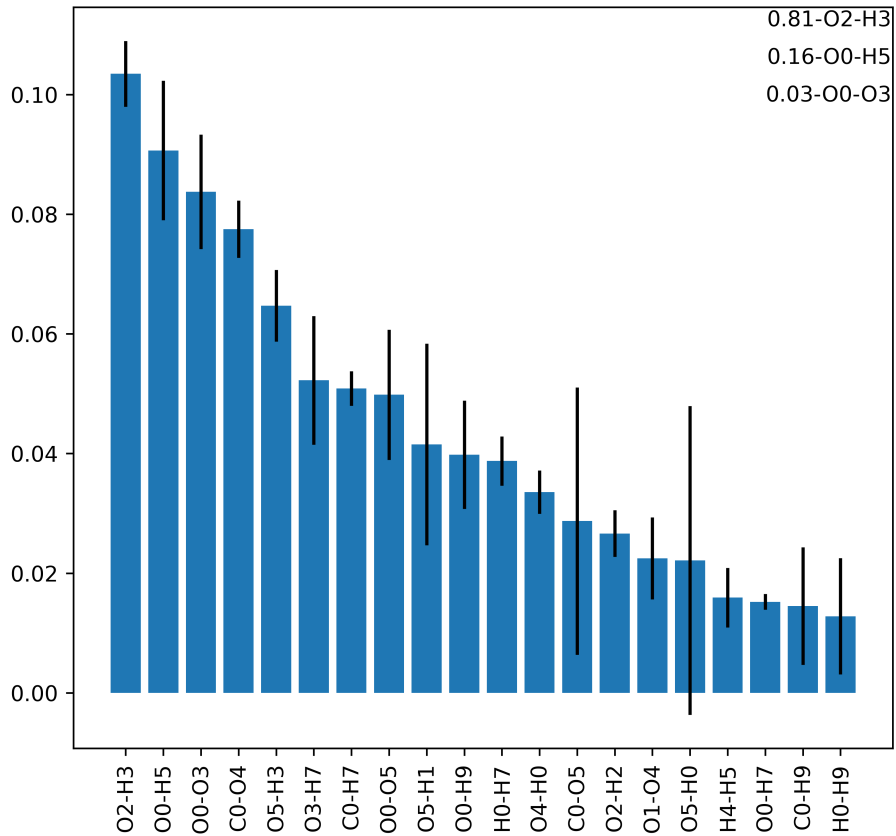


Figure 4: The importance approximations of the first split question from a random forest with depth of one for the four water case system. The bars represent the feature importance of a Random Forest, with the error bar calculated with a block-error average based on the generated trajectories.

Another feature has a probability of 16%. This is the distance between O0 and H5, the hydrogen that is connected to O3. Therefore, this condition is actually equivalent.

The next split, along the branch that leads to the highest reactive probability, is the requirement that the distance between C0 and O4 is smaller than 4.20 Å. Qualitatively, this means a water molecule has to be closer than two hydrogen bonds away from FA. If this second requirement is also satisfied, the probability of a reactive path is of 30%.

Still along the branch towards the highest reactive probability, the distance between O5 and H0 being smaller than 5.17 Å represents the last split here considered. This indicates that a second water molecule has to be within two hydrogen bonds away. When this additional requirement is satisfied, the probability of a reactive path reaches 58%.

## Formic Acid with 6 water molecules

Figure 5 reports the analysis results obtained from the simulation of the proton transfer reaction for formic acid in a 6 water molecule cluster. In the figure, the decision tree and the *xyz* structure which highlights the most probable splits which determine a reactive path are included.

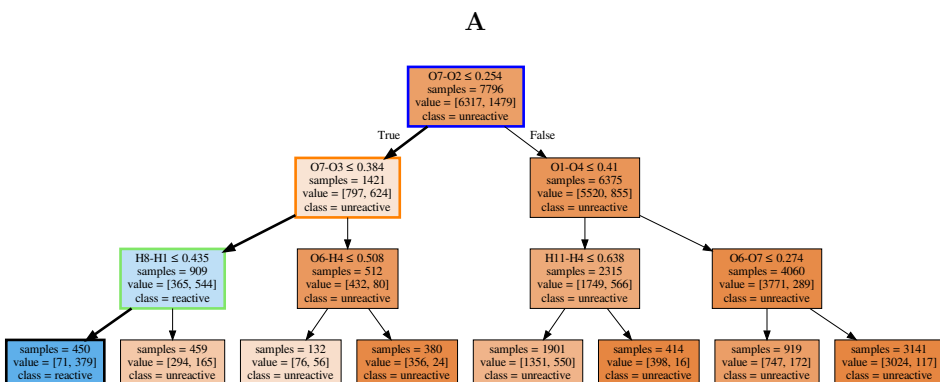
In this system the deprotonation reaction of FA appears to primarily require the distance between O7 and O2 to be smaller than 2.54 Å. The first water molecule that accepts the proton from FA has to be sufficiently close. When this condition occurs, the probability to obtain a reactive path raises from 19% to 44%.

From Figure 6 (bottom) the probability that this is the most important feature is 3%. One other equivalent feature has a probability of 16%, which involves the distance between O0 and H5, the hydrogen that is connected to O3.

The next split, still along the branch that led to the highest reactive stance, is the distance between O7 and O3 being smaller than 3.84 Å. If a second water molecule is sufficiently close to the FA oxygen, the probability of a reactive path is 60%.

Still along the branch towards the highest reactive stance, the distance between the





B

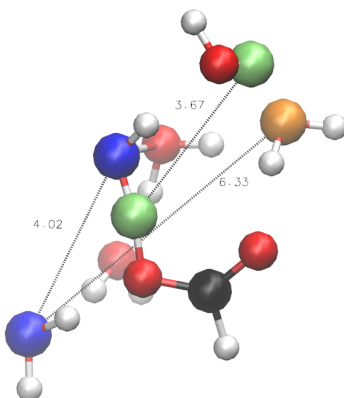


Figure 5: Decision tree for the system with 6 water molecules around the formic acid molecule. Three splits divide the input, each square reports 1) the question which splits the data, 2) the number of samples going into the node, 3) the number of [unreactive, reactive] samples going into the node and 4) the majority class of the node. At each split, the True branch is on the left, False on the right. The color indicates the ratio between unreactive (brown) and reactive (blue) samples included in a node. Wider arrows have been used to link the decision tree split with the atoms involved. In panel B, a 3D representation of the system is provided. The atoms highlighted in blue, yellow, and green, correspond to the atoms involved in the first, second, and third split of the decision tree reported in the panel B. In red are the oxygen and in white the hydrogen atoms not indicated by the decision tree reported in the panel B.

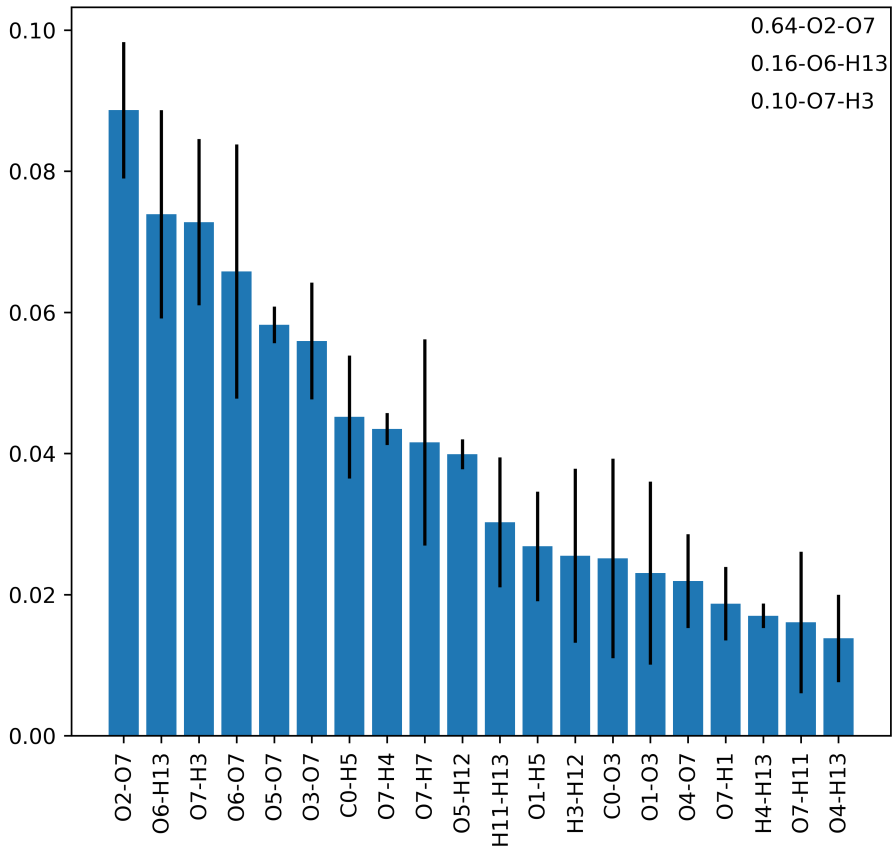


Figure 6: The importance approximations of the first split question from a random forest with depth of one for the four water systems. The bars represent the feature importance of a random forest, with the error bar calculated with a block-error average based on the generated trajectories.

hydrogen H8 and H1 being smaller than 4.35 Å represents the last split here considered. This feature can be interpreted as a structural requirement for the water complex to be reactive, and may also implicitly involve some requirement involving atomic orientations/angles. In such conditions, the probability for the path to be reactive is 84%.

The result reported by the decision trees generated by the index variant and index invariant representations are physically consistent but different in atom selection. That is, different atom pairs for the different representations are identified as most relevant. Yet, the water structure and formic acid orientation are equivalent. It should also be noted that the index variant representation results in a numerically more stable generation of random forests, as the importance values for the symmetric features are summed together. For the distribution of importance of the random forest reported in Figures 4 and 6, the first splits are thus identified with a higher probability.

## References

- (1) Young, G.; Householder, A. S. Discussion of a set of points in terms of their mutual distances. *Psychometrika* **1938**, *3*, 19–22.



## Paper D

# A Comprehensive Guide for Assessing Covalent Inhibition in Enzymatic Assays Illustrated with Kinetic Simulations

Elma Mons, Sander Roet, Robbert Q. Kim, and Monique  
P.C. Mulder

*Current Protocols* **2022**, 2, e419;  
doi: 10.1002/cpz1.419



# A Comprehensive Guide for Assessing Covalent Inhibition in Enzymatic Assays Illustrated with Kinetic Simulations

Elma Mons,<sup>1,2,4</sup> Sander Roet,<sup>3</sup> Robbert Q. Kim,<sup>1</sup> and Monique P. C. Mulder<sup>1,4</sup>

<sup>1</sup>Department of Cell and Chemical Biology, Oncode Institute, Leiden University Medical Center, Leiden, The Netherlands

<sup>2</sup>Current: Institute of Biology Leiden, Leiden University, Leiden, The Netherlands

<sup>3</sup>Department of Chemistry, Norwegian University of Science and Technology, Trondheim, Norway

<sup>4</sup>Corresponding authors: [m.w.e.mons@biology.leidenuniv.nl](mailto:m.w.e.mons@biology.leidenuniv.nl); [m.p.c.mulder@lumc.nl](mailto:m.p.c.mulder@lumc.nl)

Published in the Chemical Biology section

Covalent inhibition has become more accepted in the past two decades, as illustrated by the clinical approval of several irreversible inhibitors designed to covalently modify their target. Elucidation of the structure-activity relationship and potency of such inhibitors requires a detailed kinetic evaluation. Here, we elucidate the relationship between the experimental read-out and the underlying inhibitor binding kinetics. Interactive kinetic simulation scripts are employed to highlight the effects of *in vitro* enzyme activity assay conditions and inhibitor binding mode, thereby showcasing which assumptions and corrections are crucial. Four stepwise protocols to assess the biochemical potency of (ir)reversible covalent enzyme inhibitors targeting a nucleophilic active site residue are included, with accompanying data analysis tailored to the covalent binding mode. Together, this will serve as a guide to make an educated decision regarding the most suitable method to assess covalent inhibition potency. © 2022 The Authors. Current Protocols published by Wiley Periodicals LLC.

**Basic Protocol I:** Progress curve analysis of substrate association competition

**Basic Data Analysis Protocol 1A:** Two-step irreversible covalent inhibition

**Basic Data Analysis Protocol 1B:** One-step irreversible covalent inhibition

**Basic Data Analysis Protocol 1C:** Two-step reversible covalent inhibition

**Basic Data Analysis Protocol 1D:** Two-step irreversible covalent inhibition with substrate depletion

**Basic Protocol II:** Incubation time-dependent potency  $IC_{50}(t)$

**Basic Data Analysis Protocol 2:** Two-step irreversible covalent inhibition

**Basic Protocol III:** Preincubation time-dependent inhibition without dilution

**Basic Data Analysis Protocol 3:** Preincubation time-dependent inhibition without dilution

**Basic Data Analysis Protocol 3Ai:** Two-step irreversible covalent inhibition

**Alternative Data Analysis Protocol 3Aii:** Two-step irreversible covalent inhibition

**Basic Data Analysis Protocol 3Bi:** One-step irreversible covalent inhibition

**Alternative Data Analysis Protocol 3Bii:** One-step irreversible covalent inhibition

**Basic Data Analysis Protocol 3C:** Two-step reversible covalent inhibition

**Basic Protocol IV:** Preincubation time-dependent inhibition with dilution/competition

**Basic Data Analysis Protocol 4:** Preincubation time-dependent inhibition with dilution

Mons et al.

1 of 85

**Basic Data Analysis Protocol 4Ai:** Two-step irreversible covalent inhibition  
**Alternative Data Analysis Protocol 4Aii:** Two-step irreversible covalent inhibition

**Basic Data Analysis Protocol 4Bi:** One-step irreversible covalent inhibition  
**Alternative Data Analysis Protocol 4Bii:** One-step irreversible covalent inhibition

Keywords: biochemical potency • covalent inhibition • enzyme kinetics  
• irreversible inhibition • simulations

#### How to cite this article:

Mons, E., Roet, S., Kim, R. Q., & Mulder, M. P. C. (2022). A comprehensive guide for assessing covalent inhibition in enzymatic assays illustrated with kinetic simulations. *Current Protocols*, 2, e419. doi: 10.1002/cpz1.419

## INTRODUCTION

Traditionally, drug design efforts were focused on small molecules that interact with their biological target through noncovalent interactions in a reversible manner. In contrast, covalent inhibitors have the ability to form a much stronger covalent bond with a nucleophilic amino acid residue at the target protein, which is positioned in close proximity to a reactive (electrophilic) moiety in the inhibitor (Ward & Grimster, 2021). Risks associated with covalent reactions that can take place not only with the desired target but also with off-target proteins, often undiscovered until late-stage clinical development, resulted in drug discovery programs moving away from candidates bearing intrinsically reactive electrophilic moieties (Bauer, 2015; Singh, Petter, Baillie, & Whitty, 2011). Nonetheless, the clinical success of covalent drugs that were being used in the clinic long before their mechanism of action was elucidated, which include aspirin and penicillin, along with the more recent clinical approval and success of targeted covalent inhibitors (TCIs) bearing moderately reactive electrophilic warheads, ultimately triggered the current resurgence of covalent drugs (Abdeldayem, Raouf, Constantinescu, Moriggl, & Gunning, 2020; De Cesco, Kurian, Dufresne, Mittermaier, & Moitessier, 2017; Singh et al., 2011).

The covalent inhibitor development process typically involves identification of noncovalent inhibitors by high-throughput screening (HTS), followed by modification with a moderately reactive electrophilic warhead to improve inhibition potency and selectivity (Engel et al., 2015; Zhang, Hatcher, Teng, Gray, & Kostic, 2019). Alternatively, an electrophilic fragment that forms a covalent bond with the desired enzyme target is first identified in covalent fragment-based drug discovery (Dalton & Campos, 2020; Kathman & Statsyuk, 2019; Resnick et al., 2019), followed by optimization of the noncovalent affinity and positioning of the electrophile. A prerequisite here is that the molecular target must contain a nucleophilic residue (e.g., cysteine, serine, lysine) to form a covalent bond with the electrophilic warhead of the inhibitor (Lagoutte, Patouret, & Winssinger, 2017; Ray & Murkin, 2019). Whether covalent adduct formation is reversible or irreversible depends on the selected electrophilic warhead (Bradshaw et al., 2015; Gehringer & Laufer, 2019; Lee & Grossmann, 2012; Shindo & Ojida, 2021). The PK-PD decoupling is one of the major advantages of irreversible inhibition: an infinite target residence time, resulting in a prolonged therapeutic effect after the inhibitor has been cleared from circulation (Abdeldayem et al., 2020; Barf & Kaptein, 2012; Gabizon & London, 2020; Kim, Hwang, Kim, & Park, 2021). Here, restoration of enzyme activity can only be achieved by *de novo* protein synthesis. At the same time, if the consequences of continued on-target inhibition are poorly understood, this same property can provide a safety concern.



Consequently, inhibitors with a reversible covalent binding mode have become increasingly popular, with (tunable) target residence times ranging from several hours to multiple days (Bradshaw et al., 2015; Owen Dafydd et al., 2021; Serafimova et al., 2012).

Although traditional methods to evaluate inhibitor potency, such as determining half-maximal inhibitory concentration (IC<sub>50</sub> values), are sufficient to identify hits in high-throughput screens, a more detailed kinetic evaluation is required to elucidate the structure-activity relationship (SAR) of irreversible covalent inhibitors (De Cesco et al., 2017; Harris et al., 2018; Holdgate, Meek, & Grimley, 2017). There are many extensive reviews on the history, development, and success of covalent inhibitors (Abdeldayem et al., 2020; De Cesco et al., 2017; Johnson, Weerapana, & Cravatt, 2010; Lagoutte et al., 2017), and experimental methods to assess undesired time-dependent inactivation (TDI) of CYP450 enzymes have been excellently reviewed (Stresser, Mao, Kenny, Jones, & Grime, 2014), but a comprehensive overview of experimental methods compatible with the desired covalent binding mode of TCIs targeting nucleophilic active-site residues has been missing. In the *Strategic Planning* section, we will introduce our customized set of interactive kinetic simulation scripts to study the kinetic concepts of different experimental methods, followed by a general background on (covalent) inhibitor binding modes, the assumptions on experimental enzyme activity assay conditions, and an introduction on time-dependent inhibitor kinetics. Our findings are discussed in detail in the section *Experimental Methods and Data Analysis*, where stepwise protocols are provided for four experimental methods with data analysis tailored to the different covalent binding modes. All are accompanied by an online available set of kinetic simulation scripts and troubleshooting guidelines, allowing readers to evaluate their covalent (ir)reversible inhibitor.

## STRATEGIC PLANNING

This guide has been composed to aid readers that have identified an (ir)reversible covalent inhibitor and are contemplating which experimental method to select for the follow-up SAR analysis. Here, the performance of the enzymatic assay is not expected to be troublesome, but the challenge lies in the design of an assay method that complies with (often implied but not explicitly mentioned) assumptions on experimental conditions, and recognition of artifacts/errors in the interpretation of experimental outcome. As such, we assume that a functioning enzymatic assay with a robust read-out is already in place, and we will focus on the connection between (algebraic) data analysis methods and the respective assumptions on experimental conditions. It is important to note that this work is tailored to enzyme *activity* assays with a (fluorescence) read-out upon substrate processing to form a detectable product, and as such may not be compatible with other assay formats such as ligand *binding* competition assays or direct detection of the covalent enzyme-inhibitor adduct.

In the section '*Kinetic Simulations*', we introduce the interactive kinetic simulation scripts used to illustrate the methods and kinetic concepts in this work. All figures are composed with *in silico* data generated in kinetic simulations, and can be recreated with the information in this section. The section '*Inhibitor Binding Modes*' provides an overview of the (covalent) inhibition binding modes compatible with the methods in this work. It is paramount to select the appropriate algebraic model for data analysis, as the inhibitor binding mode changes the obtainable parameters as well as the compatibility with experimental methods. Covalent EI\* adduct formation should be validated by direct detection with MS, X-ray crystallography or NMR (Harris et al., 2018; Licican et al., 2020; Mons et al., 2019; Mons et al., 2021). Reversibility of covalent adduct formation is commonly assessed in rapid/jump dilution or washout assays with detection of regained enzymatic activity after dilution/washout (Copeland, Basavapathruni, Moyer, & Scott, 2011), MS detection of unbound inhibitor upon denaturation or digestion-mediated

dissociation (Bradshaw et al., 2015), or competitive binding of a (selective) irreversible (activity-based) probe (Liclican et al., 2020; Smith et al., 2017). It is important to note that noncovalent binding can also irreversibly inhibit enzyme activity by aggregation or precipitation (Auld, Inglese, & Dahlin, 2017).

Next, we investigated which assumptions on experimental enzyme activity assay conditions are embedded in the algebraic models used for kinetic analysis. Our findings are outlined in the section ‘*Critical Parameters: Assumptions on Experimental Assay Conditions*’, highlighting which assumptions are crucial and what the consequences are when these assumptions are violated. Finally, we provide a kinetic background on time-dependent (covalent) inhibition in the section ‘*Time-dependent Inhibitor Potency*’. Readers new to the field of enzyme inhibition kinetics are strongly encouraged to familiarize themselves with the work of Copeland for a general introduction into enzyme kinetics (Copeland, 2000, 2013e) before studying advanced kinetic concepts associated with (ir)reversible covalent enzyme inhibition and their relation to experimental enzyme activity read-out.

### **Kinetic Simulations**

Keeping assay requirements in mind, it may seem a daunting task to design, perform, and analyze proper inhibition experiments. In general, practice is the best teacher to get a feeling for these assays and the expected output. Kinetic simulations are essential to understand the importance of reaction conditions and support assay design optimization (Potratz, 2018). In such simulations, one can freely change the parameters to visualize the effect on the output and validate that kinetic parameters found after data analysis correlate with the input values. This design precludes assay artifacts and human error, and also outputs the underlying concentrations of the different reaction species (e.g., unbound enzyme, enzyme-substrate complex), illustrating the relevance of the experimental assay conditions. Finally, kinetic simulations can validate if fitted experimental parameters correlate with the experimental read-out (Pollard & De La Cruz, 2013) and aid the rational design of follow-up experiments by predicting the outcome.

Here, we use a set of customized kinetic simulation scripts based on numerical integration of the differential equations (Walkup et al., 2015) to simulate the time-dependent product concentration as well as the underlying concentrations of various enzyme species (e.g., unbound, bound to inhibitor or substrate). Some concentrations are essentially constant under specific assay conditions, and treating these parameters as constants rather than variables reduces the computing/simulation time. An overview of our kinetic scripts and the assumptions on experimental assay conditions can be found in Table 1. Since understanding kinetics can be greatly facilitated by the ability to adjust reaction conditions and changing parameters without using expensive reagents, we have made interactive versions of these simulation scripts available free of charge at <https://tinyurl.com/kineticsimulations>. We encourage our readers to perform simulations with their own kinetic parameters to visualize how the underlying concentrations of enzyme species affect the detected read-out, and to get a feeling for realistic values and assay conditions. We selected one model inhibitor for each binding mode to generate the figures that exemplify the methods described (the kinetic parameters of each model inhibitor can be found in Table S1 in Supporting Information). All figures in this work can be recreated with the information in Table 1 and Table S1.

Our kinetic simulation scripts are tailored to competitive inhibition, where an intrinsically reactive inhibitor bearing an electrophilic warhead covalently targets a nucleophilic amino acid residue at the enzymatic substrate binding site, thus blocking substrate access (Copeland, 2013e; Holdgate et al., 2017). Other covalent binding modes [e.g., prodrugs (Strelow, 2017), covalent allosteric inhibitors (Lu & Zhang, 2017), and multi-step mechanism-based inhibitors (Tuley & Fast, 2018; Yang, Jamei, Yeo, Tucker,

**Table 1** Kinetic Simulation Scripts Used in this Work<sup>a</sup>

Reaction dynamics	Script	Simulation constants	Experimental restrictions	
$  \begin{array}{c}  E + I \xrightleftharpoons[k_4]{k_3} EI \xrightarrow{k_{\text{met}}} E + P \\  \uparrow \downarrow \\  \begin{array}{c}  k_1 \\  \downarrow \\  ES \\  \downarrow \\  k_2 \\  \downarrow \\  E + P  \end{array}  \end{array}  $	KinGen	Unbound inhibitor Unbound substrate  Volume	$[I]_0 = [I]_{t'} = [I]_t$ $[S]_0 = [S]_t$  $V_{t'} = V_t$	$[I]_0 > 10[E]_0$ $[S]_0 > 10[E]_0$ $[P] < 0.1[S]_0$ $V_{\text{sub}} \ll V_{t'}$
	KinSubDpl	Unbound inhibitor Volume	$[I]_0 = [I]_{t'} = [I]_t$ $V_{t'} = V_t$	$[I]_0 > 10[E]_0$ $V_{\text{sub}} \ll V_{t'}$
	KinVol	Unbound inhibitor	$[I]_0 = [I]_{t'}$ $= (1 + (V_{\text{sub}}/V_{t'})) \times [I]_t$	$[I]_0 > 10[E]_0$
		Unbound substrate	$[S]_0 = [S]_t$	$[S]_0 > 10[E]_0$ $[P] < 0.1[S]_0$
	KinInhDpl	Volume	$V_{t'} = V_t$	$V_{\text{sub}} \ll V_{t'}$
$  \begin{array}{c}  E_{\text{deg}} \uparrow \\  \uparrow \downarrow \\  \begin{array}{c}  E + I \xrightleftharpoons[k_4]{k_3} EI \xrightarrow{k_{\text{met}}} E + P \\  \uparrow \downarrow \\  \begin{array}{c}  k_1 \\  \downarrow \\  ES \\  \downarrow \\  k_2 \\  \downarrow \\  E + P  \end{array}  \end{array}  \end{array}  $	KinDeg <sup>b</sup>	Unbound inhibitor Unbound substrate	$[I]_0 = [I]_{t'} = [I]_t$ $[S]_0 = [S]_t$	$[I]_0 > 10[E]_0$ $[S]_0 > 10[E]_0$ $[P] < 0.1[S]_0$ $V_{\text{sub}} \ll V_{t'}$
		Volume	$V_{t'} = V_t$	
		KinVolDeg <sup>b</sup>	Unbound inhibitor Unbound substrate	$[I]_0 = [I]_{t'}$ $= (1 + (V_{\text{sub}}/V_{t'})) \times [I]_t$ $[S]_0 = [S]_t$

$[I]_0$  = unbound inhibitor concentration at onset of inhibition, before (optional) enzyme binding.  $[I]_{t'}$  = unbound inhibitor concentration during preincubation, after (optional) enzyme binding.  $[I]_t$  = unbound inhibitor concentration during incubation, after (optional) enzyme binding.  $[S]_0$  = unbound substrate concentration at onset of product formation, before enzyme binding.  $[S]_t$  = unbound substrate concentration during incubation, after (optional) enzyme binding and product formation.  $V_{t'}$  = reaction volume during preincubation.  $V_{\text{sub}}$  = volume containing substrate.  $V_t$  = reaction volume during incubation ( $V_t = V_{\text{sub}} + V_{t'}$ ).

<sup>a</sup> Available at <https://tinyurl.com/kineticsimulations>.

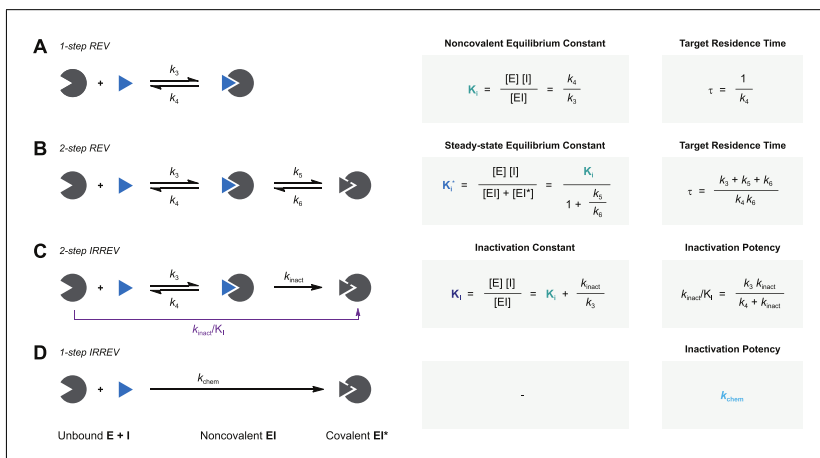
<sup>b</sup> First-order spontaneous enzyme degradation/denaturation.

& Rostami-Hodjegan, 2005)] are outside the scope of this work, although the described experimental protocols can be useful in specific cases. For further instructions and detailed information on restrictions, we refer to the webpage itself.

At the start of the simulations, we define the (pre)incubation time. The preincubation time is the elapsed time since the onset of enzyme inhibition by mixing enzyme and inhibitor, but before the onset of product formation by adding substrate. The incubation time is the elapsed time since onset of product formation: *after* substrate addition. In this work, we will distinguish between incubation and preincubation by using different symbols for preincubation  $t'$  (enzyme and inhibitor) and incubation  $t$  (enzyme, substrate and inhibitor) in all figures and equations to avoid confusion.

### Inhibitor Binding Modes

Reversible noncovalent inhibitors inhibit enzymatic activity by formation of noncovalent EI complex in a single reaction step (Fig. 1A). When the initial unbound inhibitor concentration is equal to inhibition constant  $K_i$ , the concentration of unbound enzyme E will be equal to the concentration of inhibitor-bound enzyme complex EI after steady-state equilibrium has been reached. For traditional fast-binding reversible inhibitors this equilibrium will be reached almost instantly, as association rate constant  $k_3$  and dissociation rate constant  $k_4$  are fast. In this work, the term ‘reaction completion’ relates to the endpoint of enzyme-inhibitor binding, which refers to reaching an equilibrium for



**Figure 1** Schematic overview of inhibitor binding modes (Tuley & Fast, 2018). E = unbound enzyme. I = unbound inhibitor. EI = noncovalent enzyme-inhibitor complex. EI\* = covalent enzyme-inhibitor complex. An overview of kinetic constants can be found in Table S2 (see Supporting Information). Details on equilibrium constants are available in the Supporting Information. **(A)** Classic one-step reversible inhibition. Inhibitor potency ranking based on inhibition constant  $K_i$  (M) or target residence time  $\tau$  (s). **(B)** Two-step reversible covalent inhibition. Inhibitor potency ranking based on steady-state inhibition constant  $K_i^*$  (M) for total  $E + I \leftrightarrow EI + EI^*$  equilibrium or target residence time  $\tau$  (s). **(C)** Two-step irreversible covalent inhibition (affinity label model). Inhibitor potency ranking based on inactivation efficiency: maximum rate of covalent adduct formation over inactivation constant  $k_{inact}/K_i$  ( $M^{-1}s^{-1}$ ). **(D)** One-step irreversible covalent inhibition (residue-specific reagent model). Inhibitor potency ranking based on inactivation efficiency:  $k_{chem}$  ( $M^{-1}s^{-1}$ ) =  $k_{obs}/[I]$  ( $M^{-1}s^{-1}$ ).

reversible inhibitors (Fig. 1A and 1B) or reaching full inactivation for irreversible inhibitors (Fig. 1C and 1D). Contrary to classic fast-binding inhibitors, time-dependent or slow-binding inhibition is observed when the steady-state equilibrium or irreversible inactivation is reached relatively slowly on the assay timescale (Copeland, 2013, 2013b, d). Typically, this is observed for inhibitors with a covalent binding mode (Fig. 1B-D), as formation of a covalent adduct is not an instantaneous process.

Reversible covalent adduct formation (Fig. 1B) is a two-step process consisting of (rapid) initial association to form noncovalent EI complex (*rapid equilibrium approximation*, discussed in more detail in the section ‘*Critical Parameters: Assumptions on Experimental Assay Conditions*’) preceding covalent EI\* adduct formation. Covalent EI\* adduct is at equilibrium with the noncovalent EI complex, as covalent adduct formation is reversible ( $k_6 > 0$ ), with inhibition constant  $K_i$  reflecting the initial noncovalent  $E + I \leftrightarrow EI$  equilibrium and steady-state inhibition constant  $K_i^*$  reflecting the steady-state (overall)  $E + I \leftrightarrow EI + EI^*$  equilibrium. Development of reversible covalent inhibitors typically involves optimization of overall affinity (reflected in low  $K_i^*$  values), preferably by slowing dissociation rates (Fig. 1B). A slow off-rate ( $k_{off}$ ) is favorable, as this is reciprocal with the drug-target residence time  $\tau$  ( $\tau = 1/k_{off}$ ), and a longer residence time has been linked to superior therapeutic potency (Copeland, 2010; Copeland, Pompliano, & Meek, 2006). An overview of relevant kinetic parameters can be found in Table S2 (see Supporting Information).

Inhibition is considered irreversible when its residence time exceeds the normal lifespan of the target enzyme (Holdgate et al., 2017). Dissociation from covalent EI\* adduct is negligible, resulting in full enzyme engagement when reaction completion is reached for irreversible covalent inhibitors (Fig. 1C and 1D). The irreversible binding mode changes the obtainable kinetic parameters to rank inhibitor potency, as the

biochemical  $IC_{50}$  may vary depending on the (pre)incubation time (Holdgate et al., 2017; Singh et al., 2011). The potency of two-step irreversible inhibitors that engage in an initial noncovalent enzyme-inhibitor complex EI prior to formation of covalent adduct EI\* is driven by noncovalent affinity reflected in inactivation constant  $K_I$  along with the maximum rate of inactivation  $k_{inact}$  (Fig. 1C). Rate constant  $k_{inact}/K_I$  is generally accepted as a more suitable measure of two-step irreversible inhibitor potency (Holdgate et al., 2017; Schwartz et al., 2014; Singh et al., 2011; Strelow, 2017), in an analogous fashion to  $k_{cat}/K_M$  reflecting the efficiency of enzymatic substrate conversion (detailed comparison can be found in Table S3 in Supporting Information). The binding mode becomes one-step when noncovalent equilibrium is non-existent, for example for highly reactive thiol-alkylating reagents (McWhirter, 2021; Strelow, 2017), with the parameter  $k_{chem}$  or  $k_{obs}/[I]$  reflecting potency/efficiency (Fig. 1D).

Drug development of irreversible covalent inhibitors is typically geared towards simultaneous improvement of the binding affinity (reflected in a lower  $K_I$  value) and faster covalent bond formation (reflected in a higher  $k_{inact}$  value) to generate irreversible covalent inhibitors with a high  $k_{inact}/K_I$  value for the desired enzyme target (Mah, Thomas, & Shafer, 2014; Schwartz et al., 2014), while minimizing the intrinsic reactivity with undesired enzymes such as GSH (Guan, Williams, Pan, & Liu, 2021; Lonsdale et al., 2017; Martin, MacKenzie, Fletcher, & Gilbert, 2019). Typical reported  $k_{inact}/K_I$  values of irreversible inhibitors range from  $10^3$ - $10^7$   $M^{-1}s^{-1}$  for kinase inhibitors (Schwartz et al., 2014; Telliez et al., 2016; Zhai, Ward, Doig, & Argyrou, 2020),  $10^1$ - $10^5$   $M^{-1}s^{-1}$  for protease inhibitors (Meara & Rich, 1995; Mons et al., 2019; Rocha-Pereira et al., 2014),  $10^2$ - $10^4$   $M^{-1}s^{-1}$  for other target classes (Fell et al., 2020; Hansen et al., 2018; Lanman et al., 2020), to  $10^{-2}$ - $10^2$   $M^{-1}s^{-1}$  for covalent fragments (Johansson et al., 2019; Kathman, Xu, & Stasyuk, 2014). Ranges of clinically relevant  $k_{inact}/K_I$  values are highly dependent on the nucleophilicity of the targeted amino acid (cysteine typically being more reactive than serine) and concentration of naturally present competitors (e.g., ATP-competitive inhibitors need to overcome competition by ATP at physiological concentrations far exceeding the  $K_{M,ATP}$ ).

### Critical Parameters: Assumptions on Experimental Assay Conditions

Experimental conditions should meet certain criteria in order to use algebraic fitting methods. In this paragraph, we focus on the assumptions (*Michaelis–Menten Enzyme Kinetics*, *Enzyme Stability*, *Constant Uninhibited Product Formation Velocity*, *Rapid Equilibrium Approximation*, *Pseudo First-order Reaction Kinetics without Inhibitor Depletion*) on the experimental conditions that are embedded in algebraic equations to analyze time-dependent (covalent) inhibition. Generally, these assumptions involve simplifying the enzyme-inhibitor binding reaction to a single rate-determining step along with fixing inhibitor/substrate concentrations to a constant value. There are two distinct types of algebraic analysis: linear regression (fitting straight curves, compatible with commonly available software such as Excel) and nonlinear regression (fitting exponential curves, requiring sophisticated data fitting software). Linear regression was the predominant method to analyze kinetic data, but has now been surpassed by the more accurate nonlinear regression (Perrin, 2017). For our analyses, we use least-squares nonlinear regression with GraphPad Prism (RRID:SCR\_002798), but other software packages are available too (Rufier, 2021). Please consult the detailed (online) guide on how to implement user-defined equations for nonlinear regression in GraphPad Prism (Motulsky & Christopoulos, 2003; also see Internet Resources section at end of article).

To use algebraic fitting, the experiment should meet all the required conditions outlined below. More complex systems (such as bisubstrate assay or other binding modes like allostery) violate one or more of these and require a different method of fitting. For such systems, numerical integration with dedicated software packages [e.g., KinTek (Johnson,

2009), DynaFit (Kuzmič, 2009)] is recommended. These packages are very powerful, and can fit anything with good error even when the model does not reflect the biological situation (Mayer, Khairy, & Howard, 2010). For these complex systems, it is crucial to ensure that the initial values are reasonable and the amount of (orthogonal) data is sufficient for the amount of parameters that are fitted. The first step, however, whether working with complex systems or reactions with a single rate-determining step, should always be optimization of the experimental conditions.

### ***Michaelis–Menten enzyme kinetics***

All experimental methods in this manuscript are based on enzyme activity assays with multiple turnovers per enzyme, with enzyme release after product formation. We assume that the uninhibited enzymatic substrate processing reaction ( $E + S \rightleftharpoons ES \rightarrow E + P$ ) complies with Michaelis–Menten enzyme kinetics (Pollard & De La Cruz, 2013; Rufer, 2021). The concentration of unbound substrate has to be constant ( $[S]_t = [S]_0$ ) and not depleted by engagement in a (non)covalent complex ES ( $[ES]_t < 0.1[S]_0$ ) or conversion into product. Therefore, substrate is added in a large excess over the enzyme ( $[S]_0 > 10[E]_0$ ), and the uninhibited velocity of product formation ( $v^{\text{ctrl}}$ ) is calculated over the linear part corresponding to less than 10% substrate conversion ( $[P]_t < 0.1[S]_0$ ) (Wu, Yuan, & Hodge, 2003). The signal corresponding to 10% substrate conversion can be estimated from a product calibration/titration curve (Dharadhar et al., 2019; Janssen et al., 2019) to avoid substrate depletion. The effect of substrate depletion can be investigated with the kinetic simulation script **KinSubDpl**. More complex enzymatic (bisubstrate) assays (Copeland, 2000) are outside of the scope of this work. However, the methods described herein could still be applicable under pseudo-single substrate (Hit-and-Run) conditions.

### ***Enzyme stability***

Unless otherwise noted, time-dependent decrease of enzyme activity is attributed solely to the presence of a (slow-binding) inhibitor. It is thus assumed that the enzyme activity is constant throughout the whole experiment, although this does not necessarily reflect the actual experimental situation. Recombinant enzymes do not have an eternal life; thus, time-dependent loss of enzyme activity will inevitably occur due to spontaneous protein denaturation, degradation, or unfolding (Miyawaki, Kanazawa, Maruyama, & Dozen, 2017). The Selwyn test is a relatively simple test to see if time-dependence of uninhibited enzyme activity is due to (spontaneous) enzyme inactivation (Selwyn, 1965). Spontaneous enzyme degradation/denaturation is similar to radioactive decay in a sense that inactivation is a first-order reaction (*degradation rate* =  $k_{\text{degE}} \times [E]$ ). Enzyme stability might be promoted by optimization of the assay buffer, and is less significant at shorter (pre)incubation times, but degradation cannot completely be avoided. Therefore, we included data analysis methods to account for spontaneous first-order enzyme degradation/denaturation. Cannibalistic proteases (Ferrall-Fairbanks, Kieslich, & Platt, 2020) follow a second-order (auto)proteolysis rate (*degradation rate* =  $k_{\text{degE}} \times [E]^2$ ) and are as such outside of the scope of these methods. In simulations to illustrate the methods described herein (with kinetic simulation scripts **KinDeg** and **KinVolDeg**), we assumed that first-order decay is uniform for all enzyme species ( $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}} = k_{\text{degEI}^*}$ ) and combined the individual degradation rates into the enzyme degradation rate constant  $k_{\text{deg}}$ .

### ***Constant uninhibited product formation velocity***

The uninhibited controls should be linear for the whole measurement when analyzing time-dependent inhibition. There are various factors contributing to a slight time-dependent decrease of product formation velocity in the absence of inhibitor (Copeland, 2000), thus violating this assumption. An overview of common troubleshooting options

is listed in Table 3 (located in the troubleshooting section at the end of this document). As discussed above, substrate depletion ( $[P] > 0.1[S]_0$ ) negatively influences the linearity over time, as does product inhibition ( $[P] > 0.1K_{D,P}$ ). Fortunately, this can be avoided by decreasing the enzyme concentration and/or shortening the incubation time to reduce substrate turnover, thereby lowering the absolute and relative product concentration. Other factors, such as quenching of the fluorescent product signal by photobleaching (Johnson, 2010), can make the results look nonlinear. This effect can be reduced by increasing the measurement interval and/or reducing the number of excitation cycles. Finally, optimization of assay conditions can minimize the effect of spontaneous loss of enzyme activity ( $k_{deg} > 0$ ), but cannot be resolved completely. In this work, we will refer to the overall rate of nonlinearity in the uninhibited control ( $k_{obs}$  of  $[I] = 0$ ) with the symbol  $k_{ctrl}$ , regardless of the underlying mechanism that causes the time-dependent decrease of product formation velocity.

### **Rapid equilibrium approximation**

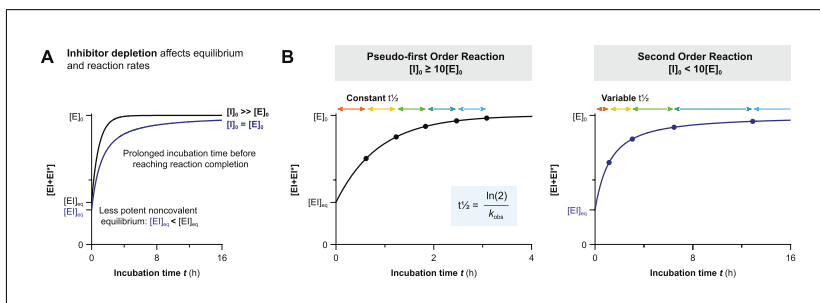
Algebraic analysis of (covalent) inhibition is based on the assumption that time-dependent inhibition is driven by a single rate-determining step. For two-step covalent inhibitors (Fig. 1B and 1C), this means that the noncovalent  $E + I \rightleftharpoons EI$  equilibrium that precedes covalent  $EI^*$  adduct formation should be reached almost instantly after the onset of inhibition. After this rapid equilibrium, a much slower step of covalent adduct formation follows ( $k_{inact} \ll k_4$ ). Whether the noncovalent equilibrium indeed is reached rapidly is an intrinsic inhibitor property, and (kinase) inhibitors with a low-nM noncovalent potency are likely to violate this assumption: the association rate constant is diffusion-limited ( $k_3 \leq 10^9 \text{ M}^{-1}\text{s}^{-1}$ ), and thus  $k_4$  must be relatively slow if  $K_i \leq 10^{-8} \text{ M}$  (Kuzmič, 2020a). Unfortunately, a slow initial, noncovalent step is not easily recognized from raw kinetic data, resulting in overestimation of the rate of inactivation  $k_{inact}$  and underestimation of the inactivation constant  $K_i$  with algebraic rather than numerical data analysis.

The inactivation constant  $K_i$  approximates inhibition constant  $K_i$  ( $K_i \approx K_i$ ) when covalent bond formation is driven by the rate-determining conversion of noncovalent complex  $EI$  into covalent adduct  $EI^*$  ( $k_{inact} \ll k_4$ ) (Fig. 1C), analogous to the Briggs–Haldane treatment of enzyme-substrate kinetics where  $K_M \approx K_S$  if  $k_{cat}$  is rate-limiting (McWhirter, 2021). Consequently,  $K_i$  and  $K_i$  may have the same value, but they are not interchangeable, and it is as such recommended to report  $k_{inact}/K_i$  rather than  $k_{inact}/K_i$ .

### **Pseudo first-order reaction kinetics without inhibitor depletion**

Algebraic analysis of (covalent) inhibition is typically based on the assumption that the unbound inhibitor concentration is a constant value ( $[I]_t = [I]_0$ ) unaffected by enzyme binding (Pollard & De La Cruz, 2013). This assumption is only valid when the inhibitor is present in large excess with respect to the enzyme ( $[I]_0 > 10[E]_0$ ) at reaction initiation. The enzyme occupancy after reaching the noncovalent equilibrium is driven solely by the excess inhibitor concentration relative to the (apparent) inhibition constant  $K_i^{app}$ :  $[EI]_{eq}/[E]_0 = 1/(1 + (K_i^{app}/[I]))$ . The effect of inhibitor depletion can be investigated with the kinetic simulation script **KinInhDpl**. Violation of this assumption results in an appreciable reduction of the remaining population of unbound inhibitor upon complexation with enzyme. Consequently, the inhibitor occupancy at equilibrium no longer reflects the apparent inhibition constant  $K_i^{app}$  because the equilibrium is now driven by both enzyme and inhibitor concentration (Fig. 2A). Algebraic correction for inhibitor depletion ( $[I]_t < [I]_0$ ) to find the equilibrium constant  $K_i$  is often performed for one-step reversible inhibitors displaying tight-binding behavior (with low inhibitor concentrations because  $K_i^{app}$  approaches  $[E]^{total}$ ), by fitting the (steady-state) equilibrium product formation velocity to (variants of)





**Figure 2** Consequences of inhibitor depletion. Simulated with **KinInhDpl** for 50 nM inhibitor **C** with 5 nM enzyme ( $[I]_0 = 10[E]_0$ ) or 50 nM enzyme ( $[I]_0 = [E]_0$ ). **(A)** Inhibitor depletion (blue line) results in lower noncovalent equilibrium occupancy  $[E]_{eq}$  calculated with Morrison's quadratic equation (available in Supporting Information) and slower reaction rates resulting in longer incubation time to reach full inactivation than for excess inhibitor (black line). **(B)** First-order reaction conditions with constant half-life  $t_{1/2}$  when inhibitor is present in excess (left). Second order reaction conditions with variable half-life  $t_{1/2}$  and longer overall reaction time when inhibitor is depleted (right).

Morrison's quadratic equation (Copeland, 2013c; Murphy, 2004) that treat the inhibitor concentration as a variable rather than a constant value (more details in Supporting Information). However, these equations are only compatible with inhibitors with a reversible binding mode after equilibrium has been reached, and are thus not suitable for irreversible inhibition.

Binding of inhibitor to enzyme is, in principle, a second-order reaction: the association rate depends on the concentration of unbound enzyme as well as unbound inhibitor, which both decrease upon formation of association product EI. Towards the end of the reaction, the reaction rate is significantly slower when less of the unbound components are left. Algebraic analysis of second-order (ir)reversible association curves is complicated (data not included, simulated with simulation script **KinInhDpl**), even for inhibitors with a one-step binding (Fig. 1); thus, it is strongly advised to analyze second-order reactions of two-step (ir)reversible inhibitors by numeric integration (Copeland, 2013a). However, as mentioned above, unbound inhibitor concentrations remain more or less constant during the reaction if the inhibitor is present in excess at reaction initiation ( $[I]_0 > 10[E]_0$ ). Consequently, the second-order binding reaction of enzyme and inhibitor behaves like a first-order reaction when the inhibitor is present in excess: pseudo-first order reaction kinetics (Copeland, 2013a). The time-dependent association reaction for a (pseudo-)first order reaction has a constant half-life  $t_{1/2}$ , and the progress curves can be fitted to standard one-phase exponential association equations (Fig. 2B, left), as will be discussed in more detail in the next section.

Second-order kinetic association reactions require a longer overall time to reach reaction completion of the enzyme-inhibitor binding reaction (inactivation or equilibrium) with a variable half-life  $t_{1/2}$  (Fig. 2B, right), because the association reaction rate slows down when the remaining unbound inhibitor concentration decreases. For two-step (ir)reversible inhibitors, the time-dependent reduction in covalent reaction rate is a direct consequence of the decreasing noncovalent occupancy upon inhibitor depletion. The rate-determining step of covalent adduct formation is preceded by noncovalent complex EI formation, and is thus limited by noncovalent occupancy, which decreased over time.

### Time-Dependent Inhibitor Potency

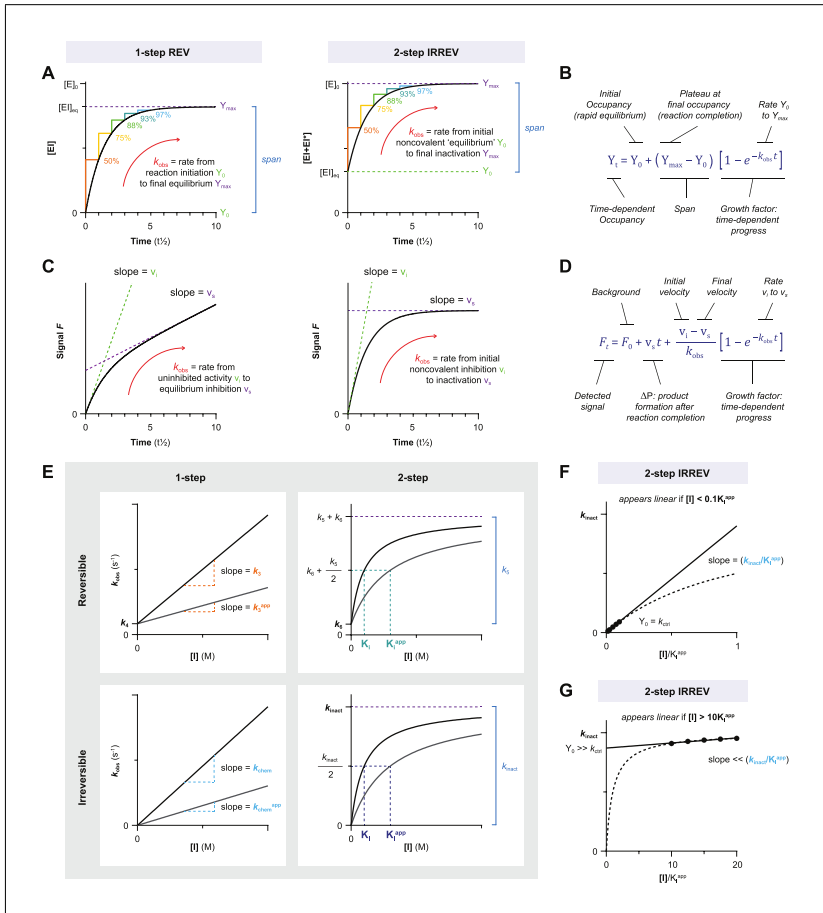
Methods to analyze time-dependent inhibitors are based on the fact that it takes time to reach completion, and we use this information to obtain kinetic parameters. Under



pseudo-first-order conditions (Copeland, 2013a) based on a single rate-determining step, inhibitor binding follows an exponential one-phase association reaction (Pollard & De La Cruz, 2013) from the rapid initial binding (*rapid equilibrium approximation*) to (slowly) reaching a plateau at *reaction completion*: equilibrium for reversible inhibitors (Fig. 3A, right) or inactivation for irreversible inhibitors (Fig. 3A, right). The incubation time to reaction completion is infinite, but after five half-lives ( $t = 5t_{1/2}$ ) reaction progress is at 97%, which is generally sufficient to be considered reaction completion (Fig. 3A). Reaction half-life  $t_{1/2}$  is inversely related to observed reaction rate  $k_{\text{obs}}$  (Copeland, 2013a):  $t_{1/2} = \text{LN}(2)/k_{\text{obs}}$ .  $k_{\text{obs}}$  is the experimental reaction rate for reaction progress from initial binding to reaction completion under the specific assay conditions. Inhibitor concentration as well as competing substrate concentration are major contributors to the observed reaction rate  $k_{\text{obs}}$ . The experimental  $k_{\text{obs}}$  value can be obtained by fitting the time-dependent binding/occupancy curve to exponential one-phase association Equation I (Fig. 3B) from initial to final enzyme occupancy.

Biochemical inhibitor potency is seldom assessed by direct observation of enzyme complex/adduct. Typically, enzyme inhibition is indirectly assessed in *in vitro* assays with a detectable read-out for product formation as a measure of (remaining) enzyme activity. Consequently, reversible enzyme inhibition may have reached the enzyme-inhibitor binding equilibrium (*reaction completion*), but not all enzyme is occupied (unless  $[I] \gg K_i^{\text{app}}$ ) so the remaining fraction of unbound enzyme continues to convert substrate into product (Fig. 3C, left). The reaction is no longer accurately reflected by Equation I (Fig. 3B), as product concentration at reaction initiation does not reflect the initial binding equilibrium, and product concentration does not reach a plateau after reaching the noncovalent equilibrium (reaction completion) for reversible inhibitors. Therefore, time-dependent product formation is fitted to exponential one-phase association Equation II (Fig. 3D) to obtain observed reaction rate  $k_{\text{obs}}$  from initial to final product formation velocity. For irreversible inhibitors, the initial velocity  $v_i$  reflects the (remaining) enzyme activity after rapid noncovalent association, and final velocity  $v_s = 0$  as this reflects full enzyme inactivation.

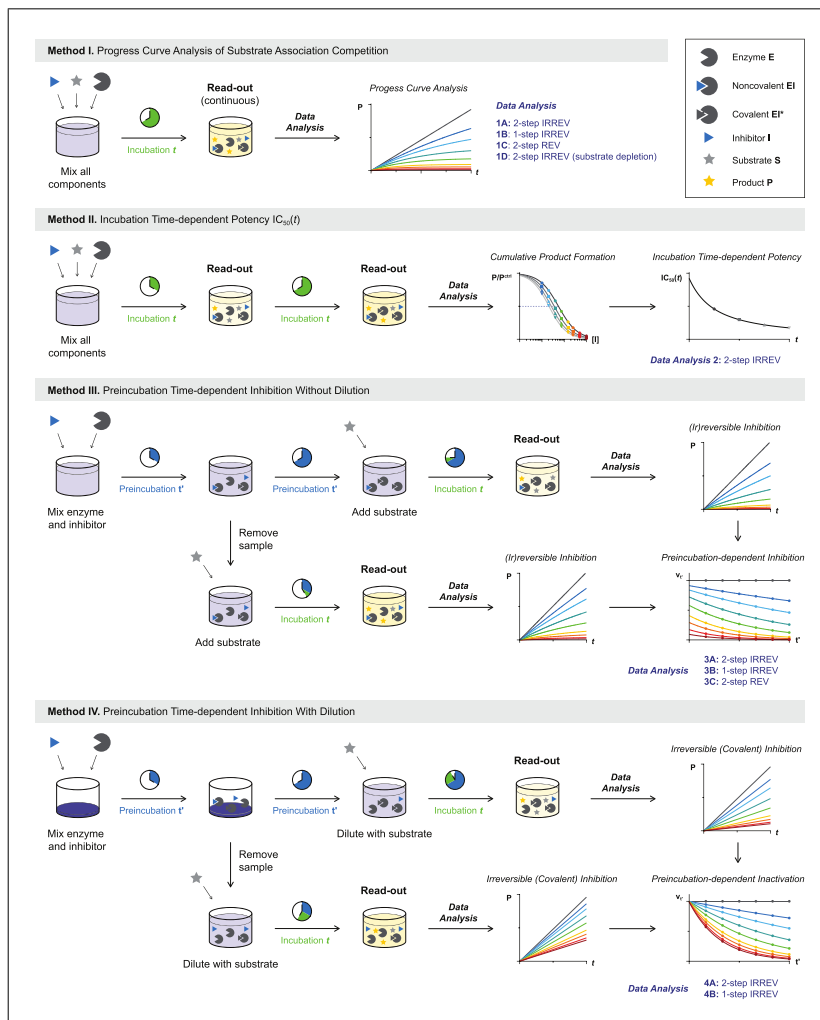
Typically, substrate competition assays are run at various inhibitor concentrations, and the concentration-dependent  $k_{\text{obs}}$  is fitted to obtain kinetic parameters (Fig. 3E). In this work, equations and simulations are tailored to competitive binding of inhibitor and substrate (Holdgate et al., 2017; Rufer, 2021). Consequently, the observed reaction rate  $k_{\text{obs}}$  (Fig. 3E) in the presence of competing substrate is slower, and apparent kinetic constants (marked with <sup>app</sup>) need to be corrected for substrate competition to reflect the kinetic inhibitor potency. Unless otherwise noted, nonlinearity in the uninhibited control  $k_{\text{ctrl}}$  ( $k_{\text{obs}}$  of  $[I] = 0$ ) is assumed to be 0. The relation between  $k_{\text{obs}}$  and inhibitor concentration holds important information on the inhibitor binding mechanism. A linear  $k_{\text{obs}}$  increase with inhibitor concentration is a hallmark of a one-step binding mode, as reaction rates are only limited by experimental factors such as solubility. Plots of  $k_{\text{obs}}$  against two-step inhibitor concentrations are hyperbolic, as the experimental covalent EI\* association rate is limited by EI occupancy, which reaches its maximum ( $k_{\text{inact}}$  or  $k_5$ ) at saturating inhibitor concentration, as shown in Figure 3E:  $[I] > 10K_i$  for 2-step IRREV or  $[I] > 10K_i$  for 2-step REV. An exception to this general observation is inhibitors with a two-step binding mode that will display a linear relationship (Strelow, 2017) when assessed at all non-saturating inhibitor concentrations (Fig. 3F) or all saturating inhibitor concentrations (Fig. 3G). These one-step binding behaviors can be distinguished from the Y-intercept ( $Y_0 = k_{\text{ctrl}}$  for  $[I] \ll K_i^{\text{app}}$  and  $Y_0 > k_{\text{ctrl}}$  for  $[I] \gg K_i^{\text{app}}$ ) along with the noncovalent inhibition of enzyme activity ( $v_i = v^{\text{ctrl}}$  for  $[I] \ll K_i^{\text{app}}$  and  $v_i < v^{\text{ctrl}}$  for  $[I] \gg K_i^{\text{app}}$ ).



**Figure 3** Time-dependent Inhibition and Reaction Completion. Simulated with **KinGen** for 1 pM enzyme with substrate **S1**. **(A)** Time-dependent enzyme occupancy simulated for 50 nM one-step reversible inhibitor **A** (left) or two-step irreversible inhibitor **C** (right) in presence of 100 nM substrate **S1**. Each half-life  $t_{1/2}$ , the occupancy increases by 50% (of the remaining span). After  $5t_{1/2}$ , occupancy is at 97% of its maximum (equilibrium concentration  $[E]_{eq}$  or total enzyme concentration  $[E]_0$ ) and generally considered as reaction completion. Half-life  $t_{1/2}$  is inversely related with observed reaction rate  $k_{obs}$  (under pseudo-first order conditions). **(B)** Bounded exponential association Equation I from initial occupancy (rapid equilibrium) to final occupancy (reaction completion). **(C)** Progress curve of time-dependent product formation for enzyme inhibition in Figure 3A. Product formation velocity (slope, in AU/s), reflecting the (remaining) enzyme activity decreases until reaction completion is reached (steady-state equilibrium or inactivation). **(D)** Exponential association Equation II from initial velocity  $v_i$  (rapid equilibrium) to final velocity  $v_s$  (reaction completion). **(E)**  $k_{obs}$  curves in absence (black,  $[S] = 0$ ) or presence (gray,  $[S] = 2K_M$ ) of competing substrate. Apparent values are not yet corrected for substrate competition. **(F)** Two-step irreversible covalent inhibitors display one-step behavior at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_i$ ). Fit straight line with Y-intercept =  $k_{ctrl}$  to obtain  $k_{chem} = (k_{inact}/K_i)$  from the linear slope. **(G)** Two-step irreversible covalent inhibitors display one-step behavior at saturating inhibitor concentrations ( $[I] > 10K_i$ ). Distinguish from non-saturating inhibitor concentrations in Figure 3F: Y-intercept  $> k_{ctrl}$  when fitting a straight line to the  $k_{obs}$  curve.

## EXPERIMENTAL METHODS AND DATA ANALYSIS

We will discuss four methods in this work (Progress Curve Analysis of Substrate Association Competition, Incubation Time–Dependent Potency  $IC_{50}(t)$ , Preincubation Time–Dependent Inhibition without Dilution, and Preincubation Time–Dependent Inhibition with Dilution/Competition) with accompanying data analysis protocols depending on the inhibitor binding mode (Fig. 4; also see Table 2). For each method, we will start with an overview of the general conceptual background and assay design considerations. Subsequent data analysis is subdivided into protocols tailored to a specific inhibitor binding mode, and for each data analysis protocol we will illustrate the ‘ideal’ situation with kinetic simulations to guide interpretation of results. A practical comment on the nomenclature used: we use the word ‘fit’ for nonlinear fits of raw data (in e.g., GraphPad as part



**Figure 4** Schematic overview of experimental protocols to analyze covalent inhibitor potency included in this work. Incubation time–dependent enzyme inhibition in *Method I* and *II*. Preincubation time–dependent enzyme inhibition in *Method III* and *IV*. Data Analysis protocols are tailored to 2-step IRREVERSIBLE inhibition (shown in Fig. 1C), 1-step IRREVERSIBLE inhibition (shown in Fig. 1D), or 2-step REVERSIBLE inhibition (shown in Fig. 1B).

Table 2 Concise Summary of Methods

Method	Data analysis Protocol	Binding mode	Readout and experimental conditions <sup>a</sup>	Obtainable kinetic parameters	Comments/remarks	Literature reference			
I	1A	2-step IRREV	Continuous	$k_{\text{inact}}, K_I$ & $k_{\text{inact}}/K_I$	Progress curve analysis is favored for very potent inhibitors as competing substrate is present during incubation. Optimization of reaction conditions to minimize assay artefacts can be challenging but rewards with the most simple experimental procedure.	Copeland, 2013b			
				1B			1-step IRREV	Continuous	$k_{\text{chem}}$
				2-step IRREV			Continuous	$k_{\text{inact}}/K_I$	
1C	2-step REV	Continuous	Continuous $[I] \ll K_i^{\text{app}}$ $k_{\text{ctrl}} \ll k_6$	$K_i^*$	Progress curve analysis is disfavored for 2-step reversible inhibitors as algebraic correction for spontaneous loss of enzyme activity is NOT possible.				
II	1D	2-step IRREV	Continuous $[P]_t > 0.1[S]_0$ $[S] \ll 0.1K_M$	$k_{\text{inact}}, K_I$ & $k_{\text{inact}}/K_I$	Algebraic correction for substrate depletion.	Kuzmič et al., 2015			
III	2	2-step IRREV	Continuous/Quenched $k_{\text{ctrl}} = 0$	$k_{\text{inact}}, K_I$ & $k_{\text{inact}}/K_I$	Incubation time-dependent potency enables use of quenched assays but is sensitive to spontaneous loss of enzyme activity.	Krippendorff et al., 2009			
				3Ai			2-step IRREV	Continuous/Quenched $[S] \ll K_M$ $V_{\text{sub}} \ll V_t$	$k_{\text{inact}}, K_I$ & $k_{\text{inact}}/K_I$
	3Aii	2-step IRREV	Continuous/Quenched $[S] \ll K_M$ $V_{\text{sub}} \ll V_t$	$k_{\text{inact}}, K_I$ & $k_{\text{inact}}/K_I$					

(Continued)

**Table 2** Concise Summary of Methods, *continued*

Method	Data analysis Protocol	Binding mode	Readout and experimental conditions <sup>a</sup>	Obtainable kinetic parameters	Comments/remarks	Literature reference
3Bi		1-step IRREV	Continuous/Quenched [S] << K <sub>M</sub> V <sub>sub</sub> << V <sub>t</sub>	k <sub>chem</sub>		
		2-step IRREV	Continuous/Quenched [S] << K <sub>M</sub> V <sub>sub</sub> << V <sub>t</sub> [I] << K <sub>I</sub>	k <sub>inact</sub> /K <sub>I</sub>		
3Bii		1-step IRREV	Continuous/Quenched [S] << K <sub>M</sub> V <sub>sub</sub> << V <sub>t</sub>	k <sub>chem</sub> or k <sub>obs</sub> /I		
		2-step IRREV	Continuous/Quenched [S] << K <sub>M</sub> V <sub>sub</sub> << V <sub>t</sub> [I] << K <sub>I</sub>	k <sub>inact</sub> /K <sub>I</sub>		
3C		2-step REV	Continuous/Quenched [S] << K <sub>M</sub> V <sub>sub</sub> << V <sub>t</sub>	K <sub>i</sub> *	Favored for 2-step REV inhibitors, with algebraic correction for spontaneous loss of enzyme activity by normalization.	
4Ai		2-step IRREV	Continuous/Quenched [S] >> K <sub>M</sub> V <sub>sub</sub> >> V <sub>t</sub>	k <sub>inact</sub> · K <sub>I</sub> & k <sub>inact</sub> /K <sub>I</sub>	Preincubation without dilution is favored for inhibitors with low (noncovalent) affinity, or to study the contribution of covalent bond formation. <i>Data Analysis Protocols 4Ai</i> and <i>4Bi</i> are favored for comparison of multiple inhibitors on a single target. <i>Data Analysis Protocols 4Aii</i> and <i>4Bii</i> are favored for selectivity evaluation of a single inhibitor on multiple targets.	Kitz & Wilson, 1962
		2-step IRREV	Continuous/Quenched [S] >> K <sub>M</sub> V <sub>sub</sub> >> V <sub>t</sub>	k <sub>inact</sub> · K <sub>I</sub> & k <sub>inact</sub> /K <sub>I</sub>		

(Continued)

**Table 2** Concise Summary of Methods, *continued*

Method	Data analysis Protocol	Binding mode	Readout and experimental conditions <sup>a</sup>	Obtainable kinetic parameters	Comments/remarks	Literature reference
4Bi		1-step IRREV	Continuous/Quenched [S] >> K <sub>M</sub> V <sub>sub</sub> >> V <sub>t</sub>	k <sub>chem</sub>		
		2-step IRREV	Continuous/Quenched [S] >> K <sub>M</sub> V <sub>sub</sub> >> V <sub>t</sub> [I] << K <sub>I</sub>	k <sub>i,naet</sub> /K <sub>I</sub>		
4Bii		1-step IRREV	Continuous/Quenched [S] >> K <sub>M</sub> V <sub>sub</sub> >> V <sub>t</sub>	k <sub>chem</sub> or k <sub>obs</sub> /I		
		2-step IRREV	Continuous/Quenched [S] >> K <sub>M</sub> V <sub>sub</sub> >> V <sub>t</sub> [I] << K <sub>I</sub>	k <sub>i,naet</sub> /K <sub>I</sub> or k <sub>obs</sub> /I		

<sup>a</sup> General assay conditions for all methods (unless otherwise noted/specified): [I] > 10[E], [S] > 10[E], [P]<sub>t</sub> < 0.1[S]<sub>0</sub>.

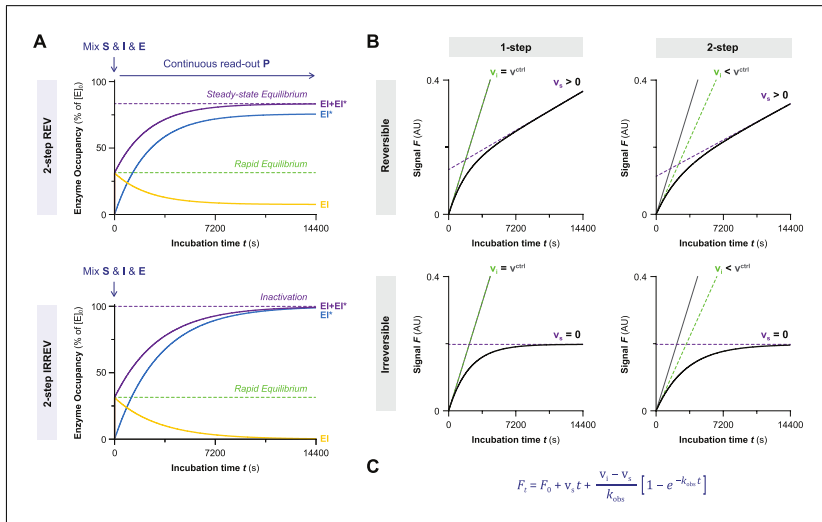
of data analysis protocols) and ‘calculate’ to denote that we calculate parameters from experimental values (in e.g., EXCEL as part of sample calculations). Furthermore, pointers on identification of deviations such as nonlinearity in the uninhibited control ( $k_{\text{ctrl}} > 0$ ) will be given along with algebraic corrections or troubleshooting options to resolve issues.

*Methods I and II* are based on incubation time–dependent enzyme inhibition (Fig. 4). Here, substrate and inhibitor are mixed, and the reaction is initiated by addition of enzyme: i.e., simultaneous onset of product formation and enzyme inhibitor. *Methods III and IV* are based on enzyme inhibition after preincubation. Here, enzyme is preincubated with inhibitor before substrate addition. Two major factors contribute to selection of the appropriate experimental method for your enzymatic inhibition assay: the available enzyme activity assay and the inhibitor binding mode. Recombinant enzyme inhibition is assessed in an *in vitro* enzyme activity assay with detectable product formation (Acker & Auld, 2014; Bisswanger, 2014). This can be a continuous read-out for enzymatic processing of fluorogenic substrates (e.g., fluorescence intensity, FRET) or be a stopped/quenched assay that may require a secondary development/quenching or separation step to detect the formed product (or remaining substrate) such as LC/MS-based assays, conversion of radiolabeled substrate, and commercial assay technologies including ADP-Glo™ (Promega) ATP consumption/ADP production assays, HTRF® KinEASE™ (Cisbio) and Z'-LYTE (Invitrogen) phosphorylation assays, and Amplex® Red (Invitrogen) hydrogen peroxide/peroxidase assays (Acker & Auld, 2014; Bisswanger, 2014). *Method I* is only compatible with homogeneous enzymatic assays that allow continuous read-out, such as cleavage of fluorogenic reporter peptides by proteases. *Methods II-IV* are also compatible with quenched/stopped assays with development step prior to read-out.

## METHOD I: PROGRESS CURVE ANALYSIS OF SUBSTRATE ASSOCIATION COMPETITION

Progress curve analysis is an established method for kinetic analysis of slow-binding inhibitors based on continuous detection of product formation after the substrate processing/product formation reaction has been initiated by addition of enzyme to a mixture of inhibitor and substrate (Fig. 5A). A single measurement at each inhibitor concentration is sufficient, which is convenient when comparing the potency of multiple inhibitors on the same target. However, this method requires the availability of an activity assay format with a continuous read-out, thereby limiting the substrates that can be used. Additionally, assay optimization for progress curve analysis is labor intensive: it is not uncommon to perform multiple pilot experiments to find suitable concentrations of substrate, enzyme, and inhibitor that ensure linear product formation in the uninhibited control (consult Table 3 in the troubleshooting section near the end of the article for troubleshooting).

For ‘slow-binding’ inhibitors, the slope of time-dependent product formation exponentially decreases from initial product formation velocity  $v_i$  (rapid noncovalent inhibition) to the final product formation velocity  $v_s$  (reaction completion) (Fig. 5B) (Copeland, 2013b). The progress curve of time-dependent product formation (as detected signal  $F_t$  in AU) is fitted to a general exponential inhibitor association Equation II (Fig. 5C) to obtain the observed rate of reaction completion  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i$  (in AU/s) to final velocity  $v_s$  (in AU/s). One-step or two-step binding modes can be identified by (visual) inspection of the initial velocity (Fig. 5B). The value of initial velocity  $v_i$  is inhibitor concentration–dependent for two-step (ir)reversible inhibitors that form a rapid (noncovalent) equilibrium ( $v_i < v^{\text{ctrl}}$ ) because the noncovalent enzyme-inhibitor complex already inhibits the enzyme activity (*rapid equilibrium approximation*). Similarly, the value of initial velocity  $v_i$  is equal to the uninhibited velocity  $v^{\text{ctrl}}$  in lieu of a



**Figure 5** Method I: Progress curve analysis of substrate association competition. Simulated with **KinGen** for 1 pM enzyme and 100 nM substrate **S1**. **(A)** The reaction between enzyme, inhibitor, and substrate is initiated by addition of enzyme. Product formation is monitored continuously to detect the time-dependent enzyme activity. Top: simulated for 50 nM reversible two-step inhibitor **B**. Bottom: simulated for 50 nM irreversible two-step inhibitor **C**. Enzyme inhibition increases with time-dependent formation of covalent  $EI^*$  until reaching reaction completion. Initially, total enzyme occupancy  $[E + EI^*]$  reflects the rapid noncovalent equilibrium  $[E]_{eq}$ . At reaction completion ( $t > 5t_{1/2}$ ), total enzyme occupancy  $EI + EI^*$  reflects the steady-state equilibrium (reversible) or inactivation (irreversible). **(B)** Typical progress curves for enzyme activity in presence of time-dependent inhibitors. Time-dependent product formation decreases exponentially from initial velocity  $v_i$  (dashed green line) to the steady-state velocity  $v_s$  (dashed purple line) at reaction completion ( $t > 5t_{1/2}$ ).  $v_i = v^{ctrl}$  when  $[I] \ll K_i^{app}$  (and for one-step inhibitors) with  $v^{ctrl}$  = linear product formation in uninhibited control (gray line). Simulated for 50 nM one-step reversible inhibitor **A**, two-step reversible inhibitor **B**, one-step irreversible inhibitor **D**, or two-step irreversible inhibitor **C**. **(C)** General exponential association Equation II to fit progress curves of time-dependent inhibition. Parameters are constrained depending on the inhibitor binding mode. Irreversible inhibition:  $v_s = 0$  (inactivation at reaction completion). One-step inhibition:  $v_i = v^{ctrl}$  (noncovalent complex is not significant at non-saturating inhibitor concentrations).  $F_t$  = time-dependent signal resulting from product formation (in AU).  $F_0$  = Y-intercept = background signal at reaction initiation (in AU).  $v_i$  = initial product formation velocity (in AU/s).  $v_s$  = final/steady-state product formation velocity (in AU/s).  $t$  = incubation time after enzyme addition (in s).  $k_{obs}$  = observed rate of time-dependent inhibition from initial  $v_i$  to final  $v_s$  (in  $s^{-1}$ ). Also fit the uninhibited/fully inhibited controls to obtain reference values for uninhibited velocity  $v^{ctrl}$  and the rate of nonlinearity in the uninhibited control  $k_{ctrl}$ .

rapid initial binding step, as can be observed for two-step (ir)reversible inhibitors at non-saturating concentrations ( $[I] \ll K_i^{app}$ ) and one-step (ir)reversible inhibitors ( $v_i < v^{ctrl}$ ). Irreversible inhibitors are expected to reach 100% inhibition at reaction completion for all inhibitor concentrations, provided inhibitor is present in large excess and the reaction does not exceed the dynamic enzyme lifetime. Therefore, the final velocity  $v_s$  is restrained to full inhibition ( $v_s = 0$ ) for two-step irreversible inhibitors (*Data Analysis 1A*) and one-step irreversible inhibitors (*Data Analysis 1B*). Two-step reversible inhibitors will reach a reversible steady-state equilibrium ( $v_s \geq 0$ ) upon reaction completion (*Data Analysis 1C*). Be aware that the product formation progress curve is not only linear for fast-binding inhibitors but will also appear linear for slow-binding inhibitors if reaction completion is much slower than the time course of the assay ( $t \ll t_{1/2}$ ). Importantly, the noncovalent equilibrium is assumed to be reached instantly for two-step inhibitors (rapid equilibrium approximation). An algebraic solution to analyze irreversible two-step inhibitors violating the rapid equilibrium approximation is available as a preprint (Kuzmič, 2020a).



It is crucial to have linear product formation in the uninhibited control ( $F^{\text{ctrl}}$ ), as progress curve fitting for time-dependent (ir)reversible inhibition relies on the assumption that uninhibited product formation is absolutely linear. This ideal situation is often not feasible to achieve experimentally, as there are many factors contributing to a slight time-dependent decrease of product formation velocity in the uninhibited control, and not all of them are resolvable (common troubleshooting options are listed in Table 3 in the troubleshooting section near the end of the article). It is possible to correct algebraically. Algebraic correction for nonlinearity in the uninhibited control  $k_{\text{ctrl}}$  caused by spontaneous enzyme degradation/denaturation is possible for irreversible inhibitors (*Data Analysis 1A-B*). Furthermore, it is also possible to perform an algebraic correction for substrate depletion for two-step irreversible inhibitors (*Data Analysis 1D*) (Kuzmič, Solowiej, & Murray, 2015). Ultimately, numerical integration is the preferred method in complex systems where multiple events contribute to the observed nonlinearity.

### Progress Curve Analysis of Substrate Association Competition

The protocol below provides a generic set of steps to accomplishing this type of measurement. A practical example with specific reagents, and assay conditions for progress curve analysis of covalent Cathepsin K inhibitors can be found in Mons et al. (2019).

#### Materials

- 1 × Assay/reaction buffer supplemented with co-factors and reducing agent
- Active enzyme, 4 × solution in assay buffer
- Substrate with continuous read-out, 4 × solution in assay buffer
- Positive control: vehicle/solvent as DMSO stock, or 2% solution in assay buffer
- Negative control: known inhibitor or alkylating agent as DMSO stock, or 2 × solution in assay buffer
- Inhibitor: as DMSO stock, or serial dilution of 2 × solution in assay buffer with 2% DMSO
- 384-well low volume microplate with nonbinding surface (e.g., Corning 3820 or 4513) for incubation and read-out
- Optical clear cover/seal (e.g., Perkin Elmer TopSeal-A Plus, #6050185, Corning 6575 Universal Optical Sealing Tape or Duck Brand HP260 Packing Tape)
- 1.5 ml (Eppendorf) microtubes to prepare stock solutions
- Optional:* 96-well microplate to prepare serial dilution of inhibitor concentration
- Microplate reader equipped with appropriate filters to detect product formation (e.g., CLARIOstar microplate reader)
- Optional:* Automated (acoustic) dispenser (e.g., Labcyte ECHO 550 Liquid Handler acoustic dispenser)

*Before you start*, optimize assay conditions in the uninhibited control to ensure compliance with assumptions and restrictions for progress curve analysis—most importantly linear product formation in the uninhibited control for the duration of the experiment ( $k_{\text{ctrl}} = 0$ ) — by activating the enzyme before reaction initiation (e.g., preincubation with reducing agent for proteases, or ATP for kinases and ligases), testing the enzyme activity on the (fluorogenic) substrate in absence of inhibitor, and adjusting the enzyme and substrate concentration ( $[S]_0 > 10[E]_0$ ) to reach maximum 10% substrate conversion at the end of the measurement window ( $[P]_t < 0.1[S]_0$ ). Further optimization typically involves tuning the reader settings for optimal sensitivity, measurement of a calibration curve for product concentration (Dharadhar et al., 2019; Janssen et al., 2019), and calculation of the  $Z'$ -score from the uninhibited and inhibited controls (ideally 8 replicates) in a separate experiment (Zhang, Chung, & Oldenburg, 1999) to validate that enough product is formed for a good signal/noise ratio ( $Z' > 0.5$ ) at the end of the measurement. Consult Table 3 in the troubleshooting section near the end of the article for common optimization and troubleshooting options. The read-out of product formation must be

homogeneous/continuous. Product formation of substrates with a less sensitive read-out (e.g., fluorescence polarization) may generate a relatively low product signal relative to the unprocessed substrate, and substrate depletion is unavoidable to generate a sufficient  $Z'$ -score (Zhang et al., 1999). Algebraic analysis of two-step irreversible inhibition with substrate depletion ( $[P]_t < 0.1[S]_0$ ) can be performed with *Data Analysis Protocol 1D* after completion of *Basic Protocol 1*, steps 2-6.

1. Add inhibitor or control to each well with the uninhibited control for full enzyme activity containing the same volume vehicle/solvent instead of inhibitor (we use DMSO in this protocol). Add a constant volume of serially diluted inhibitor in assay buffer supplemented with DMSO (e.g., 10.2  $\mu$ l of 2 $\times$  solution containing 2% DMSO), or add inhibitor and controls by (acoustic) dispensing of the pure DMSO stocks, with DMSO backfill to a constant volume (e.g., 0.2  $\mu$ l), followed by addition of assay buffer to each well (e.g., 10  $\mu$ l) and gentle shaking (300 rpm) to homogenize the solution.

Typically, measurements are performed in triplicate (or more replicates) with at least 8 inhibitor concentrations. Inhibitor concentrations might need optimization, but a good starting point is  $0.1-10\times IC_{50}$ ; the highest inhibitor concentration should correspond to maximum 90% initial (noncovalent) inhibition ( $v_i > 0.1v^{ctrl}$ ), as it can be difficult to accurately detect the increase from 90% to 100% inhibition.

2. Add substrate in assay buffer to each well (e.g., 5  $\mu$ l of 4 $\times$  solution) and homogenize the solutions by gentle shaking (300 rpm).

The order of substrate or inhibitor addition is not important *per se*, as long as enzyme is the last reagent to be added, and DMSO stocks are added prior to buffered (aqueous) solutions. Optionally, gently centrifuge the plate (1 min at 1000 rpm) to ensure that assay components are not stuck at the top of the well.

3. Add active enzyme in assay buffer to each well (e.g., 5  $\mu$ l of 4 $\times$  solution), with minimal delay between addition to the first and the last well. Optionally, gently centrifuge the plate (1 min at 1000 rpm) if bubbles are formed (especially for buffers containing surfactants), as these will induce assay artifacts, and to ensure assay components are in solution together rather than stuck to the wall at the top of the well.

Manual addition of enzyme solution and physically moving the plate to the plate reader introduces a delay that may slightly affect the accuracy of the measurement, as it can be variable (depending on the total number of wells, distance to the machine and walking pace of the researcher). This should not be significant if the delay is short compared to the total reaction time, but it can affect the outcome in the data analysis when  $t_0$  is actually 1-2 min. One method to monitor the delay between reaction initiation (onset of product formation and inhibition) and the start of product detection in step 6 is evaluation of the Y-intercept values (as discussed in Table 3). Alternatively, enzyme addition with an injector built into the plate reader minimizes the delay between reaction initiation (onset of product formation and inhibition) and starting the measurement.

4. Seal the wells by applying an optical clear cover.

Continuous kinetic measurements are subject to assay artifacts such as drift due to evaporation. In our experience, application of an optical clear cover/seal prior to measurement improves the assay robustness and resolves significant aberrant nonlinearity unrelated to enzyme activity.

5. Measure product formation in microplate reader by detection of the product read-out.

A typical assay measurement window is 60-240 min, with a measurement interval of 1-2 min. The inhibitor-binding reaction does not have to reach completion (100%

inhibition for irreversible inhibitors, equilibrium for reversible inhibitors) within this window, but data will be more reliable when completion is reached before the end of the measurement (Fig. 5B).

- Proceed to Basic Data Analysis Protocols to calculate the appropriate kinetic parameters for each covalent binding mode: *Data Analysis Protocol 1A* for two-step irreversible inhibitors, *Data Analysis Protocol 1B* for one-step irreversible inhibitors, *Data Analysis Protocol 1C* for two-step reversible inhibitors, or *Data Analysis Protocol 1D* for two-step irreversible inhibitors with substrate depletion.

EXP Conditions	Data Analysis Protocol		
	2-step IRREV	1-step IRREV	2-step REV
$k_{\text{ctrl}} = 0$	1A	1B	1C
$k_{\text{degE}} > 0$	1A	1B	–
$[P]_t > 0.1[S]_0$	1D	–	–

Exemplary assay concentrations.

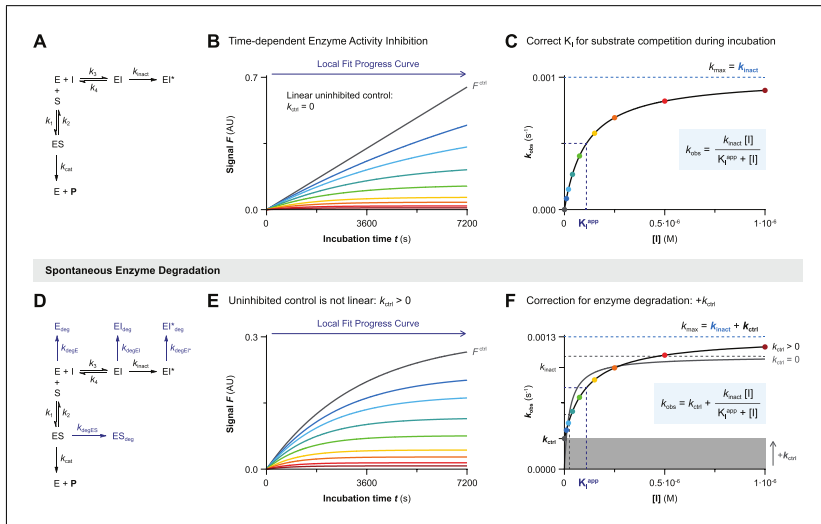
	Concentration during incubation $t$		
	[stock]	V ( $\mu\text{l}$ )	[conc] <sub><math>t</math></sub>
<b>Enzyme</b>	4 nM	5	0.99 nM
<b>Inhibitor</b>	20 nM	10.2	<b>10.10 nM</b>
<b>Substrate</b>	4 $\mu\text{M}$	5	<b>0.99 <math>\mu\text{M}</math></b>
<i>Total</i>		20.2	

### Data Analysis 1A: Progress Curve Analysis for Two-Step Irreversible Covalent Inhibition

The progress curve of time-dependent product formation of each inhibitor concentration is fitted to exponential Equation II (Fig. 5C) constraining final velocity to 100% inhibition ( $v_s = 0$ ) at reaction completion (Fig. 6A and 6B). The inhibitor concentration-dependent observed rate of inactivation  $k_{\text{obs}}$  reflects the rate from initial velocity  $v_i$  (rapid noncovalent equilibrium) to final velocity  $v_s$  (inactivation at reaction completion). The plot of inhibitor concentration-dependent  $k_{\text{obs}}$  reaches maximum rate of inactivation  $k_{\text{inact}}$  in the presence of saturating inhibitor concentration ( $[I] \gg K_i^{\text{app}}$ ) with the Y-intercept at 0 when the progress curve in absence of inhibitor is strictly linear (Fig. 6C). Importantly, the inhibitor concentration that results in half-maximum enzyme inactivation ( $k_{\text{obs}} = \frac{1}{2} \times k_{\text{inact}}$ ) has to be corrected for competition by the substrate during incubation but maximum rate of inactivation  $k_{\text{inact}}$  is unaffected.

#### Warnings and remarks

A linear plot of inhibitor concentration-dependent  $k_{\text{obs}}$  (with Y-intercept =  $k_{\text{ctrl}}$ ) and an initial velocity independent of inhibitor concentration ( $v_i = v^{\text{ctrl}}$ ) are indicative of a one-step binding mechanism: the inhibitor concentration is not saturating ( $[I] \leq 0.1K_i^{\text{app}}$  and  $[I] \leq 0.1K_i^{\text{app}}$ ). This can be resolved by increasing the inhibitor concentration, reducing the substrate concentration, or processing the data with *Basic Data Analysis Protocol 1B*. Inhibitors with a high noncovalent potency ( $[I] \gg K_i^{\text{app}}$ ) might exhibit tight-binding behavior: complete inactivation is reached at reaction initiation ( $v_i = 0$ ), even at the lowest inhibitor concentration, without violating the pseudo-first order reaction conditions ( $[I]_0 \geq 10[E]_0$ ). This can be resolved by lowering the inhibitor concentration, but only if the assay robustness is sufficient to also lower the enzyme concentration, and/or by increasing the concentration of competing substrate, thus increasing the apparent inhibition constant  $K_i^{\text{app}}$ . Unfortunately, algebraic correction for progress curve analysis of one-step inhibitors (Copeland, 2013b) with inhibitor depletion ( $[I]_0 < 10[E]_0$ ) is not



**Figure 6** Data Analysis 1A: Progress curve analysis for two-step irreversible covalent inhibition. Simulated with **KinGen** (A-C) or **KinDeg** (D-F) for inhibitor **C** with 1 pM enzyme and 100 nM substrate **S1**. (A) Schematic enzyme dynamics during incubation for two-step irreversible covalent inhibition. (B) Time-dependent product formation in absence of inhibitor  $F^{ctrl}$  or in presence of inhibitor. The progress curve for each inhibitor concentration is fitted individually to Equation II (Fig. 5C) (constraining  $v_s = 0$ ) to obtain the observed rate of inactivation  $k_{obs}$ . (C) Inhibitor concentration-dependent  $k_{obs}$  reaches  $k_{inact}$  at saturating inhibitor concentration ( $k_{max} = k_{inact}$ ). Half-maximum  $k_{obs} = \frac{1}{2}k_{inact}$  is reached when inhibitor concentration equals the apparent inactivation constant  $K_I^{app}$ . (D) Schematic enzyme dynamics during incubation for two-step irreversible covalent inhibition with spontaneous loss of enzyme activity. Simulated with  $k_{degE} = k_{degES} = k_{degEI} = 0.0003 \text{ s}^{-1}$ . (E) Time-dependent product formation in absence of inhibitor  $F^{ctrl}$  is not linear because  $k_{ctrl} > 0$ . The progress curves for each inhibitor concentration and uninhibited control are fitted individually to Equation II (Fig. 5C) (constraining  $v_s = 0$ ) to obtain the observed rates of inactivation  $k_{obs}$ . (F) Inhibitor concentration-dependent  $k_{obs}$  with spontaneous enzyme degradation increases with  $k_{ctrl}$ , but the span from  $k_{min}$  ( $= k_{ctrl}$ ) to  $k_{max}$  ( $= k_{inact} + k_{ctrl}$ ) still equals  $k_{inact}$ . Fit with algebraic correction for nonlinearity (black line,  $k_{ctrl} > 0$ ). Ignoring the nonlinearity (gray line, constrain  $k_{ctrl} = 0$ ) results in underestimation of  $K_I^{app}$  (overestimation of potency) and overestimation of  $k_{inact}$ .

compatible with two-step inhibition. Numeric fitting is a possibility to fit progress curves with depletion of substrate as well as inhibitor (Kuzmič, 2015). Alternatively, tight-binding two-step irreversible covalent inhibition can be assessed with *Method IV* if covalent adduct formation is relatively slow.

Spontaneous enzyme degradation/denaturation causes a nonlinearity in the uninhibited control ( $k_{ctrl} > 0$ ) that violates the assumption that time-dependence in the inhibitor-treated samples is a direct effect of the inhibitor (Fig. 6D and 6E). The first-order enzymatic degradation rate contributes to  $k_{obs}$  independent of inhibitor concentration ( $k_{degE} = k_{degES} = k_{degEI}$ ). Consequently, the Y-intercept of the  $k_{obs}$  against inhibitor concentration plot now corresponds to observed rate  $k_{ctrl}$  in absence of inhibitor, and  $k_{max}$  is higher ( $k_{max} = k_{inact} + k_{ctrl}$ ) (Fig. 6F). Performing a simple algebraic correction for the observed nonlinearity due to spontaneous enzyme degradation results in good estimates for  $k_{inact}$  and  $K_I^{app}$  (Fig. 6F). Ignoring the nonlinearity in the uninhibited control by restraining  $k_{ctrl} = 0$  implies that all time-dependent loss of enzyme activity should be attributed to inhibitor-mediated inactivation, resulting in an underestimation of inactivation constant  $K_I^{app}$  (overestimation of potency) and overestimation of  $k_{inact}$ . This effect is less pronounced when spontaneous enzyme degradation is much slower than the maximum rate of covalent adduct formation ( $k_{inact} \gg k_{ctrl}$ ). It is important to note that stabilization

of the enzyme species by (noncovalent) inhibitor binding also decreases the contribution of  $k_{\text{ctrl}}$  to the observed rate  $k_{\text{obs}}$  at saturating inhibitor concentrations ( $k_{\text{max}} = k_{\text{inact}}$ ). This impairs the accuracy of the algebraic correction unless  $k_{\text{ctrl}}$  is relatively small ( $k_{\text{max}}$  approaches  $k_{\text{inact}}$  if  $k_{\text{inact}} \gg k_{\text{ctrl}}$ ).

This algebraic correction does not accurately correct for nonlinearity due to substrate depletion ( $[P]_t > 0.1[S]_0$ ): substrate depletion is dependent on the total product formation and does not (significantly) contribute to  $k_{\text{max}}$  at saturating inhibitor concentration because enzyme inhibition reduces the total amount of product formed ( $k_{\text{max}} = k_{\text{inact}}$ ). Please consult *Data Analysis 1D* for algebraic correction of nonlinearity due to substrate depletion.

## Two-Step Irreversible Covalent Inhibition

Processing of raw data obtained with *Basic Protocol 1* for two-step irreversible covalent inhibitors.

1. Plot signal  $F$  against incubation time  $t$ .

Plot signal (in AU) on the Y-axis against incubation time (in s) on the X-axis for each inhibitor concentration and the controls (Fig. 6B). Product formation in the uninhibited control  $F^{\text{ctrl}}$  should be linear. Consult Table 3 for troubleshooting of nonlinearity of the uninhibited control. Optionally, perform background correction to correct for assay artifacts such as bleaching and drift that cause a negative final velocity ( $v_s < 0$  AU/s) in the fully inhibited control. This correction can be subtraction of the background in presence of substrate (and inhibitor) but absence of enzyme, or subtraction of the fully inhibited control.

2. Fit signal  $F_t$  against  $t$  to obtain  $k_{\text{obs}}$

Fit signal  $F_t$  against incubation time  $t$  to Equation II (Fig. 6B/E). Constrain final velocity  $v_s = 0$  (in AU/s) for background-corrected product formation, or  $v_s =$  value for full inhibition control. A lack of initial noncovalent complex ( $v_i = v^{\text{ctrl}}$ ) is indicative of one-step binding behavior.

$$F_t = v_s t + \frac{v_i - v_s}{k_{\text{obs}}} [1 - e^{-k_{\text{obs}} t}] + F_0$$

Equation II

Equation II for nonlinear regression of user-defined explicit equation  $Y = (v_s * X) + ((v_i - v_s) / k_{\text{obs}}) * (1 - \text{EXP}(-k_{\text{obs}} * X)) + Y_0$  with  $Y =$  signal  $F_t$  (in AU) and  $X =$  incubation time  $t$  (in s) to find  $Y_0 =$  Y-intercept  $F_0 =$  background signal at  $t = 0$  (in AU),  $v_i =$  initial slope  $v_i$  (in AU/s),  $v_s =$  final slope  $v_s$  (in AU/s) and  $k_{\text{obs}} =$  observed reaction rate  $k_{\text{obs}}$  (in  $s^{-1}$ ).

3. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $s^{-1}$ ) on the Y-axis against inhibitor concentration (in M) after reaction initiation by enzyme addition (in the final solution) on the X-axis (Fig. 6C/F). The plot of  $k_{\text{obs}}$  against  $[I]$  should reach a maximum  $k_{\text{obs}}$  at saturating inhibitor concentration. Note that a linear curve is indicative of one-step binding behavior at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_1^{\text{app}}$  in Fig. 3F) with  $v_i = v^{\text{ctrl}}$  (low initial inhibition). Proceed to *Basic Data Analysis Protocol 1B*, step 4, after it has been validated that the linear curve is not resultant from saturating inhibitor concentrations ( $[I] \gg 10K_1^{\text{app}}$  in Fig. 3G) as identified by  $v_i \ll v^{\text{ctrl}}$  (significant initial inhibition), by repeating the measurement with a higher competitive substrate concentration (increase  $K_1^{\text{app}}$ ) and/or lower inhibitor concentration.

4. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{inact}}$  and  $K_{\text{I}}^{\text{app}}$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration to Equation VII to obtain maximum inactivation rate constant  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and apparent inactivation constant  $K_{\text{I}}^{\text{app}}$  (in M). Constrain  $k_{\text{ctrl}} = k_{\text{obs}}$  of the uninhibited control (Fig. 6F). Calculate inactivation constant  $K_{\text{I}}$  (in M) and irreversible covalent inhibitor potency  $k_{\text{inact}}/K_{\text{I}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) with *Sample Calculation 1 & 2*.

$$k_{\text{obs}} = k_{\text{ctrl}} + \frac{k_{\text{inact}} [I]}{K_{\text{I}}^{\text{app}} + [I]}$$

#### Equation VII

Equation VII for nonlinear regression of user-defined explicit equation  $Y = Y_0 + ((k_{\text{max}} * X) / (K_{\text{I}}^{\text{app}} + X))$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) and  $X =$  inhibitor concentration (in M) to find  $Y_0 =$  rate of nonlinearity in uninhibited control  $k_{\text{ctrl}}$  (in  $\text{s}^{-1}$ ),  $k_{\text{max}} =$  maximum reaction rate  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and  $K_{\text{I}}^{\text{app}} =$  Apparent inactivation constant  $K_{\text{I}}^{\text{app}}$  (in M).

5. EXTRA: Plot and fit  $v_i$  against  $[I]$  to obtain  $K_{\text{I}}^{\text{app}}$ .

Inhibition constant  $K_{\text{I}}$  can be calculated from the initial velocity  $v_i$  (obtained in step 3), reflecting the rapid (initial) noncovalent enzyme-inhibitor equilibrium. Plot the mean and standard deviation of  $v_i$  (in AU/s) on the Y-axis against inhibitor concentration on the X-axis (similar to Fig. 8D). Fit  $v_i$  against  $[I]$  to four-parameter nonlinear regression Hill Equation VIII (Copeland, 2013e) to obtain apparent inhibition constant  $K_{\text{I}}^{\text{app}}$  (in M). Constrain the top to the uninhibited  $v_i$  (maximum velocity =  $v_i^{\text{ctrl}}$ ) and the bottom to the fully inhibited  $v_i$  ( $v_i^{\text{min}} =$  minimum velocity). For (background-)corrected product formation  $v_i^{\text{min}} = 0$ ). Calculate inhibition constant  $K_{\text{I}}$  (in M) with *Sample Calculation 3*.

$$v_i = v_i^{\text{min}} + \frac{v_i^{\text{ctrl}} - v_i^{\text{min}}}{1 + \left(\frac{[I]}{K_{\text{I}}^{\text{app}}}\right)^h}$$

#### Equation VIII

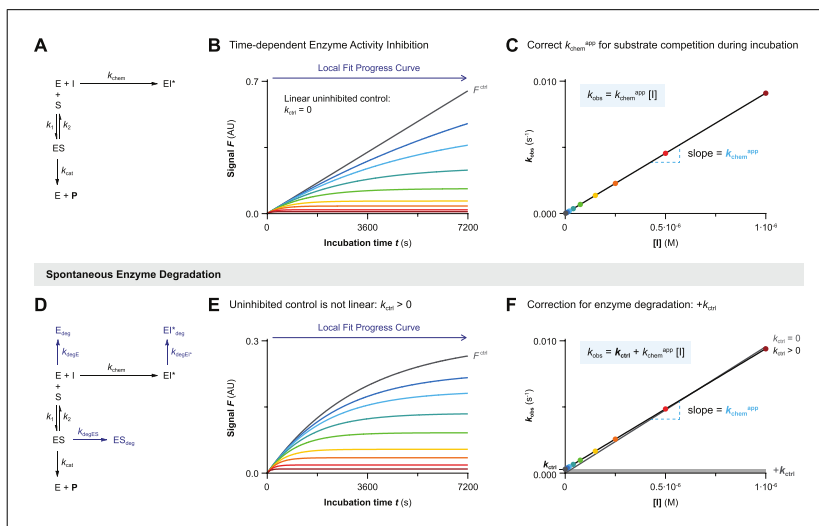
Equation VIII for nonlinear regression of four-parameter dose-response equation  $Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + (X/\text{IC}_{50})^{\text{HillSlope}})$  with  $Y =$  initial product formation velocity  $v_i$  (in AU/s),  $X =$  inhibitor concentration (in M),  $\text{Bottom} =$  velocity in fully inhibited control  $v_i^{\text{min}}$  (in AU/s), and  $\text{Top} =$  maximum velocity in uninhibited control  $v_i^{\text{ctrl}}$  (in AU/s) to find  $\text{HillSlope} =$  Hill coefficient  $h$  (unitless) and  $\text{IC}_{50} =$  apparent inhibition constant  $K_{\text{I}}^{\text{app}}$  (in M).

6. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental progress curves to the progress curves simulated with scripts **KinGen** and **KinDeg** (using experimental rate constant  $k_{\text{inact}} = k_5$ ) to confirm that the calculated kinetic constants are in accordance with the experimental data.

### Data Analysis 1B: Progress Curve Analysis for One-Step Irreversible Covalent Inhibition

The progress curve of time-dependent product formation of each inhibitor concentration is fitted to exponential Equation II (Fig. 5C) constraining final velocity to inactivation ( $v_s = 0$ ) at reaction completion (Fig. 7A and 7B). The initial velocity  $v_i$  equals the uninhibited product formation velocity ( $v_i = v_i^{\text{ctrl}}$ ), as noncovalent inhibitor binding does not contribute to enzyme inhibition by one-step irreversible inhibitors. A linear plot of inhibitor concentration-dependent  $k_{\text{obs}}$  is indicative of a one-step binding mechanism with  $k_{\text{chem}}^{\text{app}}$  as the slope (Fig. 7C). Two-step irreversible covalent inhibitors also have a linear



**Figure 7** Data Analysis 1B: Progress curve analysis for one-step irreversible covalent inhibition. Simulated with **KinGen** (A-C) or **KinDeg** (D-F) for inhibitor **D** with 1 pM enzyme and 100 nM substrate **S1**. (A) Schematic enzyme dynamics during incubation for one-step irreversible covalent inhibition. (B) Time-dependent product formation in absence of inhibitor  $F^{\text{ctrl}}$  or in presence of inhibitor. The progress curve for each inhibitor concentration is fitted individually to Equation II (Fig. 5C) (constraining  $v_s = 0$ ) to obtain the observed rate of inactivation  $k_{\text{obs}}$ .  $v_i = v^{\text{ctrl}}$  for one-step irreversible inhibitors and two-step irreversible inhibitors at non-saturating concentrations ( $[I] \ll K_1^{\text{app}}$ ). (C) Inhibitor concentration-dependent  $k_{\text{obs}}$  increases linearly with inhibitor concentration, with  $k_{\text{chem}}^{\text{app}}$  as the slope. (D) Schematic enzyme dynamics during incubation for one-step irreversible covalent inhibition with spontaneous loss of enzyme activity. Simulated with  $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}} = 0.0003 \text{ s}^{-1}$ . (E) Time-dependent product formation in absence of inhibitor  $F^{\text{ctrl}}$  is not linear because  $k_{\text{ctrl}} > 0$ . The progress curves for each inhibitor concentration and uninhibited control are fitted individually to Equation II (Fig. 5C) (constraining  $v_s = 0$ ) to obtain the observed rates of inactivation  $k_{\text{obs}}$ . (F) Inhibitor concentration-dependent  $k_{\text{obs}}$  with spontaneous enzyme degradation/denaturation increases by  $k_{\text{ctrl}}$ . Fit with algebraic correction for nonlinearity (black line,  $k_{\text{ctrl}} = 0$ ) or ignoring nonlinearity (gray line, constrain  $k_{\text{ctrl}} = 0$ ). Ignoring the nonlinearity (assuming Y-intercept = 0) results in overestimation of  $k_{\text{chem}}^{\text{app}}$  (steeper slope).

$k_{\text{obs}}$  against inhibitor concentration plot at non-saturating concentrations ( $[I] \leq 0.1K_1^{\text{app}}$ ) with  $k_{\text{chem}}^{\text{app}} = k_{\text{inact}}/K_1^{\text{app}}$ .

### Warnings and remarks

The slope has to be corrected for substrate competition to obtain the inactivation constant  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ). Substrate will occupy a fraction of the unbound enzyme to reach the noncovalent  $\text{E} + \text{S} \rightleftharpoons \text{ES}$  equilibrium (how much depends on  $[\text{S}]/K_M$ ), thus reducing the unbound enzyme concentration. It may seem counterintuitive to correct for substrate competition, as the pseudo-first-order rate of covalent adduct formation ( $k_{\text{obs}} = k_{\text{chem}}^{\text{app}}[\text{I}]$ ) does not seem to involve unbound enzyme (provided inhibitor is present in large excess), but formation of  $\text{EI}^*$  is limited by the available unbound enzyme at that moment and it is not possible to form covalent adduct  $\text{EI}^*$  when competing substrate blocks access to the enzyme active site.

It is important to have linear product formation in the uninhibited control ( $k_{\text{ctrl}} = 0$ ) or to perform an algebraic correction for nonlinearity in the uninhibited control ( $k_{\text{ctrl}} > 0$ ) caused by spontaneous first-order enzyme degradation/denaturation (Fig. 7D-F). Failure to correct for the contribution of enzyme degradation when fitting the observed rate of inactivation  $k_{\text{obs}}$  against inhibitor results in overestimation of  $k_{\text{chem}}^{\text{app}}$  (Fig. 7F, gray line). The contribution of nonlinearity  $k_{\text{ctrl}}$  becomes less pronounced at elevated inhibitor

concentrations as  $k_{\text{ctrl}}$  becomes significantly smaller than  $k_{\text{obs}}$  ( $k_{\text{ctrl}} \ll k_{\text{chem}}^{\text{app}}[\text{I}]$ ). (De)stabilization of enzyme upon inhibitor binding ( $k_{\text{degEI}^*}$ ) does not affect  $k_{\text{obs}}$ , as EI\* formation is already irreversible, thus removing the species from the available pool of catalytic enzyme. To our knowledge, methods to algebraically correct for substrate depletion have not been reported.

### One-Step Irreversible Covalent Inhibition

Processing of raw data obtained with *Basic Protocol 1* for one-step irreversible covalent inhibitors and two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[\text{I}] \leq 0.1K_{\text{i}}^{\text{app}}$ ).

1. Plot signal  $F$  against incubation time  $t$ .

Plot signal (in AU) on the Y-axis against incubation time (in s) on the X-axis for each inhibitor concentration and the controls (Fig. 7B). Product formation in the uninhibited control  $F^{\text{ctrl}}$  should be linear. Consult Table 3 for troubleshooting of nonlinearity of the uninhibited control. Optionally, perform background correction to correct for assay artifacts such as bleaching and drift that cause a negative final velocity ( $v_s < 0$  AU/s) in the fully inhibited control. This correction can be subtraction of the background in presence of substrate (and inhibitor) but absence of enzyme, or subtraction of the fully inhibited control.

2. Fit  $F_t$  against  $t$  to obtain  $k_{\text{obs}}$ .

Fit signal  $F_t$  against incubation time  $t$  to Equation II (Fig. 7B/E). Constrain final velocity  $v_s = 0$  (in AU/s) for background-corrected product formation, or  $v_s =$  value for full inhibition control. Initial velocity  $v_i$  should be a shared value because noncovalent inhibition does not significantly contribute to the initial inhibition for inhibitors displaying one-step behavior.

$$F_t = v_s t + \frac{v_i - v_s}{k_{\text{obs}}} [1 - e^{-k_{\text{obs}} t}] + F_0$$

#### Equation II

Equation II for nonlinear regression of user-defined explicit equation  $Y = (v_s * X) + ((v_i - v_s) / k_{\text{obs}}) * (1 - \text{EXP}(-k_{\text{obs}} * X)) + Y_0$  with  $Y =$  signal  $F_t$  (in AU) and  $X =$  incubation time  $t$  (in s) to find  $Y_0 = Y$ -intercept  $F_0 =$  background signal at  $t = 0$  (in AU),  $v_i =$  initial slope (in AU/s),  $v_s =$  final slope (in AU/s) and  $k_{\text{obs}} =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

3. Plot  $k_{\text{obs}}$  against  $[\text{I}]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) after reaction initiation by enzyme addition (in the final solution) on the X-axis (Fig. 7B/E). The plot of  $k_{\text{obs}}$  against inhibitor concentration  $[\text{I}]$  is linear for one-step irreversible inhibitors and for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[\text{I}] \ll 0.1K_{\text{i}}^{\text{app}}$ ).

4. Fit  $k_{\text{obs}}$  against  $[\text{I}]$  to obtain  $k_{\text{chem}}^{\text{app}}$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration to Equation IX to obtain apparent inhibitor potency  $k_{\text{chem}}^{\text{app}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) from the linear slope. Constrain Y-intercept  $k_{\text{ctrl}} = k_{\text{obs}}$  of the uninhibited control (Fig. 7F). Calculate  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) reflecting inhibitor potency for one-step irreversible covalent inhibition with *Sample Calculation 4*. Calculate  $k_{\text{inact}}/K_{\text{i}}^{\text{app}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) and  $k_{\text{inact}}/K_{\text{i}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[\text{I}] \leq 0.1K_{\text{i}}^{\text{app}}$ ) with *Sample Calculation 5 and 6*.

$$k_{\text{obs}} = k_{\text{ctrl}} + k_{\text{chem}}^{\text{app}} [\text{I}]$$

#### Equation IX



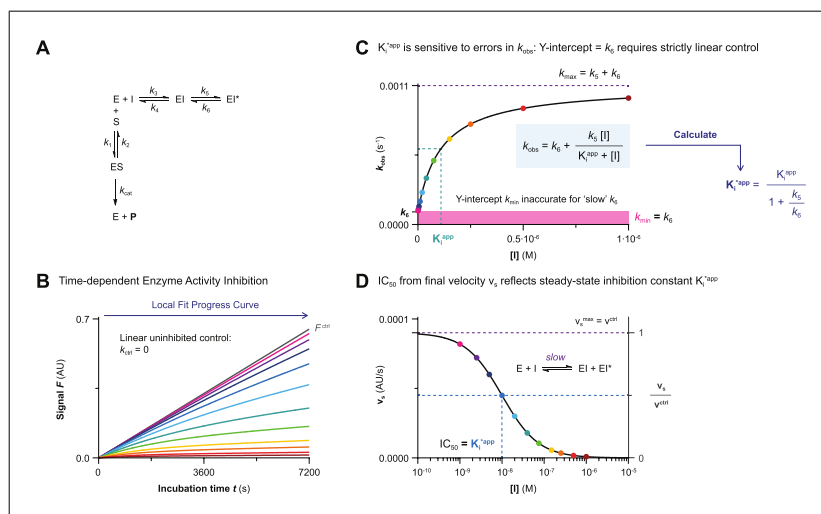
Equation IX for nonlinear regression of straight line  $Y = Y_{\text{Intercept}} + \text{Slope} \cdot X$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) and  $X =$  inhibitor concentration (in  $\text{M}$ ) to find  $Y_{\text{Intercept}} =$  rate of nonlinearity in uninhibited control  $k_{\text{ctrl}}$  (in  $\text{s}^{-1}$ ) and  $\text{Slope} =$  apparent inactivation rate constant  $k_{\text{chem}}^{\text{app}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ).

5. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

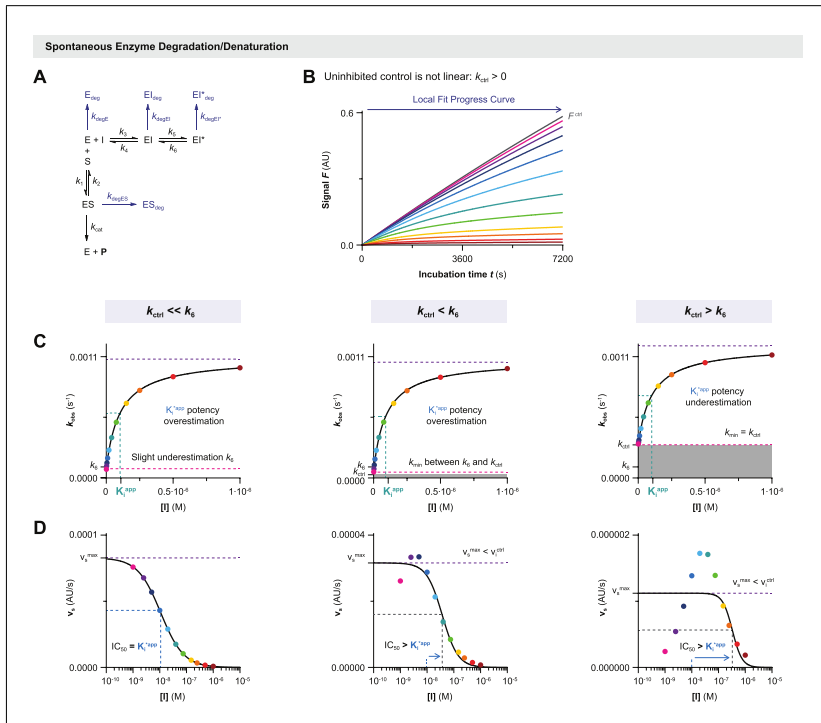
Proceed to *Kinetic Simulations 1* to compare the experimental progress curves to the progress curves simulated with scripts **KinGen** and **KinDeg** (using experimental rate constant  $k_{\text{chem}} = k_3$ ) to confirm that the calculated kinetic constants are in accordance with the experimental data.

**Data Analysis 1C: Progress Curve Analysis for Two-Step Reversible Covalent Inhibition**

The progress curve of time-dependent product formation of each inhibitor concentration (Fig. 8A and 8B) is fitted to exponential Equation II (Fig. 5C). The inhibitor concentration-dependent observed rate for reaction completion  $k_{\text{obs}}$  reflects the rate from initial velocity  $v_i$  (rapid noncovalent equilibrium) to final velocity  $v_s$  (slow steady-state equilibrium). Contrary to irreversible inhibition, steady-state velocity  $v_s$  is not constrained to inactivation ( $v_s > 0$ ) because the reversible steady-state equilibrium is reached at reaction completion. Maximum rate of reaction completion  $k_{\text{max}}$  is reached in the presence of saturating inhibitor concentration ( $[I] \gg K_i^{\text{app}}$ ), and the covalent association rate constant  $k_5$  is obtained from the span between  $k_{\text{min}}$  and  $k_{\text{max}}$ . Interestingly, the Y-intercept  $k_{\text{min}}$  is equal to covalent dissociation rate constant  $k_6$ ; therefore, the  $k_{\text{obs}}$  of uninhibited control ( $k_{\text{ctrl}}$ ) is excluded from the fit (Fig. 8C).



**Figure 8** Data Analysis 1C: Progress curve analysis for two-step reversible covalent inhibition. Simulated with **KinGen** for inhibitor **B** with 1 pM enzyme and 100 nM substrate **S1**. **(A)** Schematic enzyme dynamics during incubation for two-step reversible covalent inhibition. **(B)** Time-dependent product formation in absence of inhibitor  $F^{\text{ctrl}}$  or in presence of inhibitor. The progress curve for each inhibitor concentration is fitted individually to Equation II (Fig. 5C) to obtain the observed rate of inactivation  $k_{\text{obs}}$  and steady-state velocity  $v_s$ . **(C)** Inhibitor concentration-dependent  $k_{\text{obs}}$  equals  $k_{\text{max}}$  at saturating inhibitor concentration ( $k_{\text{max}} = k_5 + k_6$ ) and approaches  $k_6$  in absence of inhibitor ( $k_{\text{min}} = k_6$ ). Half-maximum  $k_{\text{obs}} = k_{\text{min}} + \frac{1}{2}(k_{\text{max}} - k_{\text{min}}) = k_6 + \frac{1}{2}k_5$  is reached when inhibitor concentration equals the apparent inhibition constant  $K_i^{\text{app}}$ . Steady-state inhibition constant  $K_i^{\text{app}}$  has to be calculated from the fitted values of  $k_5$ ,  $k_6$  and  $K_i^{\text{app}}$ , thus being very sensitive to errors and (non)linearity in the uninhibited background (illustrated in Fig. 9). **(D)** Steady-state inhibition constant  $K_i^{\text{app}}$  is equal to the  $\text{IC}_{50}$  of steady-state velocity  $v_s$ .



**Figure 9** Data Analysis 1C: Progress curve analysis for two-step reversible covalent inhibition is not compatible with spontaneous enzyme degradation/denaturation. Simulated with **KinDeg** for inhibitor **B** with 1 pM enzyme and 100 nM substrate **S1**. **(A)** Schematic enzyme dynamics during incubation for two-step reversible covalent inhibition with spontaneous loss of enzyme activity due to degradation/denaturation. **(B)** Time-dependent product formation in absence of inhibitor  $F^{ctrl}$  is not linear because  $k_{ctrl} > 0$ . The progress curve for each inhibitor concentration is fitted individually to Equation II (Fig. 5C) to obtain the observed rate of inactivation  $k_{obs}$  and steady-state velocity  $v_s$ . Simulated for  $k_{ctrl} = 0.00003 \text{ s}^{-1}$ . **(C)** Inhibitor concentration–dependent  $k_{obs}$  is driven by spontaneous enzyme degradation at low inhibitor concentrations, thus lowering the Y-intercept ( $k_{min}$  approaches  $k_{ctrl}$ ). Ignoring the nonlinearity in the uninhibited control  $k_{ctrl}$  results in poor fits with underestimation of  $k_6$  even if  $k_{ctrl}$  is slower than  $k_6$ . Simulated for  $k_{ctrl} = k_{degE} = k_{degES} = k_{degEI} = k_{degEI^*}$  with  $k_{ctrl} = 0.00003 \text{ s}^{-1}$  (left),  $k_{ctrl} = 0.00003 \text{ s}^{-1}$  (middle) and  $k_{ctrl} = 0.0003 \text{ s}^{-1}$  (right). **(D)** Final velocity  $v_s$  has been ‘contaminated’ by the contribution of irreversible inactivation to the time-dependent inhibition, and approaches  $v_s = 0$  at low inhibitor concentrations. Final velocity  $v_s$  no longer reflects the steady-state equilibrium:  $IC_{50}$  is larger than  $K_i^{*app}$  (underestimation of steady-state potency) unless  $k_{ctrl}$  is much smaller than  $k_6$ .

Steady-state inhibition constant  $K_i^{*app}$  can be calculated from the fitted values of  $K_i^{app}$ ,  $k_5$ , and  $k_6$ , but this is not the preferred approach, as a small error in  $k_6$  has huge implications for the calculation of  $K_i^*$ . Other methods such as jump dilution assays generate more reliable estimates of  $k_6$ , which is especially important for very potent two-step reversible covalent inhibitors: relatively small  $k_6$ -values cannot accurately be estimated from the Y-intercept (Copeland, 2013e; Copeland et al., 2011). Generally, more reliable estimates of the apparent steady-state inhibition constant  $K_i^{*app}$  are generated from the dose-response curve of steady-state velocity  $v_s$  against inhibitor concentration (Fig. 8D).

### Warnings and remarks

It is crucial to have strictly linear product formation in the uninhibited control ( $k_{\text{ctrl}} = 0$ ) because it is not possible to perform an algebraic correction for spontaneous enzyme degradation/denaturation (Fig. 9A). Unfortunately, potent reversible covalent inhibitors are likely to violate this condition. Contrary to irreversible covalent inhibitors that become more potent with a faster  $k_{\text{inact}}$ , reversible covalent inhibitors are more potent if they have a longer residence time  $\tau$ , which is driven by a slow dissociation rate  $k_6$  (Fig. 1B) (Copeland, 2010; Copeland et al., 2006). Violation of this assumption ( $k_{\text{ctrl}} > 0$ ) can be identified by fitting the uninhibited product formation  $F^{\text{ctrl}}$  to Equation II (Fig. 5C): initial velocity  $v_i^{\text{ctrl}}$  should not be larger than steady-state  $v_s^{\text{ctrl}}$ . The consequence of nonlinearity in the uninhibited control is ‘contamination’ of reaction rate  $k_{\text{obs}}$  and final velocity  $v_s$  (based on the reversible reaction to reach steady-state equilibrium:  $v_s > 0$ ) with the rate of enzyme degradation  $k_{\text{ctrl}}$  (based on an inactivation reaction:  $v_s = 0$ ). Y-intercept approaching  $k_{\text{ctrl}}$  instead of  $k_6$  even though the uninhibited control is not included in the fit is an indication that spontaneous enzyme degradation dominates  $k_{\text{obs}}$  at low inhibitor concentrations (Fig. 9C). This ‘red flag’ should not be ignored, as it will result in over/underestimation of kinetic parameters. To our knowledge, models to perform an algebraic correction have not been reported. Calculating steady-state inhibition constant  $K_i^*$  from final velocity  $v_s$  also results in an underestimation of the steady-state potency because the contribution of spontaneous enzyme degradation to final velocity  $v_s$  is dominant at low inhibitor concentrations (Fig. 9D). Underestimation of the steady-state potency of reversible covalent inhibitors that have a relatively slow  $k_6$  is more severe than for the less potent counterpart with a faster  $k_6$ . We were able to find reasonable estimates of  $K_i^*$  when the contribution of nonlinearity was significantly smaller than covalent adduct dissociation ( $k_{\text{ctrl}} \ll k_6$ ). Preincubation time-dependent inhibition (*Method III*) is a more suitable method to analyze two-step reversible inhibition affected by enzyme instability: it is possible to algebraically correct for enzyme instability in this method (*Data Analysis 3C*).

### Two-Step Reversible Covalent Inhibition

Processing of raw data obtained with *Basic Protocol 1* for two-step reversible covalent inhibitors.

1. Plot signal  $F$  against incubation time  $t$ .

Plot signal (in AU) on the Y-axis against incubation time (in s) on the X-axis for each inhibitor concentration and the controls (Fig. 8B). Product formation in the uninhibited control  $F^{\text{ctrl}}$  should be linear. Consult Table 3 for troubleshooting of nonlinearity of the uninhibited control. Optionally, perform background correction to correct for assay artifacts such as bleaching and drift that cause a negative final velocity ( $v_s < 0$  AU/s) in the fully inhibited control. This correction can be subtraction of the background in the presence of substrate (and inhibitor) but absence of enzyme, or subtraction of the fully inhibited control.

2. Fit  $F_t$  against  $t$  to obtain  $k_{\text{obs}}$  and  $v_s$ .  
Fit signal  $F_t$  against incubation time  $t$  to Equation II (Fig. 8B) to obtain final product formation velocity  $v_s$  (in AU/s) and the observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial equilibrium  $v_i$  to steady-state equilibrium  $v_s$ . Do not constrain initial velocity  $v_i$  or final velocity  $v_s$ . Also fit the progress curve of the uninhibited control ( $F^{\text{ctrl}}$ ) to validate that product formation is strictly linear ( $v_i^{\text{ctrl}} = v_s^{\text{ctrl}}$ ), because algebraic correction for nonlinearity in the uninhibited control is not possible (Fig. 9). The observed rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) reflects the exponential reaction rate from initial noncovalent equilibrium ( $v_i$ ) to final steady-state equilibrium ( $v_s$ ).

$$F_t = v_s t + \frac{v_i - v_s}{k_{\text{obs}}} [1 - e^{-k_{\text{obs}} t}] + F_0$$

**Equation II**

Equation II for nonlinear regression of user-defined explicit equation  $Y = (v_s * X) + ((v_i - v_s) / k_{\text{obs}}) * (1 - \text{EXP}(-k_{\text{obs}} * X)) + Y_0$  with  $Y = \text{signal } F_t$  (in AU) and  $X = \text{incubation time } t$  (in s) to find  $Y_0 = Y\text{-intercept } F_0 = \text{background signal at } t = 0$  (in AU),  $v_i = \text{initial slope}$  (in AU/s),  $v_s = \text{final slope}$  (in AU/s), and  $k_{\text{obs}} = \text{observed reaction rate } k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

3. Plot and fit  $v_s$  against  $[I]$  to obtain  $K_i^{*app}$ .

Apparent steady-state inhibition constant  $K_i^{*app}$  (in M) can be calculated from the final velocity  $v_s$  (obtained in the previous step) reflecting enzyme activity after reaching the steady-state inhibitor equilibrium (reaction completion). Plot the mean and standard deviation of  $v_s$  (in AU/s) on the Y-axis against inhibitor concentration (in M) on the X-axis and fit to four-parameter nonlinear regression Hill Equation X (Copeland, 2013e) to obtain apparent steady-state inhibition constant  $K_i^{*app}$  (in M) (Fig. 8D). Constrain the top to uninhibited velocity  $v^{ctrl}$  (maximum velocity =  $v_s^{max}$ ) and the bottom to the fully inhibited  $v_s$  ( $v_s^{min}$ , minimum velocity). For (background-) corrected product formation,  $v_s^{min} = 0$ . Accurate values are only obtained when uninhibited product formation is strictly linear ( $k_{ctrl} = 0$ ) or when the rate of spontaneous inactivation  $k_{ctrl}$  is much smaller than the covalent dissociation  $k_6$  (Fig. 9). Validate that  $v_s$  is not driven by spontaneous enzyme degradation ( $k_{ctrl} \ll k_6$ ) by also fitting without constraints for  $v_s^{max}$ . Calculate steady-state inhibition constant  $K_i^*$  (in M) with *Sample Calculation 7*.

$$v_s = v_s^{min} + \frac{v^{ctrl} - v_s^{min}}{1 + \left(\frac{[I]}{K_i^{*app}}\right)^h}$$

**Equation X**

Equation X for nonlinear regression of four-parameter dose-response equation  $Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + (X / \text{IC50})^{\text{HillSlope}})$  with  $Y = \text{final product formation velocity } v_s$  (in AU/s),  $X = \text{inhibitor concentration}$  (in M),  $\text{Bottom} = \text{velocity in fully inhibited control } v_s^{min}$  (in AU/s) and  $\text{Top} = \text{maximum velocity in uninhibited control } v^{ctrl}$  (in AU/s) to find  $\text{HillSlope} = \text{Hill coefficient } h$  (unitless) and  $\text{IC50} = \text{apparent steady-state inhibition constant } K_i^{*app}$  (in M).

4. *Optional*: Plot and fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $K_i^{app}$ ,  $k_5$ , and  $k_6$ .

This is an optional data processing step to obtain kinetic parameters by fitting to the observed rate  $k_{\text{obs}}$  (obtained in *Data Analysis 1C*, step 2), and is used to validate  $K_i^{*app}$  values found in the previous step, to check if nonlinearity in the uninhibited control  $k_{ctrl}$  (in  $\text{s}^{-1}$ ) affects the fit, and/or to generate experimental  $k_5$  and  $k_6$  values to use in kinetic simulations. Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) on the X-axis (Fig. 8C). Exclude  $k_{\text{obs}}$  of uninhibited control ( $k_{ctrl}$ ) from the fit. Fit  $k_{\text{obs}}$  against inhibitor concentration to Equation XI to obtain rate constants for the covalent association  $k_5$  (in  $\text{s}^{-1}$ ) and covalent dissociation  $k_6$  (in  $\text{s}^{-1}$ ), as well as apparent noncovalent inhibition constant  $K_i^{app}$  (in M) reflecting the rapid (initial) noncovalent equilibrium. Use the inhibitor concentration after reaction initiation by enzyme addition (in the final solution). Accurate values are only obtained when uninhibited product formation is strictly linear ( $k_{ctrl} = 0$ ). Y-intercept approaching  $k_{ctrl}$  despite the uninhibited control not being included in the fit is a red flag that should not be ignored, as this is indicative of spontaneous enzyme degradation rather than  $k_6$  dominating  $k_{\text{obs}}$  at low inhibitor concentrations,

for which algebraic corrections are not available (Fig. 9). Calculate noncovalent inhibition constant  $K_i$  (in M) with *Sample Calculation 3* and proceed to calculate steady-state inhibition constant  $K_i^*$  (in M) with *Sample Calculation 8*. Optionally, perform step 6 of *Data Analysis 1A* to obtain apparent noncovalent inhibition constant  $K_i^{\text{app}}$  (in M) from the initial velocity  $v_i$  (obtained in *Data Analysis Protocol 1C* step 2).

$$k_{\text{obs}} = k_6 + \frac{k_5 [I]}{K_i^{\text{app}} + [I]}$$

**Equation XI**

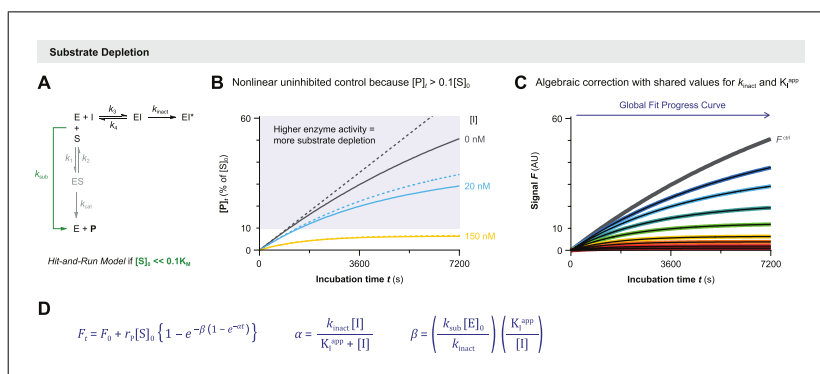
Equation XI for nonlinear regression of user-defined explicit equation  $Y = Y_0 + ((k_{\text{max}} \cdot X) / (K_i^{\text{app}} + X))$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) and  $X =$  inhibitor concentration (in M) to find  $Y_0 =$  covalent dissociation rate constant  $k_6$  (in  $\text{s}^{-1}$ ),  $k_{\text{max}} =$  covalent association rate constant  $k_5$  (in  $\text{s}^{-1}$ ) and  $K_i^{\text{app}} =$  Apparent inhibition constant  $K_i^{\text{app}}$  (in M).

5. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental progress curves to the progress curves simulated with scripts **KinGen** and **KinDeg** to confirm that the calculated kinetic constants are in accordance with the experimental data. Experimental estimates of  $k_5$  and  $k_6$  are generated in the previous step of this protocol.

### Data Analysis 1D: Algebraic Correction for Substrate Depletion in Progress Curve Analysis for Two-Step Irreversible Covalent Inhibition

Scientists from BioKin and Pfizer (Kuzmič et al., 2015) derived an algebraic model for two-step irreversible covalent inhibitors to correct for nonlinearity caused by substrate depletion (Fig. 10A). Substrate depletion causes a nonlinearity in the uninhibited control



**Figure 10** Data Analysis 1D: Algebraic correction for substrate depletion in progress curve analysis for two-step irreversible covalent inhibition. Simulated with **KinSubDpl** for inhibitor **C** with 100 pM enzyme and 10 nM substrate **S1**. **(A)** Enzyme dynamics for two-step irreversible covalent inhibition. Algebraic correction for substrate depletion is restricted to a Hit-and-Run model ( $E + S \rightarrow E + P$ ) for product formation. **(B)** Substrate depletion ( $[P]_t > 0.1[S]_0$ , blue area) results in a decrease of product formation in the uninhibited control (solid line) compared to product formation, assuming substrate conversion does not affect product formation rates (dashed line, simulated with **KinGen**). The contribution of substrate depletion to nonlinearity increases with higher enzyme activity (less inhibition). **(C)** Time-dependent product formation in the absence of inhibitor  $F^{\text{ctrl}}$  or in presence of inhibitor with time-dependent loss of enzyme activity due to substrate depletion. Inhibitor-treated progress curves are globally fitted to Equation III with shared values for  $k_{\text{inact}}$  and  $K_i^{\text{app}}$ . **(D)** Equation III. Algebraic model to correct for substrate depletion at low substrate concentrations (Kuzmič et al., 2015).  $F_0 =$  Y-intercept = background signal at reaction initiation (in AU).  $r_p =$  product coefficient for detected signal  $F$  per formed product  $[P]$  (in AU/M).  $k_{\text{sub}} =$  reaction rate constant for Hit-and-Run model of enzymatic product formation  $E + S \rightarrow E + P$  (in  $\text{M}^{-1}\text{s}^{-1}$ ).

because the unbound substrate concentration is no longer constant ( $[S]_t < [S]_0$ ) when a significant fraction of the substrate has been converted into product ( $[P]_t > 0.1[S]_0$ ). The contribution of substrate depletion to the progress curve is directly related to the enzyme activity, as  $>10\%$  substrate conversion is more likely to be exceeded when enzyme activity is high (Fig. 10B). Algebraic correction is performed by globally fitting all progress curves in presence of inhibitor to Equation III with shared values for  $k_{\text{inact}}$  and  $K_{\text{I}}^{\text{app}}$  (Fig. 10C and 10D). Substrate depletion should be the only factor contributing to nonlinearity, because the uninhibited control is not included in the global fit. Violation of this (and other) assumption requires data analysis by numerical solving (Kuzmič, 2015).

### Warnings and remarks

The authors demonstrate their algebraic model to correct for substrate depletion with the EGFR inhibitor afatinib in a homogeneous kinase activity assay. A bisubstrate kinase activity assay is different from our simulations with a single substrate, but this algebraic model can be applied in both systems: product formation in single-substrate as well as bisubstrate reactions can be simplified to a Hit-and-Run model ( $E + S \rightarrow E + P$ ) with rate constant  $k_{\text{sub}} = k_{\text{cat}}/K_{\text{M}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) as long as the substrate concentration is far below its  $K_{\text{M}}$  ( $[S] < 0.1K_{\text{M}}$ ) (Fig. 10A). The accuracy of  $k_{\text{inact}}$  and  $K_{\text{I}}$  was very good with low substrate concentrations ( $[S] \leq 0.01K_{\text{M}}$ ). A slightly higher substrate concentration ( $[S] \geq 0.1K_{\text{M}}$ ) resulted in underestimation of  $k_{\text{inact}}$  and overestimation of  $K_{\text{I}}$ , but a good estimation of overall second-order inactivation rate constant  $k_{\text{inact}}/K_{\text{I}}$ . Importantly, a calibration/titration (Dharadhar et al., 2019; Janssen et al., 2019) should be performed prior to data analysis to determine product coefficient  $r_{\text{P}}$  (in AU/M) that transforms the detected signal  $F_t$  (in AU) into product concentration  $[P]_t$  (in M).

### Two-Step Irreversible Covalent Inhibition With Substrate Depletion

Processing of raw data obtained with *Basic Protocol 1* for two-step irreversible covalent inhibitors with nonlinearity in the uninhibited control resultant from substrate depletion ( $[P]_t < 0.1[S]_0$ ).

*Before you start*, validate compliance with essential assay reaction conditions such as the Hit-and-Run model. This algebraic correction for substrate depletion (Kuzmič et al., 2015) has additional requirements for assay conditions, and is only compatible with two-step irreversible inhibition (Fig. 10). Validate that the product formation reaction complies with the Hit-and-Run model  $E + S \rightarrow E + P$  (Fig. 10A): substrate concentration must be far below the  $K_{\text{M}}$  ( $[S]_0 < 0.1K_{\text{M}}$ ) to calculate the pseudo-first order reaction rate constant for enzymatic product formation  $k_{\text{sub}} = k_{\text{cat}}/K_{\text{M}}$  ( $\text{M}^{-1}\text{s}^{-1}$ ). Observed nonlinearity in the uninhibited control should be fully attributed to substrate depletion. Convert the maximum signal  $F^{\text{ctrl}}$  (in AU) into product concentration (in M) using the product coefficient  $r_{\text{P}}$  (in AU/M product) as determined in a separate product calibration experiment (Dharadhar et al., 2019; Janssen et al., 2019). Validate that the total substrate conversion to product exceeds 10% of the initial substrate concentration ( $[P^{\text{ctrl}}]_t > 0.1[S]_0$ ), and that substrate depletion is the only factor that contributes to the observed nonlinearity: uninhibited product formation should be linear when incubation times are shorter ( $[P]_t < 0.1[S]_0$ ) or enzyme concentration is lower. Alternatively, perform kinetic analysis by numeric solving if one or more assumptions are violated (Kuzmič, 2015).

$$[P]_t = \frac{F_t - F_0}{r_{\text{P}}}$$

Calculate:  $P_t = (F_t - F_0) / r_{\text{P}}$  with  $P_t$  = product concentration at the end of the incubation  $[P]_t$  (in M),  $F_t$  = signal in uninhibited control at the end of the incubation time  $F_t$  (in AU),  $F_0$  = substrate background signal  $F_0$  (in AU) and  $r_{\text{P}}$  = product coefficient  $r_{\text{P}}$  (in AU/M product).

1. Plot signal  $F$  against incubation time  $t$ .

Plot signal (in AU) on the Y-axis against incubation time (in s) on the X-axis for each inhibitor concentration (Fig. 10C). Label the columns with the inhibitor concentration (in M).

2. Perform background correction.

Correct for assay artifacts such as fluorescence bleaching and drift that cause a declining signal in the fully inhibited control. This correction can be subtraction of the time-dependent background in absence of enzyme but in presence of substrate (and inhibitor), or subtraction of the fully inhibited control. Consult the guidelines of your data fitting software for instructions on background corrections (GraphPad Prism; see Internet Resources).

3. Globally fit  $F_t$  against  $t$  to obtain  $k_{\text{inact}}$  and  $K_I^{\text{app}}$ .

Globally fit the progress curves of time-dependent signal  $F_t$  for all inhibitor concentrations to Equation III. Exclude the dataset of the fully inhibited control from the fit. Constrain  $[E]_0$  (in M),  $[S]_0$  (in M), and  $[I] = [I]_0$  (in M) to their theoretical values. Originally,  $[I]_0$  was locally optimized (Kuzmič, 2015), but we used fixed values of  $[I]_0$  in GraphPad Prism. Constrain product coefficient  $r_p$  (in AU/M product) to the value determined in a separate product calibration experiment. Constrain  $k_{\text{inact}}$ ,  $K_I$ , and  $k_{\text{sub}}$  to a shared value for all datasets that must be greater than 0, and provide initial values that are in the anticipated range. Note that Equation III is in agreement with *equation C.16* of the original publication (Kuzmič et al., 2015), but  $[I]_0$  and  $k_{\text{inact}}$  were unintentionally displaced in *Equation III*. Calculate inactivation constant  $K_I$  (in M) and irreversible covalent inhibitor potency  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) with *Sample Calculations 1 and 2*.

$$F_t = F_0 + r_p[S]_0 \left\{ 1 - e^{-\beta(1-e^{-\alpha t})} \right\}$$

$$\alpha = \frac{k_{\text{inact}} [I]}{K_I^{\text{app}} + [I]}$$

$$\beta = \left( \frac{[E]_0 k_{\text{sub}}}{k_{\text{inact}}} \right) \left( \frac{K_I^{\text{app}}}{[I]} \right)$$

**Equation III**

Equation III for nonlinear regression of user-defined explicit equation:

$$a = k_{\text{inact}} * I_0 / (I_0 + K_I^{\text{app}})$$

$$b = (E_0 * k_{\text{sub}} / k_{\text{inact}}) * (K_I^{\text{app}} / I_0)$$

$$P = S_0 * (1 - \exp(-b * (1 - \exp(-a * X))))$$

$$Y = Y_0 + (r_p * P)$$

with  $Y$  = time-dependent signal  $F_t$  (in AU),  $X$  = incubation time  $t$  (in s),  $r_p$  = product coefficient  $r_p$  (AU/M product),  $E_0$  = maximum unbound enzyme concentration at reaction initiation  $[E]_0$  (in M),  $S_0$  = maximum unbound substrate concentration at reaction initiation  $[S]_0$  (in M) and  $I_0$  = maximum unbound inhibitor concentration  $[I]$  (in M) to find globally shared values for  $k_{\text{sub}}$  = product formation rate constant  $k_{\text{sub}} = k_{\text{cat}}/K_M$  (in  $\text{M}^{-1}\text{s}^{-1}$ ),  $k_{\text{inact}}$  = maximum rate of inactivation  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and  $K_I^{\text{app}}$  = apparent inactivation constant  $K_I^{\text{app}}$  (in M).

4. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental progress curves to the progress curves simulated with script **KinSubDpl** (using experimental rate constant

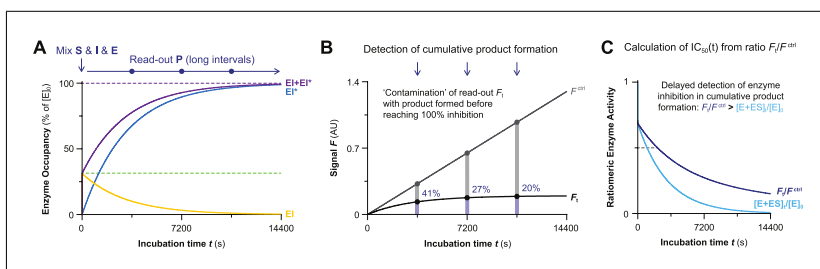
$k_{\text{inact}} = k_5$ ) to confirm that the calculated kinetic constants are in accordance with the experimental data.

## METHOD II: INCUBATION TIME-DEPENDENT POTENCY $IC_{50}(t)$

The observed potency of irreversible inhibitors increases with longer (pre)incubation time, as more enzyme is irreversibly bound. In this method, sometimes dubbed ‘the Krippendorff method’, the time-dependence of potency  $IC_{50}(t)$  is utilized to directly find the relevant kinetic parameters for two-step irreversible covalent inhibition. Contrary to progress curve analysis (*Method I*), this method is compatible with quenched/stopped assays that require a development/separation/quenching step before read-out, as continuous measurement of product formation is not required (but optional).

The incubation time-dependent potency  $IC_{50}(t)$  reflects the inhibitor concentration resulting in a 50% decrease of cumulative product formed  $F_t$  during incubation compared to cumulative product formed in the uninhibited control  $F^{\text{ctrl}}$ . Enzymatic product formation is initiated by enzyme addition without preincubation of enzyme and inhibitor (Fig. 11A). Fractional cumulative product formation  $F_t/F^{\text{ctrl}}$  decreases with longer incubation times (Fig. 11B). Importantly, this does not reflect the current enzyme activity because read-out  $F_t$  reflects that the cumulative product formed during incubation will be ‘contaminated’ with product that was formed before full inhibition was reached. Consequently, incubation time-dependent potency  $IC_{50}(t)$  calculated from the fractional product formation  $F_t/F^{\text{ctrl}}$  against inhibitor concentration will increase with longer incubation times (for slow-binding inhibitors), but will underestimate the potency compared to the values based on the current enzyme activity  $[E+ES]_t/[E]_0$  (Fig. 11C).  $IC_{50}(t)$  does not approach  $K_i^{\text{app}}$  (two-step reversible inhibition) or  $1/2[E]_0$  (irreversible inhibition) at infinite incubation times.

An implicit algebraic model based on multipoint  $IC_{50}(t)$  values has been derived (Krippendorff, Neuhaus, Lienau, Reichel, & Huisinga, 2009) for two-step irreversible covalent inhibitors (*Data Analysis 2*). Additionally, a two-point  $IC_{50}(t)$  method for two-step irreversible covalent inhibitors as well as a one-point  $IC_{50}(t)$  method for one-step irreversible covalent inhibitors have been reported in a preprint (Kuzmič, 2020b). To our knowledge, algebraic methods to calculate  $K_i^{\text{app}}$  (two-step reversible covalent inhibitors) from (end-point)  $IC_{50}(t)$  values have not been reported.



**Figure 11** Method II: Incubation time-dependent potency  $IC_{50}(t)$ . Simulated with **KinGen** for 50 nM inhibitor **C** with 1 pM enzyme and 100 nM substrate **S1**. **(A)** The reaction between enzyme, inhibitor, and substrate is initiated by addition of enzyme. Enzyme inhibition increases with time-dependent formation of covalent  $EI^*$  until reaching reaction completion. **(B)** Read-out of cumulative product formation (reflected in signal  $F_t$ ) in presence of two-step covalent inhibitor relative to product formed the uninhibited control ( $F^{\text{ctrl}}$ ) decreases upon longer incubation. **(C)** Cumulative product  $F_t$  (navy line) is ‘contaminated’ with product formed prior to reaching 100% inhibition even if the current enzyme activity (blue line) is fully inhibited.



**Incubation Time-Dependent Potency  $IC_{50}(t)$** 

The below protocol provides a generic set of steps to accomplishing this type of measurement.

**Materials**

- 1 × Assay/reaction buffer supplemented with co-factors and reducing agent
- Active enzyme, 4 × solution in assay buffer
- Competitive substrate with continuous or quenched read-out, 4 × solution in assay buffer
- Positive control: vehicle/solvent as DMSO stock, or 2% solution in assay buffer
- Negative control: known inhibitor or alkylating agent as DMSO stock, or 2 × solution in assay buffer
- Inhibitor: as DMSO stock, or serial dilution of 2 × solution in assay buffer with 2% DMSO
- Optional:* Development/quenching solution
- 384-well low volume microplate with nonbinding surface (e.g., Corning 3820 or 4513) for incubation and/or read-out
- Optical clear cover/seal (e.g., Perkin Elmer TopSeal-A Plus, #6050185, Corning 6575 Universal Optical Sealing Tape or Duck Brand HP260 Packing Tape) for *continuous* read-out, or a general microplate cover/lid (e.g., Corning 6569 Microplate Aluminum Sealing Tape) for non-continuous read-out
- 1.5 ml (Eppendorf) microtubes to prepare stock solutions
- Optional:* 96-well microplate to prepare serial dilution of inhibitor concentration
- Optional:* Microtubes to perform incubations (e.g., Eppendorf Protein Lobind Microtubes, #022431018)
- Microplate reader equipped with appropriate filters to detect product formation (e.g., CLARIOstar microplate reader)
- Optional:* Automated (acoustic) dispenser (e.g., Labcyte ECHO 550 Liquid Handler acoustic dispenser)

*Before you start*, optimize assay conditions in the uninhibited control to ensure compliance with assumptions and restrictions (Fig. 13) as outlined for *Basic Protocol I*. It is crucial to ensure that uninhibited product formation is linear with incubation time for the duration of the measurement: no enzyme degradation ( $k_{deg} = 0$ ) or other factors contributing to a nonlinearity in product formation in the uninhibited control ( $k_{ctrl} = 0$ ) are allowed, as correction for nonlinearity is not possible in *Data Analysis 2*. This method is compatible with homogeneous (continuous) assays but also with assays that require a development/quenching step to visualize formed product.

1. Add inhibitor or control (e.g., 0.2  $\mu$ l) and assay buffer (e.g., 10  $\mu$ l) to each well with the uninhibited control for full enzyme activity containing the same volume of vehicle/solvent instead of inhibitor as outlined in step 1 of *Basic Protocol I*.

Typically, measurements are performed in triplicate (or more replicates) with at least 8 inhibitor concentrations spanning the  $IC_{50}(t)$ . Inhibitor concentrations might need optimization, but a good starting point is  $[I] = 0.1-5 \times IC_{50}(t)$  at the shortest incubation time  $t$ . Alternatively, larger-volume incubations can be performed in (Eppendorf) Protein Lobind microtubes, from which aliquots are transferred to a microplate after the indicated incubation time. Whether incubation in tube or plate is performed is a matter of personal preference, compatibility with lab equipment and automation, and convenience of dispensing small volumes

2. Add substrate in assay buffer to each well (e.g., 5  $\mu$ l of 4 × solution) and homogenize the solutions by gentle shaking (1 min at 300 rpm).

The order of substrate or inhibitor addition is not important *per se*, as long as DMSO stocks are added prior to buffered (aqueous) solutions and the enzyme is the last

reagent to be added, to avoid unintentional preincubation (Fig. 13A). Inhibitor binding mode must be competitive with substrate. Optionally, gently centrifuge the plate or microtubes (1 min at 1000 rpm) to ensure assay components are not stuck at the top of the well.

3. Add active enzyme in assay buffer to each well (e.g., 5  $\mu$ l of 4 $\times$  solution) or tube as outlined in step 3 of *Basic Protocol 1*.

The accuracy of the measurement improves if the incubation time is monitored precisely.

4. Seal the wells by applying an (optical clear) cover or lid, or close the caps of microtubes to prevent evaporation of assay components during incubation.
5. *Optional*: Transfer aliquots (e.g., 20  $\mu$ l) from the reaction mixture to the microplate after each time point, if incubation is performed in large volumes (in Protein Lobind microtubes or 96-well NBS plate) rather than incubation of replicates in a 384-well microplate.
6. *Quenching*: Add development solution to the reaction mixture in the microplate to quench the product formation reaction for assay formats that require a development/quenching step to visualize formed product.

Incubation time  $t$  is the elapsed time between reaction initiation by enzyme addition (step 3) and (optional) quenching of the enzyme activity by addition of development/quenching solution (step 6).

7. Measure formed product after incubation by detection of the product read-out in microplate reader.

Follow manufacturer's advice on waiting time after addition of development solution before read-out. A typical assay measurement window is >2 hr, measuring cumulative product formation every 5-30 min (Fig. 11). The best results are obtained when inhibitor concentrations cover at least 50% of the DRC at all incubation times (Fig. 12C) and there is a significant decrease from the earliest to the last  $IC_{50}(t)$  value (Fig. 12D).

8. Proceed to *Basic Data Analysis Protocol 2* to calculate relevant kinetic parameters for two-step irreversible covalent inhibition

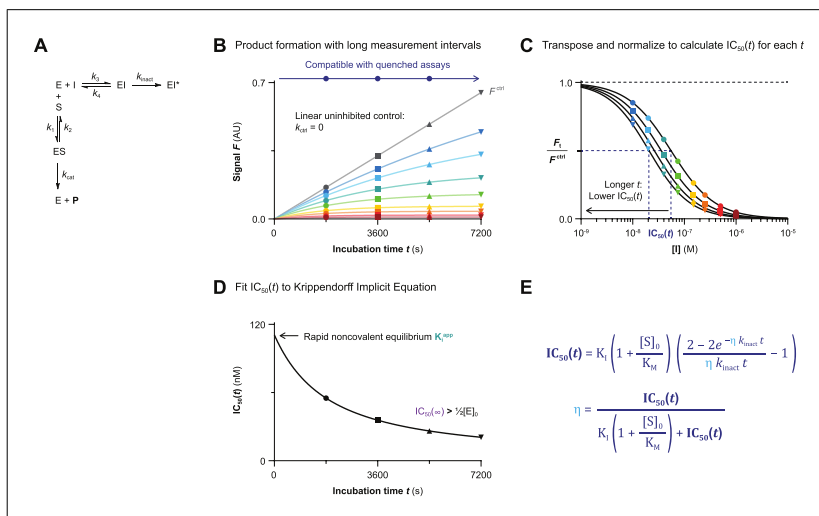
EXP Conditions	Data Analysis Protocol		
	2-step IRREV	1-step IRREV	2-step REV
$k_{ctrl} = 0$	2	–	–

Exemplary assay concentrations.

	Concentration during incubation $t$		
	[stock]	V ( $\mu$ l)	[conc] <sub><math>t</math></sub>
<b>Enzyme</b>	4 nM	5	0.99 nM
<b>Inhibitor</b>	20 nM	10.2	<b>10.10 nM</b>
<b>Substrate</b>	4 $\mu$ M	5	<b>0.99 <math>\mu</math>M</b>
<i>Total</i>		20.2	

### Data Analysis 2: Incubation Time–Dependent Potency $IC_{50}(t)$ for Two-Step Irreversible Covalent Inhibition

Krippendorff and co-workers report an algebraic model to calculate  $k_{inact}$  and  $K_I$  of irreversible covalent inhibitors from the incubation time–dependent potency  $IC_{50}(t)$  after multiple incubation times (Krippendorff et al., 2009). Detection of cumulative product



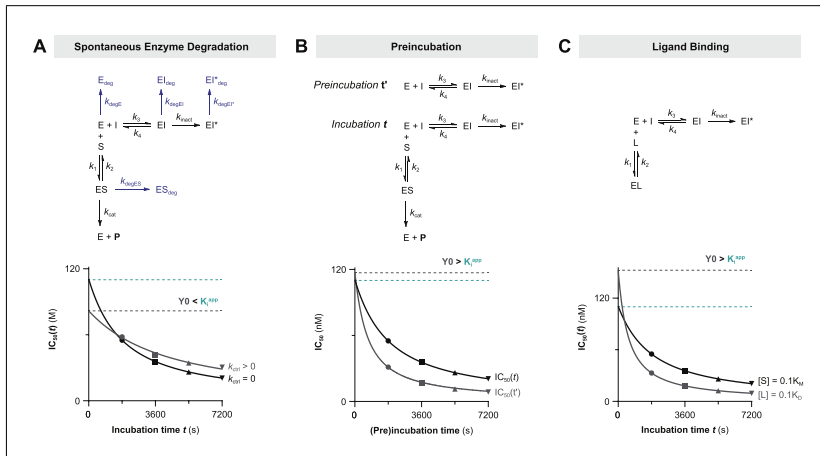
**Figure 12** Data Analysis 2: Incubation time–dependent potency  $IC_{50}(t)$  for two-step irreversible covalent inhibition. Simulated with **KinGen** for inhibitor **C** with 1 pM enzyme and 100 nM substrate **S1**. **(A)** Schematic enzyme dynamics during incubation for two-step irreversible covalent inhibition. **(B)** Time-dependent cumulative product formation in absence of inhibitor  $F^{ctrl}$  or in presence of inhibitor  $F_t$  is detected with longer measurement intervals compatible with quenched assays. **(C)** Incubation time-dependent potency  $IC_{50}(t)$  reflects the inhibitor concentration that reduces cumulative product formation during incubation by 50% compared to the uninhibited control. **(D)** Incubation time–dependent potency  $IC_{50}(t)$  against incubation time is fitted to Equation IV.  $IC_{50}(0)$  approaches apparent noncovalent inhibition constant  $K_i^{app}$  but  $IC_{50}(0)$  is never included in the fit because product formation does not start until initiation of the incubation ( $F_0 = F^{ctrl} = 0$ ). **(E)** Implicit algebraic Equation IV (Krippendorff et al., 2009).

formation after several incubation times is compatible with continuous assays, but more importantly also with stopped/quenched assays that require a development step to visualize product formation (Fig. 12A and 12B). Incubation time–dependent potency  $IC_{50}(t)$  is calculated for each incubation time from fractional product formation  $F_t/F^{ctrl}$  (Fig. 12C) and plotted against the incubation time (Fig. 12D). Finally, the authors derived *implicit* algebraic Equation IV (Fig. 12E) to calculate  $k_{inact}$  and  $K_I$  from the incubation time–dependent potency  $IC_{50}(t)$ . This method is restricted to substrate-competitive irreversible (multi-step) covalent inhibitors:  $k_{inact}$  and  $K_I$  do not have a biological meaning for reversible inhibitors or for one-step covalent inhibitors.

### Warnings and remarks

This method requires software (e.g., GraphPad Prism) that allows fitting a model defined by an implicit equation (where  $Y$  appears on both sides of the equal sign). Product formation in the uninhibited control should be strictly linear ( $k_{ctrl} = 0$ ): normalization of cumulative product formation ( $F_t/F^{ctrl}$ ) does not correct for spontaneous loss of enzyme activity or substrate depletion. It is relatively easy to miss violations of this assumption because nonlinearity in the uninhibited control ( $k_{ctrl} > 0$ ) is not evident from visual inspection of the dose-response curves (Fig. 12B). Violation of this assumption results in a significant underestimation of  $k_{inact}$  and  $K_I$  values, also when nonlinearity is relatively small ( $k_{ctrl} \ll k_{inact}$ ) (Fig. 13A).

Another important assumption is that the onset of product formation and enzyme inhibition occur simultaneously: inhibition and product formation are both initiated by addition of enzyme, without preincubation of enzyme and inhibitor prior to substrate addition. Unfortunately, numerous publications refer to preincubation of enzyme and



**Figure 13** Experimental Restrictions to fitting Equation IV (Fig. 12E) in Data Analysis 2. **(A)** Enzyme degradation/denaturation simulated with **KinDeg** for inhibitor **C** with 1 pM enzyme, 100 nM substrate **S1**, and  $k_{\text{ctrl}} = k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}} = k_{\text{degEI}^*}$  with  $k_{\text{ctrl}} = 0 \text{ s}^{-1}$  (black) or  $k_{\text{ctrl}} = 0.0003 \text{ s}^{-1}$  (gray). The rate of inactivation  $k_{\text{inact}}$  is significantly underestimated and the potency of inactivation constant  $K_i$  is overestimated when  $k_{\text{ctrl}} > 0$ . **(B)** Preincubation time–dependent potency  $\text{IC}_{50}(t')$  simulated with **KinGen** for inhibitor **C** with 1 pM enzyme and 100 nM substrate **S1**. The rate of inactivation  $k_{\text{inact}}$  is overestimated, resulting in overestimation of the inactivation efficiency  $k_{\text{inact}}/K_i$  when preincubation-dependent  $\text{IC}_{50}(t')$  (gray) is fitted instead of incubation-dependent  $\text{IC}_{50}(t)$  (black). Accurate values for preincubation-dependent potency can be obtained by performing Data Analysis 3A (Fig. 15). **(C)** Ligand binding assay simulated with **KinGen** for inhibitor **C** with 1 pM enzyme and 100 nM ligand **L1**. The rate of inactivation  $k_{\text{inact}}$  is overestimated while the potency of inactivation constant  $K_i$  is underestimated, resulting in overestimation of the inactivation efficiency  $k_{\text{inact}}/K_i$  when time-dependent  $\text{IC}_{50}(t)$  from ligand binding inhibition (gray) is fitted instead of substrate cleavage (black).

inhibitor as ‘incubation’, resulting in the understandable but incorrect fitting of preincubation time–dependent potency  $\text{IC}_{50}(t')$  to the Krippendorff model (Kuzmič, 2020b). Preincubation-dependent potency  $\text{IC}_{50}(t')$  is calculated from product formation velocity  $v_t'$ , reflecting the enzyme activity after preincubation rather than cumulative product formation  $F_t/F^{\text{ctrl}}$ . Enzyme activity  $v_t'$  is not ‘contaminated’ by product formed prior to read-out because product formation is initiated *after* the preincubation. Furthermore, substrate does not compete with inhibitor for enzyme binding during preincubation. Fitting  $\text{IC}_{50}(t')$  values to the Krippendorff model resulted in an overestimation of  $k_{\text{inact}}$  and an overestimation of the overall inactivation potency  $k_{\text{inact}}/K_i$  (Fig. 13B).

This method is not compatible with ligand binding competition assays (such as the Lanthascreen kinase binding assay) where inhibitor binding competes with ligand (tracer) binding to form enzyme–ligand complex EL as the detectable product (Fig. 13C). The enzyme–ligand equilibrium after incubation in presence of inhibitor reflects the current inhibitor competition and is unaffected by binding equilibria prior to read-out (not cumulative). Furthermore, unbound enzyme is not released after formation of product EL, thereby limiting the product formation to a single turnover per enzyme. Fitting  $\text{IC}_{50}(t)$  values obtained in ligand-binding assays (simulated with  $k_{\text{cat}} = 0$ ) to the Krippendorff model result in overestimation of  $k_{\text{inact}}$  and/or unstable parameters.

## Two-Step Irreversible Covalent Inhibition

Processing of raw data obtained with *Basic Protocol I* or *Basic Protocol II* for two-step irreversible covalent inhibitors.

1. Plot signal  $F$  against incubation time  $t$ .

Plot cumulative signal (in AU) on the Y-axis against incubation time (in s) on the X-axis for each inhibitor concentration and for the controls (Fig. 12B). Label the columns with the inhibitor concentration (in M). It is not possible to algebraically correct for spontaneous loss of enzyme activity. Validate that the product formation in the uninhibited control  $F^{\text{ctrl}}$  is linear ( $v_i = v_s$ ) by performing steps 1-3 of *Basic Data Analysis Protocol 1A* with  $k_{\text{obs}} = k_{\text{ctrl}}$ . Consult Table 3 for troubleshooting of nonlinearity of the uninhibited control.

2. Perform background correction.

Correct for assay artifacts such as fluorescence bleaching and drift that cause a declining signal in the fully inhibited control. This correction can be subtraction of the time-dependent background in absence of enzyme but in presence of substrate (and inhibitor), or subtraction of the fully inhibited control.

3. Transpose to plot signal  $F$  against inhibitor concentration  $[I]$ .

For each incubation time, transpose the X and Y values to plot signal  $F_t$  (in AU) on the Y-axis against inhibitor concentration (in M) on the X-axis. Also include product formation in the uninhibited control  $F^{\text{ctrl}}$  ( $[I] = 0$ ).

4. Normalize  $F_t/F^{\text{ctrl}}$ .

Normalize  $F_t$  (in AU) to lowest value = 0 (in AU) and highest value = uninhibited product formation  $F^{\text{ctrl}}$  (in AU) to obtain fractional product formation in presence of inhibitor  $F_t/F^{\text{ctrl}}$  (Fig. 12C). Consult the guidelines of your data fitting software for instructions on data normalization to the positive and negative controls (GraphPad; see Internet Resources).

5. Plot and fit  $F_t/F^{\text{ctrl}}$  against  $[I]$  to obtain the incubation time-dependent potency  $IC_{50}(t)$ .

Plot the dose-response curve of fractional signal  $F_t/F^{\text{ctrl}}$  against inhibitor concentration (in M), and fit to four-parameter nonlinear regression Hill Equation XII (Copeland, 2013e) to obtain the incubation time-dependent potency  $IC_{50}(t)$  (in M) (Fig. 12C). Use the inhibitor concentration *during* incubation: after reaction initiation by enzyme addition but before (optional) addition of development solution (*Basic Protocol II*, step 3).

$$\frac{F_t}{F^{\text{ctrl}}} = \frac{1}{1 + \left(\frac{IC_{50}(t)}{[I]}\right)^h}$$

**Equation XII**

Equation XII for nonlinear regression of four-parameter dose-response equation  $Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + (IC_{50}/X)^{\text{HillSlope}})$  with  $Y$  = fractional product signal  $F_t/F^{\text{ctrl}}$  (unitless),  $X$  = inhibitor concentration  $[I]$  (in M),  $\text{Bottom}$  = normalized fully inhibited product signal = 0 (unitless), and  $\text{Top}$  = normalized uninhibited product signal  $F^{\text{ctrl}}/F^{\text{ctrl}} = 1$  (unitless) to find  $\text{HillSlope}$  = Hill coefficient  $h$  (unitless) and  $IC_{50}$  = incubation time-dependent potency  $IC_{50}(t)$  (in M).

6. Plot and fit  $IC_{50}(t)$  against  $t$  to obtain  $k_{\text{inact}}$  and  $K_I$ .

Plot the mean and standard deviation of  $IC_{50}(t)$  (in M) on the Y-axis against incubation time  $t$  (in s) on the X-axis (Fig. 12D). The rate of covalent bond formation at

saturation inhibitor concentration  $k_{\text{inact}}$  and inactivation constant  $K_I$  are obtained by solving implicit Equation IV (Krippendorff et al., 2009) (Fig. 12E). Use the substrate concentration *during* incubation (*Basic Protocol II*, step 3): after reaction initiation by enzyme addition but before (optional) addition of development/quenching solution. It is important that the Michaelis constant  $K_M$  be accurate for the reaction conditions (buffer, temperature, substrate), as this value is directly used to correct inactivation constant  $K_I$  for substrate competition. Consult the guidelines of your data-fitting software (GraphPad; see Internet Resources for website) for instructions on solving implicit equations (where  $Y$  appears on both sides of the equal sign). Proceed to *Sample Calculation 2* to calculate irreversible covalent inhibitor potency  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) with propagation of error.

$$\text{IC}_{50}(t) = K_I \left( 1 + \frac{[S]_0}{K_M} \right) \left( \frac{2 - 2e^{-\eta k_{\text{inact}} t}}{\eta k_{\text{inact}} t} - 1 \right) \quad \text{with } \eta = \frac{\text{IC}_{50}(t)}{K_I \left( 1 + \frac{[S]_0}{K_M} \right) + \text{IC}_{50}(t)}$$

**Equation IV**

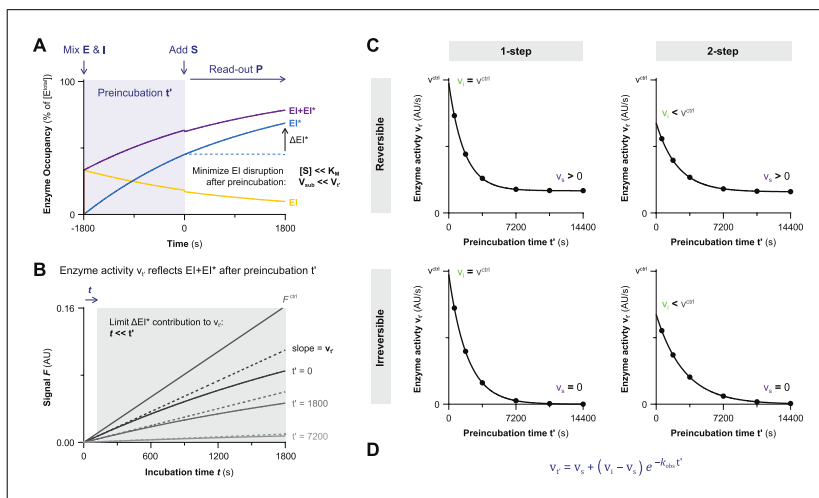
Equation IV for nonlinear regression of user-defined implicit equation  $Y = (K_I * (1 + (S/K_M))) * ((2 - (2 * \text{EXP}(- (Y / (K_I * (1 + (S/K_M)))) + Y)) * k_{\text{inact}} * X)) / ((Y / (K_I * (1 + (S/K_M))) + Y) * k_{\text{inact}} * X) - 1)$ , with  $Y$  = incubation time-dependent potency  $\text{IC}_{50}(t)$  (in M),  $X$  = incubation time  $t$  (in s),  $S$  = maximum unbound substrate concentration at reaction initiation  $[S]_0$  (in M), and  $K_M$  = Michaelis constant  $K_M$  (in M) to find  $k_{\text{inact}}$  = inactivation rate constant  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and  $K_I$  = inactivation constant  $K_I$  (in M).

**7. Optional:** Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental read-out to the product formation simulated with scripts **KinGen** and **KinDeg** (using experimental rate constant  $k_{\text{inact}} = k_5$ ) to confirm that the calculated kinetic constants are in accordance with the experimental data and found  $\text{IC}_{50}(t)$  values.

**METHOD III: PREINCUBATION TIME-DEPENDENT INHIBITION WITHOUT DILUTION**

Preincubation of enzyme and inhibitor prior to initiation of product formation by addition of substrate is an established method for kinetic analysis of slow-binding (ir)reversible (covalent) inhibitors (Copeland, 2013b; Ito et al., 1998). In the benchmark protocol by Ito and co-workers, a low substrate concentration ( $[S] \ll K_M$ ) is added in a relatively small volume ( $V_{\text{sub}} \ll V_t$ ) to keep the noncovalent enzyme-inhibitor  $E + I \leftrightarrow EI$  equilibrium intact. However, (partial) disruption of the noncovalent equilibrium does not affect the accuracy of preincubation experiments for irreversible inhibition, as is illustrated by *Method IV*. Product formation is inhibited by formation of  $EI$  and  $EI^*$  during preincubation in absence of competing substrate (Fig. 14A). Preincubation time-dependent product formation velocity  $v_t$  reflects the total inhibition by noncovalent as well as covalent inhibitor binding, and is calculated after a relatively short incubation time ( $t \ll t'$ ) to minimize additional (time-dependent) inhibition of enzyme activity during incubation resultant from enzyme-inhibitor complex/adduct formation during incubation (Fig. 14B). Enzyme activity after preincubation in the presence of time-dependent inhibitors  $v_t$  decreases exponentially from rapid (initial) equilibrium  $K_i^{\text{app}}$  ( $Y$ -intercept:  $v_i$ ) to reach a plateau at reaction completion ( $t' > 5t/2$ ), corresponding to the steady-state equilibrium ( $v_s > 0$ ) or inactivation ( $v_s = 0$ ) (Fig. 14C). Observed rate of reaction completion  $k_{\text{obs}}$  (from enzyme activity without preincubation  $v_i$  to final enzyme activity  $v_s$ ) is obtained by fitting to bounded exponential decay Equation V (Fig. 14D). Importantly, this equation fits enzyme activity  $v_t$  (in AU/s) rather than directly fitting product signal  $F$  (in AU).



**Figure 14** Method III: Preincubation time–dependent inhibition without dilution. Simulated with **KinGen** for 1 pM enzyme and 100 nM substrate **S1**. **(A)** Enzyme is preincubated with inhibitor to form noncovalent complex EI and covalent adduct EI\* in absence of competing substrate, followed by addition of substrate. Addition of a low substrate concentration in a small volume to avoid disruption of the noncovalent  $E + I \rightleftharpoons EI$  equilibrium. Simulated for 50 nM inhibitor **C** with preincubation  $t' = 1800$  s. **(B)** Preincubation time–dependent enzyme activity  $v_t$  is obtained from the slope of (initial) linear product formation velocity with a short incubation time  $t$  relative to preincubation  $t'$  to minimize  $\Delta EI^*$  formation after substrate addition. This measurement is performed separately for each preincubation time, thus requiring more material than incubation time–dependent inhibition protocols with continuous product read-out. Simulated for 50 nM inhibitor **C** with preincubation  $t' = 1800$  s. **(C)** Enzyme activity  $v_t$  of time-dependent inhibitors decreases exponentially from rapid (initial) equilibrium  $K_i^{app}$  (Y-intercept = enzyme activity without preincubation  $v_i$ ) to reaching reaction completion ( $t' > 5t/2$ ): inactivation for irreversible inhibitors ( $v_s = 0$ ) and steady-state equilibrium  $K_i^{app}$  for reversible inhibitors ( $v_s > 0$ ). Enzyme activity without preincubation  $v_i$  equals the uninhibited enzyme activity  $v^{ctrl}$  for one-step inhibitors and for two-step inhibitors at non-saturating concentration ( $[I] \ll K_i^{app}$ ). Simulated for 50 nM one-step reversible inhibitor **B**, two-step reversible inhibitor **B**, one-step irreversible inhibitor **D**, and two-step irreversible inhibitor **C**. **(D)** General bounded exponential decay Equation V to fit preincubation time–dependent enzyme activity  $v_t$  (in AU/s) against preincubation time  $t'$  (in s). Parameters are constrained depending on the inhibitor binding mode. Irreversible inhibition:  $v_s = 0$  (inactivation at reaction completion). One-step inhibition:  $v_i = v^{ctrl}$  (non-covalent complex is not significant at non-saturating inhibitor concentrations).  $v_t$  = preincubation time–dependent enzyme activity (in AU/s).  $v_i$  = Enzyme activity based without preincubation (in AU/s).  $v_s$  = Enzyme activity after preincubation ( $t' > 5t/2$ ) based on reaching reaction completion (in AU/s).  $t'$  = preincubation time of enzyme and inhibitor before substrate addition (in s).  $k_{obs}$  = observed rate of time-dependent inhibition from initial  $v_i$  to final  $v_s$  (in  $s^{-1}$ ).

Algebraic analysis by linear regression to obtain  $k_{obs}$  from the (initial) linear slope of  $\ln(\text{enzyme activity})$  against preincubation time  $t'$  is still frequently reported. This is probably because linear regression is part of benchmark protocols (Ito et al., 1998; Kitz & Wilson, 1962) for kinetic analysis of preincubation time–dependent enzyme inactivation. It is important to note that these benchmark protocols were published before dedicated data analysis software for nonlinear regression was available (Perrin, 2017). Visualization of this ‘linear’ relationship is possible by plotting the enzyme activity against preincubation time  $t'$  on a semilog scale (illustrated in Fig. S1 in Supporting Information).

Preincubation assays are generally disfavored because their experimental execution requires more material and is more laborious than substrate competition assays with continuous read-out (*Method I* and *II*). Here, substrate has to be added after the indicated preincubation time, thus requiring multiple individual measurements for each inhibitor

concentration. However, preincubation experiments are still favored when reaction completion is too slow for detection during the normal time course of an substrate competition assay ( $t \ll t_{1/2}$  in *Method 1*): substrate competition reduces the (covalent) reaction rate and inhibitor solubility limits the maximum inhibitor concentration. Instead, preincubation is performed in the absence of competing substrate, thus reaching the maximum reaction rate at a low inhibitor concentration. Therefore, preincubation experiments are frequently conducted for compounds that display one-step irreversible inhibition behavior because they have a poor noncovalent affinity, such as covalent fragments (Kathman & Statsyuk, 2019). Additionally, preincubation times can exceed the maximum incubation time of progress curve analysis, which is limited by linear product formation ( $[P]_t > 0.1[S]_0$ ), as the onset of product formation does not start until preincubation is completed.

This method is less suitable for enzymatic assays with a relatively slow uninhibited product formation velocity  $v^{\text{ctrl}}$ , as assay sensitivity might be insufficient to produce enough product signal  $F_t$  during a short incubation time. Reaction completion ( $t' > 5t_{1/2}$ ) and/or full inhibition ( $v_t = 0$ ) should not be reached before the first (shortest) preincubation time because it will be impossible to detect time-dependent changes in enzyme activity. This can be resolved by increasing the measurement interval (shorter  $dt'$ ), reduction of the inhibitor concentration, or selection of a different experimental protocol. This method is compatible with two-step irreversible inhibition (*Data Analysis 3A*) and one-step irreversible inhibition (*Data Analysis 3B*), but also with (two-step) reversible inhibition (*Data Analysis 3C*).

The protocol below provides a generic set of steps to accomplishing this type of measurement. Specific reagents, and assay conditions for preincubation time-dependent inhibition of irreversible covalent papain inhibitor fragments can be found in this reference (Kathman et al., 2014).

### **Materials**

- 1 × Assay/reaction buffer supplemented with co-factors and reducing agent
- Active enzyme, 2 × solution in assay buffer
- Substrate with continuous or quenched read-out, 11 × solution in assay buffer
- Positive control: vehicle/solvent as DMSO stock, or 2% solution in assay buffer
- Negative control: known inhibitor or alkylating agent as DMSO stock, or 2 × solution in assay buffer
- Inhibitor: as DMSO stock, or serial dilution of 2 × solution in assay buffer with 2% DMSO
- Optional:* Development/quenching solution
- 1.5 ml (Eppendorf) microtubes to prepare stock solutions
- 384-well low volume microplate with nonbinding surface (e.g., Corning 3820 or 4513) for preincubation and/or read-out
- General microplate cover/lid (e.g., Corning 6569 Microplate Aluminum Sealing Tape) if preincubation is conducted in a microplate
- Optional:* 96-well microplate to prepare serial dilution of inhibitor concentration
- Optional:* Microtubes to perform preincubations (e.g., Eppendorf Protein Lobind Microtubes, #022431018)
- Microplate reader equipped with appropriate filters to detect product formation (e.g., CLARIOstar microplate reader)
- Optional:* Automated (acoustic) dispenser (e.g., Labcyte ECHO 550 Liquid Handler acoustic dispenser)

### **Preincubation Time-Dependent Inhibition Without Dilution**

*Before you start*, optimize assay conditions in the uninhibited control to ensure compliance with assumptions and restrictions, as outlined in the *Critical Parameters*:



*Assumptions on Experimental Assay Conditions* section and *Basic Protocol 1*. Consult Table 3 in the troubleshooting section for common optimization and troubleshooting options. Specific adjustments for *Method III* are that substrate concentration should be relatively low ( $[S]_0 \ll K_M$ ) to minimize disruption of the noncovalent  $E + I \rightleftharpoons EI$  equilibrium or reduction of reaction rates by competition (Fig. 14A); adjustment of the enzyme concentration might be required to ensure that maximum 10% of the substrate is processed during the read-out ( $[P]_t < 0.1[S]_0$ ) and product formation is linear in the uninhibited control. Furthermore, incubation time  $t$  must be relatively short to minimize additional time-dependent enzyme inhibition after substrate addition. As a rule of thumb, incubation must be much shorter than the shortest preincubation ( $t \ll t'$ ), unless the product formation read-out is continuous (more details in *Data Analysis 3*, step 3). Validate that enough product is formed for a good signal/noise ratio ( $Z' > 0.5$ ) by calculating the  $Z'$ -score from the uninhibited and inhibited controls (ideally 8 replicates) in a separate experiment (Zhang et al., 1999). This method is compatible with homogeneous (continuous) assays but also with assays that require a development/quenching step to visualize formed product. Note that this protocol was designed for preincubation and read-out in a 384-well microplate.

1. Add inhibitor or control (e.g., 0.2  $\mu$ l) and assay buffer (e.g., 10  $\mu$ l) to each well with the uninhibited control for full enzyme activity containing the same volume vehicle/solvent instead of inhibitor as outlined in step 1 of *Basic Protocol 1*.

Gently shake to mix DMSO with the aqueous buffer. Typically, measurements are performed in triplicate (or more replicates) with at least 8 inhibitor concentrations for at least 5 preincubation times. Inhibitor concentrations might need optimization, but a rational starting point is to use inhibitor concentrations below 5 times the  $IC_{50}$  at the shortest preincubation time  $t'$ : inhibition is expected to improve in a time-dependent manner and the best results are obtained when full inhibition is not achieved already at the shortest preincubation time (Fig. 14C). Alternatively, larger-volume preincubations (e.g., >200  $\mu$ l) can be performed in (Eppendorf) microtubes from which aliquots (e.g., 20.2  $\mu$ l) are transferred to a microplate after the indicated preincubation time. Whether preincubation is performed in a tube or microplate is a matter of personal preference, compatibility with lab equipment and automation, and convenience of dispensing small volumes.

2. Add active enzyme in assay buffer to each well (e.g., 10  $\mu$ l of 2 $\times$  solution) or tube to start preincubation of enzyme with inhibitor and homogenize the solution by gently shaking (1 min at 300 rpm). Alternatively, dispensing the enzyme at a high flow rate will also mix the components.

The order of enzyme and inhibitor addition is not important *per se*, as long as DMSO stocks are added prior to buffered (aqueous) solutions. Inhibitor must be present in excess during preincubation ( $[I]_0 > 10[E]_0$ ). Optionally, gently centrifuge the plate or microtubes (1 min at 1000 rpm) to ensure assay components are not stuck at the top of the well.

3. Seal the wells with a cover or lid, and close the caps of microtubes to prevent evaporation of assay components during preincubation.
4. *Optional*: Transfer aliquots (e.g., 20.2  $\mu$ l) from the reaction mixture to the microplate after completion of preincubation if performed in larger volumes.
5. Add substrate in assay buffer (e.g., 2  $\mu$ l of 11 $\times$  solution) to (at least) three designated replicates after preincubation time  $t'$ .

Typically, preincubation can run anywhere from several minutes to hours depending on the enzyme stability and anticipated inhibitor potency, with superior accuracy if

the preincubation time is monitored precisely. Substrate should be added in a negligible volume ( $V_{\text{sub}} < 0.1V_t$ ) to minimize disruption of the noncovalent equilibria by dilution ( $V_t = V_t'$ ) (Fig. 14A). Because at steady-state the equilibrium can be disrupted by dilution in too much competitive substrate, keep the substrate volume  $V_{\text{sub}}$  and substrate concentration low ( $[S]_0 < 0.1K_M$ ) for successful analysis of two-step reversible inhibitors (*Data Analysis 3C*). Optionally, homogenize the solutions by gentle shaking (300 rpm) and centrifuge the plate or microtubes (1 min at 1000 rpm) to ensure that assay components are not stuck at the top of the well.

6. *Quenching*: Add development solution to the reaction mixture in the microplate to quench the product formation reaction if read-out of product formation requires a development/quenching step to visualize formed product after incubation time  $t$ .

Follow manufacturer's advice on waiting time after addition of development solution before read-out. Incubation time  $t$  is the elapsed time between onset of product formation by substrate addition (step 5) and addition of development/quenching solution (step 6). A possible advantage to the use of a quenched assay is the possibility to store the samples after addition of quenching/development solution (step 6) and measure product formation (step 7) in all samples after completion of the final preincubation rather than performing multiple separate measurements (after each preincubation time).

7. Measure formed product after incubation by detection of the product read-out in microplate reader.

Incubation time (after substrate addition) is relatively short ( $t \ll \text{LN}(2)/k_{\text{obs}}$ ) to minimize additional (time-dependent) inhibition of enzyme activity during incubation (Fig. 14B).

8. Repeat *Basic Protocol III*, steps 4-7 for at least another four preincubation times.

Preincubation time  $t'$  is the elapsed time between onset of inhibition by mixing enzyme and inhibitor (step 2) and addition of substrate (step 5). A typical preincubation assay consists of multiple hours of measuring enzyme activity every 5-30 min, depending on enzyme stability and inhibitor reaction rates. Best results are obtained if the incubation time  $t$  used to calculate enzyme activity is kept constant at all preincubation times.

9. Proceed to *Basic Data Analysis Protocol 3* to convert the raw experimental data into preincubation time-dependent enzyme activity.

### **BASIC DATA ANALYSIS PROTOCOL 3**

#### **Preincubation Time-Dependent Inhibition Without Dilution**

Processing of raw experimental data obtained with *Basic Protocol III* for all inhibitor binding modes illustrated in Figure 1.

1. Plot signal  $F$  against incubation time  $t$ .

Plot signal  $F$  (in AU) on the Y-axis against the incubation time (in s) on the X-axis for each inhibitor concentration and for the controls (Fig. 14B). *Do this separately for each preincubation time.* Proceed to step 3 of this protocol for continuous read-out assays that require a longer incubation time to produce enough product for a good signal/noise ratio.

2. Fit  $F_t$  against  $t$  to obtain  $v_t'$ .

Fit signal  $F$  (in AU) against incubation time  $t$  (in s) to Equation XIII (Fig. 15B/ Fig. 16B, left) to obtain preincubation time-dependent product formation velocity  $v_t'$  (in

AU/s) from the linear slope. Consult Table 3 for troubleshooting if product formation is not linear.

$$F_t = F_0 + v_t t$$

**Equation XIII**

Equation XIII for nonlinear regression of straight line  $Y = Y_{\text{Intercept}} + \text{Slope} * X$  with  $Y = \text{signal } F_t$  (in AU) and  $X = \text{incubation time } t$  (in s) to find  $Y_{\text{Intercept}} = \text{background signal at reaction initiation } F_0$  (in AU) and  $\text{Slope} = \text{preincubation time-dependent product formation velocity } v_t$  (in AU/s).

3. *Alternative for continuous:* Fit  $F_t$  against  $t$  to obtain  $v_t$ .

This is an alternative method to obtain  $v_t$  from the initial velocity for assays with a continuous readout, using the initial velocity in progress curve analysis (*Method I*). Fit signal  $F_t$  against incubation time  $t$  to exponential association Equation XIV (Fig. 15B/ Fig. 16B, right) to obtain preincubation time-dependent product formation velocity  $v_t$  (in AU/s) from the initial velocity. This resolves issues with low signal/noise ratios for continuous read-out assays where  $v_t$  is not linear (due to additional covalent modification during the incubation) by allowing longer incubation times to produce sufficient signal.

$$F_t = v_s t + \frac{v_t - v_s}{k} [1 - e^{-kt}] + F_0$$

**Equation XIV**

Equation XIV for nonlinear regression of user-defined explicit equation  $Y = (v_s * X) + ((v_i - v_s) / k_{\text{obs}}) * (1 - \text{EXP}(-k_{\text{obs}} * X)) + Y_0$  with  $Y = \text{signal } F_t$  (in AU) and  $X = \text{incubation time } t$  (in s) to find  $Y_0 = Y\text{-intercept } F_0 = \text{background signal at } t = 0$  (in AU),  $v_i = \text{initial slope} = \text{preincubation time-dependent product formation velocity } v_t$  (in AU/s),  $v_s = \text{final slope}$  (in AU/s) and  $k_{\text{obs}} = \text{non-linearity reaction rate } k$  (in  $s^{-1}$ ).

4. Proceed to Data Analysis Protocols to obtain the appropriate kinetic parameters for each covalent binding mode: *Data Analysis Protocol 3Ai* or *3Aii* for two-step irreversible inhibitors, *Data Analysis Protocol 3Bi* or *3Bii* for one-step irreversible inhibitors, and *Basic Data Analysis Protocol 3C* for two-step reversible inhibitors.

Selection of a data analysis method for inhibitors with an irreversible binding mode depends on the desired visual representation as well as personal preference. Generally, *Basic Data Analysis Protocols 3Ai* and *3Bi* have less data processing/manipulation and are more informative for comparison of various inhibitors on a single enzyme target, as they are compatible with assessment of inhibitor potency simultaneous with visual assessment of time-dependent enzyme stability  $k_{\text{ctrl}}$  (Figs. 15F and 16F). *Alternative Data Analysis Protocols 3Aii* and *3Bii* involve normalization of the enzyme activity that aids visual assessment of inhibitory potency of a single inhibitor on multiple enzyme targets (that might have a variable stability) (Figs. 15H and 16H).

EXP Conditions	Data Analysis Protocol		
	2-step IRREV	1-step IRREV	2-step REV
$k_{\text{ctrl}} = 0$	3Ai	3B	3C
$k_{\text{degE}} = 0$	3Ai/3Aii	3Bi/3Bii	3C

Exemplary assay concentrations during preincubation and during incubation.

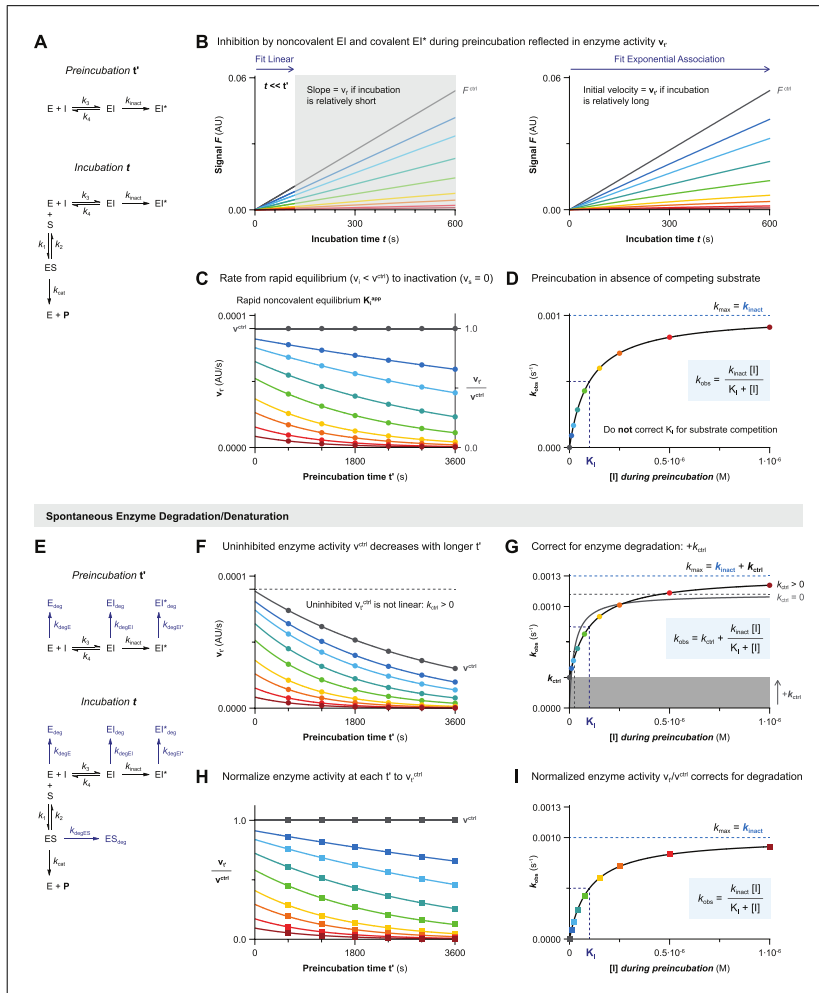
	Concentration during preincubation $t'$			Concentration during incubation $t$		
	[stock]	V ( $\mu$ l)	[conc] $_{t'}$	[stock]	V ( $\mu$ l)	[conc] $_t$
Enzyme	2 nM	10	0.99 nM	-	-	0.90 nM
Inhibitor	20 nM	10.2	<b>10.10 nM</b>	-	-	9.19 nM
Substrate	-	-	-	11 $\mu$ M	2	<b>0.99 <math>\mu</math>M</b>
<i>Total</i>		20.2			22.2	

### Data Analysis 3A: Preincubation Time-Dependent Inhibition Without Dilution for Two-Step Irreversible Covalent Inhibition

Time-dependent product formation is fitted to a straight line for each inhibitor concentration to obtain the enzyme activity after preincubation  $v_{t'}$  (in AU/s) from the linear (initial) slope (Fig. 15A and 15B, left). It is important that the incubation time be relatively short ( $t < 0.1t'_{1/2}$ ) to minimize artifacts caused by significant formation of covalent adduct EI\* after substrate addition ( $\Delta EI^*$ ) because  $v_{t'}$  should reflect the enzyme activity at the end of preincubation. As a rule of thumb, incubation time  $t$  should be much shorter than the shortest preincubation time  $t'$ . A short incubation time may result in insufficient product formation for a robust signal, which can be resolved by increasing the incubation time and obtaining enzyme activity  $v_{t'}$  from the initial velocity of the exponential association progress curve, provided that the assay is compatible with progress curve analysis (continuous read-out) (Fig. 15B, right). Enzyme activity after preincubation  $v_{t'}$  is fitted to bounded exponential decay Equation V (Fig. 14D) (constraining  $v_s = 0$ ) for each inhibitor concentration to obtain the observed rate of reaction completion  $k_{obs}$  from enzyme activity without preincubation (Y-intercept at  $v_i$ ) to reaching the final enzyme inactivation (plateau at  $v_s = 0$ ) (Fig. 15C). Enzyme activity without preincubation in presence of inhibitor  $v_i$  is lower than the uninhibited enzyme activity  $v^{ctrl}$  for two-step (ir)reversible inhibitors, because  $v_i$  reflects the rapid noncovalent equilibrium ( $K_i^{app}$ ) after substrate addition (Copeland, 2013b). The plot of inhibitor concentration-dependent  $k_{obs}$  reaches maximum rate of inactivation  $k_{inact}$  in presence of saturating inhibitor concentration ( $[I] \gg K_I$ ) with the Y-intercept at  $k_{ctrl} = 0$  when uninhibited enzyme activity  $v^{ctrl}$  is independent of preincubation time (Fig. 15D). Inhibitor concentrations should correspond with the inhibitor concentration *during preincubation* (rather than after substrate addition). Correction of inactivation constant  $K_I$  for substrate competition is not necessary because preincubation is performed in absence of substrate.

#### Warnings and Remarks

The rapid noncovalent  $E + I \rightleftharpoons EI$  equilibrium does not significantly contribute to inhibition at non-saturating inhibitor concentrations ( $[I] \ll K_i^{app}$ ), resulting in one-step binding behavior (Fig. 3F). This will be apparent from the observation that initial velocity  $v_i$  is independent of inhibitor concentration ( $v_i = v^{ctrl}$ ) along with a linear plot of  $k_{obs}$  against  $[I]$ . This is resolved either by increasing the inhibitor concentration or performing *Data Analysis 3B*. Increasing the substrate concentration can resolve issues with assay sensitivity associated with short incubation times, as this will result in a higher product signal. However, substrate addition in a relatively large volume ( $V_{sub} > 0.1V_{t'}$ ) and/or addition of a competitive substrate concentration ( $[S] > 0.1K_M$ ) causes (partial) disruption of the reversible equilibrium, although this does not affect the accuracy of  $k_{obs}$  for irreversible inhibitors. In fact, disruption of the noncovalent complex can be employed to detect covalent adduct formation of two-step irreversible inhibitors that exhibit tight-binding behavior (Copeland, 2013c; Murphy, 2004) resulting from very potent noncovalent inhibition, as will be discussed in *Method IV*.



**Figure 15** Data Analysis 3A: Preincubation time–dependent inhibition without dilution for two-step irreversible covalent inhibition. Simulated with **KinGen** (A–D) or **KinDeg** (E–I) for inhibitor **C** with 1  $\mu\text{M}$  enzyme and 100 nM substrate **S1**. **(A)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after substrate addition for two-step irreversible covalent inhibition. **(B)** Time-dependent product formation after preincubation in absence of inhibitor  $F^{\text{ctrl}}$  or in presence of inhibitor ( $t' = 1800$  s). Left: Enzyme activity after preincubation  $v_I$  is obtained from the linear slope if the incubation time is relatively short ( $t \ll t'$ ): gray area is excluded from the fit. Right: Enzyme activity after preincubation  $v_I$  is obtained from the initial velocity of the exponential association progress curve of each inhibitor concentration. **(C)** Preincubation time–dependent enzyme activity  $v_I$  is fitted to Equation V (Fig. 14D) (constraining  $v_s = 0$ ) for each inhibitor concentration to obtain observed rates of inactivation  $k_{\text{obs}}$ . Alternatively,  $v_I$  can be normalized to a fraction of the uninhibited enzyme activity  $v^{\text{ctrl}}$ . **(D)** Inhibitor concentration–dependent  $k_{\text{obs}}$  reaches  $k_{\text{inact}}$  at saturating inhibitor concentration ( $k_{\text{max}} = k_{\text{inact}}$ ). Half-maximum  $k_{\text{obs}} = 1/2 k_{\text{inact}}$  is reached when inhibitor concentration equals the inactivation constant  $K_1$ ; no correction for substrate competition because  $v_I$  reflects the enzyme activity after preincubation in absence of competing substrate. **(E)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after substrate addition for two-step irreversible covalent inhibition with spontaneous enzyme degradation/denaturation. Simulated with  $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}} = 0.0003 \text{ s}^{-1}$ . **(F)** Uninhibited enzyme activity after preincubation  $v_I^{\text{ctrl}}$  is not linear. Preincubation time–dependent enzyme activity  $v_I$  is fitted to Equation V (Fig. 14D) (constraining  $v_s = 0$ ) for each inhibitor concentration to obtain

*(legend continues on next page)*

observed rates of inactivation  $k_{\text{obs}}$ , as well as fitting uninhibited activity  $v_t^{\text{ctrl}}$  to obtain the rate of nonlinearity  $k_{\text{ctrl}}$ . **(G)** Inhibitor concentration-dependent  $k_{\text{obs}}$  with spontaneous enzyme degradation increases with  $k_{\text{ctrl}}$  but the span from  $k_{\text{min}}$  ( $= k_{\text{ctrl}}$ ) to  $k_{\text{max}}$  ( $= k_{\text{inact}} + k_{\text{ctrl}}$ ) still equals  $k_{\text{inact}}$ . Fit with algebraic correction for nonlinearity (black line,  $k_{\text{ctrl}} > 0$ ). Ignoring the nonlinearity (gray line, constrain  $k_{\text{ctrl}} = 0$ ) results in underestimation of  $K_i$  (overestimation of potency) and overestimation of  $k_{\text{inact}}$ . **(H)** Normalized enzyme activity  $v_t/v_t^{\text{ctrl}}$  is fitted to Equation V (Fig. 14D) (constraining  $v_s = 0$ ) for each inhibitor concentration to obtain corrected observed rates of inactivation  $k_{\text{obs}}$ . **(I)** Inhibitor concentration-dependent  $k_{\text{obs}}$  has been corrected for enzyme degradation by fitting normalized enzyme activity  $v_t/v_t^{\text{ctrl}}$  and does not require further corrections.

Uninhibited enzyme activity  $v_t^{\text{ctrl}}$  decreases when preincubation is long enough for significant spontaneous enzyme degradation ( $t' \gg 0.1t_{1/2}$ ) (Fig. 15F). A simple algebraic correction for spontaneous enzyme degradation results in good estimates for  $k_{\text{inact}}$  and  $K_i$  if all enzyme species have the same first-order enzymatic degradation rate ( $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}}$ ) (Fig. 15G). Alternatively, normalizing the enzyme activity  $v_t$  to uninhibited enzyme activity  $v_t^{\text{ctrl}}$  at each preincubation time corrects for enzyme degradation (Fig. 15H), and  $k_{\text{obs}}$  obtained from normalized enzyme activity  $v_t/v_t^{\text{ctrl}}$  results in good estimates of  $k_{\text{inact}}$  and  $K_i$  without further correction (Fig. 15I).

### Two-Step Irreversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol III* that has been processed according to *Basic Data Analysis Protocol 3* for two-step irreversible covalent inhibitors.

1. Plot  $v_t$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 14C/Fig. 15C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations. Check whether the uninhibited enzyme activity is independent of preincubation time ( $v_0^{\text{ctrl}} = v_t^{\text{ctrl}}$ , Fig. 15C): an algebraic correction for enzyme instability ( $k_{\text{ctrl}} > 0$ , Fig. 15F) can be performed in step 4 of this protocol by accounting for nonlinearity in the uninhibited control in the secondary  $k_{\text{obs}}$  plot (Fig. 15G). Alternatively, proceed to *Alternative Data Analysis Protocol 3Aii* to correct for enzyme instability ( $v_0^{\text{ctrl}} > v_t^{\text{ctrl}}$ ) by normalization of the enzyme activity  $v_t/v_t^{\text{ctrl}}$  (Fig. 15H-I).

2. Fit  $v_t$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Fit the mean and standard deviation of  $v_t$  against preincubation time  $t'$  (Fig. 15C/F) to Equation V. Constrain  $v_s =$  value in fully inhibited control to obtain the observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i$  (Y-intercept) to full inactivation (Plateau = 0). A lack of initial noncovalent complex ( $v_i = v_0^{\text{ctrl}}$ ) is indicative of one-step binding behavior.

$$v_{t'} = v_s + (v_i - v_s) e^{-k_{\text{obs}}t'}$$

#### Equation V

Equation V for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y =$  preincubation time-dependent product formation velocity  $v_t$  (in AU/s),  $X =$  preincubation time  $t'$  (in s), and  $\text{Plateau} =$  final velocity in the fully inhibited control  $v_s$  (in AU/s) to find  $Y_0 =$  Y-intercept = initial velocity  $v_i$  (in AU/s) and  $k =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

3. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) during preincubation (before addition of substrate) on the X-axis (Fig. 15D/G). The plot of  $k_{\text{obs}}$  against  $[I]$  should reach a maximum  $k_{\text{obs}}$  at saturating

inhibitor concentration. Note that a linear curve is indicative of one-step binding behavior at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$  in Fig. 3F) with  $v_i = v_0^{\text{ctrl}}$  (shared Y-intercept in the previous step). Proceed to *Basic Data Analysis Protocol 3Bi* step 4 after it has been validated that the linear curve is not resultant from saturating inhibitor concentrations ( $[I] \gg 10K_I$  in Fig. 3G) as identified by  $v_i < v_0^{\text{ctrl}}$ , by repeating the measurement with lower inhibitor concentrations.

4. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{inact}}$  and  $K_I$ .

Fit  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) against inhibitor concentration *during preincubation* (in M) to Equation XV to obtain maximum inactivation rate constant  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and inactivation constant  $K_I$  (in M). Constrain  $k_{\text{ctrl}} = k_{\text{obs}}$  of the uninhibited control (Fig. 15G). Inactivation constant  $K_I$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate irreversible covalent inhibitor potency  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) with propagation of error with *Sample Calculation 2*.

$$k_{\text{obs}} = k_{\text{ctrl}} + \frac{k_{\text{inact}} [I]}{K_I + [I]}$$

**Equation XV**

Equation XV for nonlinear regression of user-defined explicit equation  $Y = Y_0 + ((k_{\text{max}} * X) / (K_I + X))$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) and  $X =$  inhibitor concentration during preincubation (in M) to find  $Y_0 =$  rate of nonlinearity in uninhibited control  $k_{\text{ctrl}}$  (in  $\text{s}^{-1}$ ),  $k_{\text{max}} =$  maximum reaction rate  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and  $K_I =$  Inactivation constant  $K_I$  (in M).

5. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental read-out to the product formation simulated with scripts **KinGen** and **KinDeg** (using experimental rate constant  $k_{\text{inact}} = k_5$ ) to confirm that the calculated kinetic constants are in accordance with the experimental data. Also perform simulations with **KinVol** and **KinVolDeg** to confirm that addition of substrate does not significantly affect the noncovalent interactions.

## Two-Step Irreversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol III* that has been processed according to *Basic Data Analysis Protocol 3* for two-step irreversible covalent inhibitors.

1. Plot  $v_t$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 14C/Fig. 15C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations.

2. Normalize  $v_t$  to obtain  $v_t/v^{\text{ctrl}}$ .

Normalize  $v_t$  (in AU/s) of each inhibitor concentration and the controls to lowest value = 0 (or full inhibition control) and highest value = uninhibited product formation  $v_t^{\text{ctrl}}$  (in AU/s) to obtain normalized enzyme activity  $v_t/v^{\text{ctrl}}$  (Fig. 15H). Perform this correction *separately* for each preincubation time.

3. Plot and fit  $v_t/v^{\text{ctrl}}$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Plot the mean and standard deviation of  $v_t/v^{\text{ctrl}}$  on the Y-axis against preincubation time  $t'$  (in s) on the X-axis (Fig. 15H). Fit to exponential decay Equation XVI to obtain

**ALTERNATIVE  
DATA  
ANALYSIS  
PROTOCOL 3Aii**

**Mons et al.**

**49 of 85**

$k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i/v_0^{\text{ctrl}}$  to full inactivation (Plateau = 0). A lack of initial noncovalent complex ( $v_i/v_0^{\text{ctrl}} = 1$ ) is indicative of one-step binding behavior.

$$\left(\frac{v_{t'}}{v_{t'}^{\text{ctrl}}}\right) = \left(\frac{v_i}{v_0^{\text{ctrl}}}\right) e^{-k_{\text{obs}}t'}$$

**Equation XVI**

Equation XVI for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y =$  normalized preincubation time-dependent product formation velocity  $v_{t'}/v_{t'}^{\text{ctrl}}$  (unitless),  $X =$  preincubation time  $t'$  (in s), and  $\text{Plateau} =$  normalized final velocity  $v_s/v_s^{\text{ctrl}} = 0$  (unitless) to find  $Y_0 =$   $Y$ -intercept = normalized initial velocity  $v_i/v_0^{\text{ctrl}}$  (unitless) and  $k =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

4. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) *during preincubation* (before addition of substrate) on the X-axis (Fig. 15I). The plot of  $k_{\text{obs}}$  against  $[I]$  should have a Y-intercept = 0, and reach a maximum  $k_{\text{obs}}$  at saturating inhibitor concentration. Note that a linear curve is indicative of one-step binding behavior at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$  in Fig. 3F) with  $v_i = v_0^{\text{ctrl}}$  (shared Y-intercept = 1 in the previous step). Proceed to *Basic Data Analysis Protocol 3Bii* step 5 after it has been validated that the linear curve is not resultant from saturating inhibitor concentrations ( $[I] \gg 10K_I$  in Fig. 3G) as identified by  $v_i \ll v_0^{\text{ctrl}}$  (shared Y-intercept = 0 in the previous step), by repeating the measurement with lower inhibitor concentrations.

5. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{inact}}$  and  $K_I$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* to Equation XVII to obtain maximum inactivation rate constant  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and inactivation constant  $K_I$  (in M) (Fig. 15I). Do not correct for enzyme instability ( $k_{\text{ctrl}} > 0$ ), as this correction has already been performed by normalizing  $v_{t'}$  to  $v_{t'}/v_{t'}^{\text{ctrl}}$  in step 2 of this protocol. Inactivation constant  $K_I$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate irreversible covalent inhibitor potency  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) with propagation of error with *Sample Calculation 2*.

$$k_{\text{obs}} = \frac{k_{\text{inact}} [I]}{K_I + [I]}$$

**Equation XVII**

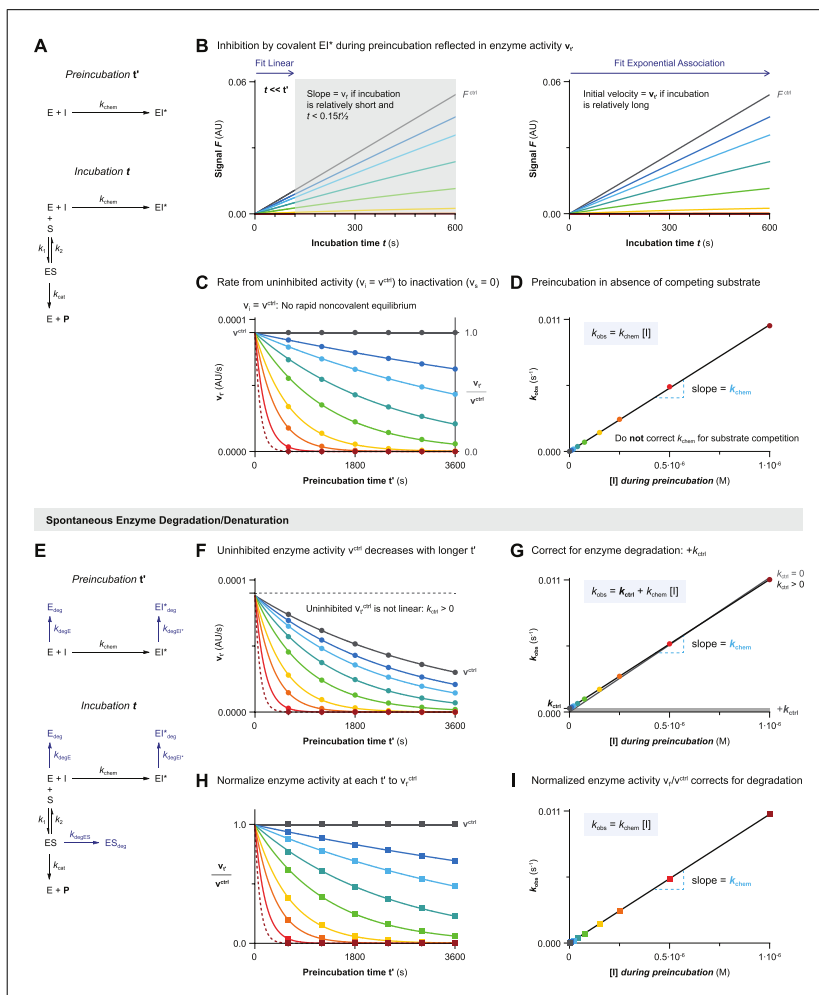
Equation XVII for nonlinear regression of user-defined explicit equation  $Y = Y_0 + ((k_{\text{max}} * X) / (K_I + X))$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ),  $X =$  inhibitor concentration during preincubation (in M) and  $Y_0 = 0$  (in  $\text{s}^{-1}$ ) to find  $k_{\text{max}} =$  maximum reaction rate  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and  $K_I =$  Inactivation constant  $K_I$  (in M).

6. *Optional:* Validate experimental kinetic parameters with kinetic simulations by proceeding to *Basic Data Analysis Protocol 3Ai*, step 5.

### Data Analysis 3B: Preincubation Time-Dependent Inhibition Without Dilution for One-Step Irreversible Covalent Inhibition

Time-dependent product formation is fitted to a straight line for each inhibitor concentration to obtain the enzyme activity after preincubation  $v_{t'}$  (in AU/s) from the linear slope (Fig. 16A and 16B, left). Incubation must be short enough to minimize formation of covalent adduct  $EI^*$  after substrate addition ( $t \ll t/2$ ); otherwise  $k_{\text{chem}}$  will be overestimated. Similar to *Data Analysis 3A*, preincubation-dependent enzyme activity  $v_{t'}$  can





**Figure 16** Data Analysis 3B: Preincubation time–dependent inhibition without dilution for one-step irreversible covalent inhibition. Simulated with **KinGen** (A–D) or **KinDeg** (E–I) for inhibitor **D** with 1  $\mu\text{M}$  enzyme and 100 nM substrate **S1**. **(A)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after substrate addition for one-step irreversible covalent inhibition. **(B)** Time-dependent product formation after preincubation in absence of inhibitor  $F^{\text{ctrl}}$  or in presence of inhibitor ( $t' = 1800$  s). Left: Enzyme activity after preincubation  $v_i$  is obtained from the linear slope if the incubation time is relatively short ( $t \ll t'$ ): gray area is excluded from the fit. Right: Enzyme activity after preincubation  $v_i$  is obtained from the initial velocity of the exponential association progress curve of each inhibitor concentration. **(C)** Preincubation time–dependent enzyme activity  $v_i$  is fitted to Equation V (Fig. 14D) (constraining  $v_s = 0$ ) for each inhibitor concentration to obtain observed rates of inactivation  $k_{\text{obs}}$ .  $v_i = v^{\text{ctrl}}$  for one-step irreversible inhibitors and two-step irreversible inhibitors at non-saturating concentrations ( $[I] \ll K_i^{\text{app}}$ ). Alternatively,  $v_i$  can be normalized to a fraction of the uninhibited enzyme activity  $v_i^{\text{ctrl}}$ . **(D)** Inhibitor concentration–dependent  $k_{\text{obs}}$  increases linearly with inhibitor concentration, with  $k_{\text{chem}}$  as the slope. No correction for substrate competition because  $v_i$  reflects the enzyme activity after preincubation in absence of competing substrate. **(E)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after substrate addition for one-step irreversible covalent inhibition with spontaneous enzyme degradation/denaturation. Simulated with  $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}^*} = 0.0003 \text{ s}^{-1}$ . **(F)** Uninhibited enzyme activity after preincubation  $v_i^{\text{ctrl}}$  is not linear:  $k_{\text{ctrl}} > 0$ . Preincubation time–dependent enzyme activity  $v_i$  is fitted to Equation V (Fig. 14D) (constraining  $v_s = 0$  and shared *(legend continues on next page)*

value for  $v_i$  = uninhibited enzyme activity without preincubation  $v_0^{\text{ctrl}}$  for each inhibitor concentration to obtain observed rates of inactivation  $k_{\text{obs}}$ , as well as fitting uninhibited activity  $v_i^{\text{ctrl}}$  to obtain the rate of nonlinearity  $k_{\text{ctrl}}$ . **(G)** Inhibitor concentration–dependent  $k_{\text{obs}}$  with spontaneous enzyme degradation/denaturation increases by  $k_{\text{ctrl}}$ . Fit with algebraic correction for nonlinearity (black line,  $k_{\text{ctrl}} > 0$ ) or ignoring nonlinearity (gray line, constrain  $k_{\text{ctrl}} = 0$ ). Ignoring the nonlinearity (assuming Y-intercept = 0) results in overestimation of  $k_{\text{chem}}$  (steeper slope). **(H)** Normalized enzyme activity  $v_i/v_0^{\text{ctrl}}$  is fitted to Equation V (Fig. 14D) (constraining  $v_s = 0$  and Y-intercept =  $v_i/v_0^{\text{ctrl}} = 1$ ) for each inhibitor concentration to obtain corrected observed rates of inactivation  $k_{\text{obs}}$ . **(I)** Inhibitor concentration-dependent  $k_{\text{obs}}$  has been corrected for enzyme degradation/denaturation by fitting normalized enzyme activity  $v_i/v_0^{\text{ctrl}}$  and does not require further corrections.

also be obtained from the initial velocity of the exponential association progress curve, provided that the read-out is continuous (Fig. 16B, right). Enzyme activity after preincubation  $v_i$  is fitted to bounded exponential decay Equation V (Fig. 14D) to obtain observed rate of reaction completion  $k_{\text{obs}}$  from uninhibited enzyme activity without preincubation (Y-intercept at  $v_i = v^{\text{ctrl}}$ ) to reaching the final enzyme inactivation (constraining  $v_s = 0$ ) (Fig. 16C). Inhibited enzyme activity without preincubation is equal to uninhibited enzyme activity ( $v_i = v^{\text{ctrl}}$ ), as rapid noncovalent inhibitor binding does not contribute to enzyme inhibition by one-step irreversible inhibitors. The slope of the linear plot of  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* is equal to  $k_{\text{chem}}$  (Fig. 16D), which should not be corrected for substrate competition as preincubation is performed in absence of competing substrate.

### Warnings and remarks

Substrate addition in a relatively large volume ( $V_{\text{sub}} > 0.1V_i$ ) and/or addition of a competitive substrate concentration ( $[S] > 0.1K_M$ ) does not significantly affect the accuracy of  $k_{\text{obs}}$  because one-step irreversible inhibition does not involve a rapid noncovalent equilibrium that can be disrupted (also see *Method IV*). Increasing the substrate concentration can resolve issues with assay sensitivity: higher substrate concentration results in a higher product concentration after the same incubation time ( $v^{\text{ctrl}} = V_{\text{max}}[S]/([S]+K_M)$ ), which in turn will result in a better signal to noise ratio.

Uninhibited enzyme activity  $v^{\text{ctrl}}$  decreases with longer preincubation due to spontaneous enzyme degradation (Fig. 16E and 16F). This especially affects assays where preincubation is long enough for significant enzyme degradation ( $t' > 0.1t_{1/2}$ ). Algebraic correction for spontaneous enzyme degradation ( $k_{\text{degE}} = k_{\text{degES}}$ ) in the secondary  $k_{\text{obs}}$  plot is relatively simple (Fig. 16G). Alternatively, correction for enzyme degradation is performed by normalizing enzyme activity  $v_i$  to uninhibited enzyme activity  $v_i^{\text{ctrl}}$  at each preincubation time (Fig. 16H and 16I). Stabilization of enzyme upon inhibitor binding ( $k_{\text{degEI}^*} < k_{\text{degE}}$ ) does not affect  $k_{\text{obs}}$ , as EI\* formation is already irreversible thus removing the species from the available pool of catalytic enzyme.

### One-Step Irreversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol III* that has been processed according to *Basic Data Analysis Protocol 3* for one-step irreversible covalent inhibitors and two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ).

1. Plot  $v_i$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_i$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 14C/Fig. 16C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations. Check whether the uninhibited enzyme activity is independent of

preincubation time ( $v_0^{\text{ctrl}} = v_t^{\text{ctrl}}$ , Fig. 16C): an algebraic correction for enzyme instability ( $k_{\text{ctrl}} > 0$ , Fig. 16F) can be performed in step 4 of this protocol by accounting for nonlinearity in the uninhibited control in the secondary  $k_{\text{obs}}$  plot (Fig. 16G). Alternatively, proceed to *Alternative Data Analysis Protocol 3Bii* to correct for enzyme instability ( $v_0^{\text{ctrl}} > v_t^{\text{ctrl}}$ ) by normalization of the enzyme activity  $v_t/v_t^{\text{ctrl}}$  (Fig. 16H-I).

2. Fit  $v_t$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Fit the mean and standard deviation of  $v_t$  against preincubation time  $t'$  (Fig. 16C/F) to Equation V. Constrain  $v_s =$  value in fully inhibited control to obtain the observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i$  (Y-intercept) to full inactivation (Plateau = 0). A lack of initial noncovalent complex ( $v_i = v_0^{\text{ctrl}}$ ) is indicative of one-step binding behavior.

$$v_t = v_s + (v_i - v_s) e^{-k_{\text{obs}}t'}$$

#### Equation V

Equation V for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y =$  preincubation time-dependent product formation velocity  $v_t$  (in AU/s),  $X =$  preincubation time  $t'$  (in s), and  $\text{Plateau} =$  final velocity in the fully inhibited control  $v_s$  (in AU/s) to find  $Y_0 =$  Y-intercept = initial velocity  $v_i =$  uninhibited initial velocity  $v_0^{\text{ctrl}}$  (in AU/s, shared value) and  $k =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

3. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) *during preincubation* (before addition of substrate) on the X-axis (Fig. 16D/G). The plot of  $k_{\text{obs}}$  against inhibitor concentration  $[I]$  is linear for one-step irreversible inhibitors and for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$ ).

4. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{chem}}$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* (in M) to Equation XVIII to obtain inhibitor potency  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) from the linear slope. Constrain Y-intercept  $k_{\text{ctrl}} = k_{\text{obs}}$  of the uninhibited control (Fig. 16G). Inhibitor potency  $k_{\text{chem}}$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ) with propagation of error with *Sample Calculation 9*.

$$k_{\text{obs}} = k_{\text{ctrl}} + k_{\text{chem}} [I]$$

#### Equation XVIII

Equation XVIII for nonlinear regression of straight line  $Y = \text{YIntercept} + \text{Slope} * X$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) and  $X =$  inhibitor concentration during preincubation (in M) to find  $\text{YIntercept} =$  rate of nonlinearity in uninhibited control  $k_{\text{ctrl}}$  (in  $\text{s}^{-1}$ ) and  $\text{Slope} =$  inactivation rate constant  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ )

5. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental read-out to the product formation simulated with scripts **KinGen** and **KinDeg** (using experimental rate constant  $k_{\text{chem}} = k_3$ ) to confirm that the calculated kinetic constants are in accordance with the experimental data. Also perform simulations with **KinVol** and **KinVolDeg** to confirm that addition of substrate does not significantly affect the reaction rates by dilution and/or competition.

### One-Step Irreversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol III* that has been processed according to *Basic Data Analysis Protocol 3* for one-step irreversible covalent inhibitors and two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ).

1. Plot  $v_t'$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t'$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 14C/Fig. 16C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations.

2. Normalize  $v_t'$  to obtain  $v_t'/v_t'^{\text{ctrl}}$ .

Normalize  $v_t'$  (in AU/s) of each inhibitor concentration and the controls to lowest value = 0 (or full inhibition control) and highest value = uninhibited product formation  $v_t'^{\text{ctrl}}$  (in AU/s) to obtain normalized enzyme activity  $v_t'/v_t'^{\text{ctrl}}$  (Fig. 16H). Perform this correction *separately* for each preincubation time.

3. Plot and fit  $v_t'/v_t'^{\text{ctrl}}$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Plot the mean and standard deviation of  $v_t'/v_t'^{\text{ctrl}}$  on the Y-axis against preincubation time  $t'$  (in s) on the X-axis (Fig. 16H). Fit to exponential decay Equation XVI to obtain  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i/v_0^{\text{ctrl}}$  to full inactivation (Plateau = 0). A lack of initial noncovalent complex ( $v_i/v_0^{\text{ctrl}} = 1$ ) is indicative of one-step binding behavior.

$$\left( \frac{v_t'}{v_t'^{\text{ctrl}}} \right) = e^{-k_{\text{obs}}t'}$$

Equation XVI

Equation XVI for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y$  = normalized preincubation time-dependent product formation velocity  $v_t'/v_t'^{\text{ctrl}}$  (unitless),  $X$  = preincubation time  $t'$  (in s),  $Y_0$  = Y-intercept = normalized initial velocity  $v_i/v_0^{\text{ctrl}} = 1$  (unitless), and Plateau = normalized final velocity  $v_s/v_s^{\text{ctrl}} = 0$  (unitless) to find  $k$  = observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

4. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) *during preincubation (before addition of substrate)* on the X-axis (Fig. 16I). The plot of  $k_{\text{obs}}$  against inhibitor concentration  $[I]$  is linear for one-step irreversible inhibitors and for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$ ).

5. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{chem}}$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* to Equation XIX to inhibitor potency  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) from the linear slope (Fig. 16I). Do not correct for enzyme instability ( $k_{\text{ctrl}} > 0$ ), as this correction has already been performed by normalizing  $v_t'$  to  $v_t'/v_t'^{\text{ctrl}}$  in step 2 of this protocol. Inhibitor potency  $k_{\text{chem}}$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ) with propagation of error with *Sample Calculation 9*. Alternatively, inhibitor potency  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ )

or  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) can be directly calculated from a single  $k_{\text{obs}}$  ( $\text{s}^{-1}$ ) and  $[I]$  (in  $\text{M}$ ) with *Sample Calculation 10*.

$$k_{\text{obs}} = k_{\text{chem}} [I]$$

#### Equation XIX

Equation XIX for nonlinear regression of straight line  $Y = Y\text{Intercept} + \text{Slope} \cdot X$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ),  $X =$  inhibitor concentration during preincubation (in  $\text{M}$ ), and  $Y\text{Intercept} = 0$  (in  $\text{s}^{-1}$ ) to find  $\text{Slope} =$  inactivation rate constant  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ).

6. *Optional*: Validate experimental kinetic parameters with kinetic simulations by proceeding to *Basic Data Analysis Protocol 3Bi* step 5.

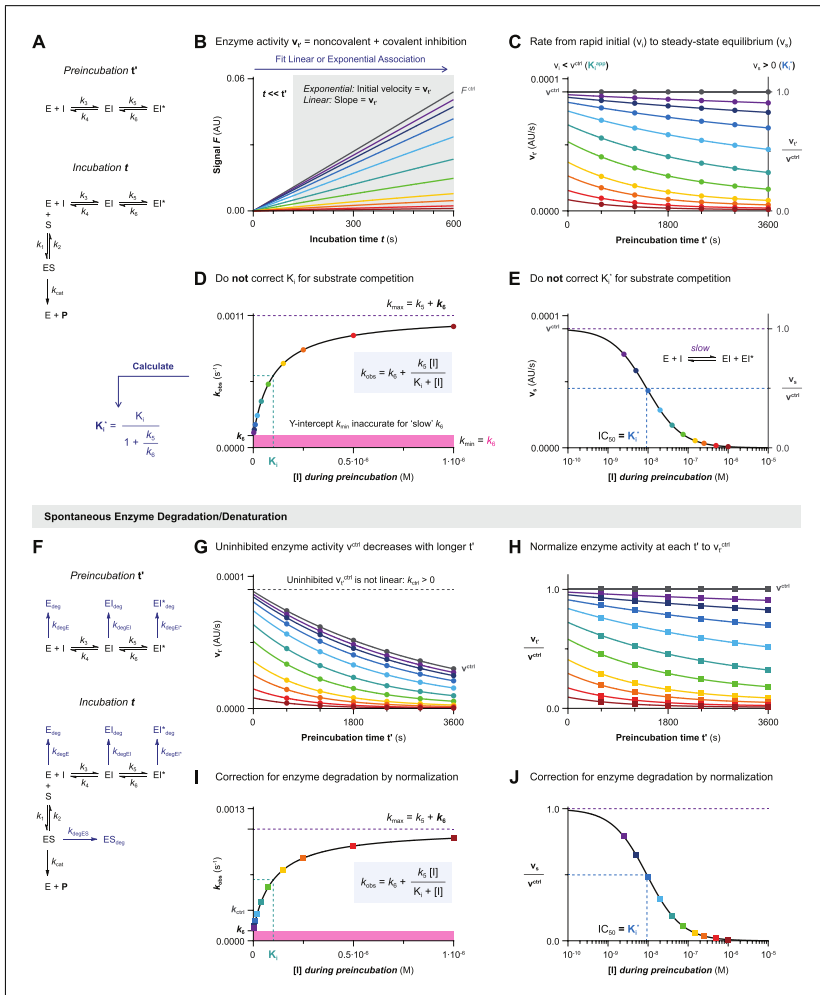
### Data Analysis 3C: Preincubation Time–Dependent Inhibition Without Dilution for Reversible Covalent Inhibition

Time-dependent product formation is fitted to a straight line for each inhibitor concentration to obtain the enzyme activity after preincubation  $v_t'$  (in  $\text{AU/s}$ ) from the linear slope (Fig. 17A and 17B). Again, it is important that the incubation time be much shorter than the shortest preincubation time  $t'$  ( $t \ll t'$ ), but enzyme activity  $v_t'$  can also be calculated from the initial velocity of the exponential association progress curve, provided that the assay is compatible with progress curve analysis (continuous read-out). Enzyme activity after preincubation  $v_t'$  is fitted to bounded exponential decay Equation V (Fig. 14D) for each inhibitor concentration to obtain observed rate of reaction completion  $k_{\text{obs}}$  from rapid noncovalent equilibrium ( $Y$ -intercept at  $v_i < v^{\text{ctrl}}$ ) to slowly reaching steady-state equilibrium (plateau at  $v_s > 0$ ) (Fig. 17C). Enzyme activity without preincubation in presence of inhibitor  $v_i$  is lower than the uninhibited enzyme activity  $v^{\text{ctrl}}$  for two-step (ir)reversible inhibitors because  $v_i$  reflects the rapid noncovalent equilibrium ( $K_i^{\text{app}}$ ) after substrate addition (Copeland, 2013b). Contrary to irreversible inhibition, the plateau ( $v_s > 0$ ) does not approximate enzyme inactivation but reaches the steady-state equilibrium ( $K_i^*$ ) instead. Steady-state inhibition constant  $K_i^*$  can be calculated from the fitted values of  $K_i$ ,  $k_5$  and  $k_6$  (Fig. 17D), but this is not the preferred approach as a small error in  $k_6$  has huge implications for the calculation of  $K_i^*$  (as illustrated in Fig. 9). Generally, more reliable estimates of the steady-state inhibition constant  $K_i^*$  are generated from the dose-response curve of steady-state velocity  $v_s$  against inhibitor concentration *during preincubation* (Fig. 17E).

#### Warnings and remarks

Steady-state inhibition constant  $K_i^*$  reflects the reversible  $E + I \rightleftharpoons EI + EI^*$  equilibrium that can be disrupted by substrate addition in a relatively large volume ( $V_{\text{sub}} > 0.1V_t'$ ) and/or addition of a competitive substrate concentration ( $[S] > 0.1K_M$ ). Simulations with high substrate concentration ( $[S] = 10K_M$ ) show that the  $\text{IC}_{50}$  of the dose-response curve for steady-state velocity  $v_s$  was slightly higher than steady-state inhibition constant  $K_i^*$ , but still significantly lower than  $K_i^{\text{app}}$ , as covalent dissociation will not be significant as long as the incubation time is significantly shorter than the dissociation half-life ( $t \ll t_{1/2\text{diss}}$ ). Altogether, fitting exponential association rather than increasing the substrate concentration is the desired solution to resolve issues with assay sensitivity associated with short incubation times. Alternatively, reasonable estimates of the steady-state inhibition constant  $K_i^*$  were obtained from the endpoint preincubation time–dependent potency  $\text{IC}_{50}(t')$  with minimal substrate competition ( $[S] \ll K_M$ ) and preincubation times exceeding the required time to reach reaction completion at all inhibitor concentrations ( $t' > 5t_{1/2}$ ).

As mentioned before, spontaneous loss of enzyme activity due to first-order degradation and/or denaturation of enzyme species ( $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}}$ ) results in a preincubation time–dependent decrease of uninhibited enzyme activity  $v^{\text{ctrl}}$  (Fig. 17E and 17F). The biggest advantage of *Method III* over *Method I (Data Analysis 1C)* is that it is



**Figure 17** Data Analysis 3C: Preincubation time–dependent inhibition without dilution for two-step reversible covalent inhibition. Simulated with **KinGen** (A-E) or **KinDeg** (F-J) for inhibitor **B** with 1 pM enzyme and 100 nM substrate **S1**. **(A)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after substrate addition for two-step reversible covalent inhibition. **(B)** Time-dependent product formation after preincubation in absence of inhibitor  $F^{ctrl}$  or in presence of inhibitor ( $t' = 1800$  s). Enzyme activity after preincubation  $v_t$  is obtained from the linear slope if the incubation time is relatively short ( $t \ll t'$ ): gray area is excluded from the fit. Alternatively, enzyme activity after preincubation  $v_t$  is obtained from the initial velocity of the exponential association progress curve of each inhibitor concentration. **(C)** Preincubation time–dependent enzyme activity  $v_t$  is fitted to Equation V (Fig. 14D) for each inhibitor concentration to obtain observed rates of inactivation  $k_{obs}$  and steady-state velocity  $v_s$  (plateau  $> 0$ ). Alternatively,  $v_t$  can be normalized to a fraction of the uninhibited enzyme activity  $v^{ctrl}$ . **(D)** Inhibitor concentration–dependent  $k_{obs}$  equals  $k_{max}$  at saturating inhibitor concentration ( $k_{max} = k_5 + k_6$ ) and approaches  $k_6$  in absence of inhibitor ( $k_{min} = k_6$ ). Half-maximum  $k_{obs} = k_{min} + \frac{1}{2}(k_{max} - k_{min}) = k_6 + \frac{1}{2}k_5$  is reached when inhibitor concentration equals the inhibition constant  $K_i$ . Steady-state inhibition constant  $K_i^*$  has to be calculated from the fitted values of  $k_5$ ,  $k_6$ , and  $K_i$ , thus being very sensitive to errors and (non)linearity in the uninhibited background (illustrated in Fig. 8G). No correction for substrate competition because  $v_t$  reflects the enzyme activity after preincubation in absence of competing substrate. **(E)** Steady-state inhibition constant  $K_i^*$  corresponds with the  $IC_{50}$  of steady-state velocity  $v_s$  obtained by fitting the dose-response curve to the Hill equation (Copeland, 2013e). (legend continues on next page)

No correction for substrate competition because  $v_t$  reflects the enzyme activity after preincubation in absence of competing substrate. (F) Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after substrate addition for two-step reversible covalent inhibition with spontaneous enzyme degradation. Simulated with  $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}} = k_{\text{degEI}^*} = 0.0003 \text{ s}^{-1}$ . (G) Uninhibited enzyme activity after preincubation  $v_t^{\text{ctrl}}$  is not linear. Fitting preincubation time-dependent enzyme activity  $v_t$  to Equation V (Fig. 14D) for each inhibitor concentration gives observed rates of inactivation  $k_{\text{obs}}$ , as well as the rate of nonlinearity  $k_{\text{ctrl}}$  for uninhibited activity  $v_t^{\text{ctrl}}$ . Inhibitor concentration-dependent  $k_{\text{obs}}$  and steady-state velocity  $v_s$  will be driven by spontaneous enzyme degradation if enzyme activity is not normalized. (H) Enzyme activity  $v_t$  is normalized to the uninhibited enzyme activity  $v_t^{\text{ctrl}}$  after each preincubation time before fitting to Equation V (Fig. 14D). (I) Inhibitor concentration-dependent  $k_{\text{obs}}$  has been corrected for enzyme degradation/denaturation by fitting normalized enzyme activity  $v_t/v_t^{\text{ctrl}}$  and does not require further corrections (even if  $k_{\text{ctrl}} > k_6$ ). (J) Steady-state velocity  $v_s$  has been corrected for enzyme degradation/denaturation by fitting normalized enzyme activity  $v_t/v_t^{\text{ctrl}}$  and does not require further corrections (even if  $k_{\text{ctrl}} > k_6$ ). Final velocity  $v_s$  obtained from uncorrected  $v_t$  is 'contaminated' by the contribution of irreversible inactivation to the time-dependent inhibition, and does not result in accurate estimates of steady-state inhibition constant  $K_i^*$  (illustrated in Fig. 8H).

possible to perform an algebraic correction for the enzyme instability in kinetic analysis of two-step reversible covalent inhibitors with *Data Analysis 3C*. Enzyme activity  $v_t$  is normalized to uninhibited enzyme activity  $v_t^{\text{ctrl}}$  at each preincubation time (Fig. 17G), and the normalized enzyme activity after preincubation  $v_t/v_t^{\text{ctrl}}$  is fitted to bounded exponential decay Equation V (Fig. 14D) for each inhibitor concentration to obtain observed rate of reaction completion  $k_{\text{obs}}$  and steady-state velocity  $v_s$ . Kinetic analysis of  $k_{\text{obs}}$  (Fig. 17H) and steady-state velocity  $v_s$  (Fig. 17I) against inhibitor concentration *during preincubation* result in good estimates of the kinetic parameters without further correction, even when  $k_{\text{ctrl}}$  is faster than the covalent dissociation rate  $k_6$  ( $k_{\text{ctrl}} > k_6$ ). We strongly advise that enzyme activity be normalized prior to analysis of reversible covalent inhibition even when  $k_{\text{ctrl}}$  is not directly obvious from the product formation in the uninhibited control  $v_t^{\text{ctrl}}$ .

## Two-Step Reversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol III* that has been processed according to *Basic Data Analysis Protocol 3* for two-step reversible covalent inhibitors.

1. Plot  $v_t$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 14C/Fig. 17C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations. Enzyme activity is never truly independent of preincubation time ( $v_0^{\text{ctrl}} > v_t^{\text{ctrl}}$ , Fig. 17G) and kinetic analysis of reversible inhibitors is very sensitive to small deviations (illustrated in Fig. 9). Therefore, correction for enzyme instability is always performed by normalization of the enzyme activity  $v_t/v_t^{\text{ctrl}}$  in the next step (Fig. 17F-J).

2. Normalize  $v_t$  to obtain  $v_t/v_t^{\text{ctrl}}$ .

Normalize  $v_t$  (in AU/s) of each inhibitor concentration and the controls to lowest value = 0 (or full inhibition control) and highest value = uninhibited product formation  $v_t^{\text{ctrl}}$  (in AU/s) to obtain normalized enzyme activity  $v_t/v_t^{\text{ctrl}}$  (Fig. 17H). Perform this correction *separately* for each preincubation time.

3. Plot and fit  $v_t/v_t^{\text{ctrl}}$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$  and  $v_s/v_s^{\text{ctrl}}$ .

Plot the mean and standard deviation of  $v_t/v_t^{\text{ctrl}}$  on the Y-axis against preincubation time  $t'$  (in s) on the X-axis (Fig. 17H). Fit to exponential decay Equation XX to obtain  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_t/v_0^{\text{ctrl}}$  reflecting rapid noncovalent equilibrium



(Y-intercept  $v_i/v_0^{\text{ctrl}} \leq 1$ ) to the final velocity  $v_s/v_s^{\text{ctrl}}$  reflecting steady-state equilibrium (Plateau  $v_s/v_s^{\text{ctrl}} \geq 0$ ).

$$\left(\frac{v_t'}{v_t'^{\text{ctrl}}}\right) = \left(\frac{v_s}{v_s^{\text{ctrl}}}\right) + \left(\frac{v_i}{v_0^{\text{ctrl}}} - \frac{v_s}{v_s^{\text{ctrl}}}\right) e^{-k_{\text{obs}}t'}$$

**Equation XX**

Equation XX for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y =$  normalized preincubation time-dependent product formation velocity  $v_t'/v_t'^{\text{ctrl}}$  (unitless),  $X =$  preincubation time  $t'$  (in s) to find  $Y_0 =$  Y-intercept = normalized initial velocity  $v_i/v_0^{\text{ctrl}}$  (unitless), Plateau = normalized final velocity  $v_s/v_s^{\text{ctrl}} = 0$  (unitless), and  $k =$  observed reaction rate  $k_{\text{obs}}$  (in  $s^{-1}$ ).

4. Plot and fit  $v_s/v_s^{\text{ctrl}}$  against  $[I]$  to obtain  $K_i^*$ .

Steady-state inhibition constant  $K_i^*$  (in M) can be calculated from  $v_s/v_s^{\text{ctrl}}$  (obtained in the previous step) reflecting remaining fractional enzyme activity after reaching the steady-state inhibitor equilibrium (reaction completion) (Fig. 17J). Plot the mean and standard deviation of  $v_s/v_s^{\text{ctrl}}$  on the Y-axis against inhibitor concentration (in M) *during preincubation* (before addition of substrate) on the X-axis (Fig. 17J), and fit the dose-response curve to four-parameter nonlinear regression Hill Equation XXI (Copeland, 2013e) to obtain steady-state inhibition constant  $K_i^*$  (in M). The maximum product formation velocity at reaction completion corresponds with the uninhibited enzyme activity  $v_s^{\text{ctrl}}/v_s^{\text{ctrl}} = 1$  and minimum velocity  $v_s^{\text{min}}/v_s^{\text{ctrl}} = 0$  for (background-)corrected enzyme activity in the full inhibition control. Steady-state equilibrium constant  $K_i^*$  (in M) does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate.

$$\left(\frac{v_s}{v_s^{\text{ctrl}}}\right) = \frac{1}{1 + \left(\frac{[I]}{K_i^*}\right)^h}$$

**Equation XXI**

Equation XXI for nonlinear regression of four-parameter dose-response equation  $Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + (X/\text{IC50})^{\text{HillSlope}})$  with  $Y =$  fractional steady-state product formation velocity  $v_s/v_s^{\text{ctrl}}$  (unitless),  $X =$  inhibitor concentration during preincubation (in M), Bottom = velocity in fully inhibited control  $v_s^{\text{min}}/v_s^{\text{ctrl}} = 0$  (unitless), and Top = uninhibited enzyme activity  $v_s^{\text{ctrl}}/v_s^{\text{ctrl}} = 1$  (unitless) to find Hill slope = Hill coefficient  $h$  (unitless) and IC50 = steady-state inhibition constant  $K_i^*$  (in M).

5. *Optional*: Plot and fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $K_i$ ,  $k_5$ , and  $k_6$ .

This is an optional data processing step to obtain kinetic parameters by fitting to the observed rate  $k_{\text{obs}}$  (obtained in *Data Analysis 3C*, step 3), and can be used to validate  $K_i^*$  values found in the previous step or to find values for  $k_5$  and  $k_6$  to use in kinetic simulations (next step in this protocol). Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $s^{-1}$ ) on the Y-axis against inhibitor concentration *during preincubation* (in M) on the X-axis (Fig. 17I). Exclude the uninhibited control ( $k_{\text{ctrl}} = 0$  for normalized enzyme activity) from the fit because Y-intercept =  $k_6$  rather than  $k_{\text{ctrl}}$ . Fit  $k_{\text{obs}}$  against inhibitor concentration to Equation XXII to obtain rate constants for the covalent association  $k_5$  (in  $s^{-1}$ ) and covalent dissociation  $k_6$  (in  $s^{-1}$ ) as well as noncovalent inhibition constant  $K_i$  (in M) reflecting the rapid (initial) noncovalent equilibrium. Noncovalent equilibrium constant  $K_i$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Proceed



to *Sample Calculation 8* to calculate steady-state inhibition constant  $K_i^*$  (in M) from experimental values of  $K_i$ ,  $k_5$ , and  $k_6$ .

$$k_{\text{obs}} = k_6 + \frac{k_5 [I]}{K_i + [I]}$$

**Equation XXII**

Equation XXII for nonlinear regression of user-defined explicit equation  $Y = Y_0 + ((k_{\text{max}} \cdot X) / (K_i + X))$  with  $Y$  = observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) and  $X$  = inhibitor concentration during preincubation (in M) to find  $Y_0$  = covalent dissociation rate constant  $k_6$  (in  $\text{s}^{-1}$ ),  $k_{\text{max}}$  = covalent association rate constant  $k_5$  (in  $\text{s}^{-1}$ ) and  $K_i$  = inhibition constant  $K_i$  (in M).

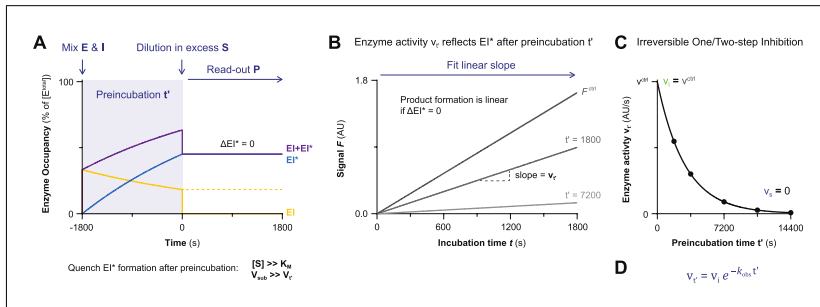
6. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental read-out to the product formation simulated with scripts **KinGen** and **KinDeg** to confirm that the calculated kinetic constants are in accordance with the experimental data. Also perform simulations with **KinVol** and **KinVolDeg** to confirm that addition of substrate does not significantly affect the noncovalent interactions/equilibria or reaction rates by dilution and/or competition. Experimental estimates of  $k_5$  and  $k_6$  are generated in the previous step of this protocol.

#### METHOD IV: PREINCUBATION TIME-DEPENDENT INHIBITION WITH DILUTION/COMPETITION

Preincubation time-dependent inhibition with dilution and/or competition is a variant of *Method III* reported for kinetic analysis of irreversible covalent inhibitors (Kitz & Wilson, 1962). Enzyme and inhibitor are preincubated in absence of competing substrate to form noncovalent EI complex and covalent EI\* adduct, followed by dilution in a 10-100-fold larger volume ( $V_{\text{sub}} \gg V_t$ ) and/or addition of a high concentration of competing substrate ( $[S] \gg K_M$ ) (Fig. 18A). The inhibitor concentration after substrate addition is far below the equilibrium concentration ( $[I]_t \ll 0.1K_i^{\text{app}}$ ), thereby inducing dissociation of inhibitor from the noncovalent inhibitor-enzyme complex EI and quenching the formation of covalent EI\* during incubation ( $\Delta[\text{EI}^*]_t = 0$ ). The approach is two-pronged: either dilution (reducing  $[I]_t$ ) or saturating substrate concentration (increasing  $K_i^{\text{app}}$  and decreasing  $k_{\text{chem}}^{\text{app}}$ ) can be sufficient as long as covalent EI\* adduct formation is fully quenched, for example by dissociation of noncovalent EI complex. Preincubation time-dependent product formation velocity  $v_t$  reflects the inhibition by covalent EI\* adduct formed during preincubation, and is calculated from the linear slope of product formation (Fig. 18B). Enzyme activity  $v_t$  decreases exponentially from 0% covalent adduct without preincubation ( $Y\text{-intercept} = v^{\text{ctrl}}$ ) to reach a plateau at 100% covalent adduct upon reaction completion ( $t' > 5t_{1/2}$ ) for irreversible covalent inhibitors (Fig. 18C). Observed rate of reaction completion  $k_{\text{obs}}$  (from 0-100% inhibition) is obtained by fitting to bounded exponential decay Equation VI (Fig. 18D). This is a simplified version of Equation V (Fig. 14D) in *Method III* (constraining  $v_s = 0$ ) because we only consider two-step irreversible inhibition (*Data Analysis 4A*) and one-step irreversible inhibition (*Data Analysis 4B*). Reversible (two-step) covalent inhibition with a slow rate of covalent dissociation  $k_6$  ( $t_{1/2\text{diss}} = \text{LN}(2)/k_6$ ) can be analyzed with preincubation dilution assays using the initial product formation velocity after rapid/jump dilution (Copeland, 2013e; Copeland et al., 2011) but will not be discussed here because the (slow) dissociation of covalent EI\* adduct can complicate the algebraic analysis.

Generally, preincubation assays are disfavored because their experimental execution requires more material and measurements than incubation assays with continuous read-out. However, as already mentioned in *Method III*, preincubation methods are favored



**Figure 18** Method IV: Preincubation time–dependent inhibition with dilution/competition. Simulated with **KinVol** for 100 pM enzyme and 50 nM two-step irreversible inhibitor **C** (before dilution) in  $V_T = 1$  and 10  $\mu\text{M}$  substrate **S1** in  $V_{\text{sub}} = 99$  corresponding with 100-fold dilution in excess substrate ( $[\text{S}] = 10K_M$ ). **(A)** Enzyme is preincubated with inhibitor to form noncovalent complex EI and covalent adduct EI\* in absence of competing substrate, followed by dilution in excess substrate. Initial noncovalent EI complex forms rapidly ( $([\text{I}]_i/K_i = 0.5)$ ) but fully dissociates upon dilution in a large volume ( $V_{\text{sub}} \gg V_T$ ) and/or addition of a high concentration of competing substrate ( $[\text{S}] > K_M$ ), as the  $\text{E} + \text{I} \leftrightarrow \text{EI}$  equilibrium has shifted towards fully unbound enzyme ( $([\text{I}]_i/K_i^{\text{app}} \ll 0.1)$ ). **(B)** Preincubation time–dependent enzyme activity  $v_T$  (in AU/s) is obtained from the (linear) slope of product formation velocity. Dilution in excess substrate quenches EI\* formation after substrate addition ( $\Delta[\text{EI}^*] = 0$ ), thus enabling longer incubation times compared to Method III. This measurement must be performed separately after each preincubation time. **(C)** Enzyme activity  $v_T$  decreases exponentially from 0% covalent adduct (Y-intercept = enzyme activity without preincubation  $v_i$ ) to 100% covalent adduct ( $v_s = 0$ ). Enzyme activity without preincubation  $v_i$  equals the uninhibited enzyme activity  $v^{\text{ctrl}}$  for one-step as well as two-step irreversible inhibitors: dilution in excess substrate should induce full dissociation of noncovalently bound inhibitor ( $([\text{I}]_i \ll 0.1K_i^{\text{app}})$ ), and covalent adduct does not form instantly. **(D)** Bounded exponential decay Equation VI to fit preincubation time–dependent enzyme activity  $v_T$  (in AU/s) after dilution in (excess) competing substrate against preincubation time  $t'$  (in s) for irreversible one- and two-step inhibition. This is a simplified version of Equation V (Fig. 14D) constraining  $v_s = 0$  (inactivation at reaction completion).  $v_i$  = enzyme activity without preincubation (in AU/s) = uninhibited enzyme activity  $v^{\text{ctrl}}$  because covalent adduct has not yet been formed and noncovalent complex has been disrupted by dilution in excess substrate.  $v_T$  = preincubation time–dependent enzyme activity (in AU/s) reflecting covalent EI\* adduct formed.  $t'$  = preincubation time of enzyme and inhibitor before substrate addition (in s).  $k_{\text{obs}}$  = observed rate of time–dependent inhibition from initial  $v_i$  to final  $v_s$  (in  $\text{s}^{-1}$ ).

for inhibitors that have a slow covalent reaction rate and/or a poor noncovalent affinity. Additionally, dilution in excess substrate can resolve issues for enzyme assays that do not generate enough product for a robust signal (slow  $v^{\text{ctrl}}$ ), as the maximum incubation time to calculate  $v_T$  is not limited by formation of EI\* during incubation ( $\Delta[\text{EI}^*]_i = 0$ ): incubation time can be longer than preincubation time. It is important to mention that there is still a limit to the incubation time: competition and/or dilution cannot fully mitigate the covalent adduct formation reaction but it can be reduced to a negligible rate during the incubation. Finally, this method allows the assessment of covalent adduct formation potency without contamination by reversible inhibition. This can be beneficial in the analysis of two-step covalent inhibitors that exhibit tight-binding behavior (customary for kinase inhibitors that have to compete with ATP): very potent noncovalent affinity ‘shields’ or ‘contaminates’ the rate of covalent adduct formation in the other protocols but not in this method, as detection is based solely on inhibition by covalent EI\* adduct. However, the enzyme concentration during incubation is much lower than during preincubation, and inhibitor has to be present in excess during preincubation (*pseudo-first order conditions*), thus limiting the inhibitor concentration to higher concentrations than with other methods, which might be impractical.

Be aware that dilution in (excess) substrate will change the absolute enzyme/inhibitor concentrations from preincubation to incubation, and make sure to calculate the desired

enzyme concentration during incubation accordingly. Reaction completion ( $v_t < 0.1v_t^{\text{ctrl}}$ ) should not be reached before the first (shortest) preincubation time because it will be impossible to detect time-dependent changes in enzyme activity. This can be resolved by increasing the measurement interval (shorter  $dt'$ ) or reducing the inhibitor concentration whenever possible. This method is less suitable for inhibitors with a very fast covalent adduct formation  $k_{\text{inact}}$  because preincubation is performed in absence of competing substrate (thus allowing the maximum rate of covalent adduct formation possible at this inhibitor concentration).

### Preincubation Time–Dependent Inhibition with Dilution/Competition

The protocol below provides a generic set of steps to accomplish this type of measurement. Specific reagents, and assay conditions for preincubation time–dependent inhibition with dilution of two-step irreversible covalent acetylcholinesterase inhibitors, can be found in Kitz & Wilson (1962).

#### Materials

- 1 × Assay/reaction buffer supplemented with co-factors and reducing agent
- Active enzyme, 200× solution in assay buffer
- Substrate with continuous or quenched read-out, 1 × solution in assay buffer
- Positive control: vehicle/solvent as DMSO stock, or 2% solution in assay buffer
- Negative control: known inhibitor or alkylating agent as DMSO stock, or 200× solution in assay buffer
- Inhibitor: as DMSO stock, or serial dilution of 200× solution in assay buffer with 2% DMSO
- Optional:* Development/quenching solution
- 1.5 ml (Eppendorf) microtubes to prepare stock solutions
- 384-well low volume microplate with nonbinding surface (e.g., Corning 3820 or 4513) for preincubation
- General microplate cover/lid (e.g., Corning 6569 Microplate Aluminum Sealing Tape) to seal 384-well plate during preincubation
- 96-well low volume microplate with nonbinding surface (e.g., Corning 3650 or 3820) for quenching and read-out
- Optional:* 96-well microplate to prepare serial dilution of inhibitor concentration
- Optional:* Microtubes to perform preincubations (e.g., Eppendorf Protein Lobind Microtubes, #022431018)
- Optional:* 384-well low volume microplate with nonbinding surface (e.g., Corning 3820 or 4513) for read-out
- Microplate reader equipped with appropriate filters to detect product formation (e.g., CLARIOstar microplate reader)
- Optional:* Automated (acoustic) dispenser (e.g., Labcyte ECHO 550 Liquid Handler acoustic dispenser)

*Before you start*, optimize assay conditions in the uninhibited control to ensure compliance with assumptions and restrictions, as outlined in *Basic Protocol I*. Consult Table 3 in the troubleshooting section for common optimization and troubleshooting options.

#### Specific adjustments for Method IV

Substrate should be added in a large volume ( $V_{\text{sub}} \gg V_t$ ) and/or at a high concentration ( $[S]_0 \gg K_M$ ) to quench time-dependent enzyme inhibition (Fig. 18A). Enzyme concentration after dilution  $[E^{\text{total}}]_t$  should be adjusted to correspond to maximum 10% substrate conversion until the end of the incubation in the uninhibited control ( $[P]_t < 0.1[S]_0$ ), and substrate should be present in excess ( $[S]_0 > 10[E^{\text{total}}]_t$ ). Preincubation-dependent enzyme activity should be calculated from initial, linear product formation after substrate addition. Validate that enough product is formed for a good signal/noise

ratio ( $Z' > 0.5$ ) by calculating the  $Z'$ -score from the uninhibited and inhibited controls (ideally 8 replicates) in a separate experiment (Zhang et al., 1999). This method is compatible with homogeneous (continuous) assays but also with assays that require a development/quenching step to visualize formed product. Note that preincubation in very small volumes ( $< 10 \mu\text{l}$ ) is not representative/reliable and the volume after 100-fold dilution in substrate will often exceed the maximum well volume of assay plates. Therefore, preincubation is typically performed in a larger volume (tube or plate) from which aliquots are removed at the end of the preincubation. In this protocol, we perform incubations in triplicate ( $20 \mu\text{l}$  per replicate) in a 384-well plate, from which  $2\text{-}\mu\text{l}$  aliquots are removed and quenched in  $198 \mu\text{l}$  substrate in a 96-well plate that is also used for read-out. Optionally, it is possible to then transfer  $20 \mu\text{l}$  to a 384-well plate for read-out, but multiple transfers of assays solutions will introduce errors. Alternatively, preincubation can be performed in microtubes or a 96-well plate.

1. Add inhibitor or control (e.g.,  $0.2 \mu\text{l}$ ) and assay buffer (e.g.,  $10 \mu\text{l}$ ) to each well with the uninhibited control for full enzyme activity containing the same volume vehicle/solvent instead of inhibitor, as outlined in step 1 of *Basic Protocol III*.

Gently shake to mix DMSO with the aqueous buffer. Typically, measurements are performed in triplicate (or more replicates) with at least 8 inhibitor concentrations for at least 5 preincubation times. Inhibitor concentrations might need optimization, but a rational starting point is to use inhibitor concentrations below 5 times the  $\text{IC}_{50}$  at the shortest preincubation time  $t'$ : inhibition is expected to improve in a time-dependent manner, and the best results are obtained when full inhibition is not achieved already at the shortest preincubation time (Fig. 18C). Whether preincubation is performed in a tube or microplate is a matter of personal preference, compatibility with lab equipment and automation, and convenience of dispensing small volumes.

2. Add active enzyme in assay buffer to each well (e.g.,  $10 \mu\text{l}$  of  $200\times$  solution) or tube to start preincubation of enzyme with inhibitor and homogenize the solution by gently shaking (1 min at 300 rpm). Alternatively, dispensing the enzyme at a high flow rate will also mix the components.

The order of enzyme and inhibitor addition is not important *per se*, as long as DMSO stocks are added prior to buffered (aqueous) solutions. Inhibitor must be present in excess during preincubation ( $[\text{I}]_0 > 10[\text{E}]_0$ ). Optionally, gently centrifuge the plate or microtubes (1 min at 1000 rpm) to ensure assay components are not stuck at the top of the well.

3. Seal the wells with a cover or lid, and close the caps of microtubes to prevent evaporation of assay components during preincubation.
4. Remove a single aliquot in volume  $V_t$  (e.g.,  $2 \mu\text{l}$ ) from the reaction mixture, and transfer to a 96-well microplate already containing a large volume (volume  $V_{\text{sub}}$ ) of substrate (e.g.,  $198 \mu\text{l}$  of  $1\times$  solution in assay buffer) after preincubation time  $t'$ .

Substrate should be added in a large volume ( $V_t \ll V_t'$ ) and/or at a high concentration ( $[\text{S}] \gg K_M$ ) to quench time-dependent addition enzyme inhibition during incubation by dilution ( $[\text{I}]_t \ll [\text{I}]_t'$ ) or competition (increasing  $K_i^{\text{app}}$  or decreasing  $k_{\text{chem}}^{\text{app}}$ ). Dilution to inhibitor concentration far below the equilibrium concentration ( $[\text{I}]_t \ll K_i^{\text{app}}$ ) promotes dissociation of noncovalently bound inhibitor after substrate addition (Fig. 18A). The accuracy of the measurement improves if the preincubation time is monitored precisely. Optionally, homogenize the solutions by gentle shaking (300 rpm) and centrifuge the plate (1 min at 1000 rpm) to ensure assay components are not stuck at the top of the well.

5. *Quenching*: Add development solution to the reaction mixture in the microplate to quench the product formation reaction if read-out of product formation requires a development/quenching step to visualize formed product after incubation time  $t$ .

Follow manufacturer's advice on waiting time after addition of development solution before read-out. Incubation time  $t$  is the elapsed time between onset of product formation by substrate addition (step 4) and addition of development/quenching solution (step 5). A possible advantage to the use of a quenched assay is the ability to store the samples after addition of quenching/development solution (step 5) and measure product formation (step 6) in all samples after completion of the final preincubation rather than performing multiple separate measurements (after each preincubation time).

6. *Optional*: Transfer aliquot (e.g., 20  $\mu$ l) to a 384-well microplate for read-out.

Typically, the total volume after dilution in substrate solution ( $V_t = V_{\text{sub}} + V_t'$ ) exceeds the maximum well volume of a 384-well microplate. Transfer an appropriate amount of reaction mixture (at least two technical replicates) to a microplate. This step can be skipped if read-out is performed in a 96-well plate.

7. Measure formed product after incubation by detection of the product read-out in microplate reader.

Incubation time  $t$  (after substrate addition) is arbitrary as long as product formation is linear in uninhibited as well as inhibited samples (Fig. 18B).

8. Repeat *Basic Protocol IV*, steps 4-7, for at least another four preincubation times.

Preincubation time  $t'$  is the elapsed time between onset of inhibition by mixing enzyme and inhibitor (step 2) and addition of substrate (step 4). A typical preincubation assay is multiple hours measuring enzyme activity every 5-30 min, depending on enzyme stability and inhibitor reaction rates. Best results are obtained if the incubation time  $t$  used to calculate enzyme activity is kept constant at all preincubation times.

9. Proceed to *Basic Data Analysis Protocol 4* to convert the raw experimental data into preincubation time-dependent enzyme activity.

### Preincubation Time-Dependent Inhibition With Dilution

Processing of raw experimental data obtained with *Basic Protocol IV* for irreversible inhibitors.

1. Plot signal  $F$  against incubation time  $t$ .

Plot signal  $F$  (in AU) on the Y-axis against the incubation time (in s) on the X-axis for each inhibitor concentration and for the controls (Fig. 19B, Fig. 20B). *Do this separately for each preincubation time.*

2. Fit  $F_t$  against  $t$  to obtain  $v_t'$ .

Fit signal  $F_t$  against incubation time  $t$  to Equation XIII (Fig. 19B, Fig. 20B) to obtain preincubation time-dependent product formation velocity  $v_t'$  (in AU/s) from the linear slope (Fig. 18B). Linear product formation is indicative of effective disruption of additional covalent modification during incubation by dilution in excess substrate (Fig. 18A). If product formation is not linear: consult Table 3 for troubleshooting or proceed to *Basic Data Analysis Protocol 3*.

$$F_t = F_0 + v_t't$$

Equation XIII for nonlinear regression of straight line  $Y = Y_{\text{Intercept}} + \text{Slope} \cdot X$  with  $Y = \text{signal } F_t$  (in AU) and  $X = \text{incubation time } t$  (in s) to find

YIntercept = background signal at reaction initiation  $F_0$  (in AU) and Slope = preincubation time–dependent product formation velocity  $v_t$  (in AU/s).

- Proceed to Data Analysis Protocols to obtain the appropriate kinetic parameters for each covalent binding mode: *Data Analysis Protocol 4Ai* or *4Aii* for two-step irreversible inhibitors and *Data Analysis Protocol 4Bi* or *4Bii* for one-step irreversible inhibitors.

Selection of a Data Analysis Method for inhibitors with an irreversible binding mode depends on the desired visual representation as well as personal preference. Generally, *Basic Data Analysis Protocols 4Ai* and *4Bi* have less data processing/manipulation and are more informative for comparison of various inhibitors on a single enzyme target, as they are compatible with assessment of inhibitor potency simultaneous with visual assessment of time-dependent enzyme stability  $k_{\text{ctrl}}$  (Fig. 19F and 19G and Figs. 20F and 20G). *Alternative Data Analysis Protocols 4Aii* and *4Bii* involve normalization of the enzyme activity that aids visual assessment of inhibitory potency of a single inhibitor on multiple enzyme targets (that might have a variable stability) (Fig. 19H and 19I and Fig. 20H and 20I).

EXP Conditions	Data Analysis Protocol		
	2-step IRREV	1-step IRREV	2-step REV
$k_{\text{ctrl}} = 0$	4Ai/4Aii	4Bi/4Bii	–
$k_{\text{degE}} > 0$	4Ai/4Aii	4Bi/4Bii	–

Exemplary assay concentrations during preincubation and during incubation.

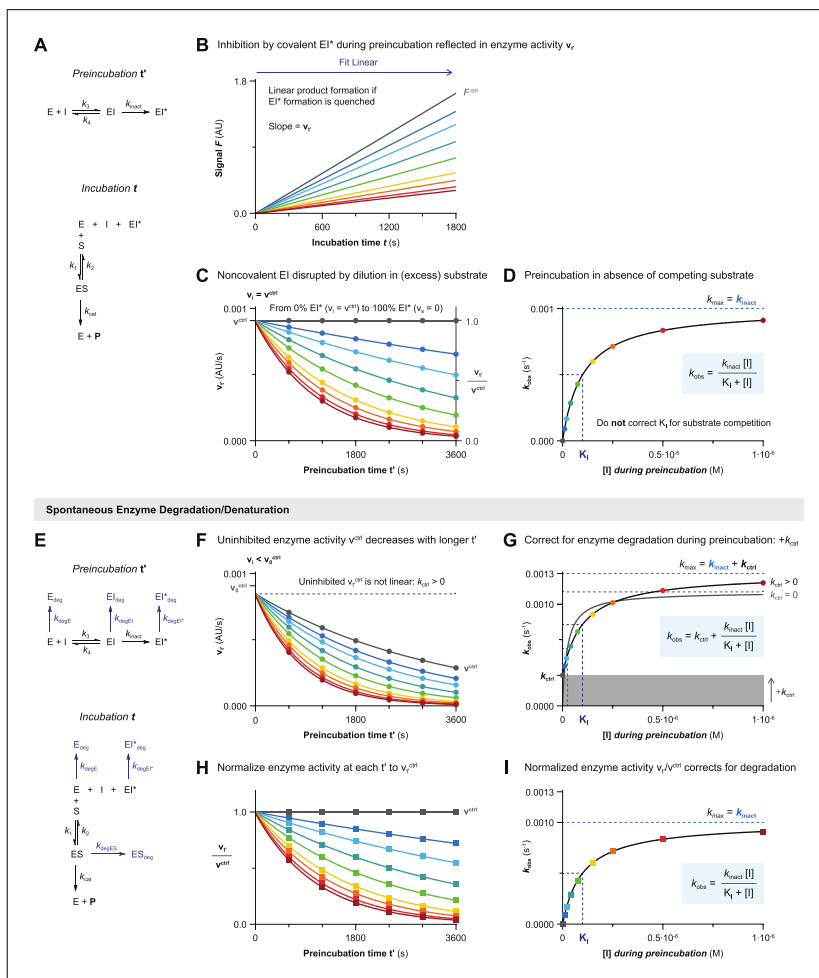
	Concentration during preincubation $t'$			Concentrations during incubation $t$		
	[stock]	V ( $\mu\text{l}$ )	[conc] $_t'$	[stock]	V ( $\mu\text{l}$ )	[conc] $_t$
Enzyme	200 nM	10	99 nM	–	1	1.0 nM
Inhibitor	2000 nM	10.2	<b>1010 nM</b>	–	1	10 nM
Substrate	–	–	–	10 $\mu\text{M}$	198	<b>9.9 <math>\mu\text{M}</math></b>
<i>Total</i>		20.2			200	

#### Data Analysis 4A: Preincubation Time–Dependent Inhibition With Dilution/Competition for Two-Step Irreversible Covalent Inhibition

Kinetic analysis of enzyme activity with dilution/competition after preincubation in the presence of a two-step covalent inhibitor is similar to data analysis of preincubation without dilution/competition (*Data Analysis 3A*), with the exception that longer incubation times are possible to calculate enzyme activity  $v_t$  from the slope (Fig. 19A and 19B), and enzyme activity without preincubation  $v_i$  should be equal to the uninhibited enzyme activity  $v^{\text{ctrl}}$  (Fig. 19C). Contrary to *Method III*, this does not imply that the inhibitors show one-step behavior: it merely confirms that extensive dilution/substrate competition successfully induced inhibitor dissociation from noncovalent EI complex to unbound enzyme. It is essential to plot the rate of covalent adduct formation  $k_{\text{obs}}$  against the inhibitor concentration *during preincubation* (Fig. 19D) to obtain kinetic parameters:  $k_{\text{obs}}$  is based on the formation of EI\* during preincubation, and the inhibitor concentration during preincubation is much higher than the inhibitor concentration after dilution in substrate ( $[I]_{t'} \gg [I]_t$ ).

#### Warnings and remarks

Insufficient dilution/competition will partially disrupt noncovalent EI complex, resulting in a time-dependent decrease of enzyme activity due to formation of EI\* after substrate addition (Fig. 19B) and deviation from  $v_i = v^{\text{ctrl}}$ , as noncovalent complex EI contributes to inhibition without preincubation (Fig. 19C). Increasing substrate concentration and/or



**Figure 19** Data Analysis 4A: Preincubation time–dependent inhibition with dilution/competition for two-step irreversible covalent inhibition. Simulated with **KinVol** (A–D) or **KinVolDeg** (E–I) for inhibitor **C** with 100 pM enzyme in  $V_t = 1$  ( $[E^{\text{total}}]_t = 100$ ,  $[E^{\text{total}}]_t = 1$ ) and 10  $\mu\text{M}$  substrate **S1** ( $[S] = 10K_M$ ) in  $V_{\text{sub}} = 99$ . **(A)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after dilution in excess substrate for two-step irreversible covalent inhibition. **(B)** Time-dependent product formation after preincubation ( $t' = 1800$  s) in absence of inhibitor  $F^{\text{ctrl}}$  or in presence of various inhibitor concentrations. Enzyme activity after preincubation  $v_t$  is obtained from the linear slope. **(C)** Preincubation time–dependent enzyme activity  $v_t$  is fitted to Equation VI (Fig. 18D) for each inhibitor concentration with global shared value for  $v_t$  ( $v_t = v^{\text{ctrl}}$ ) to obtain observed rates of inactivation  $k_{\text{obs}}$ . Alternatively,  $v_t$  can be normalized to a fraction of the uninhibited enzyme activity  $v_t^{\text{ctrl}}$ . **(D)** Half-maximum  $k_{\text{obs}} = 1/2k_{\text{inact}}$  is reached when inhibitor concentration during preincubation equals the inactivation constant  $K_i$ ; no correction for substrate competition because  $v_t$  reflects the remaining unbound/noncovalent enzyme activity after preincubation in absence of competing substrate. **(E)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after dilution in excess substrate for two-step irreversible covalent inhibition with spontaneous enzyme degradation/denaturation. Simulated with  $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}} = 0.0003 \text{ s}^{-1}$ . **(F)** Uninhibited enzyme activity after preincubation  $v_t^{\text{ctrl}}$  decreases with longer preincubation. Enzyme activity  $v_t$  is fitted to Equation VI (Fig. 18D) for each inhibitor concentration during preincubation with globally shared value for  $v_t$  ( $v_t = v^{\text{ctrl}}$ ) to obtain observed rates of inactivation  $k_{\text{obs}}$ , as well as fitting uninhibited activity  $v_t^{\text{ctrl}}$  to obtain the rate of

(legend continues on next page)



nonlinearity  $k_{\text{ctrl}}$ . **(G)** Inhibitor concentration-dependent  $k_{\text{obs}}$  with spontaneous enzyme degradation increases with  $k_{\text{ctrl}}$  but the span from  $k_{\text{min}}$  ( $= k_{\text{ctrl}}$ ) to  $k_{\text{max}}$  ( $= k_{\text{inact}} + k_{\text{ctrl}}$ ) still equals  $k_{\text{inact}}$ . Fit with algebraic correction for nonlinearity (black line,  $k_{\text{ctrl}} > 0$ ). Ignoring the nonlinearity (gray line, constrain  $k_{\text{ctrl}} = 0$ ) results in underestimation of  $K_i$  (overestimation of potency) and overestimation of  $k_{\text{inact}}$ . **(H)** Normalized enzyme activity  $v_t/v_0^{\text{ctrl}}$  is fitted to Equation VI (Fig. 18D) for each inhibitor concentration *during preincubation* (constrain  $v_t/v_0^{\text{ctrl}} = 1$ ) to obtain corrected observed rates of inactivation  $k_{\text{obs}}$ . **(I)** Inhibitor concentration-dependent  $k_{\text{obs}}$  has been corrected for enzyme degradation by fitting normalized enzyme activity  $v_t/v_0^{\text{ctrl}}$  and does not require further corrections.

dilution in a larger volume might resolve this. Alternatively, enzyme activity with partial disruption of noncovalent EI analyzed with *Data Analysis 3A* still results in reliable estimates of  $k_{\text{obs}}$ . Please note that, although detection based only on covalent adduct formation allows analysis of two-step inhibitors displaying tight-binding behavior (very high noncovalent affinity resulting in full inhibition at all inhibitor concentrations), these inhibitor concentrations are saturating if they comply with the rapid equilibrium approximation ( $K_i \approx K_I$ ); thus, it would only be possible to determine the lower limit of  $k_{\text{inact}}$  and the upper limit of  $K_I$  (Fig. 2G).

Correction for enzyme (in)stability during preincubation by correcting for the rate of spontaneous degradation  $k_{\text{ctrl}}$  has been reported (Obach, Walsky, & Venkatakrishnan, 2007) for dilution experiments with irreversible covalent inhibitors (Fig. 19E-G). Alternatively, enzyme activity after preincubation  $v_t$  can be normalized to the uninhibited enzyme activity after preincubation  $v_t^{\text{ctrl}}$  (Fig. 19H and 19I).

### Two-Step Irreversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol IV* that has been processed according to *Basic Data Analysis Protocol 4* for two-step irreversible inhibitors.

1. Plot  $v_t$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 19C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations. Check whether the uninhibited enzyme activity is independent of preincubation time ( $v_0^{\text{ctrl}} = v_t^{\text{ctrl}}$ , Fig. 19C): an algebraic correction for enzyme instability ( $k_{\text{ctrl}} > 0$ , Fig. 19F) can be performed in step 4 of this protocol by accounting for nonlinearity in the uninhibited control in the secondary  $k_{\text{obs}}$  plot (Fig. 19G). Alternatively, proceed to *Alternative Data Analysis Protocol 4Bii* to correct for enzyme instability ( $v_0^{\text{ctrl}} > v_t^{\text{ctrl}}$ ) by normalization of the enzyme activity  $v_t/v_t^{\text{ctrl}}$  (Fig. 19H and 19I).

2. Fit  $v_t$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Fit the mean and standard deviation of  $v_t$  against preincubation time  $t'$  (Fig. 19C/F) for each inhibitor concentration to bounded exponential decay Equation VI (Fig. 18D) with shared value for initial velocity  $v_i$  to obtain the observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i$  (Y-intercept) to full inactivation ( $v_s$  in fully inhibited control). A lack of initial noncovalent complex ( $v_i = v_0^{\text{ctrl}}$ ) is indicative of effective disruption of noncovalent interactions by dilution in excess substrate. Validate this by fitting without constraints for  $v_i$ . Proceed to *Basic Data Analysis Protocol 3Ai* if deviations ( $v_i < v_0^{\text{ctrl}}$ ) are observed.

$$v_{t'} = v_0^{\text{ctrl}} e^{-k_{\text{obs}}t'}$$

#### Equation VI

Equation VI for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y =$  preincubation time-dependent product formation velocity  $v_t$  (in AU/s),  $X =$  preincubation time  $t'$



(in s) and Plateau = final velocity  $v_s = 0$  or  $v_s$  in fully inhibited control (in AU/s) to find  $Y_0 = Y$ -intercept = initial velocity  $v_i =$  uninhibited velocity  $v_0^{\text{ctrl}}$  (in AU/s, shared value) and  $k =$  observed reaction rate  $k_{\text{obs}}$  (in  $s^{-1}$ ).

3. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $s^{-1}$ ) on the Y-axis against inhibitor concentration (in M) *during preincubation (before addition of substrate)* on the X-axis (Fig. 19D/G). The plot of  $k_{\text{obs}}$  against  $[I]$  should reach a maximum  $k_{\text{obs}}$  at saturating inhibitor concentration. Note that a linear curve is indicative of one-step binding behavior at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$  in Fig. 3F) with  $v_i = v_0^{\text{ctrl}}$  (shared Y-intercept in the previous step). Proceed to *Basic Data Analysis Protocol 4Bi* step 4 after it has been validated that the linear curve is not resultant from saturating inhibitor concentrations ( $[I] \gg 10K_I$  in Fig. 3G) as identified by  $v_i \ll v_0^{\text{ctrl}}$ , by repeating the measurement with lower inhibitor concentrations.

4. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{inact}}$  and  $K_I$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* to Equation XV to obtain maximum inactivation rate constant  $k_{\text{inact}}$  (in  $s^{-1}$ ) and inactivation constant  $K_I$  (in M). Constrain  $k_{\text{ctrl}} = k_{\text{obs}}$  of the uninhibited control (Fig. 19G). Inactivation constant  $K_I$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate irreversible covalent inhibitor potency  $k_{\text{inact}}/K_I$  (in  $M^{-1}s^{-1}$ ) with propagation of error with *Sample Calculation 2*.

$$k_{\text{obs}} = k_{\text{ctrl}} + \frac{k_{\text{inact}} [I]}{K_I + [I]}$$

**Equation XV**

Equation XV for nonlinear regression of user-defined explicit equation  $Y = Y_0 + ((k_{\text{max}} * X) / (K_I + X))$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $s^{-1}$ ) and  $X =$  inhibitor concentration during preincubation (in M) to find  $Y_0 =$  rate of nonlinearity in uninhibited control  $k_{\text{ctrl}}$  (in  $s^{-1}$ ),  $k_{\text{max}} =$  maximum reaction rate  $k_{\text{inact}}$  (in  $s^{-1}$ ), and  $K_I =$  Inactivation constant  $K_I$  (in M).

5. *Optional*: Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental read-out to the product formation simulated with scripts **KinVol** and **KinVolDeg** (using experimental rate constant  $k_{\text{inact}} = k_5$ ) to confirm that the calculated kinetic constants are in accordance with the experimental data.

## Two-Step Irreversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol IV* that has been processed according to *Basic Data Analysis Protocol 4* for two-step irreversible inhibitors.

1. Plot  $v_t$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 19C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations.

2. Normalize  $v_t$  to obtain  $v_t/v_t^{\text{ctrl}}$ .

Normalize  $v_t$  (in AU/s) of each inhibitor concentration and the controls to lowest value = 0 (or full inhibition control) and highest value = uninhibited product formation  $v_t^{\text{ctrl}}$  (in AU/s) to obtain normalized enzyme activity  $v_t/v_t^{\text{ctrl}}$  (Fig. 19H). Perform this correction *separately* for each preincubation time.

**ALTERNATIVE  
DATA  
ANALYSIS  
PROTOCOL 4Aii**

Mons et al.

**67 of 85**

3. Plot and fit  $v_{t'}/v^{\text{ctrl}}$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Plot the mean and standard deviation of  $v_{t'}/v^{\text{ctrl}}$  on the Y-axis against preincubation time  $t'$  (in s) on the X-axis (Fig. 19H). Fit to exponential decay Equation XVI to obtain  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i/v_0^{\text{ctrl}}$  to full inactivation (Plateau = 0). A lack of initial noncovalent complex ( $v_i = v_0^{\text{ctrl}}$ ) is indicative of effective disruption of noncovalent interactions by dilution in excess substrate. Validate this by fitting without constraints for  $v_i$ . Proceed to *Basic Data Analysis Protocol 3Aii* if deviations ( $v_i < v_0^{\text{ctrl}}$ ) are observed.

$$\left( \frac{v_{t'}}{v_{t'}^{\text{ctrl}}} \right) = e^{-k_{\text{obs}}t'}$$

**Equation XVI**

Equation XVI for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y =$  normalized preincubation time-dependent product formation velocity  $v_{t'}/v^{\text{ctrl}}$  (unitless),  $X =$  preincubation time  $t'$  (in s),  $Y_0 =$  Y-intercept = normalized initial velocity  $v_i/v_0^{\text{ctrl}} = 1$  (unitless), and Plateau = normalized final velocity  $v_s/v_s^{\text{ctrl}} = 0$  (unitless) to find  $k =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

4. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) *during preincubation* (before addition of substrate) on the X-axis (Fig. 19I). The plot of  $k_{\text{obs}}$  against  $[I]$  should reach a maximum  $k_{\text{obs}}$  at saturating inhibitor concentration. Note that a linear curve is indicative of one-step binding behavior at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$  in Fig. 3F) with  $v_i = v_0^{\text{ctrl}}$  (shared Y-intercept = 1 in the previous step). Proceed to *Basic Data Analysis Protocol 4Bii* step 5 after it has been validated that the linear curve is not resultant from saturating inhibitor concentrations ( $[I] \gg 10K_I$  in Fig. 3G) as identified by  $v_i \ll v_0^{\text{ctrl}}$  (shared Y-intercept = 0 in the previous step), by repeating the measurement with lower inhibitor concentrations.

5. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{inact}}$  and  $K_I$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* to Equation XVII to obtain maximum inactivation rate constant  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and inactivation constant  $K_I$  (in M) (Fig. 19I). Do not correct for enzyme instability ( $k_{\text{ctrl}} > 0$ ), as this correction has already been performed by normalizing  $v_{t'}$ . Inactivation constant  $K_I$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate irreversible covalent inhibitor potency  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) with propagation of error with *Sample Calculation 2*

$$k_{\text{obs}} = \frac{k_{\text{inact}} [I]}{K_I + [I]}$$

**Equation XVII**

Equation XVII for nonlinear regression of user-defined explicit equation  $Y = Y_0 + ((k_{\text{max}} * X) / (K_I + X))$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ),  $X =$  inhibitor concentration during preincubation (in M), and  $Y_0 = 0$  (in  $\text{s}^{-1}$ ) to find  $k_{\text{max}} =$  maximum reaction rate  $k_{\text{inact}}$  (in  $\text{s}^{-1}$ ) and  $K_I =$  Inactivation constant  $K_I$  (in M).

6. *Optional:* Validate kinetic parameters with kinetic simulations by proceeding to *Basic Data Analysis Protocol 4Ai* step 5.

## Data Analysis 4B: Preincubation Time-Dependent Inhibition With Dilution/Competition for One-Step Irreversible Covalent Inhibition

Kinetic analysis of enzyme activity with dilution/competition after preincubation in presence of a one-step covalent inhibitor is almost identical to data analysis of preincubation without dilution in excess substrate (*Data Analysis 3B*), with the exception that longer incubation times are possible to calculate enzyme activity  $v_t$  from the slope (Fig. 20A-C). It is essential to plot the rate of covalent adduct formation  $k_{\text{obs}}$  against the inhibitor concentration *during preincubation* (Fig. 20D) to obtain kinetic parameters:  $k_{\text{obs}}$  is based on the formation of EI\* during preincubation, and the inhibitor concentration during preincubation will be much higher than the inhibitor concentration after dilution in substrate ( $[I]_t \gg [I]_i$ ).

### Warnings and remarks

Dilution/competition does not disrupt any noncovalent EI complex, as this is non-existent for one-step inhibitors, but the rate of covalent adduct formation  $k_{\text{obs}}$  should be negligible after dilution in excess substrate, to prevent formation of covalent EI\*. Insufficient dilution and/or competition ( $\Delta[\text{EI}^*]_t > 0$ ) can result in time-dependent decrease of enzyme activity due to formation of EI\* after substrate addition (Fig. 20B). Increasing substrate concentration and/or dilution in a larger volume might resolve this if necessary, but simply performing analysis with *Data Analysis Protocol 3B* also results in reliable estimates of  $k_{\text{obs}}$ . Inhibitor concentrations that reach reaction completion during the shortest preincubation time should be excluded from the fit (highest concentration in Fig. 20C) as these fits are not reliable.

Correction for enzyme (in)stability during preincubation by correcting for the rate of spontaneous degradation  $k_{\text{ctrl}}$  has been reported (Obach et al., 2007) for dilution experiments with irreversible covalent inhibitors (Fig. 20E-G). Alternatively, enzyme activity after preincubation  $v_t$  can be normalized to the uninhibited enzyme activity after preincubation  $v_t^{\text{ctrl}}$  (Fig. 20H and 20I).

### One-Step Irreversible Covalent Inhibition

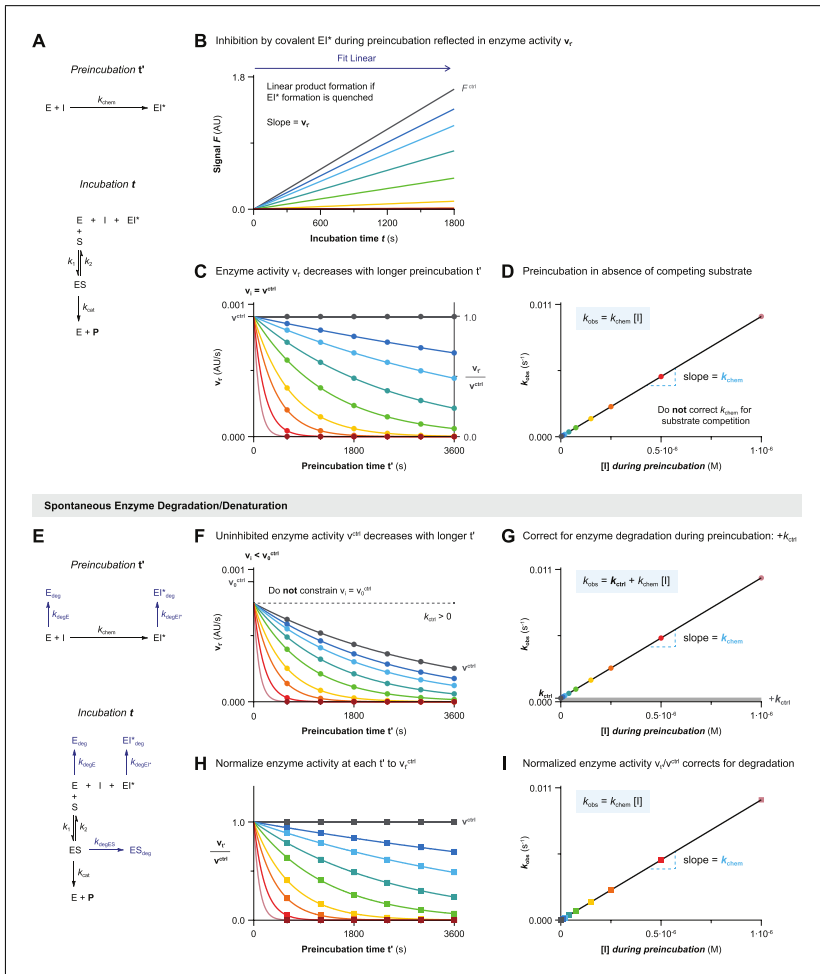
Processing of experimental data obtained with *Basic Protocol IV* that has been processed according to *Basic Data Analysis Protocol 4* for one-step irreversible covalent inhibitors and two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ).

1. Plot  $v_t$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 20C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations. Check whether the uninhibited enzyme activity is independent of preincubation time ( $v_0^{\text{ctrl}} = v_t^{\text{ctrl}}$ , Fig. 20C): an algebraic correction for enzyme instability ( $k_{\text{ctrl}} > 0$ , Fig. 20F) can be performed in step 4 of this protocol by accounting for nonlinearity in the uninhibited control in the secondary  $k_{\text{obs}}$  plot (Fig. 20G). Alternatively, proceed to *Alternative Data Analysis Protocol 4Bii* to correct for enzyme instability ( $v_0^{\text{ctrl}} > v_t^{\text{ctrl}}$ ) by normalization of the enzyme activity  $v_t/v_t^{\text{ctrl}}$  (Fig. 20H and 20I).

2. Fit  $v_t$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Fit the mean and standard deviation of  $v_t$  against preincubation time  $t'$  (Fig. 20C/F) for each inhibitor concentration to bounded exponential decay Equation VI (Fig. 18D). Constrain initial velocity  $v_i$  to a shared value to obtain observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i$  (Y-intercept) to full inactivation ( $v_s = 0$  or value in fully inhibited control).



**Figure 20** Data Analysis 4B: Preincubation time–dependent inhibition with dilution/competition for one-step irreversible covalent inhibition. Simulated with **KinVol** (A–D) or **KinVolDeg** (E–I) for inhibitor **D** with 100 pM enzyme in  $V_T = 1$  ( $[E^{\text{total}}]_t = 100$ ,  $[E^{\text{total}}]_t = 1$ ) and 10  $\mu\text{M}$  substrate **S1** ( $[S] = 10K_M$ ) in  $V_{\text{SUB}} = 99$ . **(A)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after dilution in excess substrate for one-step irreversible covalent inhibition. **(B)** Time-dependent product formation after preincubation ( $t' = 1800$  s) in absence of inhibitor  $E^{\text{ctrl}}$  or in presence of various inhibitor concentrations. Enzyme activity after preincubation  $v_i$  is obtained from the linear slope. **(C)** Preincubation time–dependent enzyme activity  $v_i$  is fitted to Equation VI (Fig. 18D) for each inhibitor concentration during preincubation with globally shared value for  $v_i$  ( $v_i = v_i^{\text{ctrl}}$ ) to obtain observed rates of inactivation  $k_{\text{obs}}$ . Alternatively,  $v_i$  can be normalized to a fraction of the uninhibited enzyme activity  $v_i^{\text{ctrl}}$ . The highest inhibitor concentration should be excluded:  $v_{600} = 0$ . **(D)** Inhibitor concentration-dependent  $k_{\text{obs}}$  increases linearly with inhibitor concentration during preincubation, with  $k_{\text{chem}}$  as the slope. No correction for substrate competition because  $v_i$  reflects the remaining unbound enzyme activity after preincubation in the absence of competing substrate. **(E)** Schematic enzyme dynamics during preincubation in absence of substrate and during incubation after dilution in excess substrate for one-step irreversible covalent inhibition with spontaneous enzyme degradation/denaturation. Simulated with  $k_{\text{degE}} = k_{\text{degES}} = k_{\text{degEI}} = 0.0003 \text{ s}^{-1}$ . **(F)** Uninhibited enzyme activity after preincubation  $v_i^{\text{ctrl}}$  decreases with longer preincubation. Enzyme activity  $v_i$  is fitted to Equation VI (Fig. 18D) for each inhibitor concentration during preincubation with globally shared value for  $v_i$  ( $v_i = v_i^{\text{ctrl}}$ ) to obtain observed rates of inactivation  $k_{\text{obs}}$ , along with fitting uninhibited (legend continues on next page)

activity  $v_t^{\text{ctrl}}$  to obtain the rate of nonlinearity  $k_{\text{ctrl}}$ . **(G)** Inhibitor concentration-dependent  $k_{\text{obs}}$  with spontaneous enzyme degradation/denaturation increases by  $k_{\text{ctrl}}$ . Fit with algebraic correction for nonlinearity (black line,  $k_{\text{ctrl}} > 0$ ) or ignoring nonlinearity (gray line, constrain  $k_{\text{ctrl}} = 0$ ). Ignoring the nonlinearity (assuming Y-intercept = 0) results in overestimation of  $k_{\text{chem}}$  (steeper slope). **(H)** Normalized enzyme activity  $v_t/v_0^{\text{ctrl}}$  is fitted to Equation VI (Fig. 18D) for each inhibitor concentration *during preincubation* (constrain  $v_t/v_0^{\text{ctrl}} = 1$ ) to obtain corrected observed rates of inactivation  $k_{\text{obs}}$ . **(I)** Inhibitor concentration-dependent  $k_{\text{obs}}$  has been corrected for enzyme degradation by fitting normalized enzyme activity  $v_t/v_0^{\text{ctrl}}$  and does not require further corrections.

$$v_t = v_0^{\text{ctrl}} e^{-k_{\text{obs}} t'}$$

#### Equation VI

Equation VI for nonlinear regression of exponential one-phase decay equation  $Y = (Y0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y =$  preincubation time-dependent product formation velocity  $v_t$  (in AU/s),  $X =$  preincubation time  $t'$  (in s), and  $\text{Plateau} =$  final velocity  $v_s = 0$  or  $v_s$  in fully inhibited control (in AU/s) to find  $Y0 =$  Y-intercept = initial velocity  $v_i =$  uninhibited velocity  $v_0^{\text{ctrl}}$  (in AU/s, shared value) and  $k =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

3. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) *during preincubation (before addition of substrate)* on the X-axis (Fig. 20D/G). The plot of  $k_{\text{obs}}$  against inhibitor concentration  $[I]$  is linear for one-step irreversible inhibitors and for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$ ).

4. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{chem}}$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* (in M) to Equation XVIII to obtain inhibitor potency  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) from the linear slope. Constrain Y-intercept  $k_{\text{ctrl}} = k_{\text{obs}}$  of the uninhibited control (Fig. 20G). Inhibitor potency  $k_{\text{chem}}$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ) with propagation of error with *Sample Calculation 9*.

$$k_{\text{obs}} = k_{\text{ctrl}} + k_{\text{chem}} [I]$$

#### Equation XVIII

Equation XVIII for nonlinear regression of straight line  $Y = \text{YIntercept} + \text{Slope} * X$  with  $Y =$  observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) and  $X =$  inhibitor concentration during preincubation (in M) to find  $\text{YIntercept} =$  rate of nonlinearity in uninhibited control  $k_{\text{ctrl}}$  (in  $\text{s}^{-1}$ ) and  $\text{Slope} =$  inactivation rate constant  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ).

5. *Optional:* Validate experimental kinetic parameters with kinetic simulations.

Proceed to *Kinetic Simulations 1* to compare the experimental read-out to the product formation simulated with scripts **KinVol** and **KinVolDeg** (using experimental rate constant  $k_{\text{chem}} = k_3$ ), to confirm that the calculated kinetic constants are in accordance with the experimental data.

### One-Step Irreversible Covalent Inhibition

Processing of experimental data obtained with *Basic Protocol IV* that has been processed according to *Basic Data Analysis Protocol 4* for one-step irreversible covalent inhibitors and two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ).

**BASIC DATA  
ANALYSIS  
PROTOCOL 4Bii**

Mons et al.

**71 of 85**

1. Plot  $v_t$  against preincubation time  $t'$  for each inhibitor concentration.

Plot the mean and standard deviation of  $v_t$  (in AU/s) on the Y-axis against preincubation time  $t'$  (in s) on the X-axis for each inhibitor concentration and the uninhibited control (Fig. 20C). Validate that inhibitor concentrations are not too high: inhibition should be less than 100% at the shortest  $t'$  for at least six inhibitor concentrations.

2. Normalize  $v_t$  to obtain  $v_t/v^{\text{ctrl}}$ .

Normalize  $v_t$  (in AU/s) of each inhibitor concentration and the controls to lowest value = 0 (or full inhibition control) and highest value = uninhibited product formation  $v_t^{\text{ctrl}}$  (in AU/s) to obtain normalized enzyme activity  $v_t/v^{\text{ctrl}}$  (Fig. 20H). Perform this correction separately for each preincubation time.

3. Plot and fit  $v_t/v^{\text{ctrl}}$  against preincubation time  $t'$  to obtain  $k_{\text{obs}}$ .

Plot the mean and standard deviation of  $v_t/v^{\text{ctrl}}$  on the Y-axis against preincubation time  $t'$  (in s) on the X-axis (Fig. 20H). Fit to exponential decay Equation XVI to obtain  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) from initial velocity  $v_i/v_0^{\text{ctrl}}$  (shared value) to full inactivation (Plateau = 0).

$$\left( \frac{v_t}{v_t^{\text{ctrl}}} \right) = e^{-k_{\text{obs}}t'}$$

#### Equation XVI

Equation XVI for nonlinear regression of exponential one-phase decay equation  $Y = (Y_0 - \text{Plateau}) * \text{EXP}(-k * X) + \text{Plateau}$  with  $Y$  = normalized preincubation time-dependent product formation velocity  $v_t/v^{\text{ctrl}}$  (unitless),  $X$  = preincubation time  $t'$  (in s),  $\text{Plateau}$  = normalized final velocity  $v_s/v_s^{\text{ctrl}} = 0$  (unitless), and  $Y_0$  = Y-intercept = normalized initial velocity  $v_i/v_0^{\text{ctrl}} = 1$  (unitless) to find  $k$  = observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ).

4. Plot  $k_{\text{obs}}$  against  $[I]$ .

Plot the mean and standard deviation of  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ) on the Y-axis against inhibitor concentration (in M) *during preincubation (before addition of substrate)* on the X-axis (Fig. 20I). The plot of  $k_{\text{obs}}$  against inhibitor concentration  $[I]$  is linear for one-step irreversible inhibitors and for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \ll 0.1K_I$ ).

5. Fit  $k_{\text{obs}}$  against  $[I]$  to obtain  $k_{\text{chem}}$ .

Fit  $k_{\text{obs}}$  against inhibitor concentration *during preincubation* to Equation XIX to obtain inhibitor potency  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) from the linear slope (Fig. 20I). Do not correct for enzyme instability ( $k_{\text{ctrl}} > 0$ ), as this correction has already been performed by normalizing  $v_t$  to  $v_t/v^{\text{ctrl}}$  in step 2 of this protocol. Inhibitor potency  $k_{\text{chem}}$  does not have to be corrected for substrate competition because preincubation is conducted in absence of competing substrate. Calculate  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) for two-step irreversible inhibitors at non-saturating inhibitor concentrations ( $[I] \leq 0.1K_I$ ) with propagation of error with *Sample Calculation 9*. Alternatively, inhibitor potency  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) or  $k_{\text{inact}}/K_I$  (in  $\text{M}^{-1}\text{s}^{-1}$ ) can be directly calculated from a single  $k_{\text{obs}}$  ( $\text{s}^{-1}$ ) and  $[I]$  (in M) with *Sample Calculation 10*.

$$k_{\text{obs}} = k_{\text{chem}} [I]$$

#### Equation XIX

Equation XIX for nonlinear regression of straight line  $Y = Y\text{Intercept} + \text{Slope} * X$  with  $Y$  = observed reaction rate  $k_{\text{obs}}$  (in  $\text{s}^{-1}$ ),  $X$  = inhibitor concentration during preincubation (in M), and  $Y\text{Intercept} = 0$  (in  $\text{s}^{-1}$ ) to find  $\text{Slope}$  = inactivation rate constant  $k_{\text{chem}}$  (in  $\text{M}^{-1}\text{s}^{-1}$ ).

6. *Optional*: Validate experimental kinetic parameters with kinetic simulations by proceeding to *Basic Data Analysis Protocol 4Bi* step 5.

## SAMPLE CALCULATIONS

The fits as obtained in the basic protocols described above still have to be converted into inhibition parameters. These are fairly straightforward linear calculations and can be performed with more basic software like Microsoft Excel. For each equation, the full right side of the equal sign is known, so it becomes a linear calculation to obtain the parameter on the left side of it.

All calculations used are listed here in order of appearance in the manuscript. We have outlined the key assumptions and a little background on the used variables for improved readability and direct applicability after following the basic protocols.

### Materials

Experimental/fitted values found in *Data Analysis Protocols 1-4*

Software to perform linear calculations (e.g., EXCEL)

#### Sample Calculation 1. Calculate $K_I$ from $K_I^{app}$

Apparent inactivation constant  $K_I^{app}$  (in M) found in *Data Analysis Protocols* (1A or 1D) for competitive two-step irreversible inhibitors is corrected for substrate competition to obtain inactivation constant  $K_I$  (in M), with propagation of error. Use substrate concentration  $[S]$  (in M) after reaction initiation and  $K_M$  (in M) as determined for these specific assay conditions (buffer, temperature, enzyme, substrate). Proceed to *Sample Calculation 2* to calculate  $k_{inact}/K_I$ .

$$K_I = \frac{K_I^{app}}{\left(1 + \frac{[S]}{K_M}\right)} \text{ with}$$

$$\sigma_{K_I} = \sqrt{\left(\frac{1}{1 + \frac{[S]}{K_M}}\right)^2 \sigma_{K_I^{app}}^2 + \left(-\frac{K_I^{app} K_M}{(K_M + [S])^2}\right)^2 \sigma_{[S]}^2 + \left(\frac{K_I^{app} [S]}{([S] + K_M)^2}\right)^2 \sigma_{K_M}^2}$$

#### Sample Calculation 2. Calculate $k_{inact}/K_I$ from $k_{inact}$ and $K_I$

Irreversible covalent inhibitor potency  $k_{inact}/K_I$  (in  $M^{-1}s^{-1}$ ) is calculated from  $k_{inact}$  (in  $s^{-1}$ ) and  $K_I$  (in M) values found in *Data Analysis Protocols* (1A, 1D, 2, 3Ai, 3Aii, 4Ai or 4Aii) and *Sample Calculation 1* for two-step irreversible inhibitors, with propagation of error.

$$\left(\frac{k_{inact}}{K_I}\right) = \frac{k_{inact}}{K_I} \text{ with } \sigma_{\frac{k_{inact}}{K_I}} = \left(\frac{k_{inact}}{K_I}\right) \sqrt{\left(\frac{\sigma_{k_{inact}}}{k_{inact}}\right)^2 + \left(\frac{\sigma_{K_I}}{K_I}\right)^2}$$

#### Sample Calculation 3. Calculate $K_i$ from $K_I^{app}$

Apparent inhibition constant  $K_i^{app}$  (in M) found in *Data Analysis Protocols* (1A, 1C, 3Ai, 3Aii or 3C) for competitive two-step (ir)reversible inhibitors is corrected for substrate competition (Cheng & Prusoff, 1973) to obtain inhibition constant  $K_i$  (in M) for the initial noncovalent equilibrium. Use substrate concentration  $[S]$  (in M) after reaction initiation and  $K_M$  (in M) as determined for these specific assay conditions (buffer, temperature, enzyme, substrate). Inhibition constant  $K_i$  approximates inactivation constant  $K_I$  for two-step irreversible inhibitors if covalent bond formation is rate-limiting (*rapid equilibrium assumption*).

$$K_i = \frac{K_i^{app}}{\left(1 + \frac{[S]}{K_M}\right)} \text{ with}$$

$$\sigma_{K_i} = \sqrt{\left(\frac{1}{1 + \frac{[S]}{K_M}}\right)^2 \sigma_{k_{chem}^{app}}^2 + \left(-\frac{K_i^{app} K_M}{(K_M + [S])^2}\right)^2 \sigma_{[S]}^2 + \left(\frac{K_i^{app} [S]}{([S] + K_M)^2}\right)^2 \sigma_{K_M}^2}$$

**Sample Calculation 4. Calculate  $k_{chem}$  from  $k_{chem}^{app}$**

Apparent inhibitor potency  $k_{chem}^{app}$  (in  $M^{-1}s^{-1}$ ) found in *Data Analysis Protocol 1B* for competitive one-step irreversible inhibitors is corrected for substrate competition to obtain inhibition potency  $k_{chem}$  (in  $M^{-1}s^{-1}$ ) with propagation of error. Use substrate concentration  $[S]$  (in  $M$ ) after reaction initiation and  $K_M$  (in  $M$ ) as determined for these specific assay conditions (buffer, temperature, enzyme, substrate).

$$k_{chem} = k_{chem}^{app} \left(1 + \frac{[S]}{K_M}\right) \text{ with}$$

$$\sigma_{k_{chem}} = \sqrt{\left(1 + \frac{[S]}{K_M}\right)^2 \sigma_{k_{chem}^{app}}^2 + \left(\frac{k_{chem}^{app}}{K_M}\right)^2 \sigma_{[S]}^2 + \left(-\frac{k_{chem}^{app} [S]}{K_M^2}\right)^2 \sigma_{K_M}^2}$$

**Sample Calculation 5. Calculate  $k_{inact}/K_I^{app}$  from  $k_{chem}^{app}$**

The linear slope  $k_{chem}^{app}$  (in  $M^{-1}s^{-1}$ ) found in *Data Analysis Protocol 1B* for two-step irreversible inhibitors equals  $k_{inact}/K_I^{app}$  when all inhibitor concentrations are non-saturating ( $[I] \leq 0.1K_I^{app}$ ). It is not possible to obtain individual values of  $k_{inact}$  and  $K_I$  from a linear graph, but it is possible to estimate the upper and lower limits:  $K_I^{app}$  is much larger than the highest inhibitor concentration if this concentration is non-saturating ( $K_I^{app} \gg [I]_{max}$ ). An unchanged slope upon constraining the Y-intercept  $k_{ctrl}$  (step 5) to the experimental value for the uninhibited control validates that all inhibitor concentrations are non-saturating (Fig. 3F) rather than saturating (Fig. 3G). Proceed to *Sample Calculation 6* to calculate  $k_{inact}/K_I$ .

$$k_{chem}^{app} = \left(\frac{k_{inact}}{K_I^{app}}\right)$$

**Sample Calculation 6. Calculate  $k_{inact}/K_I$  from  $k_{inact}/K_I^{app}$**

Apparent inactivation potency  $k_{inact}/K_I^{app}$  (in  $M^{-1}s^{-1}$ ) found in *Data Analysis Protocols (1A or 1D)* or calculated in *Sample Calculation 5* for competitive two-step irreversible inhibitors is corrected for substrate competition to obtain  $k_{inact}/K_I$  (in  $M$ ) with propagation of error. Use substrate concentration  $[S]$  (in  $M$ ) after reaction initiation and  $K_M$  (in  $M$ ) as determined for these specific assay conditions (buffer, temperature, enzyme, substrate).

$$\frac{k_{inact}}{K_I} = \left(\frac{k_{inact}}{K_I^{app}}\right) \left(1 + \frac{[S]}{K_M}\right) \text{ with}$$

$$\sigma_{\frac{k_{inact}}{K_I}} = \sqrt{\left(1 + \frac{[S]}{K_M}\right)^2 \sigma_{\left(\frac{k_{inact}}{K_I^{app}}\right)}^2 + \left(\frac{\left(\frac{k_{inact}}{K_I^{app}}\right)}{K_M}\right)^2 \sigma_{[S]}^2 + \left(-\frac{\left(\frac{k_{inact}}{K_I^{app}}\right) [S]}{K_M^2}\right)^2 \sigma_{K_M}^2}$$

**Sample Calculation 7. Calculate  $K_i^*$  from  $K_i^{*app}$**

Apparent steady-state inhibition constant  $K_i^{*app}$  (in  $M$ ) found in *Data Analysis Protocols (1C or 3C)* for competitive two-step reversible covalent inhibitors is corrected for substrate competition to obtain steady-state inhibition constant  $K_i^*$  (in  $M$ ). Use substrate concentration  $[S]$  (in  $M$ ) after reaction initiation and  $K_M$  (in  $M$ ) as determined for these



specific assay conditions (buffer, temperature, enzyme, substrate).

$$K_i^* = \frac{K_i^{*app}}{\left(1 + \frac{[S]}{K_M}\right)} \text{ with}$$

$$\sigma_{K_i^*} = \sqrt{\left(\frac{1}{1 + \frac{[S]}{K_M}}\right)^2 \sigma_{K_i^{*app}}^2 + \left(-\frac{K_i^{*app} K_M}{(K_M + [S])^2}\right)^2 \sigma_{[S]}^2 + \left(\frac{K_i^{*app} [S]}{([S] + K_M)^2}\right)^2 \sigma_{K_M}^2}$$

**Sample Calculation 8. Calculate  $K_i^*$  from  $K_i$ ,  $k_5$ , and  $k_6$**

Steady-state inhibition constant  $K_i^*$  (in M) of two-step reversible inhibitors can be calculated from experimental values of  $K_i$  (in M),  $k_5$  (in  $s^{-1}$ ), and  $k_6$  (in  $s^{-1}$ ) found with *Data Analysis Protocols 1C* or *3C*, and *Sample Calculation 3*. Reliable (relatively) small  $k_6$ -values can only be obtained with more sensitive methods such as rapid dilution assays (Copeland, 2013e; Copeland et al., 2011). The uninhibited control must be strictly linear ( $k_{ctrl} = 0$ ) for values found with *Data Analysis Protocol 1C*. This calculation is not the preferred method to obtain  $K_i^*$  due to its sensitivity to (experimental) errors in  $k_6$  and contribution of  $k_{ctrl}$ : values obtained in *Data Analysis Protocol 1C* or *3C* and *Sample Calculation 7* should generally be considered as more reliable.

$$K_i^* = \frac{K_i}{\left(1 + \frac{k_5}{k_6}\right)} \text{ with}$$

$$\sigma_{K_i^*} = \sqrt{\left(\frac{1}{1 + \frac{k_5}{k_6}}\right)^2 \sigma_{K_i}^2 + \left(-\frac{K_i k_6}{(k_6 + k_5)^2}\right)^2 \sigma_{k_5}^2 + \left(\frac{K_i k_5}{(k_5 + k_6)^2}\right)^2 \sigma_{k_6}^2}$$

**Sample Calculation 9. Calculate  $k_{inact}/K_I$  from  $k_{chem}$**

The linear slope  $k_{chem}$  (in  $M^{-1}s^{-1}$ ) found in *Data Analysis Protocols (3Bi, 3Bii, 4Bi or 4Bii)* for two-step irreversible inhibitors equals  $k_{inact}/K_I$  when all inhibitor concentrations are non-saturating ( $[I] \leq 0.1K_I$ ). It is not possible to obtain individual values of  $k_{inact}$  and  $K_I$  from a linear graph, but it is possible to estimate the upper and lower limits:  $K_I$  is much larger than the highest inhibitor concentration if this concentration is non-saturating ( $K_I \gg [I]_{max}$ ). An unchanged slope upon constraining the Y-intercept  $k_{ctrl}$  to the experimental value for the uninhibited control in step 4 of *Basic Data Analysis Protocols (3Bi and 4Bi)* validates that all inhibitor concentrations are non-saturating (Fig. 3F) rather than saturating (Fig. 3G).

$$k_{chem} = \left(\frac{k_{inact}}{K_I}\right)$$

**Sample Calculation 10. Calculate  $k_{chem}$  or  $k_{inact}/K_I$  from  $k_{obs}$  and  $[I]$**

Divide the  $k_{obs}$ -value (in  $s^{-1}$ ) obtained in *Alternative Data Analysis Protocols (3Bii or 4Bii)* by its corresponding inhibitor concentration (in M) to calculate irreversible inhibitor potency  $k_{chem}$  (in  $M^{-1}s^{-1}$ ) or  $k_{inact}/K_I$  (in  $M^{-1}s^{-1}$ ). This calculation is only accurate for normalized  $k_{obs}$  values (unaffected by contribution of  $k_{ctrl}$ ), in absence of competing substrate, and (only applicable for two-step irreversible inhibitors) at non-saturating inhibitor concentration.

$$k_{chem} = \frac{k_{obs}}{[I]}$$

$$\left(\frac{k_{inact}}{K_I}\right) = \frac{k_{obs}}{[I]}$$

## KINETIC SIMULATIONS

The figures illustrating the basic protocols are generated using kinetic simulation scripts. These scripts are available online (<https://tinyurl.com/kineticsimulations>) and can be used to validate the obtained kinetic parameters or help in optimizing your assay. On a more educational level, these scripts can show what your assay result could look like when using wildly different parameters to obtain more insight into how these affect your assay.

### Materials

- Kinetic Simulation Script (<https://tinyurl.com/kineticsimulations>)
- Software to open csv file (e.g., EXCEL)
- Data fitting software (e.g., GraphPad Prism)
- Experimental values found in *Data Analysis Protocols 1-4*

### Kinetic Simulation 1. Validation of experimental values

Perform kinetic simulations to validate that calculated kinetic parameters are in accordance with experimental RAW data. A tutorial on how to perform kinetic simulations can be found on the website of our kinetic simulation scripts. Estimate microscopic rate constants from reported (literature) values, or use association rate constants  $k_1 = k_3 = 10^6\text{-}10^9 \text{ M}^{-1}\text{s}^{-1}$  (rapid noncovalent association) to calculate the dissociation rate constants from the experimental equilibrium constants:  $k_4 = K_1 \times k_3$  (Table S2 in Supporting Information) and  $k_2 = (K_M \times k_1) - k_{\text{cat}}$  (Table S3 in Supporting Information). Ideally, also simulate the HTS reaction conditions to validate that the calculated kinetic constants give rise to the experimental inhibition/IC<sub>50</sub> (Pollard & De La Cruz, 2013).

### Kinetic Simulation 2. Rational design of validation assays

Perform kinetic simulations with the calculated kinetic parameters to rationalize assay conditions for subsequent validation assays such as the minimum/maximum (pre)incubation times for reversibility assays or MS-detection of the covalent adduct (equations can be found in the Supporting Information).

## COMMENTARY

### Background Information

The background of covalent inhibition kinetics and critical parameters for enzyme activity assays can be found in the *Strategic Planning* section. It is recommended to refer to this section before setting up your kinetic inhibition experiments as well as the core references by Copeland (Copeland, 2000, 2013e) to get a general background on enzyme activity assays. We would like to reiterate that good experimental performance is essential for obtaining reliable parameters for your covalent inhibitor.

Our kinetic simulation scripts can help validate the found values by ‘rerunning’ the experiment without human error or experimental artifacts. Not only will this give insight into the reliability of your assay, but it can also help to improve the assay setup and can show what wildly different values of concentrations

would do for the readout. In fact, figures in this manuscript have been created this way, and can as such be reproduced. Keep in mind that these are simulations, and real-life examples will always deviate due to machine artifacts or pipetting errors. Nevertheless, with a working activity assay and these instructions in hand, adequate analysis of covalent inhibitors should be very feasible.

### Troubleshooting

Like with any experimental method, our described methods will also require the necessary optimization. Since data analysis depends heavily on the experimental input, it is very important to optimize assay conditions, rather than trying to apply data corrections, to obtain reliable kinetic parameters. As the assay conditions will vary widely, depending on the enzyme used (Bisswanger, 2014), we can only

**Table 3** Troubleshooting and Optimization Experimental Assay Conditions

Problem	Possible cause	Solutions
Difference positive and negative control is not significant (poor $Z'$ -score)	Enzyme is not active (enough)	Increase [E] (not always possible with very potent inhibitors) Increase [S] to increase absolute maximum signal Optimize buffer components Switch to a substrate that is processed faster Activate enzyme with fresh reagents (e.g., DTT, ATP) in single-use aliquots Minimize freeze/thaw cycles
	Signal product is not significant compared to substrate	Change fluorophore/read-out Optimize buffer components
	Negative control or inhibitor does not inhibit	Change to reported (specific) inhibitor Use thiol-alkylating reagent (e.g., NEM, IAc) for cysteines Use no-enzyme as negative control Increase concentration of inhibitor Make fresh dilution/aliquots of inhibitor solution
	DMSO in positive control acts as inhibitor	<i>Validate</i> : compare enzyme activity with/without DMSO Reduce DMSO to max. 1% of final solution
	Machine settings/sensitivity	Check if [P] is within the sensitivity range of used machine Optimize gain settings for [P] = 0–20% [S] <sub>0</sub> Check if correct wavelengths/settings are selected
	Pipetting error	Frequently replace pipette tips to avoid contamination of positive control with inhibitor (from negative control) Avoid well-to-well contamination by using an automated dispenser
Nonlinear uninhibited product formation curve $F_{\text{ctrl}}$	Substrate depletion ( $[P]_t > 0.1[S]_0$ )	Decrease [E] Increase [S] Shorter incubation time
	Spontaneous inactivation of enzyme ( $k_{\text{deg}} > 0$ )	Optimize buffer conditions for stability Use non-binding surface plates Shorter incubation time
	Drift/evaporation	Cover/seal plate with optical clear cover Shorter incubation time
	Pre-steady state kinetics (lag phase)	Increase [S] to reach $E + S \rightleftharpoons ES$ equilibrium faster Preincubate enzyme with reducing agent/ATP
	Solution is not homogeneous	Introduce mixing step before addition of final component
	Fluorescence bleaching/quenching	Optimize excitation conditions (e.g., lower no. of flashes) Longer measurement intervals/less measurements
	Linear inhibited progress curve $F_t$	Inhibition is not time-dependent (or $k_{\text{obs}}$ is too slow)

(Continued)

**Table 3** Troubleshooting and Optimization Experimental Assay Conditions, *continued*

Problem	Possible cause	Solutions
$F_0$ is not constant	Delay between enzyme addition and read-out	Reduce [E] (less substrate conversion during delay) Correcting $t = 0$ for actual time after addition Use injector in plate reader Validate row effect: change lay-out of plate (first well has higher $F_0$ than last well, but containing same components) and reduce number of samples in one measurement.
	Fluorescence interference inhibitor	Validate: check $F_0$ for inhibitor (no substrate and enzyme), substrate (no enzyme) and substrate and inhibitor (no enzyme) Exclude high [I] Background subtraction (subtract values substrate/inhibitor without enzyme from enzyme/substrate/inhibitor signal)
	Pipetting error substrate	Check for bubbles when pipetting Use low-binding tips
Full initial inhibition for all [I] ( $v_i = 0$ )	Noncovalent affinity is too potent ( $[I] \gg K_i^{app}$ )	Reduce [I] Higher [S] to increase competition (higher $K_i^{app}$ ) Use method based on covalency ( <i>Method IV</i> or direct detection)
	$k_{obs}$ is too fast for detection/resolvable range (inhibition is not slow-binding)	Shorter minimal (pre)incubation time Higher [S] to increase competition (slower $k_{obs}$ ) Reduce [I] (slower $k_{obs}$ )
$k_{obs}$ values are low compared to uninhibited control $k_{ctrl}$	Enzyme is unstable (high $k_{ctrl}$ )	Optimize assay conditions to improve linearity of uninhibited control (lower $k_{ctrl}$ ) Use preincubation protocol ( <i>Method III &amp; IV</i> ): higher $k_{obs}$ without competition
	Enzyme is not reactive (low $k_{obs}$ )	Optimize buffer conditions to increase enzyme reactivity Add (fresh) reagents (e.g., DTT, ATP) in single-use aliquots Validate with different enzyme batch/construct Too many freeze/thaw cycles
	Low inhibitor concentration ( $[I] \ll K_i^{app}$ )	Decrease [S] to reduce competition Increase [I] Use preincubation protocol ( <i>Method III &amp; IV</i> ): higher $k_{obs}$ without competition
	Slow reaction $k_{obs}$	Reduce [S] (less competition) Longer (pre)incubation time ( $t > 0.1t_{1/2}$ ) Use preincubation protocol ( <i>Method III &amp; IV</i> ): higher $k_{obs}$ without competition Optimize buffer conditions to increase enzyme reactivity
$k_{obs}$ vs [I] is linear	Inhibitor has 1-step binding mode	Validate: Y-intercept = $k_{ctrl}$ in $k_{obs}$ vs [I] plot Validate: $v_i = v_0^{ctrl}$ in $[P]_t$ vs $t$ or $v_t$ vs $t'$ plots Increase [I] to exclude 2-step $[I] \ll K_i^{app}$ Decrease [S] to exclude 2-step $[I] \ll K_i^{app}$
	2-step IRREV inhibitor is non-saturating ( $[I] \ll K_i^{app}$ )	Validate: Y-intercept = $k_{ctrl}$ in $k_{obs}$ vs [I] plot Validate: $v_i = v_0^{ctrl}$ in $[P]_t$ vs $t$ or $v_t$ vs $t'$ plots Fit $k_{obs}$ vs [I] to linear function for combined value $k_{inact}/K_i$ Increase [I] Decrease [S] to reduce competition (lower $K_i^{app}$ ) Use preincubation protocol ( <i>Method III &amp; IV</i> ): no competition

*(Continued)*

**Table 3** Troubleshooting and Optimization Experimental Assay Conditions, *continued*

Problem	Possible cause	Solutions
	2-step IRREV inhibitor is saturating ( $[I] \gg K_i^{app}$ )	<i>Validate:</i> Y-intercept $> k_{ctrl}$ in $k_{obs}$ vs $[I]$ plot <i>Validate:</i> $v_i < v_0^{ctrl}$ in $[P]_t$ vs $t$ or $v_t$ vs $t'$ plots Decrease $[I]$ Increase $[S]$ to increase competition (higher $K_i^{app}$ )
$k_{obs}$ decreases with increasing $[I]$	Inhibitor concentration beyond resolvable range: noncovalent affinity is too potent ( $[I] \gg K_i^{app}$ )	Optimize $[I]$ range ( $v_i = 0.1-0.9 \times v^{ctrl}$ ) Increase $[S]$ (increase competition to increase $K_i^{app}$ ) Exclude unlikely values from fit
	Incorrect formula to calculate $k_{obs}$	Validate if correct equation is used to determine $k_{obs}$ ; reversible covalent/irreversible covalent, one-step/two-step etc.

give general pointers on the optimization of the assay conditions (Table 3). Luckily, many model substrates come with a satisfactory user manual or are described in extensive methods papers (e.g., (Dharadhar et al., 2019; Janssen et al., 2019)). These resources generally state reagents required for the reaction (e.g., fresh reducing agent, for cysteine-based catalysis) or additives that stabilize the readout (such as BSA or Tween-20, to prevent aspecific aggregation). The control for full inhibition of (catalytic) cysteines is typically a thiol-alkylating reagent such as iodoacetamide (IAc) or N-methylmaleimide (NEM), or a known inhibitor.

As the assay performance is essential to get reliable fits, we recommend focusing on potential experimental problems before looking into issues with fitting. A great guide for general assay optimization can be found at the National Center for Advancing Translational Sciences (Assay Guidance Manual [Internet], 2004-2021; see Internet Resources). Here, we have supplied a comprehensive troubleshooting table with potential solutions that deal with various issues causing a troublesome readout. For the top half of the table, these solutions are generally related to the assay conditions and can generally be executed in the optimization stage.

The latter half of the table is more geared towards after the data analysis of an initial experiment. The problems and accompanying solutions deal more with the experimental setup: how much inhibitor or substrate one needs to add becomes more apparent after these first data points. Some solutions, like

changing inhibitor or substrate concentrations, can be simulated with our set of interactive kinetic simulation scripts. For better understanding and help in optimizing, we recommend simulating these conditions with our scripts to see what would happen when changing the concentrations.

## Abbreviations and Symbols

### Abbreviations

ATP	Adenosine Triphosphate
AU	Arbitrary Units
CYP450	Cytochrome P450
IAc	Iodoacetamide
IRREV	Irreversible
MS	Mass Spectrometry
NBS	Non-binding Surface
NMR	Nuclear Magnetic Resonance
M	Concentration in mol/L
NEM	N-ethylmaleimide
REV	Reversible
SAR	Structure-Activity Relationship
TCI	Targeted Covalent Inhibition
TDI	Time-Dependent Inactivation
PK-PD	Pharmacokinetics- Pharmacodynamics
E	unbound enzyme
I	unbound inhibitor
EI	noncovalent enzyme-inhibitor complex
EI*	covalent enzyme-inhibitor adduct
S	unbound substrate
ES	noncovalent enzyme-substrate complex
P	(detectable) product

## Symbols

$k_1$	Second-order association rate constant for $E + S \rightleftharpoons ES$ reaction (in $M^{-1}s^{-1}$ )
$k_2$	First-order dissociation rate constant for $E + S \rightleftharpoons ES$ equilibrium (in $s^{-1}$ )
$k_3$	Second-order association rate constant for $E + I \rightleftharpoons EI$ reaction (in $M^{-1}s^{-1}$ )
$k_4$	First-order dissociation rate constant for $E + I \rightleftharpoons EI$ equilibrium (in $s^{-1}$ )
$k_5$	First-order association rate constant for $EI \rightarrow EI^*$ reaction (in $s^{-1}$ )
$k_6$	First-order dissociation rate constant for $EI \rightleftharpoons EI^*$ equilibrium (in $s^{-1}$ )
$F_t$	Detected signal reflecting product formation in presence of inhibitor after incubation $t$ (in AU)
$F^{ctrl}$	Detected signal reflecting product formation in the uninhibited control (in AU)
$F_0$	Background signal at reaction initiation (in AU)
$r_P$	Product coefficient for detected signal per formed product (in AU/M)
$v_i$	Initial product formation velocity in presence of inhibitor (in AU/s)
$v_s$	Steady-state/final product formation velocity in presence of inhibitor (in AU/s)
$v_t'$	Product formation velocity after preincubation $t'$ (in AU/s)
$v_t^{ctrl}$	Product formation velocity in the uninhibited control (in AU/s)
$v_t'^{ctrl}$	Product formation velocity in the uninhibited control after preincubation $t'$ (in AU/s)
$v_0^{ctrl}$	Product formation velocity in the uninhibited control without preincubation: $t'=0$ (in AU/s)
$t$	Incubation time after onset of product formation (in s)
$t'$	Preincubation time after onset of enzyme inhibition (in s)
$t_{1/2}$	Half-life for reaction progress (in s).
$t_{1/2}^{diss}$	Half-life for dissociation reaction (in s)
$\tau$	Target residence time (in s)
$k_{obs}$	Observed reaction rate constant (in $s^{-1}$ )
$k_{max}$	Maximum reaction rate constant at saturating inhibitor concentration for 2-step inhibition (in $s^{-1}$ )
$k_{inact}$	Inactivation rate constant for $EI \rightarrow EI^*$ at saturating inhibitor concentration for 2-step irreversible inhibition (in $s^{-1}$ )
$k_{ctrl}$	Reaction rate constant for nonlinearity or loss of enzyme activity in uninhibited control (in $s^{-1}$ )
$k_{degE}$	Enzyme degradation rate constant for $E \rightarrow E_{deg}$ (in $s^{-1}$ )
$k_{cat}$	Product formation rate constant for $ES \rightarrow E + P$ (in $s^{-1}$ ) at saturating substrate concentration
$k_{sub}$	Reaction rate constant for $E + S \rightarrow E + P$ (in $M^{-1}s^{-1}$ ) ( $= k_{cat}/K_M$ if $[S] \ll 0.1K_M$ )
$k_{chem}$	Reaction rate constant for $E + I \rightarrow EI^*$ of 1-step irreversible inhibitors (in $M^{-1}s^{-1}$ )
$k_{off}$	Overall dissociation rate constant from bound to unbound enzyme $EI + EI^* \rightarrow E + I$ (in $s^{-1}$ )
$K_i$	Inhibition/dissociation constant (in M) for noncovalent $E + I \rightleftharpoons EI$ equilibrium of two-step inhibition
$K_i^{app}$	Apparent noncovalent inhibition constant (in M): with substrate competition
$K_i^*$	Steady-state inhibition constant (in M) for $E + I \rightleftharpoons EI + EI^*$ equilibrium of two-step reversible inhibition
$K_i^{*app}$	Apparent steady-state inhibition constant (in M): with substrate competition
$K_I$	Inactivation constant for $E + I \rightarrow EI^*$ (in M) of two-step irreversible inhibition
$K_I^{app}$	Apparent inactivation constant (in M): with substrate competition
$K_M$	Michaelis-Menten constant for $E + S \rightarrow E + P$ (in M)
$k_{inact}/K_I$	Inactivation efficiency: reaction rate constant for $E + I \rightarrow EI^*$ of 2-step irreversible inhibitors (in $M^{-1}s^{-1}$ )
$IC_{50}$	Inhibitor concentration resulting in half-maximum inhibition (in M)
$IC_{50}(t)$	Inhibitor concentration resulting in half-maximum inhibition after incubation time $t$ (in M)
$[E^{total}]$	Combined total concentration of all enzyme species ( $E^{total} = E + EI + EI^* + ES + E_{deg} + EI_{deg} + EI_{deg}^* + ES_{deg}$ )
$[E]_0$	Unbound enzyme concentration at reaction initiation (before binding to inhibitor/substrate)

[I] <sub>0</sub>	Unbound inhibitor concentration at onset of inhibition (before binding to enzyme)
[S] <sub>0</sub>	Unbound substrate concentration at onset of product formation (before binding to enzyme)
[EI] <sub>eq</sub>	Noncovalent EI concentration at (steady-state) equilibrium
[X] <sub>0</sub>	Concentration of component X at reaction initiation (before binding to other reaction components)
[X] <sub>t</sub>	Concentration of component X at incubation time <i>t</i>
[X] <sub>t'</sub>	Concentration of component X at preincubation time <i>t'</i>
V <sub>t</sub>	Incubation reaction volume containing enzyme, inhibitor and substrate ( $V_t = V_{t'} + V_{\text{sub}}$ )
V <sub>t'</sub>	Preincubation reaction volume containing enzyme and inhibitor
V <sub>sub</sub>	Volume containing substrate

### Acknowledgments

This work was supported by the EU/EFPIA/OICR/McGill/KTH/Diamond Innovative Medicines Initiative 2 Joint Undertaking (EUOPEN grant no. 875510). The authors would like to thank Dr. Anthe Janssen for proof reading. In memory of Prof. Dr. Huib Ovaa, his passion for science will always be an inspiration to us.

### Author Contributions

**Elma Mons:** conceptualization, formal analysis, investigation, visualization, writing original draft, writing review and editing; **Sander Roet:** conceptualization, resources, software, writing review and editing; **Robbert Kim:** supervision, validation, writing original draft, writing review and editing; **Monique Mulder:** project administration, supervision, writing review and editing.

### Conflict of Interest

The authors declare no conflict of interest.

### Data Availability Statement

The authors confirm that the data supporting the findings of this study are available within the article [and/or] its supplementary materials.

### Literature Cited

Abdeldayem, A., Raouf, Y. S., Constantinescu, S. N., Moriggl, R., & Gunning, P. T. (2020). Advances in covalent kinase inhibitors doi: 10.1039/C9CS00720B]. *Chemical Society Reviews*, 49(9), 2617–2687. doi: 10.1039/C9CS00720B.

Acker, M. G., & Auld, D. S. (2014). Considerations for the design and reporting of enzyme assays in high-throughput screening applications. *Perspectives in Science*, 1(1), 56–73. doi: 10.1016/j.pisc.2013.12.001.

Assay Guidance Manual (2004–2021). Eli Lilly & Company and the National Center for Advancing Translational Sciences. <http://www.ncbi.nlm.nih.gov/books/NBK53196/>.

Auld, D. S., Inglese, J., & Dahlin, J. L. (2017). Assay Interference by Aggregation. In *Assay Guidance Manual [Internet]*. Eli Lilly & Company and the National Center for Advancing Translational Sciences. Available at <https://www.ncbi.nlm.nih.gov/books/NBK442297/>.

Barf, T., & Kaptein, A. (2012). Irreversible protein kinase inhibitors: Balancing the benefits and risks. *Journal of Medicinal Chemistry*, 55(14), 6243–6262. doi: 10.1021/jm3003203.

Bauer, R. A. (2015). Covalent inhibitors in drug discovery: From accidental discoveries to avoided liabilities and designed therapies. *Drug Discovery Today*, 20(9), 1061–1073. doi: 10.1016/j.drudis.2015.05.005.

Bisswanger, H. (2014). Enzyme assays. *Perspectives in Science*, 1(1), 41–55. doi: 10.1016/j.pisc.2014.02.005.

Bradshaw, J. M., McFarland, J. M., Paavilainen, V. O., Bisconte, A., Tam, D., Phan, V. T., ... Taunton, J. (2015). Prolonged and tunable residence time using reversible covalent kinase inhibitors. *Nature Chemical Biology*, 11(7), 525–531. doi: 10.1038/nchembio.1817.

Cheng, Y.-C., & Prusoff, W. H. (1973). Relationship between the inhibition constant (KI) and the concentration of inhibitor which causes 50 per cent inhibition (I50) of an enzymatic reaction. *Biochemical Pharmacology*, 22(23), 3099–3108. doi: 10.1016/0006-2952(73)90196-2.

Copeland, R. A. (2000). *ENZYMES: A Practical Introduction to Structure, Mechanism, and Data Analysis, Second Edition*. John Wiley & Sons, Inc. doi: 10.1002/0471220639.

Copeland, R. A. (2010). The dynamics of drug-target interactions: Drug-target residence time and its impact on efficacy and safety. *Expert Opinion on Drug Discovery*, 5(4), 305–310. doi: 10.1517/17460441003677725.

Copeland, R. A. (2013a). *APPENDIX 1: Kinetics of Biochemical Reactions*. In *Evaluation of Enzyme Inhibitors in Drug Discovery: A Guide for Medicinal Chemists and Pharmacologists* (Second Edition, pp. 471–482). John Wiley & Sons, Inc. doi: 10.1002/9781118540398.app1.

Copeland, R. A. (2013b). Chapter 6. Slow Binding Inhibitors. In *Evaluation of Enzyme Inhibitors in Drug Discovery: A Guide for Medicinal Chemists and Pharmacologists* (Second



- Edition, pp. 203–244). John Wiley & Sons, Inc. doi: 10.1002/9781118540398.ch6.
- Copeland, R. A. (2013c). Chapter 7. Tight binding inhibition. In *Evaluation of Enzyme Inhibitors in Drug Discovery: A Guide for Medicinal Chemists and Pharmacologists* (Second Edition, pp. 245–285). John Wiley & Sons, Inc. doi: 10.1002/9781118540398.ch7.
- Copeland, R. A. (2013d). Chapter 9. Irreversible Enzyme Inactivators. In *Evaluation of Enzyme Inhibitors in Drug Discovery: A Guide for Medicinal Chemists and Pharmacologists* (Second Edition, pp. 345–382). John Wiley & Sons, Inc. doi: 10.1002/9781118540398.
- Copeland, R. A. (2013e). *Evaluation of Enzyme Inhibitors in Drug Discovery: A Guide for Medicinal Chemists and Pharmacologists* (Second Edition). John Wiley & Sons, Inc. doi: 10.1002/9781118540398.
- Copeland, R. A., Basavapathruni, A., Moyer, M., & Scott, M. P. (2011). Impact of enzyme concentration and residence time on apparent activity recovery in jump dilution analysis. *Analytical Biochemistry*, 416(2), 206–210. doi: 10.1016/j.ab.2011.05.029.
- Copeland, R. A., Pompliano, D. L., & Meek, T. D. (2006). Drug–target residence time and its implications for lead optimization. *Nature Reviews Drug Discovery*, 5(9), 730–739. doi: 10.1038/nrd2082.
- Dalton, S. E., & Campos, S. (2020). Covalent small molecules as enabling platforms for drug discovery. *ChemBioChem*, 21(8), 1080–1100. doi: 10.1002/cbic.201900674.
- De Cesco, S., Kurian, J., Dufresne, C., Mittermaier, A. K., & Moitessier, N. (2017). Covalent inhibitors design and discovery. *European Journal of Medicinal Chemistry*, 138, 96–114. doi: 10.1016/j.ejmech.2017.06.019.
- Dharadhar, S., Kim, R. Q., Uckelmann, M., & Sixma, T. K. (2019). Chapter Thirteen - Quantitative analysis of USP activity in vitro. In M. Hochstrasser (Ed.), *Methods in Enzymology* (Vol. 618, 281–319). Academic Press. doi: 10.1016/bs.mie.2018.12.023.
- Engel, J., Richters, A., Getlik, M., Tomassi, S., Keul, M., Termathe, M., ... Rauh, D. (2015). Targeting drug resistance in EGFR with covalent inhibitors: A structure-based design approach. *Journal of Medicinal Chemistry*, 58(17), 6844–6863. doi: 10.1021/acs.jmedchem.5b01082.
- Fell, J. B., Fischer, J. P., Baer, B. R., Blake, J. F., Bouhana, K., Briere, D. M., ... Marx, M. A. (2020). Identification of the clinical development candidate MRTX849, a covalent KRASG12C inhibitor for the treatment of cancer. *Journal of Medicinal Chemistry*, 63(13), 6679–6693. doi: 10.1021/acs.jmedchem.9b02052.
- Ferrall-Fairbanks, M. C., Kieslich, C. A., & Platt, M. O. (2020). Reassessing enzyme kinetics: Considering protease-as-substrate interactions in proteolytic networks. *Proceedings of the National Academy of Sciences*, 117(6), 3307. doi: 10.1073/pnas.1912207117.
- Gabizon, R., & London, N. (2020). A fast and clean BTK inhibitor. *Journal of Medicinal Chemistry*, 63(10), 5100–5101. doi: 10.1021/acs.jmedchem.0c00597.
- Gehring, M., & Laufer, S. A. (2019). Emerging and Re-Emerging Warheads for Targeted Covalent Inhibitors: Applications in Medicinal Chemistry and Chemical Biology. *Journal of Medicinal Chemistry*, 62(12), 5673–5724. doi: 10.1021/acs.jmedchem.8b01153.
- Guan, I., Williams, K., Pan, J., & Liu, X. (2021). New cysteine covalent modification strategies enable advancement of proteome-wide selectivity of kinase modulators. *Asian Journal of Organic Chemistry*, 10(5), 949–963. doi: 10.1002/ajoc.202100036.
- Hansen, R., Peters, U., Babbar, A., Chen, Y., Feng, J., Janes, M. R., ... Zarrinkar, P. P. (2018). The reactivity-driven biochemical mechanism of covalent KRASG12C inhibitors. *Nature Structural & Molecular Biology*, 25(6), 454–462. doi: 10.1038/s41594-018-0061-5.
- Harris, C. M., Foley, S. E., Goedken, E. R., Michalak, M., Murdock, S., & Wilson, N. S. (2018). Merits and pitfalls in the characterization of covalent inhibitors of bruton's tyrosine Kinase. *SLAS DISCOVERY: Advancing the Science of Drug Discovery*, 23(10), 1040–1050. doi: 10.1177/2472555218787445.
- Holdgate, G. A., Meek, T. D., & Grimley, R. L. (2017). Mechanistic enzymology in drug discovery: A fresh perspective [Review Article]. *Nature Reviews Drug Discovery*, 17, 115. doi: 10.1038/nrd.2017.219.
- Ito, K., Iwatsubo, T., Kanamitsu, S., Ueda, K., Suzuki, H., & Sugiyama, Y. (1998). Prediction of pharmacokinetic alterations caused by drug-drug interactions: Metabolic interaction in the liver. *Pharmacological Reviews*, 50(3), 387. Retrieved from <http://pharmrev.aspetjournals.org/content/50/3/387.abstract>.
- Janssen, A. P. A., van Hengst, J. M. A., Béquignon, O. J. M., Deng, H., van Westen, G. J. P., & van der Stelt, M. (2019). Structure kinetics relationships and molecular dynamics show crucial role for heterocycle leaving group in irreversible diacylglycerol lipase inhibitors. *Journal of Medicinal Chemistry*, 62(17), 7910–7922. doi: 10.1021/acs.jmedchem.9b00686.
- Johansson, H., Isabella Tsai, Y.-C., Fantom, K., Chung, C.-W., Kümper, S., Martino, L., ... Rittering, K. (2019). Fragment-based covalent ligand screening enables rapid discovery of inhibitors for the RBR E3 ubiquitin ligase HOIP. *Journal of the American Chemical Society*, 141(6), 2703–2712. doi: 10.1021/jacs.8b13193.
- Johnson, D. S., Weerapana, E., & Cravatt, B. F. (2010). Strategies for discovering and derisking covalent, irreversible enzyme inhibitors. *Future Medicinal Chemistry*, 2(6), 949–964. doi: 10.4155/fmc.10.21.
- Johnson, I. D. (2010). Introduction to fluorescence techniques. In *Molecular Probes Handbook:*



- A Guide to Fluorescent Probes and Labeling Technologies (11th edition, pp. 2–9). Life Technologies Corporation. Available at <https://www.thermofisher.com/nl/en/home/references/molecular-probes-the-handbook/introduction-to-fluorescence-techniques.html>.
- Johnson, K. A. (2009). Fitting enzyme kinetic data with kintek global kinetic explorer. *Methods in Enzymology*, 467, 601–626. doi: 10.1016/S0076-6879(09)67023-3.
- Kathman, S. G., Xu, Z., & Statsyuk, A. V. (2014). A fragment-based method to discover irreversible covalent inhibitors of cysteine proteases. *Journal of Medicinal Chemistry*, 57(11), 4969–4974. doi: 10.1021/jm500345q.
- Kathman, S. G., & Statsyuk, A. V. (2019). Methodology for identification of cysteine-reactive covalent inhibitors. In P. Hogg (Ed.), *Functional Disulphide Bonds: Methods and Protocols* (pp. 245–262). New York: Springer. doi: 10.1007/978-1-4939-9187-7\_15.
- Kim, H., Hwang, Y. S., Kim, M., & Park, S. B. (2021). Recent advances in the development of covalent inhibitors. *RSC Medicinal Chemistry*, 12(7), 1037–1045. doi: 10.1039/D1MD00068C.
- Kitz, R., & Wilson, I. B. (1962). Esters of methanesulfonic acid as irreversible inhibitors of acetylcholinesterase. *Journal of Biological Chemistry*, 237(10), 3245–3249. doi: 10.1016/S0021-9258(18)50153-8.
- Krippendorff, B.-F., Neuhaus, R., Lienau, P., Reichel, A., & Huisinga, W. (2009). Mechanism-based inhibition: Deriving  $K_I$  and  $k_{inact}$  directly from time-dependent  $IC_{50}$  values. *Journal of Biomolecular Screening*, 14(8), 913–923. doi: 10.1177/1087057109336751.
- Kuzmič, P. (2009). Chapter 10 - DynaFit—a software package for enzymology. In M. L. Johnson & L. Brand (Eds.), *Methods in Enzymology* (Vol. 467, 247–280). Academic Press. doi: 10.1016/S0076-6879(09)67010-5.
- Kuzmič, P. (2015). *Determination of  $k_{inact}$  and  $K_i$  for covalent inhibition using the Omnia® assay [BioKin Technical Note TN-2015-02]*. Woburn MA: BioKin Ltd., [Online]. Available at [www.biokin.com/TN/2015/02](http://www.biokin.com/TN/2015/02).
- Kuzmič, P. (2020a). A steady-state algebraic model for the time course of covalent enzyme inhibition. *bioRxiv*. doi: 10.1101/2020.06.10.144220.
- Kuzmič, P. (2020b). A two-point  $IC_{50}$  method for evaluating the biochemical potency of irreversible enzyme inhibitors. *bioRxiv*. doi: 10.1101/2020.06.25.171207.
- Kuzmič, P., Solowiej, J., & Murray, B. W. (2015). An algebraic model for the kinetics of covalent enzyme inhibition at low substrate concentrations. *Analytical Biochemistry*, 484, 82–90. doi: 10.1016/j.ab.2014.11.014.
- Lagoutte, R., Patouret, R., & Winssinger, N. (2017). Covalent inhibitors: An opportunity for rational target selectivity. *Current Opinion in Chemical Biology*, 39, 54–63. doi: 10.1016/j.cbpa.2017.05.008.
- Lanman, B. A., Allen, J. R., Allen, J. G., Amegadzie, A. K., Ashton, K. S., Booker, S. K., ... Cee, V. J. (2020). Discovery of a covalent inhibitor of KRASG12C (AMG 510) for the treatment of solid tumors. *Journal of Medicinal Chemistry*, 63(1), 52–65. doi: 10.1021/acs.jmedchem.9b01180.
- Lee, C.-U., & Grossmann, T. N. (2012). Reversible covalent inhibition of a protein target. *Angewandte Chemie International Edition*, 51(35), 8699–8700. doi: 10.1002/anie.201203341.
- Liclican, A., Serafini, L., Xing, W., Czerwieńiec, G., Steiner, B., Wang, T., ... Feng, J. Y. (2020). Biochemical characterization of tirabrutinib and other irreversible inhibitors of Bruton's tyrosine kinase reveals differences in on - and off - target inhibition. *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1864(4), 129531. doi: 10.1016/j.bbagen.2020.129531.
- Lonsdale, R., Burgess, J., Colclough, N., Davies, N. L., Lenz, E. M., Orton, A. L., & Ward, R. A. (2017). Expanding the armory: Predicting and tuning covalent warhead reactivity. *Journal of Chemical Information and Modeling*, 57(12), 3124–3137. doi: 10.1021/acs.jcim.7b00553.
- Lu, S., & Zhang, J. (2017). Designed covalent allosteric modulators: An emerging paradigm in drug discovery. *Drug Discovery Today*, 22(2), 447–453. doi: 10.1016/j.drudis.2016.11.013.
- Mah, R., Thomas, J. R., & Shafer, C. M. (2014). Drug discovery considerations in the development of covalent inhibitors. *Bioorganic & Medicinal Chemistry Letters*, 24(1), 33–39. doi: 10.1016/j.bmcl.2013.10.003.
- Martin, J. S., MacKenzie, C. J., Fletcher, D., & Gilbert, I. H. (2019). Characterising covalent warhead reactivity. *Bioorganic & Medicinal Chemistry*, 27(10), 2066–2074. doi: 10.1016/j.bmc.2019.04.002.
- Mayer, J., Khairy, K., & Howard, J. (2010). Drawing an elephant with four complex parameters. *American Journal of Physics*, 78(6), 648–649. doi: 10.1119/1.3254017.
- McWhirter, C. (2021). Chapter One - Kinetic mechanisms of covalent inhibition. In R. A. Ward & N. P. Grimster (Eds.), *Annual Reports in Medicinal Chemistry* (Vol. 56, pp. 1–31). Academic Press. doi: 10.1016/bs.armac.2020.11.001.
- Meara, J. P., & Rich, D. H. (1995). Measurement of individual rate constants of irreversible inhibition of a cysteine proteinase by an epoxysuccinyl inhibitor. *Bioorganic & Medicinal Chemistry Letters*, 5(19), 2277–2282. doi: 10.1016/0960-894X(95)00396-B.
- Miyawaki, O., Kanazawa, T., Maruyama, C., & Dozen, M. (2017). Static and dynamic half-life and lifetime molecular turnover of enzymes. *Journal of Bioscience and Bioengineering*, 123(1), 28–32. doi: 10.1016/j.jbiosc.2016.07.016.
- Mons, E., Jansen, I. D. C., Loboda, J., van Doo-dewaerd, B. R., Hermans, J., Verdoes, M., ... Ovaas, H. (2019). The alkyne moiety as a latent electrophile in irreversible covalent small

- molecule inhibitors of cathepsin K. *Journal of the American Chemical Society*, 141(8), 3507–3514. doi: 10.1021/jacs.8b11027.
- Mons, E., Kim, R. Q., van Doodewaerd, B. R., van Veelen, P. A., Mulder, M. P. C., & Ovaa, H. (2021). Exploring the versatility of the covalent thiol–alkyne reaction with substituted propargyl warheads: A deciding role for the cysteine protease. *Journal of the American Chemical Society*, 143(17), 6423–6433. doi: 10.1021/jacs.0c10513.
- Motulsky, H. J. (1995–2021). Graphpad Curve Fitting Guide. *GraphPad Software, LLC*. [https://www.graphpad.com/guides/prism/latest/curve-fitting/reg\\_writing\\_models.htm](https://www.graphpad.com/guides/prism/latest/curve-fitting/reg_writing_models.htm).
- Motulsky, H. J., & Christopoulos, A. (2003). *Fitting Models to Biological Data Using Linear and Nonlinear Regression: A Practical Guide to Curve Fitting*. ISBN-13: 978-0195171808; ISBN-10: 1425919448. GraphPad Software Inc. Available at <https://www.amazon.com/Fitting-Models-Biological-Nonlinear-Regression/dp/0195171802>.
- Murphy, D. J. (2004). Determination of accurate  $K_i$  values for tight-binding enzyme inhibitors: An in silico study of experimental error and assay design. *Analytical Biochemistry*, 327(1), 61–67. doi: 10.1016/j.ab.2003.12.018.
- Obach, R. S., Walsky, R. L., & Venkatakrishnan, K. (2007). Mechanism-based inactivation of human cytochrome p450 enzymes and the prediction of drug–drug interactions. *Drug Metabolism and Disposition*, 35(2), 246. doi: 10.1124/dmd.106.012633.
- Owen Dafydd, R., Allerton Charlotte, M. N., Anderson Annaliesa, S., Aschenbrenner, L., Avery, M., Berritt, S., ... Zhu, Y. (2021). An oral SARS-CoV-2 Mpro inhibitor clinical candidate for the treatment of COVID-19. *Science*, eabl4784. doi: 10.1126/science.abl4784.
- Perrin, C. L. (2017). Linear or nonlinear least-squares analysis of kinetic data? *Journal of Chemical Education*, 94(6), 669–672. doi: 10.1021/acs.jchemed.6b00629.
- Pollard, T. D., & De La Cruz, E. M. (2013). Take advantage of time in your experiments: A guide to simple, informative kinetics assays. *Molecular Biology of the Cell*, 24(8), 1103–1110. doi: 10.1091/mbc.e13-01-0030.
- Potratz, J. P. (2018). Making enzyme kinetics dynamic via simulation software. *Journal of Chemical Education*, 95(3), 482–486. doi: 10.1021/acs.jchemed.7b00350.
- Ray, S., & Murkin, A. S. (2019). New electrophiles and strategies for mechanism-based and targeted covalent inhibitor design. *Biochemistry*, 58(52), 5234–5244. doi: 10.1021/acs.biochem.9b00293.
- Resnick, E., Bradley, A., Gan, J., Douangamath, A., Krojer, T., Sethi, R., ... London, N. (2019). Rapid covalent-probe discovery by electrophile-fragment screening. *Journal of the American Chemical Society*, 141(22), 8951–8968. doi: 10.1021/jacs.9b02822.
- Rocha-Pereira, J., Nascimento, M. S. J., Ma, Q., Hilgenfeld, R., Neyts, J., & Jochmans, D. (2014). The enterovirus protease inhibitor rupintrivir exerts cross-genotypic anti-norovirus activity and clears cells from the norovirus replicon. *Antimicrobial Agents and Chemotherapy*, 58(8), 4675–4681. doi: 10.1128/AAC.02546-13.
- Rufer, A. C. (2021). Drug discovery for enzymes. *Drug Discovery Today*, 26(4), 875–886. doi: 10.1016/j.drudis.2021.01.006.
- Schwartz, P. A., Kuzmic, P., Solowiej, J., Bergqvist, S., Bolanos, B., Almaden, C., ... Murray, B. W. (2014). Covalent EGFR inhibitor analysis reveals importance of reversible interactions to potency and mechanisms of drug resistance. *Proceedings of the National Academy of Sciences*, 111(1), 173. doi: 10.1073/pnas.1313733111.
- Selwyn, M. J. (1965). A simple test for inactivation of an enzyme during assay. *Biochimica et Biophysica Acta (BBA) - Enzymology and Biological Oxidation*, 105(1), 193–195. doi: 10.1016/S0926-6593(65)80190-4.
- Serafimova, I. M., Pufall, M. A., Krishnan, S., Duda, K., Cohen, M. S., Maglathlin, R. L., ... Taunton, J. (2012). Reversible targeting of non-catalytic cysteines with chemically tuned electrophiles. *Nature Chemical Biology*, 8(5), 471–476. doi: 10.1038/nchembio.925.
- Shindo, N., & Ojida, A. (2021). Recent progress in covalent warheads for in vivo targeting of endogenous proteins. *Bioorganic & Medicinal Chemistry*, 47, 116386. doi: 10.1016/j.bmc.2021.116386.
- Singh, J., Petter, R. C., Baillie, T. A., & Whitty, A. (2011). The resurgence of covalent drugs. *Nature Reviews Drug Discovery*, 10(4), 307–317. doi: 10.1038/nrd3410.
- Smith, S., Keul, M., Engel, J., Basu, D., Eppmann, S., & Rauh, D. (2017). Characterization of covalent-reversible EGFR inhibitors. *ACS Omega*, 2(4), 1563–1575. doi: 10.1021/acsomega.7b00157.
- Strelow, J. M. (2017). A perspective on the kinetics of covalent and irreversible inhibition. *SLAS DISCOVERY: Advancing Life Sciences R&D*, 22(1), 3–20. doi: 10.1177/1087057116671509.
- Stresser, D. M., Mao, J., Kenny, J. R., Jones, B. C., & Grime, K. (2014). Exploring concepts of in vitro time-dependent CYP inhibition assays. *Expert Opinion on Drug Metabolism & Toxicology*, 10(2), 157–174. doi: 10.1517/17425255.2014.856882.
- Telliez, J.-B., Dowty, M. E., Wang, L., Jussif, J., Lin, T., Li, L., ... Thorarensen, A. (2016). Discovery of a JAK3-selective inhibitor: Functional differentiation of JAK3-selective inhibition over pan-JAK or JAK1-selective inhibition. *ACS Chemical Biology*, 11(12), 3442–3451. doi: 10.1021/acscchembio.6b00677.
- Tuley, A., & Fast, W. (2018). The taxonomy of covalent inhibitors. *Biochemistry*, 57(24), 3326–3337. doi: 10.1021/acs.biochem.8b00315.

- Walkup, G. K., You, Z., Ross, P. L., Allen, E. K. H., Daryaei, F., Hale, M. R., ... Fisher, S. L. (2015). Translating slow-binding inhibition kinetics into cellular and in vivo effects. *Nature Chemical Biology*, 11(6), 416–423. doi: 10.1038/nchembio.1796.
- Ward, R. A., & Grimster, N. P. (2021). The design of covalent-based inhibitors. *Annual Reports in Medicinal Chemistry*, 56, 2–284. Available at <https://www.sciencedirect.com/bookseries/annual-reports-in-medicinal-chemistry/vol/56/suppl/C>.
- Wu, G., Yuan, Y., & Hodge, C. N. (2003). Determining appropriate substrate conversion for enzymatic assays in high-throughput screening. *Journal of Biomolecular Screening*, 8(6), 694–700. doi: 10.1177/1087057103260050.
- Yang, J., Jamei, M., Yeo, K. R., Tucker, G. T., & Rostami-Hodjegan, A. (2005). Kinetic values for mechanism-based enzyme inhibition: Assessing the bias introduced by the conventional experimental protocol. *European Journal of Pharmaceutical Sciences*, 26(3), 334–340. doi: 10.1016/j.ejps.2005.07.005.
- Zhai, X., Ward, R. A., Doig, P., & Argyrou, A. (2020). Insight into the therapeutic selectivity of the irreversible EGFR tyrosine kinase inhibitor osimertinib through enzyme kinetic studies. *Biochemistry*, 59(14), 1428–1441. doi: 10.1021/acs.biochem.0c00104.
- Zhang, J.-H., Chung, T. D. Y., & Oldenburg, K. R. (1999). A simple statistical parameter for use in evaluation and validation of high throughput screening assays. *Journal of Biomolecular Screening*, 4(2), 67–73. doi: 10.1177/108705719900400206.
- Zhang, T., Hatcher, J. M., Teng, M., Gray, N. S., & Kostic, M. (2019). Recent advances in selective and irreversible covalent ligand development and validation. *Cell Chemical Biology*, 26(11), 1486–1500. doi: 10.1016/j.chembiol.2019.09.012.

### Internet Resources

<https://tinyurl.com/kineticsimulations>

*Interactive kinetic simulation scripts used to generate figures in this work, including a tutorial on how to use them. Note: loading this page for the first time can take up to 5 min, and involves an automatic redirection (reloading) to the Landing Page. Full URL: <https://mybinder.org/v2/gh/sroet/Elma/main?labpath=Landing%20Page.ipynb>*

<https://www.ncbi.nlm.nih.gov/books/NBK53196/>

*Online assay guidance manual for general assay optimization managed by the National Center for Advancing Translational Sciences (NCATS)*

<https://www.graphpad.com/guides/prism/latest/curve-fitting/index.htm>

*Online guide on implementing user-defined equations for nonlinear regression in GraphPad Prism*



## Paper E

# Exchanging replicas with unequal cost, infinitely and permanently

Sander Roet, Daniel T. Zhang, and Titus S. van Erp

*arXiv preprint arXiv:2205.12663v1 [physics.comp-ph];*

doi: 10.48550/arXiv.2205.12663



# Exchanging replicas with unequal cost, infinitely and permanently

Sander Roet, Daniel T. Zhang, and Titus S. van Erp\*  
*Department of Chemistry,*  
*Norwegian University of Science and Technology (NTNU),*  
*N-7491 Trondheim, Norway*  
 (Dated: May 26, 2022)

We developed a replica exchange method that is effectively parallelizable even if the computational cost of the Monte Carlo moves in the parallel replicas are considerably different, for instance, because the replicas run on different type of processor units or because of the algorithmic complexity. To prove detailed-balance, we make a paradigm shift from the common conceptual viewpoint in which the set of parallel replicas represents a high-dimensional superstate, to an ensemble based criterion in which the other ensembles represent an environment that might or might not participate in the Monte Carlo move. In addition, based on a recent algorithm for computing permanents, we effectively increase the exchange rate to infinite without the steep factorial scaling as function of the number of replicas. We illustrate the effectiveness of the replica exchange methodology by combining it with a quantitative path sampling method, replica exchange transition interface sampling (RETIS), in which the costs for a Monte Carlo move can vary enormously as paths in a RETIS algorithm do not have the same length and the average path lengths tend to vary considerably for the different path ensembles that run in parallel. This combination, coined  $\infty$ RETIS, was tested on three model systems.

Keywords: Replica Exchange | Path Sampling | infinite swapping | Markov-chain Monte Carlo

The Markov chain Monte Carlo (MC) method is one of the most important numerical techniques for computing averages in high-dimensional spaces, like the configuration space of a many particle system. The approach has applications in a wide variety of fields ranging from computational physics, theoretical chemistry, economics, and genetics. The MC algorithm effectively generates a selective random walk through state space in which the artificial steps are designed such to ensure that the frequency of visiting any particular state is proportional to the equilibrium probability of that state. The Metropolis [1] or the more general Metropolis-Hastings [2] algorithms are the most common approaches for designing such random steps (MC moves) based on the detailed-balance principle. That is, the MC moves should be constructed such that the number of transition from an old state  $s^{(o)}$  to a new state  $s^{(n)}$  is exactly balanced by the number of transitions from the new to the old state:  $\rho(s^{(o)})\pi(s^{(o)} \rightarrow s^{(n)}) = \rho(s^{(n)})\pi(s^{(n)} \rightarrow s^{(o)})$  where,  $\rho(\cdot)$  is the state space equilibrium probability density and  $\pi(\cdot)$  are the probabilities to make a transition between the two states given the set of possible MC moves. Further, the transition is split into a generation and an acceptance/rejection step such that  $\pi(s \rightarrow s') = P_{\text{gen}}(s \rightarrow s')P_{\text{acc}}(s \rightarrow s')$ . In the case that the sampled state space is the configuration space of a molecular system at constant temperature,  $P_{\text{gen}}$  might relate to moving a randomly picked particle in a random direction over a small random distance, and  $\rho(s)$  is proportional to the Boltzmann weight  $e^{-\beta E(s)}$  with  $\beta = 1/k_B T$  the inverse temperature and  $E(s)$  the state's

energy. The Metropolis-Hastings algorithm takes a specific solution for the acceptance probability

$$P_{\text{acc}}(s^{(o)} \rightarrow s^{(n)}) = \min \left[ 1, \frac{\rho(s^{(n)})P_{\text{gen}}(s^{(n)} \rightarrow s^{(o)})}{\rho(s^{(o)})P_{\text{gen}}(s^{(o)} \rightarrow s^{(n)})} \right] \quad (1)$$

The generation probabilities will cancel in the above expression if they are symmetric,  $P_{\text{gen}}(s \rightarrow s') = P_{\text{gen}}(s' \rightarrow s)$  as in the less generic Metropolis scheme. At each MC step, the new state is either accepted or rejected based on the probability above. In case of a rejection, the old state is maintained and resampled. This scheme obeys detailed-balance and if, in addition, the set of MC moves are ergodic, equilibrium sampling is guaranteed. When ergodic sampling, even if mathematically obeyed, is slowed down by a rough (free) energy landscape, Replica exchange MC becomes useful.

Replica exchange MC (or replica exchange molecular dynamics) is based on the idea to simulate several copies of the system with different ensemble definitions [3–5], most commonly ensembles with increasing temperature (parallel tempering). By performing “swaps” between adjacent replicas, the low-temperature replicas gain access to the broader space region that are explored by the high-temperature replicas. The detailed-balance and corresponding acceptance-rejection step can be derived by viewing the set of states in the different ensembles (replicas) as a single high-dimensional superstate  $S = (s_1, s_2, \dots, s_N)$  representing the system in a set of  $N$  independent “parallel universes”. The Metropolis scheme applied to the superstate yields

$$P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)}) = \min \left[ 1, \frac{\rho(S^{(n)})}{\rho(S^{(o)})} \right] \quad (2)$$

\* titus.van.erp@ntnu.no

in which the probability of the superstate equals

$$\rho(S) = \rho(s_1, s_2, \dots, s_N) = \prod_{i=1}^N \rho_i(s_i) \quad (3)$$

where  $\rho_i(\cdot)$  is the specific probability density of ensemble  $i$ . For example, the move that attempts to swap the first two states, implying  $S^{(o)} = (s_1, s_2, \dots, s_N)$  and  $S^{(n)} = (s_2, s_1, \dots, s_N)$ , will be accepted with a probability

$$P_{\text{acc}} = \min \left[ 1, \frac{\rho_1(s_2)\rho_2(s_1)}{\rho_1(s_1)\rho_2(s_2)} \right] \quad (4)$$

In a replica exchange simulation, swapping moves and standard MC or MD steps are applied alternately. Parallel computing will typically distribute the same number of processing units per ensemble to carry out the computational intensive standard moves. The swapping move is cheap, but it requires that the ensembles involved in the swap have completed their previous move. If the standard moves in each ensemble require different computing times, then several processing units have to wait for the slow ones to finish. If the disbalance per move is relatively constant, the replicas could effectively be made to progress in cohort by trying to differentiate the number of processing units per ensemble or the relative frequency of doing replica exchange versus standard moves per ensemble. However, in several MC methods this disbalance is not constant, such as with configurational bias MC [6–8] or path sampling [9]. The number of elementary steps to grow a polymer in configurational bias MC obviously depends on the polymer’s length that is being grown, but also early rejections lead to a broad distribution of the time it takes to complete a single MC move even in uniform polymer systems. Analogously, the time required to complete a MC move in path sampling simulations will depend on the length of the path being created. Other examples of complex Monte Carlo methods with a fluctuating CPU cost per move are cluster Monte Carlo algorithms [10] and event-chain Monte Carlo [11, 12].

We will show that the standard acceptance Eqs. 1 and 4 can be applied in a parallel scheme in which ensembles are updated irregularly in time and the average frequency of MC moves is different for the ensembles. In addition, we show that we can apply an infinite swapping [13] scheme between the available ensembles. For this, we develop a new protocol based on the evaluation of permanents that circumvents the steep factorial scaling. This last development is also useful for standard replica exchange.

## METHODS

**Finite swapping.** In the following, we will assume that we have two types of MC moves. One move that is CPU intensive and can be carried out within a single ensemble, and replica exchange moves between ensembles which are relatively cheap to execute. The CPU intensive

move will be carried out by a single worker (one processor unit, one node or a group of nodes) and these workers perform their task in parallel on the different ensembles. One essential part of our algorithm is that we have less workers than ensembles such that whenever the worker is finished and produced a new state for one ensemble, this state can directly be swapped with the states of any of the available ensembles (the ones not occupied by a worker). After that, the worker will randomly switch to another unoccupied ensemble for performing a CPU intensive move.

In its most basic form, the algorithm consists of the following steps:

1. Define  $N$  ensembles and let  $\rho_i(\cdot)$  be the probability distribution of ensemble  $i$ . We also define  $P_{\text{RE}}$  which is the probability for a replica exchange move.
2. Assign  $K < N$  ‘workers’ (processing units) to  $K$  of the  $N$  ensembles for performing a CPU intensive MC move. Each ensemble is at all times occupied by either 1 or 0 workers. The following steps are identical for all the workers.
3. If the worker is finished with its MC move in ensemble  $i$ , the new state is accepted or rejected according to Eq. 1 (with  $\rho_i$  for  $\rho$ ). Ensemble  $i$  is updated with the new state (or by resampling the old state in case of rejection) and is then considered to be free.
4. Take a uniform random number  $\nu$  between 0 and 1. If  $\nu > P_{\text{RE}}$  go to step 7.
5. Among the available ensembles, pick a random pair  $(i, j)$ .
6. Try to swap the states of ensembles  $i$  and  $j$  using Eq. 4 (with labels  $i, j$  instead of 1, 2). Update ensembles  $i, j$  with the swapped state or the old state in case of a rejection. Return to step 4.
7. Select one of the free ensembles at random and assign the worker to that ensemble for performing a new standard move. Go to step 3.

In this algorithm ensembles are not updated in cohort like in standard replica exchange, but updates occur at irregular intervals. In addition, the different ensemble conditions can result in systematic differences in the number of states that are being created over time. To prove that the above scheme actually samples the correct distributions requires a fundamentally new conceptual view as the superstate picture is no longer applicable. Despite that the algorithm uses the same type of Eqs. 1 and 4, as one would use in standard replica exchange, it does not rely on Eqs. 2 and 3 that are no longer valid. In the Supplementary Information (SI) we provide a proof from the individual ensemble’s perspective in which the other ensembles provide an “environment”  $\mathcal{E}$  that might, or might not, participate in the



move of the ensemble considered. By doing so, we no longer require that the number of transitions from old to new,  $S^{(o)} \rightarrow S^{(n)}$ , is the same as from new to old,  $S^{(n)} \rightarrow S^{(o)}$ . Instead, by writing  $S = (s_1, \mathcal{E})$ , from ensemble 1's perspective, we have that the number of  $(s_1^{(o)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(n)}, \mathcal{E}^{(n)})$  transitions should be equal to the number of  $(s_1^{(n)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(o)}, \mathcal{E}^{(n)})$  transitions when the standard move is applied where  $\mathcal{E}^{(n)}$  refers to *any* new environment. The SI shows a similar detailed-balance condition for the replica exchange moves. At step 6 we sample only ensemble  $i$  and  $j$  or, alternatively, all free ensembles get a sample update. This would mean resampling the existing state of those not involved in a swap ("null move"). This makes the approach more similar to the superstate sampling albeit using only free ensembles, as described in the SI. The null move does not reduce the statistical uncertainty, but we mention it here as it makes it easier to explain the infinite swapping approach. But for the detailed-balance conditions to be valid it is imperative that occupied ensembles are not sampled.

An essential aspect of the efficiency of our algorithm is that the number of workers  $K$  is less than the number of ensembles  $N$ . The case  $K = N$  is valid but would reduce the number of replica exchange moves to zero as only one ensemble is free at the maximum. Reducing the  $K/N$  ratio will generally imply a higher acceptance in the replica exchange moves as we can expect a higher number of free ensembles whose distributions have significant overlap. What gives the optimum number of workers is therefore a non-trivial question that we will further explore in the Results and Discussion section. However, for case  $K < N$  we can maximize the effect of the replica exchange moves by taking the  $P_{RE}$  parameter as high as possible. In fact, we can simulate the effect of the limit  $P_{RE} \rightarrow 1$  without having to do an infinite number of replica exchange moves explicitly. This lead to an infinite swapping [13] version of our algorithm.

**Infinite swapping.** If in the previously described algorithm we take  $P_{RE} = 1 - \delta$ , we will loop through the steps 4-6 for many iterations ( $n_{it} = \sum_{n=0}^{\infty} n(1 - \delta)^n \delta = 1/\delta$  in the limit  $\delta \rightarrow 0$ ) before getting to step 7. When  $\delta$  vanishes and  $n_{it}$  becomes infinitely large, we expect that all possible swaps will be executed an infinite number of times. Since the swaps obey detailed balance between unoccupied ensembles, these will essentially sample the distribution of Eq. 3 (for the subset  $S^*$  of unoccupied ensembles). Hence, when the loop is exited, each possible permutation  $\sigma \in S^*$  has been sampled  $n_{it} \times \rho(\sigma) / \sum_{\sigma} \rho(\sigma)$  times. By lumping all the times that the same permutation was sampled and normalizing by division with  $n_{it}$ , we simply sample all the possible permutations in one go using fractional weights that sum up to 1. This is then the only sampling step, as the single update in step 3 can be skipped due to its negligible  $1/n_{it}$  weight.

The idea of doing an "infinite number" of swapping moves has been proposed before [13–15], but here we give a different flavor to this approach by a convenient reformulation of the problem into permanents that allows

us to beat the steep factorial scaling reported in earlier works [13]. The permanents formulation goes as follows. Supposed that after step 3, there are 4 free ensembles (we name them  $e_1, e_2, e_3, e_4$ ) containing 4 states ( $s_1, s_2, s_3, s_4$ ). Which state is in which ensemble after this step is irrelevant. We can now define a weight-matrix  $W$ :

$$W = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} & W_{44} \end{pmatrix} \end{matrix}$$

where  $W_{ij} \propto \rho_j(s_i)$ . Essential to our approach is the computation of the permanent of the  $W$  matrix,  $\text{perm}(W)$ , and that of the  $W\{ij\}$ -matrices in which the row  $i$  and column  $j$  are removed.

The permanent of a matrix is similar to the determinant, but without alternating signs. We can, henceforth, write  $\text{perm}(W) = \sum_{j=1}^4 W_{1j} \text{perm}(W\{1j\})$ . As the permanent of the  $1 \times 1$  matrix is obviously equal to the single matrix value, the permanent of arbitrary dimension could in principle be solved recursively using this relation. Based on the permanents of  $W$ , we will construct a probability matrix  $P$ :

$$P = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{pmatrix} \end{matrix}$$

where  $P_{ij}$  is the chance to find state  $s_i$  in ensemble  $e_j$ . As for each permutation each state is in one ensemble and each ensemble contains one state, the  $P$ -matrix is bistochastic: both the columns and the rows sum up to 1. If we consider  $S_{ij}^*$  the set of permutations in which state  $s_i$  is in  $e_j$ , we can write  $P_{ij} = \sum_{\sigma \in S_{ij}^*} \rho(\sigma) / \sum_{\sigma' \in S^*} \rho(\sigma')$ . We can, however, also use the permanent representation in which

$$P_{ij} = \frac{W_{ij} \text{perm}(W\{ij\})}{\text{perm}(W)} \quad (5)$$

So far we have not won anything as computing the the permanent via the recursive relation mentioned above has still the factorial scaling. The Gaussian elimination approach, that allows an order  $\mathcal{O}(n^3)$  computation for determinants of  $n \times n$  matrices, won't work for permanents as only some but not all row- and column-operations have the same effect to a permanent as to a determinant. One can for instance swap rows and columns without changing the permanent. Multiplying a row by a nonzero scalar multiplies the permanent by the same scalar. Hence, this will not affect the  $P$ -matrix based on Eq. 5. Unlike the determinant, adding or subtracting to a row a scalar multiple of another row, an essential part of the Gaussian

elimination method, does change the permanent. This makes the permanent computation of a large matrix excessively more expensive than the computation of a determinant. Yet, recent algorithms based on the Balasubramanian–Bax–Franklin–Glynn (BBFG) formula [16–18] scale as  $\mathcal{O}(2^n)$ . This means that the computation of the full  $P$ -matrix scales as  $\mathcal{O}(2^n \times n^2)$ , which seems still steep but is nevertheless a dramatic improvement compared to factorial scaling.

For our target time of 1 second, for instance, we could only run the algorithm up to  $N = 7$  in the factorial approach, while we reach  $N = 12$  in the BBFG method using a mid-to-high-end laptop (DELL XPS 15 with an Intel Core i7-8750H). If matrix size of  $N = 20$  is the target, the BBFG method can perform a full  $P$ -matrix determination in  $\sim 711$  seconds, while it would take  $\sim 15.3 \times 10^6$  years in the factorial approach. The BBFG method is the fastest completely general solution for the problem of computing a  $P$ -matrix from any  $W$ -matrix. For several algorithms, the  $W$ -matrix has special characteristics that can be exploited to further increase efficiency. For instance, if by shuffling the rows and columns the  $W$ -matrix can be made into a block form, where squared blocks at the diagonal have only zero's at their right and upper side, the permanent is equal to the product of the block's permanents. For instance, if  $W_{14} = W_{24} = W_{34} = 0$  we have two blocks,  $3 \times 3$  and  $1 \times 1$ . If  $W_{13} = W_{14} = W_{23} = W_{24} = 0$ , we can identify 2 blocks of  $2 \times 2$  etc. Identification of blocks can hugely decrease the computation of a large permanent. Another speed-up can be made if all rows in the  $W$ -matrix are a sequence of ones followed by all zeros, or can be made into that form after previously mentioned column and row operations. This makes an order  $\mathcal{O}(n^2)$  approach possible. We will further discuss this in Sec. Application:  $\infty$ RETIS.

The infinite swapping approach changes the aforementioned algorithm from step 3:

3. If the worker is finished with its MC move in a specific ensemble, the new state is accepted or rejected (but not yet sampled) according to Eq. 1. The ensemble is free.
4. Determine the  $W$ -matrix based on all unoccupied ensembles, calculate the  $P$ -matrix based on Eq. 5, and update all the unoccupied ensembles by sampling all free states with the fractional probabilities corresponding to the columns in the  $P$ -matrix.
5. Pick randomly one of the free ensembles  $e_j$ .
6. Pick one of the available states ( $s_1, s_2, \dots$ ) based on a weighted random selection in which state  $s_i$  has a probability of  $P_{ij}$  to be selected.
7. The worker is assigned to do a new standard move in ensemble  $e_j$  based on previous state  $s_i$ . Go to step 3.

## APPLICATION: $\infty$ RETIS

Replica Exchange Transition interface sampling (RETIS) [19, 20] is a quantitative path sampling algorithm in which the sampled states are short molecular trajectories (paths) with certain start- and end-conditions, and a minimal progress condition. New paths are being generated by a Monte Carlo move in path space, such as the shooting move [21] in which a randomly selected phase point of the previous path is randomly modified and then integrated backward and forward in time by means of molecular dynamics (MD). The required minimal progress increases with the rank of the ensemble such that the final ensemble contains a reasonable fraction of transition trajectories. The start- and end-conditions, as well as the minimal progress, are administered by the crossings of interfaces ( $\lambda_0, \lambda_1, \dots, \lambda_M$ ) with  $\lambda_{k+1} > \lambda_k$ , that can be viewed as non-intersecting hypersurfaces in phase space having a fixed value of the reaction coordinate. A MC move that generates a trial path not fulfilling the path ensemble's criteria is always rejected. RETIS defines different path ensembles based on the the direction of the paths and the interface that has to be crossed, but all paths start by crossing  $\lambda_0$  (near the reactant state/state  $A$ ) and they end by either crossing  $\lambda_0$  again or reaching the last interface  $\lambda_M$  (near the product state/state  $B$ ). There is one special path ensemble, called  $[0^-]$ , that explores the left side of  $\lambda_0$ , the reactant well, while all other path ensembles, called  $[k^+]$  with  $k = 0, 1, \dots, M-1$ , start by moving to the right from  $\lambda_0$  reaching at least  $\lambda_k$ .

A central concept in RETIS is the so-called overall crossing probability, the chance that a path that crosses  $\lambda_0$  in the positive direction reaches  $\lambda_M$  without recrossing  $\lambda_0$ . It provides the rate of the process when multiplied with the flux through  $\lambda_0$  (obtained from the path lengths in  $[0^-]$  and  $[0^+]$  [20]) and is usually an extremely small number. The chance that any of the sampled paths in the  $[0^+]$  path ensemble crosses  $\lambda_M$  is generally negligible, but a decent fraction of those ( $\sim 0.1 - 0.5$ ) will cross  $\lambda_1$  and some even  $\lambda_2$ . Likewise, paths in the  $[k^+]$ ,  $k > 0$ , path ensembles have a much higher chance to cross  $\lambda_{k+1}$  than a  $[0^+]$ -path as they already cross  $\lambda_k$ . This leads to the calculation of  $M$  local conditional crossing probabilities, the chance to cross  $\lambda_{k+1}$  given  $\lambda_k$  was crossed for  $k = 0, 1 \dots M-1$ , whose product gives an exact expression for the overall crossing probability with an exponentially reduced CPU cost compared to MD.

The efficiency is further hugely improved by executing replica exchange moves between the path ensembles. These swaps are essentially cost-free since there is no need to simulate additional ensembles that are not already required. An accepted swapping move in RETIS provides new paths in two ensembles without the expense of having to do MD steps. The enhancement in efficiency is generally even larger than one would expect based on those arguments alone as path ensembles higher-up the barrier provide a similar effect as the high-

temperature ensembles in parallel tempering. In addition, point-exchange moves between the  $[0^-]$  and  $[0^+]$  are performed by exchanging the end- and start-points of these path that are then continued by MD at the opposite site of the  $\lambda_0$  interface.

While TIS [22] (without replica exchange) can run all path ensembles embarrassingly parallel, the RETIS algorithm increases the CPU-time efficiency, but is difficult to parallelize and open source path-sampling codes, like OpenPathSampling [23] and PyRETIS [24], implement RETIS as a fully sequential algorithm. The path length distributions are generally broad with an increasing average path length as function of the ensemble's rank. This becomes increasingly problematic the more ensembles you have as they all have to wait for the slowest ensemble. This means that while RETIS will give you the best statistics per CPU-hour, it might not give you the best statistics in wall-time. With the continuous increase in computing power, trading some CPU-time efficiency for wall-time efficiency, getting the answer faster while spending more CPU-cycles, might be preferential. Our parallel scheme can effectively deal with the unequal CPU cost of the replicas, which allows us to increase the wall-time efficiency with no or minimal reduction in CPU-time efficiency.

**The  $W$ -matrix in RETIS.** If there are  $M + 1$  interfaces,  $\lambda_0, \lambda_1, \dots, \lambda_M$ , there are also  $N = M + 1$  ensembles,  $[0^-], [0^+], [1^+], \dots, [(M - 1)^+]$ . For  $K$  workers, the size of the  $W$ -matrix is, hence, either  $(N - K + 1) \times (N - K + 1)$  or  $(N - K) \times (N - K)$  as swappings are executed when 1 of the  $K$  workers is free, while the remaining  $K - 1$  workers occupy path ensembles that are locked and do not participate in the swap. The smallest matrix occurs when one worker is occupying both  $[0^-]$  and  $[0^+]$  during the point exchange move, as described in the simulation methods.

Paths can be represented by a sequence of time slices, the phase points visited by the MD trajectory. For a path of length  $L + 1$ ,  $X = (x_0, x_1, \dots, x_L)$ , the plain path probability density  $\rho(X)$  is given by the probability of the initial phase point times the dynamical transition probabilities to go from one phase point to the next:  $\rho(X) = \rho(x_0)\phi(x_0 \rightarrow x_1)\phi(x_1 \rightarrow x_2) \dots \phi(x_{L-1} \rightarrow x_L)$ . Here, the transition probabilities depend on the type of dynamics (deterministic, Langevin, Nosé-Hoover dynamics, etc). The weight of a path within a specific path ensemble  $\rho_j(X)$  can be expressed as the plain path density times the indicator function  $\mathbf{1}_{e_j}$  and possibly an additional weight function  $w_j(X)$ :  $\rho_j(X) = \rho(X) \times \mathbf{1}_{e_j}(X) \times w_j(X)$ . The indicator function equals 1 if the path  $X$  belongs to ensemble  $e_j$ . Otherwise it is 0. The additional weight function  $w_j(X)$  is part of the high-acceptance protocol that is used in combination with the more recent path generation MC moves such as stone skipping [25] and wire-fencing [26]. Using these "high-acceptance weights", nearly all the CPU intensive moves can be accepted as they are tuned to cancel the  $P_{\text{gen}}$ -terms in Metropolis-Hastings scheme, Eq. 1, and the effect of the non-physical

weights is undone in the analysis by weighting each sampled path with the inverse of  $w_j(X)$ .

While the path probability  $\rho(s_i = X)$  is difficult to compute, determining  $\mathbf{1}_j(s_i)$  and  $w_j(s_i)$  is trivial. It is therefore a fortunate coincidence that we can replace  $W_{ij} = \rho_j(s_i)$  with

$$W_{ij} = \mathbf{1}_{e_j}(s_i)w_j(s_i) \quad (6)$$

because the  $P$ -matrix does not change if we divide or multiply a row by the same number, as mentioned in Sec. Methods. Except for  $[0^-]$ , all path ensembles have the same start and end condition and only differ with respect to the interface crossing condition. A path that crosses interface  $\lambda_k$  automatically crosses all lower interfaces  $\lambda_{l < k}$ . Reversely, if the path does not cross  $\lambda_k$ , it won't cross any of the higher interfaces  $\lambda_{l > k}$ . This implies that if the columns of  $W_{ij}$  are ordered such that the 1st column ( $e_1$ ) is the first available ensemble from the sequence ( $[0^-], [0^+], [1^+], \dots, [(M - 1)^+]$ ), the 2nd column ( $e_2$ ) is the second available ensemble etc, most rows will end with a series of zeros.

Reordering the rows with respect to the number of trailing zeros, almost always ensures that the  $W$ -matrix can be brought into a block-form such that the permanent can be computed faster based on smaller matrices. In particular, if  $[0^-]$  is part of the free ensembles, it will always form a  $1 \times 1$  block as there is always one and no more than one available path that fits in this ensemble.

If high-acceptance is not applied, we have  $w_j(X) = 1$  and each row in the  $W$ -matrix (after separating the  $[0^-]$  ensemble if it is part of the free ensembles) is a sequence of ones followed by all zeros. The  $W$ -matrix can hence be represented by an array  $(n_1, n_2, n_3, \dots, n_n)$  where each integer  $n_i$  indicates the number of ones in row  $i$ . As we show in the SI, the permanent of such a  $W$ -matrix is simply the product of  $(n_i + 1 - i)$ :  $\text{perm}(W) = \prod_i (n_i + 1 - i)$ . Further, the  $P$ -matrix can be constructed from following order  $\mathcal{O}(n^2)$  method.

The first step is to order the rows of the  $W$ -matrix such that  $n_1 \leq n_2 \leq \dots \leq n_n$ . We then fill in the  $P$ -matrix from top to bottom for each row using

$$P_{ij} = \begin{cases} 0, & \text{if } W_{ij} = 0 \\ \frac{1}{n_i + 1 - i}, & \text{if } W_{ij} = 1 \text{ and } [W_{(i-1)j} = 0 \text{ or } i = 1] \\ \left( \frac{n_{i-1} + 1 - i}{n_i - i} \right) P_{(i-1)j}, & \text{otherwise} \end{cases} \quad (7)$$

The approach is extremely fast and allows the computation of  $P$ -matrices from a large  $W$ -matrix, up to several thousands, within a second of CPU-time. The above method applies whenever the rows of the  $W$ -matrix can be transformed into sequence of ones followed by all zeros. Besides RETIS without high-acceptance, this would apply to other MC methods like subset-sampling [27] or umbrella sampling [28] with semi-infinite rectangular windows.

## RESULTS AND DISCUSSION

To test our algorithms we ran 3 types of simulations. First a memoryless single variable stochastic (MSVS) process was simulated in order to mimic a RETIS simulation in which the average path length increases linearly with the rank of the ensemble. A "path" is created by drawing 2 random numbers where the first determines how much progress a path makes and the second determines the path length. These two outcomes are variable and depend on the rank of the ensemble such that the fictitious path in ensemble  $[k^+]$  has a 0.1 probability to cross  $\lambda_{k+1}$  and has an average path length of approximately  $k/10$  seconds (see Section Materials and Methods). The worker is paused for a number of seconds equal to the path length before it can participate in replica moves to mimic the time it would take to do all the necessary MD steps. While this artificial simulation allows us to investigate the potential strength of the method to tackle extremely rare events, it cannot reveal the effect of correlations between accepted paths when fast exploration of the reaction coordinate's orthogonal directions are crucial. To analyze this effect, we also ran a 2D membrane permeation system with two slightly asymmetric channels [29]. Lastly, to study our algorithm with a more generic  $W$ -matrix that needs to be solved via BBFG formula, we also ran a set of underdamped Langevin simulations of a particle in a double well potential [30] using the recent wire fencing algorithm with the high acceptance protocol [26]. All simulation results were performed using 5 independent runs of 12 hours. Errors were based on the standard deviations from these 5 simulations, except for the MSVS process, where a more reliable statistical error was desired for the comparison with analytical results. Here, block errors were determined on each of the five simulations based on the running average of the overall crossing probability. The block errors were finally combined to obtain the statistical error in the average of the five simulations.

### Memoryless single variable stochastic (MSVS) process

Table I reports the overall crossing probabilities and their statistical errors for a system with 50 interfaces and 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50 workers. All values are within a 50% deviation from the true value of  $10^{-50}$  with the more accurate estimates for the simulations having a large number of workers. Also, the true value is within one standard deviation of the reported averages for 70% of the data points, as is expected from the standard Gaussian confidence intervals. Figure 1a shows the scaling of the MD time (solid lines) and number of MC moves (dashed lines) of the MSVS simulations (orange) compared to linear scaling (black) and the expected scaling for standard replica exchange (REPEX) in which ensembles are updated in cohort (purple).

Whereas the number of "MD steps" and MC moves quickly levels off to a nearly flat plateau in the standard approach due to workers being idle as they need to wait for the slowest worker, the replica exchange approach developed in this article shows a perfect linear scaling with respect to the MD time. The number of MC moves in the new method shows an even better than linear scaling due to the fact that the ensembles with shorter "path lengths" get simulated relatively more often with more workers, resulting in more MC moves per second. This in itself does not necessarily mean that the simulations converge much faster because the additional computational effort may not be targeted to the sampling where it is needed. If we neglect the fact that path ensemble simulations are correlated via the replica exchange moves, we can write that the relative error in overall crossing probability  $\epsilon$  follows from the relative errors in each path ensemble  $\epsilon_i$  via:  $\epsilon^2 = \sum_i \epsilon_i^2$ . It is henceforth clear that additional computational power should not aim to lower the error in a few path ensembles that were already low compared to other path ensembles. We therefore measure the effectiveness of the additional workers by calculating computational efficiencies. The efficiency of a specific computational method is here defined as the inverse computer time, CPU- or wall-time, to obtain an overall relative error equal to 1:  $\epsilon = 1$ .

In figure 1d the efficiencies based on wall-time (solid) and CPU-time (dashed) are plotted for the MSVS process. These plots depends on the ability of computing reliable statistical errors in the overall crossing probability that is an extremely small number,  $10^{-50}$ . The somewhat fluctuating behavior of these curves should hence be viewed as statistical noise as the confidence interval of these efficiencies depends on the statistical error of this error. Despite that, clear trends can be observed in which the CPU-time efficiency is more or less flat, while the wall-time efficiency shows an upward trend. If we neglect the effect of replica exchange moves on the efficiency, we can relate these numerical results with theoretical ones [20, 31] for any possible division of a fixed total CPU-time over the different ensembles. A common sense approach would be to aim for the same error  $\epsilon_i$  in each ensemble (which implies doing the same number of MC moves per ensemble) or to divide the total CPU-time evenly over the ensembles. These two strategies correspond to the case  $K = 1$  or standard RETIS and  $K = N$  or standard TIS, respectively. Ref. [31] showed that these two strategies provide the same efficiency and in the SI we derive that this leads to a wall-time efficiency as function of the number of workers ( $K$ ) equal to  $K/56250$  which is the continuous purple line in figure 1d. The optimum division, however, would give a slightly better wall-time efficiency equal to  $K/50000$  which is the continuous black line in this figure. Also shown in figure 1d are the expected theoretical efficiencies based on the numerical distribution of MC moves in each ensemble. This hybrid numerical/theoretical result is shown by the small purple dots. This shows that  $\infty$ RETIS, at

TABLE I. Results of the 3 model systems showing crossing probabilities ( $P_{\text{cross}}$ ), permeabilities (perm.), and rates for different number of workers ( $\#w$ ). All results are shown in dimensionless units. Errors are based on single standard deviations. Values shown in the lower part are a: exact result, b: Ref. [29], c: approximated value based on Kramers' theory (see SI), d: Ref. [30], and e: Ref. [26].

MSVS		two-channel system		double well with wire fencing			
$\#w$	$P_{\text{cross}}/10^{-50}$	$\#w$	$P_{\text{cross}}/10^{-5}$	perm./ $10^{-6}$	$\#w$	$P_{\text{cross}}/10^{-7}$	rate/ $10^{-7}$
1	$0.61 \pm 0.33$	1	$1.52 \pm 0.17$	$1.28 \pm 0.14$	1	$5.91 \pm 0.18$	$2.59 \pm 0.07$
5	$1.47 \pm 1.04$	2	$1.63 \pm 0.24$	$1.37 \pm 0.20$	2	$5.70 \pm 0.13$	$2.51 \pm 0.06$
10	$0.86 \pm 0.51$	3	$1.52 \pm 0.07$	$1.28 \pm 0.06$	3	$5.57 \pm 0.19$	$2.45 \pm 0.08$
15	$0.68 \pm 0.08$	4	$1.42 \pm 0.10$	$1.19 \pm 0.08$	4	$5.20 \pm 0.30$	$2.34 \pm 0.12$
20	$1.02 \pm 0.13$	5	$1.40 \pm 0.12$	$1.18 \pm 0.10$	5	$5.05 \pm 0.41$	$2.23 \pm 0.18$
25	$1.02 \pm 0.17$	6	$1.54 \pm 0.06$	$1.30 \pm 0.05$	6	$5.49 \pm 0.29$	$2.42 \pm 0.13$
30	$1.26 \pm 0.24$	7	$1.48 \pm 0.08$	$1.24 \pm 0.07$	7	$4.99 \pm 0.39$	$2.21 \pm 0.17$
35	$1.05 \pm 0.15$	8	$1.46 \pm 0.08$	$1.23 \pm 0.06$	8	$4.88 \pm 0.43$	$2.15 \pm 0.19$
40	$1.05 \pm 0.14$	9	$1.42 \pm 0.10$	$1.20 \pm 0.08$			
45	$0.93 \pm 0.09$	10	$1.44 \pm 0.08$	$1.21 \pm 0.07$			
50	$1.00 \pm 0.07$	11	$1.41 \pm 0.09$	$1.19 \pm 0.08$			
		12	$1.30 \pm 0.15$	$1.09 \pm 0.12$			
literature/theoretical result							
			$1.23 \pm 0.16^b$	$1.06 \pm 0.14^b$			$2.79 \pm 0.70^d$
						$5.84 \pm 0.13^c$	$2.58 \pm 0.06^e$
1.00 <sup>a</sup>			$1.61^c$	$1.37^c$		$5.83^c$	$2.58^c$

least for a system in which the path length grows linearly with the ensemble's rank, naturally provides a division of the computational resources that is even better than TIS ( $K = N$ ) or RETIS ( $K = 1$ ). Yet, due to statistical inaccuracies this is only evident for the  $K = 15$  case. The best wall-time efficiency is obtained for the case  $K = N$ , which is essentially equivalent of running independent TIS simulations (i.e. without doing any replica exchange moves). We do not expect this to apply to more complex systems where the replica exchange move is a proven weapon for efficient sampling.

### Two-channel simulations

In the middle column of table I we report the calculated crossing probabilities and permeabilities for 5 simulations for every number of workers. All simulations are somewhat higher, though still in good agreement with the previous simulation from Ref. [29]. We also evaluated the approximate result based on Kramers' theory (see SI) which seem to confirm the results obtained in this paper.

Figure 1b shows the scaling of the MD time (solid lines) and number of MC moves (dashed lines) of the two-channel simulations (blue) compared to linear scaling (black). We see a slightly worse than linear scaling of the MD time, which might just be due to a small positive fluctuation of the 1 worker data-point. We also see a similar more than linear scaling in the number of MC moves as with the MSVS simulations, for the same reason. In figure 1e the efficiencies based on wall-time (solid) and CPU-time (dashed) are plotted for the two-channel system. The CPU-time efficiency is more or less flat until 8 workers after which it starts to drop off. The wall-time efficiency shows an upward trend until 10 workers after

which it starts to drop off as well. We assign this drop to the reduction of replica exchange moves which is an essential aspect for sampling this system efficiently [29]. This is tangible from figure S1 in the SI where we plot fraction of trajectories, passing through  $\lambda_{M-1}$ , that are in the lower barrier channel. While from the average fraction it still looks like the simulations sampled both channels for any number of workers, 4 out of the 5 simulations in the  $K = N = 12$  case solely visited one of the two channels. This is in agreement with previous TIS results [29]. The  $K = 11$  case already provides a dramatic improvement, but is still expected to be sub optimal due to the relatively low frequency of replica exchange moves compared to  $K < 11$ . As reported in ref. [29], this ratio requires many MC moves to converge to the theoretical value of 0.71 without the added MC moves introduced in that paper. We did not simulate with these added moves and thus see the same slow convergence for all of our simulations. From this 2D system it would indicate that having half the number of ensembles as workers is a safe bet for optimum efficiency.

### Double well 1D barrier using wire fencing

In the right column of table I we report the calculated crossing probabilities and rates for the underdamped Langevin particle in the 1D double well potential. All simulations are in reasonable agreement with each other and the results of Refs. [26] and [30], as well as the approximate value based on Kramers' theory. However, while these results confirm the soundness of the method, the scaling and efficiency are less convincing. Figure 1b shows a significantly worse than linear scaling. On further inspection we found the average time per MC move was significantly smaller than our infinite-swapping goal

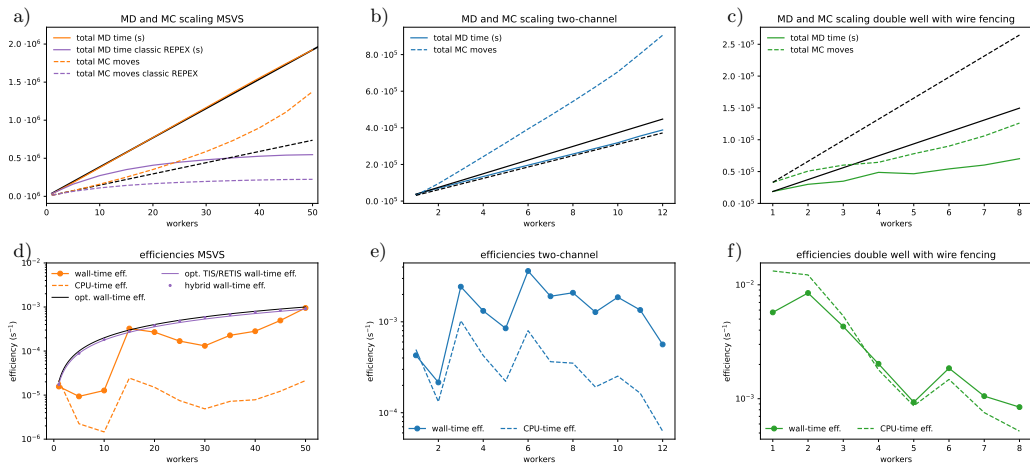


FIG. 1. The average scaling of total MD time (cumulative time spend by all the workers) (solid) and MC moves (dashed) (a-c) and wall-time (solid) and the CPU-time (dashed) efficiencies (d-f) for each number of workers. This is shown for the memoryless single variable stochastic (MSVS) process (a, d, orange), the two-channel system (b, e, blue) and the double well with wire fencing (c, f, green) simulations. Each of the data points is based on 5 independent simulations. For the scaling plots, the black lines are guides for linear scaling from the 1 worker data-point. The purple lines in the scaling plot for the MSVS simulations (a) show what the scaling would be if we had to wait for the slowest ensemble to finish for each MC move. The black line, purple line, and points in the efficiency plot of the MSVS process (d) show the optimal, optimal TIS/RETIS, and hybrid wall-time efficiency, respectively, as computed in the SI.

of 1 s when the simulation was run with more than 2 workers. This results in a bottleneck on how many MC moves can be started per second, which is the reason for the observed bad scaling. It still is slightly positive instead of flat as the infinite swapping procedure becomes quicker with more workers due to the smaller  $W$ -matrix. The same bottleneck can be seen in figure 1f where both efficiencies plummet with more than 2 workers. The reported scaling deficiency is of little significance for actual molecular systems where the creation of a full path takes minutes to hours rather than subseconds.

## CONCLUSIONS

We developed a new generic replica exchange method that is able to effectively deal with MC moves with varying CPU costs, for instance due to the algorithmic complexity of the MC moves. An essential aspect of the method is that the number of workers, who execute the ensemble’s specific MC moves in parallel, is less than the number of ensembles. Once a worker is finished with its move, replica exchange moves are carried out solely between those ensembles that are not occupied by a worker. This implies that the ensembles are updated at irregular intervals and a different number of MC moves will be executed for each ensemble. As a result, the conceptual viewpoint in which the set of replica’s are viewed as a single superstate is no longer valid and the existence of some

kind of detailed-balance relation is no longer trivial. To prove the exactness of our approach, we introduced some new conceptual views on the replica exchange methodology that is different from the common superstate principle. Instead, we show that the distributions in the new approach are conserved for each ensemble individually via a twisted detailed balance relation in which the other ensembles constitute an environment that is potentially actively involved in the MC move of the ensemble considered. In addition, the method can be combined with an infinite swapping approach without the factorial scaling based on a mathematical reformulation using permanents.

We applied the novel replica exchange technique on a path sampling algorithm, RETIS, which is a prototype of algorithm where the costs for a Monte Carlo move can vary enormously. The resulting new path sampling algorithm, coined  $\infty$ RETIS, was thereafter tested on three model systems. The results of these simulations show that the number of MD steps increase linearly with the number of workers invoked as long as the ensemble’s MC move has a lower computational cost than the replica exchange move carried out by the scheduler. The number of executed MC moves shows an even better than linear scaling. Moreover, the efficiency increases linearly with the number of workers for a low-dimensional system in which the replica exchange has little effect, while it has an optimum in more complex systems as the number of successful replica exchange moves decreases when the

number of workers is close to the number of ensembles.

In summary, the replica exchange method discussed in this paper has a clear potential to accelerate present path sampling simulations, but can also be combined with many other complex algorithms including those that are yet to be invented. With continuing trend to run progressively more massively-parallel computing jobs, our algorithm is likely to gain importance and will open up new avenues in the field of molecular simulations and beyond.

### Supporting Information Appendix (SI)

This article contains Supplementary Information

## MATERIALS AND METHODS

### Simulation methods

The implementation of  $\infty$ RETIS was structured as follows. We start  $1 \leq K \leq N$  worker- and 1 scheduler-process. Each of the worker-processes is going to process ensemble specific MC moves while the scheduler-process will do all the replica exchange moves and submits new jobs to the workers. All ensembles/trajectories that are currently being updated by a worker-process are not considered for MC moves by the scheduler, essentially being 'locked'. This means that no data is written for those ensembles and they are not valid targets for swapping moves. After a worker is done, it submits the result to the scheduler, the scheduler then unlocks the returned ensemble/trajectory and executes the replica exchange moves on all ensembles/trajectories that are not locked. It then submits a new job to the freed worker for performing a new MC move in a randomly chosen free ensemble (or two ensembles in case of a point exchange move) and locks the involved ensembles/trajectories.

In the  $\infty$ RETIS method there are two kind of ensemble moves that involve MD steps. The first one is the shooting move (either standard shooting [21] or the more recent sub-trajectory moves [25, 26]) in which a new path is being generated from an old path within a single ensemble. The second one is the point exchange move between  $[0^-]$  and  $[0^+]$ . If a worker is assigned to this task, it means that both  $[0^-]$  and  $[0^+]$  are occupied by this worker. The scheduler ensures that there is never more than 1 worker considered free at a given time. When the free worker is assigned to perform a new MC move, each of the ensembles have an equal probability to be selected. If  $[0^+]$  or  $[0^-]$  is selected and the other is also free, there is a 50% chance to perform a  $[0^-] \leftrightarrow [0^+]$  point exchange move instead of a shooting move in the selected ensemble.

### Memoryless single variable stochastic (MSVS) process

No actual MD is run for the MSVS simulations. Instead, we directly sample two random numbers,  $r_1$  and  $r_2$  from an uniform distribution  $\in [0, 1)$  to set the path's progress and the path length. A path in ensemble  $[k^+]$  is assumed to cross interface  $\lambda_{k+l}$  if  $r_1 < (0.1)^l$ . After this, we wait a random time,  $t = 0.2r_2k + 0.1$  in seconds. This was done to simulate both the increasing average simulation time and variance for outer ensembles. This setup means that we have no history dependence and allows us to compute the theoretical values show in figure 1. 5 independent  $\infty$ RETIS simulations were run with 1, 5, 10, 15,  $\dots$ , 45, 50 workers.

### Double channel simulations

In order to investigate the effect of our algorithm on the ergodicity of the sampling, a 2D two-channel simulation was run as described in reference [29]. The new RETIS moves introduced in that paper (mirror-move and target-swap move) were not used. Instead, MD was only run to do shooting moves or the  $[0^-] \leftrightarrow [0^+]$  point exchanges. As the MD for this system completed too fast, every worker was set to wait 9 times the time it took to run the MD before returning the result. 5 independent  $\infty$ RETIS simulations were run with 1, 2,  $\dots$ , 11, 12 workers.

### 1D double well with wire fencing

In order to investigate the accuracy with a  $W$  matrix that contains more numbers than 0s or 1s we simulated a 1D double-well system [30] together with the high-acceptance version of a novel path-sampling algorithm, wire fencing. The algorithm is described in reference [26], but for us the relevant part is that the high-acceptance weight is the number of frames that a path has outside the interface for each ensemble times an extra factor 2 if the path ends at the last interface. As for the two-channel system, a worker was set to wait 9 times the time it took to complete the MD move before returning the result. 5 independent  $\infty$ RETIS simulations were run with 1, 2,  $\dots$ , 7, 8 workers with interfaces placed at  $[-0.99, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, 1.0]$ .

## ACKNOWLEDGMENTS

We acknowledge funding from the Research Council of Norway through FRINATEK Project No. 275506.

*The authors declare no conflict of interest*

- 
- [1] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087 (1953).
- [2] W. Hastings, Monte-Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**, 97 (1970).
- [3] R. H. Swendsen and J. S. Wang, Replica monte-carlo simulation of spin-glasses, *Phys. Rev. Lett.* **57**, 2607 (1986).
- [4] E. Marinari and G. Parisi, Simulated tempering - a new monte-carlo scheme, *Europhysics Lett.* **19**, 451 (1992).
- [5] Y. Sugita and Y. Okamoto, Replica-exchange molecular dynamics method for protein folding, *Chem. Phys. Lett.* **314**, 141 (1999).
- [6] J. I. Siepmann and D. Frenkel, Configurational bias Monte-Carlo - a new sampling scheme for flexible chains, *Molecular Physics* **75**, 59 (1992).
- [7] T. Vlucht, R. Krishna, and B. Smit, Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite, *J. Phys. Chem. B* **103**, 1102 (1999).
- [8] D. Frenkel and B. Smit, *Understanding molecular simulations from algorithms to applications* (Academic press, San Diego, California, U.S.A., 2002).
- [9] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, Transition path sampling and the calculation of rate constants, *J. Chem. Phys.* **108**, 1964 (1998).
- [10] R. H. Swendsen and J.-S. Wang, Nonuniversal critical dynamics in monte carlo simulations, *Phys. Rev. Lett.* **58**, 86 (1987).
- [11] E. A. J. F. Peters and G. de With, Rejection-free monte carlo sampling for general potentials, *Phys. Rev. E* **85**, 026703 (2012).
- [12] M. Michel, S. C. Kapfer, and W. Krauth, Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps, *J. Chem. Phys.* **140**, 054116 (2014).
- [13] N. Plattner, J. D. Doll, P. Dupuis, H. Wang, Y. Liu, and J. E. Gubernatis, An infinite swapping approach to the rare-event sampling problem, *The Journal of Chemical Physics* **135**, 134111 (2011), <https://doi.org/10.1063/1.3643325>.
- [14] N. Plattner, J. D. Doll, and M. Meuwly, Overcoming the rare event sampling problem in biological systems with infinite swapping, *J. Chem. Theory Comput.* **9**, 4215 (2013).
- [15] J. Lu and E. Vanden-Eijnden, Methodological and computational aspects of parallel tempering methods in the infinite swapping limit, *J Stat Phys* **174**, 715 (2019).
- [16] K. Balasubramanian, *Combinatorics and diagonals of matrices*, Ph.D. thesis, Loyola College, Madras, India (1980).
- [17] E. Bax, *Finite-difference Algorithms for Counting Problems*, Ph.D. thesis, California Institute of Technology, Pasadena, United States of America (1998).
- [18] D. G. Glynn, The permanent of a square matrix, *European Journal of Combinatorics* **31**, 1887 (2010).
- [19] T. van Erp, Reaction rate calculation by parallel path swapping, *Phys. Rev. Lett.* **98**, 268301 (2007).
- [20] R. Cabriolu, K. M. S. Refsnes, P. G. Bolhuis, and T. S. van Erp, Foundations and latest advances in replica exchange transition interface sampling, *J. Chem. Phys.* **147**, 152722 (2017).
- [21] C. Dellago, P. G. Bolhuis, and D. Chandler, Efficient transition path sampling: Application to lennard-jones cluster rearrangements, *The Journal of Chemical Physics* **108**, 9236 (1998), <https://doi.org/10.1063/1.476378>.
- [22] T. S. van Erp, D. Moroni, and P. G. Bolhuis, A novel path sampling method for the sampling of rate constants, *J. Chem. Phys.* **118**, 7762 (2003).
- [23] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, Openpathsampling: A python framework for path sampling simulations. 2. building and customizing path ensembles and sample schemes, *J. Chem. Theory Comput.* **15**, 837 (2019).
- [24] E. Riccardi, A. Lervik, S. Roet, O. Aaroen, and T. S. van Erp, Pyretis 2: An improbability drive for rare events, *J. Comput. Chem.* **41**, 370 (2020).
- [25] E. Riccardi, O. Dahlen, and T. S. van Erp, Fast decorrelating monte carlo moves for efficient path sampling, *J. Phys. Chem. Lett.* **8**, 4456 (2017).
- [26] D. T. Zhang, E. Riccardi, and T. S. van Erp, Path sampling with sub-trajectory moves, In preparation **xx**, **xx** (2021).
- [27] S.-K. Au and J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, *Probabilistic Eng. Mech.* **16**, 263 (2001).
- [28] G. Torrie and J. Valleau, Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling, *J. Comp. Phys.* **23**, 187 (1977).
- [29] A. Ghysels, S. Roet, S. Davoudi, and T. S. van Erp, Exact non-markovian permeability from rare event simulations, *Phys. Rev. Research* **3**, 033068 (2021).
- [30] T. van Erp, Dynamical rare event simulation techniques for equilibrium and nonequilibrium systems, *Adv. Chem. Phys.* **151**, 27 (2012).
- [31] T. S. van Erp, Efficiency analysis of reaction rate calculation methods using analytical models I: The two-dimensional sharp barrier, *J. Chem. Phys.* **125**, 174106 (2006).



# Supporting Information for: Exchanging replicas with unequal cost, infinitely and permanently

Sander Roet, Daniel T. Zhang, and Titus S. van Erp\*  
*Department of Chemistry, Norwegian University of Science and Technology (NTNU),  
N-7491 Trondheim, Norway*

## I. SUPPORTING INFORMATION TEXT

This Supplementary Information contains the following data, derivations, and numerical examples. In Sec. II, we provide a proof that the replica exchange method with cost unbalanced replicas conserves the equilibrium distribution at the individual ensemble level. Instead of the superstate principle, the derivation is based on the individual ensemble's perspective where the other ensembles serve as an environment, which finally leads to a twisted detailed-balance relation. In Sec. III, we show a  $\mathcal{O}(n^2)$  algorithm for computing the  $P$ -matrix from a  $W$ -matrix for the case that the  $W$ -matrix consists of rows having a series with ones, followed by zeros. This is the type of matrix that is relevant for RETIS simulations based on the standard shooting move. Sec. IV presents the derivations of the theoretical results on the crossing probabilities, rate constant, and permeability via Kramer's theory that are shown in table 1 of the main article. In Sec. V the computational efficiencies, including the derivations for the most optimal efficiencies, are discussed. Finally, in Sec. VI we provide some additional simulation results on the relative transition probabilities through the lower and higher barrier channel.

## II. DETAILED-BALANCE RELATIONS

In this section, we will derive detailed-balance relations for parallel replica's that are not based on the common superstate viewpoint. These alternative relations can be used to validate the replica exchange algorithm for replica's with unequal CPU cost. Our derivation is based on the finite swapping approach, though the infinite swapping version follows automatically from this when the probability to perform a swap goes to unity ( $P_{\text{RE}} \rightarrow 1$ ) as explained in the main text. To simplify matters, we assume that we have one type of replica exchange move that is low in CPU cost and one type of ensemble move that operates within one ensemble and has a high CPU cost. The relations that we derive are, however, by no means limited to that. In fact, in the RETIS algorithm there is also a point exchange move between the  $[0^-]$  and  $[0^+]$  ensemble. In previous publications this move, annotated as  $[0^-] \leftrightarrow [0^+]$ , was categorized as a special type of swapping/replica exchange move. In this article we reserve the name swap or replica exchange to an operation that involves the swapping of full paths, which does not require any MD steps. In contrast, the  $[0^-] \leftrightarrow [0^+]$  point exchange implies the exchange of time slices at the end and start of the paths that are then extended at the other side of the  $\lambda_0$  interface. In our implementation, this  $[0^-] \leftrightarrow [0^+]$  move is carried out by a single worker that locks both the  $[0^-]$  and  $[0^+]$  ensembles during this move. As the  $[0^-]$  paths can never be swapped with any of the other paths, we can view the point exchange move as an ensemble move in ensemble  $[0^+]$ .

As explained in the main article, the replica exchange algorithm that we propose is based on a set of workers and a set of ensembles. The number of workers  $K$  is less than the number of ensembles  $N$ . Most of the time the worker is performing a CPU intensive single-ensemble move. The ensemble in which the worker operates is considered occupied/locked. Once a worker has completed a CPU intensive move, the move will be accepted or rejected, after which either a replica exchange move will be carried out with any of the unoccupied ensembles or the worker will be assigned to do a new single-ensemble move at a randomly picked free ensemble.

In order to indicate the difference between occupied and unoccupied ensembles, we introduce a new state vector that indicates both the available ensembles as in the main text and the occupied ensembles with a bar, e. g.  $S = (s_1, s_2, \bar{s}_3, s_4, \bar{s}_5)$  to show that there are 5 ensembles of which ensemble 3 and 5 are occupied by a worker. For both occupied and unoccupied ensembles, the  $s_i$ -terms reflect the most recent state that was sampled in the  $i$ th ensemble. Now our sole aim is to ensure that if we just count the instances that an ensemble  $i$  is updated with a new sample (which could be a copy of the previous sample in case of a rejected move), these should be distributed according to the correct probability density  $\rho_i$ .

---

\* titus.van.erp@ntnu.no

It is important to note that the time between two updates can vary and depends on the state that was most recently sampled. However, the waiting time between an update of a specific ensemble and the point in time that this ensemble gets occupied by a worker will depend on the states of all other ensembles, but *not* on the state in the ensemble considered. Since the ensembles are independent, this waiting time will be the same on average irrespective to this sampled state. This has as a consequence that if we take "photographs" of the state vector, at intervals or randomly, evenly distributed over time, we should again obtain the correct distributions  $\rho_i$ , for all  $i$ , of the states in ensemble  $i$  as long as we ignore the instances that this ensemble is occupied. In other words, we can write for the previous example state vector

$$\rho(S) = \rho(s_1, s_2, \overline{s_3}, s_4, \overline{s_5}) = \rho_1(s_1)\rho_2(s_2)\rho_3^u(\overline{s_3})\rho_4(s_4)\rho_5^u(\overline{s_5}) \quad (1)$$

where  $\rho_i(\cdot)$  is the statistically correct distribution of ensemble  $i$ , and  $\rho_j^u(\cdot)$  an unknown distribution for occupied ensemble  $j$  that has no clear physical interpretation. For instance, it can happen that a state  $s$  is relatively unlikely to exist in ensemble  $i$ , low  $\rho_i(s)$ , but that any MC move starting from that state takes a very long time, resulting in a high  $\rho_i^u(s)$ .

Now, let's consider the Markov chain from the perspective of ensemble 1 where we monitor its state at the point that a new MC is initiated from an old state  $s_1$ . From the viewpoint of ensemble 1, the other ensembles are viewed as an "environment" ( $\mathcal{E} = (s_2, \overline{s_3}, s_4, \overline{s_5})$  in the aforementioned example), that might or might not influence the MC move. The probability of state  $s_1$  in ensemble 1 can be written as an integral of the conditional probability given an environment:

$$\rho_1(s_1) = \int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})d\mathcal{E}. \quad (2)$$

As the ensembles are independent we can write

$$\rho_1(s_1|\mathcal{E}) = \rho_1(s_1), \quad (3)$$

but we temporarily keep the condition to clarify the logical structure of the upcoming derivation.

As stated, we assume that we employ two types of moves: 1) a CPU intensive move that modifies  $s_1$  without using the environment  $\mathcal{E}$  and 2) a swapping move. In addition, the environment might influence the relative selection probabilities for choosing either 1) or 2). Typically, this selection probability will depend on  $N_a(\mathcal{E})$ , the number of unoccupied ensembles in  $\mathcal{E}$ . Further, we need to keep in mind that during the execution of the MC move in ensemble 1, the environment changes. How much the environment changes will depend on how long it takes to fully execute the move involving ensemble 1.

To derive detailed-balance relations for the replica exchange method for cost unbalanced ensembles, we start with the more general balance concept; if we have an infinite number of states distributed according to the equilibrium distribution, all of which make a MC move at the same time, then we have to get the equilibrium distribution again. This means that the flux out off  $s_1$  should be equal to the flux into  $s_1$  which can be written as

$$\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})\pi(s_1, \mathcal{E} \rightarrow s'_1, \mathcal{E}')d\mathcal{E}d\mathcal{E}'ds'_1 = \int \rho_1(s'_1|\mathcal{E}'')\rho(\mathcal{E}'')\pi(s'_1, \mathcal{E}'' \rightarrow s_1, \mathcal{E}''')d\mathcal{E}''d\mathcal{E}'''ds'_1 \quad (4)$$

The transition probability  $\pi(\cdot)$  can be split into the transitions via the different types moves (that we will indicate with the Greek letter  $\alpha$ ) which will be selected with a probability  $P_\alpha^{\text{sel}}(\mathcal{E})$  that can depend on the environment  $\mathcal{E}$ :

$$\pi(s_1, \mathcal{E} \rightarrow s'_1, \mathcal{E}') = \sum_\alpha P_\alpha^{\text{sel}}(\mathcal{E})\pi_\alpha(s_1, \mathcal{E} \rightarrow s'_1, \mathcal{E}') \quad (5)$$

This shows another complicating factor as in standard detailed-balance we need to consider the probability that the exact reverse move will be executed once the new state has been established. However, as the environment could have changed, the reverse move might involve different selection probabilities.

By substituting Eq. 5 into Eq. 4, we get an extra summation over  $\alpha$  in addition to the integrals:

$$\sum_\alpha \int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_\alpha^{\text{sel}}(\mathcal{E})\pi_\alpha(s_1, \mathcal{E} \rightarrow s'_1, \mathcal{E}')d\mathcal{E}d\mathcal{E}'ds'_1 = \sum_\alpha \int \rho_1(s'_1|\mathcal{E}'')\rho(\mathcal{E}'')P_\alpha^{\text{sel}}(\mathcal{E}'')\pi_\alpha(s'_1, \mathcal{E}'' \rightarrow s_1, \mathcal{E}''')d\mathcal{E}''d\mathcal{E}'''ds'_1 \quad (6)$$

But at this point, we apply the first level of "detailedness" by requiring the equation to hold for *each*  $\alpha$ :

$$\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_\alpha^{\text{sel}}(\mathcal{E})\pi_\alpha(s_1, \mathcal{E} \rightarrow s'_1, \mathcal{E}')d\mathcal{E}d\mathcal{E}'ds'_1 = \int \rho_1(s'_1|\mathcal{E}'')\rho(\mathcal{E}'')P_\alpha^{\text{sel}}(\mathcal{E}'')\pi_\alpha(s'_1, \mathcal{E}'' \rightarrow s_1, \mathcal{E}''')d\mathcal{E}''d\mathcal{E}'''ds'_1 \quad (7)$$

So now we can evaluate the different moves separately. We further simplify this expression by integration out the variables  $\mathcal{E}'$  and  $\mathcal{E}'''$  using the following relation:

$$\int \pi_\alpha(s, \mathcal{E} \rightarrow s', \mathcal{E}') d\mathcal{E}' = \pi_\alpha(s, \mathcal{E} \rightarrow s', {}^a\mathcal{E}) \quad (8)$$

where  ${}^a\mathcal{E}$  refers to *any* possible environment. Substitution of Eq. 8 in Eq. 7 gives:

$$\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_\alpha^{\text{sel}}(\mathcal{E})\pi_\alpha(s_1, \mathcal{E} \rightarrow s'_1, {}^a\mathcal{E})d\mathcal{E}ds'_1 = \int \rho_1(s'_1|\mathcal{E}'')\rho(\mathcal{E}'')P_\alpha^{\text{sel}}(\mathcal{E}'')\pi_\alpha(s'_1, \mathcal{E}'' \rightarrow s_1, {}^a\mathcal{E})d\mathcal{E}''ds'_1 \quad (9)$$

First, we consider  $\alpha = 1$  referring the CPU intensive move that only operates in ensemble 1. For this move we substitute  $\alpha = 1$  in Eq. 9 and replace  $\mathcal{E}''$  and  $s'_1$  with respectively  $\mathcal{E}$  and  $s'_1$ , which is allowed since these are dummy integration variables

$$\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_1^{\text{sel}}(\mathcal{E})\pi_1(s_1, \mathcal{E} \rightarrow s'_1, {}^a\mathcal{E})d\mathcal{E}ds'_1 = \int \rho_1(s'_1|\mathcal{E})\rho(\mathcal{E})P_1^{\text{sel}}(\mathcal{E})\pi_1(s'_1, \mathcal{E} \rightarrow s_1, {}^a\mathcal{E})d\mathcal{E}ds'_1$$

Then, we fix another level of detailedness by requiring that the integrands at the left and right side of equality sign to be identical for any  $\mathcal{E}$  and  $s'_1$ . As a result,  $\rho(\mathcal{E})P_\alpha^{\text{sel}}(\mathcal{E})$  will cancel out such that we can write

$$\rho(s_1|\mathcal{E})\pi_1(s_1, \mathcal{E} \rightarrow s'_1, {}^a\mathcal{E}) = \rho(s'_1|\mathcal{E})\pi_1(s'_1, \mathcal{E} \rightarrow s_1, {}^a\mathcal{E}) \quad (10)$$

Since in move 1) the ensembles progress independently from each other, we have

$$\pi_1(s_1, \mathcal{E} \rightarrow s'_1, {}^a\mathcal{E}) = \pi_1(s_1 \rightarrow s'_1)\pi_1(\mathcal{E} \rightarrow {}^a\mathcal{E}) \quad (11)$$

The subscript "1" in  $\pi_1(\mathcal{E} \rightarrow {}^a\mathcal{E})$  might seem contradictory to the previous statement on independent progression, but it just indicates that the points in time at which the environment is evaluated relates the duration of the MC move in ensemble 1:  $\mathcal{E}$  is the environment at the start of the MC move in ensemble 1, and  ${}^a\mathcal{E}$  is that when the move is completed. As the time for a  $s_1 \rightarrow s'_1$  move is likely not the same as the time for a  $s'_1 \rightarrow s_1$  move, the final environments are likely not the same. However,  ${}^a\mathcal{E}$  refers to *any* environment. Hence, by substituting Eq. 11 into Eq. 10,  $\pi_1(\mathcal{E} \rightarrow {}^a\mathcal{E})$  does not only cancel as it appears at both sides of the equals sign, it is also equal to one. We therefore have not just one, but two very good reasons to eliminate this term such that:

$$\rho_1(s_1|\mathcal{E})\pi_1(s_1 \rightarrow s'_1) = \rho_1(s'_1|\mathcal{E})\pi_1(s'_1 \rightarrow s_1) \quad (12)$$

or, via Eq. 3:

$$\rho_1(s_1)\pi_1(s_1 \rightarrow s'_1) = \rho_1(s'_1)\pi_1(s'_1 \rightarrow s_1) \quad (13)$$

This equation essentially the same as the standard detailed balance equation such that we can adapt our acceptance according to

$$P_{\text{acc}}(s_1 \rightarrow s'_1) = \min \left[ 1, \frac{\rho_1(s'_1)P_{\text{gen}}(s'_1 \rightarrow s_1)}{\rho_1(s_1)P_{\text{gen}}(s_1 \rightarrow s'_1)} \right] \quad (14)$$

which is exactly the same as in standard Metropolis-Hastings. Still, the underlying philosophy is different from a super-state perspective as the number of transitions from old to new,  $S^{(o)} \rightarrow S^{(n)}$ , is not the same as from new to old,  $S^{(n)} \rightarrow S^{(o)}$ . Instead, by writing  $S = (s_1, \mathcal{E})$  we have that the number of  $(s_1^{(o)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(n)}, {}^a\mathcal{E}^{(n)})$  transitions should be equal to the number of  $(s_1^{(n)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(o)}, {}^a\mathcal{E}^{(n)})$  transitions. In addition, as at the end of the move we only update ensemble 1, and not those that are here considered as environment, the number of sampled states in the ensembles do not increase in cohort. Sampling all states simultaneously like in a true superstate move would imply that distributions get mixed with the unknown and unphysical  $\rho_i^u$  distributions.

For the swapping move we just consider the example of an attempted  $1 \leftrightarrow 2$  swap as all other swaps  $i \leftrightarrow j$  are completely analogous. We start again at Eq. 7 with  $\alpha = 1 \leftrightarrow 2$ , and further we split the environment  $\mathcal{E} = \{s_2, \mathcal{E}_\sharp\}$  into the part that participates in the swap move,  $s_2$ , and the rest,  $\mathcal{E}_\sharp$ :

$$\begin{aligned} & \int \rho_1(s_1|s_2, \mathcal{E}_\sharp)\rho_2(s_2)\rho(\mathcal{E}_\sharp)P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}_\sharp) \times \pi_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_\sharp \rightarrow s'_1, s'_2, \mathcal{E}'_\sharp)ds_2d\mathcal{E}_\sharp ds'_2d\mathcal{E}'_\sharp ds'_1 = \\ & \int \rho_1(s''_1|s''_2, \mathcal{E}''_\sharp)\rho_2(s''_2)\rho(\mathcal{E}''_\sharp)P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}''_\sharp) \times \pi_{1\leftrightarrow 2}(s'_1, s''_2, \mathcal{E}''_\sharp \rightarrow s_1, s''_2, \mathcal{E}''_\sharp)ds''_2d\mathcal{E}''_\sharp ds''_2d\mathcal{E}''_\sharp ds'_1 \end{aligned} \quad (15)$$

Here, we assume that the selection probability  $P_{1\leftrightarrow 2}^{\text{sel}}$  depends on  $\mathcal{E}_j$ . The chance to do a replica exchange move equals  $P_{\text{RE}}$ , but once it is decided to perform a replica exchange move, all possible swaps  $i \leftrightarrow j$  compete to be selected with an equal probability. Hence, the probability for the  $1 \leftrightarrow 2$  swap to be selected depends on the number of available ensembles, which is the total number of ensembles minus the number of occupied ones. This latter information is contained in  $\mathcal{E}_j$ .

The swapping transition probability  $\pi_{1\leftrightarrow 2}$  relates to a move that has only one possible outcome, namely the one in which the states in ensemble 1 and 2 are exchanged. Therefore,  $\pi_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s'_1, s'_2, \mathcal{E}'_j)$  is vanishing if  $s'_1 \neq s_2$  and  $s'_2 \neq s_1$ . Likewise,  $\pi_{1\leftrightarrow 2}(s'_1, s'_2, \mathcal{E}'_j \rightarrow s_1, s_2, \mathcal{E}_j)$  vanishes if  $s'_2 \neq s_1$  and  $s'_1 \neq s_2$ . We can, therefore, write

$$\begin{aligned} \pi_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s'_1, s'_2, \mathcal{E}'_j) &= \hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s_2, s_1, \mathcal{E}'_j) \delta(s_2 - s'_1) \delta(s_1 - s'_2) \\ \pi_{1\leftrightarrow 2}(s'_1, s'_2, \mathcal{E}'_j \rightarrow s_1, s_2, \mathcal{E}_j) &= \hat{\pi}_{1\leftrightarrow 2}(s'_1, s'_2, \mathcal{E}'_j \rightarrow s_1, s_2, \mathcal{E}_j) \delta(s'_2 - s_1) \delta(s'_1 - s_2) \end{aligned} \quad (16)$$

where the transition probability with the hat,  $\hat{\pi}_{1\leftrightarrow 2}$ , differs from transition probability without the hat,  $\pi_{1\leftrightarrow 2}$ , by the fact that the latter considers any potential (even if impossible) result of the swapping operation, while the former actually relates to the probability of successfully executing the move in practice in which  $s_1$  and  $s_2$  change places. Substitution of Eqs. 16 in Eq. 15 allows us to eliminate the integrals over  $s'_1$ ,  $s'_2$ ,  $s'_1$ , and  $s'_2$  via the delta-function integration property.

$$\begin{aligned} &\int \rho_1(s_1|s_2, \mathcal{E}_j) \rho_2(s_2) \rho(\mathcal{E}_j) P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}_j) \hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s_2, s_1, \mathcal{E}'_j) ds_2 d\mathcal{E}_j d\mathcal{E}'_j = \\ &\int \rho_1(s''_1|s_1, \mathcal{E}''_j) \rho_2(s_1) \rho(\mathcal{E}''_j) P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}''_j) \hat{\pi}_{1\leftrightarrow 2}(s''_1, s_1, \mathcal{E}''_j \rightarrow s_1, s_2, \mathcal{E}''_j) d\mathcal{E}''_j ds''_1 d\mathcal{E}''_j \end{aligned} \quad (17)$$

We then eliminate the integrals over  $\mathcal{E}'_j$  and  $\mathcal{E}''_j$  using a similar expression as Eq. 8.

$$\begin{aligned} &\int \rho_1(s_1|s_2, \mathcal{E}_j) \rho_2(s_2) \rho(\mathcal{E}_j) P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}_j) \hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s_2, s_1, {}^a\mathcal{E}_j) ds_2 d\mathcal{E}_j = \\ &\int \rho_1(s''_1|s_1, \mathcal{E}''_j) \rho_2(s_1) \rho(\mathcal{E}''_j) P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}''_j) \hat{\pi}_{1\leftrightarrow 2}(s''_1, s_1, \mathcal{E}''_j \rightarrow s_1, s_2, {}^a\mathcal{E}_j) d\mathcal{E}''_j ds''_1 \end{aligned} \quad (18)$$

In the next step, we change some of the dummy integration variable names:  $s''_1$  to  $s_2$  and  $\mathcal{E}''_j$  to  $\mathcal{E}_j$ .

$$\begin{aligned} &\int \rho_1(s_1|s_2, \mathcal{E}_j) \rho_2(s_2) \rho(\mathcal{E}_j) P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}_j) \hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s_2, s_1, {}^a\mathcal{E}_j) ds_2 d\mathcal{E}_j = \\ &\int \rho_1(s_2|s_1, \mathcal{E}_j) \rho_2(s_1) \rho(\mathcal{E}_j) P_{1\leftrightarrow 2}^{\text{sel}}(\mathcal{E}_j) \hat{\pi}_{1\leftrightarrow 2}(s_2, s_1, \mathcal{E}_j \rightarrow s_1, s_2, {}^a\mathcal{E}_j) d\mathcal{E}_j ds_2 \end{aligned} \quad (19)$$

and use a detailed-balance principle by stating that the equality does not only hold when integrated, but is true for any pair  $s_2, \mathcal{E}_j$ .

$$\rho_1(s_1|s_2, \mathcal{E}_j) \rho_2(s_2) \hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s_2, s_1, {}^a\mathcal{E}_j) = \rho_1(s_2|s_1, \mathcal{E}_j) \rho_2(s_1) \hat{\pi}_{1\leftrightarrow 2}(s_2, s_1, \mathcal{E}_j \rightarrow s_1, s_2, {}^a\mathcal{E}_j) \quad (20)$$

We further simplify  $\rho_1(s_1|s_2, \mathcal{E}_j)$  by  $\rho_1(s_1)$  using Eq. 3, and split  $\hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_j \rightarrow s_2, s_1, {}^a\mathcal{E}_j)$  into  $\hat{\pi}_{1\leftrightarrow 2}(s_1, s_2 \rightarrow s_2, s_1) \times \pi_{1\leftrightarrow 2}(\mathcal{E}_j \rightarrow {}^a\mathcal{E}_j)$  where the latter term cancels like before:

$$\rho_1(s_1) \rho_2(s_2) \hat{\pi}_{1\leftrightarrow 2}(s_1, s_2 \rightarrow s_2, s_1) = \rho_1(s_2) \rho_2(s_1) \hat{\pi}_{1\leftrightarrow 2}(s_2, s_1 \rightarrow s_1, s_2) \quad (21)$$

Since  $\hat{\pi}_{1\leftrightarrow 2}(s_2, s_1 \rightarrow s_1, s_2)$  is the transition probability from  $(s_1, s_2)$  to  $(s_2, s_1)$  in the first two ensembles given that the  $1 \leftrightarrow 2$  swap move was selected, and given that there are no other possible outcomes of this swap ( $P_{\text{gen}} = 1$ ), the transition probability equals the acceptance probability:

$$\rho_1(s_1) \rho_2(s_2) P_{\text{acc}}(s_1, s_2 \rightarrow s_2, s_1) = \rho_1(s_2) \rho_2(s_1) P_{\text{acc}}(s_2, s_1 \rightarrow s_1, s_2) \quad (22)$$

To satisfy this relation, Eq. (4) of the main article suffices.

$$P_{\text{acc}} = \min \left[ 1, \frac{\rho_1(s_2) \rho_2(s_1)}{\rho_1(s_1) \rho_2(s_2)} \right] \quad (23)$$

So also here, the standard replica exchange acceptance rule applies. The main difference is that ensembles are not updated in cohort. After the  $1 \leftrightarrow 2$  swap move we only update ensembles 1 and 2. Alternatively, after the  $1 \leftrightarrow 2$  swap all other free ensembles will be updated as well with "null moves". In the example of Eq. 1 this would mean that besides, ensemble 1 and 2, also ensemble 4 would be updated. As the state in this ensemble is not changing in a  $1 \leftrightarrow 2$  swap, this would imply recounting the existing  $s_4$  state. Hence, this could be viewed as a superstate move, but then without the occupied states. Resampling  $s_4$  is allowed as the chance for resampling is independent of the content of ensemble 4. However, the sampling of the ensembles 3 and 5 should, while occupied, at all cost be avoided since the time that ensembles 3 and 5 remain occupied can correlate with the values of  $s_3$  and  $s_5$ , respectively.

Like in Eq. 14, the acceptance rule of Eq. 23 is based on a twisted detailed balance relation: we require that, given an equilibrium distribution, the number of  $(s_1^{(o)}, s_2^{(o)}, \mathcal{E}_j^{(o)}) \rightarrow (s_1^{(n)}, s_2^{(n)}, a\mathcal{E}_j^{(n)})$  transitions should be equal to the number of  $(s_1^{(n)}, s_2^{(n)}, \mathcal{E}_j^{(n)}) \rightarrow (s_1^{(o)}, s_2^{(o)}, a\mathcal{E}_j^{(o)})$  transitions, where  $s_1^{(o)} = s_2^{(n)} = s_1$  and  $s_2^{(o)} = s_1^{(n)} = s_2$ . So in this section, we proved that standard acceptance-rejection rules can be applied in a parallel scheme in which replica exchange moves occur only between unoccupied ensembles, such that ensembles are not updated in cohort.

### III. MATRICES WITH CONSECUTIVE ONES AND ZEROS

If the high-acceptance approach is not applied,  $w_i(X) = 1$  in Eq. (6) of the main article and the  $W$ -matrix has rows consisting of a sequence of ones, followed a sequence of zeros. The  $P$ -matrix can then be determined from Eq. (7) of the main article which has an  $\mathcal{O}(n^2)$  scaling. In this section we provide the proof of this equation.

Let  $n_i$  be the number of ones in row  $i$ . The first step to order the rows with increasing order of  $n_i$ . For instance in the following  $5 \times 5$  matrix

$$W = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

we see that  $s_2$ , originating from an MC move in ensemble  $e_2$ , is also valid for  $e_3$  and  $e_4$ . State  $s_3$  that was created in  $e_3$  only reaches the minimal condition for that ensemble. In path sampling, where  $s_2$  and  $s_3$  are paths and  $e_2$ ,  $e_3$  and  $e_4$  refer to path ensembles  $[k^+]$ ,  $[l^+]$  and  $[m^+]$  with  $m > l > k$ , it would mean that path  $s_3$  crosses  $\lambda_l$ , but not  $\lambda_m$ , while path  $s_2$  crosses at least  $m - k$  more additional interfaces than strictly needed for being a valid trajectory in  $e_2 = [k^+]$ . As a result, the third row has fewer ones than the second row. After reordering, the  $W$ -matrix looks as follows:

$$W = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} s'_1 = s_1 \\ s'_2 = s_3 \\ s'_3 = s_2 \\ s'_4 = s_4 \\ s'_5 = s_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix} = W[n_1, n_2, n_3, n_4, n_5] = W[2, 3, 4, 4, 5]$$

where we introduced the bracket notation  $W[\cdot]$  indicating the number of ones in each row in which  $1 \leq n_1 \leq n_2 \leq n_3 \dots \leq n_n = n$ . Likewise, we always have  $n_i \geq i$ .

Based on the recursive relation,  $\text{perm}(W) = \sum_j W_{1j} \text{perm}(W\{1j\})$ , and the fact that the matrix after removing row 1 and column  $j$ ,  $W\{1j\}$ , is identical for any  $j \leq n_1$ , we can write

$$\text{perm}(W[n_1, n_2, n_3, \dots, n_n]) = n_1 \times \text{perm}(W[n_2 - 1, n_3 - 1, \dots, n_n - 1]) \quad (24)$$

The permanent of the remaining matrix  $W[n_2 - 1, n_3 - 1, \dots, n_n - 1]$  can again be written as  $(n_2 - 1) \times \text{perm}(W[n_3 - 2, \dots, n_n - 2])$  and so on. The permanent is, hence, equal to

$$\text{perm}(W[n_1, n_2, \dots, n_n]) = \prod_{i=1}^n (n_i + 1 - i) \quad (25)$$

The  $P$ -matrix follows from Eq. (5) of the main article:  $P_{ij} = W_{ij} \text{perm}(W\{ij\}) / \text{perm}(W)$ . This means that  $P_{ij} = 0$  whenever  $W_{ij} = 0$ . If  $W_{ij} = 1$ , and  $n_{i-1} < j$  or  $i = 1$ , we have that for a matrix  $W[n_1, n_2, \dots, n_{i-1}, n_i, n_{i+1}, \dots, n_n]$  the following matrix remains after removal of row  $i$  and column  $j$ :

$$W\{ij\} = W[n_1, n_2, \dots, n_{i-1}, n_{i+1} - 1, \dots, n_n - 1] \quad (26)$$

and the permanent

$$\begin{aligned} \text{perm}(W\{ij\}) &= \left( \prod_{i'=1}^{i-1} (n_{i'} + 1 - i') \right) \left( \prod_{i'=i+1}^n (n_{i'} - 1 + 1 - (i' - 1)) \right) \\ &= \left( \prod_{i'=1}^{i-1} (n_{i'} + 1 - i') \right) \left( \prod_{i'=i+1}^n (n_{i'} + 1 - i') \right) = \frac{\text{perm}(W)}{(n_i + 1 - i)} \end{aligned} \quad (27)$$

and, therefore, for this case we have

$$P_{ij} = \frac{1 \times \text{perm}(W\{ij\})}{\text{perm}(W)} = \frac{1}{(n_i + 1 - i)}. \quad (28)$$

If for some  $k < i$ ,  $n_k \geq j$ , while  $n_{k-1} < j$  or  $k = 1$ , we have that for a matrix  $W[n_1, n_2, \dots, n_{k-1}, n_k, \dots, n_i, n_{i+1}, \dots, n_n]$  the following matrix remains after removal of row  $i$  and column  $j$ :

$$W\{ij\} = W[n_1, n_2, \dots, n_{k-1}, n_k - 1, n_{k+1} - 1, \dots, n_{i-1} - 1, n_{i+1} - 1, \dots, n_n - 1] \quad (29)$$

Therefore, the permanent of  $W\{ij\}$  can be written as

$$\begin{aligned} \text{perm}(W\{ij\}) &= \left( \prod_{i'=1}^{k-1} (n_{i'} + 1 - i') \right) \left( \prod_{i'=k}^{i-1} (n_{i'} - 1 + 1 - i') \right) \left( \prod_{i'=i+1}^n (n_{i'} + 1 - 1 - (i' - 1)) \right) \\ &= \left( \prod_{i'=1}^{k-1} (n_{i'} + 1 - i') \right) \left( \prod_{i'=k}^{i-1} (n_{i'} - i') \right) \left( \prod_{i'=i+1}^n (n_{i'} + 1 - i') \right) \\ &= \frac{\text{perm}(W)}{(n_i + 1 - i)} \prod_{i'=k}^{i-1} \frac{(n_{i'} - i')}{n_{i'} + 1 - i'} \end{aligned} \quad (30)$$

This gives for  $P_{ij}$ :

$$P_{ij} = \frac{1}{(n_i + 1 - i)} \prod_{i'=k}^{i-1} \frac{(n_{i'} - i')}{n_{i'} + 1 - i'} \quad (31)$$

We can compare this result with that of one row below (row  $i + 1$ ):

$$P_{(i+1)j} = \frac{1}{(n_{i+1} + 1 - (i + 1))} \prod_{i'=k}^i \frac{(n_{i'} - i')}{n_{i'} + 1 - i'} = \frac{P_{ij}(n_i + 1 - i)}{(n_{i+1} - i)} \frac{(n_i - i)}{n_i + 1 - i} = P_{ij} \frac{n_i - i}{(n_{i+1} - i)} \quad (32)$$

Therefore, we have following recursive relations

$$P_{ij} = \begin{cases} 0, & \text{if } W_{ij} = 0 \\ \frac{1}{n_i + 1 - i}, & \text{if } W_{ij} = 1 \text{ and } [W_{(i-1)j} = 0 \text{ or } i = 1] \\ \left( \frac{n_{i-1} + 1 - i}{n_i + 1 - i} \right) P_{(i-1)j}, & \text{otherwise} \end{cases} \quad (33)$$

For the example given above, this relation gives the following  $P$ -matrix:

$$P = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \begin{matrix} s'_1 = s_1 \\ s'_2 = s_3 \\ s_3 = s_2 \\ s'_4 = s_4 \\ s'_5 = s_5 \end{matrix} & \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

This  $\mathcal{O}(n^2)$  algorithm can be done within a second for  $n \leq 3500$ , bigger than any foreseeable RETIS simulation, without even leveraging the block-diagonalization. One could swap again the second and third row to get them ordered according to the original  $s_i$ -states, though there is in principle no need for this. This is because it is irrelevant to connect the existing states to the ensembles in which they were originally created.

#### IV. KRAMER'S THEORY

For Langevin dynamics, Kramer's relation provides a way to improve upon transition state theory via an approximate expression for the transmission coefficient:

$$\kappa = (1/\omega_b) \left( -\gamma/2 + \sqrt{\gamma^2/4 + \omega_b^2} \right) \quad (34)$$

Here,  $\gamma$  is the friction coefficient of the Langevin dynamics and  $\omega_b = \sqrt{k/m}$  with  $m$  the particle's mass and  $k$  the curvature along the reaction coordinate at the transition state. The rate constant is then the product of the transmission coefficient times the transition state theory expression for the rate:

$$k = \kappa k^{\text{TST}} \quad (35)$$

For a one-dimensional motion along a coordinate  $z$ , the transition state theory expression can be expressed as [1]:

$$k^{\text{TST}} = \sqrt{\frac{k_B T}{2\pi m}} \frac{e^{-\beta V(0)}}{\int_{-\infty}^0 e^{-\beta V(z)} dz} \quad (36)$$

where  $V(\cdot)$  is the underlying potential,  $T$  the temperature,  $k_B$  the Boltzmann constant, and  $\beta = 1/k_B T$ . The transition state is here assumed to be located at  $z = 0$  and the system is in state  $A$ , the reactant state, if  $z < 0$ .

The Kramer's approximation for the rate constant  $k$  follows from Eqs. 34-36. However, other properties like crossing probabilities and the permeability through a membrane can be derived from the transmission coefficient as well.

The crossing probability  $P_A(\lambda_B|\lambda_A)$  from interface  $\lambda_A$  to interface  $\lambda_B$  follows from the main TIS/RETIS rate equation:

$$k = f_A P_A(\lambda_B|\lambda_A) \quad (37)$$

where  $f_A$  is the conditional flux through  $\lambda_A$  given the system is in state  $A$ . Here,  $\lambda_A$  and  $\lambda_B$  correspond to the first,  $\lambda_0$ , and last interface,  $\lambda_M$ , respectively. The flux  $f_A$  through  $\lambda_A$  is similar to  $k^{\text{TST}}$ , the flux through the transition state without recrossing correction, as it counts all positive crossings and is based on the same normalization (integration over state  $A$ ):

$$f_A = \sqrt{\frac{k_B T}{2\pi m}} \frac{e^{-\beta V(\lambda_A)}}{\int_{-\infty}^0 e^{-\beta V(z)} dz} \quad (38)$$

From Eqs. 34-38 we end up with an equation for the crossing probability:

$$P_A(\lambda_B|\lambda_A) = \frac{\kappa e^{-\beta V(0)}}{e^{-\beta V(\lambda_A)}} \quad (39)$$

Hence, based on the underlying potential and Kramer's expression, Eq. 34, one can obtain an approximate value for the crossing probability. Likewise, for a membrane system we can derive a Kramer's expression for the permeability  $P$  starting from Eq. 18 in Ref.[2]:

$$P = \frac{k}{(\rho_{\text{ref}})_A} = \frac{f_A P_A(\lambda_B|\lambda_A)}{(\rho_{\text{ref}})_A} \quad (40)$$

where  $\rho_{\text{ref}}$  refers to the probability density for a permeant at a location away from the membrane,  $z_{\text{ref}}$ , where  $V(\cdot)$  is considered to be flat, and the subscript  $(\cdot)_A$  indicates that it is normalized over the reactant state region  $A$ :

$$(\rho_{\text{ref}})_A = \frac{e^{-\beta V(z_{\text{ref}})}}{\int_{-\infty}^0 e^{-\beta V(z)} dz} \quad (41)$$

Note that the integral in the denominator of Eqs. 38 and 41 is usually diverging since the underlying potential  $V(\cdot)$  is generally flat away from the barrier in a membrane system. Fortunately, this integral term cancels in Eq. 40:

$$P = \sqrt{\frac{k_B T}{2\pi m}} \left( \frac{e^{-\beta V(\lambda_A)}}{e^{-\beta V(z_{\text{ref}})}} \right) P_A(\lambda_B|\lambda_A) = \sqrt{\frac{k_B T}{2\pi m}} \left( \frac{\kappa e^{-\beta V(0)}}{e^{-\beta V(z_{\text{ref}})}} \right) \quad (42)$$

where in the second equality we substituted  $P_A(\lambda_B|\lambda_A)$  using Eq. 39. Hence, based on Eq. 34 and Eq. 42, we can obtain a value for the permeability based on Kramer's theory.

The aforementioned equations can be generalized for multidimensional systems by replacing the  $V(z)$  terms with the Landau free energy  $F(z)$ . That is, for one additional degree of freedom  $y$ :

$$F(z) = -k_B T \ln \left( \int e^{-\beta V(y,z)} dy \right) \quad (43)$$

In addition, if multiple reaction channels yield competing parallel saddle points in the potential energy surface, these need to be summed up as we will do in the next section.

#### A. Kramer's relation for crossing probability of a two-channel system

The potential energy surface described in Ref. [2] is the following

$$V(y, z) = e^{-cz^2} \left( V_1 + A + A \sin \left( \frac{2\pi y}{L_y} \right) + B + B \cos \left( \frac{4\pi y}{L_y} \right) \right) \quad \text{with} \\ A = (V_2 - V_1)/2, B = V_{\text{max}}/2 - V_1/4 - V_2/4, V_1 = 10, V_2 = 11, V_{\text{max}} = 20, c = 1, L_y = 6 \quad (44)$$

Note that the potential is periodic along the  $y$ -direction such that  $V(y, z) = V(y + L_y, z)$  and that it is zero in the limit  $|z| \rightarrow \infty$ . Further, the following mass, Langevin friction coefficient and thermodynamic parameters were set in dimensionless reduced units:  $\gamma = 5$ ,  $T = m = k_B = \beta = 1$ . The first and last interfaces were set at:  $\lambda_A = -1.5$  and  $\lambda_B = 1.2$ . In this case, we have two saddle points at  $(-L_y/4, 0)$  and at  $(+L_y/4, 0)$  where the former is slightly lower in potential energy by  $1k_B T$  ( $V_1$  and  $V_2$ , respectively). The curvatures can be obtained by applying a second order Taylor expansion around  $z = 0$ :

$$V(-L_y/4, z) \approx V_1 - cV_1 z^2 \Rightarrow k_1 = 2cV_1 \\ V(+L_y/4, z) \approx V_2 - cV_2 z^2 \Rightarrow k_2 = 2cV_2$$

which gives  $w_{b,1} = \sqrt{20}$  and  $w_{b,2} = \sqrt{22}$ . As a result  $\kappa_1 = 0.5866$ ,  $\kappa_2 = 0.6002$  via Eq. 34. From this we can compute the crossing probability based on essentially Eq. 39, but using the Landau free energy,  $F(\cdot)$ , by Eq. 43, instead of the potential energy,  $V(\cdot)$ , and using both transmission coefficients for the parts along the orthogonal coordinate,  $y$ , where they are relevant:

$$P_A(\lambda_B|\lambda_A) \approx \frac{\kappa_1 \int_{-3}^0 e^{-\beta V(y,0)} dy + \kappa_2 \int_0^3 e^{-\beta V(y,0)} dy}{\int_{-3}^3 e^{-\beta V(y,\lambda_A)} dy} = 1.61 \cdot 10^{-5} \quad (45)$$

where the integrals over  $y$  are taken over one period. Note that the system in Ref. [2] actually contains 3 particles that move in this 2D potential energy surface such that the dimension of the system is actually 6. However, since we follow one single target permeant and the other particles are assumed to have no influence on the target (the interparticle interaction was set to 0 [2]), the effective dimension for our analysis is 2 with coordinates  $y$  and  $z$ .

The permeability then follows from Eq. 42 with  $V(\cdot)$  replaced by  $F(\cdot)$ , where we used the expression based on the crossing probability to have the effect of the two different transmission coefficients directly included:

$$P = \sqrt{\frac{k_B T}{2\pi m}} \left( \frac{\int_{-3}^3 e^{-\beta V(y,\lambda_A)} dy}{\int_{-3}^3 e^{-\beta V(y,z_{\text{ref}})} dy} \right) P_A(\lambda_B|\lambda_A) = \frac{1}{6} \sqrt{\frac{k_B T}{2\pi m}} \left( \int_{-3}^3 e^{-\beta V(y,\lambda_A)} dy \right) P_A(\lambda_B|\lambda_A) = 1.37 \cdot 10^{-6} \quad (46)$$

where we assumed that  $z_{\text{ref}}$  is taken far away from the membrane at  $z = 0$  such that  $z_{\text{ref}} \ll 0$  and  $V(y, z_{\text{ref}}) \approx 0$ .



## B. Kramer's relation for crossing probability of double well potential

The double well potential is given by [3]

$$V(z) = k_1 z^4 - k_2 z^2 \text{ with } k_1 = 1, \quad k_2 = 2 \quad (47)$$

which has a transition state at  $z = 0$  and minima at  $z = -1$  and  $z = 1$ . Further is given that  $T = 0.07$  and  $k_B = m = 1$  such that the transition state theory expression for the rate, Eq. 36, equals [3]:  $k^{\text{TST}} = 2.776 \cdot 10^{-7}$ .

The curvature at the transition state equals  $2k_2 = 4$  such that  $w_b = 2$ . Together with the friction coefficient of  $\gamma = 0.3$ , Kramer's relation, Eq. 34, provides a transmission coefficient:  $\kappa = 0.9278$ . Henceforth, by Eq. 35 the rate constant based on Kramer's theory equals:  $k = 2.58 \cdot 10^{-7}$ .

The crossing probability follows from Eq. 39 where in this case  $\lambda_A = -0.99$  [4]. From the previously determined value for  $\kappa$ , we get:  $P_A(\lambda_B|\lambda_A) = 5.83 \cdot 10^{-7}$

## V. COMPUTATIONAL EFFICIENCIES

In this paper, the computational efficiency is defined as

$$\text{efficiency} = \frac{1}{\tau^{\text{eff}}} \quad (48)$$

where  $\tau^{\text{eff}}$  is the efficiency time [5], which is equal to the computational cost that is needed to get a statistical relative error equal to 1 for the property that is computed. Here,  $\tau^{\text{eff}}$  could be expressed as the number of MD steps in path sampling simulations of large systems or path sampling simulations based on Ab Initio MD where the number of force calculations completely determines the total CPU cost. Expressing the efficiency time in this way has the advantage that it is hardware independent. In this article, however, we express the efficiency time in actual CPU- or wall-time seconds in order to include also the computational cost for calculating the permanents in the replica exchange move.

When a simulation is completed after a certain time  $\tau$  and the relative error  $\epsilon$  has been obtained via, e.g. independent runs, block averaging or bootstrapping, the efficiency time is estimated by

$$\tau^{\text{eff}} = \epsilon^2 \tau \quad (49)$$

Note that for serial simulations this property is in principle independent of the simulation length  $\tau$ . If we increase the simulation by a certain factor, the error should reduce by the square root of this factor such that  $\tau^{\text{eff}}$  remains unchanged. However, we should realize that there is a rather large statistical uncertainty in the estimated values for  $\tau^{\text{eff}}$  due to the fact that the statistical error in the error is generally large.

In the following, unless stated otherwise, we will refer to the CPU-time and CPU-based efficiency time when referring to  $\tau$  and  $\tau^{\text{eff}}$ . However, let us shortly discuss the wall-time efficiency that follows from the same equation, Eq. 49, but with  $\tau$  being the wall-time instead of CPU-time. In all our simulations, we fixed the wall-time to  $5 \times 12$  hours with 5 independent runs. So the wall-time is constant and independent to the number of workers that is used. However, with  $K$  workers instead of 1, the CPU-time increases by a factor  $K$ . This means that if the error would follow the same trend as in a serial run, the use of  $K$  instead of 1 worker would result in a  $\sqrt{K}$  reduction of the error. Yet, with  $\tau$  in Eq. 49 being the wall-time instead of CPU-time, the reduction in the error is not canceled by an increase in  $\tau$  and the efficiency, Eq. 48, would increase linearly with  $K$ . This would mean that we can write:

$$\text{efficiency}(\text{wall-time}) = K \times \text{efficiency}(\text{CPU-time}) \quad (50)$$

if the parallel run uses the total CPU-time as effectively as a serial simulation that runs  $K \times 5 \times 12$  hours long. However, our parallel algorithm will introduce changes in the relative CPU-time that is used for MC moves in the different ensembles. This effect was investigated for the memoryless single variable stochastic (MSVS) process. In the next subsection, we give the meaning and derivation of the continuous curves shown in Fig. 1 of the main article.

### A. Theoretical efficiencies for the MSVS process

The efficiency time can also be calculated for specific parts of the calculation. In specific, TIS/RETIS consists of different path ensemble simulations that compute a local crossing probability. In the path ensemble  $[k^+]$  which consists of paths that at least cross  $\lambda_k$ , this local crossing probability equals the fraction of paths that cross  $\lambda_{k+1}$  as

well. Based on the expected error in the local crossing probability, the CPU-based efficiency time of ensemble  $[k^+]$  can be expressed as [5]:

$$\tau_k^{\text{eff}} = \frac{1-p_k}{p_k} \mathcal{N}_k \xi_k L_k \quad (51)$$

where  $p_k$  is the local crossing probability of ensemble  $[k^+]$ ,  $L_k$  is the average path length (expressed in MD steps or CPU seconds), and  $\xi_k$  is the ratio of the average cost of a MC move to  $L_k$ . In other words,  $\xi_k L_k$  is the average computational cost for doing a MC move (creation of a trial path that might then be accepted or rejected). Finally,  $\mathcal{N}_k$  is a measure of the effective correlations between MC moves also called the "statistical inefficiency". Paths can be correlated due to rejections, which implies that the old path is recounted, or because of similarities between accepted paths. In practice,  $\mathcal{N}_k$  tends to be significantly larger than 1 while  $\xi_k$  is often smaller than 1 as many rejections occur without that a trial path needs to be fully completed. In addition, some MC moves like the replica exchange move or the time-reversal move do not require any MD steps.

In the following, we will neglect the effect that the replica exchange moves have on the errors and on the CPU-time. Under this assumption, the successive MC moves are completely independent. In addition, the ensemble moves are memoryless (hence  $\mathcal{N} = 1$ ). The overall error can thus be computed from the errors in the individual ensembles using standard error propagation rules for independent estimates. Except for the replica exchange part, the MSVS simulation is rejection-free such that we also have  $\xi = 1$ . In addition, the random artificial MD time for a path in ensemble in ensemble  $[k^+]$  was on average  $0.1k + 0.1$  seconds. To simplify our analysis, we neglect the final 0.1 addition, and state that  $L_k = ak$  with  $a = 0.1$ . Finally, we fixed the local crossing probability to  $p_k = p = 1/10$  for all ensembles  $[k^+]$  such that

$$\tau_k^{\text{eff}} = a \frac{1-p}{p} k \quad (52)$$

The relative error in estimate of the local crossing probability of ensemble  $[k^+]$  follows from Eq. 49:

$$\epsilon_k = \sqrt{\frac{\tau_k^{\text{eff}}}{\tau_k}} \quad (53)$$

with  $\tau_k$  the CPU-time that is spend to ensemble  $[k^+]$ . Given a certain division of the total simulation time  $\tau$  into the times  $(\tau_0, \tau_1, \dots, \tau_{N-1})$ , we can compute the total efficiency time by Eq. 49 with

$$\epsilon^2 = \sum_{k=0}^{N-1} \epsilon_k^2 = \sum_{k=0}^{N-1} \frac{\tau_k^{\text{eff}}}{\tau_k} \quad \text{and} \quad \tau = \sum_{k=0}^{N-1} \tau_k \quad (54)$$

The first expression is the standard error propagation rule for the error in a final estimate that is obtained from a product of independent estimates.

Now let us first consider standard TIS or the  $N = K$  case. In this simulation we would have an equal number of workers as ensembles. Each worker is solely designated to a single ensemble such that an equal amount of CPU-time is spend per ensemble when the simulation is stopped. So we can simply put  $\tau_k = 1$  such that  $\tau = N$  and

$$\epsilon^2 = \sum_{k=0}^{N-1} \tau_k^{\text{eff}} = a \frac{1-p}{p} \sum_{k=0}^{N-1} k = a \frac{1-p}{p} \frac{1}{2} (N-1)N \approx \frac{a}{2} \frac{1-p}{p} N^2 \quad (55)$$

where in the last equality we assumed  $N \gg 1$ . The efficiency time for TIS is hence

$$\tau^{\text{eff}} \approx \frac{1}{2} a \frac{1-p}{p} N^3, \quad \text{for TIS or } K = N \quad (56)$$

For serial RETIS, each ensemble is updated by a MC move before a next cycle of moves is started. As a result, in each ensemble the same number of MC moves are carried out such that  $\tau_k \propto L_k \propto k$ . By taking  $\tau_k = k$ , we get that  $\tau = (N-1)N/2 \approx N^2/2$  and

$$\epsilon^2 = \sum_{k=0}^{N-1} \frac{\tau_k^{\text{eff}}}{\tau_k} = aN \frac{1-p}{p} \quad (57)$$

and the CPU-based efficiency time is exactly the same

$$\tau^{\text{eff}} \approx \frac{1}{2} a \frac{1-p}{p} N^3, \quad \text{for RETIS or } K = 1 \quad (58)$$

This is in agreement with Ref. [5] which stated that an equal division of CPU-time or aiming for the same error in each ensemble gives the same efficiency. Since the local crossing probability is the same for each ensemble,  $p_k = p$ , aiming for the same error in each ensemble is equivalent to having the same number of MC moves per ensemble (if the statistical inefficiencies,  $\mathcal{N}_k$ , are the same). The optimal division of CPU-time over the different ensembles is, however,  $\tau_k \propto \sqrt{\tau_k^{\text{eff}}}$  [5]. By taking  $\tau_k = \sqrt{k}$ , the total CPU-time becomes

$$\tau = \sum_{k=0}^{N-1} \sqrt{k} \approx \int_0^N \sqrt{x} dx = \frac{2}{3} N^{3/2} \quad (59)$$

and the total error

$$\epsilon^2 = \sum_{k=0}^{N-1} \frac{\tau_k^{\text{eff}}}{\tau_k} = a \frac{1-p}{p} \sum_{k=0}^{N-1} \sqrt{k} \approx a \frac{1-p}{p} \frac{2}{3} N^{3/2} \quad (60)$$

which by Eq. 49 results in a slightly lower efficiency time than for TIS/RETIS:

$$\tau^{\text{eff}} \approx \frac{4}{9} a \frac{1-p}{p} N^3, \quad \text{for an optimal division} \quad (61)$$

Based on  $a = p = 0.1$  and  $N = 50$ , the efficiency times are  $\tau^{\text{eff}} = 56250$  for TIS/RETIS and  $\tau^{\text{eff}} = 50000$  for the optimal division. Naturally, the corresponding CPU-time efficiencies by Eq. 48 are  $1/56250$  and  $1/50000$ . Furthermore, based on Eq. 50, the optimal wall-time efficiency and the optimal TIS/RETIS wall-time efficiency are given by  $K/50000$  and  $K/56250$ , respectively. These are the continuous black and purple curves in Fig.1d of the main article.

It is interesting to observe that the optimal TIS/RETIS CPU-time efficiency is only 12.5% lower than the optimal CPU-time efficiency. This seems to suggest that it is difficult to improve the CPU-time efficiency of TIS and RETIS unless the division of CPU-time is exactly targeted to do so. On the other hand, one can easily get a much worse CPU-time efficiency when errors in some ensembles are reduced to unnecessary small values while the other ensemble errors are ignored. Based on the fact that  $\tau_k \propto \sqrt{\tau_k^{\text{eff}}}$  gives the optimum, the optimum division of MC moves is obtained when in ensembles  $[k^+]$  the number of MC moves is proportional to  $\sqrt{\tau_k^{\text{eff}}/L_k}$ . For the MSVS system this means that the number of executed MC moves in each ensemble should optimally be taken as  $\propto 1/\sqrt{k}$  for  $k = 1, 2, \dots, M-1$  (to account for  $k = 0$  we should have kept the neglected 0.1 addition in the path length to avoid divergence). This means that it is actually good to execute more MC moves at the lower rank ensembles (low  $k$ ) than at the higher rank (high  $k$ ). However, this should not be exaggerated since too many MC moves in the low ranked ensembles will just result in inefficient use of CPU-time as discussed above. Based on the numerical sampling ratios, we determined the CPU-time spend in each ensemble,  $\tau_k$ , by multiplying these ratios by  $L_k = ak$ . We then estimated the error based on Eqs. 54 and 52. The resulting efficiency, based on the actual sampling ratios of  $\infty$ RETIS, turned out to give a slightly better CPU-time efficiency than that of TIS/RETIS. The resulting wall-time efficiencies of this hybrid theoretical/numerical result is shown by the purple dots in Fig.1d as well. This shows that  $\infty$ RETIS can actually improve both the CPU- and wall-time efficiency compared to TIS/RETIS. The latter is expected based on the brute force principle that more CPU power is used per second. The former is more subtle and related to the fact that  $\infty$ RETIS leads to a more efficient distribution of the CPU-time among the different ensembles compared to TIS or RETIS.

## VI. ADDITIONAL SIMULATION RESULTS

### A. Ratios of channels crossings

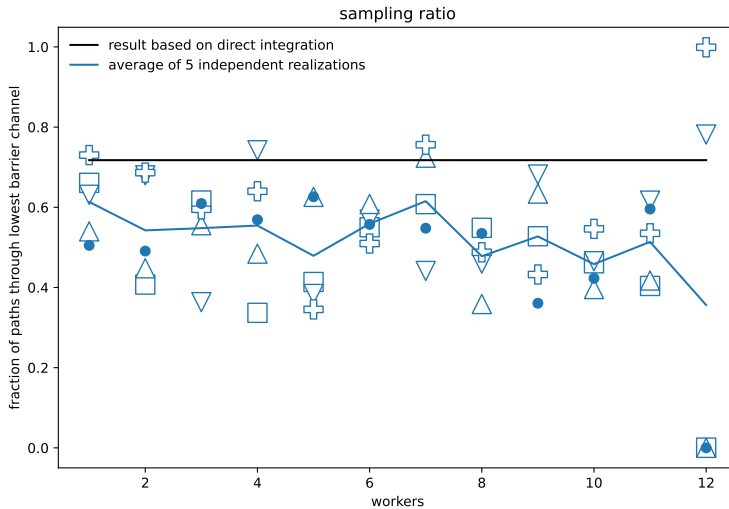


FIG. 1. The ratio of first crossings points for the last ensemble in the more favorable channel. The blue icons shows the sampled ratio for each simulation, the blue line is the average of 5 simulations for each amount of workers and the black line is the expected value from direct integration of  $\exp(-\beta V(y, z))$  over  $y$  with  $z$  fixed at  $z = \lambda_{10} = -0.2$ . The 3 icons overlapping at 0.0 for 12 workers is the result of the known ergodicity issues of the TIS algorithm due to the lack of replica exchange moves.

- 
- [1] D. Frenkel and B. Smit, *Understanding molecular simulations from algorithms to applications* (Academic press, San Diego, California, U.S.A., 2002).
- [2] A. Ghysels, S. Roet, S. Davoudi, and T. S. van Erp, Exact non-markovian permeability from rare event simulations, *Phys. Rev. Research* **3**, 033068 (2021).
- [3] T. van Erp, Dynamical rare event simulation techniques for equilibrium and nonequilibrium systems, *Adv. Chem. Phys.* **151**, 27 (2012).
- [4] D. T. Zhang, E. Riccardi, and T. S. van Erp, Path sampling with sub-trajectory moves, In preparation **xx**, **xx** (2021).
- [5] T. S. van Erp, Efficiency analysis of reaction rate calculation methods using analytical models I: The two-dimensional sharp barrier, *J. Chem. Phys.* **125**, 174106 (2006).

## Paper F

# Path sampling simulations reveal how the Q61L mutation alters the dynamics of KRas

Sander Roet, Ferry Hooft, Peter G. Bolhuis,  
David W.H. Swenson, and Jocelyne Vreede

*Manuscript*



# Path sampling simulations reveal how the Q61L mutation alters the dynamics of KRas

Sander Roet,<sup>†</sup> Ferry Hooft,<sup>‡</sup> Peter G. Bolhuis,<sup>‡</sup> David W.H. Swenson,<sup>¶</sup> and  
Jocelyne Vreede<sup>\*,‡</sup>

<sup>†</sup>*Department of Chemistry, Norwegian University of Science and Technology (NTNU),  
Trondheim, Norway*

<sup>‡</sup>*van 't Hoff Institute for Molecular Sciences, University of Amsterdam, Amsterdam, The  
Netherlands*

<sup>¶</sup>*Univ Lyon, ENS de Lyon, Univ Claude Bernard, CNRS, Laboratoire de Physique and  
Centre Blaise Pascal, Lyon, France*

E-mail: J.Vreede@uva.nl

## Abstract

The GTPase KRas is a signaling protein in networks for cell differentiation, growth, and division. When active, KRas tightly binds GTP. KRas mutations can affect the conversion between this rigid state and inactive, more flexible states, thus prolonging activation of signal transduction pathways, which may result in tumor formation. Transitions in KRas take place on time scales of microseconds and longer, which are difficult to characterize experimentally at high resolution in both space and time. In this work, we applied path sampling simulations to investigate the dynamic behaviour of KRas-4B (wild-type, WT) and the oncogenic mutant Q61L (Q61L). Our results show KRas visiting several states, which are the same for WT and Q61L. The multiple state transition path sampling (MSTPS) method samples transitions between

the different states simultaneously, by allowing switching between different transitions. Large differences occurred in the switching dynamics between WT and Q61L. Further investigation of the MSTPS results revealed that for Q61L a route to a flexible state is inaccessible, which shifts the equilibrium to more rigid states. The methodology presented here enables a detailed characterization of protein flexibility on time scales not accessible with brute-force molecular dynamics simulations.

## Introduction

Ras GTPases are signal transduction proteins that mediate cell growth, cell differentiation and death. Binding of guanosine triphosphate (GTP) activates signal transduction by Ras proteins, while their GTPase function inactivates signal transduction again by hydrolyzing GTP to guanosine diphosphate (GDP). Ras GTPases comprise the most frequently occurring family of oncoproteins in human cancers.<sup>1,2</sup> Mutations in Ras proteins initiate cell transformation, drive oncogenesis and promote tumor maintenance. The Ras family of oncoproteins has been studied extensively for almost three decades, as activation of Ras represents a key feature of malignant transformation for many cancers. In the cancers that contribute most heavily to worldwide mortality, Ras mutations are extremely common.<sup>3</sup> Several isoforms of Ras exist, which are implicated in different types of cancer.<sup>2,3</sup> A member of this family, KRas-4B, is often found in common and life-threatening cancers, such as lung cancer, colon cancer and pancreatic cancer.<sup>3</sup>

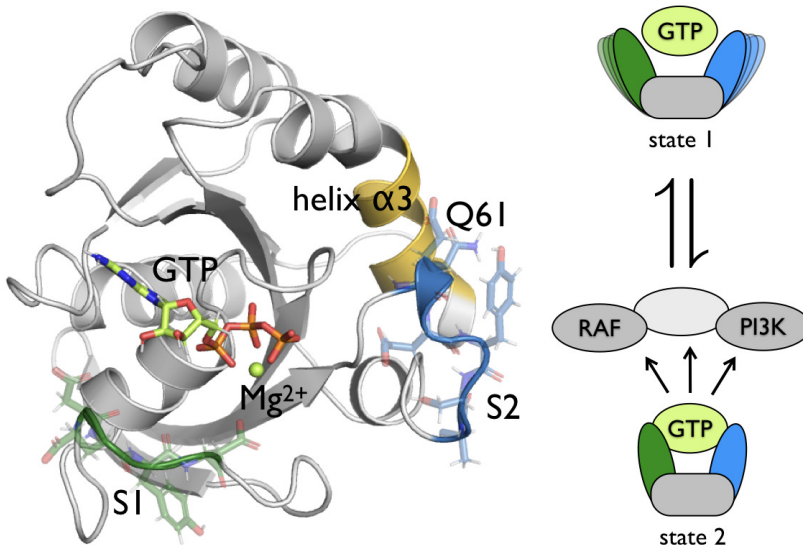
Ras proteins consist of a highly conserved catalytic domain called the G domain and a variable C domain which anchors Ras in the membrane. In this work we focus on the G domain of the KRas-4B isoform, which contains 166 residues and can be considered as the minimal signaling unit. This domain contains the guanine nucleotide binding site and two regions that can detect the nature of the bound nucleotide, switch 1, S1, and switch 2, S2. These regions, highlighted in green (S1) and blue (S2) in figure 1(left), are involved in many interactions between Ras and partners. In the GTP-bound state, Ras interacts with



downstream effectors such as the Raf and PI3K kinases.<sup>4</sup> After hydrolysis of GTP, these loop regions adopt a more open conformation<sup>5</sup> and exhibit more flexibility, causing Ras to lose the ability to bind to downstream effectors. While bound to GTP, Ras exists in a dynamic equilibrium between a weakly populated state 1 and a dominant state 2.<sup>6,7</sup> Conformational state 1 is more flexible and open than the closed, ordered state 2, as schematically shown in figure 1(right). Crystal structures of Ras bound with GTP analogues are typically in the state 2 form of Ras.<sup>4,8</sup> However, <sup>31</sup>P-NMR studies report that the switch regions can also adopt disordered conformations when bound to GTP, similar to the GDP-bound state.<sup>9</sup> This work will focus on the transition between the ordered (state 2) and disordered (state 1) states of GTP-bound KRas-4B.

Even though the role of Ras mutations in tumor formation has long been recognised, Ras is considered undruggable.<sup>12,13</sup> Targeting direct competition with GTP binding is difficult, as Ras has a picomolar affinity for GTP, with micromolar concentrations of GTP in cancer cells. The absence of a hydrophobic pocket for the binding of small molecules complicates the development of allosteric inhibitors of Ras. Obtaining a more detailed understanding of the dynamics underlying the activation of Ras could provide new insights that eventually could lead to new therapeutic leads. However, probing the dynamics of Ras at sufficient resolution in both space and time proves to be very difficult experimentally.<sup>14</sup>

Molecular dynamics (MD) simulations are well suited to obtain high resolution insights into protein dynamics.<sup>14</sup> While it is currently possible to run microseconds of straightforward MD, an investigation of the mechanism and kinetic aspects of protein conformational transitions is not feasible. Transitions occurring on microsecond or longer timescales involves high free energy barriers separating stable conformational states. During an MD simulation, most of the time is spent in the stable states, waiting for a barrier crossing, resulting in poor sampling of the transitions. Protein flexibility often involves more than two stable configurations, requiring the sampling of several transitions. The transition path sampling (TPS) algorithm<sup>15</sup> addresses this timescale problem by focusing the MD simulations on the



**Figure 1: Ras structure and function.** Structure of GTP-bound KRas in the active state 2 (left) and a schematic representation of the inactive state 1 and the active state 2 of GTP-bound KRas (right). In the protein drawing, the switch regions are highlighted in green for S1 and blue for S2, helix  $\alpha 3$  is highlighted in yellow. The protein is shown as a ribbon with an transparent stick representation for the amino acids in S1 and S2. GTP is shown as solid sticks, with carbon atoms colored in green, oxygen in red, nitrogen in blue and phosphorus in orange.  $Mg^{2+}$  is shown as a green ball. Note that no consensus has been reached yet on the residues ranges that correspond to S1 and S2.<sup>10,11</sup> We chose to use a narrow definition that corresponds to the residues that are important for the conformational changes in this study, using residues 30-33 for S1 and residues 60-66 for S2. In the schematic drawing the S1 region is represented in green, the S2 region in blue, and the rest of the protein in grey. State 1 corresponds to the conformational state in which S1 and S2 are more flexible and not bound to GTP. State 2 corresponds to the conformational state in which both S1 and S2 are bound to GTP. State 2 activates downstream effectors like RAF and PI3K by binding them.

barrier regions. TPS is a Monte Carlo (MC) simulation in the space of trajectories and collects an ensemble of short reactive trajectories connecting a predefined initial and final state, without prior knowledge of the transition state region. The speed-up gained by using TPS and related techniques is tremendous. Assuming a transition rate in the order of  $10 \mu s^{-1}$ , observing a single transition would require on average  $10 \mu s$  of MD. In contrast, when using TPS, the barrier region is sampled using MD trajectories of only tens of nanoseconds, thus

providing a speed up in the order of several thousand to a million. Even though path sampling methods like TPS were originally developed for two states, they have been extended to be used with multiple stable states.<sup>16</sup> In this setup, transitions between any two stable states can be sampled, and therefore it is possible to generate trajectories that connect different pairs of states. The frequency of such switching between transitions depends on the barrier separating different transition channels. Analysis of the switching behavior between these transition channels provides useful insight into the overall dynamics.

It is still an open question how Ras converts from ordered to less ordered conformational states. Ras-activating mutations include substitutions at glutamine (Q) 61,<sup>17</sup> which affect the conformational equilibrium of Ras. Changing Q61, located in S2, results in reduced GTPase activity in Ras<sup>18</sup> and an altered conformational space for KRas-4B.<sup>19</sup> Replacing Q61 by leucine (L) results in an oncogenic mutant.<sup>10</sup> By changing a hydrophilic glutamine into a hydrophobic leucine,<sup>20</sup> the hypothesis is that the conformational space of GTP-bound KRas-4B will change, and alter the transition between state 1 and state 2. In particular the effect of mutations on these transitions is unclear. In this work we present multiple state TPS simulations of KRas-4B and the oncogenic Q61L mutant, showing that indeed S2 displays different dynamics for the two systems. In particular, the WT switches frequently from one transition to another, while the Q61L hardly switches at all. Here, we present a way to qualitatively analyse the kinetics of the switching behavior. Closer examination reveals that the WT S2 can reach the flexible open state via a channel that is not accessible for Q61L. Both WT and Q61L can reach the open state by S2 sliding along a slightly hydrophobic pocket of the  $\alpha$ 3-helix. However, the Q61L mutation prevents direct solvation of S2, which is possible for the WT protein. As a result, the open, inactive state will occur less frequently in Q61L and the protein is more likely to be in an ordered state. Our results show that our methodology is able to map out the dynamics of a Ras protein and can indicate differences in dynamics between a WT protein and an oncogenic mutant. Moreover, the methodology presented here is able to reveal details on the nature of the altered behavior as caused by

the mutation. As such, this work is an example of using MSTPS simulations to characterize protein flexibility on time scales of microseconds and longer.

## Results and discussion

### Identification of conformational states

The crystal structure of GppNHp bound HRas (PDB: 4EFL)<sup>21,22</sup> was used as a structural template to model the sequence of WT and Q61L KRas-4B with GTP bound. With these two structures we performed four 100 ns MD simulations to explore the conformational space of KRas, for both WT and Q61L. These simulations resulted in the characterization of two stable states for S1, S1-D33 and S1-open, and two stable states for S2, S2-GTP, and S2-open. After initial TPS simulation ended up with trajectories that did not end in any of these states, a third state was found for S1, S1-30-32, as well as S2, S2- $\alpha$ 3. All the stable states are shown in figure 2. When S1 is in the S1-D33 state, the side chain of D33 is involved in (water-mediated) hydrogen bond interactions with GTP. For the S1-30-32 state, S1 has shifted along GTP, compared to the S1-D33 state, to form one or more hydrogen bonds between the side chains of residues D30, E31, or Y32 and GTP. The conformations in which S1 has no hydrogen bond interaction with GTP and where it is oriented away from GTP are classified as the S1-open state. The S2-GTP state corresponds to the conformation of S2 where it forms hydrogen bonds with GTP. Two states can occur when S2 is oriented away from GTP. In the S2- $\alpha$ 3 state, S2 has multiple interactions between its side chains and the  $\alpha$ 3-helix. In the S2-open state S2 has no binding interactions with GTP. The parameters for defining these states are listed in Appendix S1. Within the timescale of the path sampling simulations, the conformation of S1 has little effect on the conformation of S2 or vice versa. The same stable states were found for both WT and Q61L, and were stable for at least 100 ns of MD. Note that no consensus has been reached yet on the range of residues that correspond to each switch region.<sup>10,11</sup> We chose to use a narrow definition that includes the residues that

are important for the conformational changes in this study, using residues 30-33 for S1 and residues 60-66 for S2. Adding more residues will not change the stable state definitions.

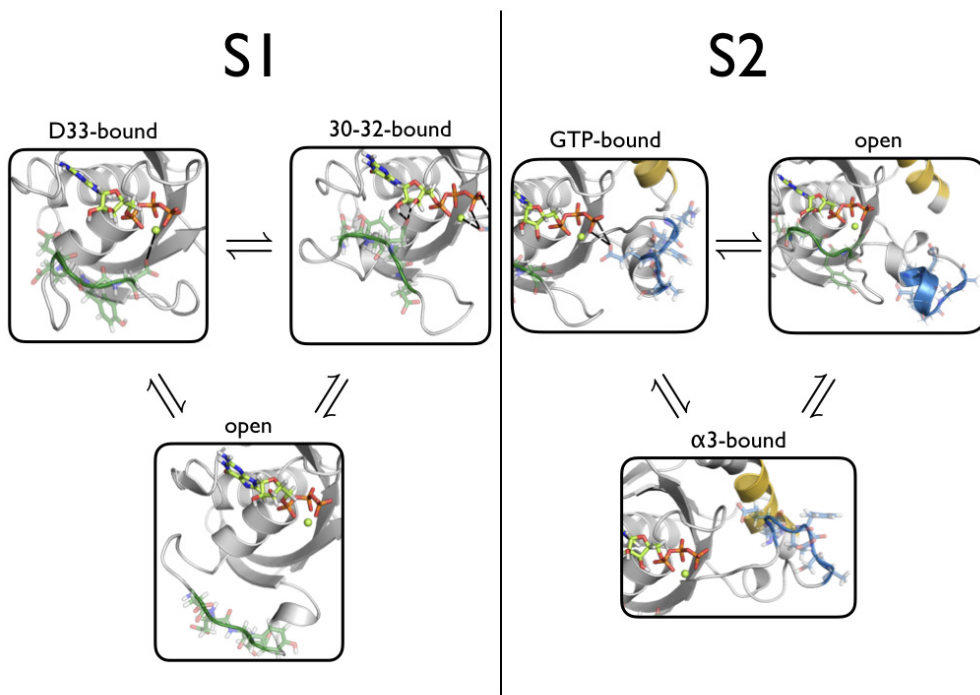


Figure 2: **Stable states of KRas.** The stable states found for S1 and S2 are shown with the same coloring as figure 1(left). The S1-D33 state corresponds to the conformation in which D33 in S1 has a hydrogen bond with GTP. The S1-30-32 state corresponds to the conformation in which one or more hydrogen bonds occur between residues 30-32 and GTP. The S1-open state corresponds to the conformation in which S1 has no interactions with GTP and is oriented away from GTP. For S2 the S2-GTP state corresponds to the conformation in which S2 has one or more hydrogen bonds with GTP. The S2-open state corresponds to a state in which S2 has no interactions with GTP and is oriented away from GTP. The S2- $\alpha$ 3 state corresponds to a conformation in which S2 has no interactions with GTP, but instead has 4 interactions with the  $\alpha$ 3-helix.

## Mapping conformational transitions

Using MSTPS, we investigated the transitions between the stable states as identified in the MD simulations for S1 and S2 separately. For both S1 and S2 three pairs of transitions are observed: S1-D33  $\leftrightarrow$  S1-30-32, S1-D33  $\leftrightarrow$  S1-open, and S1-30-32  $\leftrightarrow$  S1-open for S1, and S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3, S2-GTP  $\leftrightarrow$  S2-open, and S2- $\alpha$ 3  $\leftrightarrow$  S2-open for S2. For both WT and Q61L we performed one MSTPS simulation for S1 starting at the S1-30-32  $\leftrightarrow$  S1-open transition, and three independent MSTPS simulations for S2, each starting in a different transition, resulting in eight MSTPS simulations in total. The statistics of the MSTPS simulations are listed in table 1 and indicate a good acceptance ratio of 33% or higher, and an aggregate simulation time of microseconds.

Table 1: Statistics of the MSTPS simulations. MC steps indicates the number of Monte-Carlo trials, also called shooting moves. Accepted steps refers to the number of MC trials that were accepted and the acceptance is  $\frac{\text{accepted steps}}{\text{MC steps}}$ . Decorrelated trajectories indicate the number of accepted trajectories that do not have any frames in common. The total simulation time is the total time of MD performed by the MD engine in the MSTPS simulations.

	S1 WT	S1 Q61L	S2 WT	S2 Q61L					
	sim 1	sim 1	sim 1	sim2	sim 3	sim 1	sim 2	sim 3	
MC steps	1000	1000	2000	2000	2000	2000	2000	2000	
Accepted steps	355	334	766	824	787	688	748	759	
Acceptance	35.5%	33.4%	38.3%	41.2%	39.4%	34.4%	37.4%	38.0%	
Decorrelated trajectories	50	57	131	151	128	99	129	125	
Average path length (ns)	5.94	2.28	3.49	1.26	1.50	1.42	1.36	1.98	
Total simulation time ( $\mu$ s)	4.88	2.67	6.01	2.15	2.10	2.26	1.83	3.74	

Path sampling simulations for proteins (with stochastic dynamics and diffuse barriers) commonly employ the stochastic, or “one-way” shooting algorithm,<sup>23</sup> which improves the acceptance ratio. In this algorithm a trial move replaces only part of the trajectory (forward or backward). Therefore, successive trajectories will have segments with overlapping frames and at least two trials (one forward and one backward) are needed for an accepted trajectory to have no frames in common with the original. These no-overlap trajectories are referred to as “decorrelated”, and are required for sufficient sampling. Table1 shows that each simulation

generated on average 100 decorrelated trajectories.

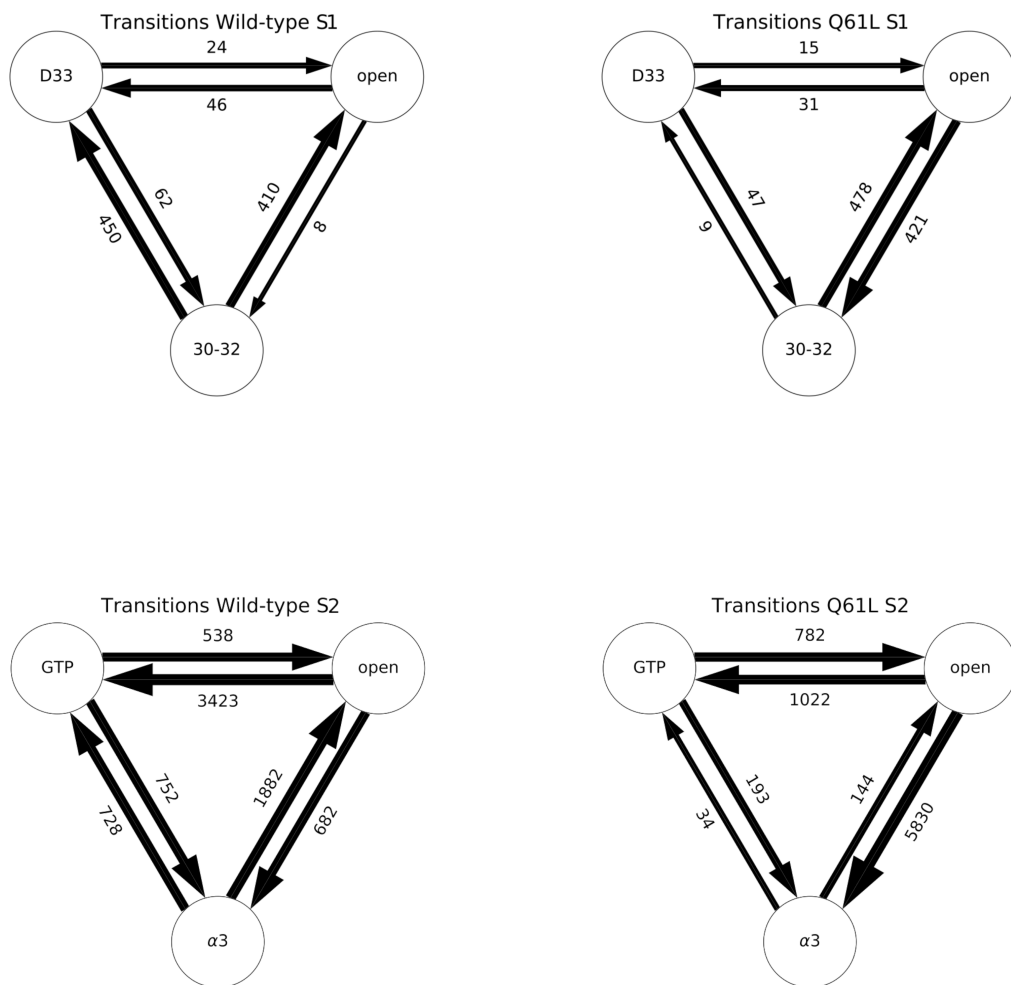


Figure 3: **Transitions of S1, and S2 in WT and Q61L.** A schematic representation of the number of samples per transition for S1(top) and S2(bottom) for the WT (left) and Q61L (right) TPS simulations. The circles correspond to the stable states of S1, with D33 being the S1-D33 state, 30-32 being the S1-30-32 state, and open the S1-open state. For the S2 plots, GTP is the S2-GTP state, open the S2-open state, and  $\alpha 3$  the S2- $\alpha 3$  state. The arrows represent the sampled transition, pointing in the direction that was considered forward during the simulation. The labels are the number of accepted paths in each transition, and the width of the arrows is scaled with a  $10 \log$  scale of this number.

The transitions sampled in the MSTPS simulations are summarized in figure 3. Between

each pair of states there are two possible transitions, corresponding to what is the forward time direction in the path. Both the WT and the Q61L simulations have sampled all allowed transitions for both S1 and S2. At infinite sampling, the relative sampling frequency for the two transitions between a given pair of states will be identical.

When looking at the transitions sampled for S1 WT, the number of samples differ for the directions of the 30-32  $\leftrightarrow$  D33 and the 30-32  $\leftrightarrow$  open by more than an order of magnitude, this indicates that this simulation did not converge. For the S1 Q61L system, the number of samples in both directions of each transition is within the same order of magnitude, which is a first indication that this simulation might be converged. While the WT simulation has not converged, for all three of the transitions the WT has at least one sampling direction with the same order of magnitude as Q61L (like the D33  $\rightarrow$ 30-32 transition, which has 62 and 47 samples for respectively the WT and Q61L). This, together with the fact that this switch is unaltered by the mutation, makes us believe that all transitions would be the same order of magnitude between S1 Q61L and S1 WT when the WT simulation converges.

For S2, while the transitions are different between the wild-type and Q61L, the counts for most transition pairs are within an order of magnitude of each other, indicating acceptable sampling. The exception is the S2-open and S2- $\alpha$ 3 pair in Q61L, where the S2-open $\rightarrow$ S2- $\alpha$ 3 transition is sampled 5830 times, and the S2-open $\leftarrow$ S2- $\alpha$ 3 transition only 114 times. This difference in sampling the S2-open and S2- $\alpha$ 3 transition in WT and Q61L suggests that the mutation has altered the transition region. Such an alteration can happen in two ways, that do not exclude each other. The mutation changes the stability of one or more of the stable states, thus making transitions less (or more) likely. The mutation can also alter the mechanism of the transition, which has a direct effect on the transition region.

Although MSTPS only samples one transition at a time, switching between transitions can occur when there are more than two states. For example, given states  $A$ ,  $B$ , and  $C$ , an initial  $A \rightarrow B$  trajectory can produce a trial  $A \rightarrow C$  trajectory if a forward trial ends in state  $C$ . Such a transition of transitions is called a “switch”. The Methods section contains



a more detailed explanation. Analyzing the switching behavior provides useful insight into the transition region. A lack of switching between two states indicates there is a large (free energy) barrier in the transition region between the channels for the individual reactions.

Conversely, many switching events suggests a flatter, more diffusive landscape in the transition region.

Figure 14 in appendix S4 plots the transition sampled in each accepted path as a function of the number of MC steps for both S1 simulations. Figures 9 and 10 in appendix S2 plot the sampled transitions per MC step for the S2 WT and S2 Q61L simulations respectively.

For both the S1 simulations MC steps resulting in accepted and decorrelated (no-overlap) trajectories are distributed mostly uniformly throughout both simulations. At around step 900 in the Q61L simulation, no acceptance occurs for several MC steps, indicating the simulation is trapped in the  $D33 \leftarrow 30-32$  transition and the probability of accepting a new (part of a) path has become very low.

For the S2 WT simulations, MC steps resulting in accepted trajectories and decorrelated trajectories are distributed uniformly throughout all 3 simulations. Throughout most of the WT simulations, switching occurs on average every 16 MC steps, indicating that the simulation loses memory of the starting transition path. The second part of the simulation starting from an S2- $\alpha 3$  to S2-open transition is an exception, as this simulation remains in the S2-open  $\rightarrow$  S2-GTP transition for over 1000 MC steps. For the S2 Q61L simulations the accepted and decorrelating MC steps are also distributed uniformly throughout all three simulations. All simulations spend a significant amount of simulation steps in the S2-open  $\rightarrow$  S2- $\alpha 3$  transition, possibly indicating that the barrier separating the open state from the S2- $\alpha 3$  state is lower in Q61L. The number of switches occurring between the transitions is much lower than in the WT simulations. The lack of switching also explains why the Q61L simulations are less well sampled than the WT simulations.

Figure 4 shows the number of switches between the six transitions occurring for S1 and the six transitions of S2 as arrows with a thickness relative to the number of switches. The

switching looks very similar between WT and Q61L for S1, however this is not the case for S2. Clearly, the number of switches between transitions in S2 is much lower for the mutant than for the WT, indicating that the WT has a lower free energy barrier between the different transition channels than Q61L. Further analysis is done only on S2 in the following sections, as the S2 data showed the strongest difference.

## Kinetic analysis of switching of S2

To quantify the relative frequency or “population” of each transition and the switching rate between transitions, we applied a kinetics analysis approach as developed for Replica Exchange MD<sup>24</sup> on our MSTPS data. In this analysis, rate constants are estimated from the number of transitions between stable states and the average residence times in the stable states. To apply this analysis to our MSTPS data, time, stable states and transitions correspond to respectively the MC trials, the transitions between the states and the switches (transitions of transitions). Such a kinetics analysis results in rate constants that measure the switching rate between transitions in units of MC steps, see the Methods section for more detail. We analyzed the kinetics by including all accepted trajectories, or only decorrelated trajectories, see Figure 5. For one Q61L simulation, starting from S2-GTP  $\rightarrow$  S2-open, no switching occurs out of the S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3 transition and we therefore excluded this simulation from the switching analysis. The “population” of the S2- $\alpha$ 3  $\leftrightarrow$  S2-open transition is almost twice as high for Q61L (63.23%) than for WT (36.09%), while the S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3 is less likely for Q61L (6.22%) than for WT (21.97%). When including all accepted paths, most switching rates are lower for Q61L than for WT, except for the switches out of the S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3 transition.

The largest relative difference between WT and Q61L are observed for the switching rates out of the S2- $\alpha$ 3  $\leftrightarrow$  S2-open transition. When including only decorrelated trajectories, all switching rates are lower for Q61L, with the largest relative difference for the transition into the S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3 transition. These observations indicate that the barrier separat-

ing S2- $\alpha$ 3 and S2-open states is lower in Q61L than in WT. In addition, it is difficult to obtain decorrelated paths for the S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3 transition. This is also apparent from figures 9 and 10 in Appendix S2, where the residence time in the S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3 transitions is almost always only a single MC step. Less switching would occur if the transition channels in the free energy landscape are less deep. An alternative view is that part of the region between transitions would be less favorable for Q61L compared to WT.

## Path densities reveal two channels for S2

To further investigate the origin of the difference in switching kinetics, we projected the trajectory space sampled in the combined TPS simulations in a path density histogram, see the Methods section for an explanation on how path density histograms are computed. Figure 6(top) shows the path density for the WT (left) and Q61L (right) S2 TPS simulations, projected in the plane of the distances between S2 and GTP, and between S2 and residues H95, Y96, Q99, and R102 of the  $\alpha$ 3-helix (see table 4 in Appendix S1 for definitions of these distances). Note that path densities do not show stable states, as the trajectories are stopped when reaching one. Comparing the two path density plots indicate that the WT simulations sample a larger region on the vertical axis, as the WT path density extends to above 1.25 nm, while the Q61L path density is more confined to the region below 1.25 nm in S2 -  $\alpha$ 3 distance.

Inspection of the least changed path, a trajectory connecting paths on top of the transition barrier, in figures 11 and 12 in Appendix S3, confirms that the observed switching is indeed a diffusive process and that the Q61L mutation constrains the dynamics. This indicates that S2 can move away from the  $\alpha$ 3-helix more easily in the WT protein. The WT histogram even shows a second channel for transitions between the S2-GTP and the S2-open states, at a distance of more than 1.75 nm from the  $\alpha$ 3-helix. The Q61L simulations sample configurations closer to the  $\alpha$ 3-helix, as indicated by the density below 0.8 nm on the vertical axis. A more pronounced negative correlation exists between the S2 -  $\alpha$ 3-helix and the

S2 - GTP distances. The further away S2 is from GTP, the closer it is to  $\alpha 3$ . Furthermore, the Q61L simulations do not sample the second channel at all. As three independent simulations were performed for both WT and Q61L, each initiated from a different transition, the absence of direct solvation transitions for Q61L are likely to be a direct consequence of the mutation.

The two conformations plotted in figure 6(bottom) illustrate the difference between the two reaction mechanisms or channels. The image on the left shows the WT protein in the S2-open state with S2 at a distance of at least 1.75 nm from the  $\alpha 3$ -helix. On the right Q61L is shown in the S2-open state with S2 closer than 1.25 nm to the  $\alpha 3$ -helix, see video supplements 1 and 2 for movies of typical trajectories for each of the two reaction channels. The distance of S2 to  $\alpha 3$  is indicative of the different mechanisms. The channel far away from the  $\alpha 3$ -helix represents a mechanism involving water molecules solvating S2, resulting in S2 extending into the solvent, away from both GTP and the  $\alpha 3$ -helix. The channel close to the  $\alpha 3$ -helix represents S2 moving along a hydrophobic pocket on the  $\alpha 3$ -helix. In this reaction mechanism, S2 can either enter the S2- $\alpha 3$  state by forming four contacts between S2 and the  $\alpha 3$ -helix, or by sliding along the helix until entering S2-open. The Q61L mutation changes a hydrophilic residue to a hydrophobic one, thus lowering the affinity of S2 for water. Therefore, the solvated transition channel, which is easily accessible for the WT protein, becomes much less likely for Q61L. Moreover, the mutated S2 has stronger interactions with the  $\alpha 3$ -helix, as shown by the higher path density in the channel close to  $\alpha 3$ -helix (Figure 6(top, right)), indicating that for Q61L it is harder to escape from the  $\alpha 3$ -state. The increased stability of the  $\alpha 3$ -state renders the S2-GTP $\leftrightarrow$ S2-open transition less likely. Figure 7 summarizes this conclusion.

### **Q61L has a higher propensity for a more structured S2-open**

Visual inspection of the transition paths shows that in some WT trajectories the  $\alpha 2$ -helix (residues 65–73, overlapping with part of S2), unfolds when entering the S2-open state,

but retains its shape for the Q61L mutant. Probability histograms of the S2-open state obtained from the transition path ensemble by projection on the number of helical hydrogen bonds in the  $\alpha$ 2-helix and the S2- $\alpha$ 3-helix distance shown in figure 13 in Appendix S3 further substantiates this observation. Therefore, we can conclude that the S2-open state contains multiple sub-states, characterized by the conformation of the  $\alpha$ 2-helix and the S2- $\alpha$ 3 distance. Furthermore, these probability histograms show that Q61L has a higher propensity compared to WT for the more structured conformations of the S2-open state. The  $\alpha$ 2 helix plays a vital role in binding other proteins,<sup>5</sup> suggesting that these structured sub-states are more similar to the active state 2 than the inactive state 1.

The more open and flexible sub-states of the S2-open state are less likely to be recognized by downstream effectors. A  $\beta$ -strand in the PI3 kinase interacts with KRas via both S1 and S2,<sup>4</sup> which can only occur when both S1 and S2 are in a closed conformation. The more open conformations are harder to reach in Q61L, and indeed, we only observe these flexible sub-states in the WT simulations, thus providing an explanation for the increased probability of Q61L to bind a downstream effector. This prediction may be tested by repeating the NMR experiment as performed by Geyer et al.,<sup>9</sup> comparing the effect of the Q61L mutation on the switching frequency. Alternatively, the lifetime of the S2- $\alpha$ 3 state could be measured using <sup>15</sup>N NMR spectroscopy, by labeling nitrogen atom NE2 in the Q95 side chain, located in the  $\alpha$ 3-helix. Finally, the  $\alpha$ 3-helix identified as important for the transitions between the ordered, active state 2 and the flexible, inactive state 1 might provide a new target for the development of compounds that could ameliorate the effect induced by the Q61L mutation.

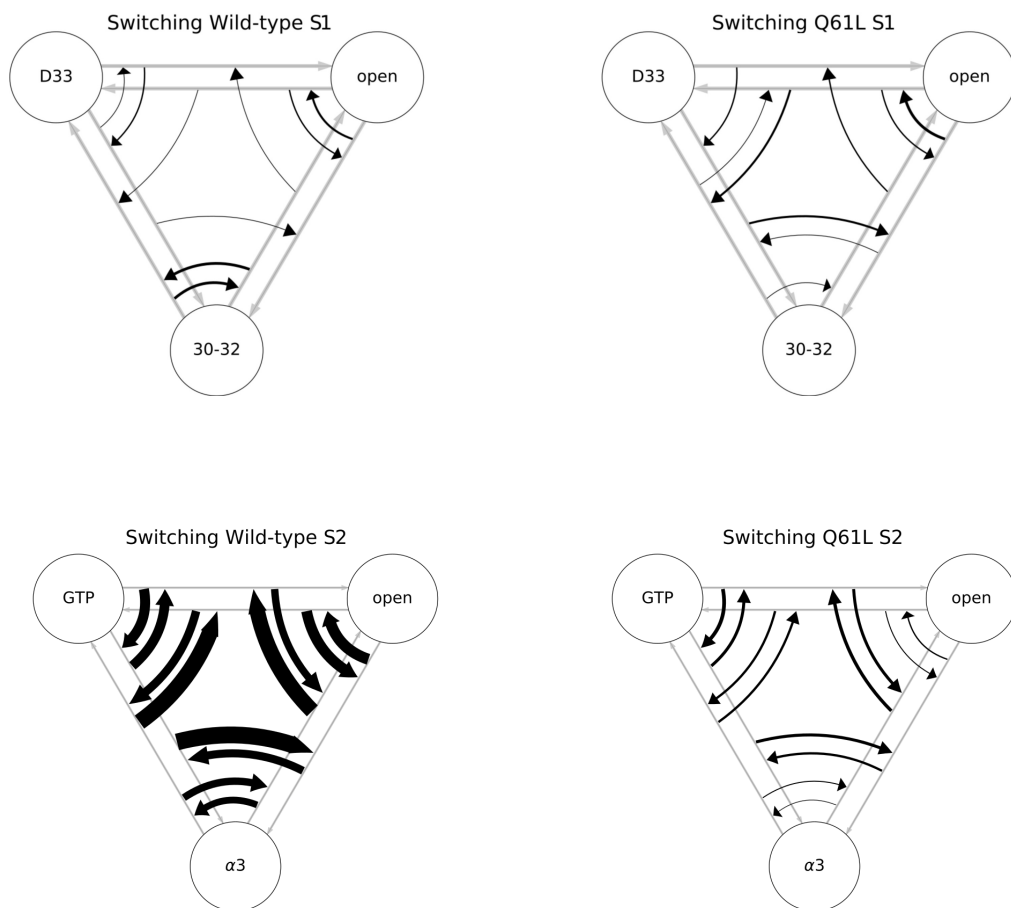


Figure 4: **Switches between transitions.** A schematic representation of the amount of switching between different sampled transitions in WT (left) and Q61L (right), for both S1(top) and S2(bottom). The circles represent the stable states and the gray arrows show the unscaled transitions. The same state abbreviations were used as in figure 3. Each of the black arrows represents a switch between the transitions, scaled linearly to the number of times this switch occurred.

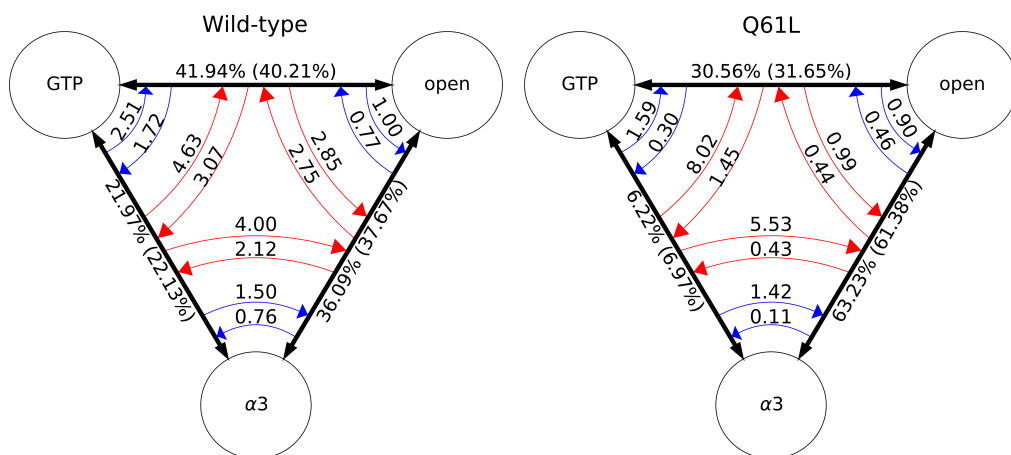


Figure 5: **Population analysis.** (left) All WT S2 simulations (right) Q61L simulations starting in the S2- $\alpha 3 \rightarrow$  S2-GTP transition and the S2-open  $\rightarrow$  S2- $\alpha 3$  transition. The circles represent the stable states and the same state abbreviations are used as in figure 3. The black arrows are the time combined transitions, the red arrows are the switching rates obtained from all accepted paths, and the blue arrows are the switching rates observed from only using decorrelated paths. The labels of the transitions arrows are the population percentages from all accepted paths, with the decorrelated data in parentheses. The labels of the switching arrows are the rates of switching per 100 MC steps.

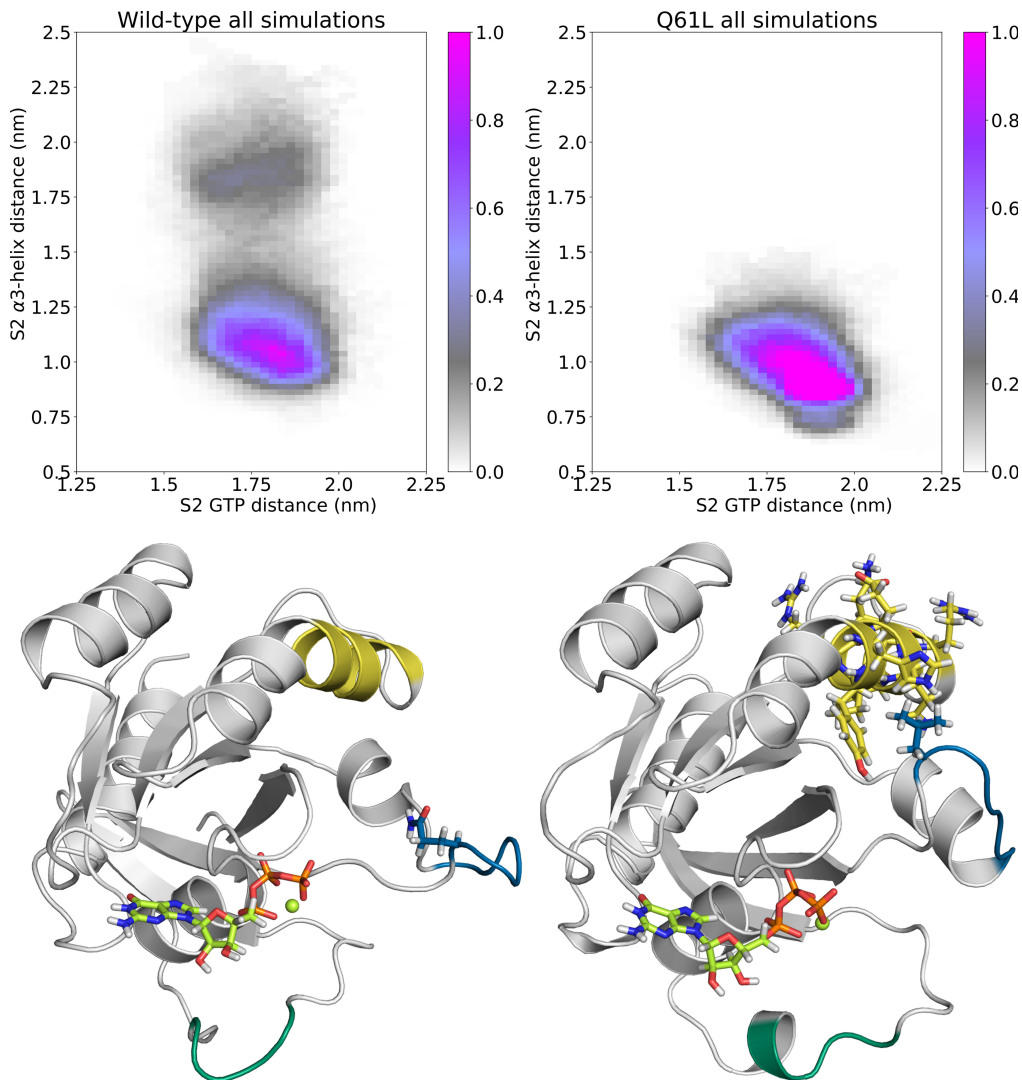


Figure 6: **Mechanistic differences between WT and Q61L.** (top) Path density histograms of the WT (left) and Q61L (right) simulations with on the x-axis the distance between the circular mean center of mass (cCOM) of S2 and on the y-axis the cCOM of GTP and on the y-axis the distance between the cCOM of S2 and the cCOM of residues 95, 96, 99, and 102. The bin widths are 0.25 Å for both axes. The coloring shows the sampled configuration of the transitions, weighted per trajectory, and normalized to 1. The stable states are not defined entirely based on these coordinates, but S2-GTP corresponds roughly to the area left of 1.6 nm on the x-axis, S2-open to the area right of 2 nm on the x-axis, and S2- $\alpha$ 3 at intermediate distances on the x-axis, and below 0.8 on the y-axis. (bottom) Snapshots of the S2-open-state with the same coloring as figure 1(left) from (left) the channel that is far away from the  $\alpha$ 3-helix in the WT simulation and (right) the channel close to the  $\alpha$ 3-helix in the Q61L simulation.



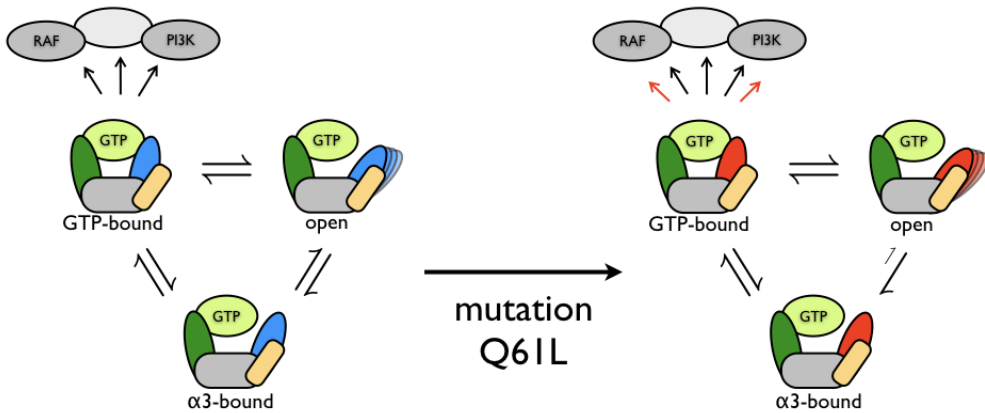


Figure 7: **Schematic overview of the effect of the Q61L mutation on the dynamics of KRas.** (left) WT (right) Q61L. The S1 region is represented in green, the S2 region in blue for WT and red for Q61L, the  $\alpha$ 3-helix in yellow, and the rest of the protein in grey. Downstream effectors are also shown in grey. Assuming only the S2 GTP-bound state triggers the downstream effectors, the Q61L mutation alters the conformational space such that one channel to reach the open state becomes very unlikely. This would lead to either a shift in the equilibrium distribution between the open and GTP-bound state or to transitions occurring more frequently. Both of these effects would lead to an increased probability to encounter downstream effectors while in the GTP-bound state, which would trigger the downstream signaling networks.

# Methods

## Structure generation

The initial GTP-bound KRas-4B structure was constructed from the crystal structure of GppNHp bound HRas (PDB-code: 4EFL).<sup>21,22</sup> This was done by first using homology modeling (MODELLER v9.16),<sup>25</sup> using sequential alignment to convert HRas to KRas-4B. Then the GppNHp was manually modified into GTP, by changing the nitrogen into an oxygen and removing the attached hydrogen. Finally, structures of the protein and the GTP were combined into a single file. The initial structure for the mutant (Q61L) was made from this structure by mutating the glutamine (Q) 61 of this final structure into a leucine (L), using MODELLER.<sup>25</sup>

The initial structures were put inside a dodecahedral periodic box with a minimum distance between the structures and the side of the box of 1 nm. This resulted in boxes with volumes of 228.154 nm<sup>3</sup> and 230.723 nm<sup>3</sup> for the wild type (WT) and Q61L, respectively. The boxes were filled with TIP3P water.<sup>26</sup> 51 of the waters were replaced by 30 Na<sup>+</sup> and 21 Cl<sup>-</sup> ions to neutralize the systems and achieve a physiological salt concentration of 0.15 M NaCl. This resulted in total system sizes of 22561 and 22857 atoms for the WT and Q61L, respectively.

## Molecular Dynamics

### Procedure

The initial systems were equilibrated in four steps, consisting of energy minimization, an isothermal equilibration, an isothermal-isobaric equilibration and a 1 ns molecular dynamics simulation. The equilibrated structures were used to run four 100 ns molecular dynamics simulations for both WT and Q61L.

## Settings

In the molecular dynamics simulations the atomic interactions were described by the AMBER99SB-ILDN<sup>27</sup> force field, extended with optimized parameters for the triphosphate chain of GTP.<sup>28</sup> Long-range electrostatic interactions were treated via the Particle Mesh Ewald method.<sup>29</sup> The short-range non-bonded interactions (e.g. electrostatics and Van der Waals interactions) were cut off at 1.1 nm.

All of the equilibration was performed with GROMACS v.4.6.5.<sup>30</sup> The leap-frog integrator was used with a time step of 2 fs. Temperature was kept constant at 310 K using the v-rescale thermostat<sup>31</sup> using two temperature coupling groups: the first group consisted of the protein, GTP and  $\text{Mg}^{2+}$ , while the second group consisted of water,  $\text{Na}^+$ , and  $\text{Cl}^-$ . The pressure was kept constant using the Parrinello-Rahman barostat<sup>32</sup> at a pressure of 1 bar. All bond lengths were constrained using the LINCS algorithm.<sup>33</sup>

The 100 ns production runs were performed with OpenMM (7.1.0.dev-5e53567).<sup>34</sup> The constraints were changed to only affect bonds including a hydrogen atom, using SHAKE,<sup>35</sup> the integrator was the Velocity Verlet with velocity randomization (VVVR) integrator<sup>36</sup> from OpenMMTools v.0.14<sup>37</sup> and the barostat was the Monte Carlo barostat.<sup>38</sup> The production simulations were run using the CUDA platform of OpenMM on NVIDIA GeForce GTX TITAN X GPUs.

## Collective variables and stable states

The long molecular dynamics runs were visually analyzed to identify stable states, using VMD.<sup>39</sup> Five types of collective variable functions were used to define the stable states, which are described in table 2 in Appendix S1.

For both WT and Q61L the relevant collective variables can be found in table 3 and 4, for S1 and S2, respectively. These collective variables are comprised of the collective variable types described in table 2. The stable states for S1 were S1-D33, S1-30-32, and S1-open, and for S2 were S2-GTP, S2- $\alpha 3$ , and S2-open. The definitions of all stable states can be

found in table 5.

## Transition Path Sampling (TPS)

In the long molecular dynamics simulations some transitions spontaneously occurred once in several 100 ns simulations. These transitions were used as the starting transition path for TPS.<sup>15,23</sup> One TPS simulation was performed for S1, starting from the S1-30-32 to S1-open transition. For S2 three TPS simulations were performed, each starting from a different transition. This was done for both WT and Q61L. The initial trajectories were first equilibrated with a TPS simulation until the first decorrelated transition path (a transition path that has no frames in common with the original path) was obtained. This decorrelated path was used as the starting point for the production TPS simulations.

### Settings

The TPS simulations were performed with `OpenPathSampling(0.1.0.dev-c192493)`.<sup>40,41</sup> Multiple state TPS (MSTPS)<sup>16</sup> was performed with an all-to-all flexible length ensemble, excluding self-transitions. All-to-all means all transitions connecting two states are allowed. A self-transition is a path that starts in a state and returns to that same state after crossing the boundaries set by the state definitions. We used the one-way shooting algorithm,<sup>42</sup> with uniform shooting point selection. For the S1 simulations, 1000 shooting trials were performed, while for each of the S2 simulations 2000 shooting trials were performed.

### Analysis

All analysis of the TPS simulations was performed using the tools included in the `OpenPathSampling` package,<sup>40,41</sup> extended with custom Python code. `Matplotlib`<sup>43</sup> was used for plotting the graphs and triangles.

The supporting figures 9 and 10 show the type of transition as a function of the MC trial for S2 of WT and Q61L.

## Path density histograms

Path density histograms (pdhs) are two-dimensional histograms that show the configurations in a transition path, projected on collective variables. Each path is weighed with its MC weight, and divided by the number of total MC trials. For example, if a trajectory visits a histogram bin, the count of that bin is increased by the MC weight of that trajectory. It does not matter how often the trajectory visits a bin, it counts the trajectory only once. The path density gives the reactive flux of trajectories, whereas regular projection would give a configurational density which is usually overwhelmed by intermediate states.

## Switching analysis

In MSTPS simulations, more than one transition is possible (e.g.,  $A \rightarrow B$ ,  $A \rightarrow C$ ,  $B \rightarrow A$ , etc.), however, only one transition is sampled at a given MC step of the MSTPS simulation. With one-way shooting, an initial  $A \rightarrow B$  trajectory can produce a trial  $A \rightarrow C$  trajectory if a forward trial ends in state  $C$  as schematically shown in figure 8. Such a transition of transitions is called a “switch.” Analyzing the switching behavior provides useful insight into the transition region.

With one-way shooting, switching between a transition  $A \rightarrow B$  and its reversed version,  $B \rightarrow A$ , requires at least three sequential switches: e.g., starting from an  $A \rightarrow B$  transition, a transition from  $A \rightarrow C$  can be generated, followed by a  $B \rightarrow C$  transition, from which the next shot can result in a  $B \rightarrow A$  transition, see figure 8 for a visualisation. One-way shooting can only change the starting or ending state with a backward or forward shot respectively, but cannot change both in the same MC step. As MSTPS samples an equilibrium distribution, the number of paths collected from the  $A \rightarrow B$  transition should be similar to the number of paths from  $B \rightarrow A$ , reversed in time, which provides a measure for convergence of the simulation.

This measure helps to provide heuristics to assess the convergence of the MSTPS simulations. First, it is an estimate of the ergodicity of the simulation whether all transitions are

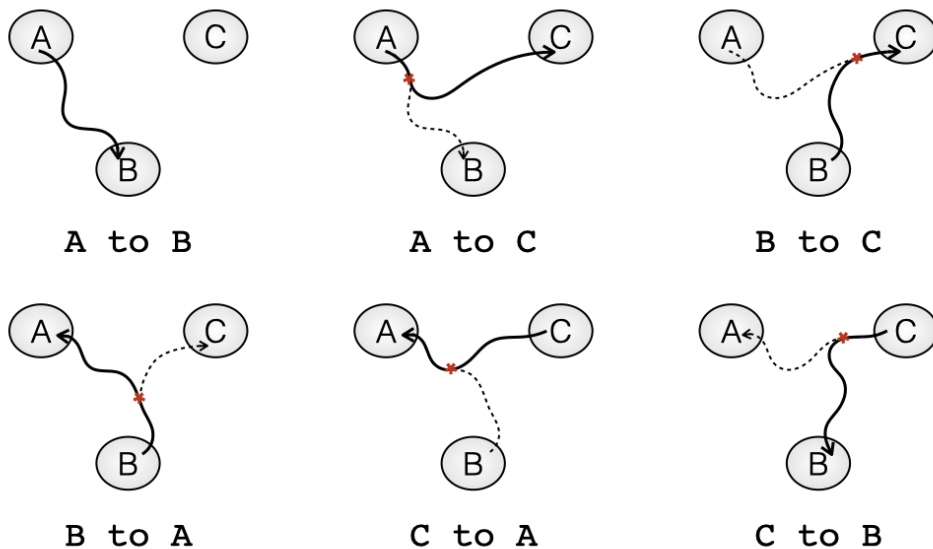


Figure 8: **Schematic overview of switching transitions in a three state system.** The three states are labeled A, B and C. The current path is indicated by a solid line. The previous accepted path is indicated by a dashed line. Shooting points are indicated by an asterisk.

visited. Second, forward and backward versions of transitions with the same pair of states (e.g.,  $A \rightarrow B$  and  $B \rightarrow A$ ) should have similar statistics in all ways. Furthermore, the fraction of MC steps spent in the two transitions between the same pair of states should be the same. The same goes for the path length distributions.

## Kinetics analysis

As we assume that the switching samples an equilibrium distribution, the probability  $P_i$  of sampling a transition  $i$  is given by:

$$P_i = \frac{1}{\sum_{j \neq i} n_{ji}} \sum_{j \neq i} \frac{n_{ij} + n_{ji}}{\frac{1}{P_i} + \frac{1}{P_j} \frac{t_j}{t_i}} \quad (1)$$

where  $n_{ij}$  is the number of switches from  $i$  to  $j$ , and  $t_i$  is the number of MC steps sampling transition  $i$ . As the sum of all probabilities is equal to one  $\sum_i P_i = 1$ , equation 1 can be solved for all  $P_i$ . From the probabilities the switching rate from  $i$  to  $j$ ,  $k_{ij}$ , can be calculated by:

$$k_{ij} = \frac{n_{ij} + n_{ji}}{t_i + t_j \frac{P_i}{P_j}} \quad (2)$$

The values for  $n$  and  $t$  are taken from the MSTPS simulations. This analysis is adapted from.<sup>24</sup>

## Conclusion

In this work we investigated the conformational space and dynamic behavior of KRas in complex with GTP using multiple state transition path sampling. The loops in KRas that interact with GTP each visit three different conformational states. Surprisingly, these conformational states do not change upon introducing the Q61L mutation, located in region S2. However, the mutation has a significant effect on the transitions between the conformational states of region S2. This effect could be a result of changes in the relative free energies of the conformational states or changes in the transition mechanisms, or a combination of both. While the WT protein frequently changes from one transition to another, the mutant hardly changes at all. Closer examination of the various transitions revealed that S2 in the WT protein is more likely to be solvated than in the Q61L mutant. The Q61L mutation prevents direct solvation of S2, which is an accessible route for the WT protein. Both WT and

mutant can reach the opened up state by S2 sliding along a slightly hydrophobic pocket on the  $\alpha$ 3-helix. Our results show that the MSTPS methodology in combination with the novel switching analysis, is able to map out the dynamics of a Ras protein, indicate differences in dynamics between the WT protein and an oncogenic mutant, and reveal details on the nature of the altered behavior as caused by the mutation.



## Supporting Information Available

**Appendix S1: Stable state definitions** The stable states are defined by ranges in collective variables. This appendix provides a guide to these stable state definitions. Table 2 gives the types of collective variables, while Tables 3 and 4 list the collective variables used to define the stable states for S1 and S2, respectively. Table 5 gives the ranges in collective variable space for the stable states found for S1 and S2.

Table 2: List of the different collective variable types.

CV type	Description
Minimum distance	The smallest distance between two groups of atoms. Using MDTraj, <sup>44</sup> distances of every atom pair were calculated. The lowest is the minimum distance.
Circular mean center of mass (cCOM)*	The circular mean center of mass (cCOM) is a center of geometry calculation that allows for periodicity. The system is first mapped onto a cube, followed by the calculation of the center of mass, using the procedure of ref. <sup>45</sup> Then the cCOM is mapped back onto the original axes.
Number of hydrogen bonds	The number of hydrogen bonds is calculated by counting how many of the possible donor-acceptor pairs form a hydrogen bond. A hydrogen bond in this code is defined by having a $H_{donor}$ -acceptor distance smaller than 0.25 nm and having an $X_{donor}$ - $H_{donor}$ -acceptor angle larger than $\frac{2}{3}\pi$ rad.
Number of water mediated hydrogen bonds	The number of water mediated hydrogen bonds is calculated by first selecting all water oxygens that are within a distance of 0.35 nm from both input groups. If a water forms a hydrogen bond with both groups, it is counted as a water mediated hydrogen bond. The hydrogen bond calculation is done as described above.
Number of bonds	The number of bonds is the number of pairs, from a given list of pairs, for which the minimum distance is smaller than 0.35 nm. The minimum distance is defined in the minimum distance cv type.

\* Note that this circular mean center of mass does not give the actual center of mass.

In the CV type column the name of the collective variable type as used in table 3 and 4 are shown, with their description in the Description column.

Table 3: **List of the relevant collective variables for the stable state definitions of S1.**

CV	Description
d_GTP_asp30 <sup>a</sup>	The minimum distance between the C <sub>γ</sub> of aspartic acid 30 and the heavy atoms of GTP, including Mg <sup>2+</sup> .
d_GTP_glu31 <sup>a</sup>	The minimum distance between the C <sub>δ</sub> of glutamic acid 31 and the heavy atoms of GTP, including Mg <sup>2+</sup> .
d_GTP_tyr32 <sup>a</sup>	The minimum distance between the side-chain oxygen of tyrosine 32 and the heavy atoms of GTP, including Mg <sup>2+</sup> .
d_GTP_asp33 <sup>a</sup>	The minimum distance between the C <sub>γ</sub> of aspartic acid 33 and the heavy atoms of GTP, including Mg <sup>2+</sup> .
n_hbonds_GTP_asp30 <sup>c</sup>	The number of hydrogen bonds between the side-chain oxygens of aspartic acid 30 and the hydroxyl groups on the ribose of GTP.
n_hbonds_GTP_tyr32 <sup>c</sup>	The number of hydrogen bonds between the hydroxyl oxygen of tyrosine 32 and the hydroxyl groups on the ribose of GTP.
n_hbonds_tyr32_GTP <sup>c</sup>	The number of hydrogen bonds between the oxygens of GTP and the hydroxyl group of tyrosine 32.
n_hbonds_ile55_tyr40 <sup>c</sup>	The number of hydrogen bonds between the backbone carbonyl of isoleucine 55 and the backbone amide of tyrosine 40.
n_hbonds_GTP_S1 <sup>c</sup>	The number of hydrogen bonds between the backbone carbonyls of valine 29 and aspartic acid 30 and the hydroxyls on the ribose of GTP.
n_h_med_bonds_GTP_asp33 <sup>d</sup>	The number of water mediated hydrogen bonds between all oxygens of aspartic acid 33 and all oxygens of GTP.
n_h_med_bonds_MG_asp33 <sup>d</sup>	The number of water mediated hydrogen bonds between all oxygens of aspartic acid 33 and Mg <sup>2+</sup> .
n_h_med_bonds_GTP_nnb <sup>d</sup>	The number of water mediated hydrogen bonds between the side-chain oxygens of aspartic acid 30, glutamic acid 31 and tyrosine 32 and all oxygens of GTP.
n_h_med_bonds_MG_nnb <sup>d</sup>	The number of water mediated hydrogen bonds between the side-chain oxygens of aspartic acid 30, glutamic acid 31 and tyrosine 32 and Mg <sup>2+</sup> .

<sup>a</sup> This collective variable uses the minimum distance as described in table 2.

<sup>c</sup> This collective variable uses the number of hydrogen bonds as described in table 2.

<sup>d</sup> This collective variable uses the number of water mediated hydrogen bonds as described in table 2.

Table 4: **List of the relevant collective variables for the stable state definitions of S2**

CV	Description
d_gly12_gly60 <sup>a</sup>	The minimum distance between the heavy atoms of glycine 12 and the heavy atoms of glycine 60.
d_gly12_gln61 <sup>a,wt</sup>	The minimum distance between the heavy atoms of glycine 12 and the side-chain heavy atoms of glutamine 61.
d_gly12_leu61 <sup>a,Q61L</sup>	The minimum distance between the heavy atoms of glycine 12 and the side-chain heavy atoms of leucine 61.
d_GTP_glu62 <sup>a</sup>	The minimum distance between the C <sub>δ</sub> of glutamic acid 62 and the heavy atoms of GTP, including Mg <sup>2+</sup> .
d_GTP_glu63 <sup>a</sup>	The minimum distance between the C <sub>δ</sub> of glutamic acid 63 and the heavy atoms of GTP, including Mg <sup>2+</sup> .
d_cCOM_GTP_S2 <sup>a,b</sup>	The minimum distance between the circular mean center of mass of all atoms of residues 61 to 66 and the circular mean center of mass of all atoms of GTP, including Mg <sup>2+</sup> .
n_S2_α3 <sup>e</sup>	The number of combinations between the sets of {histidine 95, tyrosine 96, glutamine 99, arginine 102} and {{61} <sup>*</sup> , glutamic acid 62, glutamic acid 63, tyrosine 64} for which the minimal distance between the side-chain heavy atoms of the residue from the first set and all heavy atoms of the residue from the second set is smaller than 0.35 nm.

<sup>a</sup> This collective variable uses the minimum distance as described in table 2.

<sup>b</sup> This collective variable uses the circular mean center of mass as described in table 2.

<sup>e</sup> This collective variable uses the number of bonds as described in table 2.

<sup>wt</sup> Only used in wild-type KRas.

<sup>Q61L</sup> Only used in the Q61L mutant of KRas.

<sup>\*</sup> {61} is glutamine 61 for wild-type KRas and leucine 61 for the Q61L mutant of KRas.

These definitions apply to both WT and Q61L. The CV column shows the collective variable names, with their description in the Description column.

Table 5: List of the stable state definitions for KRas.

State	CV	Constraint(s)	Logic
S1-D33	d_GTP_asp33	$0.0 \leq x \leq 0.43$ nm	} $or^a$
	n_h_med_bonds_GTP_asp33	$1.9 \leq x \leq 5.0$ bonds	
	n_h_med_bonds_MG_asp33	$0.9 \leq x \leq 5.0$ bonds	
S1-30-32	d_GTP_asp30	$0.0 \leq x \leq 0.35$ nm	} $and^b$
	n_hbonds_GTP_asp30	$1.9 \leq x \leq 5.0$ bonds	
	d_GTP_glu31	$0.0 \leq x \leq 0.42$ nm	} $or^a$
	n_hbonds_GTP_tyr32	$0.9 \leq x \leq 4.0$ bonds	
	n_hbonds_tyr32_GTP	$0.9 \leq x \leq 4.0$ bonds	
	n_h_med_bonds_GTP_nnbound	$1.9 \leq x \leq 5.0$ bonds	
	n_h_med_bonds_MG_nnbound	$0.9 \leq x \leq 5.0$ bonds	
d_GTP_asp33	$1.1 \leq x \leq 5.0$ nm	} $and^b$	
S1-open	d_GTP_asp30	$1.05 \leq x \leq 5.0$ nm	} $and^b$
	d_GTP_glu31	$1.55 \leq x \leq 5.0$ nm	
	d_GTP_tyr32	$1.25 \leq x \leq 5.0$ nm	
	d_GTP_asp33	$1.25 \leq x \leq 5.0$ nm	
	n_hbonds_ile55_tyr40	$-0.1 \leq x \leq 0.1$ bonds	
	n_hbonds_GTP_S1	$-0.1 \leq x \leq 0.1$ bonds	
S2-GTP	d_gly12_gly60	$0.0 \leq x \leq 0.3$ nm	} $or^a$
	d_gly12_{61} <sup>*</sup>	$0.0 \leq x \leq 0.3$ nm	
	d_GTP_glu62	$0.0 \leq x \leq 0.65$ nm	
	d_GTP_glu63	$0.0 \leq x \leq 0.65$ nm	
	d_cCOM_GTP_S2	$0.0 \leq x \leq 1.6$ nm	
S2- $\alpha$ 3	n_S2_ $\alpha$ 3	$3.5 \leq x \leq 4.5$ bonds	} $and^b$
	d_cCOM_GTP_S2	$1.78 \leq x \leq 1.85$ nm	
S2-open	d_gly12_gly60	$0.6 \leq x \leq 5.0$ nm	} $and^b$
	d_gly12_{61} <sup>*</sup>	$0.8 \leq x \leq 5.0$ nm	
	d_GTP_glu62	$1.0 \leq x \leq 5.0$ nm	
	d_GTP_glu63	$1.0 \leq x \leq 5.0$ nm	
	d_cCOM_GTP_S2	$1.975 \leq x \leq 5.0$ nm	

\* {61} is glutamine 61 for the wild type and leucine 61 for the Q61L mutant. <sup>a</sup> One or more of the conditions must be true. <sup>b</sup> All conditions must be true.

The State column are the names of the stable states. Every stable state is build by combining the Constraints and Logic columns. For example in set notation the S2-GTP state corresponds to  $((\{x \mid d\_gly12\_gly60(x) \in [0.0, 0.3]\} \cup \{x \mid d\_gly12\_{61}(x) \in [0.0, 0.3]\} \cup \{x \mid d\_GTP\_glu62(x) \in [0.0, 0.65]\} \cup \{x \mid d\_GTP\_glu63(x) \in [0.0, 0.65]\}) \cap \{x \mid d\_cCOM\_GTP\_S2(x) \in [0.0, 1.6]\})$  in words this would be:  $((0.0 \leq d\_gly12\_gly60(x) \leq 0.35$  or  $0.0 \leq d\_gly12\_{61}(x) \leq 0.35$  or  $0.0 \leq d\_GTP\_glu62(x) \leq 0.65$  or  $0.0 \leq d\_GTP\_glu63(x) \leq 0.65$ ) and  $0.0 \leq d\_cCOM\_GTP\_S2(x) \leq 1.6$ )

**Appendix S2: Transitions as function of the Monte-Carlo (MC) steps** The figures listed in this appendix show the type of transition as sampled for each step in the TPS simulations.

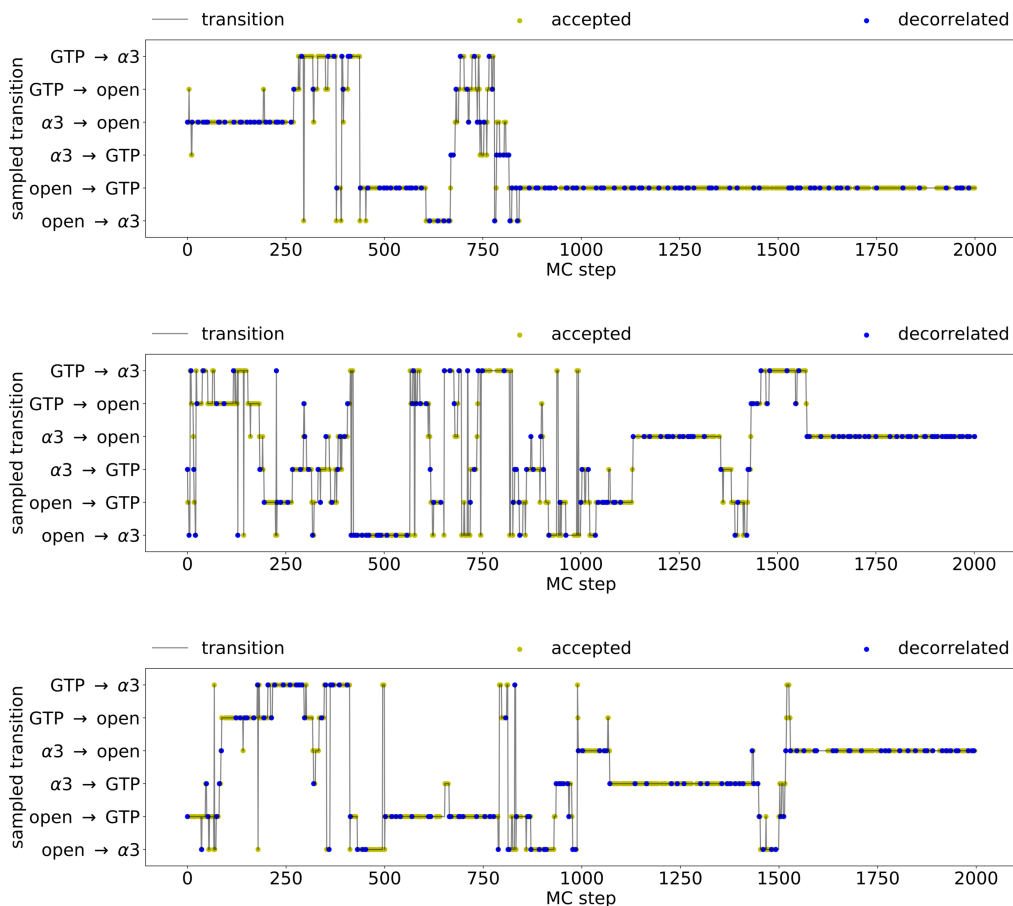


Figure 9: **Transitions as function of the Monte-Carlo step for the WT simulations.** The simulations started from (top) the S2- $\alpha 3$  to S2-open transition, (middle) the S2- $\alpha 3$  to S2-GTP transition and (bottom) the S2-open to S2-GTP transition. The x-axis shows the number of the MC steps. The y-axis shows the sampled transition. The y-axis lists all transitions that can occur for S2. The gray lines represents the trial moves, with the accepted MC steps highlighted as yellow dots and the accepted MC steps that lead to a new decorrelated trajectory with a blue dot.

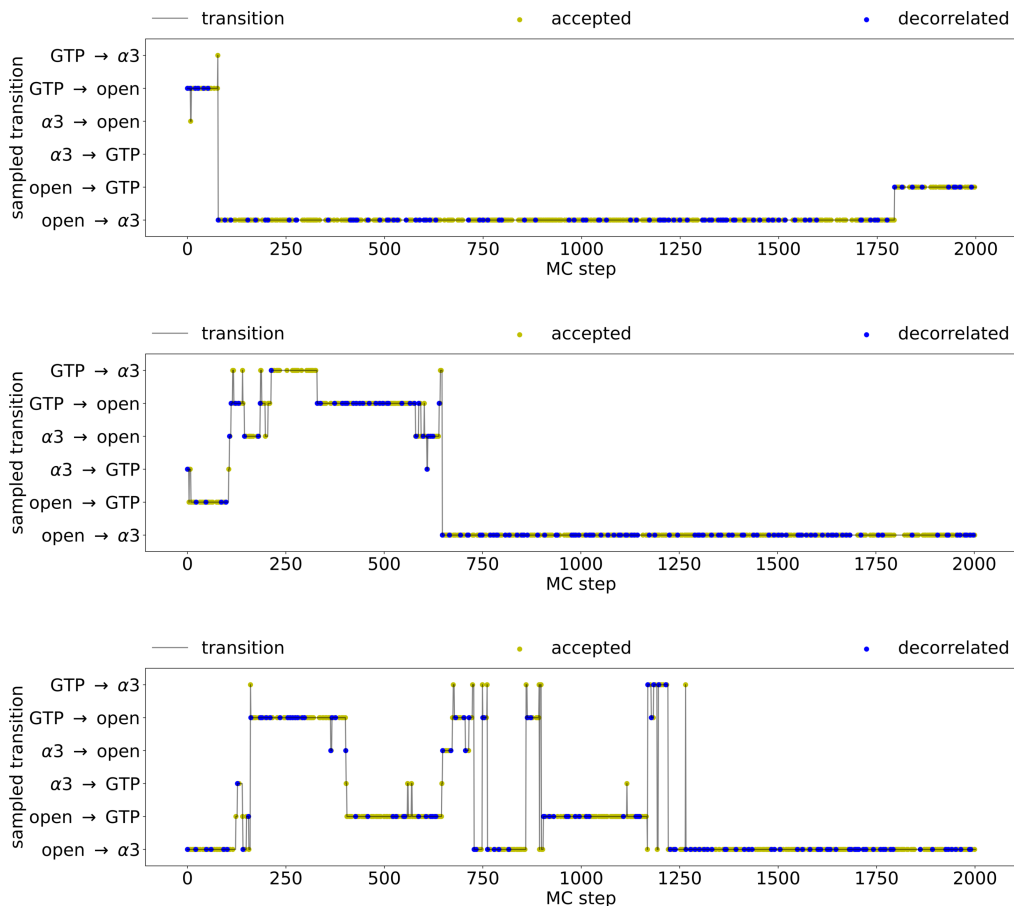


Figure 10: **Transitions as function of the Monte-Carlo step for the Q61L simulations.** The simulations started from (top) the S2-GTP to S2-open transition, (middle) the S2- $\alpha 3$  to S2-GTP transition and (bottom) the S2-open to S2- $\alpha 3$  transition. The x-axis shows the number of the MC steps. The y-axis shows the sampled transition. The y-axis lists all transitions that can occur for S2. The gray lines represents the trial moves, with the accepted MC steps highlighted as yellow dots and the accepted MC steps that lead to a new decorrelated trajectory with a blue dot.

**Appendix S3: MSTPS results for S2** The sampling statistics of the S2 MSTPS simulations are shown in table 1. The number of Monte Carlo (MC) trials was equal for all simulations. The acceptance is between 34 % and 42 %, which is reasonable considering the theoretical maximum of 67 %. This theoretical maximum is due to the fact that in our shooting algorithm only self transitions are forbidden. This leads to a maximum acceptance of  $\frac{(N-1)}{N}$ , for  $N$  number of states. With  $N = 3$  this leads to the theoretical maximum acceptance of 67% for this MSTPS study. The number of decorrelated trajectories is satisfactory for all simulations, and are spread well throughout the simulation as shown by the blue dots in the figures in Appendix . The average path length and total simulation time are only different for WT simulation 1. This simulation enters a different transition channel than the other simulations, which would explain these altered numbers.

The Least Changed Path (LCP) is the sequence of frames that, together, represent all accepted trajectories of a path sampling simulation. When running backwards in the simulation (from the last MC step to the first) a sequence of frames that are in between the shooting point of the latest trajectory and the shooting point of the trajectory in which that shooting point of the latest trajectory is replaced, on the last accepted trajectory before this next trajectory is added to the LCP. This is continued until the first MC step is reached. These LCPs represent the barrier region that is sampled during the TPS simulation.<sup>46</sup> Figure 11 (WT) and figure 12 (Q61L) show the LCPs for all simulations, projected on top of the combined pdhs from figure 6. The colouring is based on the first sampled transition of each frame of the LCP and is red for S2-GTP  $\leftrightarrow$  S2-open, blue for S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3, and yellow for S2- $\alpha$ 3  $\leftrightarrow$  S2-open. For the WT, all transitions sample the same diffuse barrier region, as indicated by the overlap of the clouds, which supports the hypothesis that the switching is also a diffusive process. For the extra channel for the S2-GTP $\leftrightarrow$ S2-open transition, this mostly occurs in simulation 1 of WT, but it is also observed in simulation 2 and 3. For the Q61L simulations, the LCP is more constrained to a value of under 1.3 nm for the S2- $\alpha$ 3-distance. Also, the clouds overlap less with each other, making switching more unlikely.



Simulation 1 and 3 of Q61L also show sampling of an extra S2- $\alpha$ 3  $\leftrightarrow$  S2-open channel, at values of 0.75 nm or lower for the S2- $\alpha$ 3-distance.

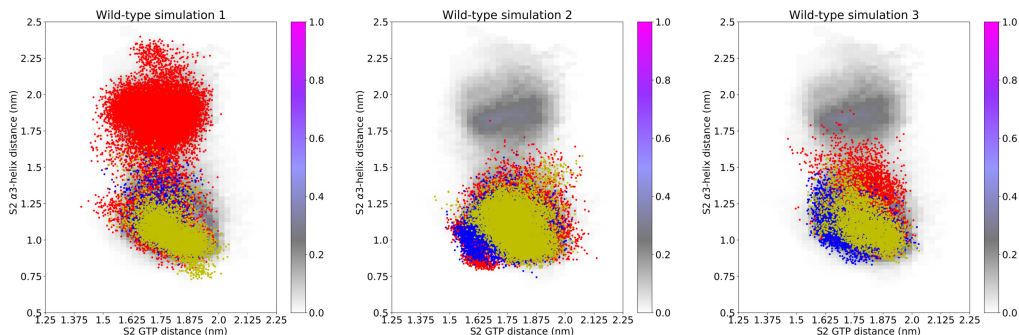


Figure 11: **The Least-Changed-Paths of the WT simulations.** The frames of the LCP of each WT simulation, shown on top of the combined path density histogram, as shown in (figure 6(top)). The color of each frame represents the first transition sampled by that frame, red for S2-GTP  $\leftrightarrow$  S2-open, blue for S2-GTP  $\leftrightarrow$  S2- $\alpha$ 3, and yellow for S2- $\alpha$ 3  $\leftrightarrow$  S2-open. The numbering of the simulations is in the order of figure 9.

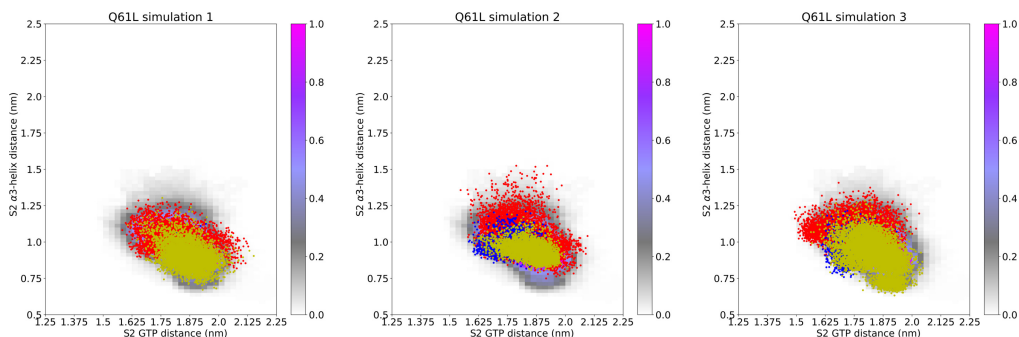
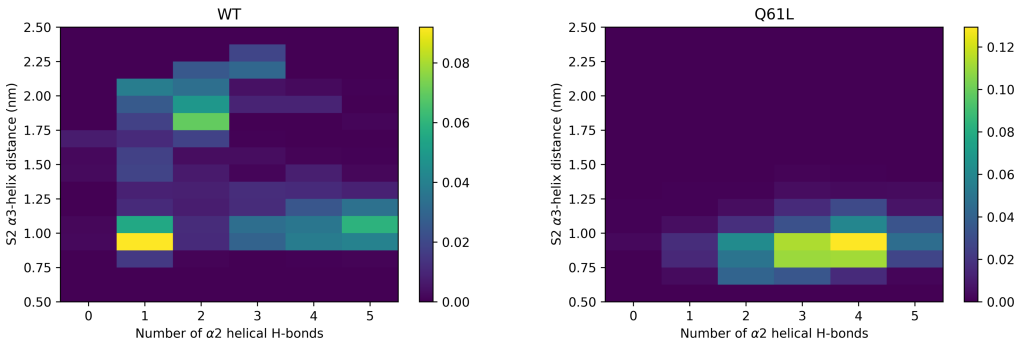


Figure 12: **The Least-Changed-Paths of the Q61L simulations.** The LCP are shown on top of the combined path density histogram (figure 6(top)). The color of each frame represents the first transition sampled by that frame, and is the same as in figure 11. The numbering of the simulations is in the order of figure 10.

Visual inspection of the transition paths as sampled for WT shows that in some paths helix  $\alpha$ 2 (residues 65-73) contained within S2, unfolds when entering the open state, but retains its shape in the S2-open state for the Q61L mutant. Two-dimensional probability histograms of the S2- $\alpha$ 3 distance and the number of helical hydrogen bonds of the  $\alpha$ 2-helix

(residues 65-73), for frames in the S2-open state, are shown in figure 13 for both WT and Q61L. Looking at the WT plot, there are two maxima for states in the reaction channel close to the  $\alpha$ 3-helix (under 1.5 nm on the y-axis), one where the  $\alpha$ 2-helix has all 5 helical H-bonds and one where it has only 1 helical H-bond. For the WT S2-open states away from the  $\alpha$ 3-helix (above 1.5 nm on the y-axis), the  $\alpha$ 2-helix has lost part of its helical structure, as indicated by a distribution around 2 helical H-bonds. When looking at the transition region between these two reaction channels at around 1.5 on the y-axis, helix  $\alpha$ 2 has lost most of its helical hydrogen bonds, which may indicate a correlation between the unfolding of helix  $\alpha$ 2 and the switching between the two reaction channels. The probability histogram for the S2-open frames of Q61L show a maximum at 4 helical H-bonds and S2 close to helix  $\alpha$ 3. These observations suggest that Q61L has a more structured open state.



**Figure 13: Two-dimensional probability histogram of the S2- $\alpha$ 3-helix distance and the number of helical hydrogen bonds in  $\alpha$ 2-helix for the S2-open state.** These are shown for (left) WT and (right) Q61L. The y-axis is the cCOM of S2 to the  $\alpha$ 3-helix (as used in figure 6). The x-axis is the number of hydrogen bonds (as described in table 2) between the backbone O of residue  $i$  and the backbone NH of residue  $i + 4$  for  $i \in [65, 69]$ . The colors indicate the probability.

**Appendix S4: MSTPS results for S1** The transitions as function of the MC trials of the S1 simulations are shown in figure 14. The x-axes represent the number of the MC trials, while the y-axis shows the sampled transition. Like in the supplements of figure 3, the y-axis lists the transitions, ordered such that the simulation can only switch to the transitions

directly above or below the current transition, or between the top and bottom transition. This ordering is possible, because only either the initial or the final state can change per MC step due to the one-way shooting algorithm. With a forward shot the final state can change and a backward shot may change the initial state. The trial moves are shown as a gray line, with the accepted MC steps highlighted as yellow dots and the accepted MC steps that result in a new decorrelated trajectory with a blue dot. The accepted and decorrelating MC steps are distributed well throughout both simulations. The number of switches that occur between the transitions is similar for both WT and Q61L. Both simulations spend a significant amount of simulation steps in the 30-32  $\rightarrow$  open transition.

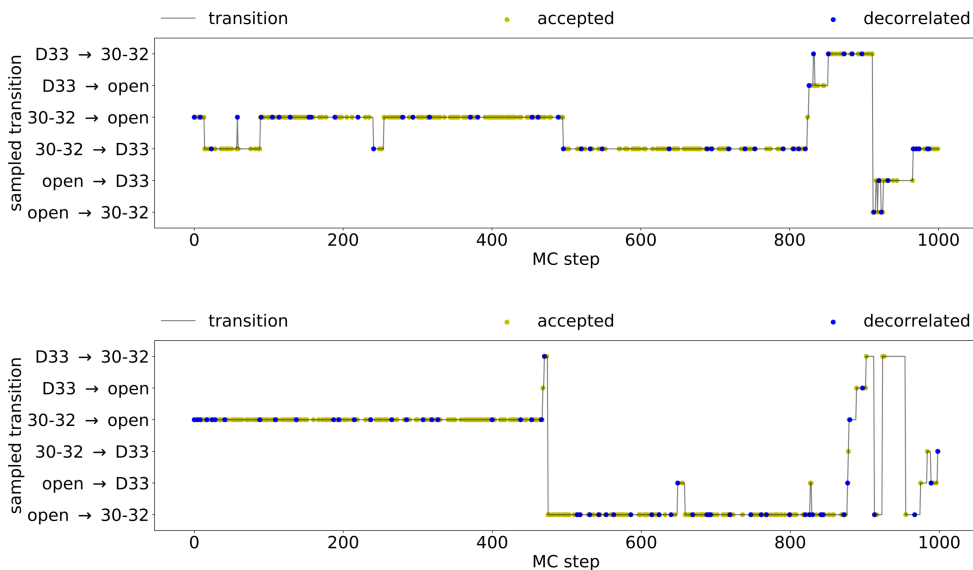


Figure 14: **Transitions as function of the Monte-Carlo step for the S1 simulations.** (top) WT (bottom) Q61L. The same axis setup and labeling is used as in the supplements for figures 9 and 10. Here D33 corresponds to the S1-D33 state, 30-32 to the S1-30-32 state, and open to the S1-open state.

The sampling and switching behaviour of both simulations is summarized in figure 15.

This could be attributed to a lack of convergence of the simulations. The number of switches that occur between the transitions is similar for both the WT and the Q61L simu-

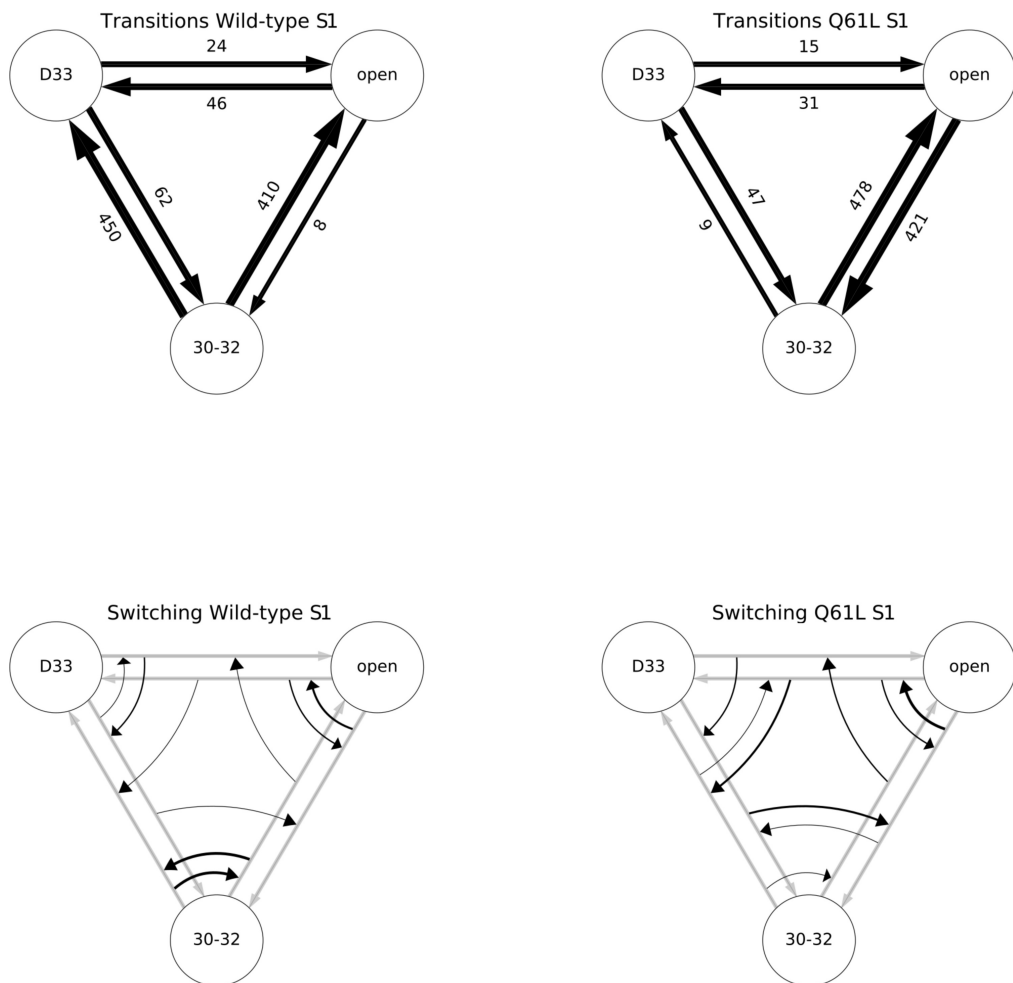


Figure 15: **Results of the S1 path sampling simulations.** (left) Transitions in the (top, left) WT and (bottom, left) Q61L simulations. (right) Switches in the (top, right) WT and (bottom, right) Q61L simulations. The same labeling for the states is used as in figure 14.

lation.

Representative trajectories of all sampled transitions of both the WT and the Q61L were visually compared. No distinct differences in transition mechanisms between WT and the Q61L were observed. In conclusion, these results suggest that the mutation in S2 has little effect on the dynamical behavior of S1.

**S5 Video** **Movie of a typical trajectory of the S2-GTP and the S2-open transition for WT.** The frames in these movies are rendered with the switch regions highlighted in green for S1 and blue for S2. The  $\alpha$ 3-helix is highlighted in yellow. The protein is shown as a ribbon with an transparent stick representation for the amino acids in S1 and S2. GTP is shown as solid sticks, with carbon atoms colored in green, oxygen in red, nitrogen in blue and phosphorus in orange.  $Mg^{2+}$  is shown as a green ball.

**S6 Video** **Movie of a typical trajectory of the S2-GTP and the S2-open transition for Q61L.** The frames in these movies are rendered with the switch regions highlighted in green for S1 and blue for S2. The  $\alpha$ 3-helix is highlighted in yellow. The protein is shown as a ribbon with an transparent stick representation for the amino acids in S1 and S2. GTP is shown as solid sticks, with carbon atoms colored in green, oxygen in red, nitrogen in blue and phosphorus in orange.  $Mg^{2+}$  is shown as a green ball.

## References

- (1) Downward, J. Targeting RAS signalling pathways in cancer therapy. *Nature Reviews Cancer* **2003**, *3*, 11–22.
- (2) Lau, K. S.; Haigis, K. M. Non-redundancy within the RAS oncogene family: Insights into mutational disparities in cancer. *Molecules and Cells* **2009**, *28*, 315–320.
- (3) Prior, I. A.; Lewis, P. D.; Mattos, C. A Comprehensive Survey of Ras Mutations in Cancer. *Cancer Research* **2012**, *72*, 2457–2467.
- (4) Pacold, M.; Suire, S.; Perisic, O.; Lara-Gonzalez, S.; Davis, C.; Walker, E.; P.T., H.; Stephens, L.; Eccleston, J.; Williams, R. Crystal Structure and Functional Analysis of Ras Binding to Its Effector Phosphoinositide-3-Kinase- $\gamma$ . *Cell* **2000**, *103*, 931–944.
- (5) Boriack-Sjodin, P.; Margarit, S.; Bar-Sagi, D.; Kuriyan, J. The structural basis of the activation of Ras by Sos. *Nature* **1998**, *394*, 337.
- (6) Spoerner, M.; Hermann, C.; Vetter, I.; Kalblitzer, H.; Wittinghofer, A. Dynamic properties of the Ras switch I region and its importance for binding to effectors. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 4944.
- (7) Kobayashi, C.; Saito, S. Relation between the conformational heterogeneity and reaction cycle of Ras: molecular simulation of Ras. *Biophys. J.* **2010**, *99*, 3726.
- (8) Spoerner, M.; Hozsa, C.; Poetzl, J.; Reiss, P., K. and Ganser; Geyer, M.; Kalblitzer, H. Conformational states of human rat sarcoma (Ras) protein complexed with its natural ligand GTP and their role for effector interaction and GTP hydrolysis. *J. Biol. Chem.* **2010**, *285*, 39768.
- (9) Geyer, M.; Schweins, T.; Herrmann, C.; Prisner, T.; Wittinghofer, A.; Kalbitzer, H. R. Conformational Transitions in p21ras and in Its Complexes with the Effector Protein

- Raf-RBD and the GTPase Activating Protein GAP†. *Biochemistry* **1996**, *35*, 10308–10320.
- (10) Hobbs, G. A.; Der, C. J.; Rossman, K. L. RAS isoforms and mutations in cancer at a glance. *Journal of Cell Science* **2016**, *129*, 1287–1292.
- (11) Dharmiah, S.; Tran, T. H.; Messing, S.; Agamasu, C.; Gillette, W. K.; Yan, W.; Waybright, T.; Alexander, P.; Esposito, D.; Nissley, D. V.; McCormick, F.; Stephen, A. G.; Simanshu, D. K. Structures of N-terminally processed KRAS provide insight into the role of N-acetylation. *Scientific Reports* **2019**, *9*.
- (12) Chuang, H.-C.; Huang, P.-H.; Kulp, S.; Chen, C.-S. Pharmacological strategies to target oncogenic KRAS signaling in pancreatic cancer. *Pharmacological Research* **2017**, *117*, 370–376.
- (13) Meng, M.; Zhong, K.; Jiang, T.; Liu, Z.; Kwan, H.; Su, T. The current understanding on the impact of KRAS on colorectal cancer. *Biomedicine & Pharmacotherapy* **2021**, *140*, 111717.
- (14) Prakash, P.; Gorfe, A. A. Overview of simulation studies on the enzymatic activity and conformational dynamics of the GTPase Ras. *Molecular Simulation* **2014**, *40*, 839–847.
- (15) Dellago, C.; Bolhuis, P. G.; Csajka, F. S.; Chandler, D. Transition path sampling and the calculation of rate constants. *The Journal of Chemical Physics* **1998**, *108*, 1964–1977.
- (16) Rogal, J.; Bolhuis, P. G. Multiple state transition path sampling. *J. Chem. Phys.* **2008**, *129*, 224107.
- (17) Hunter, J.; Manadhar, A.; Carrasco, M.; Gurbani, D.; Gondi, S.; Westover, K. Biochemical and Structural Analysis of Common Cancer-Associated KRAS Mutations. *Mol. Cancer Res.* **2015**, *9*, 1325.

- (18) Buhrman, G.; Kumar, V.; Cirit, M.; Haugh, J.; Mattos, C. Allosteric modulation of Ras-GTP is linked to signal transduction through RAF kinase. *J. Biol. Chem.* **2011**, *286*, 3323.
- (19) Fetics, S.; Guterres, H.; Kearney, B.; Buhrman, G.; Ma, B.; Nussinov, R.; Mattos, C. Allosteric effects of the oncogenic RasQ61L mutant on Raf-RBD. *Structure* **2015**, *23*, 505–516.
- (20) Kyte, J.; Doolittle, R. F. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology* **1982**, *157*, 105 – 132.
- (21) Muraoka, S.; Shima, F.; Araki, M.; Inoue, T.; Yoshimoto, A.; Ijiri, Y.; Seki, N.; Tamura, A.; Kumasaka, T.; Yamamoto, M.; Kataoka, T. Crystal structures of the state 1 conformations of the GTP-bound H-Ras protein and its oncogenic G12V and Q61L mutants. *FEBS Letters* **2012**, *586*, 1715–1718.
- (22) Muraoka, S.; Shima, F.; Araki, M.; Inoue, T.; Yoshimoto, A.; Ijiri, Y.; Seki, N.; Tamura, A.; Kumasaka, T.; Yamamoto, M.; Kataoka, T. Crystal structure of H-Ras WT in complex with GppNHp (state 1). 2012; <http://dx.doi.org/10.2210/pdb4ef1/pdb>.
- (23) Bolhuis, P. G.; Chandler, D.; Dellago, C.; Geissler, P. L. TRANSITION PATH SAMPLING: Throwing Ropes Over Rough Mountain Passes, in the Dark. *Annual Review of Physical Chemistry* **2002**, *53*, 291–318.
- (24) Stelzl, L. S.; Hummer, G. Kinetics from Replica Exchange Molecular Dynamics Simulations. *Journal of Chemical Theory and Computation* **2017**, *13*, 3927–3935.
- (25) Webb, B.; Sali, A. *Current Protocols in Bioinformatics*; John Wiley & Sons, Inc., 2002; pp 5.6.1–5.6.37.
- (26) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L.

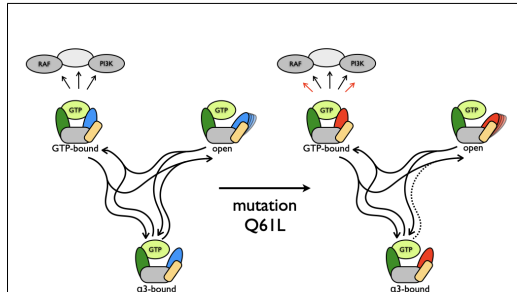


- Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics* **1983**, *79*, 926.
- (27) Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins: Structure, Function, and Bioinformatics* **2010**, *78*, 1950–1958.
- (28) Meagher, K. L.; Redman, L. T.; Carlson, H. A. Development of polyphosphate parameters for use with the AMBER force field. *Journal of Computational Chemistry* **2003**, *24*, 1016–1025.
- (29) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *The Journal of Chemical Physics* **1995**, *103*, 8577.
- (30) Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435–447.
- (31) Bussi, G.; Donadio, D.; Parrinello, M. Canonical sampling through velocity rescaling. *The Journal of Chemical Physics* **2007**, *126*, 014101.
- (32) Parrinello, M.; Rahman, A. Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.* **1981**, *52*, 7182.
- (33) Hess, B. P-LINCS: A Parallel Linear Constraint Solver for Molecular Simulation. *Journal of Chemical Theory and Computation* **2008**, *4*, 116–122.
- (34) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology* **2017**, *13*, 1–17.

- (35) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of Computational Physics* **1977**, *23*, 327 – 341.
- (36) Sivak, D. A.; Chodera, J. D.; Crooks, G. E. Time Step Rescaling Recovers Continuous-Time Dynamical Properties for Discrete-Time Langevin Integration of Nonequilibrium Systems. *The Journal of Physical Chemistry B* **2014**, *118*, 6466–6474.
- (37) Chodera, J.; Rizzi, A.; Naden, L.; Beauchamp, K.; Grinaway, P.; Fass, J.; Rustenburg, B.; Ross, G.; Simmonett, A.; Swenson, D. choderalab/openmmtools: 0.14.0 - Exact treatment of alchemical PME electrostatics, water cluster test system, optimizations. 2018; <https://doi.org/10.5281/zenodo.1161149>.
- (38) Åqvist, J.; Wennerström, P.; Nervall, M.; Bjelic, S.; Brandsdal, B. O. Molecular dynamics simulations of water and biomolecules with a Monte Carlo constant pressure algorithm. *Chemical Physics Letters* **2004**, *384*, 288 – 294.
- (39) Humphrey, W.; Dalke, A.; Schulten, K. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics* **1996**, *14*, 33–38.
- (40) Swenson, D. W. H.; Prinz, J.-H.; Noe, F.; Chodera, J. D.; Bolhuis, P. G. OpenPathSampling: A Python Framework for Path Sampling Simulations. 1. Basics. *Journal of Chemical Theory and Computation* **2019**, *15*, 813–836, PMID: 30336030.
- (41) Swenson, D. W. H.; Prinz, J.-H.; Noe, F.; Chodera, J. D.; Bolhuis, P. G. OpenPathSampling: A Python Framework for Path Sampling Simulations. 2. Building and Customizing Path Ensembles and Sample Schemes. *Journal of Chemical Theory and Computation* **2019**, *15*, 837–856.
- (42) Bolhuis, P. G. Transition path sampling on diffusive barriers. *Journal of Physics: Condensed Matter* **2003**, *15*, S113.

- (43) Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* **2007**, *9*, 90–95.
- (44) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MD-Traj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **2015**, *109*, 1528 – 1532.
- (45) Bai, L.; Breen, D. Calculating Center of Mass in an Unbounded 2D Environment. *Journal of Graphics Tools* **2008**, *13*, 53–60.
- (46) Juraszek, J.; Vreede, J.; Bolhuis, P. G. Transition path sampling of protein conformational changes. *Chemical Physics* **2012**, *396*, 30–44.

## Graphical TOC Entry



Schematic overview of the effect of the Q61L mutation on the dynamics of KRAs. KRas and its mutant Q61V show similar conformational transitions, at different frequencies.

ISBN 978-82-326-6856-4 (printed ver.)  
ISBN 978-82-326-6406-1 (electronic ver.)  
ISSN 1503-8181 (printed ver.)  
ISSN 2703-8084 (online ver.)



**NTNU**

Norwegian University of  
Science and Technology