ISBN 978-82-326-2188-0 (printed ver.)
ISBN 978-82-326-2189-7 (electronic ver.)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Vinay Kumar Gautam

Doctoral theses at NTNU, 2017:58

NTNU

Doctoral thesis

NTNU
Norwegian University of
Science and Technology
Thesis for the degree of
Philosophiae Doctor
Faculty of Information Technology and
Electrical Engineering
Department of Computer Science

Doctoral theses at NTNU, 2017:58

Vinay Kumar Gautam

Dynamically Controlled DNA Tiles

Looking Beyond the Self-assembly of Static Two-dimensional DNA-based Computational Structures

NTNU
Norwegian University of
Science and Technology

Vinay Kumar Gautam

Dynamically Controlled DNA Tiles

Looking Beyond the Self-assembly of Static Two-dimensional DNA-based Computational Structures

Thesis for the degree of Philosophiae Doctor

Trondheim, March 2017

Norwegian University of Science and Technology
Faculty of Information Technology and
Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology and
Electrical Engineering
Department of Computer Science

ISBN 978-82-326-2188-0 (printed ver.)
ISBN 978-82-326-2189-7 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2017:58

Printed by NTNU Grafisk senter

To my late father, Mr. J. P. Gautam

Abstract

Algorithmic self-assembly of DNA tiles enables computation and construction of DNA structures at the nanoscale, where the assembly process is guided by specific DNA interactions between complementary sticky-ends of tiles. Tile assembly was originally proposed as a unidirectional (passive) process, where tiles once assembled could not be disassembled and reassembled reliably and therefore, limiting its scope for designing complex, cheap and practically useful computational DNA systems.

A dynamically controlled tile assembly process would not only help in solving the limitations imposed by passive nature of tile assembly, but would also open up new avenues in biologically-inspired nanosystem design. However, enabling dynamic control in the self-assembly of DNA tiles — an attribute ubiquitous in the self-assembly happening at various scales in the living world — is a technical challenge.

This thesis addresses closely related issues of reliable and inexpensive assembly of tiles by combining DNA tiles and DNA strand displacement based dynamically controlled switching mechanisms. Building on the existing building blocks of the tile self-assembly, (self)-switchable and (externally)-switchable DNA tile structures are proposed.

Self-switchable tiles include Double Crossover DNA molecular structures and especially designed protection elements, together forming protected tiles termed Enveloped Tiles. Integrated protection elements of the Enveloped Tiles protect them from illegitimate binding as the tiles self-assemble cooperatively. A successful docking of correctly matching Enveloped Tile, which is favoured against the docking of a mismatched tile, releases its protection using an intramolecular DNA displacement process. The feasibility of the DNA molecular structure of Enveloped Tiles is established using DNA sequence design and thermodynamics calculations. The process of releasing protection of Enveloped Tile is studied using Coarse-grained molecular dynamics simulations. A kinetic modeling of the Enveloped Tile assembly process quantitatively demonstrates the error prevention feature. This work thus provides a theory-to-practice approach for a reliable construction of algorithmically self-assembling DNA tile systems.

Externally-switchable tiles are built around Double Crossover DNA molecular structures by adding a short DNA sequence with their sticky ends. The additional DNA sequences serve as switching points for the self-assembled tile structures that can be dissociated dynamically by a inhibitor DNA signal, which is generated by an integrated DNA reaction system. Using these tiles, therefore, hierarchical assembly pathways involving several steps of assembly, disassembly and reassembly can be designed. In this framework, we first introduce conceptual design of a minimal system for self-replication of algorithmically programmed DNA tile patterns and then study its behaviour using both the abstract and kinetic models of tile assembly. Simulations of the kinetic model, which capture the realism of the DNA tile assembly process, show that the tile pattern self-replication occurs reliably for a set of physicochemical parameters that are typically used in the self-assembly of tile patterns.

These switchable tile designs are simple, yet capable of assembling reliable and complex nanostructures without use of biological enzymes. Thus, in a more general view, these dynamically controlled self-assembly processes are useful for a wider class of applications in algorithmically programmable tile self-assembly systems.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfillment of the requirements for the degree of philosophiae doctor.

Acknowledgements

The work presented in this thesis was carried out at the CRAB (Complex, Robust, Adaptive, Bio-inspired solutions) lab in the Intelligent Systems Group of Computer and Information Science Department, NTNU, Trondheim, Norway, during the years 2010-2015.

I recognize that this research would not have been possible without the financial assistance of Computer and Information Science Department, IME faculty, The Norwegian University of Science and Technology.

First and foremost, I would like to express sincere gratitude to my PhD advisors: Prof. Keith L. Downing, Prof. Pauline C. Haddow, and Prof. Martin Kuiper. Firstly, I would like to thank Prof. Keith Downing for his support and guidance in the latter phase of the PhD. To-the-point advice from Keith has been invaluable for connecting the dots of this challenging research and writing a coherent thesis. I am grateful to Prof. Pauline C. Haddow for supporting my PhD fellowship application, helping me to feel comfortable in this new place during early days of the PhD. Her guidance in deciding the research topic, formulation of the research questions, and scientific writing have been essential for this thesis. A very special thanks goes out to Prof. Martin Kuiper. He has been very kind to help me in this challenging research. Without his understanding in this field and kindness to answer questions with patience, I could not have achieved my research objectives. Discussions with him have been instilling, and his scientific acumen for the details will always be remembered.

I deeply thank Dr. Eugen Czeizler (post-doctoral fellow in the department of Computer and Information Science, Aalto University, Finland (now working as senior researcher in Åbo Akademi University, Finland)) for discussions, advice and helping in self-replicator design. I thank Dr. Flavio Romano (post-doctoral fellow in the theoretical chemistry Group at the department of Chemistry, University of Oxford, UK) for helping me to understand the coarse grain molecular dynamics simulator (oxDNA).

I would like to thank Prof. Agnar Aamodt, Prof. Pinar Øzturk, Prof. Björn Gambäck, Prof. Helge Langseth, and Prof. Anders Kofod-Petersen, for illuminating the Intelligent Systems Group. I express my thanks to the entire Intelligent Systems Group for enabling conducive research environment. I also thank to my office neighbors, the Teknisk Gruppe (TG), who always gave me a sense of confidence that no matter what happens with my computer, they will fix it.

I am extremely grateful to my parents for their selfless love, unconditional support and motivation. I dedicate this PhD work to my beloved father, who is no more alive, but his lessons of perseverance, simplicity, and all-is-well in the end have always been source of inspiration for me.

I thank my beloved wife Shweta Tiwari for her unconditional love, patience and encouragement especially in the latter phase of PhD when get going was tough. I thank my friends Gleb Sizov, Kai Olav Ellefsen, Lester Solbakken, Lars Bungum, Vegard Edvardsen, and Biswanath Barik for giving me company in the department.

Contents

List of Tables

List of Figures

List of Abbreviations

DNA	Deoxyribonucleic acid
DX	Double Crossover
DAO	Double Crossover and Anti-parallel with Odd spacing
TX	Triple Crossover
BT	Base Tile
PT	Protector Tile
ET	Enveloped Tiles
nt	nucleotide
bp	base pair
ss-DNA	single stranded-DNA
ds-DNA	double stranded-DNA
aTAM	abstract Tile Assembly Model
kTAM	kinetic Tile Assembly Model
CA	Cellular Automata
MD	Molecular Dynamics
VMMC	Virtual Move Monte Carlo
US	Umbrella Sampling
KDR	Kinetic Discrimination Ratio
BSD	Bimolecular Strand Displacement
USD	Unimolecular Strand Displacement
SWET	Switching Enabled Tile
CRN	Chemical Reaction Network
CST	Corner Super Tile
PTS	Pattern forming Tile Set
MTS	Mold forming Tile Set
TPSS	Tile Pattern Self-replication Simulator
aTPSM	abstract Tile Pattern Self-replication Model
kTPSM	kinetic Tile Pattern Self-replication Model

1 Introduction

1.1 Motivation

The conventional computing approach is based on logical and mathematical, abstraction, where the physical substrate used for the realization of computation model is merely an implementation detail. Therefore, computation becomes independent of the physics of the underlying substrate. The ease of implementation, universality (i.e., a Turing machine can simulate every computational model) and programmability have made conventional computing very successful. Computation, however, is not limited to the conventional approach, for example many people see information processing in natural systems as computation [142]. In natural information processing systems, a computation process is embodied in its substrate, continuously interacting with its environment, adapting and evolving.

The conventional computing approach provides multipurpose and easily programmable computing machines. However, it is quite inefficient in terms of use of space, time and energy. In contrast, biological systems are highly specialised and lacking in general control but can evolve, adapt and develop very complex organisms. From an information processing perspective, these two approaches seem mutually exclusive, as pointed out by Michael Conrad [31], "a system cannot at same time be effectively programmable, amenable to evolution by variation and selection, and computationally efficient". Although conventional computing has been an exceptionally successful paradigm, there are application areas where we may not need a powerful and programmable computer. For example, a drug delivery system may require an “unconventional computer" having only few molecules that can efficiently release a drug after sensing the local environment of a cell [17].

There have been growing interests in applying the computational perspective of natural systems to perform computations that allows harnessing underlying characteristics of the physical substrates [10, 154, 8, 87, 9]. Unconventional or Non-standard (Non-classical) computing [143] is an emerging interdisciplinary research area, which explores novel methods of computing and information processing that go beyond the conventional model, and encompasses physical, chemical and biological systems and computing architectures. Some of the major endeavours to unconventional computing using physical substrates are: DNA-based Computing [10, 159, 154], amorphous computing [8], Evolution-in-Materio [87, 62], Bacterial Computing [12], or Slime Mold Computing [9]. This paradigm has the potential to revolutionize not only our fundamental understanding of the nature of computation, but also the way in which we seek solutions to fundamental problems, manipulate material at the molecular scale, perform fabrication across spatial scales, and develop drugs.

DNA-based computing started with the seminal work of Adleman in which an unconventional approach based on synthetic DNA molecules was used to solve a Traveling Salesman Problem (TSP) [10]. Adleman's approach uses the enormous information carrying capacity of DNA molecules to represent potential solutions of a TSP. Further, these DNA molecules are tested using biochemistry techniques to find out which of these solutions satisfy the criteria of being an actual solution of the TSP. Although it is a tedious task to perform these biochemistry steps in a laboratory, parallelism of DNA operations enables a large number of solutions which are tested simultaneously, making it an interesting concept from the information processing perspective. It was later found that the parallelism in the process comes at a cost of restricted flexibility in programming these DNA molecules.

Erik Winfree used DNA tiles [157] as physical analogues of Wang tiles [149], and demonstrated that self-assembly of DNA tiles can be used to introduce programming flexibility in DNA-based computing [154]. It has been shown that DNA tile self-assembly can simulate the dynamics of bounded one-dimensional cellular automaton [112] and therefore is capable of performing Turing universal computations [159, 70]. A variety of computational lattices [112, 78, 14, 21] of both *Double Crossover(DX)* [157] tiles and *Triple Crossover* [78] tiles have been designed and successfully assembled experimentally in the laboratory. Further, a design of a transducer [21], which simulates a finite state automaton, has been demonstrated in the framework of DNA tile self-assembly. The transducer design opens up possibilities for the implementation of computational tile lattices with programmed inputs. Further, complex and generalized computing machines with several iterative finite state automata can be implemented using such transducers.

Self-assembled structures of DNA tiles are typically produced by a thermal annealing, and therefore the structure is at its thermodynamic minimum energy and static in nature. There is no local control over assembling tiles in such thermodynamics driven tile assembly processes. To enable a sequence-specific and dynamic local control in the DNA assembly process, Yurke et al. [168] demonstrated toehold-mediated DNA Strand Displacement [173]. The toehold-mediated strand displacement has been widely used in the design of dynamic DNA systems, such as molecular motors [168, 16], walkers [133, 166], DNA logic circuits [129, 106, 105], and enzyme-free catalytic systems [129, 173, 174, 165, 175]. A review article, Zhang et al. [170], presents detailed state-of-the-art work in dynamic DNA nanosystems.

The connection between DNA self-assembly, computation and dynamic control at the nanoscale, therefore, opens up a powerful paradigm to develop new computation and material manipulation methodologies by exploiting the physico-chemical characteristics of the information rich DNA substrate. A better understanding of the intertwined nature of information and matter at the nanoscale would enable not only the engineering of robust, inexpensive and novel nanosystems, but would also help us in understanding the ways in which information is created and used to guide increasing levels of complexity in nature.

1.2 DNA-based Computing and Nanotechnology

DNA-based engineering started around four decades ago with the pioneering work of Ned Seeman [130]. The approach has emerged as a promising technique to implement DNA-based molecular computation [10, 159, 154, 158, 109, 70] and to construct two-dimensional arrays [154, 112, 73, 14, 21] and three-dimensional structures [25, 176, 56]. In addition to these static self-assembled DNA-based nanostructures, there have also been parallel developments in the field of DNA-based dynamic systems design [168, 173, 174, 172, 105, 129, 165], which started with the introduction of the DNA Strand Displacement mechanism by Yurke et el. [168].

Self-assembly of DNA tiles [157], which is based on the abstract mathematical computation model of tiling [149], has been used to assemble computational lattices of tiles [112, 14]. It has been suggested that the DNA-tile self-assembly is capable of simulating complex and powerful computation process, i.e., a Turing universal computing machine [159]. Further, Jonoska et al. [70] reported the design of a transducer, a finite state automaton model, that can be used to simulate a universal computer in the DNA-tile self-assembly framework. Such a transducer design with a one-dimensional programmable input has been experimentally demonstrated in Chakrborty et el. [21]. Benenson et al. [17] also reported a finite state automaton using duplex DNA molecules and enzyme molecules.

Self-assembly of DNA is guided by weak and reversible, but specific local interactions that are governed by topology and hydrogen bonds between complementary DNA sequences. DNA as a substrate has well-known mechanical and physico-chemical properties that can be abstracted away to design realistic models of DNA self-assembly processes enabling scalability [172]. Algorithmic self-assembly of DNA tiles [157] has been used to implement two-dimensional lattices using a set of designed tiles with DNA sticky ends that guide the local interactions between tiles. The self-assembly starts growing the tile lattice from a supplied seed structure and thus, the equilibrium structure assembles out of cooperative binding of the tiles. In static DNA self-assembly, the ultimate goal is to assemble target structures; there is no local DNA sequence-specific control, except the influence of physico-chemical factors (e.g., temperature and concentration), over local interactions between individual components forming static structures.

On the other hand, dynamic systems of DNA have been designed using toehold-mediated strand displacement reactions [168, 174]. In a strand-displacement reaction, complementary single-stranded DNA domains (toeholds) co-localize separate DNA molecules to facilitate branch migration processes, which can reconfigure the molecule or release a previously attached single-stranded species into solution. The dissociated single-stranded species can then engage in further strand displacement reactions, thus enabling the implementation of cascaded DNA reaction systems [172, 140, 105].

The above two approaches have largely been applied separately for the design of either self-assembled DNA structures or dynamic control behaviours using DNA molecules. In the self-assembled DNA structures, such as DNA tile lattices, sticky ends of the tiles are predesigned and they follow the programmed route to assemble the static structure; there is no local control between the interacting tiles, once they are set to assemble. Further, the tiles can not be reused or reconfigured post-assembly. Therefore, self-assembled DNA structures have no local control over the interacting DNA molecules, and are typically non-reactive to its immediate environment. The toehold-mediated dynamic systems [168, 174] of DNA have separately been used to design a variety of control applications, which involve repeated reconfiguration of DNA molecules, dynamically controlling local interactions among the molecules and their environment.

Together these approaches can enable a dynamically controlled DNA-based framework [171, 71, 114, 119, 121], where local interactions between DNA molecules can be controlled so as to precisely and robustly self-assemble programmable DNA structures. Such framework would also enable a dynamic process of computation with programmed local and global interactions of DNA molecules, which could produce continuous structural changes by reconfiguration and reuse of DNA-based building blocks.

This thesis focuses on the design of dynamically controlled DNA assembly mechanisms that integrate the DNA tile self-assembly and DNA-toehold strand displacement. Particularly, the following two closely related issues have been addressed using such an approach.

In algorithmic self-assembly of DNA tiles, a set of DNA tiles are designed to serve as a “program” [121] for the assembly of a pattern. The basic mechanism underlying algorithmic self-assembly [112] of tiles is cooperative binding between locally interacting tiles: complementary sticky ends of tiles bind cooperatively to form a thermodynamically stable attachment. Typically, a tile attaching by at least two sticky ends is considered thermodynamically favourable, that is, it does not fall off and gets buried inside the growing aggregate. However, a tile that attaches by fewer than two sticky ends is thermodynamically less favourable, and therefore it may fall off from the engaged site of the aggregate. A deadlock situation arises when an unfavourably attached tile fails to fall off before a second tile attaches and makes it thermodynamically stable, and thus a mismatched tile gets fixed. The rate of such assembly errors is too high for production of practically useful patterns using algorithmic self-assembly.

To address this issue, we present the design of Enveloped Tile (ET) by integrating switchable protection elements with the basic building blocks of double crossover (DX) DNA tiles (Base Tiles*(BTs)*). The switchable protection elements, termed Protector Tiles *(PTs)* bind with the input and output sticky ends of *BTs*, and thus, *ETs* stay deactivated as there are no exposed DNA domains, except short toeholds on the input sticky ends of *BTs*, available for binding. However, during self-assembly as an *ET* binds at a vacant site of the growing tile lattice, triggered by a correct matching at both its inputs, the PT is released and the *BT* binds stably. But, in case of a mismatched binding at either of the inputs, the *ET* is rejected from the site after an initial 'proofreading' engagement. Thus, the *ET* enables a reliable self-assembly process by suppressing assembly errors.

Second, we introduce an externally switchable mechanism in the tile assembly framework so as to dynamically assemble and disassemble the tile structures. In the externally switchable tile assembly framework, we design a minimal system of self-replication of 2-D rectangular patterns of tiles. The self-replicator design adheres to simplicity and implementation feasibility in four aspects: 1) DX tiles are used; 2) all glues are of strength 1; 3) tiles do not carry signals; 4) the replication process is enzyme free. Pattern replication starts with formation of a mold structure around the “L”-shaped seed with the help of a set of SWitch-Enabled Tiles (SWET) that can be activated to switch their binding state from bound (ON) to free (OFF). Further, the assembled mold gets dissociated from the seed structure by a toehold-mediated switching control, which is cyclically triggered at precise time intervals. The dissociated mold structure grows a new copy of the “L”-shaped seed while the dissociated seed structure reiterates the process. The remaining pattern is grown on these self-replicating seed structures by supplying the system with an appropriate set of pattern forming tiles.

1.3 Research Context

This research has been conducted in line with the research themes pursued at the Complex, Robust, Adaptive and Bio-inspired Solutions (CRAB lab) jointly headed by prof. Pauline C. Haddow and prof. Keith Downing in the Department of Computer and Information Science, NTNU. Although there are a variety of research areas covering Evolvable Hardware (EHW), Artificial Neural Networks (ANN), Robotics, Reconfigurable Computing Platforms, and Systems Biology etc., that have been pursued in the course of time at the CRAB lab, the general focus has been in creating bottom-up techniques for investigating the emergence of complex, robust and/or adaptive solutions, whether natural or artificial. In this pursuit many existing and newer bio-inspired techniques are investigated and applied to various application areas as well as to the design of future computational materials.

This thesis started initially with the aim to investigate unconventional computing substrates that can directly be used for the implementation of biological features, such as robustness, autonomous reproduction, reconfiguration, hierarchy, and adaptation etc. Towards this quest, we found out that DNA self-assembly is a promising technique that allows engineering of programmable systems using nanoscale DNA motifs (DNA tiles). Due to the interdisciplinary nature of this research field and our lack of prior knowledge in some of these disciplines, we opted to limit the preliminary ambitious goal of our research to a reasonable research objective, as described in Section 1.4

1.4 Research Questions

Motivated by recent developments in the field of DNA-based nanoengineering and intrigued by crucial challenges facing the field today, this research has been guided by the following main research question:

Main research question: *How to introduce dynamic control in the self-assembly of DNA tiles so as to reliably assemble programmable patterns of tiles in two-dimensions, and autonomously reproduce (self-replicate) them?*

To answer the main research question, it has further been divided in the following two sub-questions:

SQ-1: *Can we enhance the correctness of algorithmically self-assembling DNA tile systems by introducing kinetic control in the local interactions of tiles ?*

Self-assembly of DNA tiles has been proposed as a general method for information-guided molecular computation [154, 113, 112, 156, 14] and construction of complex, periodic and aperiodic patterns [157, 112, 14, 15, 126]. Theoretically these works have demonstrated significant complexity [113, 11], but the reliability of assembled patterns is often poor in practice (experiments in tile self-assembly have reported assembly error rates as large as 10%) [112, 14, 15, 128]. Assembly errors are introduced mainly due to cooperative binding between tiles, where a weakly attached tile — that must fall off quickly for a correct growth —is stabilized by the attachment of another tile in its neighborhood [156, 49].

Previously, assembly error prevention/correction techniques have been suggested, where original tiles are replaced by redundant blocks of n×n (2×2 or 3×3) tiles [156, 65], or a molecular tile structure is designed that introduces kinetic control in the self-assembly process of tiles [49, 83]. Redundant tile blocks reduce errors [121] in the self-assembly of tiles. However, it requires to design n^2 tiles with orthogonal sticky end sequences for each redundant block. Further, the assembled computational lattices using this approach have an increase in their size by a factor of n^2 [156] for n×n redundant tile blocks. The second approach introduces new designs of molecular tile structures [49, 83] that consist of two layers: sticky ends of original tile are protected by a protective layer. Self-assembly of protected tiles is kinetically driven: the protective layer kinetically discriminates between a mismatched attachment and a correct attachment. When the attaching tile matches correctly, its protective layer releases and the original tile gets assembled with the growing aggregate.

Following these error suppression approaches, particularly kinetically driven self-assembly of protected tiles [49] and Activatable tiles [83], we wondered whether a general protection mechanism of DNA tiles can be introduced to render kinetically controlled local interactions between tiles. The error-suppressing protection mechanism should include: 1) sticky ends of tiles should be fully protected to safeguard them against any illegitimate attachments; 2) there should be a strong kinetically-based discrimination between tiles assembling erroneously and correctly; 3) following a correct attachment during the assembly, protected tiles should self-trigger the complete dissociation of its PT, thus clearing the assembled tile for subsequent attachment by arriving tiles.

SQ-2: *How should DNA tile self-assembly be used to autonomously reproduce (self-replicate) algorithmically self-assembled rectangular patterns of tiles?*

Self-assembly and self-replication are two closely related terms used in biology and molecular synthesis. A self-replicating chemical system [148] includes two elements: a template molecule and a few substrate molecules capable of self-assembling an exact replica of the template molecule. The assembled replica must be able to dissociate from the template so as to produce two copies of the template: the former template and the newly created template. These templates need to then be able to catalyse a reiteration of the process by self-assembly of two new replicates on the two templates. Such a process theoretically results in an exponential amplification of the number of templates.

Previously, Schulman [120, 128] demonstrated the pattern self-replication in the tile self-assembly framework. The underlying principle is that an externally forced random fragmentation of a self-assembling pattern of tiles would create two new nucleation sites for regrowth of patterns. Such a pattern growth followed by fragmentation would eventually result into an amplified number of copies of the tile patterns. In a similar effort, Abel et al. [6] proposed a theoretical design of tile pattern self-replication, where post-assembly dissociation of tile patterns is realized using enzymes. Although this is a shape-independent replication mechanism applicable to both a precise and an infinite yield, usage of enzymes could pose practical limitations. Keenan et al. [75] studied the tile pattern self-replication using signal-controlled tiles that are triggered to dissociate intermediate assemblies by passing signals. Although the technique provides an innovative approach for enzyme-free self-replication of tile patterns, design and implementation of signal-passing tiles and their self-assembly would pose significant practical challenges [80, 95].

To answer the second part of the main research question, we study self-replication in the tile self-assembly framework. We ask whether the tile self-assembly methodology, which was initially introduced as a static process, can be equipped with dynamic control so as to autonomously reproduce many copies of an algorithmically self-assembled pattern of tiles. Using motivation from previous work in tile pattern self-replication, we wanted to design a self-replicating system for algorithmically programmed tile pattens that adheres to simplicity and implementation feasibility in following aspects: 1) it should be using well-studied tile self-assembly framework based on DX DNA tiles; 2) centralized inhibitor signalling is used to dynamically control the disassembly and reassembly of intermediate tile structures; 3) the replication process is enzyme free.

1.5 Thesis Structure

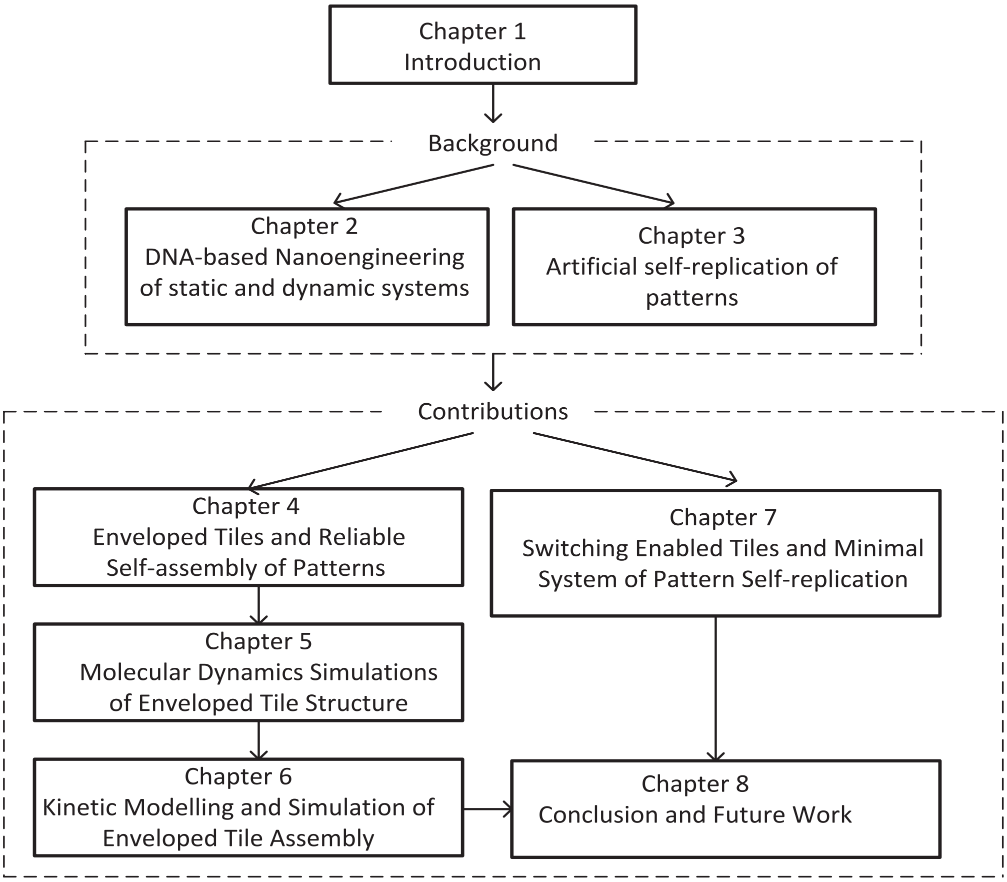

Chapter 1
Introduction
Background
Chapter 2
DNA-based Nanoengineering of static and dynamic systems
Chapter 3
Artificial self-replication of patterns
Contributions
Chapter 4
Enveloped Tiles and Reliable Self-assembly of Patterns
Chapter 5
Molecular Dynamics Simulations of Enveloped Tile Structure
Chapter 6
Kinetic Modelling and Simulation of Enveloped Tile Assembly
Chapter 7
Switching Enabled Tiles and Minimal System of Pattern Self-replication
Chapter 8
Conclusion and Future Work

Figure 1.1: The structure of this thesis.

The remainder of the thesis is organized as follows (Figure 1.1). Chapter 2 presents the background of DNA-based engineering. DNA tile motifs as building blocks of the programmable self-assembly and tile assembly models have been described. Following is a description of the DNA strand displacement mechanism and its use in the design of DNA-based dynamic systems. In the end, it presents work that introduces dynamic control in the self-assembly of DNA nanostructures.

Chapter 3 reviews selectively the background work in artificial self-replication in a chronological order. It covers the self-replication studied in cellular automata model, physical self-replication systems, and chemical self-replication systems. Self-replicating systems in these areas have been compared and contrasted. In the end, we discuss recent related work in the tile pattern self-replication studied in the framework of tile self-assembly.

Chapter 4 introduces the concept of error-prevention in the tile self-assembly using protected tiles. Design of an error preventing protected tile called ET is described in detail. Following is the mechanism of ETs that enables reliable self-assembly of tile structures. In the end, an example of an ET set design for the Sierpinski Pattern assembly is explained. This chapter uses material from the published article:
Vinay K Gautam, Pauline C. Haddow, and Martin Kuiper, "Reliable Self-assembly by Self-triggered Activation of Enveloped DNA Tiles" [51].

Chapter 5 presents coarse-grained Molecular Dynamics simulations to study the thermodynamic and kinetic feasibility of the ET assembly process. We simulated free-energy profiles of the PT during its release from a self-assembling ET. The results establish that at room temperature a correctly self-assembling ET would successfully release its PT, while its base element will assemble with the growing aggregate.

In Chapter 6, a detailed kinetics modeling of the ET self-assembly process is described. Using kinetic simulations, error prevention/correction and assembly timing of the ET self-assembly are studied and compared with state-of-the-art error prevention/correction methods. The result shows that the ETs can serve as building blocks of a reliable self-assembly of tiles.

Chapter 7 outlines minimal requirements for an autonomous self-replication of tile structures in the tile self-assembly framework. Autonomous self-replication is a subset of static tile self-assembly that requires dynamically controlled disassembly and reassembly of tile structures. To enable dynamically controlled switching in the tile self-assembly, we present Switching Enabled Tiles, which can be set ON (assembled state) or OFF (disassembled state) with the help of an externally supplied inhibitor signal of DNA, which is generated by an integrated reaction system of DNA molecules. Following is the description of tile pattern self-replication simulator that has been developed to study the pattern self-replication dynamics. The simulator uses abstract and kinetic models of tile self-assembly, thus it comes in two versions. This chapter uses material from the published article:
Vinay K Gautam , Eugen Czeizler, Pauline C Haddow, and Martin Kuiper, "Design of a Minimal System for Self-replication of Rectangular Patterns of DNA Tiles" [50].

Finally, Chapter 8 concludes by discussing the main contributions of this thesis and motivations for further research.

2

Background: DNA-based Nanoengineering of Static and Dynamic Systems

This chapter starts with an introduction of the self-assembly mechanism and its use as an engineering concept. Following is a description of DNA polymer as an engineering material for nanoscale system design. Further, DNA self-assembly is introduced with a brief mention of the recent exciting developments in this field. This is followed by a description of two major themes in DNA-based engineering: first, DNA tile-based static self-assembly that is used to implement computation and manufacture static structures of DNA, and second, strand displacement reaction-based dynamic DNA system design. At the end of this chapter, we provide more specific background details of the issues this thesis is built on: previous work related to integration of the two themes in DNA-based engineering, and efforts taken to introduce error resilience in the error-prone tile assembly process.

2.1 Self-assembly

Inside every cell, hierarchical structures e.g, cytoskeleton, microtubules, actin filaments and others, are assembled autonomously via specific and local interactions between a variety of biomolecules, such as DNA, RNA, proteins and others [44, 151, 85]. These self-assembling systems of remarkable complexity and functionality have been designed by nature through selection and evolution. Although a detailed understanding of the principles behind nature's way of designing remains elusive, the self-assembly process provides an attractive route to engineering artificial systems ranging from nano to millimeter scales [151, 152, 59]. The multidisciplinary nature of self-assembly makes it hard to find a universal definition. However, in a general context, self-assembly can be defined as a process of autonomous organization of ordered structures through selective affinity (binding) of pre-existing building blocks [151]. A typical self-assembly system has four requirements: *building blocks* that self-assemble into ordered structures, *binding forces* that dictate the local interactions between the building blocks, a medium in which self-assembly takes place, and a *driving force* within the assembly medium that drives the assembly system towards forming ordered structures of building blocks.

Building Blocks: A self-assembling system consists of a set of building blocks existing as either separate components (e.g., molecules) or a linked chain of multiple intramolecular interacting sites (e.g., protein sheets). Self-assembly is executed through spontaneous physical and/or chemical processes enabling local interactions between the building blocks. Typically, the favored local interactions are those resulting in ordered configurations of minimum free-energy.

Binding Forces: Self-assembling building blocks interact using 'weaker' bonds so as to access different configurations during the self-assembly process. Typically, stable binding between two building blocks requires multiple such weaker bonds to be formed simultaneously. For example, molecular self-assembly is guided by noncovalent interactions, such as hydrogen bonding, metal coordination, van der Waals forces, electrostatic interactions, and hydrophobic forces. On the other hand, interactions in a macroscopic self-assembling system are often guided by external forces, such as gravitational, magnetic, capillary, and electormagnetic [151].

Medium: To enable local interactions between the building blocks in a self-assembling system, it is essential that building blocks should be able to move inside the reaction medium. Molecular self-assembly is performed in the liquid medium where building blocks easily diffuse and interact locally to form ordered structures. In a liquid medium, environmental factors, such as, temperature, concentration of building blocks, pH, and salt concentration influence the interaction properties of building blocks.

Driving Force: A physical/chemical self-assembling system requires an overall driving force to drive the process of ordered structure formation through local interactions of building blocks. For example, self-assembly systems at the molecular scale that produce ordered structures of minimum free-energy are driven by thermodynamics of interacting building blocks. In a thermodynamically driven self-assembling system local interactions between the building blocks are required to be reversible in order to form minimum free-energy structures at equilibrium. Therefore, such processes are often very slow and require fine control over different environmental factors during execution. However, macroscopic self-assembling systems based on magnetic and gravitational forces do not require these equilibrium states and thus are driven usually fast.

2.1.1 Types of Self-assembly

All self-assembling systems spontaneously create structures of higher order through local interactions between building blocks, i.e., a self-assembling structure builds itself. Self-assembly can be classified into two broader categories: *static* and *dynamic* [151].

Static Self-assembly: The ordered structures formed by static self-assembly are at local or global equilibrium. This may require some form of agitation or stirring to drive the static self-assembling systems, but structures once assembled, are stable, at their minimum free-energy (equilibrium), and do not dissipate energy. In nature, assembly of virus particles and folding reactions of some of the proteins are common examples of static self-assembly. Most research in self-assembly has been focused on these static examples.

Dynamic Self-assembly: It requires an external supply or dissipation of energy to drive local interactions of self-assembling building blocks forming higher order structures. Thus, a dynamic self-assembly system is driven away from equilibrium, and the assembled structures may not necessarily be in their minimum free-energy state. Although dynamic self-assembly is ubiquitous in several biological processes of remarkable adaptability, complexity and functionality [74, 85], there is very little understanding of its mechanisms and the ways in which they can be applied to the engineering of novel systems that otherwise can not be designed using static self-assembly.

2.1.2 Self-assembly as a Design Concept

Self-assembly in nature is compelling for engineering of artificial systems due to several reasons. First, an understanding of mechanisms governing spontaneous formation of biological systems would enable engineering of artificial systems with complexity and novelty unimaginable for the top-down design approach. Second, such a bottom-up design approach might be the only route to designing nanoscale systems, where it is not possible to have individual control over a large number of nanoscale components.

To design a self-assembly system, one has to decide the structure of building blocks and the types of bonds governing location and connectivity of the building blocks. For example, a two-dimensional target structure can be assembled using a set of square shaped building blocks having a set of complementary bonds (glues) associated with their edges. A set of building blocks together with glues, is equivalent to instructions that guide the self-assembling system to produce a particular structure from myriad possibilities. Based on the types of building blocks, self-assembly systems can be classified into the following three types.

Periodic Self-assembly (PSA): In a PSA, as illustrated in Figure 2.1, there are two types of building blocks, each building block has a unique bond, but all the bonds within a single building block are the same. However, bonds between the two types of blocks are complementary. Therefore, there is a single possibility of binding between any two building blocks having complementary bonds. This type of self-assembly produces periodic pattern, as shown in Figure 2.1(b). There is no control over the final size of the assembled structurein a PSA: as long as there is a constant supply of the individual building blocks, the pattern would grow. Also, it lacks programability i.e., the final assembled pattern is fixed, and therefore if one wishes to modify the pattern, the building blocks needs to be modified.

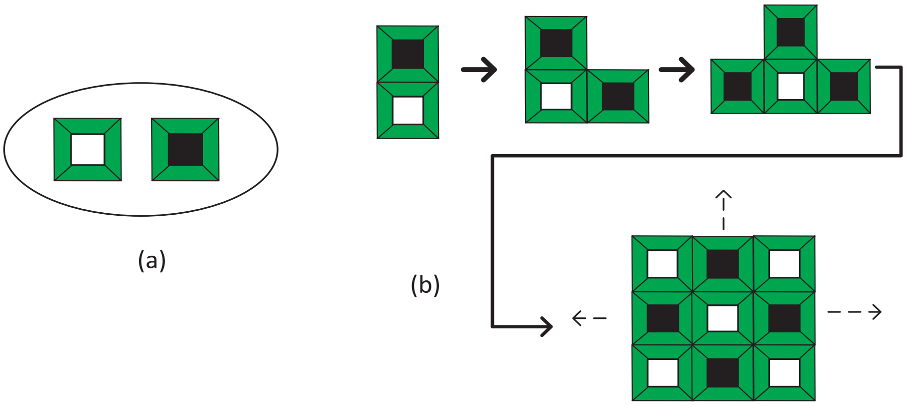


Figure 2.1: A periodic self-assembly system. (a) A set of two building blocks; all four bonds within a block have the same binding logic (shown by green edges of the square block). (b) A few steps of the periodic self-assembly.

Uniquely Addressable Self-assembly (UASA): In a UASA, as shown in Figure 2.2, each building block of a self-assembly system has a unique set of bonds. Therefore, every position within the assembled pattern is occupied by a distinct building block. The position is decided by other adjacent building blocks. A UASA system requires a large number of building blocks, thus limiting the size and complexity of the assembled structures.

Algorithmic (or Programmable) Self-assembly (ASA): The ASA introduces a form of programmable control over the self-assembly process, while also enabling the design of complex structures through a smaller set of components. The set of building blocks for an ASA are designed such that they encode both the configuration of final assembled structure and the sequence of steps required to produce it.

Using only four types of building blocks, a programmable pattern is self-assembled, as shown in Figure 2.3. The edges of the building blocks are designed such that each edge has two options of tiles that can attach with it. If such building blocks are set

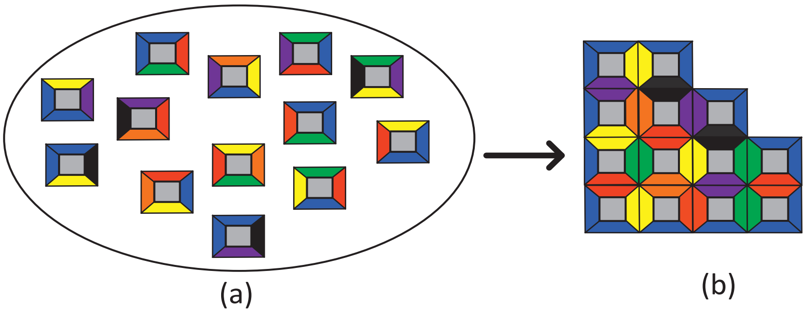


Figure 2.2: A uniquely addressable self-assembly system.(a) A set of building blocks; each block has a unique combination of bonds. (b) A uniquely addressable self-assembled pattern.

to self-assemble under the condition that each assembling block must attach with at least two other blocks, this would result in a programmable self-assembled pattern, as shown in Figure 2.3(b). In programmable self-assembly, a pre-formed seed is required to initiate (nucleate) the self-assembly process. The role of seed and nucleation processes are explained in Section 2.4.

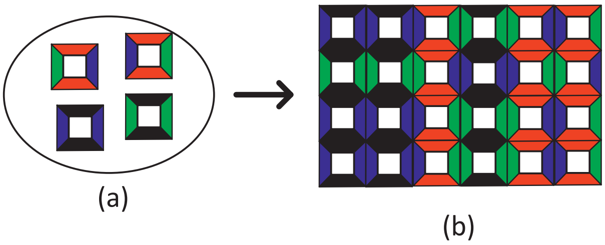


Figure 2.3: An algorithmic self-assemmbly system.(a) A set of building blocks designed for an algorithmic self-assembly of pattern. (b) Self-assembled pattern of the building blocks.

2.2 DNA as a Substrate for Nanoscale Engineering

DNA is a linear polymer existing in two forms, shown in Figure 2.4: single-stranded DNA (ss-DNA) and double-stranded DNA(ds-DNA). A ss-DNA, as shown in Figure 2.4(a), consists of repeating units of four different nucleotide bases named as A (Adenine), T (Thymine), G (Guanine), C (Cytosine), while a ds-DNA is formed out of binding between two ss-DNA that are complementary and anti-parallel as shown in Figure 2.4(b) . Two ss-DNA are said to be complementary if their base sequences are such that each 'A ' in one ss-DNA has a facing 'T' in the other ss-DNA, while 'C' faces 'G'. The complementary binding rule — A : T and C : G — is the canonical Watson-Crick base-pairing. The two ends of a ss-DNA are chemically asymmetric and are referred to as the 5' and 3' ends, respectively. The two ss-DNA that bind to form ds-DNA are anti-parallel i.e., if one is in the 3'→ 5' orientation, then the other should be in 5'→ 3'.

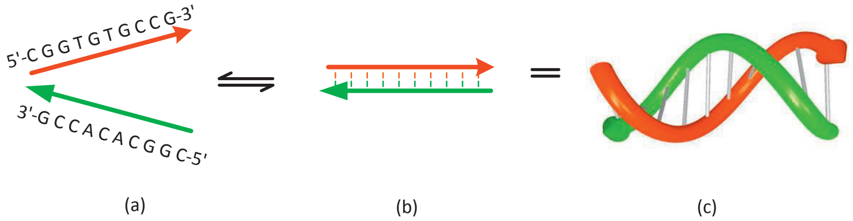


Figure 2.4: DNA structure.(a) Two complementary ss-DNA molecules each 10 bases long. (b) ds-DNA resulting from hybridisation of the complementary ss-DNA molecules, and (c) the helix structure of the ds-DNA molecule.

The basic topological properties of ds-DNA and ss-DNA are illustrated in Figure 2.5. The shape of ds-DNA is a double helix, while ss-DNA is flexible. The helix in ds-DNA makes a complete turn about its axis every 10.5 bases [130, 47]. The width of a ds-DNA is 2 nm, and the distance between consecutive base pairs along the helix is 0.34 nm. The distance between consecutive bases in ss-DNA is ≈ 0.7 nm [139].The ds-DNA has higher stiffness than the ss-DNA — the persistence length[1] of ds-DNA is larger than 150 base-pairs [84], while ss-DNA has a persistence length of ≈ 3 bases [163].

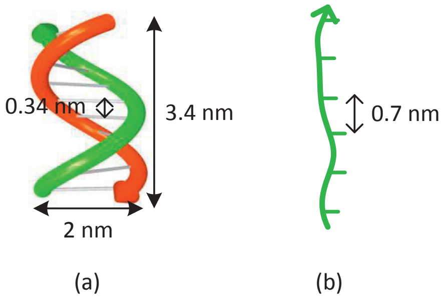


Figure 2.5: Topology of DNA.(a) Width, length of one helix turn, and spacing between two consecutive bases in the ds-DNA. (b)Spacing between consecutive bases in the ss-DNA.

DNA looks to be a suitable physical substrate for molecular computation and nanoengineering [132], at least from the following four perspectives:

[1] The persistence length of a polymer is a measure of its elasticity. It is is a way of measuring the limit at which a polymer ceases to be treated like an elastic object and can be treated statically [84].

First, thermodynamic properties of DNA are well known and can be predicted reliably using the the Nearest-Neighbour (NN) model [115]. Therefore, given a sequence of DNA and reaction conditions (e.g.,salt concentrations and temperature), one can calculate entropy (ΔS°), enthalpy (ΔH°), and free energy (ΔG°) using a number of freely available packages: Mfold [177], NUPACK [169] etc.

Second, it is possible to do reverse engineering of DNA, i.e., given a target secondary structure[2]. One can use the software package (e.g., NUPACK [169]) to search heuristically for a set of potential DNA sequences that produce the target structure with high affinity and specificity. Third, DNA nanoengineering relies mostly on ss-DNA sequences of short length (less than 100 nucleotides long), that can easily be synthesized chemically and bought from synthesis companies. The rapidly falling prices of oligonucleotide synthesis, purification and sample preparation has served as one of the major driving forces for DNA research. Fourth, DNA engineered molecular systems are bio-compatible, thus suitable for in-vivo integration.

Thermodynamics and Kinetics of DNA Hybridisation Reaction: Hybridisation is the chemical process driving the transformation of ss-DNA to ds-DNA, and vice versa [40]. A hybridisation reaction of DNA molecules can be driven reversibly by varying the reaction temperature under given reaction conditions. For example, lowering the temperature causes two complementary ss-DNA molecules to form a relatively stable ds-DNA, and the process is termed *thermal annealing*. Alternatively, a ds-DNA dissociates into its constituent ss-DNA if the temperature is raised, and the process is termed *denaturing or melting*. The reaction temperature at which half of the ds-DNA has dissociated into the ss-DNA form is known as the *melting temperature*(T_m)of the DNA.

In the following, we illustrate the thermodynamics and kinetics of DNA hybridization reaction using the ss-DNA molecules shown in Figure 2.4(a)).

$$ssDNA_1 + ssDNA_2 \underset{k_r}{\overset{k_f}{\rightleftharpoons}} dsDNA$$

The equilibrium constant for the above reaction is

$$K_{eq} = \frac{[dsDNA]}{[ssDNA_1][ssDNA_2]} \tag{2.1}$$

[2] The reverse engineering of DNA sequences is currently limited to only non-tertiary (DNA structures free from pseudoknots [36]) target structures. Although the Sequence Symmetry Minimization (SSM) [131] technique has been in use for sequence design of tertiary structures of DNA (e.g., DNA tiles [157]), there is no theoretical evidence if such designed sequences would yield the target structure with optimum probability [35], because the sequence search criteria is based on the optimisation of desired base-pairing and minimization of spurious base-pairing. But it does not consider other competing structures which might form with higher probability out of the optimised sequences

Further, (K_{eq}) and standard free energy (ΔG°) of the above reaction are related as:

$$\Delta G^{\circ} = -RT\ ln(K_{eq}) \tag{2.2}$$

where R is the Gas Constant (R = 1.98722 cal $K^{-1}\ mol^{-1}$) and T is the thermodynamic temperature in Kelvin (K).

The ΔG° of the reaction is calculated as follows:

$$\Delta G^{\circ} = \Delta G^{\circ}(ssDNA_1) + \Delta G^{\circ}(ssDNA_2) - \Delta G^{\circ}(dsDNA)$$

Free energy, ΔG°, thus determines whether the reaction will be favourable or not. In this reaction, $\Delta G^{\circ} < 0$ means the hybridisation reaction can be driven to produce ds-DNA species by supplying an appropriate amount of ss-DNA species.

To understand the thermodynamics and kinetics of the DNA hybridization, let's use the following analysis.

Using the NN model, the standard free energy of the ds-DNA is $\Delta G^{\circ} \approx -14$ Kcal/mol at T = 310 K, which corresponds to $K_{eq} \approx 10^{10}\ M^{-1}$ using (2.2). Let's consider the initial concentrations of the reactions species as follows: $[ssDNA_1] = [ssDNA_2] = 1\mu M$ and $[dsDNA] = 0$. Using (2.1), the equilibrium concentration of the ds-DNA is $\approx 10^4 \mu M$, which is very large in comparison to the starting concentrations of the ss-DNA species ([ss-DNA] is only 0.01% of [ds-DNA]). Thus, the hybridisation reaction is driven 99.99% to completion. Now, imagine the DNA hybridisation reaction having $\Delta G^{\circ} \approx -4$ kcal/mol. For this reaction, $K_{eq} \approx 10^3 M^{-1}$. Considering the same initial concentrations of ss-DNAs and ds-DNA as in the previous case, the equilibrium concentration of ds-DNA is $\approx 10^{-3}\mu M$, which is merely 0.01% of the the initial concentration of the ss-DNAs. Thus, the reaction has not proceeded forward in this case, and the yield of the ds-DNA is very low. It could be seen from these examples that the equilibrium yield of DNA hybridisation reaction depends on the standard free energy of the reactants. In DNA-based engineering, the concentration of DNA species is typically in the μM range, thus DNA sequences are rationally designed to produce desired hybridization yields.

Further considering the reversible nature of the DNA hybridisation process, the equilibrium constant (2.1) can be represented by $K_{eq} = \frac{k_f}{k_r}$, where k_f and k_r are the rate constants of forward and backward reactions. The forward rate constant has a typical value, $k_f = 3 \times 10^6 M^{-1}\ s^{-1}$, for the diffusion-limited DNA hybridisation reactions [107]. The backward rate constants for the aforementioned DNA hybridisation reactions ($\Delta G^{\circ} \approx -14$ and $\Delta G^{\circ} \approx -4$) can be calculated as $k_r = 10^{-4}s^{-1}$ and $k_r = 10^3 s^{-1}$, respectively. The dissociation time for the former reaction is in the timescale of hours, while for the latter it is in milliseconds. Therefore, dissociation reactions of long duplexes of DNA (>12 base pairs) are very slow for any practical use. However, dissociation time of long DNA duplexes can be reduced using toehold-mediated strand displacement, explained in Section 2.5.

DNA-based Nanoscale Systems: Over the past two decades, there have been exciting developments in the field of DNA-based system design. There are two major themes in the field: a) DNA-based programmable self-assembly design for the implementation of molecular computation and the construction of static patterns and structures at the nanoscale; b) dynamic DNA system design, which uses toehold-mediated strand displacement [173] reactions to engineer DNA systems with complex behaviors.

The two main approaches in DNA self-assembly are 'DNA Origami' [161, 73, 38, 34] and tile self-assembly [154]. Both approaches have the common goal of creating nanostructures through programmable self-assembly of DNA. DNA Origmai uses a long single-stranded DNA and multiple short DNA strands (DNA staples) that fold the single strand into desired shape. Although DNA origami has emerged as an ideal method for one-pot self-assembly of arbitrary patterns and structures in 2D and 3D, use of single strand of DNA limits the size of the largest self-assembled origami structure to about 100 nanometer. The other approach, called DNA tile self-assembly [154, 78, 167, 61], uses structural building blocks of DNA (DNA tiles) to construct arbitrary structures and patterns and perform computation at the nanoscale. Tile self-assembly is controlled by the design of tile motifs and physico-chemical parameters of the self-assembly. The design of tile motifs includes: selection of appropriate structures (e.g., DX DNA junction [47], Holliday junction [130], Triple Crossover (TX) [78], Three-point star junction [167], T-junction [61]) for the tile motifs and sticky ends, and design of the DNA sequences that fold into the intended tile structures. Physical parameters, such as temperature and concentration of tile motifs play crucial roles in the wet-lab implementation of the tile self-assembly. Typically, tile self-assembly has been used to form structures that are at their minimum thermodynamic free-energy, and thus are static in nature. One of the strengths of the tile self-assembly is its scalability, that is, using a set of DNA tiles and appropriate laboratory conditions, one can construct programmable structures of micrometer size or even larger.

The other theme in the DNA-based engineering covers dynamic systems of DNA that use toehold-mediated strand displacement mechanism [168, 174]. The toehold-mediated strand displacement mechanism enables kinetic control in the DNA binding process that otherwise has no control except the thermodynamic one. DNA strand displacement reaction systems have been used to design catalytic systems of DNA [173, 172, 174, 175, 170], molecular motors [16], actuators [135, 136, 164],and other dynamic DNA circuits[105, 140]. Sections 2.3 - 2.5 present the basic principles of DNA tile self-assembly and dynamic systems of DNA.

2.3 Self-assembly of DNA Tiles

A connection between algorithmic self-assembly and computation was studied by Wang in his theoretical tiling model [149]. Wang tiling theory demonstrates the implementation of a Turing machine by a finite set of square tiles with four colored edges. The Wang tiling asks if a given finite plane can be covered by assembling a finite set of colored tiles such that the adjacent tiles have matching colored edges.

Double Crossover Molecular DNA Tiles

DNA tile self-assembly [157] enables programmable self-assembled structures of DNA molecules based on the mathematical tiling theory of Wang [149]. DX molecular DNA tiles [157], the building blocks of the tile self-assembly, consist of four (≈ 50 nucleotide) ss-DNA molecules, synthesized for a given DNA tile design. Figure 2.6 illustrates the construction of a DX molecular DNA tile with four DNA strands. As shown in (a), each ss-DNA consists of a sequence of nucleotides (A, T, G, C). The tiles self-assemble through the bonding of these ss-DNAs at room temperature. The bonding process occurs when two complimentary strands meet and their base pairs: A-T and G-C, bind. Any left-over bases from each of the bonded strands form a sticky end(s) — as shown in (b). As the term implies, this end is available to "stick" or bond to another strand. DX molecular DNA tiles are square shaped structures where sticky-ends are represented by their respective square edges — as illustrated in (c).

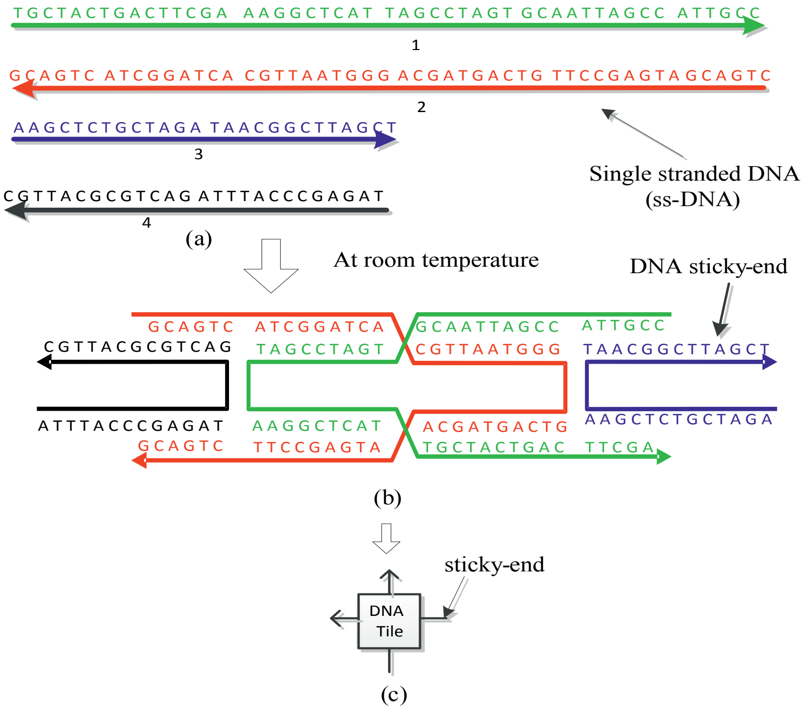


Figure 2.6: DX DNA Tile Structure(a) four ss-DNA (b) assembled DNA tile (c) abstract representation

The DNA strands intended to assemble DNA tiles are designed using Sequence Symmetry Minimization(SSM) [131], a heuristic search technique to design DNA sequences so as to prevent potential spurious bindings between different DNA sequences. The other commonly used CAD tools for the design, and analysis of DNA structures include: NUPACK [4], DNA Sequence Design [3], DNA SequenceGenerator and DNA SequenceCompiler [42], caDNAno [1] and CanDo [2].

Further, the tile self-assembly process itself can be realistically simulated using simulators such as the kinetic Tile Assembly Model (kTAM) [5]. In recent years, the availability of such simulation tools along with the reduced cost and availability of DNA molecules, has increased the interest in development of novel DNA self-assembly systems that are not just limited to DNA tiles.

2.4 Tile Assembly Models

The physical implementation of tile self-assembly in a wet-lab is often time-consuming, expensive and challenging with respect to reproducibility of results. Simulation of realistic models of DNA self-assembly provides a cheaper, faster (and more reliable) media in which to explore and refine new avenues of research, prior to experimentation. There are two simulation models of tile self-assembly, developed by Winfree [113, 154]: 1) The abstract Tile Assembly Model (*aTAM*), and 2) The kinetic Tile Assembly Model (*kTAM*).

2.4.1 The Abstract Tile Assembly Model

The *aTAM* is based on Wang's tiling theory [149], which requires creation of a finite set of square shape tiles that are abstract representations of DX DNA molecules [47] shown in Figure 2.7(a). In aTAM, a tile *t* is represented by a quadruple $(\sigma_S(t), \sigma_W(t), \sigma_N(t), \sigma_E(t))$, where $\sigma \in \Sigma$ is glue type associated with the four sides (North(N), South(S), West(W), East(E)) of a rotationally asymmetric unit square. The glue type, Σ, is a finite set, which is used to derive a glue strength function $(s : \Sigma \times \Sigma \rightarrow N)$ for a legitimate tile association between two glues of tiles. The glue strength function is symmetric, i.e., s(σ_1, σ_2) = s(σ_2, σ_1) $\forall \sigma_1, \sigma_2 \in \Sigma$.

A tile pattern assembly system (TPAS) $\mathcal{T} = (T, S, s, \tau)$ consists of a finite set T of tile types, an assembly S termed as seed assembly, a glue strength function s and a temperature parameter $\tau \in Z^+$. A tile assembly system has a temperature 'τ' if any larger structure of tiles cannot be dissociated into smaller assemblies without breaking bonds of total strength at least 'τ'. Alternatively, a tile can join the assembly as long as the sum of the strengths of the bonds that it makes with tiles already in the assembly is at least τ.

Figure 2.7 illustrates the self-assembly process of the Sierpinski pattern [156, 112] at temperature 2 ($\tau = 2$). The tile set comprises a seed tile, two boundary tiles and four rule tiles - see Figure 2.7(a). Tile edges are marked by non-negative integers illustrating their respective glue strengths. The South and West glues of the tiles are designed as inputs and the North and East glues are outputs.

Tile pattern assembly in the aTAM starts from a given seed structure that nucleates the pattern formation which grows into a finite or infinite pattern as more tiles join - see Figure 2.7(b). Tiles join by forming bonds with strength at least of τ (e.g., a $\tau = 2$ assembly requires tiles to bind with total strength at least 2). For a given TPAS, a pattern assembly P is said to be terminal, if no tile can be added further that satisfies the $\tau - stability$ criteria.

The aTAM has given insights to important theoretical aspects of the tile assembly systems, e.g, 1) what can or can't be self-assembled?, and 2) if something can be assembled, how efficient it could be?

2.4.2 The Kinetic Tile Assembly Model

The *aTAM* provides theoretical analyses of computing power of tile assembly, while the *kTAM* provides a plausible model of tile assembly by considering tile binding as a reversible physico-chemical process. In *kTAM* [154], each tile addition step is a reversible process governed by the tile concentration, local reaction temperature and the length of the tile's sticky ends. Furthermore, to simplify the complex process, it assumes that 1) the tile concentration is constant for each tile type, 2) only one tile can attach/detach from the growing aggregate at a time, 3) tiles do not change orientation, and 4) the binding strength between orthogonal pairs of sticky ends is negligible.

Figure 2.7(c) shows the underlying reversible kinetics of tile assembly in the *kTAM*. The rate of tile attachment (r_f) at a binding site of aggregate is directly proportional to the tile concentration. The concentration of each type of tile (except seed tile) can be given by $e^{-G_{mc}}$, where G_{mc} is the decrease in entropy while a tile binds at a vacant site. Therefore, the attachment rate (r_f) can be given by (2.3), where k_f is the reaction rate constant.

$$r_f = k_f e^{-G_{mc}} \quad . \tag{2.3}$$

Similarly, the tile detachment process is controlled by the energy required to break any single tile-aggregate bond and denoted by G_{se}. The value of G_{se} depends on the sticky-end length (s) and the temperature (T), where $G_{se} \approx (4000/T - 11)s$. The tile detachment reaction rate ($r_{r,b}$) involving b tile bonds is given by (2.4). Tile kinetic rates, $r_{r,2}$ and $r_{r,1}$ in Figure 2.7(c), are the detachment rates of tiles involving double and single bonds, respectively.

$$r_{r,b} = k_f e^{-bG_{se}} \quad . \tag{2.4}$$

The parameters, G_{mc} and G_{se}, govern the dynamics of tiles in the kTAM: a larger value of G_{mc} implies lower tile concentration and consequently slower forward reaction rates (or vice versa). Similarly, a larger value of G_{se} results in a slower detachment rate. The ratio ($G_{mc}/G_{se} = \tau$) is often used in the analysis of the kTAM. Growth dynamics

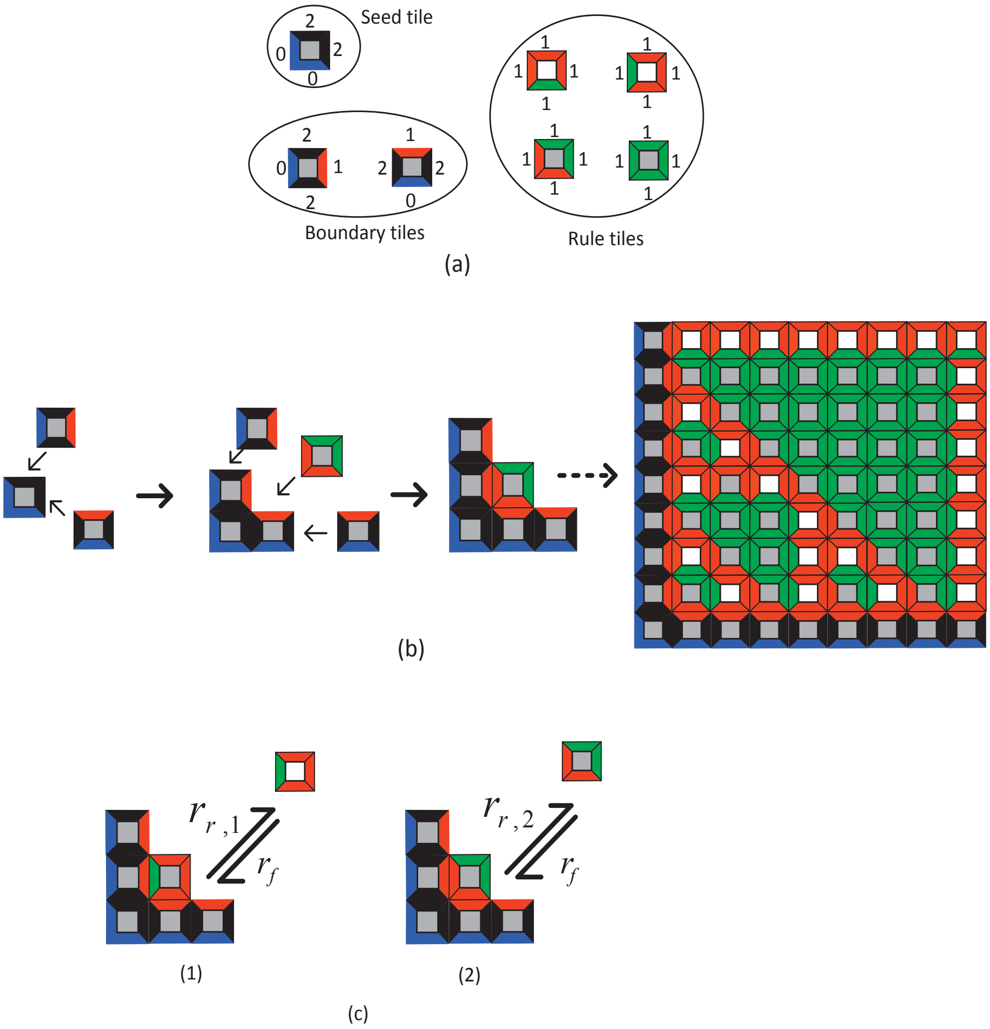


Figure 2.7: Sierpinski pattern self-assembly.(a) Sierpinski tile set (XOR tile set). (b) Steps of self-assembly of Sierpinski Pattern of size 9x9. (c) Kinetics of tile assembly in *kTAM*.

of tile self-assembly can be divided into three regions : 1) $\tau < 1$, 2) $1 < \tau < 2$, and 3) $\tau > 2$, as shown in Figure 2.8. If $\tau < 1$, then the forward rate is greater than the reverse rate and therefore, the assembly will grow faster, resulting in random assembled structures with a high error rate. If $1 < \tau \leq 2$, then it is possible to drive self-assembly to form computational structures with lower error rates. Near thermodynamic equilibrium ($G_{mc} \approx 2G_{se}$), optimal growth may be obtained with low error rates where most errors are of the form of kinetic trapping. Finally, if $\tau > 2$, self-assembly becomes infeasible because of the relatively large backward rate causing tiles to fall apart immediately after coming in contact with the aggregate.

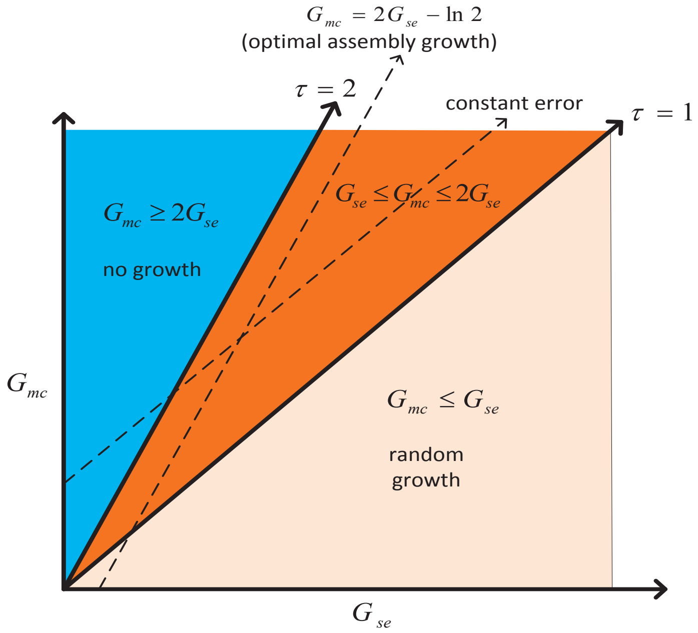


Figure 2.8: Phase diagram of the tile self-assembly dynamics in the kTAM. Random growth region: $\tau < 1(G_{mc} < G_{se})$. Algorithmic growth region: $1 < \tau < 2(G_{se} < G_{mc} < 2G_se)$. No growth region: $\tau > 2(G_{mc} > 2G_{se})$

The *kTAM* enables analyses of the assembly errors and growth rate for a given tile set. However, the aforementioned assumptions in the *kTAM* oversimplify the physico-chemical process of tile self-assembly. For example: 1) the tile concentration would not remain constant, as the lattice grows tile concentration would decrease, and 2) tiles may easily flip across the axes and therefore, orthogonal sticky end sequences of tiles should be designed after taking this aspect into consideration. Nonetheless, the growth and error rates derived by the *kTAM* simulations comply with the experimental observations near equilibrium [89, 124].

2.5 Engineering of Dynamic DNA-based Systems

In static self-assembly of DNA systems, as described earlier, thermodynamically stable DNA structures self-assemble by hybridization between complementary DNA domains as the system is annealed slowly. Therefore, local interactions between different DNA domains have no control other than to ‘hybridize’ or ‘not to hybridize’. However, to design a self-assembly system with complex, dynamic and autonomous behaviors, local interactions should be controllable so as to assemble and disassemble different molecular entities. The toehold-mediated DNA strand displacement [168, 173], described below, is used to design dynamic control over assembly and disassembly of DNA molecules. The toehold-mediated strand displacement has been widely used in the design of dynamic DNA systems, such as tweezers, motors [168, 16], walkers [133, 166], DNA logic circuits [129, 106, 105], and enzyme-free catalytic systems [129, 173, 174, 165, 175]. A review article, Zhang et al. [170], presents detailed state-of-the-art work in dynamic DNA self-assembly.

2.5.1 Toehold-mediated DNA Strand Displacement

Toehold-mediated DNA strand displacement is a DNA-based reaction mechanism, where one DNA molecule is displaced by a second DNA molecule. The reaction is driven forward by a net loss in the configuration entropy of the DNA reaction system. A toehold-mediated DNA strand displacement mechanism is illustrated in Figure 2.9. Typically, there are two DNA molecules used in a displacement reaction (see Figure 2.9(1)): 1) a partially bound duplex DNA [TB^*B] that usually comes with a short overhang of DNA, termed *toehold* (T); 2) a single strand of DNA, termed as *invader* DNA strand [T^*B']. The invader DNA strand binds with the toehold of the duplex (see, Figure 2.9(2)) and starts the srand displacement process (see the intermediate step, Figure 2.9(3)), which ultimately results in displacing the DNA molecule (branch (B)) from the duplex. The intermediate step is known as *branch migration*, where duplex base-pairs in the incumbent side of the invader-duplex junction break away and subsequently new base-pairs are formed between the duplex and the *invader*. Thus, the incumbent strand will eventually be displaced by the invader as base-pairs of the incumbent are exchanged with the *invader's* base-pairs, completing strand displacement, as shown in Figure 2.9(4). However, if a base-pair breaks on the invader side of the junction, it is known as *fraying* [141], and the invader may eventually dissociate from the duplex if all its toehold base-pairs simultaneously fray away.

Yurke et al. [168] demonstrated the first dynamic DNA self-assembly system of a molecular *tweezer*, shown in Figure 2.10, using the DNA strand displacement mechanism. The molecular *tweezer*, in *open* state, consists of three DNA strands that form two stiff arms of DNA duplexes connected by a small flexible hinge of single-stranded DNA and two short overhangs of DNA protruding from the two arms. In the *open* state, the two ends of the *tweezer* are separated, while supplying a fourth strand *F*, known as fuel, the two arms are brought closer as the fuel binds with the free ends, thus driving the tweezer to the *closed* state. Further, the tweezer is switched from the *closed* to the *open* state through a strand displacement reaction triggered by a fifth strand, ($\bar{F}$). The $\bar{F}$ strand is complementary to the *F*. Therefore binding at the overhanging end of the F, initiates downward strand displacement which ultimately results in a double-stranded waste product ($F\bar{F}$), and the tweezer opens up. Thus, the tweezer can be cycled between the two states by supplying *F* and $\bar{F}$ in succession. This presents a dynamically controlled self-assembly system of DNA that functions far from equilibrium.

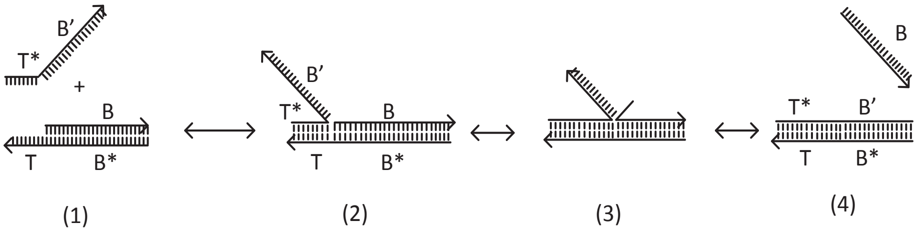


Figure 2.9: Toehold-mediated strand displacement process.(1) A typical strand displacement system consisting of a duplex with an overhanging short *toehold*(T), and an *invader* DNA strand($[T^{*}B']$), where (*) is used to denote complementary DNA sequence. (2) Strand displacement process is initiated as the *invader* binds with the *toehold* of the duplex. (3) Intermediate steps of *branch migration*.(4) A successful *branch migration* displaces the *incumbent* strand from the duplex, and displacement is completed.

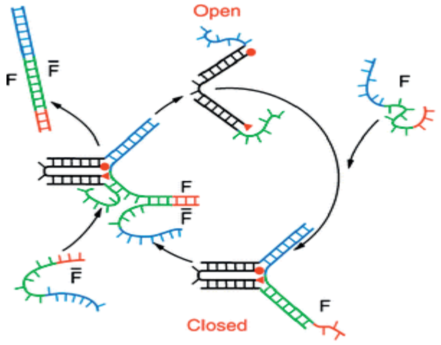


Figure 2.10: Dynamic DNA system of molecular tweezer [168]. Tweezer operates between *open* and *closed* states through a successive supply of fuel strands (*F*) and removal strands($\bar{F}$)

The toehold has a major role in the design of a strand displacement process. First, its length and sequence composition have a significant influence over the kinetic rate of strand displacement [174] — kinetic rate varies a millionfold over a toehold length six bases or less, and saturates for longer toeholds. Second, toeholds also serve as recognition domains for input DNA strands (*invaders*). The first feature allows engineering of the non-equilibrium systems of competing DNA reactions [168, 174, 52], while the second feature allows designing cascaded strand displacement systems through sequestered toeholds that are activated conditionally [129, 165].

A sequestered toehold introduces several interesting features into strand displacement systems. First, it enables the design of multiple DNA complexes with reactive motifs buried either within duplex regions [129] or inside hairpin loops [36], as shown in Figure 2.11(a,c). Second, DNA complexes and hairpins are designed with mutually non-reactive toeholds. Thus, to activate the strand displacement process, an input is required that triggers the downward strand displacement reactions as sequestered toeholds are revealed, Figure 2.11(b,d). Using sequestered toeholds thus avoids the need for multiple inputs.

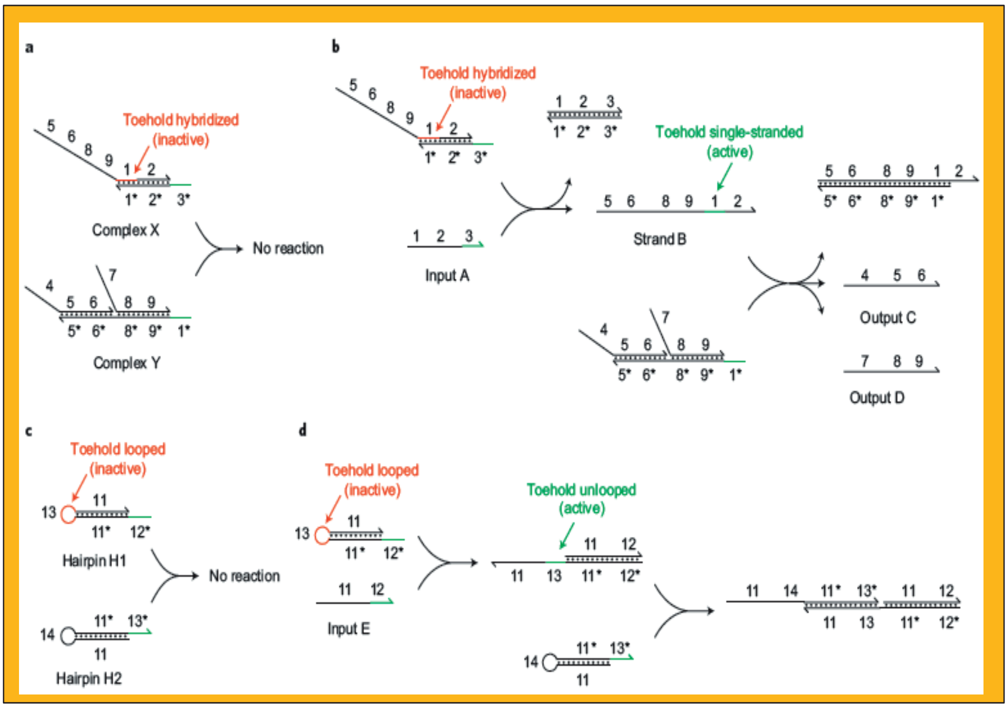


Figure 2.11: Strand displacement systems with sequestered toeholds [170].(a) Complex X and complex Y have non-reactive toeholds, represented by 1^* and 3^*, respectively. (b) Input strand A triggeres the strand displacement in complex X that activates toehold domain *1* for a downward strand displacement of complex Y. (c) and (d) represent a toehold activation and downward strand displacement in hairpin based complexes.

2.6 Combining the Two Aspects of DNA-based Approaches

Despite being closely related, the two aforementioned approaches in DNA-based engineering have been applied to design DNA systems that are loosely divided into two classes: structural DNA nanosystems [154, 112, 73, 14, 16, 125, 128] and dynamic

DNA nanosystems [168, 129, 173, 165, 174, 172, 106, 105]. The former class of DNA systems are designed using static self-assembly process, where structural (spatial arrangement) aspect of a system is the main focus. However, the latter class of DNA systems have implemented dynamic behaviors using cascaded reaction networks of DNA strand species [129, 173, 172, 170, 105], where structural aspects are not the prime concern. These approaches have been studied extensively using both theoretical and realistic models [154, 113, 156, 140], and state-of-the-art DNA systems of both types are scalable and complex.

Recent work [171, 114, 119] suggests that the two approaches can be combined to introduce dynamic control in the self-assembling structural DNA systems. Previous work [36, 165], though not designed for dynamic control of a spatial DNA system, has proposed mechanisms for the dynamic control of sequentially driven DNA reaction networks. For example, Dirk et al. [36] demonstrated the design of a system with protected DNA components, where the components do not react together until an initiator DNA strand is introduced that triggers the self-assembly of the components. Another work by Yin et al. [165] proposed implementation of programmable chemical assembly pathways through DNA hairpins-based toehold sequestering that triggers the order and timing of strand displacement processes. A more recent work by Sadowski et al. [114] demonstrated kinetically controlled self-assembly of a DNA tetrahedron.

Zhang et al. [171] demonstrated a method for the integration of the two most popular approaches: the DNA tile self-assembly and strand displacement reaction systems. The basic idea was to control the formation of DNA nanotubes, which is self-assembled using DNA tiles, using an integrated control mechanism that is driven by a system of toehold-mediated DNA strand displacement reactions. Recently, in a theoretical approach, Schiefer and Winfree [119] discussed the generalisation of the integration mechanism and its challenges.

2.7 Summary

This chapter provides a brief introduction to the self-assembly from a design perspective, illustrates the basic physicochemical properties of the DNA, and discusses the basic concepts used in the two major approaches of DNA-based nanosystem design. The concept of DNA tile design, self-assembly of DNA tile system, and modeling of the tile self-assembly process have been illustrated. The approach of dynamic DNA-based system design and the basic mechanism of toehold-mediated strand displacement have been introduced. In the end, we mention the recent efforts that are trying to introduce dynamic control in the DNA-based nanosystems.

3

Background: Artificial Self-replication of Patterns

To illustrate the main concepts used in the designs of artificial self-replicating systems, this chapter presents a selected review of work in both cellular automata models of self-replication and kinematic self-replication models. Starting with the cellular automata based models first, von Neumann's universal constructor machine [18] and seminal work of Langton [79] are described. This is followed by a review of kinematic self-replicating systems underlying self-assembly principle. This includes: physical self-replicating systems of Penrose [102], and Griffith [58]. Following are descriptions of molecular self-replicating systems, that include: the catalytic model by Kiedrowski [134], physics simulation of self-replicating molecules [138, 41], and self-replication model based on artificial chemistry [67]. At the end of this chapter, we present the state-of-the-art in DNA tile pattern self-replicators that are closely related to the tile pattern self-replicator discussed in Chapter 7.

3.1 Introduction

In 1950, John von Neumann presented five models for the realization of self-reproducing computing machines [18]. Two of his main models, the kinematic model and the cellular automata model, have been developed further by his successors [13, 30, 102]. The kinematic approach has been used to design self-replicating systems, where it is not merely the pattern of information that is replicated but real physical copies of given structure are produced. The second model of self-replication, the popular cellular automata (CA) model, was described in much detail and has been studied rigorously over the last 65 years. In this approach, self-replication is studied as a process whereby a pattern of information is capable of creating copies of itself in the cellular automata framework. Research in cellular automata has provided a platform to develop understanding of fundamental information processing principles underlying self-replication. This understanding has given useful insights to the engineering of programmable self-replicating computing systems, and helped in understanding the principles behind self-organization and information processing in living systems.

CA models of self-replication have given insights into simple sets of requirements for the self-replication of information carrying structures. Likewise, physical/chemical self-replicating systems with simple requirements have also been reported [102, 53, 58]. Although CA models of self-replicating loops have produced interesting studies of emergence and life, they are far from physical realism. Using physical/chemical components, kinematic self-replication systems with simple requirements have been implemented [102, 53, 58]. Specifically, self-replicating models and systems of molecular components are promising for both low-cost manufacturing at the nanoscale and realization of programmable self-replicating information structures.

DNA tile self-assembly has recently emerged as a technique for the implementation of programmable pattern of nanostructural building blocks, termed DNA tiles. Although there are certain distinctions[1] between the CA and DNA tile self-assembly process, DNA tiles are nothing but physical realization of cellular automata [112]. Considering our interest in studying the self-replication in the DNA tile self-assembly framework, our major focus herein has been to give an account of a chronological and conceptual development of artificial self-replicating systems that has happened over the last 65 years.

3.2 Historical Background

Self-replication and self-reproduction are two important terms used for explaining life on earth. The former represents the process of creating identical copies of genetic information contained within cellular DNA, while the later symbolizes the process of producing new organisms through mutation, crossover and division. Remarkable robustness shown by natural systems is attributed to their inherent redundancy, which is ultimately a consequence of self-replication/self-reproduction both at cellular level and organism level. Fascinated by robustness in biological organisms, in 1951, John von Neumann [18] introduced the concept of artificial automata to design self-replicating computing machines. He described five distinct approaches, illustrated below, for the realization of an artificial self-replicating automaton([18], pp. 91 - 99). The terms, self-replication and self-reproduction, have been interchangeably used throughout the chapter.

1. **Kinematic Model:** In a series of lectures delivered in year 1949 at the University of Illinois, John von Neumann explained his vision of a physical non-biological self-replicating system, termed kinematic self-reproducing automaton model. The model illustrates an abstract design of a hypothetical machine using a large number of physical components (spare parts) as its source of raw material. The machine uses instructions stored in a memory tape to fetch the physical

[1] State change in the CA model occurs synchronously and in discrete time, while DNA tiles asynchronously assemble in continuous time to form structures.

components using a manipulator, assemble them into a replica of itself, and then copies the instructions from memory tape into the newly created replica. A possible prototype of such machine was illustrated using eight components, that included four logic components (sensors and switches) to send and receive stimuli, and four muscle-like components along with joining and cutting tools to form flexible mechanical structure. Using physical as well as electronics components, Von Neumann identified the minimal requirements for such a centralized self-reproducing machine (akin to a robot constructing another robot), but he faced difficulty in analyzing it with mathematical rigor. Therefore, he came up with a new abstract model for self-replication based on CA model, described below.

2. **Cellular Automata Model:** Following input from his mathematician friend, Stanislas Ulam, John von Neumann developed the CA model, which was easy to analyze mathematically. A cellular automaton consists of a grid of cells that interact locally with their neighbors. Each cell in the cellular space has a state that is updated based on some fixed local rule (generally a mathematical function) defined by current state of the cell and the states of the cells in its neighborhood. An initial configuration of cells (at $t = 0$) is selected to start with, and as the time advances in discrete steps ($t = 1, t = 2$ and so on), cells update their states simultaneously to simulate self-organizing behaviors. Typically, the local rule in a CA is applied synchronously to all the cells simultaneously, though exceptions are known, such as the stochastic cellular automaton [147] and asynchronous cellular automaton [29].

   Von Neumann developed a complex design of self-reproducing automaton, where each cell has 29 possible states [18]. Due to the extremely large size of such an automaton, its physical implementation was not possible, and moreover its simulation was also not feasible with the computing power available then. However, successors of von Neumann [30] later simplified the design of the automaton, which finally led to the development of Langton's loop [79]. The Langton's loop uses only 8 states in a 5-cell neighborhood, thus limiting the transition rules to just a few hundred. This drastic simplification was achieved by giving away the necessary condition of universal construction used by von Neumann. Langton loop opened the door to an interesting research area, called ALife (an acronym for Artificial Life).

3. **Excitation-threshold-fatigue Model:** As a further refinement to the self-reproducing automaton with 29 states, von Neumann envisioned its implementation using neuron-like elements. He never defined the details of these elements, but the idea behind this model was to use simplistic neuron-like functions for the realization of the automaton.

4. **Continuous Model:** To introduce continuous time dynamics into the CA model that otherwise is discrete, von Neumann used differential equations to describe the process of a self-reproducing automaton. There are not many details available for this model, except that differential equations were used to describe

the excitation, threshold and fatigue properties of neuron-like elements of the previous excitation-threshold-fatigue model.

5. **Probabilistic Model:** One of the remarkable features of biological systems is their capability of undergoing mutation and evolving. Von Neumann wanted to introduce probabilistic transition rules in CAs so as to enable mutation and evolution in the artificial self-reproducing automaton that he was studying at that time. It is not clear how he would have realized such systems, but the concept reminds as of today's genetic algorithms [54].

Research in artificial self-replicating systems over last 65 years can be categorized into two major tracks, that are, the kinematic approach and the cellular automaton approach. A selective and non-exhaustive lineage of self-replicating systems and models is given in Figure 3.1, showing both the conceptual link and the chronological development. An exhaustive literature review of the field can be found in [137] and ([45], pp. xviii).

The first track covers artificial self-replicating/self-reproducing systems in CA [18] environment. Following complex design of von Neumann's self-replicating automaton, Codd [30] tried to simplify it, which later led to the development of much simpler Langton's loop [79]. This was subsequently followed by several other abstract designs of self-replicating loops [81, 27, 28]. Mostly, research in this direction have produced simulation studies of self-replication behavior using abstract models, independent from their physical implementation.

The second track covers kinematic self-replicating systems in which the physical system, not merely the pattern of information, is assembled either autonomously or with the help of a central manipulator that controls and coordinates the assembly process. The review presented herein covers only self-assembly based self-replicating systems. However, self-replicating machines with a central manipulator have also been realized. This approach is very well explained in a remarkable book [45], where self-replication is used as a paradigm for efficiently exploring other planets. However, this kind of self-replication, using a central manipulator, is beyond the scope of this thesis.

3.3 Self-replicating Cellular Automaton Models

3.3.1 Basic Definitions

After introducing the CA model briefly in Section 3.2, basic definitions of commonly used terms in CA are given below.

Grid: It refers to an infinite or finite array of cells usually in one-dimensional, two-dimensional, or three-dimensional cellular space. A two-dimensional finite grid is shown in Figure 3.2(a),

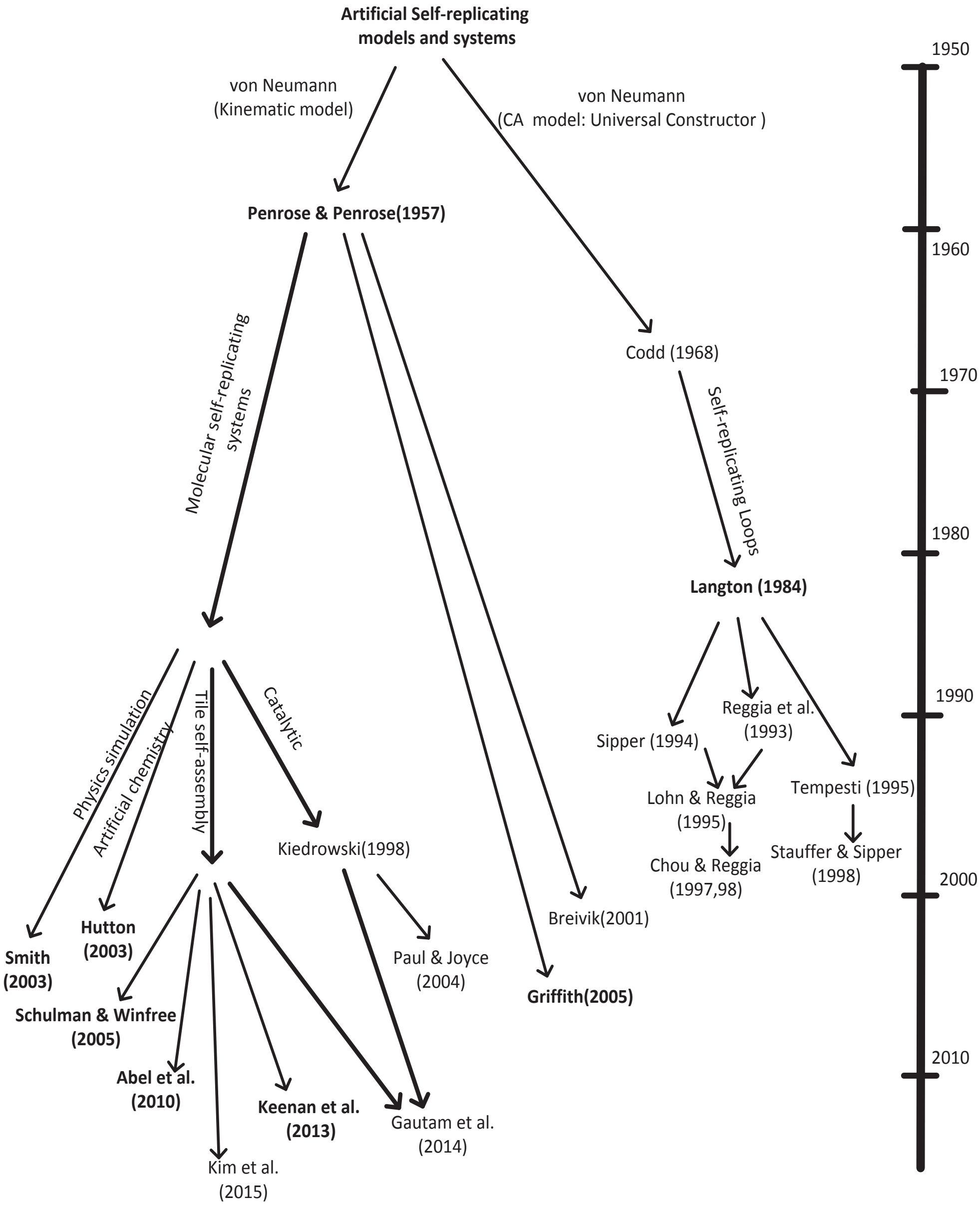


Figure 3.1: A brief, selective and non-exhaustive history of self-replicating models and systems.

Configuration: In CA notation, a configuration is an array of cells at a given time. Typical, the term refers to the states of all the cells in a CA grid. For example, an initial configuration refers to the states of cells in the grid at time t=0, as shown in

Figure 3.2(a).

State: Each cell within CA space is allowed to be in one of the finite number of states. The simplest case is a CA with binary state, where each cell can be either in state ’0’ or state ’1’. In visual representation, an CA with binary states is shown in Figure 3.2(b), where a filled cell and a blank cell are equivalent to states ’1’ and ’0’, respectively. In a complex CA, cells may have multiple states, where state transition is determined by local update rule in discrete time.

Neighborhood: The CA model uses the concept of cell-neighborhood to update the states of cells in each time step. The neighborhood of a cell consists of its surrounding (adjacent) cells, as shown in Figure 3.2(d). For one-dimensional CAs, a cell is connected to k local neighbors (cells) on either side, where $k/2$ is a parameter referred to as the radius (thus, each cell has k+1 neighbors, including itself). For two-dimensional CAs, two types of cellular neighborhoods are usually considered: 5 cells, consisting of the cell along with its four immediate non-diagonal neighbors, and 9 cells, consisting of the cell along with its eight surrounding neighbors.

Local Update Rule: State transition of each cell in an CA space is governed by local update rule, that is derived based on the states of its neighborhood cells, including its own state. Rule 90([160], Chapter 3) is one of the elementary rules in CA. In this rule, each cell’s new state is the XOR of the states of two neighboring cells, as illustrated in Figure 3.2(b).

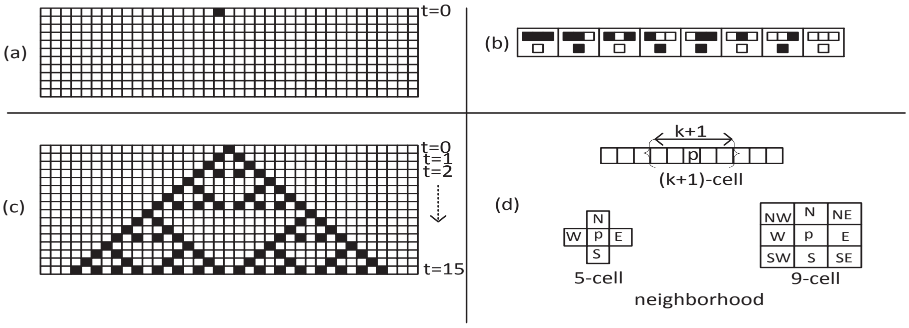


Figure 3.2: Schematic reprepresentation of cellular automaton.(a) An initial configuration (t =0) of two-dimensional grid. (b) Representation of one-dimensional CA rule 90 ([160], Chapter 3). (c) Nested pattern of Sierpinski triangle fractal produced by applying the Rule-90.(d) Neighborhood in CA: k+1 neighbourhood in one-dimensional CA, 5-cell neighborhood, and 9-cell neighboorhood, where N, E, S, W, NE, SE, SW, NW represent cells in the neighborhood of present cell, p.

3.3.2 Von Neumann's Universal Constructor

Von Neumann's self-reproducing cellular automaton is a universal constructor that can build any machine given its description. Therefore, if the universal constructor is given its own description, it would produce its own copy — self-reproduction. At a high level, as shown in Figure 3.3, the automaton consists of three main components: the memory controller, constructing unit, and constructing arm. The memory controller is capable of accessing the description of machine to be constructed. The construction unit uses this description, interprets it to control and coordinate the actions of constructing arm that builds the machine. In the case of self-reproduction, the description of universal constructing machine stored in memory (M) is finally copied into the newly constructed machine.

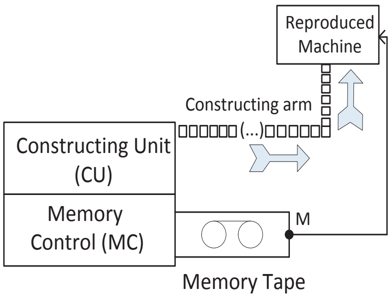


Figure 3.3: Von Neumann's self-reproducing machine

Von Neumann demonstrated the design of a universal constructor using CA. The universal constructor automaton consists of several thousand cells and it starts in a specific initial configuration. There is another cellular grid representing the memory tape, which is read by the constructing the automaton. Each cell in the automaton has 29 states and a 5-cell neighborhood is used to make a state transition. The complexity of this automaton can be imagined by the number of transition rules ($29^5 = 20,511,149$) required to exhaustively define its potential behavior. An automaton with this much complexity could not be simulated by the computers available at the time of von Neumann. Even today with modern computing resources, it would take several weeks to simulate the operation of a universal constructor to make a single copy of itself and its tape.

3.3.3 Langton's Self-replicating Loop

Trying to reduce the complexity of von Neumann's self-reproducing automaton, Edgar F. Codd in 1968 demonstrated that a similar universal constructor can be implemented in a cellular automaton with just 8 states [30]. This is a substantial reduction of the complexity in comparison to Von Neumann's self-reproducing automaton of 29 states, but the automaton still required 100, 000, 000 cells, thus, making it again infeasible for simulation or implementation.

Both von Neumann's and Codd's self-reproducing automaton use universal construction as a necessary condition for self-replication. Christopher Langton, however, posed the converse question: how simple an automaton can become to still be able to reproduce itself [79]? This led to the development of Langton's self-replicating loop, that uses just 86 cells and 8 states to self-replicate the loop.

Langton's Loop and its self-replication dynamics is illustrated in Figure 3.4. In its initial configuration, as shown in Figure 3.4a, the loop consists of a data path shielded by a square sheath of red colored cells, and a constructing arm. The data path consists of a sequence of instructions that direct the constructing arm to build the new loop while the instructions also get copied into the newly forming loop. The data path consists of *core* (cell with state '1') and *signal* (cells with states '4' and '7'). The signal states, '4' and '7', are followed by state '0', and form two important instructions: Advance Instruction (07) and Turn Instruction (04). The signals always travel in the direction of the signal states, i.e., $0 \rightarrow 7$ and $0 \rightarrow 4$. The Advance Instruction tells the constructing arm to advance by one step, while the Turn Instruction directs the arm to turn 90 degrees to the left.

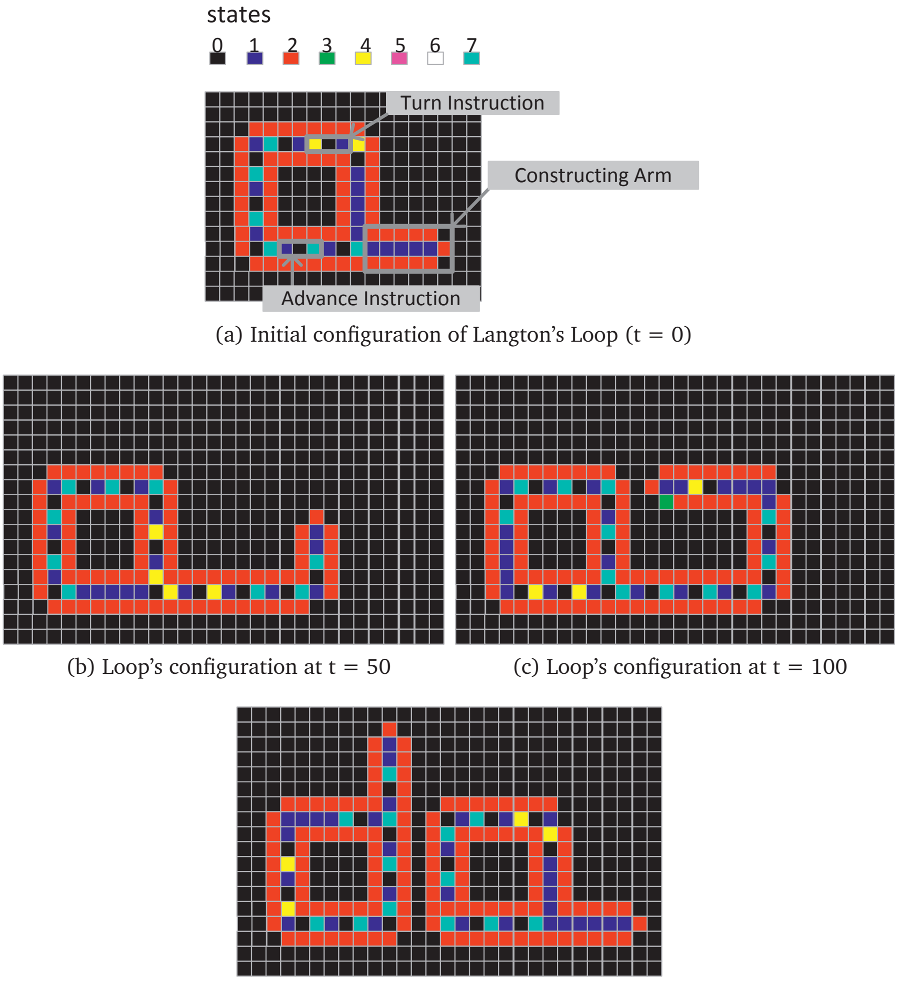
A discrete time dynamics of Langton's Loop indicating important steps in self-replication is illustrated in Figure 3.4. With time, signals travel along the data path as the instructions are executed. Signals arriving at a branching point are duplicated and proceed in both directions, as shown in Figure 3.4b. At t = 100, the arm is ready to take a third turn (indicated by state '3'), as shown in Figure 3.4c. After three turns the arm has looped back on itself: the new data path reaches the old. This causes the two loops to separate, which further grow their construction arms, as shown in Figure 3.4d, and get ready to build new copies. A loop completely surrounded by other loops or reaching to the edge of the cellular grid would however die because all cells within the loop will turn to core cells, and therefore there would not be any instructions within the data path of the loop.

3.3.4 Self-replicating Loops from Langton's Successors

After Langton's work, John Byl came up with a simpler self-replicating loop without an inner sheath layer. Byl's Loop [19] consists of 12 cells, where each cell can be in one of the 6 states (0-5). The Loop uses 43 transition rules, 5-cell neighborhood, and reproduces itself in just 25 time steps.

Reggia et al. [108] further reduced the size and complexity of self-replicating loops by removing the surrounding sheath completely from the data path. These *unsheathed loops*, also known as Chou-Reggia Loops use rotational asymmetry [2] in the cells that

[2] Codd's [30] and Langton's [79] work used strong rotational symmetry in all cell states, that is, cells have no orientation. To understand the concept of rotational symmetry, let us consider a cell in the 5-cell neighborhood, as shown in Figure 3.2(d). Let us define the cell's transition rule as $NESWp \rightarrow p'$, where



(a) Initial configuration of Langton's Loop (t = 0)

(b) Loop's configuration at t = 50

(c) Loop's configuration at t = 100

(d) Replication is complete (t =150)

Figure 3.4: Langton's Loop and its self-replication dynamics. (a) shows the initial configuration of the loop. (b), (c), and (d) represent the self-replicating loop's dynamics at t= 50, t= 100, and t=150, respectively.

significantly reduces size of the replicating loop, algorithmic complexity, and time steps required to reproduce the loops. The size of the Chou-Reggia Loop is 5 cells and it uses 8-states in a 5-cell neighborhood. The reproduction time is drastically reduced to 15 time steps.

NSEW are the states from neighboring cells (considered in clock-wise), and p and p' are the present state and the next state, respectively. A cell with rotational symmetry would have distinction between the rules $NSEWp \to p'$ and $WNESp \to p'$ because the cell is unoriented. However, in the case of a rotationally asymmetric cell, these rules can be associated with two distinct orientations of the same cell [108]

One of the limitations of Langton's Loop lies with the absence of construction capability, which was one of the main goals of the von Neumann's automaon. To overcome this limitation, G. Tempesti [145] designed a self-replicating loop with construction capabilities. Further, the loop also removes Langton loop's requirement of an infinite cellular space. Tempesti's Loop uses 10-states, 9-cell neighborhood, consists of 148 cells, and reproduces itself in 304 time steps. In contrast to the Langton Loop, Tempesti's Loop allows patterns to be written inside the loop, thus enabling functionality to be added into the self-replicating loop. Similarly, Perrier's Loop [68] also enables functionality to be added in the self-replicating loop, but at the cost of large complexity added to the automaton.

Sayama [117] introduced structurally dissolvable self-reproducing (SDSR) loop. Sayama's model produces colonies of self-replicating loops inside a finite but large cellular grid. Loops in this model get dissolved when approaching obstacles, such as reaching the boundaries of the CA grid or other loops. The model produces dynamically stable populations of self-replicating loops, thus enabling evolvable behavior. Later, Sayama introduced a further extension called evoloop [118], which introduced the concept of evolution and competition for space.

Table 3.1: Comparison of different Self-replicating Loop Models.

Loop	No. of states	Neighborhood size	Loop size	Replication period
Langton's loop [79]	8	5	86	151
Byl's loop [19]	6	5	12	25
Chou-Reggia loop [108]	8	5	5	15
Tempesti loop [145]	10	9	148	304
Perrier loop [68]	64	5	158	235
SDSR loop [117]	9	5	86	151
Evoloop [118]	9	5	149	363

In the above review of cellular automata and self-replicating loops, I have illustrated how the first artificial self-reproducing automaton was introduced by von Neumann. I have further discussed how this work was pursued by Codd and finally Langton, leading to the development of the Langton Loop, which demonstrates self-replication without using a complex automaton. This was achieved by omitting the necessity of universal construction for self-replication. Finally, I have discussed other self-replicating loops ([19, 108, 145, 68, 117, 118]) studied by Langton's successors. In the end, a comparison of different self-replicating loops is illustrated in Table 3.1.

Cellular automata have produced interesting models for studying self-replication and life-like properties in artificial systems. However, all of these models are merely abstract mathematical representation of aspects of real world processes. For example, CA models advance in discrete time, while the real world is continuous. Likewise, CA models are typically synchronous, while physical processes are asynchronous and parallel. Further, CA models do not consider the laws of thermodynamics. For example, the aforementioned self-replicating loops do not need an energy source to maintain their structure and keep producing new copies of themselves. Such perpetuating mobile phenomena without the consideration of energy input and output to the environment are clearly impossible in the real world. Therefore CA models should rather be seen as interesting abstract mathematical models, which provide insight into how self-organization and emergent phenomena arise in general. For more realistic models of self-replication and life-like systems, more complex models are needed that take these aspects into account .

3.4 Kinematic Self-replication

The following discussion reviews work in kinematic self-replication, illustrated in Figure 3.1. This includes the self-replication models that use components to autonomously reproduce physical copies of given templates. We first describe the first realization of a kinematic self-replicator of macroscopic structures by Penrose and Penrose [102]. Then we briefly illustrate physical self-replication systems inspired by Penrose's model that are recently demonstrated by Saul Griffith [58]. Following is the review of molecular self-replicating systems: 1) JohnnyVon [138, 41], a kinematic self-replicating model that uses simulated physics of molecular components as movable CA; 2) Squirm3 [67], a self-replication model using artificial chemistry; 3) Non-enzymatic self-replication systems of Nucleic Acid molecules; and 4) tile pattern self-replicating systems recently studied in the tile self-assembly framework.

3.4.1 Penrose's Self-replicator

Lionel Penrose, with his son, Roger Penrose presented the first physical implementation of self-replication[3] in 1957 using mechanical components made of plywood. It is a widely referenced example of both self-assembly and self-replication, and models range from simple one-dimensional to very complex two-dimensional. For the sake of simplicity and ease of understanding, a one-dimensional self-replication model is illustrated.

Penrose defined self-replication as a process whereby a physical system autonomously produces new copies of itself which are identical to the structure and size of the original, when the entire system is put together in a suitable environment. Penrose's first prototype of a macro-scale physical self-replicating system consisted of two types of plywood components (A and B), as shown in Figure 3.5. The components are placed inside a one-dimensional groove that gives limited freedom of movement: the components inside the grove can only shift or tilt to the left or right when the system is shaken. The components of types A and B have convex and concave surfaces,

[3] The term, self-reproduction, was used in the original article by Penrose [102]. However, following Sipper's definitions [137] of self-replication and self-reproduction, the term self-replication seems more appropriate in this context

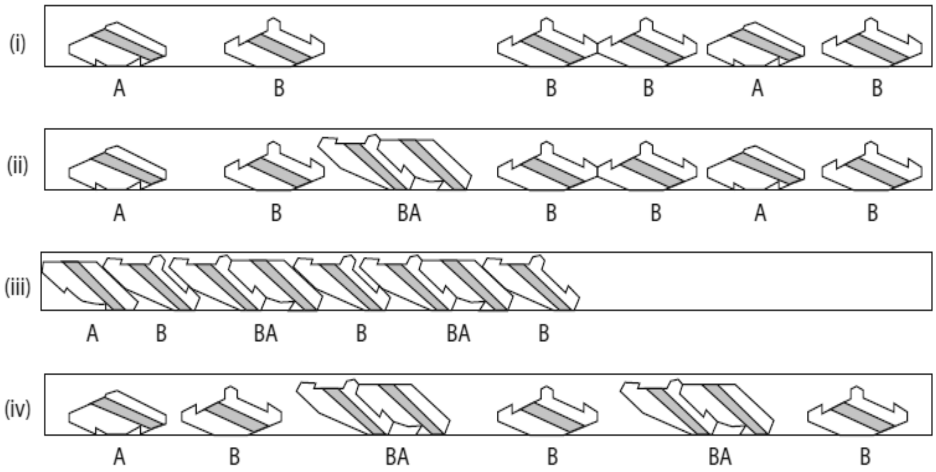


Figure 3.5: Schematic representation of Penrose's one-dimensional physical self-replicator [102]. (i) Two types of components: *A*, *B*, kept in a random order inside a groove (ii) A pre-formed 2-bit (*BA*) seed structure is introduced in the groove. (iii) The system is shaken horizontally. (iv) As the agitation subsides, weakly linked components break apart, while the seed and a new replica (*BA*) is formed.

respectively. Thus, no two components of the same type — AA or BB — can remain bound after agitation is removed. However, tilted components of opposite surfaces, such as an A colliding with an B, join together. Observe that by design the components inside the grove can not tilt automatically, unless a pre-formed BA (or AB) structure is present as a seed inside the grove, as shown in Figure 3.5(ii). The components colliding with the seed structure get tilted, and therefore tilted components that collide in the order *B*-*A*, join stably, as shown in Figure 3.5(iii). A new copy, BA, is produced as the agitation of the system is removed, as shown in Figure 3.5(iv). Due to mechanical symmetry in the components of the system, an AB seed would similarly produce new copies of AB structures.

Requirements for Physical Self-replication Penrose elucidated the following minimal set of requirements for the realization of a physical self-replication system:

1. Each component must have at least two states : *inactive* and *active*. Initially, components are in their inactive states. Inactive components do not form stable links, thus there is no spuriously assembled structure at the start. However, if an inactive component collides with an active component, the two get linked stably. Thus, if a pre-formed seed structure is introduced, an inactive component colliding with it will form stable link.

2. The system should not form any periodic assembly of components. This demands that components should have definite boundaries, and components of the same type should not form stable links: an inactive *A* type component should not form a stable link with an active *A* component when the two collide.
3. The self-replication process requires a driving force, that is, there should be a net energetic stabilizing contribution to the system, every time a component forms a stable link. For example, in the Penrose's self-replicator, the kinetic energy supplied by agitating the system is converted into the form of potential energy of linking two activated components.
4. An activated component should transfer activation to the next colliding inactive component so that the structure grows as more and more inactive components join by changing their states from inactive to active.
5. The self-replication environment should favor desired interactions between the components. In the one-dimensional case, grooves constrain the movement of the components so that they can collide more often in the desired orientation and form stable links.

The mechanical symmetry in the components of the aforementioned self-replicating system puts a limitation over the degree of complexity of seeds that can be self-replicated. To enable a desired degree of complexity in the self-replicating seeds, a new design of asymmetrical components was introduced, as shown in Figure 3.6. By rotating an asymmetric component by 180° on its vertical axis, the second component is obtained. The two such components can be designed for the implementation of self-replicating system of arbitrary seed structure in one-dimension.

A self-triggering 'hook-release' mechanism is introduced further in the system so as to autonomously dissociate the replica structure its seed. One such self-replication system is illustrated in Figure 3.6. The seed structure (a) consists of a double-column of five components, stacked one over the other (components at the bottom are not shown). When the unit on the left assembles with the seed (b), conformation between the seed and the joining unit triggers switching in one of the hooks holding the seed structure together. Further, as the unit on the right joins the seed (c), the second hook switches its state. This causes dissociation of the four-column structure into two copies of the original(d).

Using the inspiration from enzymatic allostery, Saul Griffith [58] constructed dynamically switching components that can be reversibly latched and unlatched as they interact locally. Each component consists of electromechanical switches that are controlled by an on board controlled designed as a state machine of 7 predictable states. The components self-assemble while floating on a two-dimensional air table. Using a set of such switching components, Griffith implemented a self-replicating system of an arbitrary 5-bit string of the components.

The mechanism of Griffith's self-replicating system can be understood by a CA-based abstract model, as shown in Figure 3.7. Each CA cell has four edges (square unit) and three neighbouring cells that are used to update the current state of the cell. An

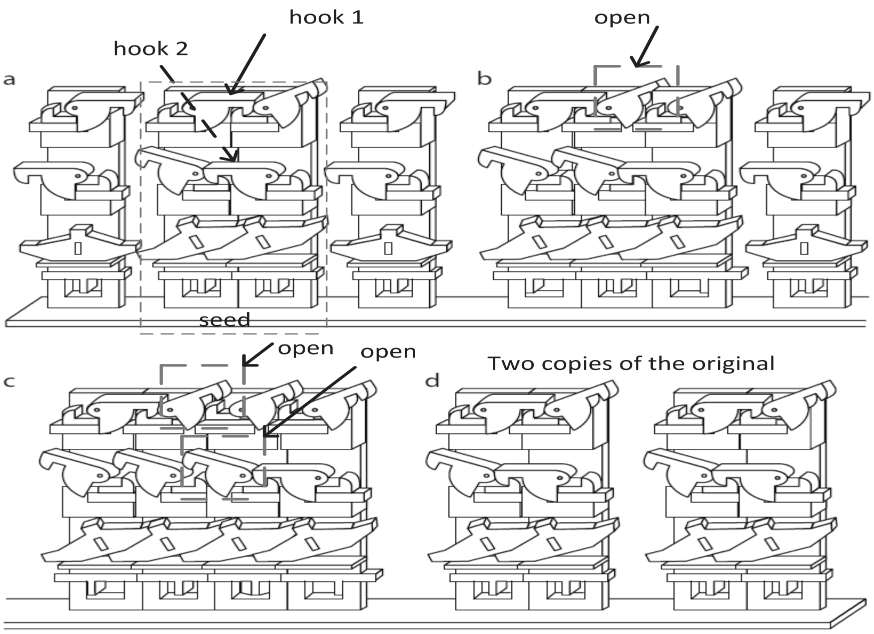


Figure 3.6: Penrose's self-replication system in 2-D using conformational swtching blocks [103].

arbitrary 5-bit string of seed structure is represented by a row of five interconnected CA cells lying on the upper row (see, state 1). The interconnecting link (latched state) between components is shown by state '1' on the corresponding edges of the cells. The second row represents sequential self-assembly of the replica of the seed as components start attaching from the left end of the seed. A free floating component has all its electromechanical switches turned off, and is represented by a state (0, 0, 0, 0) of CA cell. As a free floating component comes in the near communication range of the left component that has its bottom switch turned on (in state 1, the bottom edge of the first cells in the upper row is marked by '1'), it gets attracted towards it. The electromechanical switch of the approaching component is set into the latch mode (see, state 2, where cell state is changed from '0' to '1'), and the first component replica get added. Following this, the next component in the seed turns its bottom switch from off-to-on (see, state 3, where bottom edge of the second cell changes its state from '0' to '1'). Hence, a new active site is created that will attract a new free-floating component, and as the component comes into the near communication range, it get attached to the second component of the seed (see, state 3 and 4). In the state 5, similarly the third component approaches the newly activated component of the seed. In state 6, three changes happen: 1) the first component of the replica that is latched with the seed, is unlatched; 2) the first and second component of the replica gets latched; 3) simultaneously the third component in the replica is latched to the seed. As this process continues further (see, state 10), the complete replica is assembled, and in next few states (not shown here), it will get dissociated from the seed.

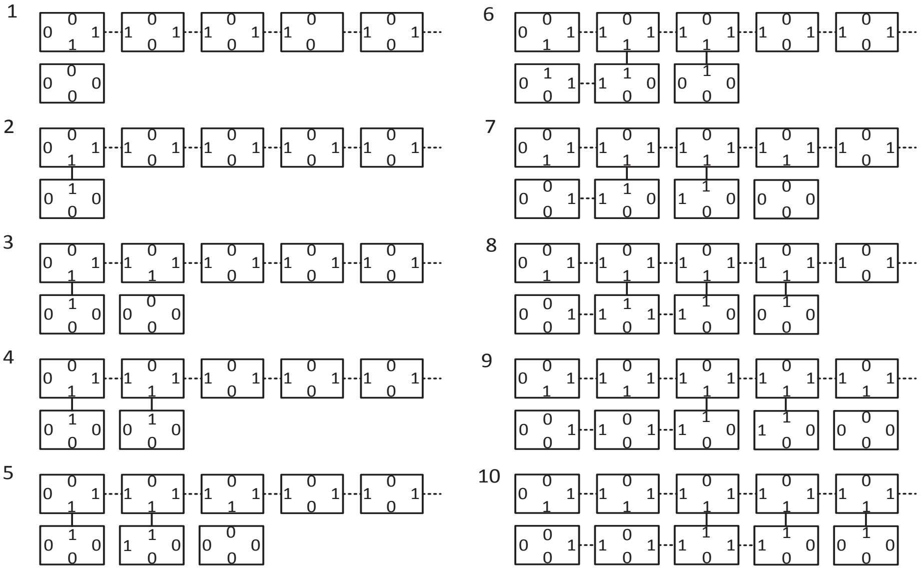


Figure 3.7: Cellular automaton representation of Griffith's self-replication system [58].

3.5 Molecular Self-replicating Systems

Self-replication in biology exploits rich chemistry of biological molecules that enable spontaneous self-organization of molecules. Over the past few decades, our understanding of physico-chemical properties of biomolecules, their synthesis and controllability in the laboratory setups, and mechanisms to program larger assemblies have increased. This has led to interesting developments in the field of nanotechnology, where programmable molecular orchestration is seen as a promising route for the realization of larger complex systems and behaviors using synthetic molecules.

Recently, molecular self-replicating systems have been studied in simulations and real implementations: simulated physics models of molecules [138, 41], artificial chemistry models [91, 67], and catalytic models of real chemical molecules [134, 72]. A review of the work in molecular self-replicating systems is given below.

3.5.1 JohnnyVon: Self-replicating Model Using Simulated Physics of Molecules

In a synchronous CA, the state transition of all cells occurs synchronously in discrete time. Such models are far from the realism of the physical world that is inherently continuous. To model automata dynamics with continuous space and virtual physics, JohnnyVon [138, 41] uses physically plausible and molecular units that interact locally in continuous space. The model uses two types of molecular units and a simulated Brownian motion allow them to drift randomly in two-dimensional space. The molecular units are called “codons” and comprise T-shaped virtual objects that possess both attractive and repulsive fields. The codons are similar to mobile automata having discrete internal states that can be set by finite state machines. Codons interact in continuous space governed by simulated force fields. Codons can be used to spontaneously self-assemble arbitrary sequences of linear template structures. However, if a pre-assembled template is introduced, it attracts floating molecular units to self-assemble its replica. The template-replica structure dissociates into the template and a replicated copy, thus the system evolves to form several copies of the template-like linear structures as time advances. The replicated copies further self-assemble to form larger nanostructures.

JohnnyVon and Penrose’s self-replicating system both autonomously reproduce given structures using simple components. However, there are certain crucial distinctions between the two. First, Penrose’s model uses macroscopic components to reproduce copies of given template, while JohnnyVon is a simulation model of self-replication based on molecular scale components. The components used in the Penrose’s model are symmetric, therefore limiting its use for replication of templates consisting of more than two components. Simulation model of JohnnyVon do not have such limitation, and therefore it is suitable for the self-replication of larger size templates. Second, Penrose’s model requires physical implementation of the replication process, making it difficult to execute experiments. On the other hand, the simulation model of JohnnyVon can be easily used to study the self-replication dynamics and analyze the system’s behavior using parameter variation.

3.5.2 Squirm3: Self-replication Using Artificial Chemistry

The JohnnyVon [138, 41] is a computational model that simulates behavior of continuously interacting molecular-like units. Due to use of physics and force fields in modeling, JohnnyVon is more suitable to study nanomanufactruing processes, such as construction of nanobots and nanostructures. Researchers [67] however more interested in the investigation of principles, such as self-organization and evolution, developed *Squirm3* [67] that is a chemical model based on Artificial Chemistry (AC) [37, 39].

The AC is an abstract model of chemical reactions. It has been used for the modeling of complex chemical behaviors using molecular-like abstract units. With a simple set of a reaction toolbox in AC, a variety of chemical behaviors, such as self-organization, self-maintenance, and self-construction have been studied [37, 39, 67]. While AC-based models are far from real chemistry of natural processes, they do give insights into some of the design constraints of molecular-like units in realizing larger chemical systems.

The Squirm3 simulation uses a AC to support self-replicating molecules. Virtual atoms occupy cells in a two-dimensional discrete grid space. The atoms move randomly in the grid space, presumably due to virtual Brownian motion, like the particles in JohnnyVon. When two atoms occupy adjacent cells, they may form a bond with each other, again like the particles in JohnnyVon. If a seed molecule, consisting of a string of bonded atoms, is placed in a soup of free atoms, it will replicate itself by a series of virtual chemical reactions with the free atoms.

Both Squirm3 and JohnnyVon use molecular-like components and local interaction rules to model complex behavior, however, there are several distinctions between the two models: first, Squirm3 uses virtual chemistry, whereas JohnnyVon uses virtual physics; second, Squirm3 uses a discrete grid space, whereas JohnnyVon uses a continuous space; third, basic components in Squirm3 are virtual atoms, whereas basic components in JohnnyVon are virtual machines (e.g., nanobots).

3.5.3 Non-enzymatic Self-replication of Nucleic Acid Molecules

Inside biological cells, DNA self-replicates by templating and catalyzing its own synthesis, assisted by a large number of enzyme macromolecules. Duplex DNA is cut selectively by enzymes and the two DNA strands are separated. Further, each DNA strand is used as a template for the enzyme-catalyzed assembly of a complementary DNA strand. Finally, enzymes separate the template and the complementary strand before the next round of replication.

Using biological inspiration of templating and catalysing its own synthesis, chemical self-replicating systems capable of transmitting or amplifying structural information contained in synthetic chemical molecules, have been widely studied in theory and experiments. The study of chemical self-replication has drawn interest from both biologists and synthetic chemists. Biologists use these systems to understand the emergence of prebiotic self-replicating molecules [72]. However, for the synthetic chemist, they represent autonomous systems, capable of manufacturing a large number of perfect copies of themselves from a single original molecule [134, 101].

Attempts to design enzyme-free simple chemical systems capable of self-replication have resulted in two minimal processes as illustrated in Figure 3.8.

The basic principle behind these models is non-enzymatic catalysis. In an auto-catalytic self-replication model, shown in Figure 3.8(a), a molecule serves as a template (T) to catalyze the formation of its own replica from a set of supplied precursors (A, B). Intermediate complex molecule, M, is ligated to form complex, D, which dissociates into the template and its replica. Auto-catalytic self-replication has the limitation that it can be used only for self-complementary templates as reported by von Kiedrowski [134]. However, such a limitation does not exist in natural self-replication of DNA inside cells, where template DNA strand is complementary rather than self-complementary. To realize the self-replication of both types of templates, a cross-catalytic model, shown in Figure 3.8(b), was introduced by von Kiedrowsky [134]. The cross-catalytic self-replicator consists of two template molecules (AA, BB) which form one another's replicas through two or more sets of smaller fragments (A, A', B, and B') of molecules [134].

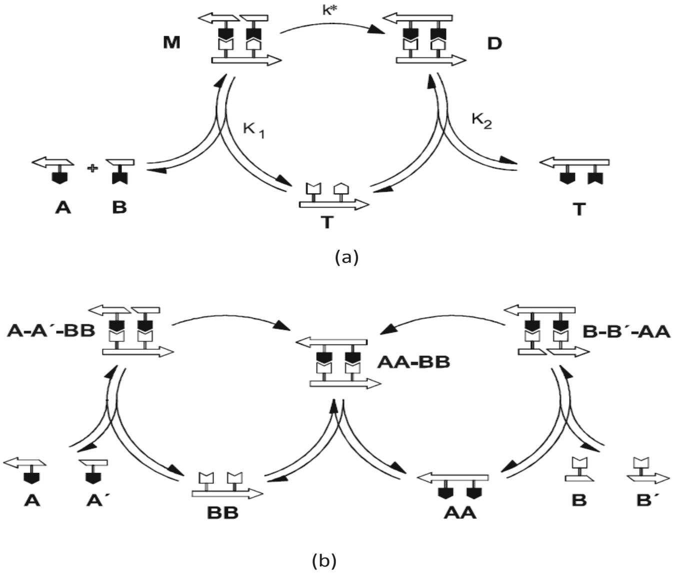


Figure 3.8: Minimal systems of non-enzymatic self-replication [100]. (a) autocatalytic self-replication cycle: two molecules AA and BB mutually catalyze each other's production through temporary base-pair formation. (b) cross-catalytic self-replication cycle: AA and BB are template molecules, where one catalyzes the production of the other. A, B, A', and B' are fragment molecules complementary to the templates.

3.6 Self-replicating Systems of 2-D Patterns of DNA Tiles

Self-assembly can be seen as an attractive route to designing artificial systems at the nanoscale with enormous precision and complexity. Engineering of synthetic chemical systems in 2-D and 3-D using biological principles, such as self-replication and reproduction is likely to be beneficial in several ways. First, it may open doors to cheaply produce completely new materials and nanomachines. Second, designing synthetic chemical systems using these principles may help to understand the fundamental processes of biological world.

One of the requirements for kinematic self-replicating systems is self-assembly: smaller components autonomously assemble to reproduce larger structures. The second requirement of such self-replicating system is a dynamic and autonomous dissociation of template structure from replicate. A self-assembling platform is thus necessary for such non-centralized self-replicating systems.

DNA-guided self-assembly has recently emerged as the most common platform for the design of complex nanostructures, including linear polymers [134, 101] and 2-D tile patterns [120, 6, 75]. Self-assembly of DNA tiles is one such approach that enables programmable pattern formation in 2-D. Earlier, tile self-assembly has been used to propose conceptual designs [120, 6] of 2-D pattern self-replication. However, tile self-assembly is inherently a static process, therefore such self-replicating systems proposed either external forced fragmentation [120, 128] or usage of enzymes [6] to separate templates from replicates. Recent work integrating dynamic control in the tile self-assembly [171, 114], has enabled the design of tile systems that can be dynamically assembled, disassembled, and reconfigured. Dynamically controlled dissociation of tiles has recently been used for the design self-replicating systems of 2-D patterns of tiles [75]. In the following, we review work in self-replicating systems of 2-D patterns of tiles realized in the tile self-assembly framework.

3.6.1 Self-replication of Combinatorial Patterns of Tiles

To explain how prebiotic self-replicator could have emerged, in 1966, Graham Cairns-Smith proposed a simple mechanism based on polytypic clay crystals that could replicate information sequences contained in the crystal layers [20]. Growth of polytypic clay crystals occurs layer-by-layer, where each layer constitutes molecules arranged in a particular orientation. If such a crystal eventually splits, for example by frictional forces of a rough surfaces, there could be several fragmented crystals, and thus multiple growth sites would simultaneously replicate the information-bearing sequence. This may result in an exponential rise in the population of the information-bearing sequences. Thus such a process might have implications for the origination of prebiotic self-replicators and primitive life.

Schulman and Winfree proposed a self-replicating system that uses Graham-Smith's approach of crystal growth and fragmentation in the tile self-assembly framework [120, 128]. Tile self-assembly has been used for the implementation of algorithmically programmed patterns of tiles [154], where self-assembling tile patterns could be designed to copy a particular linear arrangement of tiles during pattern growth. If such growing pattern is fragmented into parts, each new part of the fragmented pattern would have new growth fronts that will also copy the tile sequences in the subsequent growth. Repeating this process would therefore create a population of patterns that would ideally copy the same linear arrangements (tile assembly errors may introduce new copying sequences of tiles) of tiles and thus, an exponential replication of the tile sequences could be realized.

A self-replicating system using growth and fragmentation is illustrated in Figure 3.9

and Figure 3.10. Tiles have unique bonds that are marked by numbers on their edges, as shown in Figure 3.9(a). Matching edges in the tiles have the same number. As the pattern grows, alternating tiles in each row enforce the assembly of square shape tiles on the top and bottom layers, as shown in Figure 3.9(b). Tiles are designed such that each tile must make two correct matches to get attached stably in the growing pattern. Furthermore, a tile can not attach without matching the label on its horizontal neighbor, and thus ensuring that each incoming tile in the new column must comply with the one in the previous column. As a result, the two inner rows of the pattern copy a particular 2-bit sequence of tiles in each column. Using tile types shown in Figure 3.9(a), only one type of tile sequence can be produced as tiles self-assemble. Adding four types of more tiles in the previous tile system, as shown in Figure 3.9(c), four different 2-bit tile sequences can be copied. This could be generalized for larger tile sets, where copying of 2^n tile sequences of width n can be implemented.

To increase the rate at which new copies of the tile sequences are produced, the growing zig-zag patterns are split into pieces by external forced fragmentation, as shown in Figure 3.10. Each new piece of the pattern that is created by fragmentation has new growth fronts (GF) that become available to copy the tile sequence. Repeating the cycles of growth and fragmentation produces exponential rise information-bearing initial sequence of tiles. Due to error-prone process of tile self-assembly, sometimes erroneous tiles get assembled into the copying tile sequences. Such erroneous tile sequences introduce interesting replication dynamics into the self-replicating systems that can be used for studying evolution in simple tile self-assembly systems.

3.6.2 Self-replication of Tile Patterns Using Enzyme-based Dissociation

Abel et al. [6] introduced a conceptual design of self-replicating system of 2-D patterns in the tile self-assembly framework that uses DNA tiles, RNA tiles, and RNase enzymes. The RNA tiles are used to form a layer on the periphery of the given DNA tile pattern. DNA tiles are subsequently used to form a frame covering the layer of RNA tiles. The layer of RNA tiles is then dissolved by the addition of RNase enzyme that releases the pattern from the frame. The self-replication mechanism is applicable to any shape in 2-D, and can be tailored to produce both a precise number of copies and an infinite number of copies of the given pattern.

Producing a precise number of copies of a given tile pattern: The mechanism of self-replication consists of two phases. The first phase involves stages of formation of a double layer of RNA (in direct contact with the pattern) and DNA tiles (out layer). The RNA layer is dissolved using RNase enzyme and DNA frame is created. The second phase involves several stages that produce multiple copies of the pattern. In the first stage, a double layer of RNA tiles is grown on the inner side of the DNA frame. In the second stage, a new pattern, P1, is grown inside the frame. In the third stage, the double layer of RNA tiles is dissolved to release the pattern, P1. In the fourth stage, P1

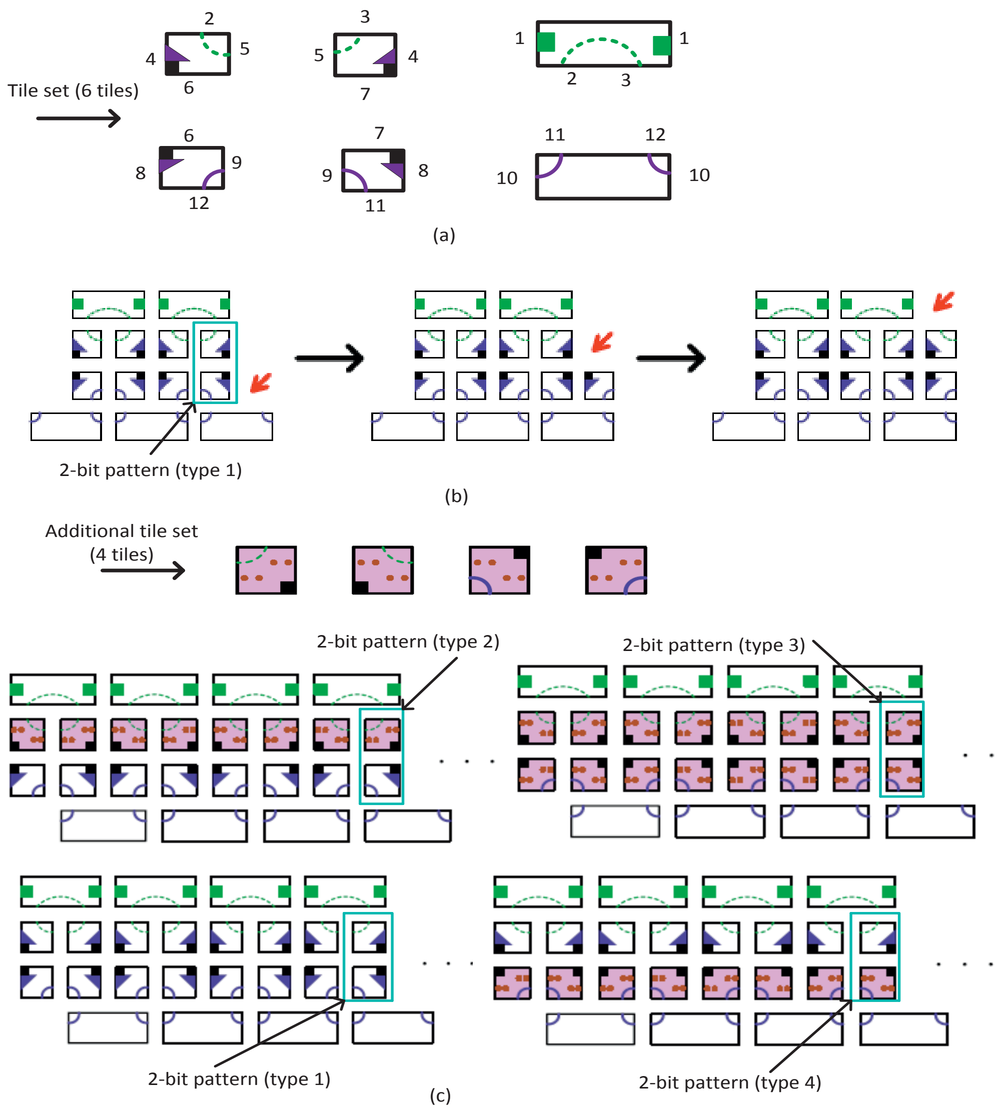


Figure 3.9: Self-assembly of zig-zag patterns of tiles. (a) Zig-zag tile set. (b) Growth of zig-zag pattern of tiles. (c) Growing multiple combinations of tile sequences in zig-zag patterns.

patterns are appended with single layer of DNA tiles that reproduces an exact replica. In the last stage, DNA frames can be terminated by adding a layer of DNA tiles on the inner side, thus halting the self-replication process.

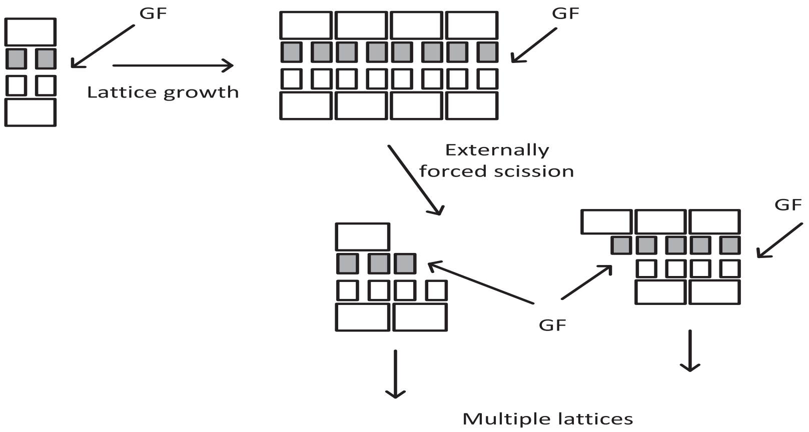


Figure 3.10: Self-replication of combinatorial sequences of tiles: schematic representation of pattern growth followed by externally forced scission that creates multiple growth fronts in the fragmented copies of the patterns.

Producing infinitely many copies of a given tile pattern: The mechanism of producing infinitely many copies of a target pattern involves several stages. First, a layer of RNA tiles is grown around the pattern. The RNA layer is extended at the edges of the pattern to form upward and downward columns. Second, using DNA tiles space between RNA columns is filled to bring the entire structure in rectangular shape. Third, sections between the columns are further grown with unique counting sequences of DNA tiles, while RNA columns are extended along the borders of the counters. Fourth, the RNA layer is dissolved using enzyme, which releases the pattern and structures of unique counting patterns. Fifth, infinitely many copies of each counting structure are grown in parallel. In the end, using these counting structures, infinitely many copies of the pattern are assembled.

As it can be seen from the general description of the self-replication mechanism that involves several stages of complicated operations. To realize such self-replicating system in practice would be a challenge, additionally, highly restrictive nature of enzyme usage could pose more limitations. Nonetheless, the conceptual self-replicating mechanism is capable of shape-independent replication applicable to both a precise and an infinite yield.

3.6.3 Template-directed Self-replication of Rectangular Patterns of Tiles

Keenan et al. [75] presented a template-directed self-replicating system of 2-D rectangular patterns in tile self-assembly framework, illustrated schematically in Figure 3.11. The self-replication model uses asynchronous signal passing tiles to autonomously dissociate templates from their replicates. Asynchronous signal passing tiles have been proposed in Signal Tile Assembly Model (STAM) [95]. The STAM is derived from the aTAM [154, 113] where tiles are modified to enable signaling. The STAM uses tiles with asynchronous signaling, that is, DNA signals are passed to asynchronously change the state of a tile glue. For example, an attached tile could be dissociated by turning off its glue or it can be instructed to bind to a particular site by turning on its glue. Typically, signaling reactions used in the STAM are: break reaction, combination reaction, glue-flip reaction. A signal propagates through the tile lattice by sending controlled signals to activate and deactivate binding between two connected or remotely placed tiles.

A signal-controlled dissociation of tile assemblies forms the basis of the formation of replicates of the template and later to trigger autonomous dissociation of the two.

The template-directed self-replication mechanism of rectangular patterns is quite similar to Kiedrowski's [134] model of non-enzymatic self-replication, described earlier in Section 3.5.3. However, the two differ in the types of replicates that are produced. A replicate could be non-terminal (ntr) or terminal (tr): a ntr can be used to catalyze the formation of more products, while a tr would produce an inert final product. Kiedrowski's model was designed to self-replicate linear polymers of nucleic acids, thus it produces only non-terminal (ntr) replicates that are used further to produce more ntrs. The pattern self-replicator however produces both ntr and tr replicates, where trs are used to produce replicated patterns and ntrs serve as templates for the formation of more ntrs and trs in subsequent cycles.

Template-directed Self-replication process of rectangular patterns is illustrated in three phases, as shown in Figure 3.11. The first phase, Figure 3.11(a), starts with the formation of a staircase structure along the west edge of a target pattern, P. The staircase structure is grown to append a unique tag (number of tiles) to each row of the target pattern. The unique tag helps in the correct reassembly of replicates to produce replicated patterns. With the formation of staircase structure, tiles in the pattern are signaled to turn off their glues, and thus the rows (P1, P2, P3, P4, P5) of the pattern get dissociated from each other.

In Phase 2, as shown in Figure 3.11(b), the dissociated rows from the first phase are used as templates for the formation of non-terminal replicates (ntr1, ntr2, ntr3, ntr4, ntr5). In the end of the phase 2, the ntrs get dissociate from the templates of the pattern. The dissociated ntrs are used in phase 3, as shown in Figure 3.11(c). Each ntr serves as a template for the formation both a new ntr (ntr') and a terminal replicate (tr) concurrently. In the end of the phase 3, the trs detach from their parent ntrs, and along with other trs form a replicated copy of the target pattern (replicated P). The trs ensure that replicated copies of pattern P are produced, while ntrs ensure that copies of P are produced at an exponential rate.

Although the technique provides an innovative approach for enzyme-free self-replication of both 2-D and 3-D structures, design and implementation of such tiles with asynchronous and non-local signaling control would be a real practical challenge [95].

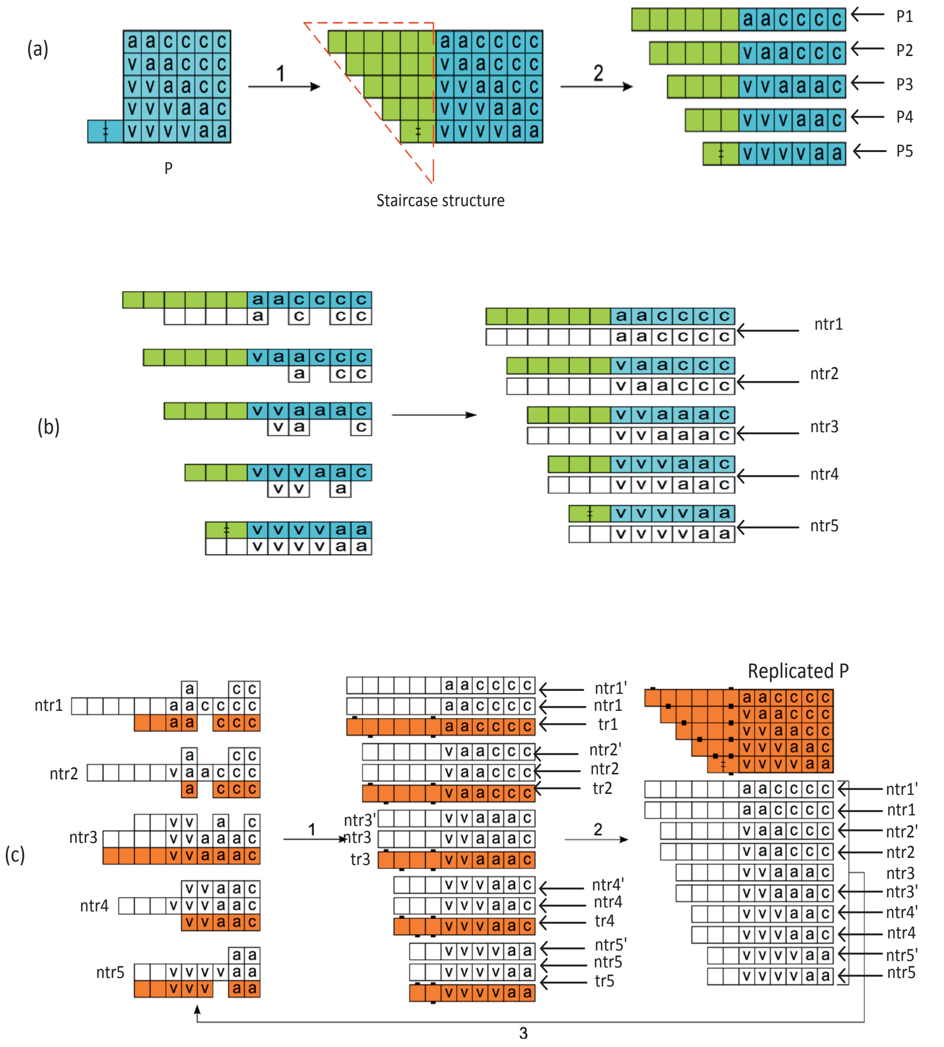


Figure 3.11: Template-directed self-replication of 2-D rectangular patterns of tiles [75]. (a) Target pattern, staircase formation (1), and autonomous dissociation of the pattern into individual rows. (b) Formation of ntrs using the dissociated rows as templates. (c) Formation of ntrs and trs concurrently using ntrs obtained from (b). In the end of the cycle, copies of trs produce replicated pattern and two sets of ntrs copies can be used to produce two new replicated patterns in the next cycle, and so on the process could produce exponential replication.

3.7 Comparison of Self-replicating Models

In the following, we present a comparison of the self-replicating systems reviewed in this chapter. Comparison, listed in Table 3.2, has been made based on the following four criteria [41].

1. **Physical Realism:** There is a varied degree of physical realism (low, medium, or high), which is incorporated into different models of self-replicating systems. A realistic model gives closer insight into the physical reality of self-replicating system, thus it helps in the engineering of a system for a desired behavior. Due to abstraction used in the CA models, they are far from realism, and thus have little relevance for physical realization and manufacturing. Models of molecular self-replicating systems, such as Squirm3, JohnnyVon and DNA tile self-assembly models incorporate physical realism. The Squirm3 model includes virtual chemistry, continuous space, mobile molecules, and a simple virtual physics. The JohnnyVon model has moderate physical plausibility, with a virtual physics that includes continuous space, Brownian motion, viscosity, momentum, and attractive and repulsive fields. The DNA tile self-assembly model, such as the kinetic Tile Assembly Model [154, 156], includes physico-chemical factors governing the assembly of tiles.

2. **Phenotype-Genotype Distinction:** A self-replicating system may produce structures that further manufacture structures of higher order. Although a self-replicating system may produce replicas with and without the presence of genotypes (seed structures), manufacturing of higher order structures using replicated copies requires the presence of phenotypes in the encoding of the structures. Von Neumann's universal constructor automaton makes a strong distinction between the phenotype (the group of cells that compose the universal constructor) and the genotype (the line of cells that compose the instruction tape, read by the universal constructor). There is no meaningful difference between genotype and phenotype in the self-replicating loops. The Squirm3 has both genotype(self-replicating strands of molecules) and phenotype (the circular barrier they build, like a cell wall). The first version of JohnnyVon, JohnnyVon 1.0 [138], has no distinction of genotype and phenotype: strands could only self-replicate. The second version of JohnnyVon, JohnnyVon 2.0 [41], however implements a clear distinction between genotype and phenotype. This distinction separates the act of self-replication from the act of building useful structures. DNA tile self-assembly have distinction between genotype and the phenotype; seed serves as genotype to produce larger complex pattern of tiles, which can be considered as phenotype of the system, as the genotype and phenotype have different encoding here.

3. **Programmability:** A programmable self-replicating system can be used for the production of a variety of structures, which may not need altering the overall implementation framework of the system. In the CA models of self-replicating systems, programmability is linked with the complexity of the self-replicator. For

example, programmability of von Neumann's universal constructor comes at the cost of complexity. Self-replicating loops are less complex than the von Neumann's self-replicator, but they sacrifice the programmability. If initial configuration of a seed loop is changed, either it ceases to self-replicate or repairs itself to come back to the initial configuration where it restarts the process.

4. **Tractability:** computationally demanding simulation models of the self-replicating systems often become intractable for larger systems, thus limiting the system that can be simulated on available computing resources. Von Neumann's universal constructor is a very complex system, which makes it intractable to simulate its behavior even today with available computing resources. Simple self-replicating loops and other self-replicating systems, such as Squirm3, JohnnyVon, and DNA tile self-assembly simulations are computationally tractable. The tractability of tile pattern self-replicators, reported in Abel et al. [6] and Keenan et al. [75], is not known due to unavailability of computer simulation model (marked by '-' Table 3.2).

Table 3.2: Comparison of self-replication models reviewed in this chapter.

Model	# 1	# 2	# 3	# 4
Universal constructor [18]	low	yes	high	low
Self-replicating loops [79]	low	no	low	high
Squirm3 [67]	medium	yes	low	high
JohnnyVon 1.0 [138]	medium	no	low	high
JohnnyVon 2.0 [41]	medium	yes	medium	high
Tile pattern self-replicator([120])	high	yes	high	high
Tile pattern self-replicator [6]	medium	yes	high	-
Tile pattern self-replicator [75]	medium	yes	high	-

where # 1, # 2, # 3, and # 4 refer to the four parameters listed above.

3.8 Summary

Over the last 65 years, artificial self-replication has been extensively studied using both CA models and physical/chemical systems. The earliest CA model of universal constructor by John von Neumann was a very complex design, and therefore, this could not be realized. This work, however, successfully put foundation to the field of artificial self-replication, produced important theoretical insights, and gradually led to the development of simple self-replicating loop designs that have been readily realizable. Although CA models of self-replicating loops have produced interesting studies of emergence and life, they are far from the physical realism. Therefore, realization of these models using known biochemical processes has not yet been possible. Progress in this direction has mainly been hindered due to lack of molecular structures and biochemical processes that can map the abstract processes of CA model.

For example, a self-replicating loop in the CA model might have a vague correspondence to a circular structure of oligonucleotide and a protein that transcribes it for the creation of replica. Nonetheless, the insight gained from the simple self-replicating structures in CA models is promising for the realization of self-replicating molecular structures embodying information processing.

In parallel to the developments of CA models of self-replication, kinematic self-replication models of physical/chemical components with simple set of requirements have also been reported [102, 53, 58]. These models incorporate physical/chemical realism, making them useful for the design of real self-replication systems. Physical/chemical self-replication models covered in this review include: Pensrose's model [102], Griffith's model [58], and molecular self-replicators. Physical self-replicators, such as Penrose's model and Griffith's model use macroscopic physical components to produce replicas of seed structures. At the macroscopic level, there are not many options to carve out distinctions in different physical components. For example, the plywood blocks used in the Penrose's model have symmetry, thus limiting the complexity of patterns that can be self-replicated. Self-replicating systems of molecules have produced both realistic simulation models and designs of molecular structures. Realistic simulation models of self-replicating molecular systems, such as Squirm3 [67] and JohnnyVon [138, 41] have given interesting insights to both manufacturing at the nanoscale and principles governing self-organization and evolution in a chemical system.

Research efforts are underway to see whether non-enzymatic catalytic systems [101] recently being developed by organic chemists, emerging approaches for the synthesis of multifunctional molecules [98, 26] and the innovative manufacturing methods currently being developed in the field of nanotechnology could be used to realize such molecular self-replicating systems. To this end, DNA tile self-assembly is one such emerging technique that enables the implementation of programmable tile patterns at the nanoscale. DNA tile self-assembly process has a correspondence to the CA, thus making it suitable for the realization of self-replicating information patterns of tiles. This approach may open up novel routes to low-cost manufacturing at the nanoscale, which is one of the central goals of nanotechnology. Also, it may help in gaining further insights into self-replication patterns and information processing in the realistic medium using molecules.

4

Enveloped Tiles and Reliable Self-assembly of Tile Patterns

Self-assembly provides a powerful way to design large complex structures using a finite number of simple building blocks, where structures emerge spontaneously via local interactions among the building blocks. Nanoscale structural building blocks of DNA, DNA Tiles, have been used to design programmable self-assembly systems. Using sets of DNA tiles, large complex tile patterns can be constructed. Although tile self-assembly could be optimised such that a favorable tile binding is strongly preferred over an unfavorable binding, such a strong discrimination is achievable under a very limited range of physical parameters (assembly temperature, tile concentration, salt concentration, and PH) that are difficult to maintain in laboratory experiments. Therefore, assembly errors occur so often that the feasibility of the technique remains limited.

In this chapter, we propose the use of an ET consisting of a DX tile as BT that carries a PT to suppress erroneous tile assembly. The design of the ET promotes the dissociation of the PT from the BT through a self-triggered activation process, which keeps the outputs of the BT protected until both BT inputs have bonded correctly to the assembly. The ET structure design, DNA sequence design, thermodynamics stability, a complete set of ETs designed for Sierpinski pattern assembly, and a simple analysis of assembly error-prevention and assembly time are discussed.

4.1 Related Work

4.1.1 Tile Assembly Errors

Nature accomplishes the spontaneous formation of complex molecular systems with the help of a variety of biomolecules. These biomolecules serve as a set of building blocks, which self-assemble into larger macromolecules through physicochemical interactions. The assembly of such molecular systems is compelling for the artificial design of nanoscale systems for at least two reasons. First, the underlying mechanism of such molecular systems is simple, yet capable of achieving complexity unimaginable for a top-down design approach. Second, such a construction approach, where a system builds itself, might be the only route to designing the nanoscale systems in the future, where it is not possible to have control over individual components. Although self-assembly techniques have been in use for chemical synthesis for a long time, the designed molecular systems were simple and non-programmable.

Self-assembly of DNA tiles [154] — the emergence of assemblies through physico-chemical interactions between basic building blocks, termed *DNA tiles (in short, tiles)*, provides a potential technology for nanostructure construction and molecular computation. Tiles may be designed with combinatorial DNA sticky ends that can bind according to the rule of DNA complementarity. Algorithmic self-assembly of tiles has shown potential to implement universal Turing machine [155] and thus, theoretically a wide variety of complex structures can be assembled via a finite set of short sticky-end sequences (5 to 10 bases long) [157, 154, 113, 132, 70, 112]. In practice, however, DNA-tile based algorithmic self-assembled structures are prone to assembly errors [156, 112, 123, 49, 23, 15].

Erroneously assembled tile structures are formed basically due to self-assembly errors contributed by mismatched tile bindings [156, 48, 23, 124]. In cooperative self-assembly of tiles, a tile binds stably at a lattice site if it attaches by two of its sticky ends termed 'inputs'. A tile attaching by single input becomes unfavorable, and detaches quickly from the engaged site. An *insufficient attachment* causes assembly error if an unfavorable tile gets trapped at the engaged site by a favorably attaching adjacent tile.

There are three basic types of assembly errors (Figure 4.1): growth errors [156], facet nucleation errors [24], and spontaneous nucleation errors [122]. The first two types of errors both result from an insufficient attachment. Growth errors occur when a tile gets trapped at a vacant site of a growing assembly even when one of its sticky ends does not match correctly. Facet nucleation errors appear when an insufficiently attached tile at a growing surface fails to detach quickly and eventually gets fixed, creating a new nucleation site that may introduce a wrong pattern in the growing lattice. If tile self-assembly process nucleates from a tile other than the designated seed, it would result in an erroneously assembled lattice, as shown by a sample self-assembly pathway in Figure 4.1(b).

As long as the self-assembly is conducted near thermodynamic equilibrium ($G_{mc} \approx 2G_{se}$), the possibility of spurious nucleation events is minimum. However, spuriously nucleated assemblies become dominant when self-assembly medium moves away from the thermodynamic equilibrium towards strong supersaturation ($G_{mc} << 2G_{se}$) [156, 126]. In a tile self-assembly experiment using a homogeneous seed (one of the tiles serves as the seed), it becomes a challenge to conduct the experiment with minimum assembly errors (that requires a slightly supersaturated medium), while maintaining favourable conditions for the initiation of assembly from a single tile seed (that requires strong supersaturation) [124, 126]. Using heterogeneous seed structures,

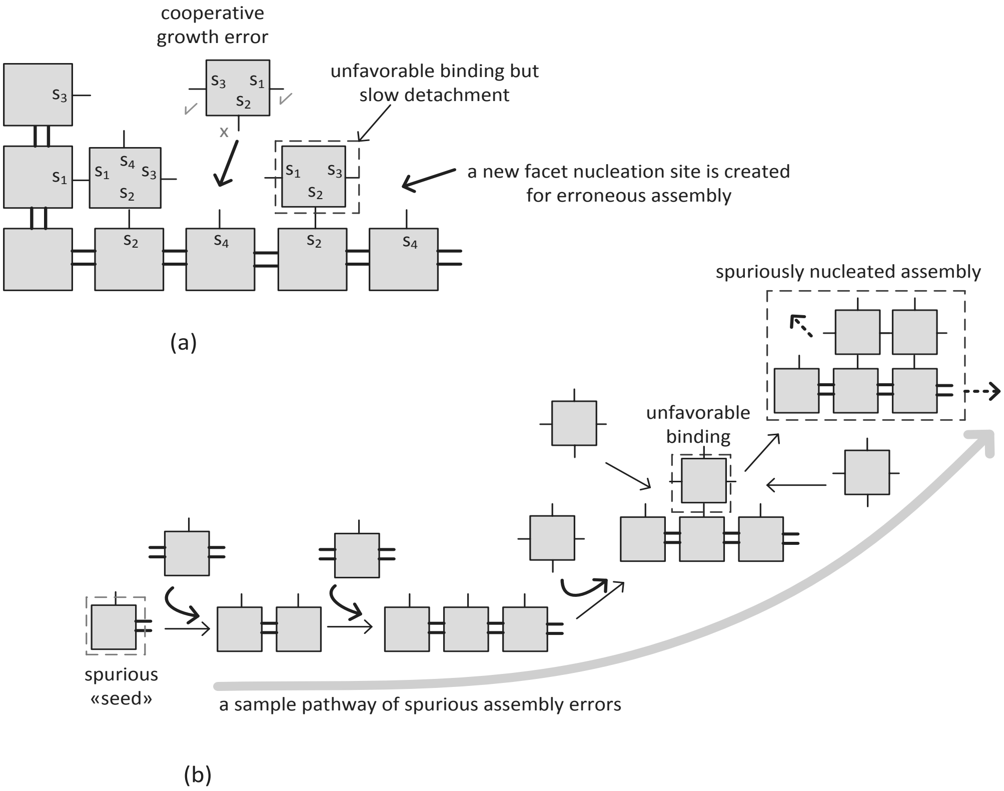

cooperative growth error
unfavorable binding but slow detachment
a new facet nucleation site is created for erroneous assembly
s3
s1
s4
s2
x
(a)
spuriously nucleated assembly
unfavorable binding
spurious «seed»
a sample pathway of spurious assembly errors
(b)

Figure 4.1: Assembly errors in cooperative self-assembly. (a) During cooperative self-assembly, an insufficient (tile binds by single sticky end) attachment of a tile may introduce two types of assembly errors: 1) growth errors due to tile trapping; 2) creation of a new nucleation site that may propagate an errorneous pattern. A few sticky end levels (s_i) are marked on the edges of tiles to show matches and mismatches between sticky ends of tiles (b) Following a nucleation of assembly from a spurious seed tile, errorneous assemblies are formed .

which are pre-assembled and introduced into the self-assembly medium, such difficulty is avoided [15, 89].

Lets use our previous understanding from the kinetic model (Section 2.4.2) to analyse the issue of assembly errors. For $G_{mc} \approx 2G_{se} - \epsilon$, a tile attaching by *b* bonds will fall off at a relative rate of $\frac{r_{r,b}}{r_f} = e^{(2-b)G_{se}-\epsilon}$. For a small positive value of ϵ and $b < 1$ i.e., tiles binding by single bond or no bond, would fall off very quickly after they approach a lattice site and therefore, can be ignored. But the tiles attaching by two or more bonds will favourably attach and thus, there would be a net forward growth of the lattice, but due to co-operative binding some mismatched tiles could get fixed in the growing lattice. The interesting case of cooperative tile assembly arises when $b = 2$ and thus, tiles binding by two bonds are slightly more favourable ($\frac{r_{r,2}}{r_f} = e^{-\epsilon}$) to remain attached than to fall off.

Imagine a situation where G_{mc} is small i.e., the forward rate r_f is large and therefore, a tile attaching with single bond or no bond would not be too quick in falling off. Therefore, this engaged tile would get fixed erroneously if there is an adjacent tile which binds with it in the meantime. These errors are the main source of poor reliability in algorithmic self-assembly of tiles. This signifies that assembly errors are linked with the rate of assembly growth. Winfree studied this link using a *kinetic trapping model* [154], which establishes a relationship between a desired error rate (ε) and growth rate (r_g) at which assembly proceeds most rapidly. This relationship between error rate and growth is given by $r_g \approx 0.75 \times 10^6 \varepsilon^2$, where $r_g \approx \frac{r_f - r_{r,2}}{2}$ and $\varepsilon \approx 2e^{-(G_{mc}-G_{se})}$. Thus, any effort to reduce the error rate (ε) by tuning physical parameters (G_{mc} and G_{se}) would result in a quadratic reduction of the growth rate.

4.1.2 Tile Assembly Error Prevention Mechanisms

Self-assembly of DNA tiles is thus inherently an unreliable process, where illegitimate binding between tiles often results in erroneously assembled structures. The algorithmic self-assembly of tiles uses cooperative binding between tiles to guide the formation of structures. The basic idea behind cooperative binding is that a tile binding with two or more bonds is more favorable than a tile binding with a single bond. For an ideal cooperative binding in the algorithmic tile self-assembly, it is assumed that each tile binds in a directional fashion i.e., it involves only two sticky ends termed inputs, while the other two sticky ends, termed outputs, remain free until the input binding is complete. However, this theoretical assumption is not true in practice. The output ends of assembling tile are open for binding with adjacent tiles, which may provide illegitimate binding for fixing the tile. The challenge of preventing such assembly errors is two fold: 1) tiles binding with two input bonds should be strongly favored against tiles binding with single bond, and 2) the output ends of tiles should not be exposed to binding until the binding at the input ends is complete.

Several mechanisms have been proposed to prevent and/or correct tile assembly errors [156, 22, 49, 83, 60]. These include: error-preventing redundant tile sets [156, 22], protection and triggered deprotection based error-preventing tiles [49, 83], and thermocyling based error-correction technique [60]. The assembly process of redundant and thermocycling based error-correction approaches is the same passive process near the thermodynamic equilibrium as proposed by Winfree [157]. In the activation-based approaches, tiles are initially deactivated due to their covered sticky ends. The deactivated tile activates as it assembles with an already unprotected seed structure or at a growing front of tile lattice, and releases its protection following the correct binding.

Redundant Tile Sets

In the algorithmic tile self-assembly, an assembly error is introduced when a mismatched tile fails to detach quickly and gets trapped by subsequent tiles joining the lattice [154]. Therefore, if the tile could not assemble without making a second mismatched binding, it would give more time to the engaged tile to detach before it becomes part of the growing tile lattice. In the redundant tile sets [156, 22, 127], each original tile is replaced with a redundant block ($k \times k$) of tiles so as to provide more time for a mismatched tile to get detached before it gets assembled with the lattice.

A *proofreading tile set* consists of blocks of $k \times k$ unique tiles derived from the original tiles. Each boundary tile within a block has its external sticky ends equal to the corresponding side of the original tile, while the internal edges of the tile have unique but complementary sticky ends. Therefore, if there is a mismatched tile binding during the growth, the corresponding block will not self-assemble without $k - 1$ more mismatched bindings. This would result in a block that is either correct or has k mismatches. In thermodynamics, this would mean that a correct assembly of $k \times k$ block is discriminated (proofread) by $e^{-kG_{se}}$ against its erroneous assembly. Thus, for a 2×2 tile set, there would be a quadratic reduction in the the errors. Although there is a substantial reduction in the error rate, it comes at the cost of k-fold increase in the overall size of the assembled structure and the time required for its assembly.

Another proofreading tile set termed *snaked proofreading* was introduced by Chen and Goel [22]. The *snaked proofreading* tile set is also designed by replacing each original tile with a $k \times k$ block of tiles. The snaked proofreading tiles also consist of null bonds on their edges, which enable multiple mismatched bindings to occur — especially on the growing facets of the lattices — before an erroneous tile block forms. This tile set shows substantial reduction of the facet nucleation errors.

Thermocycling-based Error Correction

Gu et al. [60] experimentally demonstrated a simple thermocycling-based error-correction protocol for a dynamically controlled and programmable self-assembly of DNA tiles. The basic idea is to pre-assemble DNA tiles (Triple Crossover tiles termed *Capture Tiles (CT)* in this case) with pairs of PX and JX_2 structures (PX and JX_2 structures are two configurations of the $PX - JX_2$ device discussed in [162]). These pre-assembled *CTs* are used to design dynamically programmable system of DNA tiles. Although the approach enables dynamic control and reconfigurable designs of self-assembled DNA tile patterns, it was observed that the possibility of assembly errors remains similar to the case of Winfree's [154] algorithmic self-assembly of DNA tiles. This is because of mismatched (Even if only two out of four sticky ends of a *CT* match correctly, it may get assembled) assembly of *CT* with PX and JX_2 pairs.

In order to prevent such assembly errors of *CTs*, an innovative thermocycling-based protocol at the temperature near thermodynamic equilibrium of these tiles has been demonstrated. The basic idea behind the thermocycling protocol is to prevent erroneous assembly of *CTs* with PX and JX_2 pairs. It is observed that a half-correct *CT* , having only two sticky ends matching with the PX and JX_2 pair, is thermodynamically stable below $35°\mathrm{C}$, but it gets dissociated in between $35°\mathrm{C} - 37°\mathrm{C}$. However, if a capture tile is fully-correct (matching by all four sticky ends), it dissociates only above $40°\mathrm{C}$. Therefore, there is a non-permissible range of temperature between $37°\mathrm{C} - 40°\mathrm{C}$, which could be used to prevent erroneous assembly of capture tiles. Thus, the protocol loads capture tiles one-by-one (independent of the order in which *CTs* are added) and in each cycle it first raises the temperature to the non-permissible temperature range and then cools down to $4°\mathrm{C}$. This leads to an all correctly assembled *CTs* that can be used to assemble DNA tile patterns.

Activation-based Error-correction Tiles

Protected Tiles: Fujibayashi et al. [49] introduced two mechanisms to kinetically-controlled error prevention, named Protected Tile Mechanism (PTM), and Layered Tile Mechanism (LTM). The PTM, illustrated in Figure 4.2 is based on a theoretical design of tile having a single strand of DNA that partially protects its input ends. A kinetic barrier formed by protecting DNA safeguards a tile against any illegitimate binding through its input ends. But, as the output ends are still unprotected, it is amenable to kinetic trapping described in previous section. The LTM, illustrated in Figure 4.3, uses a pair of stacked tiles such that the upper tile protects both inputs and outputs. Similar to the PTM, it also forms a kinetic barrier to suppress spurious binding at input ends of the tile, while the output ends are partially protected (3 bases out of 14 bases long sticky end). Therefore, the PTM tile does protect outputs against kinetic trapping, but this protection is just partial, as the output ends are not fully protected.

Activatable Tiles: To make an improvement to the PTM and LTM, Majumdar et al. [83] introduced Activatable Tiles. These are initially fully protected but then change conformation during assembly by dissolving the protection DNA molecule with the help of supplied enzyme. The deprotection mechanism ensures that tiles are self-assembled in directional fashion i.e., correct binding at inputs triggers activation of output ends. The Activatable tile approach provides an innovative theoretical framework, but the use of enzyme for deprotection purposes may impose some practical limitations. Also, the the activatable tiles assemble irreversibly: once assembled, a tile can not detach by itself.

The remainder of the chapter presents the design of error-correcting protected tiles (ET). It starts with an abstract idea of ET and its intended error-correction mechanism in Section 4.2. Section 4.3 discusses the designs of topology and DNA molecular structure of ET. In Section 4.4, we discuss the DNA sequence design of the ET. Section 4.5 presents the thermodynamic stability analysis, feasibility and a wet-lab protocol to prepare ETs in the laboratory. In Section 4.6 we describe the design an ET set for the algorithmic self-assembly of Sierpinski pattern [112]. Section 4.8 discusses the prospects of assembly error-correction using ET, gives a rough estimation of assembly

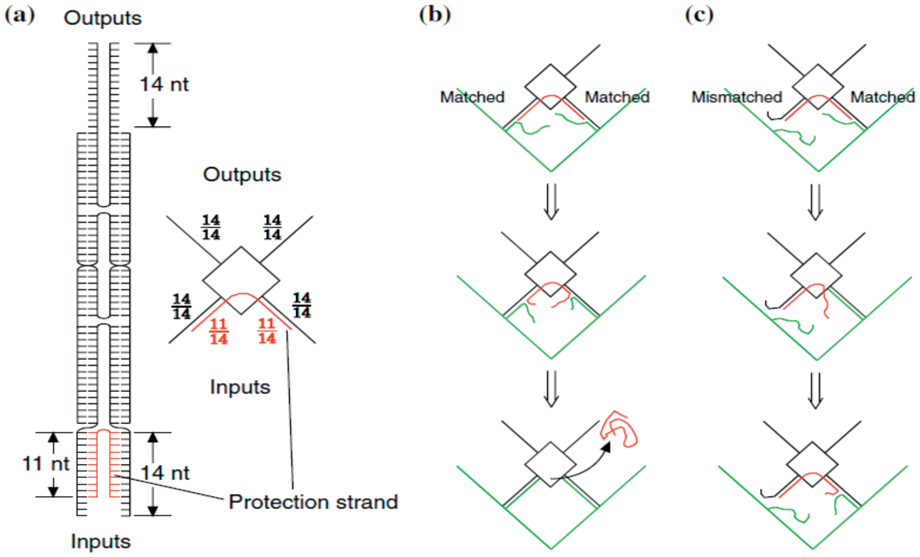


Figure 4.2: Protected tile assembly mechanism [49]. (a) Schematic representation of PT having protection strand associated with its input ends (*PT*). (b) Assembly mechanism of a correctly matching PT, and (c) Errorneous matching PT.

time in the ET assembly framework, and comments about strengths and limitations of the ET design in comparison with the other state-of-the-art error-correction techniques.

4.2 An Abstract Idea of Self-assembly Using Enveloped Tiles

An ET, illustrated using a simplistic schematic diagram in Figure 4.4a, consists of two elements: BT and PT.

The *BT* is an original tile published in Winfree at el. [157], while the molecular structure of *PT* is designed by us so as to protect the inputs and outputs of the *BT*. Sticky ends of both *BT* and *PT* are shown in protruding lines having corresponding lengths in nucleotides (*nt*) marked over them. The basic idea behind the self-assembly of *ET* is illustrated in Figure 4.4b,c.

Self-assembly starts with a set of ET – only one *ET* is shown here for simplicity — so that the tiles do not assemble until exposed to an initiating 'seed' structure (a triplet of tiles in the figure). After the entry of seed structure (which can be either a single tile or a preformed structure of tiles, as shown here), the *ET* then approaches the available

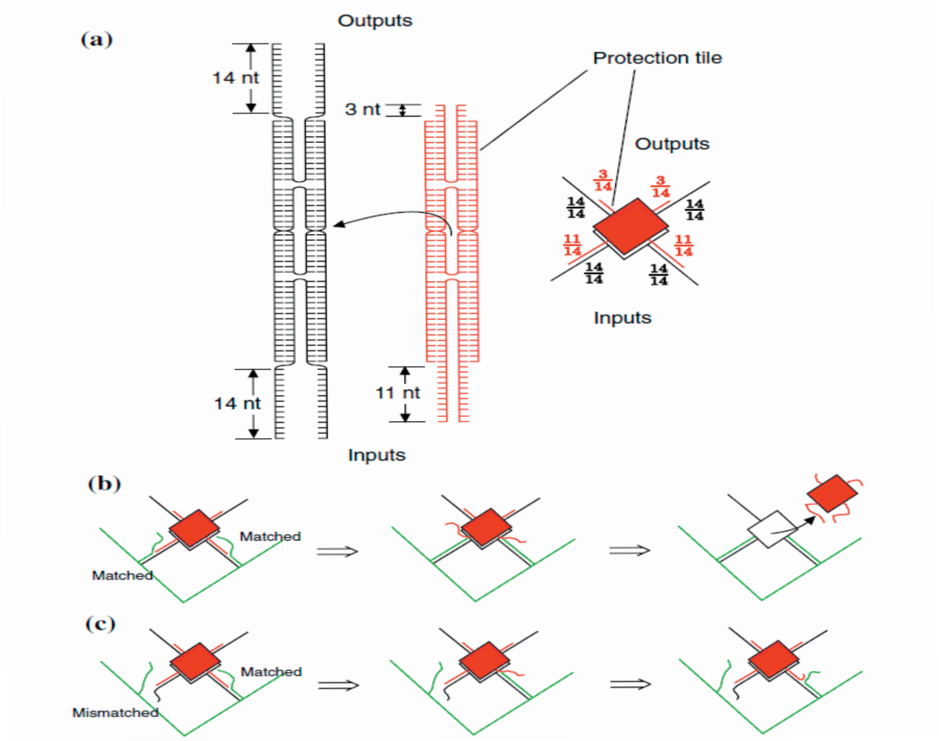


Figure 4.3: Layered tile assembly mechanism [49]. (a) Schematic representation of LT having stacked tile over the BT. (b) Assembly mechanism of a correctly matching LT, and (c) Errorneous matching LT.

site for self-assembly. If both of its input ends match correctly with the seed outputs as shown in Figure 4.4(b), the *PT* gets dissociated at the end of the assembly process, and it fixes the *BT* at the site. However, in case of a mismatch on either of the inputs, the *ET* remains only unstably connected and is ultimately rejected from the site as shown in Figure 4.4c.

4.3 Topology and DNA Molecular Structure of Enveloped Tile

Double Crossover and Anti-parallel with Odd spacing (DAO) [47] DNA molecular structures are accepted building blocks for DNA tile self-assembly [157, 112]. The *BT* of the ET is designed using the DAO. Two interconnected DAO molecules are shown in Figure 4.5a. DAO molecular structure has a 2D planar topology, assembles from

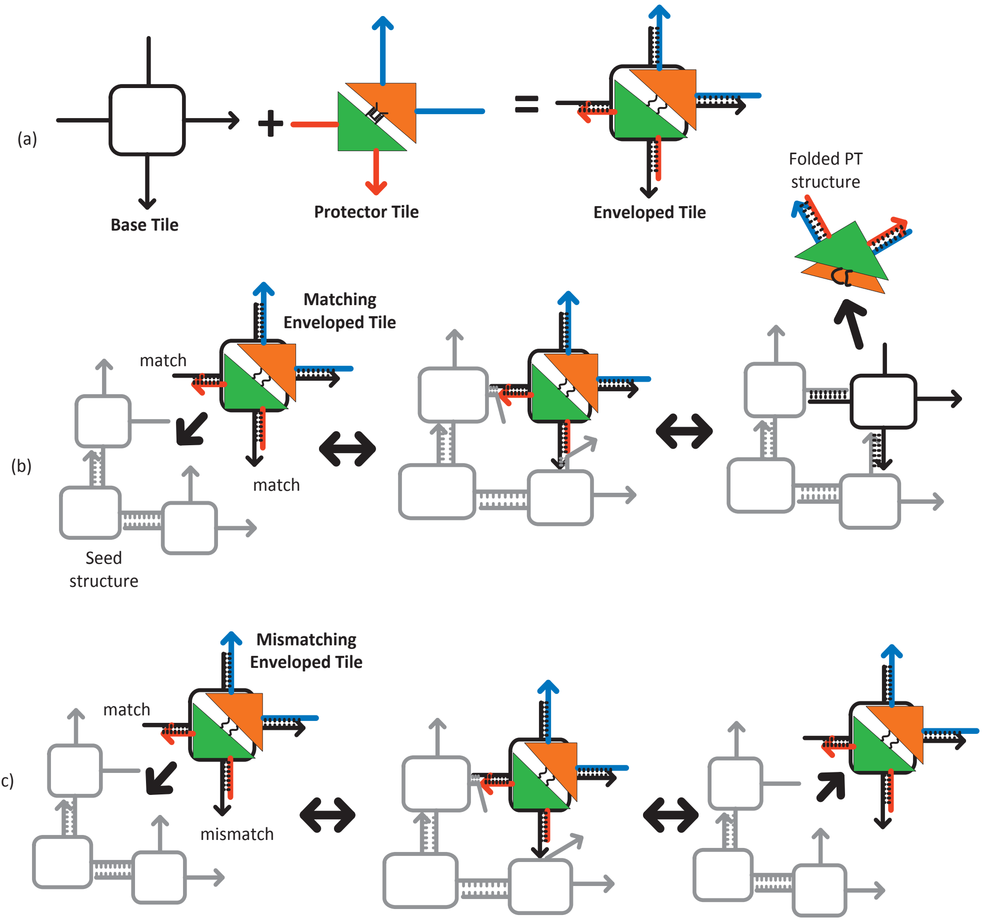


Figure 4.4: Schematic representation of ET and self-assembly process. (a) Schematics of BT, PT, and ET. Self-assembly of an ET having both of its input sticky-ends matched with the available site (b), and with one mismatched end (c).

four single strands of DNA. The DAO molecular structure forms two crossover points, where DNA strands become antiparallel forming stacking bonds and thus holding the *BT* structure into a plane. In a DAO molecule, these crossover points are separated by an odd number of DNA helix half-turns (one full turn of DNA helix is ≈ 10.5 bases long). Furthermore, intermolecular spacing between two connected tiles — as shown in Figure 4.5a — is kept equal to an odd/even number of half-turns to ensure that the assembled lattice lies in a 2-dimensional plane.

BT Structure: We adopted the design of BT from the DAO tile published by Winfree et al. [157, 112]. In the original tile, we made three changes to make the BT shown in Figure 4.5a: 1) sticky ends are increased from 5 nt to 10 nt, 2) intramolecular spacing is increased from 16 nt (3 half-turns) to 26 nt (5 half-turns), and 3) intermolecular

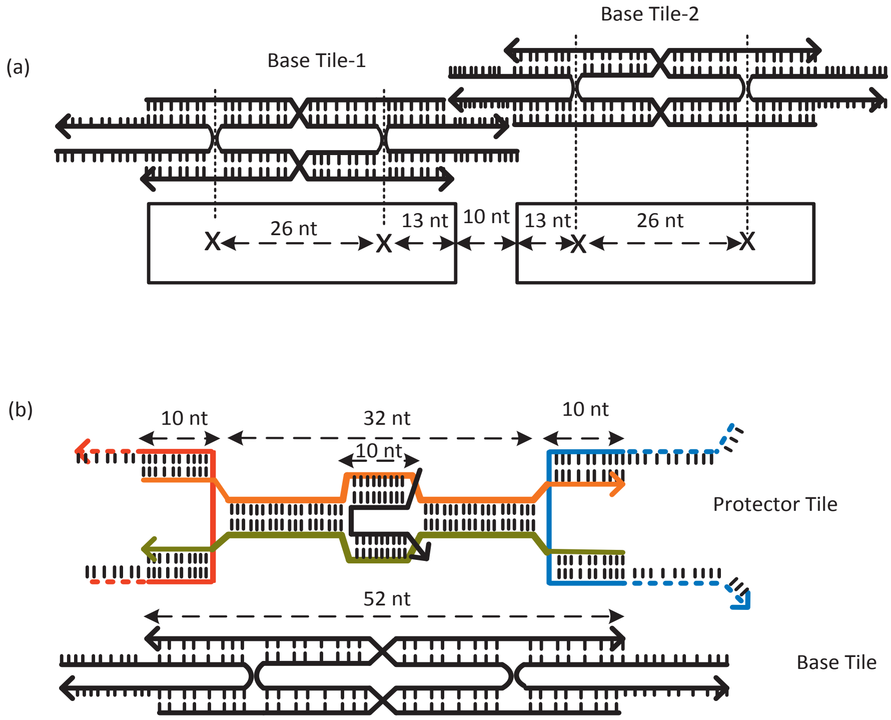


Figure 4.5: BT and PT Structures. (a) Intramolecular and intermolecular separations between crossover points (X) of BT. (b) PT structure superimposed upon a BT structure.

separation is increased from 21 nt (4 half-turns) to 36 nt (7 half-turns). These lengths of different sections were chosen to meet two criteria: 1) spacing of an odd number of half-turns between crossover points, and 2) the melting temperature of a BT should be larger than the melting temperature of the tile lattice. This difference in melting temperatures will ensure that tiles themselves first form and thus, annealing temperature can be maintained to keep the individual tiles formed, but not the aggregates of tiles. In algorithmic tile self-assembly, input and output sticky ends of a tile could be mutually complementary. Hence, temperature control is required to prevent the spurious assembly of single tile monomers.

PT Structure: The PT is the second component of an *ET* structure. The *PT* provides a unique tile design, assembling with the *BT* and thus protecting the *BT's* inputs and outputs whilst leaving protruding DNA toeholds for self-triggered activation of *BT*. Therefore, the *PT* structure should be able to superimpose upon the structure of *BT* as shown in Figure 4.5b. The length of *PT* core is designed to be equal to the length of *BT* core (52 nt long). We kept the central clamp as 10 nt long while keeping adjacent sections on each side 11 nt long. Although our choice of these lengths is based on a simple projection of the PT structure over the BT, these lengths in the central section of the *PT* can be tweaked for its superimposition upon a *BT* structure.

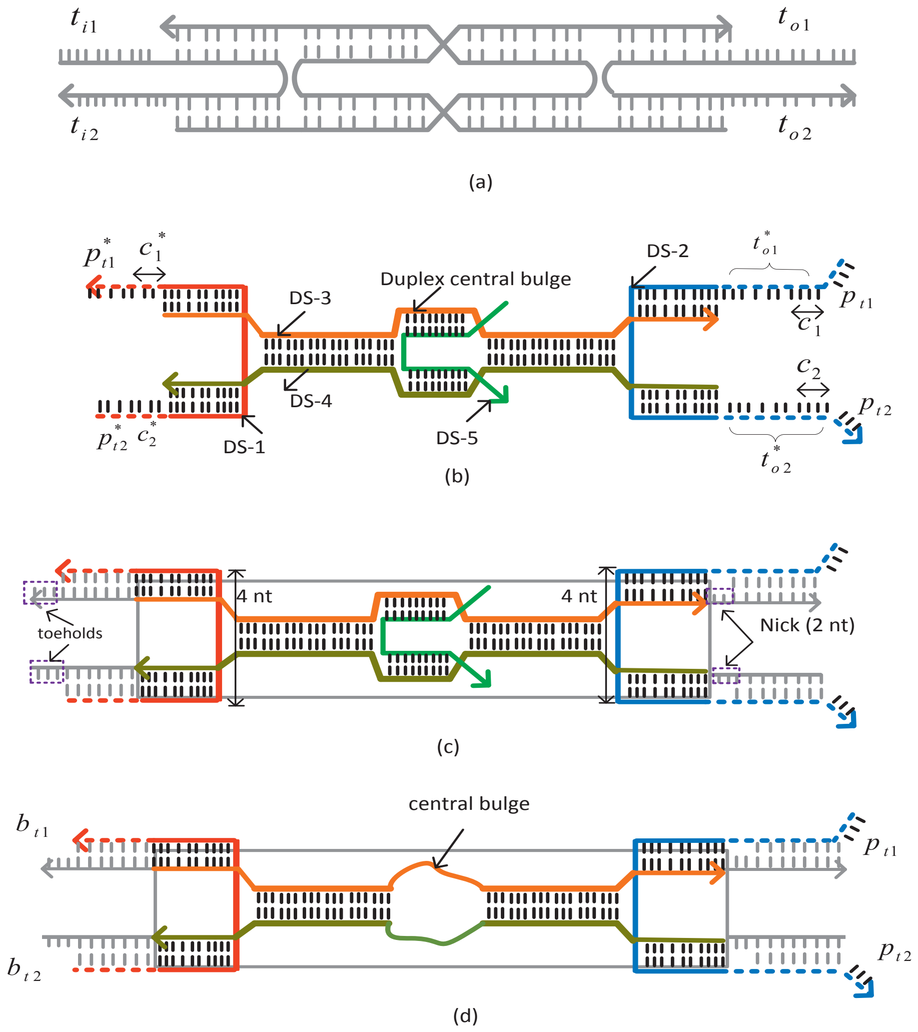


Figure 4.6: ET structure. (a) BT structure. (b) PT structure. (c) BT-PT pair with duplex central bulge.(d) ET *(ET)*.

The centerpiece of *ET* structure is the *PT* — illustrated in Figure 4.6b, involving five single stranded DNA molecules (*DS-1*, *DS-2*, *DS-3*, *DS-4*, and *DS-5*). *DS-1* and *DS-2* serve as protection for the *BT's* input and output sticky ends, respectively, and are clamped together by *DS-3* and *DS-4*. The double stranded DNA sections on both sides of the *duplex central bulge* act as *distal clamps*, linking the ET engagement at the input side to the intramolecular strand displacement events at the output side.

DS-5 forms a double-duplex central bulge together with DS-3 and DS-4, and leaves two protruding toeholds. The duplex central bulge serves two purposes. First, stiffness of duplex DNA holds the two mutually complementary ends of the *PT* structure separated. Second, after formation of the *BT-PT* pair, shown in Figure 4.6c, the *DS-5* is dissociated via a supplied DNA strand that triggers strand displacement through available toeholds of duplex central bulge. This results in a *BT-PT* pair (*ET*) having a non-duplex central bulge as shown in Figure 4.6d. This helps to fold *PT* after it is released from its input ends during the assembly of *ET*.

Each ds-DNA section (depicted by vertical dashed lines) is 10 base-pairs long and formed by two anti-parallel ss-DNAs. One side of each ss-DNA is marked by an arrowhead representing its 3' end; the other side is the 5' end. The 7 bases long sticky ends of *DS-1* (e^*_{t1}, c^*_1 and e^*_{t2}, c^*_2) are complementary to the 7 bases sections (e_{t1}, c_1 and e_{t2}, c_2), respectively. The input sticky ends of the *PT* ((e^*_{t1}, c^*_1 and e^*_{t2}, c^*_2) of *DS-1*) are also complementary to the 7 bases of the input sticky ends of the *BT* — see Figure 4.6c. Similarly, 7 bases of the output sticky ends of the *PT* (t^*_{o1} and t^*_2 of DS-2) are complementary to the output sticky ends of the *BT*.

The final structure of the *ET* is shown in Figure 4.6d. It consists of a stacked pair of *BT* and *PT*, and two set of binding toeholds: 3 *nt* toeholds (b_{t1}, b_{t2}) on the input side, and 3 *nt* toeholds (p_{t1}, p_{t2}) on the output side.

The basic idea behind the error-prevention of Enveloped Tiles is kinetic discrimination: the assembly of a correctly matching tile (both input sticky ends match correctly) is kinetically favored over the assembly of a partially-matching tile (only one out of two sticky ends matches) in the algorithmic tile self-assembly setup. To achieve this, Enveloped Tile design intends (which has been proved in the Chapter 6 of the thesis) to enforce a synchronization between the two strand displacement processes, one on each input side of the Enveloped Tile, as it assembles. Imagine a case when only one side of the ET matches: the corresponding PT arm will undergo strand displacement, which may eventually release the arm. However, the other input arm of the PT is still attached with the BT, and thus it is likely that either the strand displacement under progress in the first arm may halt or it may reverse the strand displacement (a few bases from the recently bound strand of the incoming tile) reaction due to raised increased local concentration of the dissociating strand of PT.

To raise the local concentration during the event of the partially-matching displacement, we suggest that two distal clamps (shown in Figure 4.7) can be added in the topology of PT, shown in Figure 4.6b. These distal clamps can play an important role in the case where only one side of the *PT* gets dissociated i.e. an erroneous tile, and by using this process no erroneous tile assembly happens. The clamps hold the dissociated input ends close to their binding sites (emulating a locally elevated concentration of these binding sites). This in turn prevents the inputs of the *PT* from engaging with its outputs. The clamps, therefore, reinforce the reverse branch migration so that a partially displaced *PT* can return to its original structure when released from the assembly.

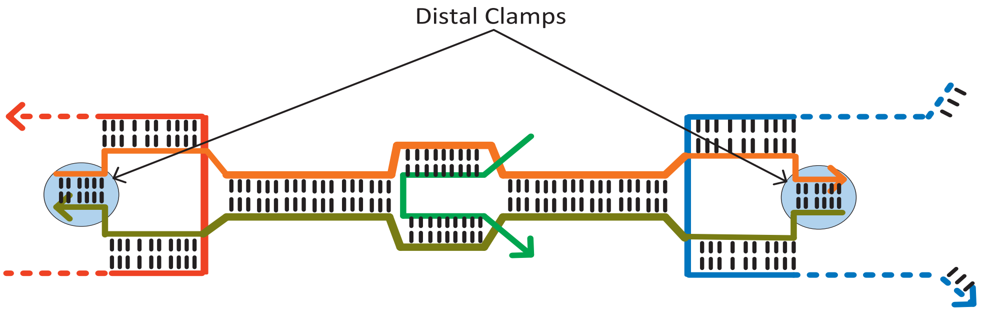


Figure 4.7: PT structure with distal clamps.

4.4 DNA Sequence Design of Enveloped Tile

The next step of DNA tile design is translation from the tile topology into the relations between strands, as shown in Figure 4.8. The basic concept is to identify the double-stranded (duplex) regions in the target topology, and apply the same color to the sections in the DNA strands that are going to form the double-stranded regions in the target structure.

In the case of ET topology, we break it down into two levels: in the first level, double-stranded regions are intramolecular (Figure 4.8a,b); in the second level, double-stranded regions are intermolecular (Figure 4.8c). If we open the DNA strands of the two DNA structures (BT and PT) separately, the topologies get translated into the color-coded DNA strands. The same color-coded sections between the strands are required to be assigned complementary DNA sequences, while the sections with dissimilar colors are required to be non-complementary. Thus, the DNA sequence design now converts into an optimization problem.

DNA sequences of structures of simple topologies can be designed using heuristic search based optimization. Starting with random sequences, search algorithms attempt to minimize the undesired complementary subsequences within a single DNA sequence and between different DNA sequences [131]. For complicated DNA structures, such as DNA tiles that include multiple DNA strands and pseudo-knotted topologies [110], there is no reliable way to design the DNA sequences. This is because of unknown thermodynamics of the pseudo-knotted.

The accepted technique of DNA sequence design for the DNA tile topology so far is trial and error method using heuristic program developed by Winfree [3] that uses thermodynamic penalty score to optimize the sequences. We used Winfree's heuristic program to design the DNA sequences (BS-1, BS-2, BS-3, BS-4) of the BT (Figure 4.8a). Sequences of the PT are designed using NUPACK [169]. Scripts to design the BT sequences and PT sequences are enclosed below:

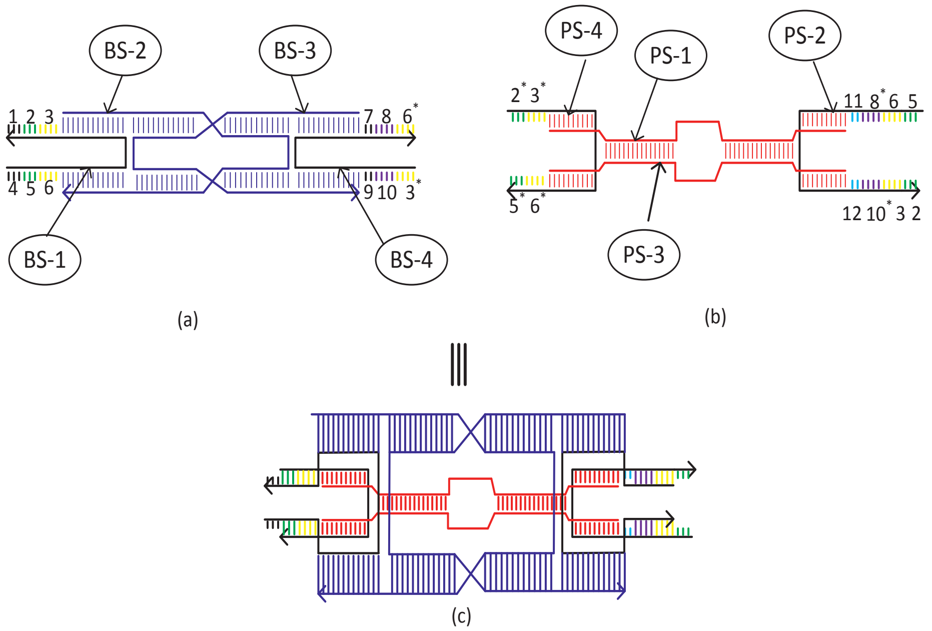


Figure 4.8: DNA sequences design for the target topology od ET. (a) Double-stranded regions in the BT topology. (b) Double-stranded regions in the PT topology.(c) The target topology of the ET and desired double-stranded regions.

DNA Sequences Design for The PT: Temperature = 25.0 °C
Material = dna
Parameters= SantaLucia1998
Dangles = 'some'
Na^+ = 0.05 M, Mg^{++} = 0.0115 M
————- 'Dot-parentheses+' Notation of the PT topology
((((((((((((((((((((((((((..........
((((((((((((((((((((((((((+.............))))))))))....
((((((((((.............+))))))))))))))))))))))))))
..........))))))))))))))))((((((((((+
.......))))))))))....)))))))))).......
————

DNA Sequences Design for the BT: The basic concept in using heuristic optimization for DNA sequence design of the BT is to initialize the lengths of DNA strands, fixed sequences from the PT (the complementary domains from level 2 topology (ET)), and the complementary regions. There are six duplex regions (see, the 'BaseTileDR' in the

Table 4.1: DNA Sequences of PT Tile.

PT DNA strands	DNA sequences
PS-1	5’-GTTGCCCTCGCGGCCAGACCTCTCGGTTCCCATACCCCGCCAATA ATGTCCGGCGAGGCGGC-3’
PS-2	5’-TGTCAGGCTACCTGCCGCCTCGCTTTTCGGTCTCGGCTTGGTGT CAGGAT-3’
PS-3	5’-GCCGAGACCGCGGACATTATTGGCGGTCCCAGTTTTCCGAGAGG TCTGGCCGGCTGCCTCGC-3’
PS-4	5’-ATCCTGAGCGAGGCAGCGTTACGAGGGCAACCCTGACA-3’

script) in the BT.

The script for the BT sequences design (Run using the SpuriousC [3])

$>> BaseTile = ['NNNTGTCAGGNNNNNNNNNNNCACCNNNNNN$
$NNNNNTCAGGATNNN'...'NNNNNNNNNNNGGACNNNNNN$
$NNNNNNNNNNNNNNNNCACCNNNNNNNNNN$
$NNNNNNNNNNNNCCTGNNNNNNNNNNN'...$
$'NNNNNNNNNNNCCTGNNNNNNNNNNNN$
$NNNNNNNNNNGTGGNNNNNNNN$
$NNNNNNNNNNNNNNGGACNNNNNNNNNNN'...$
$'CTGACACCNNNNNNNNNNNNNGTGGNNN$
$NNNNNNNNNNGTAGCCTG'];$

$>> BaseTileDR = [1\ 11\ 2\ 78\ 13; 3\ 40\ 2\ 65\ 26; 3\ 66\ 4\ 23\ 13; ...2\ 1\ 1\ 36\ 13; 2\ 14\ 3\ 39\ 26; 4\ 24\ 3\ 13\ 13];$

$>> [St, wc, eq] = constraints(DAOseq, [BaseTileHelices; BaseTileSticky], []);$

$>> save_spuriousC_files(S, St, wc, eq,' BaseTile');$

$>> unix('spuriousCtemplate = BaseTile.Stwc = BaseTile.wceq = BaseTile.eqtmax = 180score = spuriousW_verboten = 1')$

Table 4.2: DNA sequences of BT.

BT DNA strands	DNA sequences
BS-1	5’-GACTGTCAGGAGTGCGAGCTTCACCTTACGATGGTCAGGATTCC-3’
BS-2	5’-CCATCTCGTAAGGACTAGCAAGACAGAGTAAACAACGCA CCATACCGTCGATTGATTCTCATGCCTGAAGCTCGCACT-3’
BS-3	5’-TTAGAGCGGAACCTGCGTTGTTTACTCTGTCTTGCTAGTGGCATGA GAATCAATCGACGGTATGGACGAAGTGTTATG-3’
BS-4	5’-CTGACACCAACATAACACTTCGTGGTTCCGCTCTAAAGGTAGCCTG-3’

4.5 Thermodynamics Analysis of Enveloped Tile Structure

In this section we discuss the thermodynamics of the PT and BT structures using the DNA sequences. Calculating Ensemble Base Pair Fractions (EBPF) is an acceptable method to study the thermodynamic probability of base-pair formation in the target DNA structure at a given temperature. The ensemble pair fraction for a given DNA sequence s, denotes a base-pairing probability p_{ij} $(i < j)$, which is the probability that the i^{th} and j^{th} nucleotides of s, s_i and s_j, form a base-pair, considering ensemble of DNA structure.

Using NUPACK model at the following salt and parameter model, we calculate the ensemble base-pairing fractions for both the PT and BT structures.

Parameters used to calculate the thermodynamic properties of the BT and PT:
Temperature = 25.0 °C (varying from $\approx 25°$C to 70 °C)
Material = dna
Parameters= SantaLucia1998
Dangles = ’some’
Na^{+} = 0.05 M, Mg^{++} = 0.0115 M
Each DNA strand was considered at 1μ M concentration

Thermodynamics of the PT Structure: In the following we discuss the ensemble pair fraction probabilities, melting profile of the BT structure at different temperatures. As can be seen from the plots, PT structure has a melting temperature $\approx 62°\mathrm{C}$, and below this temperature all its six DRs form with high probabilities.

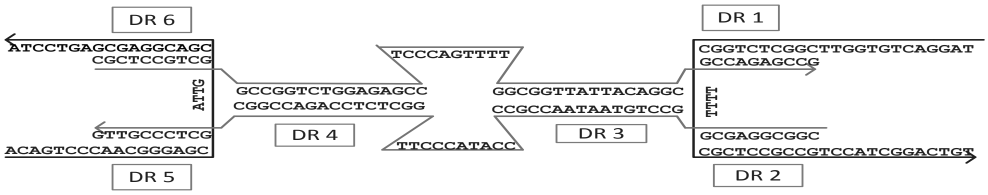


Figure 4.9: The PT structure and its Duplex Regions (DRs).

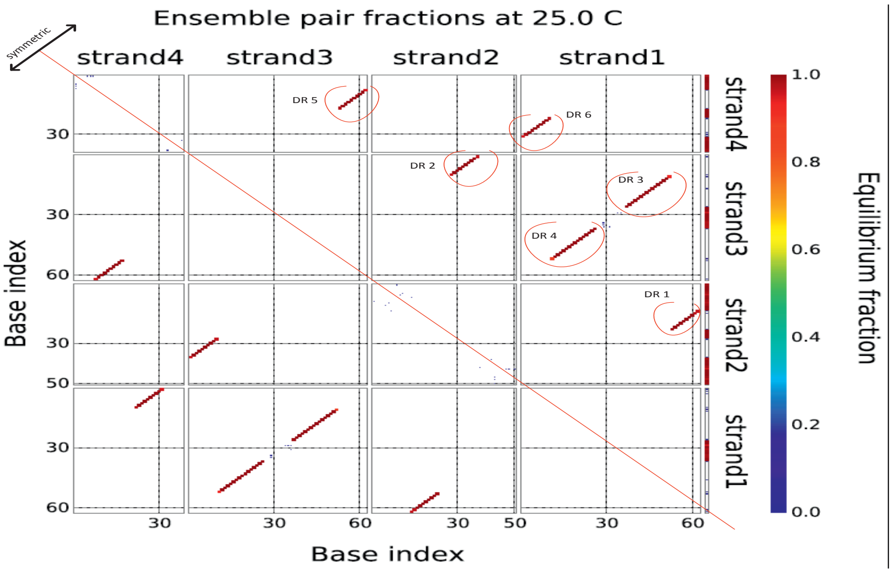


Figure 4.10: Ensemble pair fractions of the PT structure at 25 C. At this temperature, all the six DRs are formed with high probabilities

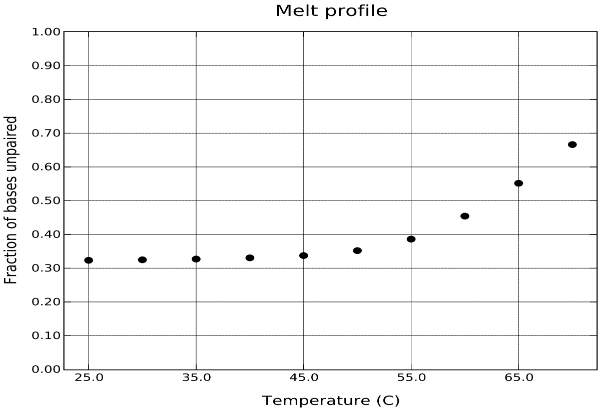
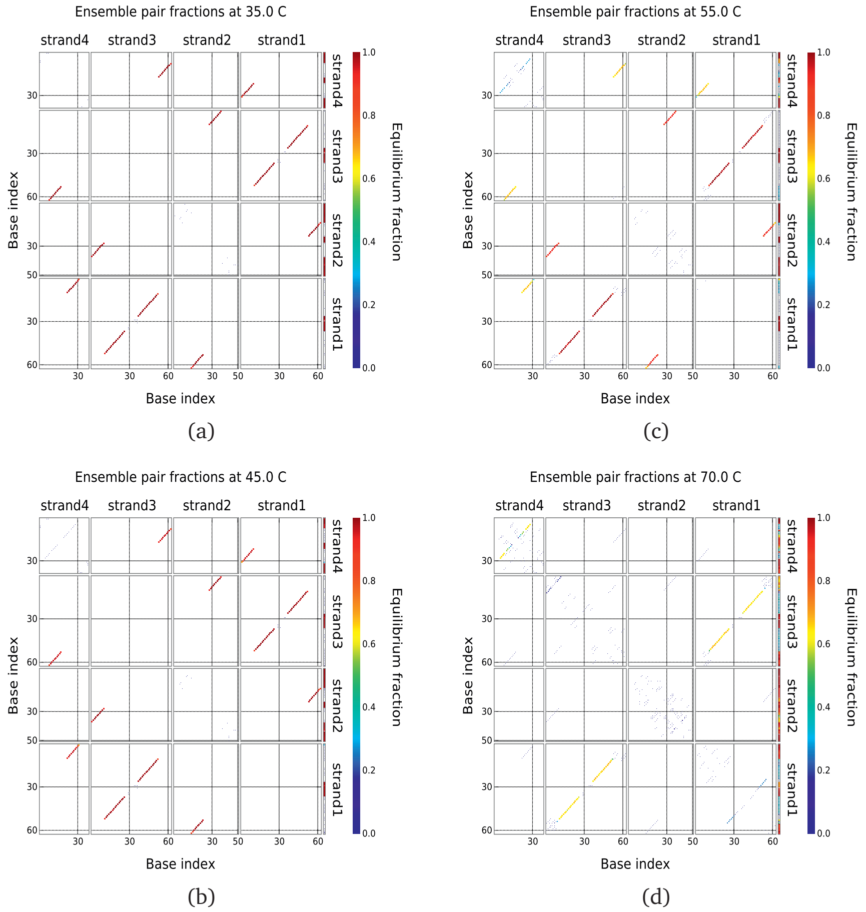


Figure 4.11: Melting profile of the PT structure.



(a)

(c)

(b)

(d)

Figure 4.12: Ensemble pair fractions of the PT at different temperatures. At T= 35 °C (a) and T= 45 °C (b), the different DRs are still formed. At T= 55 °C (c), the DR6,DR5, and DR1 has just started disappearing. Finally, at higher temperature T = 70 °C (d), the ensemble pair fraction of all the DRs is very low.

Thermodynamics of the BT Structure: Observe that the DNA sequences of the BT, which involves pseudo-knotted structure, were designed using heuristic search algorithm, thus the sequences are not perfect, but they are somewhat closer to forming the target structure of BT, as can be seen from the following ensemble pair fraction plots.

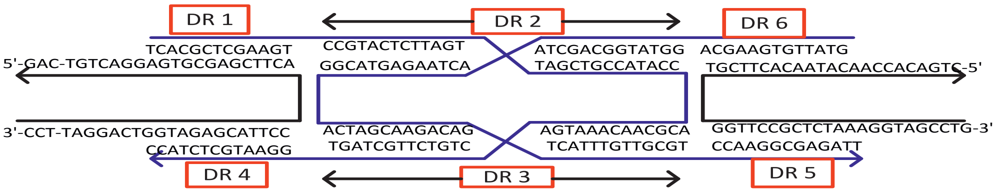


Figure 4.13: The PT structure and its Duplex Regions (DRs).

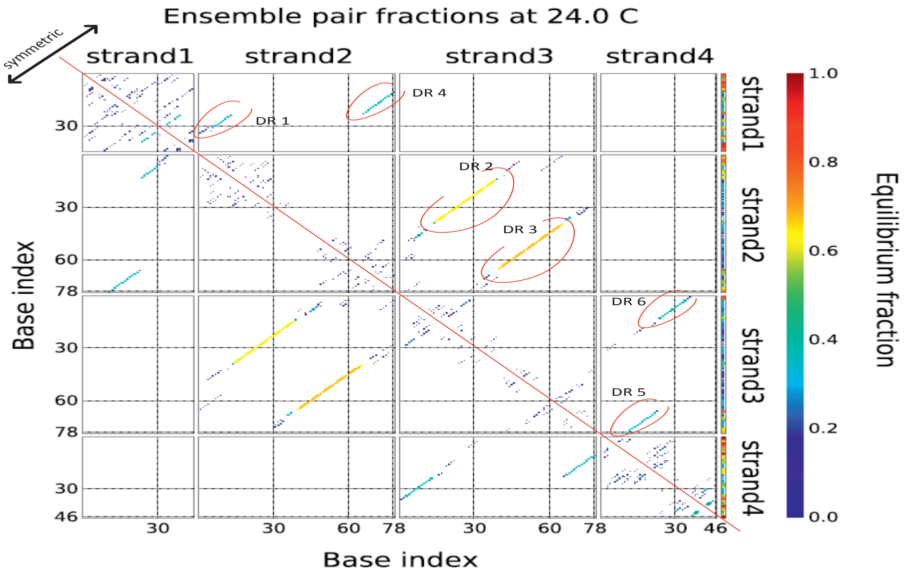


Figure 4.14: Ensemble pair fractions of the BT structure at 24 °C. At this temperature, all the six DRs are formed, but with low probabilities.

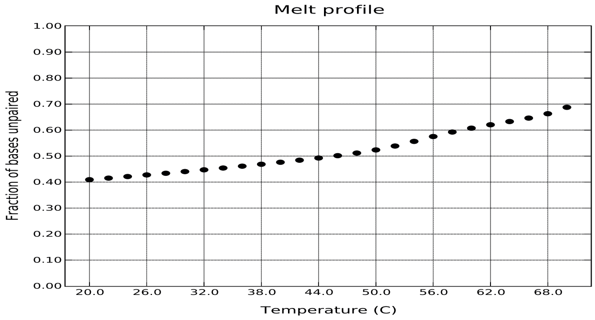
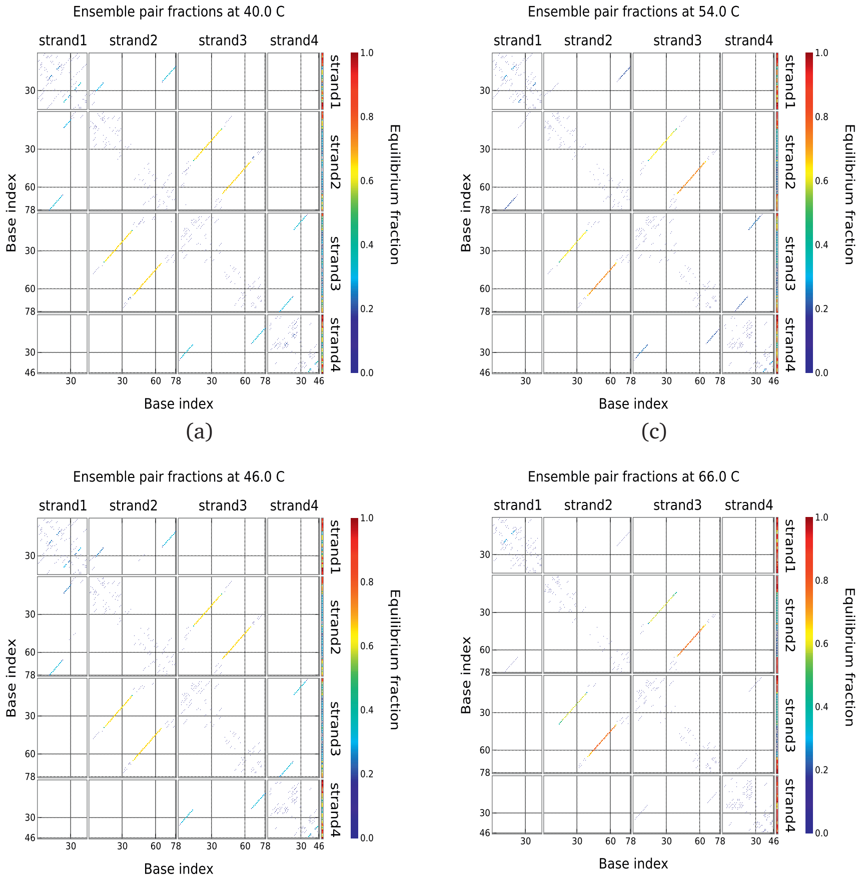


Figure 4.15: Melting profile of the BT structure.



(a)

(c)

(b)

(d)

Figure 4.16: Ensemble pair fractions of the BT at different temperatures. At T= 40 °C(a), the DR2 and DR3 are formed with not so high probabilities, but the other DRs are not formed with good probability; there are also 'spurious' base-pairing in the structure. Until the temperature, T ≈ 46 °C (b), the different DRs are visible to some extent. For higher temperatures, T= 54 °C (c), and T= 66 °C(d), the different DRs disappear, except the DR2 and DR3.

Thermodynamics of the BT-PT pair: We analyse the thermodynamic stability of the BT-PT pair at different temperatures by calculating the free energies of BT, PT, and BT-PT pair structures corresponding to their Minimum Free-Energy (MFE) structures. The MFEs at the different temperatures shown in Figure 4.17 are normalised: the free-energy of BT-PT pair at T= 25 °C is the lowest free energy, other free energies are normalized to this value. Hence it is the difference between the free-energies that is relevant for the analysis of relative stabilities of different structures. In these calculations we assume (based on the aforementioned thermodynamics of the BT and

PT structures) that the individual structures of BT and PT would be present as assembled units at T= 50 °C. If the pre-assembled BT and PT structures are mixed and annealed from (e.g., T = 40 °C) to the lower temperature (e.g., T= 25 °C), the BT-PT pair would assemble, as shown by relative MFEs of the different structures.

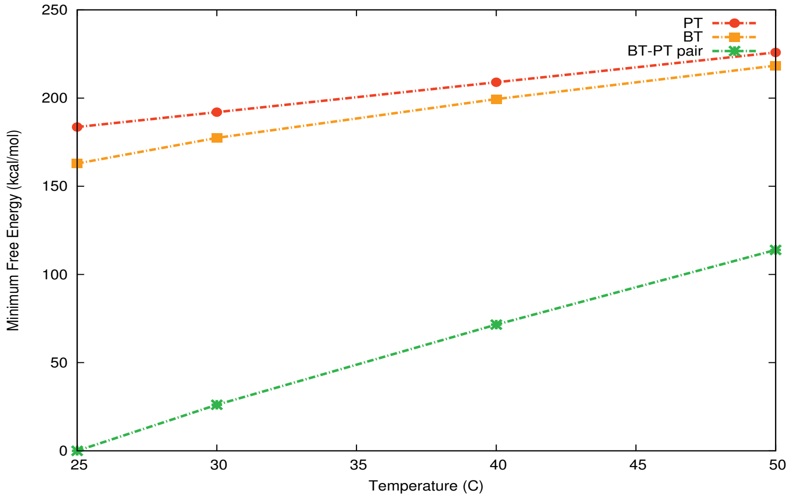


Figure 4.17: Minimum free energy of BT, PT, and BT-PT pair.

Wet-lab preparation of ETs: Having discussed the thermodynamic stability of the ET, we describe below a four-step wet-lab protocol for the preparation and self-assembly of ETs in the laboratory. First, for each ET its *BT* and *PT* are prepared separately after mixing a stoichiometric amount of their respective component DNA strands in the hybridization buffers followed by annealing from 90 °C to 40 °C. Second, the assembled BTs and PTs are mixed in *BT-PT* pairs in separate test tubes, and the mixture is again annealed from 40 °C to room temperature so as to form the individual ETs. Third, the duplex central-bulge-forming DNA strand is displaced via an external DNA strand. Fourth, the assembled *ETs* are mixed together with a supplied pre-formed seed structure at room the temperature. The seed structure nucleates the self-assembly of ETs, which grows into the programmed pattern as more and more tiles join.

4.6 An Example of Enveloped Tile Set Design

Figure 4.18 illustrates a Sierpinski tile set design using ETs. As shown in (a), each Sierpinski Rule Tile (SRT-00, SRT-11, SRT-01 and SRT-10) has a different combination of input and output sticky ends (complementary sticky ends are marked by *). The *BTs* are shown in (b) and the *PTs* are shown in (c). The Input and output sticky ends of the BTs and TEs are represented by (S^{jk}_{bi1}, S^{jk}_{bi2}),(S^{jk}_{bo1}, S^{jk}_{bo2}), and (S^{jk}_{ei1}, S^{jk}_{ei2}), (S^{jk}_{eo1}, S^{jk}_{eo2}) respectively, where $jk = 00, 11, 01, 10$ represents the type of Rule tile.

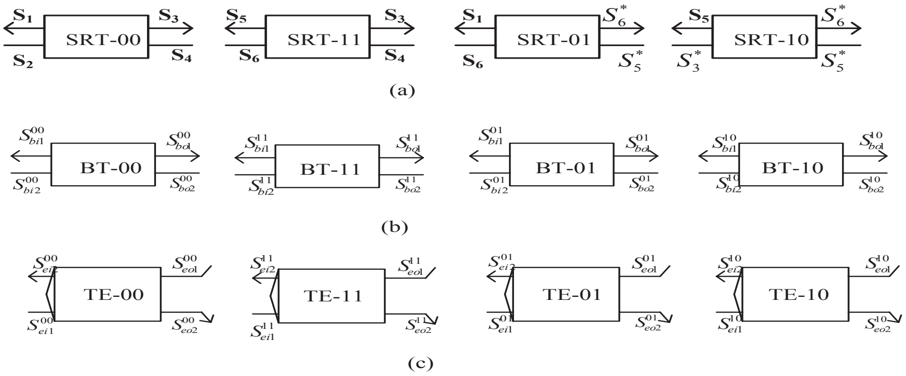


Figure 4.18: ET Set for Sierpinski Triangle: (a) Rule Tiles, (b)BTs used for Sierpinski ET set (c) Corresponding PTs.

Table 4.3: BTs Sticky Ends.

Name	DNA sequence
S_{bi1}^{00}	**5'-TCCTAGGACT-3'**
S_{bi2}^{00}	**3'-CTGTGTCAGG-5'**
S_{bo1}^{00}	**3'-AGGTACCTGA-5'**
S_{bo2}^{00}	**5'-GACCAAGTCC-3'**
S_{bi1}^{11}	**5'-CGTTAGGACT-3'**
S_{bi2}^{11}	**3'-GTTTGTCAGG-5'**
S_{bo1}^{11}	**3'-AGGTACCTGA-5'**
S_{bo2}^{11}	**5'-GACCAAGTCC-3'**
S_{bi1}^{01}	**5'-CGTTAGGACT-3'**
S_{bi2}^{01}	**3'-CTGGTTCAGG-5'**
S_{bo1}^{01}	**3'-GCAATCCTGA-5'**
S_{bo2}^{01}	**5'-CAGACAGTCC-3'**
S_{bi1}^{10}	**5'-TCCTAGGACT-3'**
S_{bi2}^{10}	**3'-GTCTGTCAGG-5'**
S_{bo1}^{10}	**3'-GCAATCCTGA-5'**
S_{bo2}^{10}	**5'-CAGACAGTCC-3'**

Table 4.4: PTs Sticky Ends.

Name	DNA sequence
S_{ei1}^{00}	**3'-ACAGTCC-5'**
S_{ei2}^{00}	**5'-ATCCTGA-3'**
S_{eo1}^{00}	**5'-TCCATGGACTGT-3'**
S_{eo2}^{00}	**3'-CTGGTTCAGGAT-5'**
S_{ei1}^{11}	**3'-ATCCTGA-5'**
S_{ei2}^{11}	**5'-ACAGTCC-3'**
S_{eo1}^{11}	**5'-TCCATGGACTGT-3'**
S_{eo2}^{11}	**3'-CTGGTTCAGGAT-5'**
S_{ei1}^{01}	**3'-ATCCTGA-5'**
S_{ei2}^{01}	**5'-CAAGTCC-3'**
S_{eo1}^{01}	**5'-CGTTAGGACTTG-3'**
S_{eo2}^{01}	**3'-GTCTGTCAGGAT-5'**
S_{ei1}^{10}	**3'-ATCCTGA-5'**
S_{ei2}^{10}	**5'-CAAGTCC-3'**
S_{eo1}^{10}	**5'-CGTTAGGACTTG-3'**
S_{eo2}^{10}	**3'-GTCTGTCAGGAT-5'**

There are three requirements for complementary DNA sequences among these sticky end sequences: between the *BT's* inputs and outputs, between the *BTs'*, and *PTs'* inputs and outputs and between the PTs' inputs and outputs. Thus the input and output sequences of the PTs are constrained to have a certain sequence of common bases c_1, c_2, shown in Figure 4.6b.

The design of the length of the sticky ends of the *PT* has to take into account a number of factors. These ends must both be able to protect the rule tile's outputs and, upon disengaging from the input sides, to bond with their input protector counterparts. Therefore, the overhangs have been designed such that the protective helix on the rule tile outputs measures 8 base pairs, keeping 2 nt unpaired nick on the inner side. Together with the toeholds, the same strands are able to form 7 nt bonds internally so as to let the *PT* dissociate from the *BT* via uni-molecular stand displacement. The DNA sequences c_1 and c_2 are common for all the *PTs* needed for a Sierpinski tile set. Furthermore, the DNA sequences ($c_1 + e_{t1}$, $c_2 + e_{t2}$) are 7 *nt* long, thus adding 3 bases for protruding toeholds on the output side of *ETs*. Tables 4.3 and 4.4 show a set of the designed DNA sequences of *BTs* and the corresponding *PTs* for the Sierpinski tiles.

4.7 Self-assembly guided by Enveloped Tiles

As illustrated in Figure 4.19, when a tile approaches a vacant site (I) of a growing tile lattice, the tile 'input toeholds' may attach with the bind site's sticky ends as the system is annealed. Due to presence of multiple types of tiles, both or single toeholds may match with the binding site ,as shown in Figure 4.19.

In case 1, the first state (kinetic discrimination of the inputs) *C(I)* consists of toehold binding followed by branch migration of 7 *nt* of the inputs of the *PT*, *C(II)*, that were bound with a Rule Tile's sticky ends. A stable binding of the Rule Tile to the assembly has thus occurred. At this stage the Rule Tile no longer needs to be protected by the *PT*. As the *PT* is released from its input side it will result in uncoiling of the clamp, *C(III)*, which primes a configuration in which the left part of the *PT* is free to swivel back towards its outputs where it is still protecting the output sticky ends of the *BT*. The released inputs start 'sniffing' the protruding toeholds at the outputs of the *PT*, *C(IV)*. The resulting toehold binding, followed by a further branch migration, *C(V)*, of the respective 4 *nt* will create a single thermodynamic component to disengage the *PT* and expose the *BT's* output sticky ends, *C(VI)*. The next incoming, matching Rule Tile provides additional thermodynamic impetus to further support migration of the *PT* away from the growing assembly.

The first stage of Case 2 is as in Case 1, except that only one of the toeholds binds, *E(I)*, and initiates branch migration at that input. All other sticky ends of the *BT* will remain protected, *E(II)*. As this will only occur on one of the input sides, all other sticky ends of the non-matching tile will remain protected by the *PT* and thus remain unbound to the assembly. Although the branch migration of *E(II)* may or may not proceed to completion, the otherwise fully protected tile will eventually disengage from the lattice

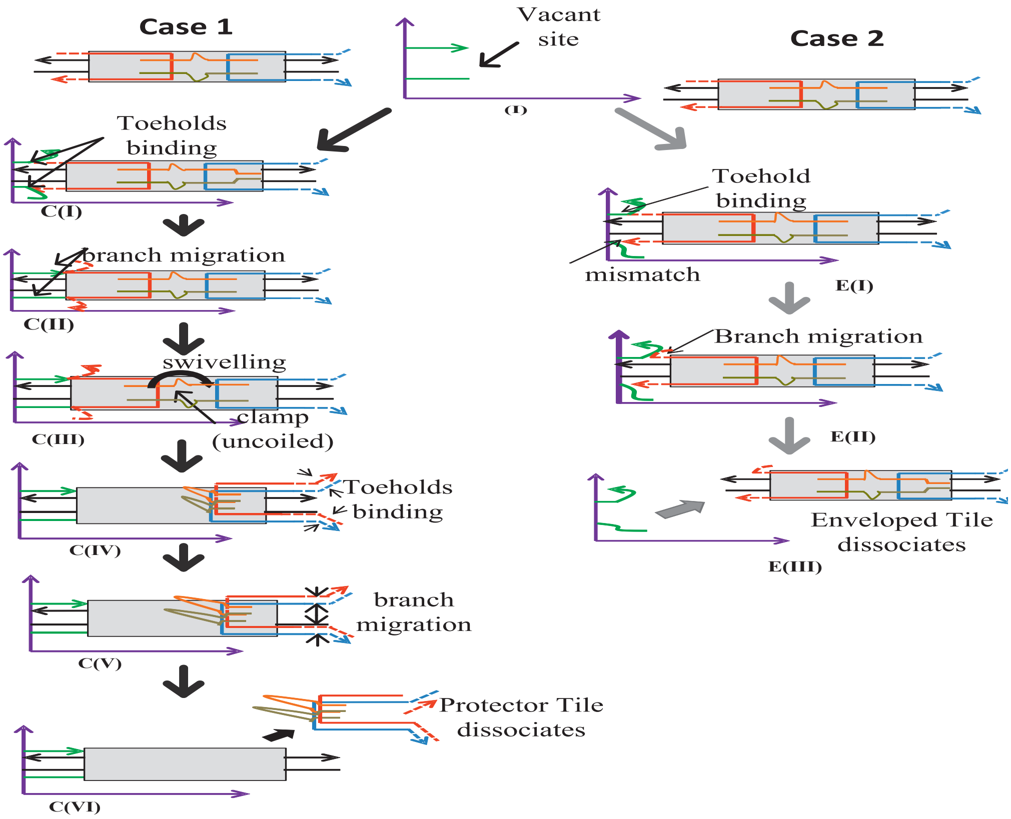


Figure 4.19: Self-assembly step states: (left) both inputs match, (right) only one input matches.

due to the inherently unstable nature of binding through a single input, *E(III)*. The uniqueness of this approach lies in the strategy applied to protect BT's output. The protection comes not just in the form of the binding of the outputs but further the distal clamp design protects the outputs from unbinding unless the tile is correctly annealed to a lattice.

4.8 Discussion

In this section we discuss the prospects of error-preventive tile assembly using ET. Following is an estimation of assembly time of an $n \times n$ tile lattice in the ET assembly framework. Detailed analyses of error-prevention and assembly time of ET are given in Chapter 6. However, the following simplistic analyses are useful to illustrate the kind of improvements we can expect to get from ET. A further discussion in this section presents the strengths and weaknesses of the ET, and a brief comparison with the other state-of-the-art techniques in the tile assembly error correction.

4.8.1 "One Size Fits All": Error-prevention prospects in Enveloped Tile Assembly

Growth Error-prevention: There are three types of errors in cooperative self-assembly of tiles, as illustrated in Section 4.1.1. The first and widely studied assembly error, growth error, occurs due to insufficient attachment (tiles binding by single sticky end) of tiles. For an error-free assembly process, tiles attaching by single sticky end should detach quickly before they are trapped by the addition of subsequent tiles. A slow detachment of an insufficiently attached tile may thus introduce growth error.

Considering the case of original tiles of Winfree's model, where tiles have no protection, and therefore kinetic rates of tile attachment involving single sticky end or two sticky ends have no discrimination. However, the ETs, introduce a kinetic discrimination between tiles binding by single sticky end and two sticky ends.

Due to fully protected nature of the ET 'outputs', there is no incoming cooperative binding from the output side of a tile as it binds using 'inputs'. However, if a mismatched ET undergoes successful deprotection during the engagement, a growth error may introduce to the assembly as the adjacent tile assembles in the lattice. However, the probability of such growth errors would be decided by the extent to which mismatching tiles are kinetically discriminated against the matching tiles, as illustrated in Figure 4.19. A strong kinetic discrimination may thus substantially reduce the possibility of such errors. The kinetic discrimination in the ET assembly mainly comes from the toehold-mediated strand displacement. Considering a kinetic discrimination of an order of magnitude of four[1], would assemble one erroneous tile out of 10000 tiles, which would be a significant improvement over a 10% error rate of the unprotected tile assembly.

Prevention of Facet Assembly Errors: In Winfree's tile assembly model [154], a facet assembly error is introduced by insufficient attachment of a tile that is followed by a subsequent favorable attachment of the adjacent tile, providing a cooperative reinforcement to the insufficiently attached tile. Mechanism of facet nucleation errors in the ET assembly is illustrated in Figure 4.20. ETs have fully protected 'outputs', therefore there is no cooperative binding possible from the protected sides of the engaged ET. However, if there is a spontaneous deprotection of the ET after a successful displacement of the engaged 'input' of the PT, new facet nucleation site would be created. Events of facet nucleation errors in the ET assembly would thus be extremely

[1] Kinetic rate of a toehold-mediated DNA strand displacement can be varied by million-fold simply by changing toehold over 1-6 nucleotides [174]. Therefore, two DNA displacement reactions that use toeholds of 2 nucleotides and 6 nucleotides, would provide a kinetic discrimination of an order of magnitude of four.

rare. It should be observed that in contrast to the partially protected tiles of Fujibayashi et al. [49], which have possibility of creating facet nucleation sites if there are unprotected tiles in the assembly mixture, ETs are immune to such errors even if there are BTs present in the same environment.

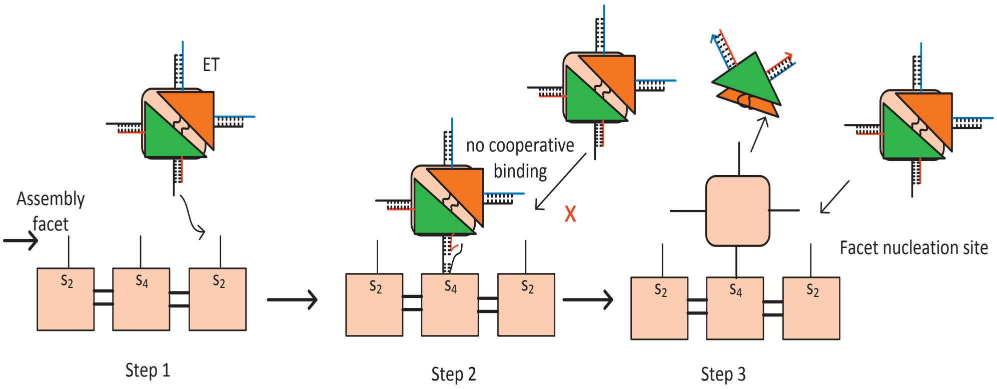


Figure 4.20: Facet nucleation errors in the ET assembly. An ET approaching a facet site (step 1) binds by its matching toehold (step 2). While ET is engaged with the facet site but its 'outputs' are still protected by the PT, there is no favourable (cooperative binding) attachment possible by a subsequent ET (case 1) or BT (case 2). A spontenesou deprotection of the PT may expose the enaged tile 'outputs', creating a facet nucleation site (step 3).

Prevention of Spurious Nucleation of Assemblies: Due to the protected nature of ETs, there is no possibility of triggering a spurious nucleation assembly, as illustrated in Figure 4.21. However, if there are unprotected tiles (BTs) in the assembly environment of the ETs, then two BTs may simultaneously bind to an ET, causing its deprotection. However, such events require three tiles to collide simultaneously (a termolecular reaction) , and thus would be rare. Furthermore, size of such spurious assemblies would be very small in the ET assembly, as subsequent tile attachments would face similar unfavourable conditions.

4.8.2 Assembly Time of Enveloped Tiles

We analyse assembly time of forming a (n-1)x(n-1) tiles lattice using an example of tile assembly that is seeded by a preformed L-shaped seed structure , as shown by a grey South-West border in Figure 4.22. Consider that the tiles self-assemble with the seed structure to fill the 4x4 space (shown by white squares).

Lets us consider Winfree's model [154] that uses unprotected tiles to self-assemble the lattice, as shown in Figure 4.22a. In this case, all the positions (✓) adjacent to the

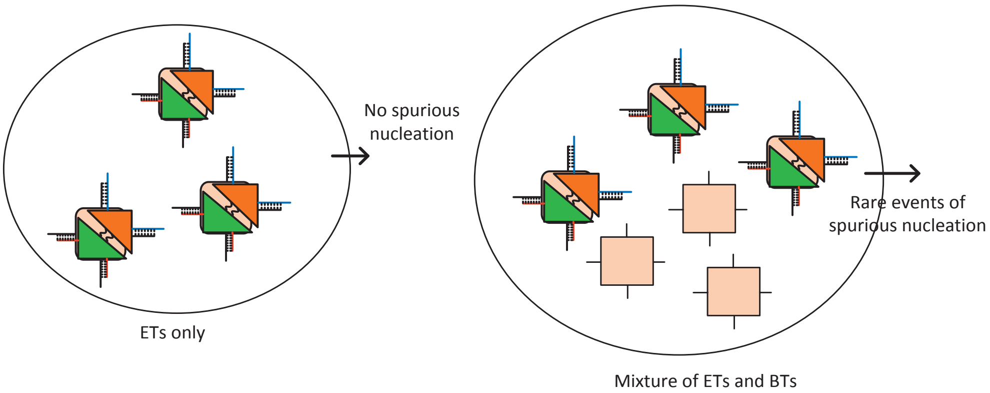


Figure 4.21: Spurious nucleation errors in ET assembly. (a) the assembly medium has only ETs, and thus trigerring a spurious nucleation event is unfavorable. (b) a mixture of ETs and BTs is present in the assembly medium. Unprotected tiles may bind with the protected tiles, deprotet them, and nucleate spurious assemblies

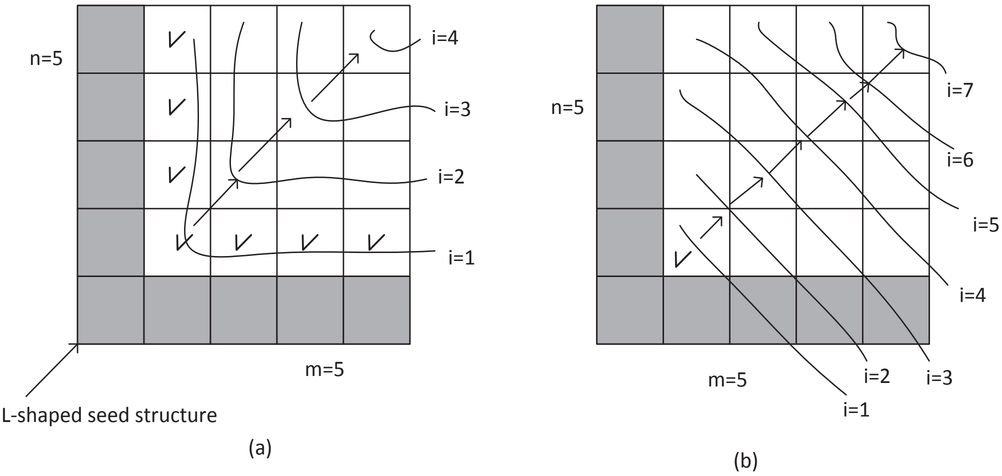


Figure 4.22: Assembly time in ET assembly.

L-shaped seed are valid binding sites for the unprotected tiles. Considering that the lattice will be formed layer-by-layer, as shown by curve lines ($1 \leq i \leq 4$). Using kinetic rate parameters of the tile assembly model (the rate of tile binding: r_f; the rate of tile detachment: $r_{r,b}$, where b is the number bonds by which the tile is attached), we can estimate assembly time of the 4x4 lattice, shown in the figure. The effective rates of tile binding by single bond and two bonds can be given by

1. An approximate effective rate of a tile binding by single bond: $r_s = (r_f - r_{r,1})$
2. An approximate effective rate of a tile binding by two bonds: $r_d \approx (r_f - r_{r,2})$.

If there are four types of tiles to form the lattice (the Sierpinski pattern assembly), the assembly time of filling a single position can be approximated as

$$t_{pw} = \frac{1}{4r_d} + \frac{3}{4r_s} \tag{4.1}$$

Approximate assembly time of a n x n lattice can be given as

$$\begin{aligned} T_{n\times n}^{winfree\ model} &= \sum_{i=1}^{n-1}(2n-2-i) \times t_p \\ &= \frac{(3n^2-7n+4)t_{pw}}{2} \end{aligned}$$

Similarly, we can estimate an approximate assembly time of 4x4 lattice formation in the ET assembly, as shown in Figure 4.22b. ET assembly process is directional: only a successful deprotection of the assembling tile facilitates the binding of subsequent tiles. Starting with the corner position (✓) adjacent to the L-shaped seed, ETs will assemble one-by-one in the subsequent activated lattice positions falling in the lines ($1 \leq i \leq 7$). Each active position may engage with a matching or mismatching ET. Number of active binding sites during the tile assembly can be given as
1. for i=1, active sites = 1
2. for i = 2, active sites = 2
3. for i = 3, active sites = 3
4. for i = 4, active sites = 4
5. for i = 5, active sites = 3
6. for i = 6, active sites = 2
7. for i = 7, active sites = 1

Considering the average assembly times of matching and mismatching tiles as t_{ma} and t_{mi}, respectively. For an assembly system of four types of ET, the assembly time t_{pe} of filling a single position in the lattice is $t_{pe} = \frac{t_{ma}}{4} + \frac{3t_{mi}}{4}$

An approximate assembly time of nxn lattice using ETs can be given as

$$\begin{aligned} T_{n\times n}^{et\ model} &= t_{pe}(n-1+\sum_{i=1}^{n-2} 2i) \\ &= (n^2-2n+1)t_{pe} \end{aligned}$$

The assembly times of the two models are related as

$$\frac{T_{n\times n}^{winfree\ model}}{T_{n\times n}^{et\ model}} = \frac{(3n^2 - 7n + 4)t_{pw}}{2(n^2 - 2n + 1)t_{pe}},$$

which gives $1 < \frac{T_{n\times n}^{winfree\ model}}{T_{n\times n}^{et\ model}} \leq \frac{3}{2}$

If the average assembly times required to fill a single site in the lattice are equal in the two models, the total assembly time of the nxn lattice in the Winfree's model would be larger than the assembly time of ETs. However, due to the complex nature of physicochemical processes involved in the assembly of ETs, the time required to assemble a single ET may be larger than the time in the unprotected case. Therefore, in practice, assembly time using ETs is expected to be larger than the unprotected tiles. However, Winfree's model requires a very slow lowering of temperature during the annealing so as to self-assemble the lattice with minimum errors at a slightly supersaturation, which takes days to conduct a laboratory implementation. In contrast to this, a laboratory implementation of ETs assembly may not require such slow annealing process, and therefore a swift annealing may compensate for a possible slowdown in the process.

Activation Mechanisms of the ET and Signal-passing Tiles

There have been growing interests in the design of DNA-based nanostructures with dynamic control so as to enable reconfiguration and activation in the assembled DNA structures [165, 83, 60, 171, 114, 97], which otherwise are static in nature. Design of the Enveloped Tile is a step in the direction of enabling dynamic control in the self-assembly of DNA tiles. We described how the Enveloped Tiles could be used to control local interactions between tiles as they assemble so that illegitimate binding could be prevented in the cooperative process of algorithmic tile assembly. The mechanism of input-binding triggered activation in the Enveloped Tile is based on the toehold-mediated strand displacement process [168, 174], making it robust against temperature changes, varying ionic concentrations and pH changes.

In addition to the error-correction feature, the Enveloped Tile presents a general mechanism to implement sequential assembly of tiles, as demonstrated in the experimental implementation of an one-dimensional tile assembly process of upto five distinct tiles using Signal-passing Tiles [97]. The designs of Enveloped Tile and Signal-passing Tile are similar, except that Enveloped Tile design takes extra care of cooperative binding so as to prevent errors in the algorithmic self-assembly. There is also theoretical work suggesting the use of Origami-based Signal-passing Tiles [95, 96]. It would be interesting to see how the design of one-dimensional Signal-passing Tile is extended to the Origami-based Tiles. This would perhaps require similar consideration for the synchronization of binding and deprotection events in the multiple sticky ends of Origami-based Signal-passing tiles, as adopted in the case of Enveloped Tiles. Therefore, the design of Enveloped Tile can be seen as an extension to the one-dimensional Signal-passing Tile.

Owing to the similarity between the two tile designs, a further interesting aspect in the study of Enveloped tiles would be to establish limits on what we can assemble robustly using algorithmic self-assembly of Enveloped Tiles within the experimental limit of seven steps, as demonstrated in the case of Signal-Passing tiles [97]. In the algorithmic self-assembly, this limit would translate to the size of largest possible tile set that can be designed using the Enveloped Tiles. Although the algorithmic assembly of simple tile lattices, such as Sierpinski pattern [112] that requires seven types of different tiles would be feasible using Enveloped Tiles, the assemblies of larger size tile sets may presently not be possible using Enveloped Tiles.

Error-correction in the ET and other related techniques

Error-correction mechanism in the Enveloped Tile is closely related to the activation-based error-correction techniques, e.g. Activatable Tiles [83] and Protected Tiles [49]. In contrast to the design of Activatable Tile [83], one of the advantages in the Enveloped Tile is that there is no usage of restriction enzymes, and therefore its assembly process is autonomous and robust against variation in the temperature and other chemical factors. Further, the Enveloped Tile has been carefully designed so as not to keep any exposed sticky ends on the output side, so long as the input sticky ends of the tile have not correctly bound with the engaged lattice site; the activation on the output sticky ends is passed from a correct binding event at both input ends of the engaged ET. However, there is such possibility of erroneous binding in Protected Tiles [49] due to partially protected output sticky ends.

In the error-correcting redundant tile sets [156, 22], each tile is replaced by a $k \times k$ block of tiles. Therefore, the error-correction in the redundant tile sets comes at a price of adding k^2 times more tiles in the finally assembled lattice, and designing such larger tile sets with orthogonal sticky ends is always challenging.

Thermocycling-based robust error-correction technique [60], as briefly described in Section 4.1.2, identifies permissive and non-permissive temperature ranges near the thermodynamic equilibrium of *Capture Tiles* and $PX - JX_2$ devices. In a cyclic-fashion, the temperature of the system is raised and lowered so as to ensure that only correctly matching Capture Tiles are added to the pair of $PX - JX_2$ devices. The approach is thermodynamic and do not involve any kind of activation of tiles. The error correction using this technique has been experimentally demonstrated to assemble individual capture Tiles, and as such there is no limitation to apply it in the algorithmic tile self-assembly. However, if there is a choice for a robust, isothermal and kinetically controlled error-correction, the Enveloped Tiles can serve as an alternative to the Thermocycling-based approach.

4.9 Summary

Correction/prevention of assembly errors is crucial for the design of algorithmic self-assembly tile systems. In this chapter, we discussed the design of error-correcting Enveloped Tile and its assembly mechanism. The ET consists of a DX molecular DNA structure of BT and a protection element that covers the inputs and outputs of the BT from illegitimate cooperative binding as it assembles. ETs initially stay inactive, but a correctly matching engagement at the inputs of the ET dissociates its protection, as tile assembles correctly. The design and self-assembly of ETs is enzyme-free, and error-prevention in the tile self-assembly is achieved without any increase in the size of the final assembled structure. Although each ET assembles sequentially i.e., the next tile joins only after the present tile is completely assembled, multiple tiles can join at distinct sites at the same time, and thus parallelism is retained.

To demonstrate the general applicability of the approach, we designed a complete ET set for the Siepinski Triangle assembly. The same approach can be adopted in principle to converting any tile set into corresponding ETs. Further, using simplistic analysis we discuss the prospects of error-prevention and assembly timing in the ET self-assembly.

The ET self-assembly seems a simple yet promising approach to designing reliable tile self-assembly systems that can be adopted for laboratory experiments. However, before conducting any laboratory experiments using ETs, it requires further study of thermodynamics, biophysics and kinetics of the deprotection mechanism of the ETs. This would allow us to explore the feasibility and design space using parameter variation. In the next chapter, we discuss the coarse-grained Molecular Dynamics simulations to study the thermodynamics and kinetics of deprotection mechanism of the ET.

5 Molecular Dynamics Simulations of Enveloped Tile Structure

In the previous chapter we discussed the topology, DNA sequence design, and thermodynamics of the ET. The ET consists of two components: DX molecular DNA tile [157] as a BT, and PT). The *PT* is associated with the *BT* and protects its sticky ends from illegitimate bindings that cause assembly errors. Therefore ETs remain inactive until a nucleating seed is exposed to them, which initiates the self-assembly of tiles. During the assembly process, ETs engage with active sites in the growing lattice, where they lose their PTs as BTs become the part of the lattice. The deprotection of ET occurs in a two-step process: first, the ET 'toeholds' bind with the sticky ends coming from an active lattice site, which causes downward branch migration in the PT 'inputs'; second, the released PT 'inputs' bind with the other side of the PT structure through a intramolecular engagement, which ultimately dissociates the PT as the BT gets fixed to the lattice site.

As described above, an *ET* goes through several physicochemical changes during its self-assembly. The deprotection of ET particularly is crucial for a successful tile self-assembly. Studying thermodynamics of such process is a first step to verify its feasibility and associated kinetics. In this chapter we present coarse-grained molecular dynamics simulations of the ET deprotection process.

5.1 Thermodynamics and Kinetics of Chemical Reactions

Characteristics of chemical reactions are often studied in terms of thermodynamics and kinetics. Thermodynamics of a chemical reaction gives an understanding of the relative stability of its reactant and product species, while the kinetics characterises how quickly or slowly the species react.

Free Energy Profile and Reaction Coordinate

A *Free Energy Profile* (Free Energy vs. reaction coordinate, as shown in Figure 5.1) is a typical way for a qualitative study of chemical reactions. Potential Energy of a chemical reaction is an aggregate of energy calculated from the atomic configurations, as the reactants are transformed into products. The *reaction coordinate* is thus a generalized coordinate that does not represent the time of progress of a reaction, but in general it is a function of the Cartesian coordinates in which the atomic configuration of the reactants and products are represented.

The energy profile of a simple chemical reaction (also termed elementary reaction) typically consists of three energy states corresponding to: 1) Reactants (R), 2) Products (P), and 3) Transition State (TS), as shown in Figure 5.1(a). Energy profiles of non-elementary reactions may consists of one or multiple intermediate states (I), as shown in Figure 5.1(b). Free energy difference $\Delta G^{\circ} = -RTln(K)$, where K is the equilibrium constant of the reaction) relates to the relative thermodynamic stability of the reactants and products species in a chemical reaction. A reaction with $\Delta G^{\circ} < 0$ is known as exothermic (thermodynamically favorable), while a reaction with $\Delta G^{\circ} > 0$ is called endothermic (thermodynamically unfavorable). The 'energy barrier' (activation energy, E_a) between reactants and products is a measure of kinetic rate of the reaction. A reaction with smaller E_a would be faster than the reaction with a larger E_a. Thus, free energy plots are very useful in a qualitative analysis of: 1) whether a reaction is favourable (exothermic) or not (endothermic); 2) which reaction has a faster kinetic rate; 3) if a reaction is thermodynamically feasible and kinetically fast enough to occur in a practically useful time scale.

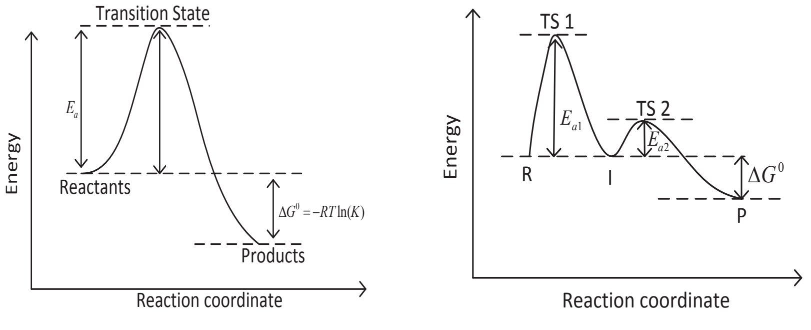


Figure 5.1: Free energy profiles of chemical reactions. (a) Chemical reaction has no intermediate state in. (b) Chemical reaction with one intermediate state.

5.2 Free-Energy Profile and Molecular Dynamics Simulations

Computer simulations provide an economical, and in a sense, more practical approach to establishing a proof of concept of a theoretical design. Molecular Dynamics (*MD*) simulations [46] are used to study the structural and dynamical properties of molecular assemblies in terms of microscopic interactions. Properties of a molecular system studied through *MD* simulations can be tested against macroscopic observations of the same molecular system obtained from laboratory experiment. If there is a mismatch between the two, the molecular model needs to be improved. On the other hand, *MD* simulations may also be used to test a theoretical model. If the simulations of a system do not match with those obtained from an approximate analytical theory of the same system, the approximate theory is flawed. Thus, the computer simulations can be considered equivalent to the experimental study in the later case. Furthermore, one can explore the design space of a molecular system through simulations, which can provide further insights to designing experiments.

MD simulations have been widely used to study molecular behaviors in biochemistry, biophysics, material science, and recently in nanotechnology. In a *MD* model, intermolecular interactions are represented by potential functions, that are modelled using quantum mechanics or classical mechanics. Quantum mechanics based *MD* models are very detailed, and thus, their use is limited to studying interactions between only a pair of molecules (e.g., nucleotide-nucleotide) due to high computation cost. In a classical mechanics based *MD* model, the potential function is reduced from a quantum level to a classical level, thus simulations can be relatively fast. Classical mechanics based *MD* models are widely used for smaller molecular systems of few molecules, such as a short DNA strand. However, for larger systems of molecules, where simulation times are larger than the microsecond range, both these approaches fail to provide reliable simulations with realistic computer resources [82].

Although theoretical methods, such as the Worm-Like Chain WLC [104] have been used to study the properties of larger molecular polymers, such models do not capture detailed biophysics and dynamics of the molecular processes. In between the detailed molecular models and theoretical models, there is a third approach of molecular dynamics modelling: coarse-grained *MD* models [116, 55, 32, 64, 76]. These models, such as *oxDNA* [92], employ a reduced number of degrees of freedom between molecules represented as rigid bodies, and the effect from solvent molecules is ignored. Although these simplifications put a limit on the realism of the oxDNA-based simulation studies, the model has produced a number of interesting studies of DNA-based systems [93]. That includes: 1) understanding the thermodynamics and kinetics of both in-equilibrium (e.g., DNA hybridization, DNA hairpin loop formation) and out-of-equilibrium DNA systems (e.g., toehold-mediated strand displacement and related dynamic systems of DNA); 2) understanding the underlying biophysics of DNA structures so as to rationally engineer DNA systems; and 3) studying the intermediate states of self-assembly processes, which can not be observed by experiments. In particular, it has successfully reproduced the thermodynamic and kinetic properties of DNA hybridization [94], bending [63], elasticity and torsion [111] etc.

A standard approach for calculating thermodynamic properties of computational models is the Metropolis algorithm [86]. A drawback with this approach is that only moving single particles at a time results in slow equilibration for systems with strong attractions. This is true for DNA strands, where collective diffusion is strongly suppressed if nucleotides are moved individually. Simulations can be made more efficient by using the Virtual Move Monte Carlo (VMMC) algorithm proposed by Whitelam and Geissler which allows for collective diffusion using cluster moves of particles

Thermodynamics simulations in the ox-DNA model are obtained using the Virtual-Move Monte Carlo (VMMC) algorithm [150]. The VMMC algorithm is a standard technique for calculating thermodynamic properties of intramolecular and intermolecular interactions. The basic idea behind the algorithm is to select a particle randomly and add the neighbouring particles based on probabilities derived from anticipated energy changes of the moves. Dynamics of a DNA system is simulated by defining *reaction coordinates*, also known as order parameters, which represent macroscopic properties common to a group of microstates of the system. Thus, using a carefully designed reaction coordinate, the simulation system is biased towards the desired behavior that provides a free-energy profile with respect to the designed reaction coordinate. The free-energy profile thus provides thermodynamic behavior which can further be used to extract the kinetic behavior of the DNA system.

Sometimes the configuration space of a simulated system faces thermodynamic barriers which are hard to cross by using direct sampling in VMMC. Simulation of such systems will take infinite time, and thus cannot feasibly explore the entire configuration space. However, the technique called *Umbrella Sampling* (US) [146] enables easier sampling of such systems. In US, the free-energy landscape near energetic barriers is first flattened by multiplying with artificial weights, and then, after sampling, the resulting distributions are demodulated again using appropriate weights. This makes sampling easier and speeds up the simulation, which otherwise would take an infinite time, while still producing relevant results as indicated by the simulations of several DNA systems [92]

Calculation of Free-Energy Profile

Several quantities of interest in a chemical system can be characterised by calculating free-energy differences between states of the system. In statistical mechanics, macroscopic quantities of a system are represented in terms of the properties of individual molecules making the system. Macroscopic quantities are defined as *ensemble* averages. An ensemble is a collection of all possible systems which have different microscopic states but have an identical macroscopic or thermodynamic state. Using statistical mechanics equations, free-energy differences may be expressed in terms of averages over ensembles of (3N) atomic configurations of a N-body system.

The *canonical* ensemble considers the system at constant volume at a fixed temperature (i.e., in contact with a thermal reservoir). Using statistical mechanics, the canonical ensemble average of a N-body system is given by

$$< \mathcal{A} >= \iint dr^N dp^N d\Omega^N \mathcal{H}(p^N, r^N, \Omega^N)\rho(p^N, r^N, \Omega^N),$$

where $\mathcal{A}(p^N, r^N, \Omega^N)$ is a microscopic property of interest and it is expressed as a function of the center of mass momenta, *p*, the center of mass positions, *r*, and angular orientations, Ω, of the N-body system. The integration is over all possible variables of *r*, *p*, and Ω.

The probability density of the ensemble is given by

$$\rho(p^N, r^N, \Omega^N) = \frac{e^{-\frac{\mathcal{H}(p^N, r^N, \Omega^N)}{k_b T}}}{Q},$$

where *Q* is the *partition* function, *H* is the *Hamiltonian*, and k_B is the Boltzmann's constant.

Simplifying the Hamiltonian, the above expression gives

$$\rho(q) \propto e^{-\frac{E(q)}{k_b T}},$$

where q is a multidimensional vector representing the microscopic coordinates of the reaction system. Due to the multidimensional nature of the vector, q, the above expression is difficult to analyse. However, by combining the variables of interest in terms of an *order parameter* of the reaction, we can study the properties the system. The *order parameter* gives information about the extent to which the reaction has progressed towards a macroscopic state of the interest.

The probability distribution for an order parameter s is

$P(s) \propto \int dq e^{-\frac{E(q)}{k_B T}} \delta(s - s(q))$

This probability can be expressed in energy units as a free energy landscape F(s):

$f(s) = -k_B T \; ln \; P(s)$

5.3 The oxDNA Model

The oxDNA [92] is a fully three-dimensional coarse-grained molecular model of DNA in which each nucleotide is presented as a three-dimensional rigid body having three interaction sites, as shown in Figure 5.2. Thus a chain of N nucleotides of DNA will consists of a total 12 N interaction parameters — 6N for spatial dimensions and 6N for momenta. In oxDNA, all interactions are modelled pairwise, and interaction potential is modelled considering backbone connectivity, excluded volume, Watson-Crick bonding, stacking between adjacent nucleotides, stacking between non-adjacent nucleotides, and cross-stacking.

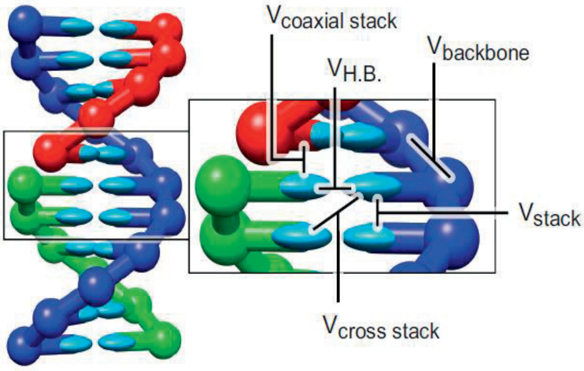


Figure 5.2: Representation of a duplex DNA constituting three different strands of DNA in oxDNA model [144]. Each nucleotide (treated as molecule) is represented by a spherical backbone and an ellipsodal base. Different interaction terms are shown in an enlarged view.

There are several simplifications that have been incorporated in oxDNA to make it computationally affordable for simulating the properties of larger DNA systems. First, it captures the duplex formation from single strands, but at higher salt concentrations, and thus, electrostatic interactions that might be significant forces at lower salt concentrations are ignored. Second, it does not capture the helical structure of the DNA duplex: the major and minor groves have no distinction in the duplex. Nonetheless, it has successfully reproduced the thermodynamics and structural properties of ss-DNA and ds-DNA [144], including detailed studies of DNA hybridization [94], toehold-mediated DNA strand displacement [141], overstretching [111] and dynamic systems, such as DNA tweezers [92] and DNA walkers [94].

The oxDNA model and its intermolecular interaction potentials have been described in detail elsewhere [93, 144][1].

[1] Implementation code of oxDNA is available for download (http://dna.physics.ox.ac.uk/) under GNU General Public License.

5.4 Simulation of Deprotection in Enveloped Tile

The basic element of this simulation study in oxDNA is a DNA structure, shown in Figure 5.3. This structure, a PT, is assembled using four ss-DNA (length of different sections is marked in the figure). The central section of the PT is a bulge of 10 nt. The two ends of PT (marked by green and red) are designed complementary so that the entire structure could form a loop (similar to a hairpin) as the two sets of complementary ends bind together, where the loop is formed by the bulge section and stems are formed by two ends of the PT.

5.4.1 The Folding Reaction of The Protector Tile

The *PT* structure (Figure 5.3) is the centerpiece in the *ETs*, and plays crucial role in a correct self-assembly of tiles. For a correct self-assembly of an *ET*, it is essential that there is a successful deprotection of the tile. The deprotection of ET involves: 1) an intramolecular engagement between the two ends of the PT, where the *PT* structure folds after it is released from the input side; 2) In a toehold-mediated displacement during the intramoleculr engagement, the PT spontaneously dissociates from the ET. Herein, we study the thermodynamics of *PT* 'folding' and the dynamics of ET 'deprotection' by simulating their free-energy profiles using oxDNA.

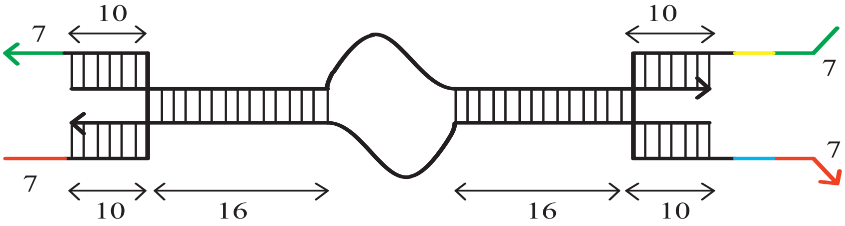


Figure 5.3: PT structure

Simulation method: Simulations of the folding reaction were performed with a virtual-move Monte Carlo (VMMC) algorithm [150] at temperature of 298 K. As the free-energy barrier between typical open (e.g., Figure 5.4) and folded states (e.g., Figure 5.6b) is large, the transition between the two macrostates constitutes a rare-event. Umbrella sampling, a technique that allows for the biased sampling of states with respect to an order parameter, was employed to sample the barrier crossing in reasonable computational time.

We use a two-dimensional order parameter Q =(Q_{ee},Q_{bp}) to characterize the transition. Q_{ee} is a discretized measure of the distance of closest approach between the

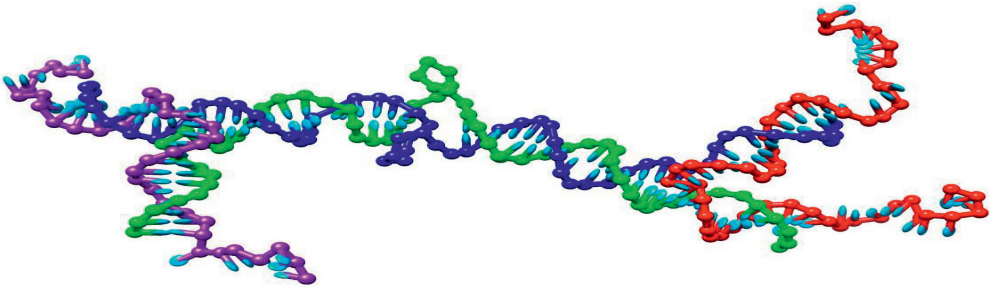


Figure 5.4: 3-D topology of the initial configuration (UNFOLDED state) of the PT Structure

complementary sticky ends. Q_{bp} is the number of base pairs formed between complementary sticky ends,where $0 \leq Q_{bp} \leq 7$.

To further improve computational efficiency, umbrella sampling was windowed to separately sample the open and folded states of each *PT* structure. For the window associated with the open state, the system was restricted to $Q_{ee} = 0$; for the window associated with the cyclized state, the system was restricted to $Q_{ee} = Q_{ee}^{min}$ (the value corresponding to the shortest distances between sticky-ends). Simulations were run until convergence to within 5% for each window.

The sampling windows overlap at $(Q_{ee}^{min}, Q_{bp}^{0})$; results were combined by normalizing each window so that the free energies were equal for this value of the order parameter. As there is only one well-defined overlap between the values of the order parameters for both windows, more complex approaches, such as the weighted histogram analysis method [77], were unnecessary. To further simplify sampling, we forbade the formation of base pairs that are not intended in the design of the system (non-native base pairs).

Results: A free-energy profile of the PT 'folding' reaction obtained from the oxDNA simulations is shown in Figure 5.5. The bulge loop is 10 nt long and consists of only 'T' bases. The reaction coordinate (represented by x-axis) for this simulation study has been designed in such a way that negative numbers correspond to UNFOLDED states of the structure, while positive numbers correspond to FOLDED states. The reaction coordinate representing FOLDED states (numbers, 1-7), denotes base pairs formed during the folding reaction. There are two branches in the energy plot for FOLDED states, one for each arm of the structure. On the y-axis, free energy (Δf) of the folding reaction is represented in the units of $\beta = \frac{1}{k_B T}$, where k_B is Boltzmann constant and T is the reaction temperature. The Free-energy of different states shown in this plot is normalized. Hence, it is the difference between free-energies of two states that is relevant.

A few observations about the free energy profile are as follows: The difference of free energies of fully FOLDED state and the Fully UNFOLDED state is negative, meaning that the most stable state at this temperature is the FOLDED state. This is desirable for the

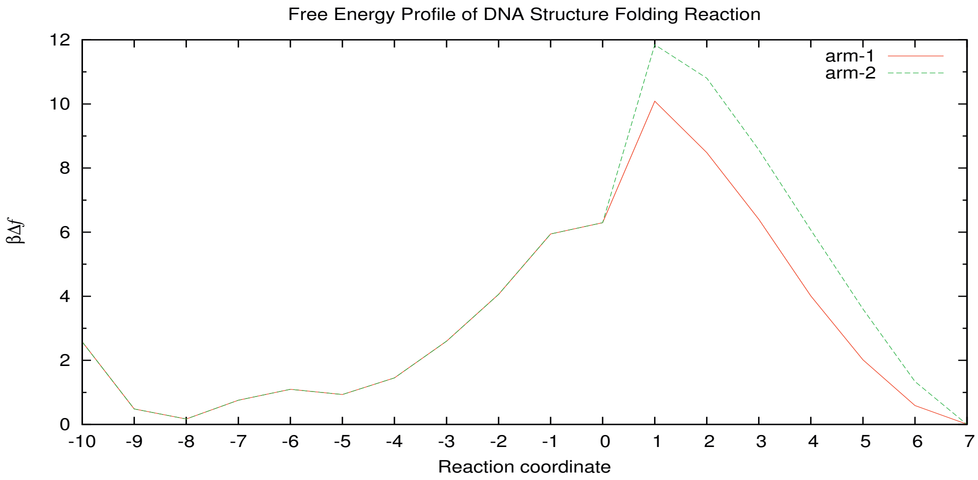
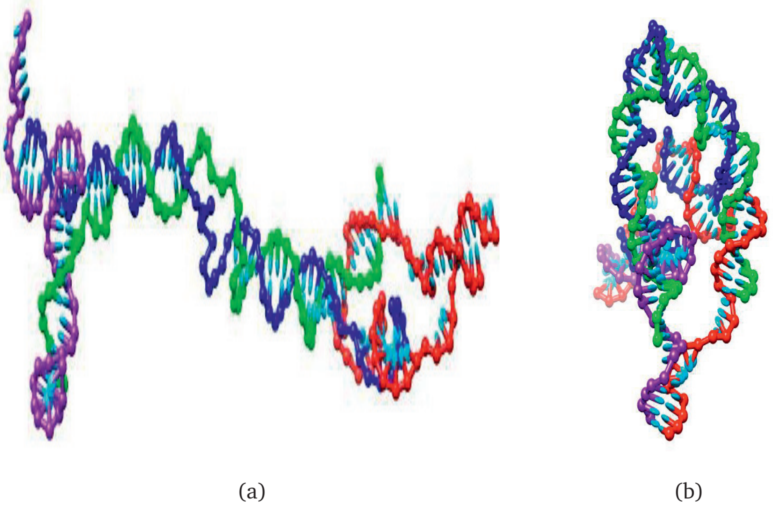


Figure 5.5: Free-energy profile of the PT folding process ($T = 25°\mathrm{C}$)

stability of the folded structure. It is intended that the FOLDED structure is stable once it forms after the PT gets detached from the BT.



(a)

(b)

Figure 5.6: 3-D topologies of the PT Structure during the folding. (a) a cross-talk state in the PT folding. (b) PT structure after folding successfully

There is a dip in the free-energy of UNFOLDED profile at reaction coordinate "-8", which signifies that there are unwanted base-pairing between two arms of the structure. This could be avoided by redesigning sequences so as to reduce the cross-talk and avoid partially bonded structures with no intended base-pairing.

There are peaks in free-energy profile at reaction coordinate value "1". Such an abrupt increase in free energy is attributed to entropy cost of localizing the first base-pair of the complementary sequences. This is well established result documented by several studies related to complementary binding of two DNA strands.

We further simulated the folding reaction of the PT structure at different temperatures, as shown in Figure 5.7. It can be observed from the free-energy profiles that the free energy of UNFOLDED states is quite similar at different temperatures. Therefore, it can be inferred that the dynamics of the folding reaction is independent of temperature as long as there is no base-pair formed. However, there is a significant difference in free-energy of FOLDED states at different temperatures, since base-pairing (hydrogen bonding) energy depends on the reaction temperature.

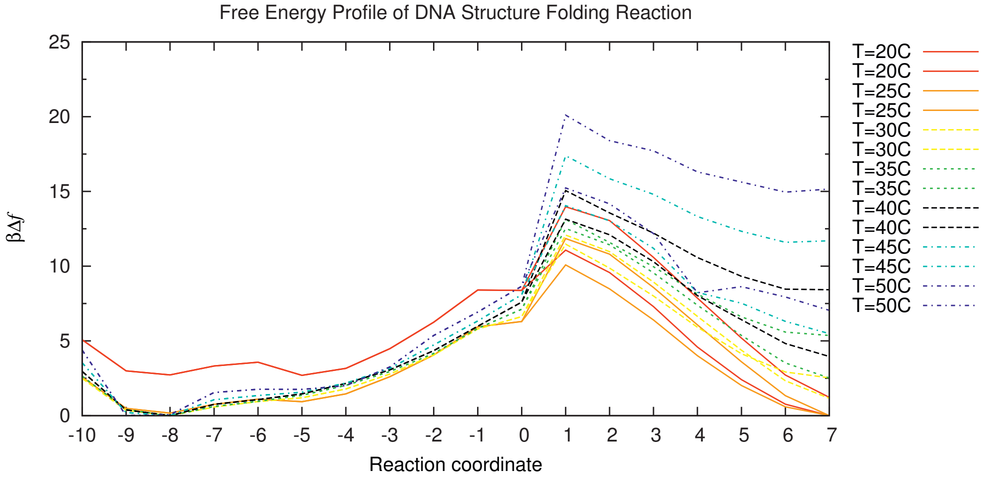


Figure 5.7: Free-energy profile of PT folding reaction at different temperatures

It can be seen from the free-energy profiles that the completely FOLDED state has lower free energy than the completely UNFOLDED state for $T \leq 40°\mathrm{C}$. Therefore, a reaction temperature should be maintained below $40°\mathrm{C}$ so as to keep the PT in a FOLDED state to avoid unintended reactions and cross-talk after it is displaced from the BT.

5.5 Summary

The free-energy profile derived from the oxDNA simulations establishes that: 1) the folding reaction of PT is thermodynamically feasible at room temperature; 2) free-energy profile of PT folding is dependent on reaction temperature. Therefore, as intended in our original hypothesis of BT-PT pair that the PT structure should be able to exist in two states, can be established from this study. This study also suggests that DNA sequences of two arms of PT should be redesigned to reduce the cross-talk so as to avoid spurious stable structures during the folding process. However, considering the

BT-PT pair together, as the PT ‘ouputs’ remain bound with the BT, such cross-talk would not be relevant for the PT folding reaction during the self-assembly process.

6 Kinetic Modelling of Enveloped Tile Assembly

Self-triggered activation in ETs is driven by two consecutive toehold-mediated strand displacement [174] steps. The first step is a bimolecular strand displacement(BSD) [168, 174] process, in which an ET approaches a vacant site and binds by its matching toehold(s). Following the toehold(s) binding, the protecting ends of the *PT* may get displaced in a branch migration [99, 168] process. In the second step, if both arms of the *PT* are released after the bimolecular strand displacement, the PT structure swivels and engages in a unimolecular strand displacement(USD) involving its own two ends. At the end of these two steps, the *(PT)* eventually dissociates completely and the BT has been assembled at the vacant site.

Having discussed the design and Molecular Dynamics study of the deprotection mechanism of the ET in the previous chapters, we present a quantitative modelling of the ET assembly mechanism in this chapter. We first describe the kinetics modelling of the toehold-mediated strand displacement using models developed by Zhang and Winfree [174], and Srinivas et al. [141]. Then we employ the kinetics and biophysics of the toehold-mediated strand displacement process to develop a basic model system for a quantitative study of the ET assembly process at the single component level. In particular, we analyse quantitatively the assembly error prevention in the cooperative assembly process of the ETs and reflect upon the design choices for a theory to practice approach of the mechanism. We also analyse quantitatively the approximate time required for the assembly of an ET. In the end, we compare these findings with the performance of other state-of-the-art error-prevention tile assembly mechanisms.

6.1 Kinetics of the toehold-mediated Strand Displacement Reaction

In Section 2.5.1, we discussed the mechanism of toehold-mediated strand displacement. In this section, we revisit the mechanism and discuss briefly the models that were previously developed by Zhang and Winfree [174], and Srinivas [141] to study the kinetics of the toehold-mediated strand displacement reaction.

The dissociation reaction of double-stranded DNA is a very slow process (discussed in Section 2.2), it may take years to dissociate a double-stranded DNA of 12 nucleotides . However, there is a intuitive way to speed up and control the dissociation rate of double-stranded DNA over a million-fold simply by adjusting the length (between 1-6 nucleotides) of a protruding single-stranded section termed 'toehold' [174]. The dissociation reaction involves toehold-mediated strand displacement ((Figure 6.1)), in which an invading DNA strand displaces a target strand from a double-stranded DNA complex. The strand displacement reaction is driven by the free energy derived from gain in enthalpy due to base-pairing of toehold and gain in configuration entropy due to releasing strand [173].

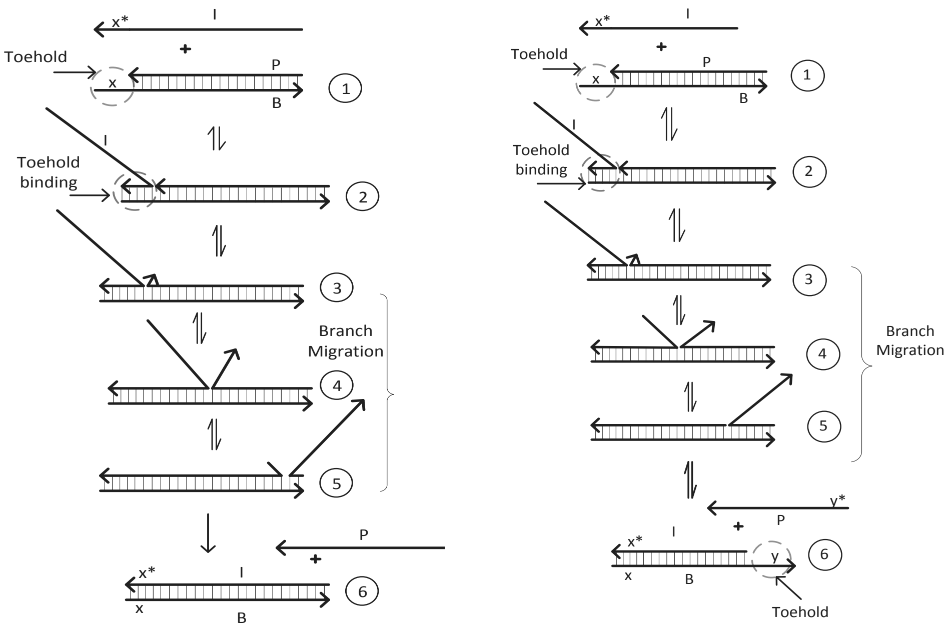
There are two quite similar variants of the strand displacement reaction [173, 174]: 1) Toehold-mediated strand displacement; 2) Toehold exchange. The Toehold-mediated strand displacement refers to the strand displacement reaction where invading DNA strand has equal length (number of nucleotides that are also complementary) as the Base strand (B) of the double-stranded DNA complex (Figure 6.1a). Thus, the final product of the reaction is a double-stranded DNA complex with no single-stranded overhang that can act as a potential toehold to reverse the reaction. The toehold exchange reaction (Figure 6.1b) differs from the formers in that the invader is not completely complementary to the base strand of the double-stranded DNA complex. Therefore in the end of the stand displacement process, the protector strand still remains attached by a few nucleotides with the DNA complex, which may dissociate spontaneously. The spontaneous dissociation of the protector strand would create a new single-stranded overhang in the DNA complex that could serve as a potential toehold for the reverse reaction. Therefore the toehold exchange reaction could be designed as a reversible process. The toehold exchange reaction offers improved kinetic control and creates two reactive DNA molecules which can be utilized in cascading the DNA reactions.

6.1.1 Three-state Phenomenological Model of DNA Strand Displacement

Here we discuss the three-step kinetic model of strand displacement that was developed by Zhang and Winfree [174]. The three steps in the model include: toehold binding by invader DNA strand, branch migration as a single step, and spontaneous dissociation of the incumbent DNA strand (protector strand in Figure 6.1).

$$\textcircled{1} \underset{k_{r(x)}}{\overset{k_{fw}}{\rightleftharpoons}} \textcircled{2} \underset{k_b}{\overset{k_b}{\rightleftharpoons}} \textcircled{3} \underset{k_b}{\overset{k_b}{\rightleftharpoons}} \textcircled{4} \underset{k_b}{\overset{k_b}{\rightleftharpoons}} \textcircled{5} \underset{k_{fw}}{\overset{k_{r(y)}}{\rightleftharpoons}} \textcircled{6} \tag{6.1}$$

The rate constant k_{fw} represents the hybridization rate of toehold binding reaction that



(a) (b)

Figure 6.1: DNA strand displacement mechanisms. (a) Toehold-mediated DNA strand displacement. (b) DNA toehold exchange. Notations used in figure are: Invader DNA strand (*I*); Base strand (*B*) and Protector strand (*P*) in double-stranded DNA complex; Toeholds (*x*, *y*); (*) represents complementary DNA domain; Progress of strand displacement processes is represented by 6 intermediate steps (① → ② → ③ → ④ → ⑤ → ⑥). Step ① represents initial stage, ② is toehold binding step, branch migration is represented by steps (③, ④, ⑤), and ⑥ represents the final step, where protector strand is displaced by the invader strand. Three steps representing the branch migration are just symbolic, in reality the process involves multiple back-and-forth steps, and it has been modelled as a 1D 'random walk' [174, 141]. Observe that the final step in (b) is reversible due to creation of toehold (y), while in (a) process is (almost) irreversible.

occurs between a toehold (x or y) and its complement. In a diffusion-limited DNA hybridization reaction, the rate constant varies within a factor of two for toehold DNA sequences [90], thus can be considered constant ($k_{fw} \approx 3 \times 10^6 M^{-1}\ s^{-1}$). The rate constants $k_{r(x)}$ and $k_{r(y)}$ represent the dissociation rates of DNA strands attached by toehold *x* (step ② in Figure 6.1a) and toehold *y* (step ⑤ in Figure 6.1b). The effective kinetic rate constant for branch migration step(s) is represented by k_b.

The kinetics of strand displacement reaction (6.1) can be simplified to an equivalent bimolecular reversible reaction kinetics between initial step (①) and final step(⑥), as given in (6.2).

$$① \underset{k_{eff(y,x)}}{\overset{k_{eff(x,y)}}{\rightleftharpoons}} ⑥ \tag{6.2}$$

The rate constants $k_{eff(y,x)}$ and $k_{eff(x,y)}$ denote effective rates of the reversible toehold-exchange reaction. An analytically derived expression for the effective rate constant $k_{eff(y,x)}$ is given below in Equation (6.3).

$$k_{eff(y,x)} = \frac{k_{fw}k_{r(y)}k_b}{k_{r(x)}k_{r(y)}+k_b(k_{r(x)}+k_{r(y)})} \tag{6.3}$$

6.1.2 Detailed Biophysical Model of DNA Strand Displacement

The phenomenological model of Zhang and Winfree [174] does not explain much about the process of branch migration; It simply fits an approximate value ($k_b \approx 1s^{-1}$) in the model. Srinivas et al. [141] observed a discrepancy in effective kinetic rates for a 1 nucleotide toehold strand displacement obtained from the phenomenological model and derived analytically. To explain this discrepancy, Srinivas et al. [141] presented a detailed model of branch migration that explicitly includes intermediates, thereby highlighting the important thermodynamic and kinetic features of the process that are not evident from the phenomenological approach. The model is briefly explained below.

The effective rate of toehold-mediated strand displacement can be given by the rate of toehold binding multiplied by the probability that the toehold successfully completes the branch migration once it is bound:

$$k_{eff(0,x)} = r_{fw} \times p_{bm|toe}, \tag{6.4}$$

where $p_{bm|toe}$ is the probability of successfully completing the branch migration once the toehold is bound.

The branch migration process is initiated by displacing the first base of the incumbent strand with a rate k_{first}. The probability of initiating the branch migration rather than falling off is

$$p_{in} = \frac{k_{first}}{k_{first} + k_{r(x)}} \tag{6.5}$$

The branch migration initiation is a slow unimolecular process, where both the incumbent and invader DNA strands are engaged in a localized reaction involving a tough fight of 'breaking and making' base pair. The model assigns an energetic cost, ΔG_{s+p} = 7.3 kcal/mol to the process, where $\Delta G_{s+p} = \Delta G_s + \Delta G_p$. Energy parameters, ΔGs and ΔGp, represent equivalent free energy costs of slowness of branch migration against fraying and penalty for initiating branch migration, respectively. The fitted values to these parameters are: $\Delta Gs = 5.3$ kcal/mol and $\Delta Gp = 2.0$ kcal/mol. The rate of branch migration initiation, k_{first}, is given as:

$$k_{first} = \frac{1}{2} \times k_{uni} \times e^{-\Delta G_{s+p}/RT},$$

where k_{uni} is kinetic rate constant for the unimolecular DNA binding reaction. A typical value of the k_{uni} is ≈ 20 times larger than the DNA hybridization rate constant k_{fw} [141]

Once the branch migration is initiated and the invading strand has displaced the first base of the incumbent strand, it has a probability $1/(b-1)$ of successfully displacing the remaining (b-1) base pairs of the incumbent strand. It is assumed that the branch migration is a 1D random walk, where the invading strand can gain or lose a base pair with equal probability. The probability that the invading strand goes back to being bound just by the toehold is $1 - 1/(b-1)$. It can then again initiate the displacement with probability p_{in}. Therefore, probability $p_{bm|toe}$ can be given as (6.6).

$$p_{bm|toe} = \frac{k_{first}}{k_{first} + k_{r(x)}} \left(\frac{1}{b-1} + \frac{b-2}{b-1} \times p_{bm|toe}\right) \tag{6.6}$$

which gives

$$p_{bm|toe} = \frac{k_{first}}{k_{first} + (b-1)k_{r(x)}} \tag{6.7}$$

Using equations, (6.4) and (6.7), rate of successful displacement is obtained

$$k_{eff(0,x)} = k_{fw} \times \frac{k_{first}}{k_{first} + (b-1)k_{r(x)}} = \frac{k_{fw}}{1 + (b-1)\frac{k_{r(x)}}{k_{first}}} \tag{6.8}$$

6.1.3 Kinetic Discrimination Regime in the Strand Displacement Reaction

One of the observed behaviours of the toehold-mediated strand displacement reactions is the presence of two kinetic regimes [174, 141]: the kinetic discrimination regime and the kinetic saturation regime. The kinetic discrimination regime occurs in the short toehold range (kinetic rates vary by million-fold over 1-6 nucleotides toeholds), while for large toeholds the kinetic rates saturate. The kinetic regimes can be analysed using the models discussed above. The two models of the strand displacement kinetics, presented in (6.3) and (6.8), are compatible, except the energetic cost parameter ΔG_{s+p} used in the detailed biophysical model. Considering the kinetic expression in Equation (6.3), the effective rate of a toehold-mediated strand displacement reaction (toehold length x) can be gives as:

$$k_{eff(0,x)} = \frac{k_{fw}k_{r(0)}k_b}{k_{r(x)}k_{r(0)} + k_b(k_{r(x)} + k_{r(0)})}$$

For short toeholds, the toehold dissociation rate $k_{r(x)}$ is large, and therefore considering $k_{r(x)} > k_b$, the above expression can be simplified as:

$$k_{eff(0,x)} \approx \frac{k_{fw}k_b}{k_{r(x)}} \tag{6.9}$$

However, for long toeholds, the dissociation rate $k_{r(x)}$ would be small due to strong binding in the toehold. Therefore, considering $k_{r(x)} < k_b$ gives the following approximate expression for the kinetic rate.

$$k_{eff(0,x)} \approx \frac{k_{fw}k_{r(0)}k_b}{k_{r(0)}k_b} = k_{fw} \tag{6.10}$$

Similarly, the kinetic model of Equation (6.8) can be approximated into the two kinetic regimes, as described below.

For the short toeholds, we can assume $1 << (b-1)\frac{k_{r(x)}}{k_{first}}$, and Equation (6.8) approximates to

$$k_{eff(0,x)} = r_{fw} \times \frac{k_{first}}{(b-1)k_{r(x)}} \tag{6.11}$$

For long toeholds, considering $1 >> (b-1)\frac{k_{r(x)}}{k_{first}}$, gives

$$k_{eff(0,x)} = k_{fw} \tag{6.12}$$

Observe that the two models give the same kinetic behaviour in the saturation regime (Equations, 6.10 and 6.12). However, the kinetic rates obtained from the two models in the kinetic regime (Equations, 6.9 and 6.11) differ. The two rates reconcile if $k_b = \frac{k_{first}}{(b-1)}$.

To simulate the kinetic rates of toehold-mediated strand displacement reactions using model of Zhang and Winfree [174], we use the following example (Table 6.1), where toehold length is varied from 0-10 nucleotides. DNA strands, B, P, and I refer to the Base Strand, Protector Strand, and Invader strand, respectively, as described in Figure 6.1. DNA sequences of different strands are listed in the second column of the table. Each of the B and I strands are 20 nucleotides long. Length of the protector strand varies from 20-10 nucleotides($P_0 - P_{10}$) so as to create toeholds of 0-10 nucleotides in the double-stranded DNA complex using strands B and P_x. The DNA sequences were designed using the *NUPACK* software. The designed DNA sequences produce only double-stranded DNA complexes between strands B, P_x,and I, and do not form any unintended DNA structures. The third and fourth columns in the table represent

Table 6.1

	DNA sequence (5'–>3')	Toehold binding energy (kcal/mol)	Toehold length (x)
P_0	CGCTCCGCCGCGTCAGAAAC	+2.50	0
P_1	CGCTCCGCCGCGTCAGAAA	+1.35	1
P_2	CGCTCCGCCGCGTCAGAA	-0.13	2
P_3	CGCTCCGCCGCGTCAGA	-1.30	3
P_4	CGCTCCGCCGCGTCAG	-2.50	4
P_5	CGCTCCGCCGCGTCA	-4.82	5
P_6	CGCTCCGCCGCGTC	-5.70	6
P_7	CGCTCCGCCGCGT	-8.15	7
P_8	CGCTCCGCCGCG	-7.07	8
P_9	CGCTCCGCCGC	-10.16	9
P_{10}	CGCTCCGCCG	-13.07	10
B	GTTTCTGACGCGGCGGAGCG		
I	CGCTCCGCCGCGTCAGAAAC		

toehold binding free energy and toehold length (x), receptively. The toehold binding energy was calculated using *NUPACK*, as described below:

Toehold Binding Free Energy $\Delta G^{\circ}(x) = \Delta G^{\circ}(BP_xI)$ - $\Delta G^{\circ}(BP_x)$, where BP_xI and BP_x represent the DNA complexes in the toehold bound state(②, Figure 6.1) and the toehold free state (states (①, Figure 6.1), respectively. Using the kinetic model of Zhang and Winfree (Equation 6.3), a simulation of the kinetic rates is shown in Figure 6.2. The x-axis shows both the toehold length (x nucleotides) and corresponding toehold binding free energy (kcal/mol), and the y-axis shows the simulated kinetic rates. As can be seen from the figure, the maximum kinetic rate occurs for x=7 ($k_{eff}(0,7) \approx 2.50 \times 10^6$) and then it saturates for longer toeholds. In the region, $0 \leq x \leq 6$, the kinetic rate varies by a million-fold. In this region therefore strong kinetic discrimination can be engineered using rationally designed toeholds. In the following section, we use this insight to discuss the kinetic discrimination in the ET docking during its assembly.

6.2 Mechanism of Enveloped Tile Assembly

We revisit the steps (Figure 4.19) involved in the self-assembly of ET. The self-assembly starts as ETs are exposed to an initiating seed structure, and the two-dimensional tile lattice grows as ETs assemble to the vacant sites created by each time a tile joins the lattice. A correctly matching *ET* is kinetically favoured over a mismatching *ET* during the docking stage. A successful *ET* docking, which involves a cooperative toehold-mediated displacement of its PT from the engaged inputs side, initiates the complete deprotection through an intramolecular engagement of the PT's just released 'inputs' with 'outputs'. The main steps of the ET self-assembly process are listed below, and illustrated in

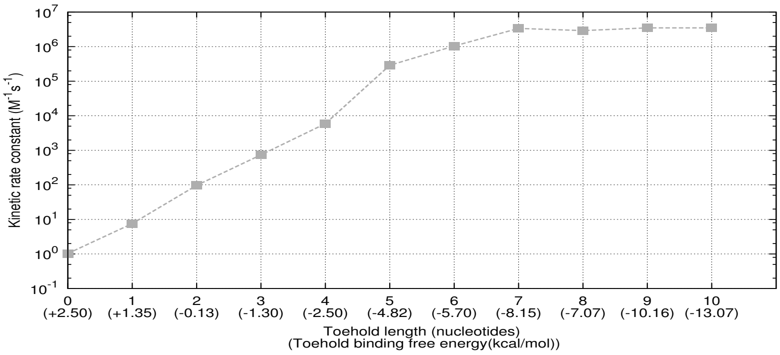


Figure 6.2: Kinetic rate regimes in the strand displacement reaction.

Figure 6.3 using schematics.

- *ET 'Docking'*: An ET approaches the available vacant site in the growing lattice of tile, and binds by its toehold(s). If the toehold(s) engagement is thermodynamically strong, it would initiate the displacement of PT 'inputs'
- *Branch Migration vs. toehold fraying [99, 141]*: In a competing process, an ET attached weakly by its toehold(s) may fray away from the engaged site.
- *Completion of Branch Migration*: If the branch migration is initiated in the PT 'inputs', it is more likely to succeed in releasing the PT from its input side.
- *PT folding*: After releasing from input side, the *PT* swivels around its flexible central section and initiates intramolecular engagement between its 'inputs' and 'outputs'.
- *BM initiation on the output side of ET*: Engagement between the two ends of the *PT* starts a unimolecular strand displacement, in which first branch migration initiates in the 'outputs' of the PT that leads the PT partially attached with its *BT*.
- *Spontaneous dissociation of PT* : In a thermally driven dissociation process the PT spontaneously dissociates, deprotecting the outputs of the engaged *BT* for subsequent assembly by incoming ETs.

6.2.1 Kinetic Discrimination in Enveloped Tile Docking

Docking step in the ET assembly is crucial for the error-prevention. We first discuss the biophysics of ET docking, and quantitatively determine the Kinetic Discrimination Ratio (KDR) between matching and mismatching ETs using sequence specific binding free energies of randomly chosen combinations of toehold DNA sequences.

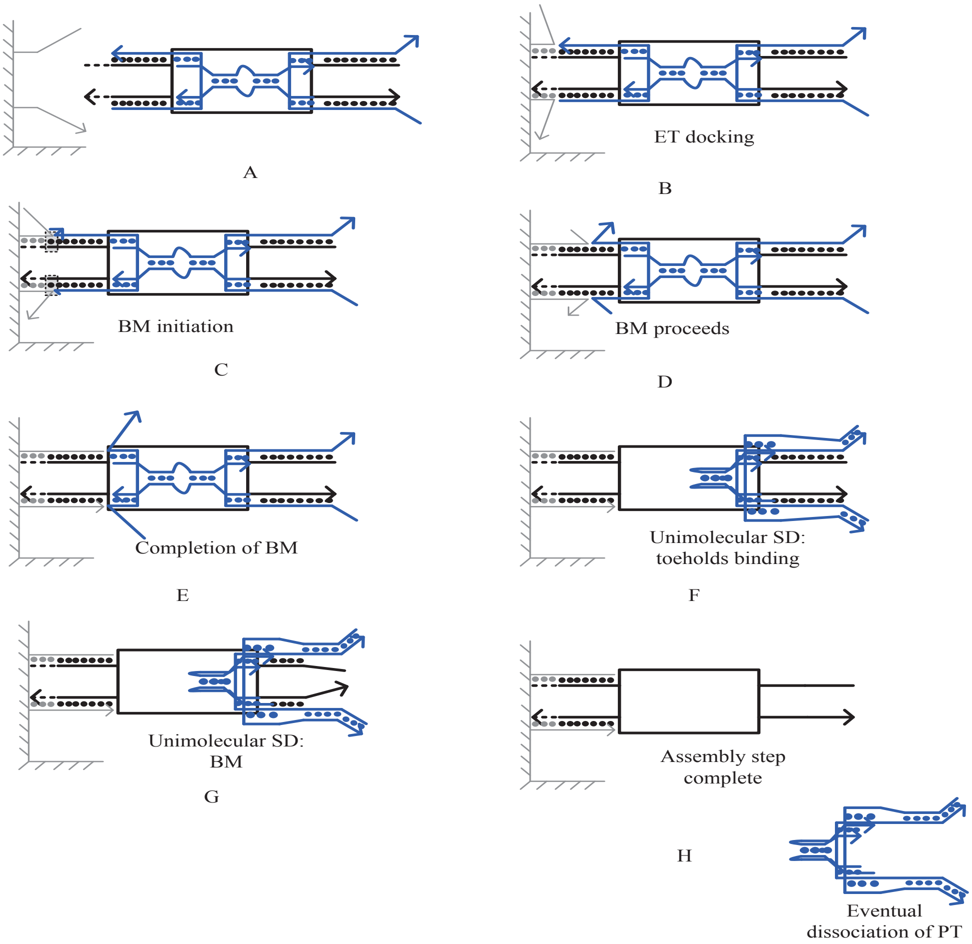


Figure 6.3: Intermediate steps in assembly process of ET (A-H): (A) ET approaches a vacant site of the growing tile-lattice by orienting its input toeholds towards free sticky-ends of the vacant site. (B) ET 'Docking'. (C) Branch Migration initiation vs. toehold fraying. (D) A successful branch migration in the fully engaged inputs of the PT releases it from the input side (E). (F) A swivelling of *PT* initiates intramolecular engagement of the PT 'inputs' with 'outputs', leading to unimolecular strand displacement(G). (H) The *PT* spontaneously dissociates from the BT, deprotecting its outputs for further assembly.

ET docking is illustrated in Figure 6.4. Observe that the two toeholds of the engaging ET are tightly held at a spatial separation of $\approx 4\ nm$ by the planar DNA structure of the BT. Lets analyse the possible biophysics of such localized binding of the two toeholds. There could be three possibilities for the binding of toeholds: 1) binding is cooperative; 2) binding is anticooperative; 3) binding is independent.

In a cooperative binding, a molecular binding thermodynamically favours the binding of subsequent molecules. For example, a haemoglobin molecule can attach four oxygen molecules. But as first oxygen molecule attaches, the affinity for binding of subsequent oxygen molecules increases [7], and the rise in the binding affinity it attributed to the structural changes in the haemoglobin protein. This phenomenon is known as cooperative binding. Similarly, in an anticooperative binding the molecular binding makes it thermodynamically less favourable for the binding of subsequent molecules. Finally, in an independent binding, a molecular binding has no influence over the binding of other molecules. Williamson [153] illustrates a thermodynamics perspective of these three types of binding mechanisms.

For an independent binding of ET toeholds, the total binding free energy of the ET would be two times the free energy of a single toehold binding, i.e., $\Delta G_{sum} = 2\Delta G_t^{\circ}$, where ΔG_t is the binding energy of single toehold . Due to localized binding of the two toeholds, it may not be an independent binding. Therefore, it could be either an anticooperative binding or cooperative binding. Imagine when one of the ET toeholds binds, the second toehold should also be confined to the binding site. The binding of the first toehold would be a diffusion-limited process, that is, the tile has to first approach the binding site, align its toehold to comply with the topology of the binding site, and form base-pairs. The free energy of the first toehold binding can be parameterised by the nearest neighbor-model of SantaLucia [115], which includes a penalty for the binding initiation ($\Delta S_{int}^{\circ} \approx -6$ cal/mol/K in 1 M NaCl at 37 $^{\circ}$C.

Binding of the second toehold (Figure 6.4B) would be a unimolecular process, where the toehold is confined to a very small volume and has limited freedom to undergo an independent translation and rotation. Therefore, the entropy loss would be less than the entropy loss in the binding of the first toehold. How much this difference would exactly be is a matter of experimental study. However, a recent study [69] shows that binding between sticky ends of two DX molecular tiles is less than the binding free energy of two similar but flexible DNA structures. This difference in the free energy, $\Delta G^{\circ} \approx -2.7$ kcal/mol (at 25 $^{\circ}$C), reflects cooperative nature of the binding.

A similar free energy difference can be considered in the toeholds binding of the ET. To include the effect of cooperativity in the toeholds binding free energies of ET, we introduce an entropy term, ΔS_c(cal/mol/K) in the overall free energy of toehold binding. A fitted value to the parameter ΔS_c is 15 cal/mol/K, which gives a cooperative binding energy contribution ($\Delta\Delta G_c^{\circ} \approx -2.7$ kcal/mol) at 25 $^{\circ}$C.

Kinetic Discrimination Ratio: To analytically determine the kinetic discrimination between a matching tile and mismatching tile during the ET docking, we calculated sequence specific binding free energies of different toehold combinations and calculated the kinetic rates of displacement of the PT inputs. The Kinetic Discrimination Ratio (KDR) for different toehold combinations (Appendix A.1) is shown in Figure 6.5. We considered toeholds of 1 nucleotides, 2 nucleotides, 3 nucleotides, and 4 nucleotides. To explain the binding free energy calculation, consider the case with 3 nucleotide case. There would be $\frac{64!}{62!2!}$ possible combinations of 3 nucleotide sequences that can

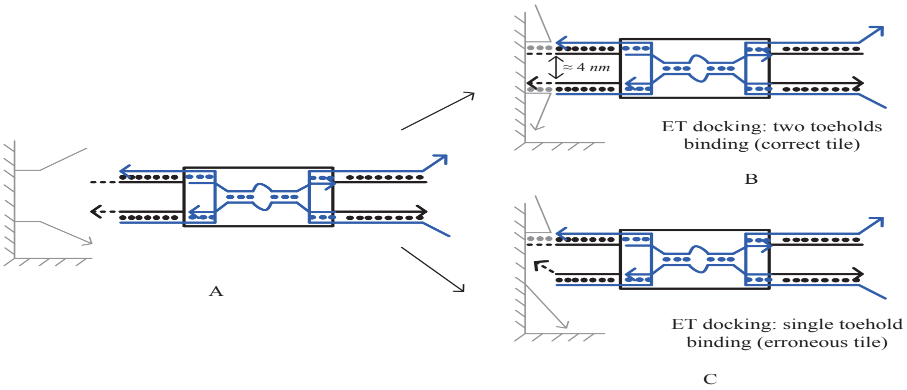


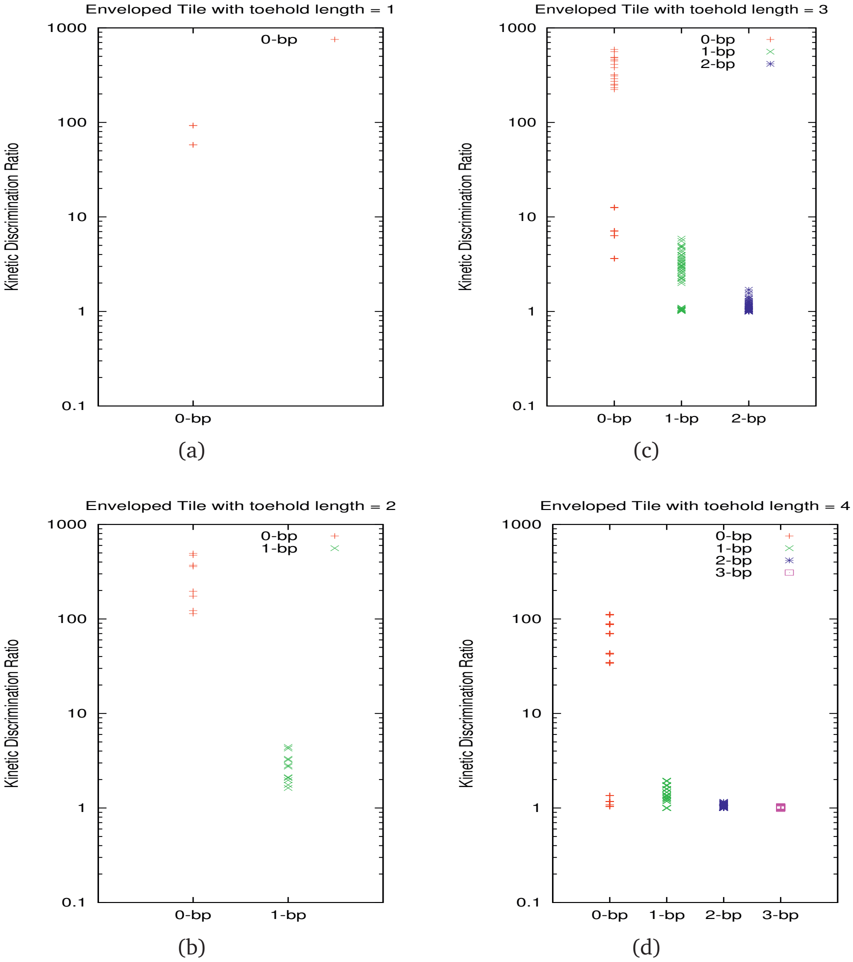
Figure 6.4: Docking of ET. (A) An ET approaching a vacant site. (B) Docking stage of ET having both toeholds correctly matching with the vacant site's sticky ends. (C) Docking stage of ET having only one toehold correctly matching with the sticky ends of vacant site.

potentially be chosen as ET toeholds. Instead of calculating the binding free energy for all these combinations, we consider the sets of toehold sequences that would give highest and least kinetic discriminations. We choose only two types of 3 nucleotides sequences: with 0% GC content and 100% GC content. The sequences with 0% GC content, which includes only A/T nucleotides, would give the weakest toehold binding, while the sequences with 100% GC content would give the strongest toehold binding. Thus, the two extreme cases would produce the smallest and the largest kinetic discriminations possible in the 3 nucleotide toeholds case.

Matching toehold free energies: Consider two DNA sequences each having 3 nucleotides (5'-TAT-3' and 3'-AAA-5') that represent the two toeholds of an ET. To calculate the total toehold binding free energy, we first calculate $\Delta G^{\circ}\binom{5'{-}TAT{-}3'}{3'{-}ATA{-}5'}$ = -1.88 kcal/mol, $\Delta G^{\circ}\binom{3'{-}AAA{-}5'}{5'{-}TTT{-}3'}$= -2.46 kcal/mol, and summed up $\Delta G^{\circ}\binom{5'{-}TAT{-}3'}{3'{-}ATA{-}5'}$ + $\Delta G^{\circ}\binom{3'{-}AAA{-}5'}{5'{-}TTT{-}3'}$ + $\Delta\Delta G^{\circ}_{c}$ (= -2.7 kcal/mol). Hence, the total toehold binding free energy, where both toeholds are matching, would be ≈ 7.04 kcal/mol. These free energies were calculated using NUPACK at the salt concentrations $Na^{+} = 0.05$ M, $Mg^{++} = 0.0115$ M and temperature 25 °C. The calculated free energies include 'all' types of stacking and dangling energies included in the NUPACK model.

Mismatching toehold free energies: Consider the case when only one of the toeholds (5'-TAT-3' or 3'-AAA-5') matches. Lets assume the toehold 5'-TAT-3' is the matching toehold, then the binding free energy contribution from single toehold binding would be $\Delta G^{\circ}_{t}\binom{5'{-}TAT{-}3'}{3'{-}ATA{-}5'}$ = -1.88 kcal/mol. For the mismatching toehold, we consider three cases: 1) non of its nucleotide has a complementary base in the binding site (0-bp); 1) one of its nucleotides has a complementary base in the binding site (1-bp); 1) two of its nucleotides has a complementary base in the binding site (2-bp). We calculate the binding free energies for all three cases of mismatched toehold binding. Case (0-bp): $\Delta G^{\circ}\binom{3'-AAA-5'}{5'--3'}$ = 0 kcal/mol. Case (1-bp): $\Delta G^{\circ}\binom{3'-AAA-5'}{5'-T-3'}$ = -0.12 kcal/mol. Case (2-bp): $\Delta G^{\circ}\binom{3'-AAA-5'}{5'-TT-3'}$ = -1.29 kcal/mol. Then we calculate the total binding free energies for these cases. For the two cases (1-bp and 2-bp) we consider the cooperative binging free energy contribution, but for the the (0-bp) case this contribution is ignored, because there is no base-paring in the mismatched toehold.

To calculate the KDR for each of the above described cases, we simulated the kinetic rates of toeholds-mediated displacements using kinetic model of Zhang and Winfree (Equation 6.3). The KDRs are shown in Figure 6.5c.



(a)

(c)

(b)

(d)

Figure 6.5: Kinetic discrimination ratios between matched and mismatched ETs during docking stage. (a) 1 nucleotide toeholds. (b) 2 nucleotides toeholds. (c) 3 nucleotides toeholds. (d) 4 nucleotides toeholds.

The maximum and minimum KDRs are $\approx$ 591 and $\approx$ 1, which occur in the (0-bp) case and (2-bp) case, respectively. Similarly, we calculated KDRs for the toeholds of 4 nucleotides (Figure 6.5d), 2 nucleotides (Figure 6.5b), and 1 nucleotide (Figure 6.5a). The maximum KDR for the 2 nucleotides toeholds is ≈ 493, while minimum is the same as in the 3 nucleotides toeholds case. For 4 nucleotides toeholds the maximum KDR is $\approx$ 111. It can be observed from these kinetic discrimination ratios that ETs should be designed with the toehold sequences of 3 or 2 nucleotides and the there should not be possible base-paring nucleotides in the sticky ends of tiles so as to strongly favor the ET docking of matching tiles against mismatching tiles.

6.3 Kinetic Modelling

Using biophysics and kinetic parameters, we develop a kinetic model of ET assembly. The forward rate of toehold binding is equivalent to the hybridization between two single strands; it is expressed as $r_{ft} = k_f[c]$, where c is the concentration of tile monomers and k_f is the kinetic rate constant of hybridization between two single DNA strands. Toehold dissociation is thermodynamically governed, having dissociation rate $r_{d,t} = k_f e^{\frac{\Delta G_t^\circ}{RT}}$, where ΔG_t° is free energy released by single toehold binding and can be be given by $\Delta G_t^\circ = \Delta H_t^\circ - T\Delta S_t^\circ$ (T is temperature in K). ΔH_t° and ΔS_t° are values of standard enthalpy and standard entropy, respectively. Using Nearest-Neighbor (NN) base-pairing energy parameters from SantaLucia et al. [115], values of ΔH_t° and ΔS_t° for a toehold of length l can be given by: $\Delta H_t^\circ \approx -8l$ kcal/mol, and $\Delta S_t^\circ \approx -22l$ cal/mol/K. Therefore, free-energy of a single toehold binding can be given by $\Delta G_t^\circ = -8000l - T(-22l + \Delta S_{int})$, where ΔS_{int} is the conformation entropy loss due to aligning two DNA strands in opposite orientations. A rough estimate of ΔS_{int}, used in other similar studies [154], is ≈ -6 cal/mol/K. Free energy of toehold binding considering the cooperative toeholds binding can be given by $\Delta G_{tc}^\circ = -8000l - T(-22l + \Delta S_{int} + \Delta S_c)$.

6.4 Error Prevention in the Enveloped Tile Self-assembly

In the self-assembly of ETs, assembly error suppression could happen in the following two stages: Error Prevention due to Kinetic Discrimination of Mismatching ET, and Error Prevention due to a Competitive Disengagement of Mismatched ET.

Error Prevention due to Kinetic Discrimination of Mismatching ET Given the rate of toehold binding r_{ft}, the toehold-mediated displacement rates of PT inputs during the ET docking can be derived using Equation (6.13). The (k_1) is displacement rate for the mismatching ET, and (k_2) is for the matching ET.

$$k_1 = r_{ft} \times p_{bm|toe} \tag{6.13}$$

$$k_2 = r_{ft} \times p_{bm|2toe} \tag{6.14}$$

Therefore, the probability (p_p) that a mismatched ET would pass the kinetic discrimination state is

$$p_p = \frac{k_1}{k_1 + k_2} \tag{6.15}$$

6.4.1 Error Prevention due to a Competitive Disengagement of Mismatched ET

If branch migration succeeds in the engaged input of the ET, as shown by state transition $A \rightleftharpoons B$ in Figure 6.6, there could be two further state transitions ($B \rightleftharpoons A$, or $B \rightleftharpoons C$) from state 'B' as described below-

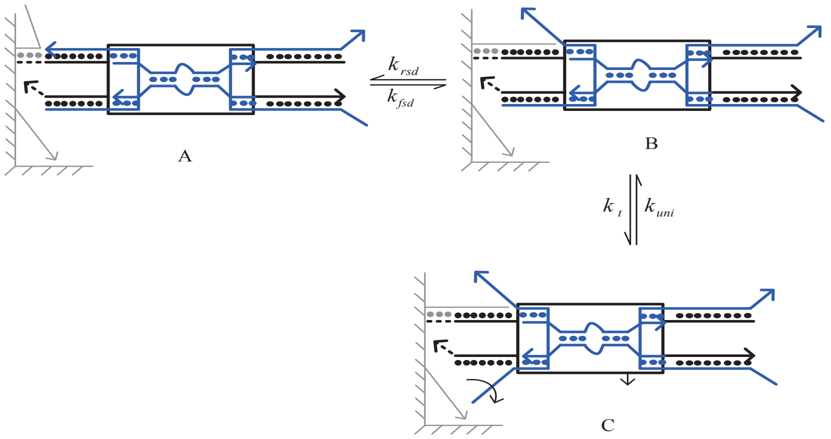


Figure 6.6: Kinetics driven states of a locked in mismatched ET: (A) *ET* binds with its single matching toehold, (B) A successful branch migration displaces the engaged PT input, and (C) the other PT input, with no conjugate, disengages spontaneously, releasing the PT from the engaged side.

Case-I Starting in state B, a spontaneous thermal process may eventually create a single base toehold by fraying one base of the newly displaced arm, and the arm undergoes a subsequent displacement leaving the mismatched tile bound by its single toehold (l bases) that may eventually dissociate and thereby clear the site for another tile to approach. Kinetics rates of transitions from A $\rightleftharpoons$ B, and from B $\rightleftharpoons$ A are k_{fsd} and k_{rsd}, respectively. It should be observed that these two kinetic rate constants are related, i.e., k_{fsd} is a strand displacement rate for a toehold length l while k_{rsd} is for a zero toehold length. Therefore, using the strand displacement kinetics model described earlier, the rake constant, k_{rsd}, is

$$k_{rsd} = \frac{k_{uni}\; e^{\frac{-\Delta G_{s+p}}{RT}}}{b} \tag{6.16}$$

Considering a subsequent dissociation of the toehold bound tile with a kinetics rate constant, $k_{dt} = k_{bi}\; e^{\frac{-l \times G_{bp}}{RT}}$, an equivalent kinetic rate constant of a reverse SD process with subsequent toehold dissociation can be given by

$$k_{eq} = \frac{k_{rsd} k_{dt}}{k_{rsd} + k_{dt}} \tag{6.17}$$

Case-II Before the aforementioned transition ($B \rightarrow A$) occurs, the other protected input of the PT spontaneously dissociates , and therefore, a subsequent intramolecular unimolecular strand displacement may ultimately deprotect the tile, fixing an erroneous tile.

Kinetic rates, k_t and k_{uni}, between B and C are related as

$$k_t = k_{uni} \times e^{-bG_{bp}} \tag{6.18}$$

The probability that a mismatched tile that passed the kinetic discrimination stage, would still remain engaged is

$$p_c = 1 - \frac{k_{eq}}{k_{eq} + k_t} \tag{6.19}$$

Therefore, an erroneous tile would ultimately assemble if it passes both the stages: the kinetic discrimination stage and the competitive disengagement stage. The probability (p_e) of erroneous ET assembly can be given by combining the two probabilities from Equations (6.15) and (6.19).

$$p_e = p_p \times p_c \tag{6.20}$$

6.5 Assembly Time of Enveloped Tiles

In this section, we model the assembly time of the enveloped tile using the processes discussed in Section 6.2.

Kinetic Discrimination Delay: We can intuitively analyse time spent in the 'docking' phase of an ET. Given $m + 1$ types of tiles participating in the assembly process, there is only one type of tile that is correct for a particular lattice site, while the rest are mismatched tiles. But all types of tiles are free to dock at an available lattice site. Therefore, time elapsed in the docking process at a vacant site will be the sum of the time delay (T_d) due to mismatched tiles and the time required for successful *docking* of a correct tile.

$$T_d = p_m \times (p_{BM-fail} \times (T_1 + T_2) + T_3 + T_4 \times p_c) \tag{6.21}$$

$$T_{docking} = T_d + T_c \tag{6.22}$$

where different terms are as follows-

- Probability of mismatched tiles approaching the vacant site → p_m (see (6.23))
- Probability that BM fails after successful toehold binding → $P_{BM-fail} = 1 - p_{bm/toe}$ (see Section 6.4 for the details of the term, $p_{bm/toe}$)
- Probability of correcting the erroneous binding of a mismatched tile by reversing the strand displacement → p_c (see, (6.24))
- Time involved in docking of mismatched tiles → T_1 (see,(6.25))
- Time taken by a mismatched tile to get dissociated by breaking its toehold binding → T_2 (see (6.26))
- Time taken by a mismatched tile to complete the BM in its engaged arm after toehold binding → T_3 (see (6.27))
- Time required for reversing the strand displacement process that has just completed its BM → T_4 (see (6.28))
- Time required for releasing the PT from its input ends when a correct tile approaches a vacant site → T_c (see (6.29))

$$p_m \approx me^{-\frac{\Delta G}{RT}} \tag{6.23}$$

where $\Delta G = 2\Delta G^{\circ}_{tc} - \Delta G^{\circ}_{t}$ and m is the number of mismatched type of tiles for a vacant site.

$$p_c = \frac{k_{eq}}{k_{eq} + k_t} \tag{6.24}$$

An expression of p_c is given in in Section 6.4.

$$T_1 \approx \frac{1}{k_f e^{-G_{mc}}} \tag{6.25}$$

$$T_2 \approx \frac{1}{k_{bi} e^{-l \times G_{bp}}} \tag{6.26}$$

$$T_3 \approx \frac{1}{r_{itb} \times p_{bm|toe}} \tag{6.27}$$

$$T_4 \approx \frac{1}{k_{eq}} \tag{6.28}$$

$$T_c \approx \frac{1}{r_{opinPT}} \tag{6.29}$$

Branch Migration Initiation: In the process of ET assembly, after an *ET* is successfully attached by its toehold(s), the subsequent complementary sections of *PT* initiates branch migration to start strand displacement. The kinetic rate of *BM* initiation is

$$r_{bmi} = \frac{k_{uni} \times e^{-\frac{\Delta G_{s+p}}{RT}}}{2} \tag{6.30}$$

where $k_{uni} = 1.8 \times 10^7$ is the rate of unimolecular reaction involving invader DNA strand and protector-base complex.

Branch Migration Completion: After toehold binding followed by a successful initiation of *BM*, an *ET* having protected inputs by b_i bases has a probability $1/(b_i - 1)$ of successfully completing the displacement, assuming that the branch migration is a random walk where the invading strand can gain or lose a base pair with equal probability. When the invading strand has displaced one base, the probability that the invading strand goes back to being bound just by the toehold is $1 - 1/(b_i - 1)$. It can then again initiate displacement with probability $p_{in} = \frac{r_{bmi}}{r_{bmi} + r_{d,2t}}$. We can thus approximate the probability, $p_{bm|2t}$, that an ET, just docked by its toeholds, will successfully complete the *BM* leading to opening up of its *PT* from input side:

$$p_{bmi|2t} = \frac{r_{bmi}}{r_{bmi} + r_{d,2t}} \left(\frac{1}{b_i - 1} + \frac{b_i - 2}{b_i - 1} \times p_{bm|2t}\right) \tag{6.31}$$

which gives:

$$p_{bmi|2t} = \frac{r_{bmi}}{r_{bmi} + (b_i - 1) r_{d,2t}} \tag{6.32}$$

Combining the kinetics from these three steps (Equations: 6.30, 6.31, 6.32) i.e., tile docking, BM initiation, and BM completion, an equivalent kinetic rate (r_{opinPT}) of opening up of PT from its input ends is given as

$$r_{opinPT} = r_{ft} \times p_{bmi|2t} \tag{6.33}$$

Folding of PT structure: After PT is released from its input side, its free input arms discover their complementary domains (output toeholds) as the PT swivels around its flexible central section. Other ends of the PT remain attached with its BT during the swiveling, thus the PT can not float away. In such localized (unimolecular) case, toehold binding will be faster than the rate of toehold binding in bulk reactions that are diffusion limited. Here in this section, we derive the kinetic rate for such unimolecular toehold binding using the method described in Genot et al. [52].

Following previous analysis, we approximate the rate (k_t) of toehold binding in localized unimolecular reaction conditions as

$$k_t = k_{bi} \times c_e \tag{6.34}$$

where k_{bi} (in /M/s) is the kinetic rate constant of a bimolecular DNA binding reaction under bulk conditions, and c_e is the effective concentration of localized toeholds.

The PT folding involves intramolecular engagement between its inputs and outputs. Let us assume that a volume of lateral size a determines the region where the released ends of the PT encounter the output ends. Each of the two binding domains (the two ends of the PT) lie at the opposite ends of the *PT* structure having end-to-end distance ≈ 30 nm, excluding the flexible central section consisting of ss-DNA. *PT* structure consists of ds-DNA (except the central section), and therefore, under the Worm-like Chain (WLC) model [104], the PT can be considered as a hinged structure having two stiff arms, each having length 10 nm. Assuming that the PT structure swivels around a flexible central section, the probability $p(a)$ of engaging the two ends of PT inside the reaction volume is $p(a) = \frac{a^3}{2\pi l^2 a}$, and therefore, effective local concentration, c_e, in Molar (M) unit can be given as

$$c_e = p(a) \times 10^{-3} \times N_a^{-1} \times a^{-3} \tag{6.35}$$

where N_a is Avogadro Number, and a is the dimension of the cube representing reaction volume within which the two ends of PT collide for toehold binding. For *PT* structure, radius (R) of hemispherical volume that its freed end can explore is ≈ 15 nm, and the dimensions of the reaction volume within which the two ends collide for toehold binding is ≈ 2.1 nm. Therefore, using Equations: (6.34) and (6.35), an approximate kinetic rate of the intramolecular toehold engagement is $k_t \approx 3000$ /s.

BM initiation on the output sides of PT: After successful toehold binding involving the two ends of *PT*, *BM* is initiated to advance the strand displacement in the protected output arms of *PT*. Kinetics of the BM initiation here is the same as described earlier in Section 6.3. Therefore, the rate of BM initiation, r_{bmo}, is same as r_{bmi} given in

Equation 6.30.

BM Completion on the output sides of PT: PT binds with *BT* using (b_o) bases in each of its two output arms. However, out of b_o bases, only $(b_o - 4)$ bases on each arm can be displaced in the *BM* here. After unimolecular toehold binding and initiation of branch migration, *BM* proceeds further to displace $(b_o - 4)$ bases in each input. The rate of *BM* completion on the output side of *PT* is given as

$$r_{bmo} = k_t \times p_{bmo|2t} \tag{6.36}$$

where $p_{bmo|2t} = \frac{r_{bmo}}{r_{bmo}+(bo-5)r_{d,2t}}$.

Spontaneous Dissociation of PT: After completion of BM, PT is still attached with the BT by 4 bases in each of its output arms. Caused by thermal fluctuations, PT may eventually dissociate completely from the BT. The rate of such thermally driven dissociation of PT at an temperature of $25°\mathrm{C}$, which is still bound by 8 bases together in both arms, can be given by extrapolating dissociation rate reported by Morrison et al. [90].

$$r_{dPT} \approx 0.2 \ sec^{-1} \tag{6.37}$$

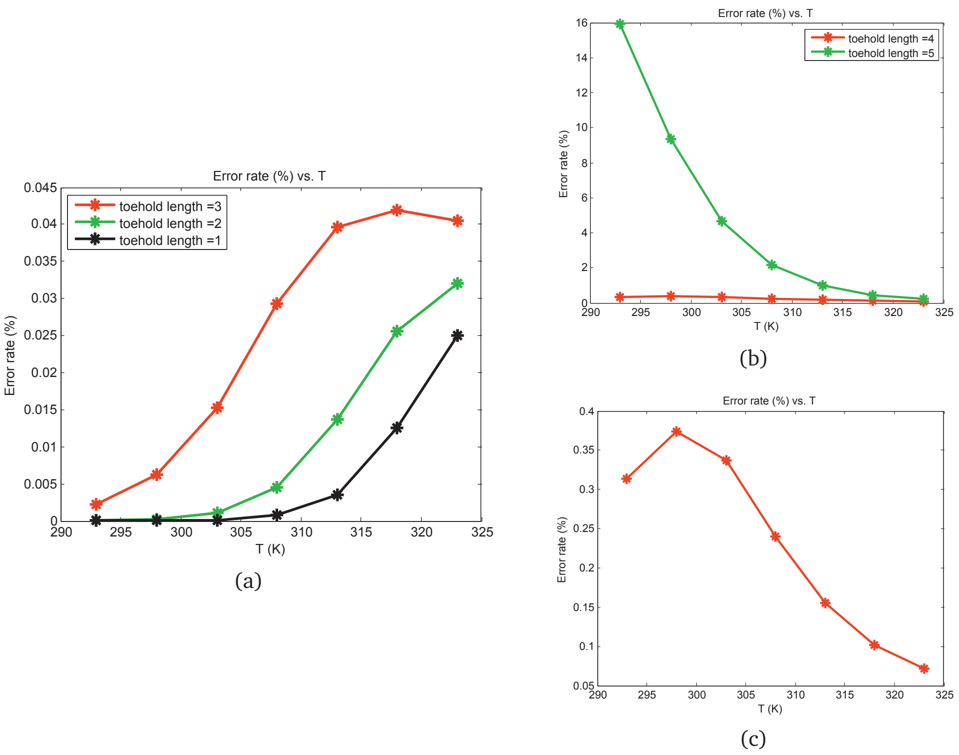
6.6 Simulations

6.6.1 Assembly Errors

Assembly errors are simulated using the kinetic modelling described in Section 6.4. Parameters used in these simulations are: concentration, $G_{mc} = 18$, where $(c = e^{-G_{mc}})$, and a free energy contribution due to co-operative binding of toeholds, $\Delta S_c = 15$ cal/mol/K.

Figure 6.7 shows assembly errors vs. assembly temperature (T) for different toehold lengths of the ET. For toehold length ≤ 3 bases, as shown in Figure 6.7a, error rate is as low as 0.001% for a low temperature range (e.g., T=298 K), but it increases with increase in temperature. For toehold lengths of 4 and 5 bases, error rate is higher, as shown in Figure 6.7b,c, but drops with increase in temperature. Therefore, for lower error rates (e.g., $\leq$ 0.005%), ETs should be designed with shorter toehold (e.g. ≤ 3 bases), and assembly can be conducted at room temperature.

Increase and decrease in the error rates for shorter and longer toeholds, as shown in Figure 6.7a and Figure 6.7b,c, respectively, is attributed to the two-step error-prevention mechanism described in Section 6.4. Due to a strong kinetic discrimination in tiles with shorter toeholds, the first step in the error prevention dominates. Further, a lower temperature would mean that the second step in the error prevention gives significant contribution. Thus, as shown in Figure 6.7a, errors are substantially suppressed at lower temperature. For higher temperature, however, the errors evade the kinetic discrimination and are not corrected effectively, thus error rate increases. In contrast to the shorter toehold regime, tiles with longer toeholds are weakly discriminated in the kinetic discrimination stage. Hence, a relatively higher error rate appears in Figure 6.7b and Figure 6.7c.



(a)

(b)

(c)

Figure 6.7: Error rate vs. assembly temperature (T in K). (a) Toehold lengths = 3,2,1 (bases). (b) Toehold lengths = 4, 5 (bases). (c) Toehold length = 4 bases (an enlarged view).

The impact of protected length *(b)* of ET inputs over error rate is shown in Figure 6.8. At room temperature (T= 298 K), the error rate falls swiftly with increase in protected length *(b)*. Error correction, described in Section 6.4, depends on temperature and protected length (b). For lower temperature, the protected length of arms is kinetically less favorable to dissociate, which gives more time for reversing strand displacement on the other arm. Therefore, increasing the protected length (b), while keeping assembly temperature low, reduces assembly errors.

Another important parameter used in these analyses is the free energy contribution due to cooperative binding of toeholds during the docking stage of ETs. As described in Section 6.3, ΔS_{sc} accounts for additional stability due to localised toehold binding. Although the exact nature and amount of this free energy contribution is a matter of

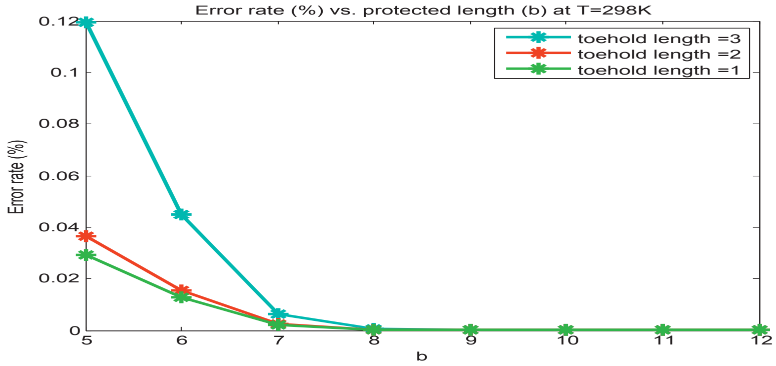


Figure 6.8: Error rate vs. protected length (b) of inputs in ETs (toehold lengths: 1,2 and 3 bases).

experimental investigation, we consider $\Delta S_{sc} = 15$ cal/mol/K for these studies. As shown in Figure 6.9, the error rate drops swiftly as ΔS_{sc} is increased and saturates for $\Delta S_{sc} \geq 15$ cal/mol/K.

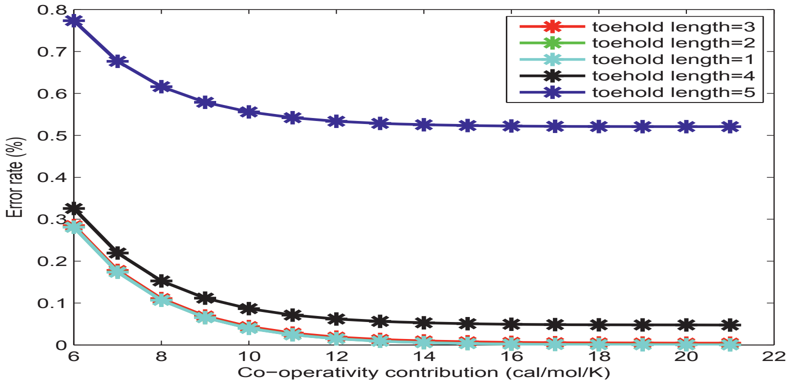


Figure 6.9: Error rate vs. energetic gain due co-operative toehold binding.

6.6.2 Assembly Time

Using the kinetic model described in Section 6.5, we approximate the assembly time of the ET. The first step in the ET assembly, the tile 'docking', is a diffusion-limited process, as shown in Figure 6.10: lower concentrations, $G_{mc} \geq 15$, ([monomer] = $e^{-G_{mc}}$) result in significant delay. Furthermore, the tile 'docking' time also depends on the toehold length of the ET, as shown in Figure 6.11. ETs with toeholds longer than 2 nucleotides have similar tile 'docking' times.

The PT folding reaction is a *USD* process, where its two ends engage in an

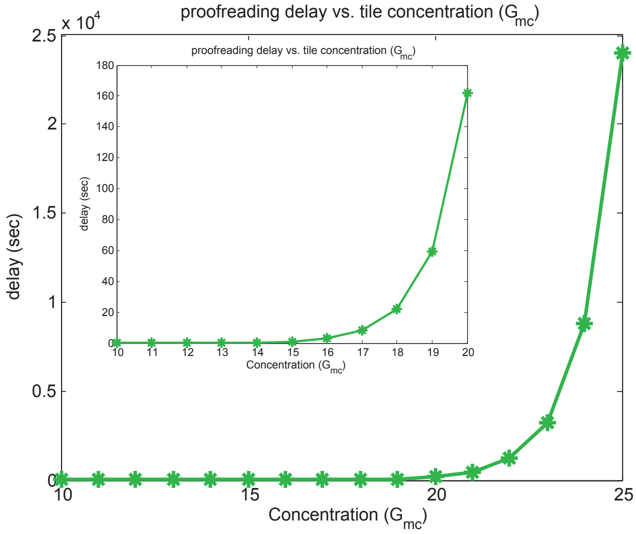


Figure 6.10: ET 'docking' time vs. tile monomer concentration (G_{mc}). An enlarged (inset) view of the plot in the lower G_{mc} range.

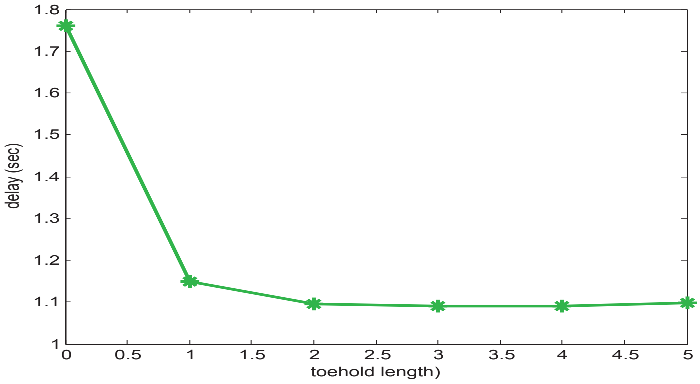


Figure 6.11: ET 'docking' time for different toehold lengths

intramolecular toehold-mediated displacement reaction. The kinetic rate of *USD* depends on the length of the available toeholds on the *ET* 'outputs', as shown in Figure 6.12. It also has greater influence of temperature for the shorter toeholds. Overall, USD is very fast in comparison to the *BSD* process described earlier.

Time required for the assembly of an ET can be estimated using these derived kinetic rates for different steps. Timing of ET assembly is limited by two steps: delay in the ET 'docking', and spontaneous dissociation of the *PT*. Due to localized toehold binding during the intramolecular engagement, the kinetic rate of *USD* is high in comparison to the other two limiting cases.

At a temperature T= 298 K and tile concentration $G_{mc} = 15$, an ET with 3 nucleotides toeholds on both sides, has kinetic rates as shown in Table 6.2

Table 6.2: Kinetic rates of different steps in the assembly of an ET

k_1	k_2
$\approx 2600\,\mathrm{sec}^{-1}$	$\approx 0.2\ sec^{-1}$

where k_1, and k_2 are the kinetic rates of the *USD* process and dissociation of PT due to thermal fluctuations, respectively.

Combining these two with the delay involved in tile 'docking', an approximate assembly time/tile is $\leq 10\ sec$. Most of the experimental assembly of the tile arrays is performed overnight. This involves precisely controlled thermal annealing protocols in which temperature is reduced very slowly so as to maintain local thermodynamic equilibrium during the course of assembly. In enveloped tile assembly, such thermodynamic equilibrium would not be required and therefore the assembly may need less time in laboratory implementations.

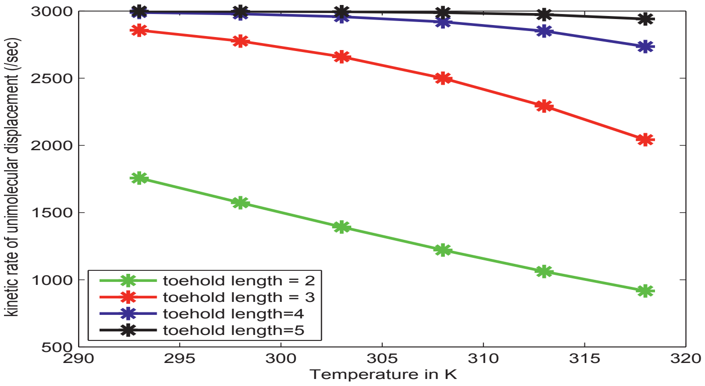


Figure 6.12: Kinetic rate of unimolecular strand displacement for different lengths of output toeholds vs. assembly temperature

6.7 Summary

In Winfree's kinetic Tile Assembly Model (kTAM) [154] of unprotected tiles, the assembly error rate (ϵ) and the assembly growth rate (*r*) are related: $r = \beta\epsilon^2$, where $\beta = 0.75 \times 10^6$. Thus, achieving lower error rates requires a substantial reduction in assembly growth rate. For example, to grow a tile pattern at a rate $r = 1$ tile/sec, the best error rate that can be achieved is 0.1%. To achieve an error rate of 0.01%, growth should be reduced by a factor of 100, which entails a waiting time of 1.6 minutes for each tile to be added. However, if one is willing to wait for an hour for each tile addition, an error rate as low as 3×10^{-4} can be achieved.

To the best of our knowledge, the maximum error improvement demonstrated in previous studies [156, 49] gives an error rate that is linearly proportional to the assembly growth rate e.g., for a 2x2 proofreading tile set [156], $r = \beta\epsilon$. Thus, there is a 10^4-fold increase in the growth rate for a target error rate of 10^{-4}. In other words, for the same physical conditions that result in a 1% error rate for the original tile set, the 2x2 proofreading tile set yields a 0.01 % error rate. Looking to the experimental feasibility of these tile assembly methods, it may be concluded that the physical conditions in which a perfect 10x10 Sierpinski pattern can be grown error-free or with a maximum of one tile faulty, can also grow a pattern of size 100x100 error-free using the proofreading tiles.

There are two important observations made from the simulations of error rates in the ET self-assembly: first, in order to achieve low error rates ($\approx 0.001\%$), the toehold length should be kept below 4 bases; second, error rates are further suppressed for longer protected lengths of *ET* 'inputs'. Based on these analyses, the ET assembly has the potential for a better error-prevention than any other state-of-the-art error-prevention tile assembly mechanisms.

Further, the assembly time of ETs is in the range of $\leq 10\ sec/$ site. Therefore, ET assembly may be designed with an error rate of 0.001%, where tiles assemble at a rate of 1 tile/10 sec. In Winfree's assembly model of unprotected tiles [154, 156], the error rate has a quadratic dependence over the assembly growth rate, thus reducing the error rate from 1% to 0.01% would result in significant slow down of assembly e.g., a single tile addition would require as long as 3 hours. Although redundant tile sets [156, 65], for example a 3x3 tile set, may reduce the assembly errors upto the range of 0.001% and without a significant slowdown in the assembly process, resource overhead due to redundancy of tiles and efforts required to design a large set of orthogonal sticky-ends may limit the prospects of their use in experiments. On the other hand, the enveloped tiles are compact, have strong error-prevention potential, and assembly process of enveloped tiles is kinetically-controlled i.e., in contrast to a precisely controlled thermal annealing in the unprotected tile assembly, enveloped tile assembly may be implemented in narrow temperature range (isothermally) near the room temperature. Combining together these features, the enveloped tiles may serve as error-resilient tile substitutes to the unprotected tiles.

7

Minimal System of Self-replicating Tile Patterns

Self-assembly of DNA tiles has been used as a programmable platform to design complex 2-D patterns and/or to perform computation as patterned structure forms. Furthermore, these patterns can be used as scaffolds to affix other molecular structures, thus providing breadboards for chemical engineering at the molecular scale. On the other hand, a self-assembling system of DNA tiles could serve as a medium for the design of synthetic replication systems [120, 75, 6]. The interest in synthetic replication systems comes from the insights they may provide on the principles behind the origination of life [103]. There is interest also from the point of view of what template replication may have to offer for manufacturing [138].

To extend the ongoing efforts in the field of DNA tile self-assembly, we posed a simple design challenge: create a minimal self-replicating system for 2-D tile patterns. The aim herein is to autonomously produce a large number of copies of the assembled tile patterns, which presently requires extensive wet-lab efforts. Self-replication of tile patterns would offer a low-cost and efficient nanomanufacturing, if it were based on an automated, dynamically-controlled assembly and disassembly of tiles — a missing feature in the algorithmic tile self-assembly framework introduced by Winfree [154].

In this chapter, we first examine the tile self-assembly framework for the realization of self-replicating tile patterns. We used Penrose's self-replicating system, illustrated in Section 3.4.1, to analyze whether the basic requirements of self-replication are met by the tile self-assembly framework. One of these requirements is the self-assembly medium itself, where smaller components autonomously assemble to produce organized structures. The second requirement of an autonomous self-replicating system involves the ability to undergo dynamic disassembly and reassembly of self-assembling structures.

DNA tile self-assembly was originally introduced as a static process: tiles could not be disassembled and reassembled dynamically. To introduce such a dynamic control in the static tile self-assembly process, we discuss a mechanism of *external switching* in tile self-assembly. The external switching mechanism that uses switching enabled tiles, is described in Section 7.3. This is followed by the design of a self-replicating system of rectangular 2-D patterns of tiles based on the abstract Tile Assembly Model. At the end of this chapter, we describe the simulator designed to study the pattern self-replication using the abstract and kinetic models of the tile self-assembly process.

7.1 Self-replication in the Tile Assembly Framework

Although Penrose's [102, 103] design of self-replication could not take the further step toward more complicated machines that actively and autonomously replicate, it serves as one of the first concepts that defined the properties of components (e.g., wooden blocks) and environment in which these components could build a self-replicating machine under external agitation.

In light of the basic properties of Penrose's self-replication model (discussed in detail in Section 3.4.1, Table 7.1 presents the features of a DNA tile self-assembly framework that satisfies these requirements. Both static self-assembly of DX Tiles and dynamically controlled self-assembly using Enveloped Tiles [50] or Signal-passing Tiles [97] have been considered. If there is a feature in the tile assembly medium that meets a particular requirement of the self-replication, it is marked with (✓); absence of the feature is marked with a (✗). The minimal requirements for a self-replication medium are: #1 Building blocks should be in two states (inactive and active); #2 Building blocks should have definite boundaries, i.e building blocks of the same type should not assemble; #3 There should be a driving force for the formation of larger structures using building blocks; #4 Building blocks should transfer the activation to the inactive building block, which collides to assemble with the structure; #5 There should be specificity in the interactions of the building blocks.

Table 7.1: Minimal requirements for a 1-D self-replication system of DNA tiles.

Minimal Requirements	DX Tiles	Enveloped Tiles
# 1	✗	✓
# 2	✗	✓
# 3	✓	✓
# 4	✗	✓
# 5	✓	✓

7.1.1 Design of a Penrose-like Self-replicating System of One-dimensional DNA-Tile Patterns

To illustrate the concept of self-replication in the tile self-assembly framework, we discuss a simple design of a Penrose-like [102, 103] system of self-replicating DNA-tile patterns. As mentioned above, a Penrose-like self-replicating system in the tile self-assembly framework would need tiles that have the property to respond to a binding event at one end of the tile by activating its other end. Such a property is not present in the simple DX or TX DNA molecular tiles, but recently introduced designs of switching tiles, such as Activatable Tiles [83], Enveloped Tiles discussed in this thesis [51] and Signal-passing Tiles [97] could meet this requirement.

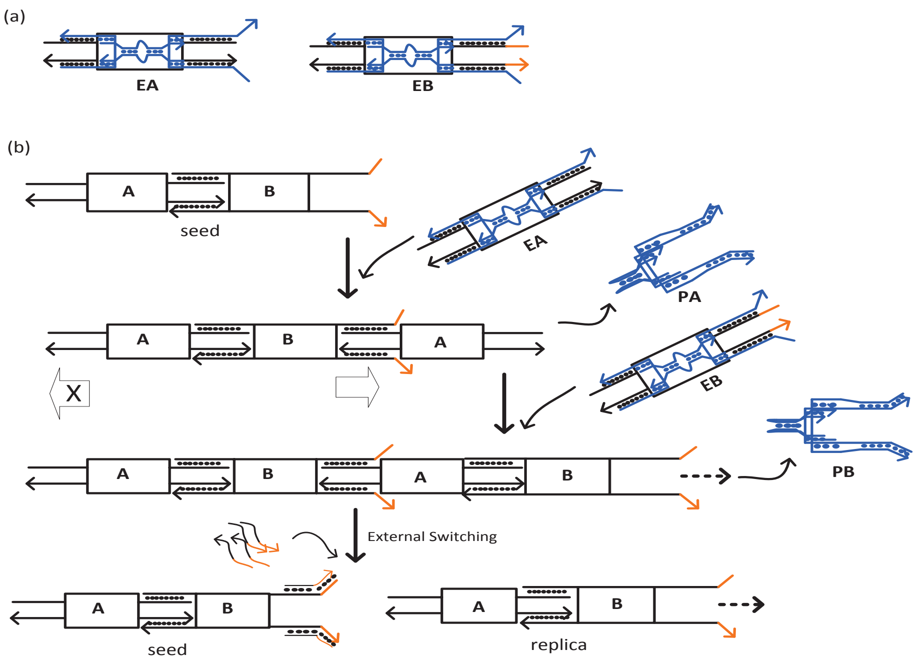


Figure 7.1: Penrose-Like one-dimensional self-replication using Enveloped Tiles. (a) System consists of two types of Enveloped Tile components: EA, EB. (b) A pre-formed two-tile seed structure (AB) is introduced in the system in which Enveloped Tiles, EA and EB self-assemble sequentially a repeating chain of AB patterns. Finally, an external switching mechanism dissoctates the original seed structure and replicated copies, thus producing multiple replicated copies of the AB tile structure.

Figure 7.1 illustrates a conceptual design of a Penrose-like self-replication system using two Enveloped Tiles. In this system, a pre-assembled seed structure of tiles (one-dimensional pattern, AB) serves as a target pattern of the replication system. Larger size ($>$ 2 tiles) linear patterns can be designed as seed (target pattern). For example, Signal-passing Tiles have experimentally been demonstrated to assemble one-dimensional lattices consisting of up to five different tiles [97] .

A further consideration in the tile self-replication system is to be given to introduce a mechanism of global switching in the tile assembly medium so as to dissociate assembled replicas of the seed. In Penrose's system [102, 103] agitation was used to dissociate the assembled block at their loose links, which resulted in separating the seed structure and assembled replica structures. In the DNA assembly medium, implementing a global switching would require a mechanism that can be activated at multiple sites in an synchronized manner. Previously, there have been suggestions for the design of global switching mechanisms in the DNA assembly medium. For example, a laser beam-controlled global switching mechanism has been suggested for an experimental implementation of a DNA-based system of molecular ping-pong game [71]. In this mechanism, the basic concept is to use infrared sensitive dyes with the DNA tiles that can be globally exposed by laser beams so as to switch between two states: from bound state to unbound and vice versa . A pH controlled dynamic and cyclic switching of DNA complexes between a duplex state and a triplex state has also been demonstrated by Minero et al. [88]. Such mechanism could be used for the design of a global switching in the DNA medium. In Section 7.3, we discuss a design of global switching based on toehold-mediated DNA strand displacement for tile pattern self-replication.

7.1.2 How to Self-replicate Algorithmic Self-assembled Two-dimensional DNA-Tile Patterns?

A wider class of 2-D tile patterns are formed by algorithmic tile self-assembly [154, 112, 14, 70], thus the work herein was set out as a design challenge to implement a system of tiles that can self-replicate algorithmically assembled rectangular patterns of tiles. Herein, we present a design of a minimal system of self-replicating 2-D rectangular patterns of tiles within the framework of the aTAM [154, 113]. The design adheres to simplicity and implementation feasibility in four aspects: 1) double crossover (DX) tiles are used; 2) all glues are of strength 1; 3) tiles do not carry signals; and 4) the replication process is enzyme free.

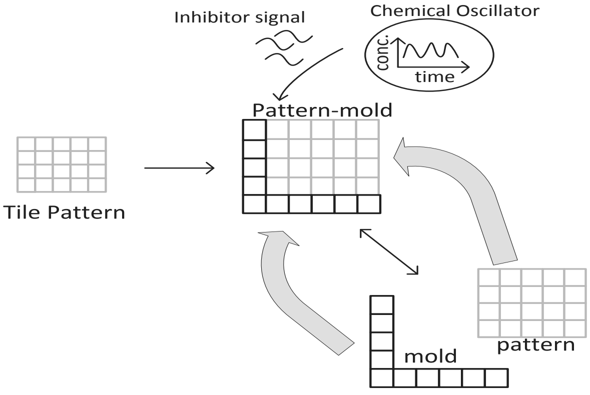


Figure 7.2: A simplistic view of the pattern self-replication system.

The pattern self-replication system, illustrated using a simple flowchart in Figure 7.2, is designed using additional tiles, which self-assemble to form a mold structure around the L-shaped South-West border of the target pattern of tiles. The assembled mold consists of switching enabled tiles that are dynamically triggered by an externally supplied inhibitor signal of DNA to dissociate the pattern and mold templates. The dissociated mold and pattern structures further catalyse the assembly of new templates of patterns and mold structures, respectively. The inhibitor signal is cyclically released by a chemical oscillator tuned to the time intervals involved in the mold formation and pattern formation. Thus, the entire process forms the basis of a cross-coupled self-replication system of 2-D patterns of tiles.

Definitions

Pattern: A target tile pattern here refers to a two-dimensional ($m \times n$) tile lattice assembled using algorithmic self-assembly [154]. We consider initialization of an L-shaped seed (an L-shaped frame of tiles that forms the South and the West boundaries of the lattice) as an input to self-assemble the rest of the pattern (pattern formation from L-shaped seed is explained in Section 7.6.1). We assume that the target pattern has its South-West corner tile as a special addressable tile; this assumption is made to ensure that mold formation starts only from this particular addressable position and nowhere else in the lattice. Further, it is assumed that target pattern is a completely connected lattice, free from any hole inside.

Algorithmic self-assembly of tiles is nothing but simulation of finite state automata. Therefore, in principle, a variety of patterns can be self-assembled in the two-dimensional $m \times n$ lattice space, and each could be used as a target pattern for self-replication. Further, DNA tile based programmable transducers [21] have demonstrated tile lattice formation from pre-assembled input configurations. Such computational transducer lattices can serve as a target pattern for self-replication in this framework.

Mold: An assembled mold of a target pattern is an L-shaped configuration of tiles that are used only to assemble the mold, and do not interfere with the tiles used in the self-assembly of pattern. Mold formation starts at the special corner tile in the target pattern and both arms of the L-shaped mold configuration simultaneously form. Observe that L-shaped mold is therefore nothing but a seed (input) to self-assemble a copy of target pattern from which it was created. It should be noted that the tile lattices assembled from such L-shaped seed structure have been shown to be terminal assembly[33]

7.2 Design of Self-replicating System of Two-dimensional Patterns of Tiles

A minimal self-replicating chemical system [148] includes two elements: a template molecule and a few substrate molecules capable of self-assembling an exact replica of the template molecule. The assembled replica must be able to dissociate from the template so as to result in two templates: the former template and the newly created template. These templates need to then be able to catalyse a reiteration of the process by self-assembly of two new replicates on the two templates. Such a process theoretically results in an exponential amplification of the number of templates, and could be adopted to design a minimal self-replication system of patterns in the tile self-assembly framework.

In the following Figure 7.3, we illustrate the self-replicating system of rectangular patterns of tiles.

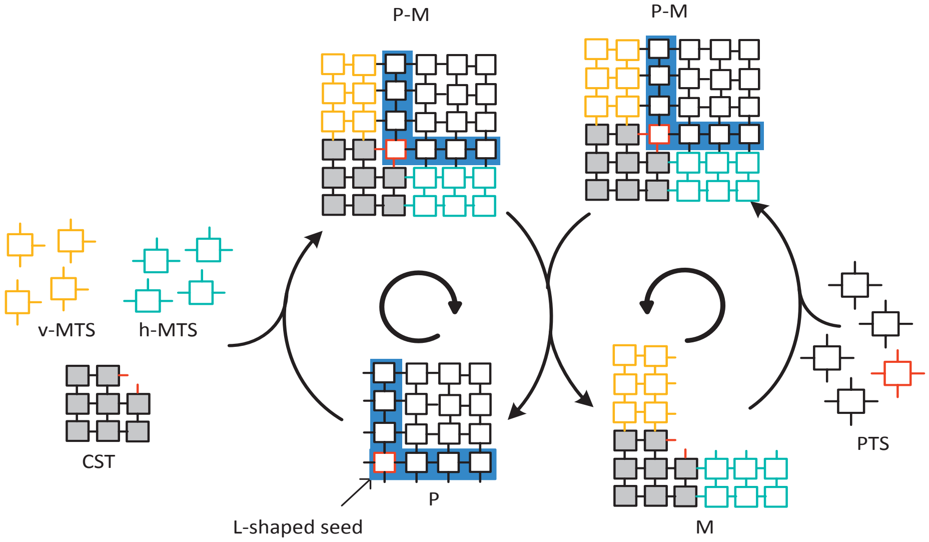


Figure 7.3: Tile pattern self-replication system. The L-shaped seed of the target pattern (P) is highlighted with a blue colour. The unique corner tile of the pattern is shown in red. Starting with the pattern structure (left cycle), pattern-mold (P-M) complex forms as CST attaches with the unique corner tile of the pattern, and further tiles from the v-MTS and h-MTS sets assemble to form the vertical and horizontal arms of the mold, reprectively. The P-M complex is dissociated into the seed and the mold (M) through external switching. The dissociated mold (M) serves as a new seed to assemble a new P-M complex (right cycle) that subsequently dissociates in the seed and the mold. Thus, the process initiates cross-coupled cycles catalyzing the formation of one another.

Let a pre-assembled target pattern, P, be self-replicated. L-shaped South-West border of the pattern P serves as seed, which enables entire rectangular pattern of tiles to be uniquely identified by the glues placed on its interior border. Considering that each tile in the pattern requires at least two bonds for a stable attachment (the algorithmic tile self-assembly at temperature(τ)-2), the formation of pattern from the L-shaped seed would be a terminal assembly process [11]. A terminal assembly system forms a unique final structure from a set of supplied components.

The replication process starts with a pre-assembled rectangular pattern (P), Corner Super Tile (CST), and a set of Mold forming Tile Set (MTS). The MTS consists of two subsets: 1) Vertical Mold forming Tile Set (v-MTS) assembles to form a vertical double layer of the mold; 2) Horizontal Mold forming Tile Set(h-MTS) assembles the horizontal arm of the mold.

We require that the target pattern contains a unique, red-coloured tile on its lower-left corner position, which is not used on any other position inside the pattern. Observe that the CST consists of eight tiles, and therefore it is stable at temperature-2. The CST is designed to bind (using two strength-1 glues) on the special red-coloured tile. Mold formation is initiated with the binding of the CST, and further proceeds as more tiles cooperatively join one by one until the entire South-West boundary of the pattern structure is covered by a double layer of tiles, creating a pattern-mold complex (*P-M*). Tiles forming the inner layer of the mold are designed as SWET type (now shown in the above schematics) with switch-enabled glue on the side that binds with the seed (pattern). The assembled pattern-mold complex undergoes a controlled dissociation, splitting into the Pattern *P* and the mold *M* structures. Observe that the dissociated mold structure has two layers of tiles, thus ensuring its stability under temperature-2 assembly framework.

In the next replication cycle, the dissociated pattern structure (*P*) repeats the left hand side pathway, and thereby, creates two (*P-M*) complexes, whereas the dissociated mold structure (*M*) drives the right hand side pathway using tiles from the PTS. Indeed, assuming we have at our disposal a tile set capable of assembling the pattern, we use the mold to reassemble the complete pattern P. Thus, by supplying the system with sufficiently many copies of the tiles within the MTS and PTS tile sets, and by continuing the process for n complete cycles, the replicator could theoretically produce 2^{n-1} copies of both the mold and the pattern structures. In a potential experimental implementation, one has to provide enough time for both the mold formation process (from a template pattern) and the pattern formation process (using the mold as a seed). Then, one adjusts the cycle of inhibitor signal supply, which triggers the pattern-mold dissociation such as to be at least as long as the maximum of the two expected time values.

7.3 A Model of Switch-enabled Tile Assembly

The abstract Tile Assembly Model (*aTAM*) [154, 113], introduced by Winfree, provides a framework where a 2-D target pattern can be self-assembled with the help of a finite set of tiles. In the aTAM, a tile is represented as a unit square with its four edges, North (N), East (E), South (S) and West (W), labelled from Σ, where Σ is a finite set of ‘glues’, including the special empty glue “0”. Therefore, a Tile *t* can be represented by the glue quadruple $\{\sigma_N(t),\ \sigma_E(t),\ \sigma_S(t),\ \sigma_W(t)\}$. A zero value of the glue denotes the absence of a sticky-end i.e., zero binding strength. We assume that our tile system is deterministic, and works under the $temperature = 2(\tau = 2)$ assumption, explained earlier in Section 2.4. Moreover, all the glues/sticky-ends used herein are assumed to be of *strength 1*.

Physical Basis of Switch-enabled Tile Assembly The concept of glue activation/deactivation has earlier been demonstrated by [83, 51], by introducing innovative mechanisms of protection and deprotection of tile sticky-ends. Signal passing and glue activation/deactivation were explored in STAM [95]. In the STAM framework, control signals pass through the tiles in order to activate or deactivate the glues of remote tiles. This signal traversal involves several concomitant strand displacement steps. Further, in the STAM framework, tiles carry the control signals, and the activation/deactivation of a remote tile would therefore depend on the success of a set of consecutive activation/deactivation events along the signal path.

In order to design a tile assembly system with attributes of active assembly, where tiles can activate/deactivate their glues through a localised strand displacement reaction, we introduce the concept of SWitch-Enabled Tile(SWET) shown in Figure 7.4(a). A (DX-)tile can be converted to a SWET by extending its sticky-end (S) with a short switching toehold (SW) that serves as a local switch between two tiles, where switching is controlled by a global signal cyclically generated by an especially designed chemical oscillator system described in Section 7.4. The switching toehold is used to mediate the binding-breaking (ON-OFF) process between a SWET and a DX-tile, see Figure 7.4(b). In the 'ON' state, a SWET is able to bind a tile using the domain (i, j) of its sticky end $(\ i, j, t_s)$, where sections *i, j,* t_s are arbitrarily chosen to be 3, 7 and 3 nucleotides long, respectively, and complementary sections are marked by (*). A periodically available DNA strand (j^*, t_s^*, sgc_1) changes the binding state from 'ON' to 'OFF', where the two tiles would eventually break apart.

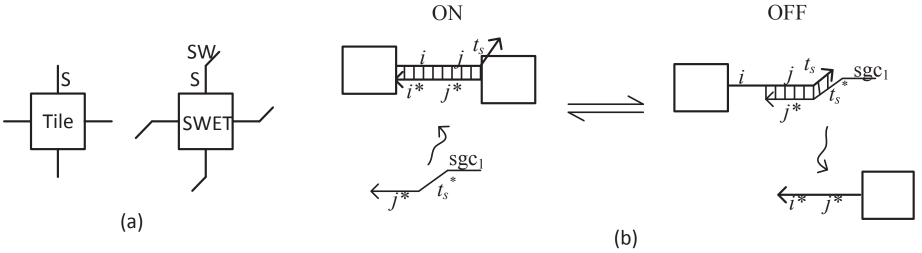


Figure 7.4: On-to-Off switching mechanism.(a) A normal tile and a SWET tile, (b) Toehold-mediated ON-OFF switching between a simple tile and a SWET tile.

The toehold-mediated switching of the SWET, from ON to OFF is essential for accomplishing dynamic assembly-disassembly of the tile structures. Such control can be achieved by a cyclic and abrupt increase in the supply of the inhibitor signal, i.e. the DNA strand (j^*, t_s^*, sgc_1) in Figure 7.4(b). This dynamic process can be implemented using an Oregonator autocatalytic reaction-system [43], which in turn can be implemented using DNA molecules as reported in [140]. Moreover, the methodology from [140], based itself on the strand displacement technique, allows for various adjustments of the autocatalytic system parameters, including the length and the amplitude of the cyclic signal, as well as the steepness of its descent. In Section 7.4 we introduce an ODE-based numeric simulation for the dynamics of one such paired SWET and inhibitor signal, showing that indeed the de-activation of the SWET is both cyclic and abrupt as shown in Figure 7.6.

7.4 Chemical Oscillator and Cyclic ON-OFF Activation of SWET

In this section we discuss the design of a chemical oscillator to trigger the dissociation of P-M complexes in each replication cycle. The oscillator cyclically releases an inhibitor signal (DNA strand (j^*, t_s^*, sgc_1) in Figure 7.4(b)) so as to switch a SWET from ON to OFF and back. Oscillator-controlled ON-to-OFF switching of the SWET(s) dissociates templates at the end of each cycle. However, as long as the mold remains bound to the pattern with only a few residual glues (from SWETs that escape OFF switching), it would reassemble instantly with the pattern upon subsequent ON switching. This would result in an overall less efficient replication cycle, as it removes free templates from the replication process. It is therefore essential that the switching from ON to OFF occurs abruptly and completely, resulting in a comprehensive splitting of all mold-pattern complexes. Herein, an Oregonator autocatalytic reaction-system [43] is used to introduce the dynamics of a chemical oscillator. We chose the Oregonator oscillator model due to three reasons: first, its dynamics fits to the need of cyclically producing inhibitor signal; second, its Chemical Reaction Network (CRN) model is widely studied and experimented; third, due to a large number of chemical reactions in the Oregonator model, its DNA implementation would produce a slow dynamic behavior [140], which would be beneficial for the ON-to-Off switching of large tile patterns.

Chemical kinetics of Oregonator reactions, adopted from Soloveichik et al. [140], are listed below (reactions:(1)-(6)).

(1): $X2 \xrightarrow{k_1} X1$ (2): $X2 + X1 \xrightarrow{k_2} \phi$ (3): $X1 \xrightarrow{k_3} 2X1 + X3$
(4): $2X1 \xrightarrow{k_4} \phi$ (5): $X3 \xrightarrow{k_5} X2$ (6): $X3 \xrightarrow{k_6} \phi$

The reversible kinetics of the ON-to-OFF switching process is given by the following reaction (7).

#	Rate constant	Species	Initial Concentration
(1)	$k_1 = 0.0871s^{-1}$	$X1$	$[X1]_0 = 8.8 \times 10^{-10}M$
(2)	$k_2 = 1.6 \times 10^9 M^{-1}s^{-1}$	$X2$	$[X2]_0 = 3.4 \times 10^{-7}M$
(3)	$k_3 = 520s^{-1}$	$X3$	$[X3]_0 = 10^{-9}M$
(4)	$k_4 = 3000M^{-1}s^{-1}$		
(5)	$k_5 = 443s^{-1}$		
(6)	$k_6 = 2.676s^{-1}$		
(7)	$k_{fw} = 4000M^{-1}s^{-1}$;	$SWon$	$[SWon]_0 = 9.8 \times 10^{-5}M$
	$k_{bw} = 0.1s^{-1}$	$SWoff$	$[SWoff]_0 = 1.8 \times 10^{-6}M$

Table 7.2: Reactions (1)-(6) form the Oregonator model; the reversible reaction (7) models the OFF/ON switching of the SWET(s)

(7): $X2 + SW_{on} \underset{k_{bw}}{\overset{k_{fw}}{\rightleftharpoons}} SW_{off}$

A bimolecular toehold exchange [174] and a unimolecular thermodynamic dissociation process represent the kinetics of the forward reaction (k_{fw}) and the backward reaction (k_{bw}), respectively. We chose k_{fw} to be $4000M^{-1}s^{-1}$ for a toehold exchange involving both the invader and the incumbent toeholds with lengths 3 nt, based on the toehold exchange model reported by Zhang and Winfree [174]. The value of $k_{bw} = 0.1s^{-1}$ for a 3 nt long duplex is derived by interpolating dsDNA dissociation kinetics data reported by Morrison and Stols [90].

A deterministic and ODE-based numerical simulation of the dynamics of the $X1$, $X2$, and $X3$ species using the COPASI software suite [66], is shown in Figure 7.5. From the deterministic time course simulations shown in Figure 7.6, it is clear that the ON-state SWET and inhibitor transitions from low to high and vice versa, are abrupt. A more realistic simulation capturing the stochastic aspect of chemical kinetics would give even steeper transitions. The time span over which a spike of the inhibitor signal has significant levels should be larger than the time required to complete a strand displacement process (ON-to-OFF switching of a SWET). As the oscillator module drives the switching module of the SWET, and both modules are implemented by strand-displacement reactions, a rational design of these reactions must satisfy different timing constraints. In order to realise such a self-replicator system with maximum yield and reliability in a wet-lab implementation, two criteria must be met. First, the dynamics of SWET switching from ON-to-OFF should be faster than the inhibitor signal dynamics. Second, the SWET switching from ON-to-OFF should be driven strongly and efficiently in the presence of the inhibitor signal, ideally approaching completion.

In the simulation shown in Figure 7.6, ON-to-OFF switching is 95% complete with arbitrarily chosen parameters of reversible kinetics ($k_{fw} = 4000M^{-1}s^{-1}$ and $k_{bw} = 0.1s^{-1}$ for a three nucleotide switching toehold) given by reaction(7) in Table 7.2. In this case, a mold with up to 20 SWET tiles would likely retain one tile that remains in the ON state (meaning a point of binding between mold and pattern) during the switching cycle, constituting a possible re-engagement point for the mold and pattern. Although this would cause a reduction of the overall replication efficiency,

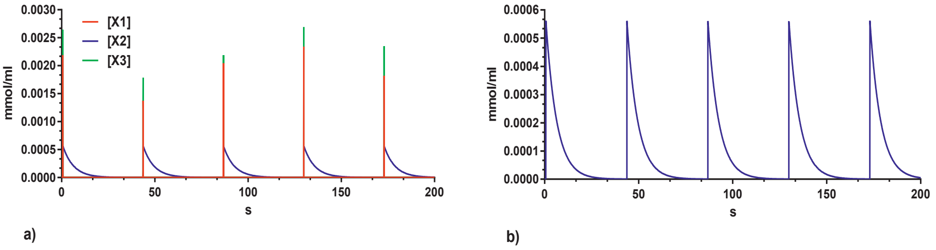


Figure 7.5: The dynamics of the Oregonator model; the system parameters are those from Table 7.2 a) all three species X1, X2, and X3; the concentration of the $X1$ and $X3$ species is overlapping in most of the cases, though at different amplitudes; b) the oscillatory dynamics of the $X2$ species.

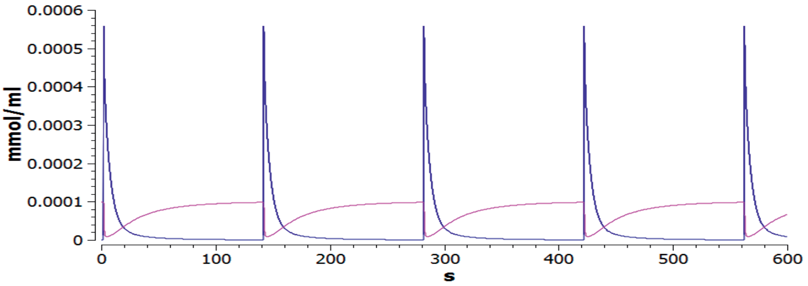


Figure 7.6: Cyclic, abrupt, and virtually complete deactivation of a SWET tile using an inhibitor signal modulated by an Oregonator autocatalytic system. The blue line represents the dynamics of the inhibitor signal, while the purple one represents the concentration of the ON-state SWET. The total concentration (ON and OFF) of SWET is $10^{-4}M$.

the ON-to-OFF switching proportions can further be very significantly improved by increasing the switching toehold length [174] in the SWET. It therefore is reasonable to assume that a rationally designed DNA-strand displacement reaction network would be able to meet both the timing and efficiency demands of SWET-switching enabled template dissociation. Taking the assumption that the tile assembly is deterministic, it can be further assumed that both the mold and the seed assemble in some finite time, say T_m and T_s respectively. For a reliable self-replication, the ON-OFF cycles of the Oregonator oscillator have to be synchronised with the times T_m and T_s, for example the ON period should be larger than the largest of the T_m, T_s. In order to test if the designed oscillator can be implemented for different time periods of oscillations so as to synchronise it with the above timing requirement, we performed a parameter scan only on the activation reaction, i.e., the backward rate constant k_{bw} of reaction (7) in Table 7.2. We observed that the time of each cycle, in between two spikes of the inhibitor signal $X2$, is tunable (see Figure 7.7).

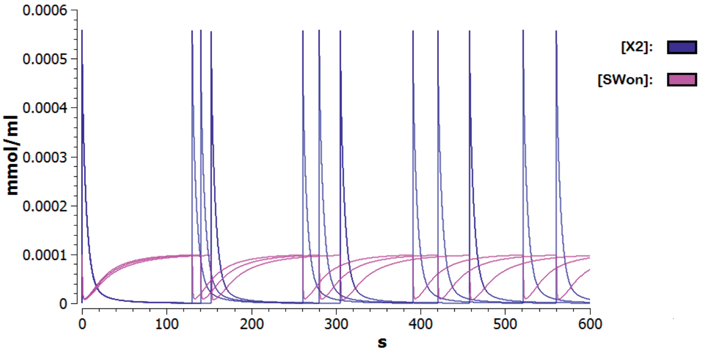


Figure 7.7: A parameter scan for the rate constant k_{bw} in Table 7.2 for the values $0.085, 0.1,$ and $0.115s^{-1}$, respectively.

7.4.1 Oregonator CRN-to-DNA Transformation

The above mass-action based oscillatory dynamics can be implemented by a set of DNA-strand displacement reactions using Soloveichik et al.'s [140] CRN-to-DNA transformation, as shown in Figure 7.8. In the CRN-to-DNA transformation, reaction species (e.g., X_1, X_2, X_3 of the Oregonator CRN) are replaced by DNA strands termed 'signal' species (see Figure 7.9). Further, a few rationally designed DNA complexes termed 'fuel' species are also introduced. The fuel species mediate the reactions between the signal species through well-known toehold-mediated DNA strand displacement mechanism [174], which also provides free-energy driving force for the overall reaction system and produces desired dynamic behavior [140]. Considering the implementation of DNA reaction system in a batch reactor, the concentration of the fuel species would reduce with the time. Therefore, the dynamic behavior of the DNA-based Oregonator reaction system may differ from the idealized CRN dynamics of the oscillator, which is based on the test tube operation. For example, DNA signal species would be at higher concentration as the fuel species would be consumed to produce inert DNA wastes, thus the oscillations may eventually cease due system reaching to its thermodynamic equilibrium. If the reaction system is constantly supplied with fresh species, for example by using reservoirs, and the product species are removed, a close to idealised oscillatory behavior can be produced for longer time.

DNA signal species (S), as shown in Figure 7.9a, consists of two types of moieties within the single stranded DNA: 1) version moiety; 2) control moiety. The version moiety of a signal species represents the history of the signal. For example, there can be multiple copies of a signal species produced from different reactions, and thus each copy of the species can be represented by assigning a separate version moiety, as shown in Figure 7.9b. Behaviors of such signal species is identical within the reaction environment. The control moiety consists of three parts: toehold control sequence 1 (Sc1), branch migration domain (Sbm), and toehold control sequence 2 (Sc2). The Sc1 and Sc2 provide toehold controls to interact with the fuel species, while the Sbm is used

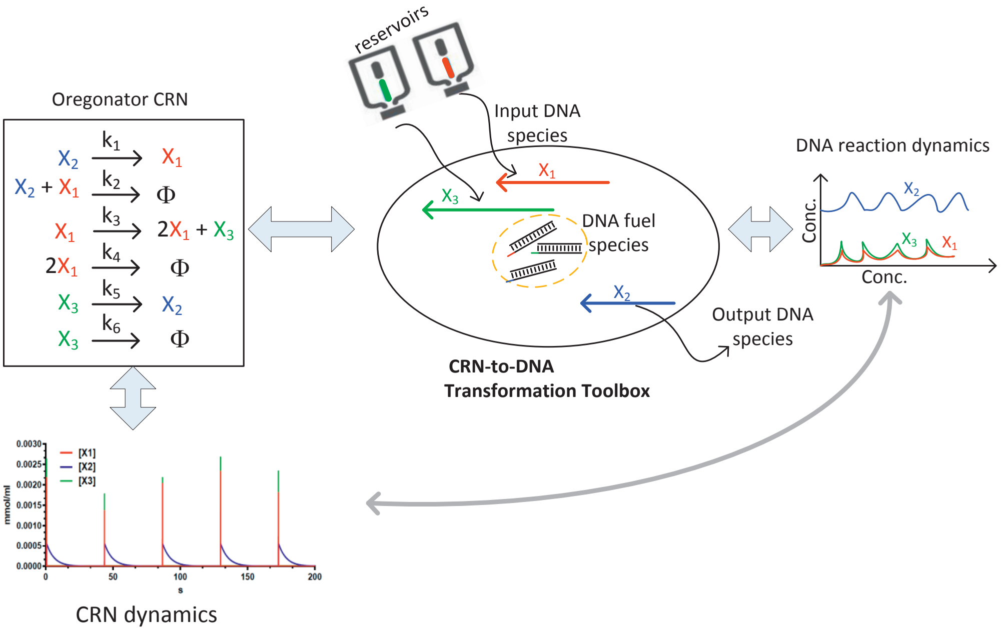


Figure 7.8: Soloveichik et al.'s [140] CRN-to-DNA transformation of the Oregonator CRN.

in the core of fuel species. Signal species are designed to have non-complementary DNA domains, hence they do not interact spuriously with each other.

In order to drive the ON-to-OFF switching of SWET(s), the inhibitor signal moiety (j^*, t_s^*, sgc_1, shown in Figure 7.4) can be used as a version control of DNA signal species representing $X2$.

7.5 An Example: Tile Sets design and topologies

In the following we discuss designs of two sets of tiles: the *PTS* used for the self-assembly of a target tile pattern, and the *MTS* (these tiles are SWET tiles) that assembles mold structure of the target pattern. All of these tiles contain only temperature = 1 glues, while the temperature of the entire system is 2. In other words, all tiles assemble through the cooperative binding process in the tile self-assembly framework.

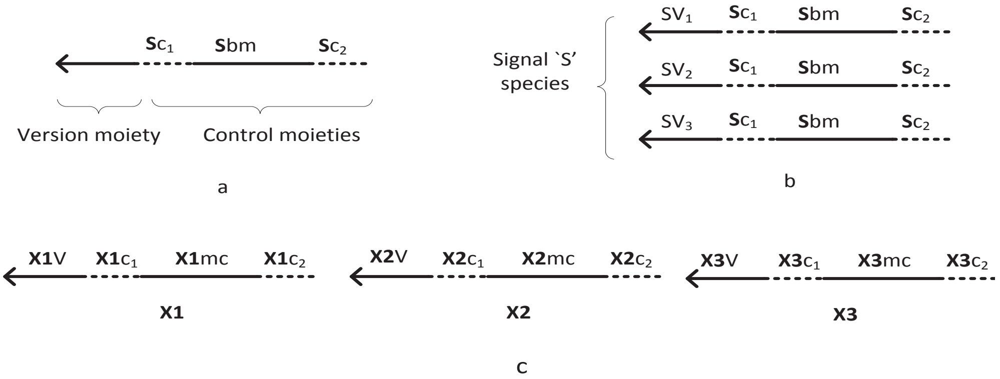


Figure 7.9: DNA signal species. a, b Representation of a DNA signal species. c DNA Singal species of the Oregonator reaction system

7.5.1 Tile Sets Design

The PTS: To illustrate the tile set design for a pattern formation, we consider a target pattern of 4x4 tiles, as shown in Figure 7.10. The pattern is assembled from the classic XOR tiles, explained earlier in Section 2.4.2. The PTS for this pattern consists of five tiles (s, a, b, c, d), as shown in the following figure. The red color tile (s) is an unique tile, which appears only once inside the pattern. The PTS tiles can be implemented using DAO molecular structures of DNA, as shown in lower-half of Figure 7.10.

The MTS: Once the PTS is established, we can design the tile set for the mold formation. This set can be split into the horizontal-Mold forming Tile Set (h-MTS), forming the horizontal arm of the mold, the vertical-Mold forming Tile Set (v-MTS), forming the vertical arm of the mold, and the Corner Supertile (CST) structure serving as a seed for the mold assembly.

The CST The CST can be seen as a merger of 8 tiles forming two layers of tiles around the corner, as shown in Figure 7.11(a). an inner layer of the mold (IML) having tiles that assemble with the unique corner tile of the Pattern and an outer layer of the mold (OML) supporting the inner layer. The inner layer consists of IML-North (a SWET enabled with switching on its East glue), IML-middle, and IML-East (a SWET enabled with switching on its North glue), forming a concave superstructure. The outer layer consists of five tiles that are designed to assemble with the tiles of the inner layer and form a stable structure at temperature-2. In both cases, we require that the glues at the concave corner of the CST match with the special red-coloured tile of the PTS. A topological view of the CST binding with the red color tile of the pattern is shown in Figure 7.12. Further, two new glues (mb and mt) are introduced. The mt is assigned to the North glue of the IML-North tile, and mb is assigned to the East glue of the IML-East tile. Each

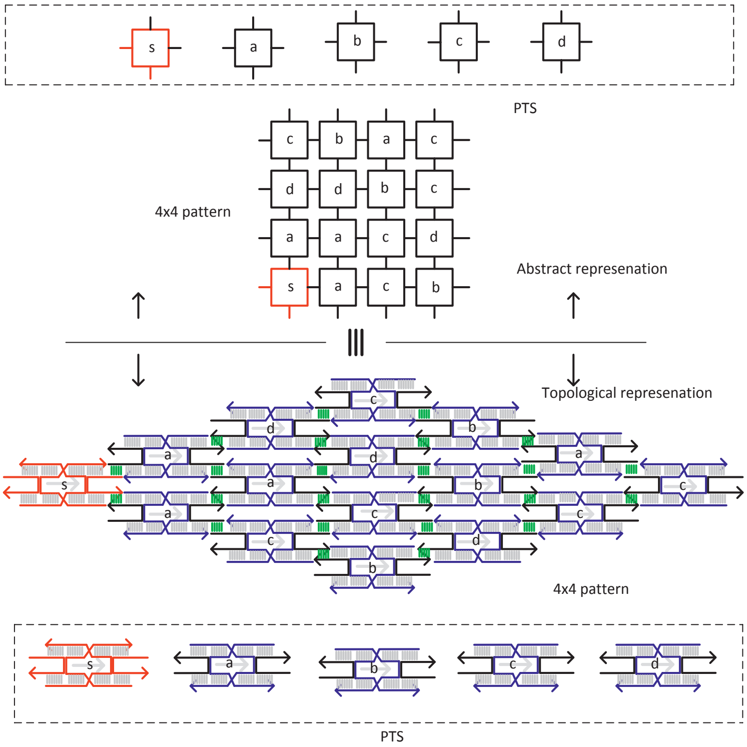


Figure 7.10: Pattern forming tile set of a 4x4 tiles pattern. Abstract view and topological view are shown in lower-half and upper-half of figure, receptively. Gray arrows inside molecular tile structures show the direction of their orientation during the assembly.

of these glues in cooperation with the glues in the pattern initiate the formation of the horizontal and vertical arms of Mold, as shown in Figure 7.13.

Implementation of the CST: The CST can be implemented using eight DAO molecular structures, as shown in Figure 7.11(b). However, the assembled 8-tile structure may not remain stable in experimental conditions, which often involves multiple rounds of thermal annealing. Therefore, the CST should be implemented such that it stays as a superstructure, and do not disintegrate during the self-replication process. There can be two such implementations of the CST, as illustrated in Figure 7.11(c).

The implementation-I has been designed by converting the topological CST structure of (b) into long continuous DNA strands. The scaffold of the CST consists of nine DNA strands. The second implementation, the implementation-II, has been designed using DNA origami [73] approach. In DNA origami, the CST can be implemented using

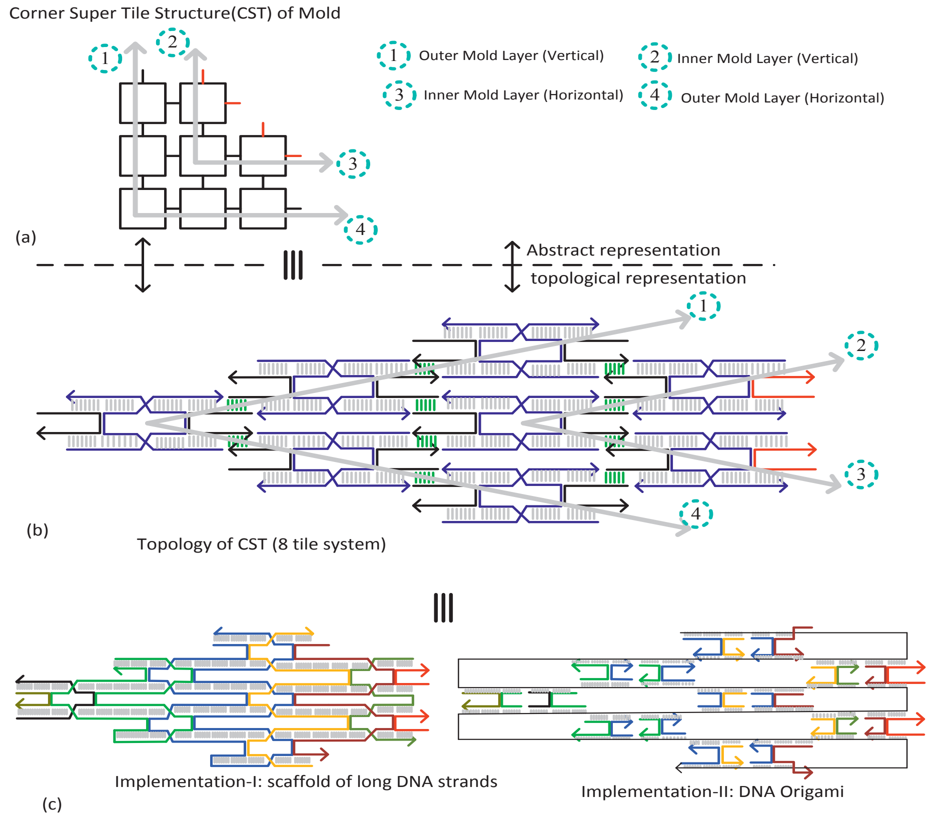

Corner Super Tile Structure(CST) of Mold
1
2
3
4
1 Outer Mold Layer (Vertical)
2 Inner Mold Layer (Vertical)
3 Inner Mold Layer (Horizontal)
4 Outer Mold Layer (Horizontal)
(a)
Abstract representation
topological representation
(b)
Topology of CST (8 tile system)
(c)
Implementation-I: scaffold of long DNA strands
Implementation-II: DNA Origami

Figure 7.11: Corner supertile structure and its implementation.(a) Abstract view of 8-tiles CST structure. Double tile layers, which initiate the formation of mold, are marked by arrows with numbers.(b) Topological representation of the CST, obtained by a direct mapping from the abstract design to DAO molecular tile structures. (c) Two implementations of the CST: implementation-I uses nine DNA strands; implementation-II uses a DNA origami approach (a single long DNA strand and 32 short DNA staple strands).

a long DNA strand and 32 short staple strands. Observe that the implementation-I uses multiple long DNA strands, thus it is prone to misfolding. However, DNA origami approach uses one long DNA strand, which is folded into the desired structure with the help of short staple strands. Therefore, in a one-pot reaction, the DNA origami approach may be better choice for the implementation of the CST.

The h-MTS and v-MTS Consider now the tiles that form the horizontal and vertical arms of the L-shaped seed of the pattern. We define the bottom-Glue (bG) and the

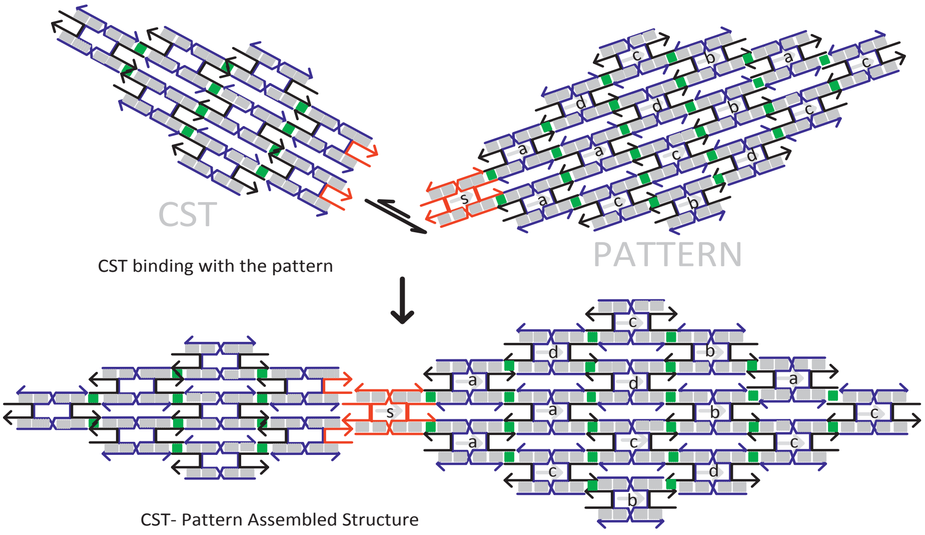

CST
PATTERN
CST binding with the pattern
CST- Pattern Assembled Structure
a
b
c
d
s

Figure 7.12: CST binding with the pattern.

left-Glue (lG) sets as containing all South glues of tiles within the bottom and left arms of the L-shaped seed of the pattern. The h-MTS tile contains $|BG|$ tiles that form the inner layer of the bottom border. For each glue $a \in BG$ we create a SWET with glue a on the North side that is enabled with switching control, and glue mbi on both its East and West sides. Similarly, the v-MTS contains $|LG|$ tiles that form the inner layer on the left border. For each glue $b \in lG$, we create a corresponding SWET with glue b on the East side that is enabled with switching control, and glue mti on its North and South sides. A fixed glue mbf is designed for South sides of the tiles of the h-MTS and v-MTS, which provides a binding for the tiles forming the outer layer of the mold.

7.6 Tile Pattern Self-Replication Simulator

The following presents description of the Tile Pattern Self-replication Simulator (TPSS) that has been developed to study the tile pattern self-replication process, discussed earlier in this chapter. The simulator comes in two versions that are based on the underlying models of tile assembly (described in Section 2.4): 1) abstract Tile Pattern Self-replication Model (aTPSM) is based on the abstract Tile Assembly Model (aTAM) [154, 113]; 2) kinetic Tile Pattern Self-replication Model (kTPSM) is based on the kinetic Tile Assembly Model (kTAM) [154]. The simulator takes an algorithmically programmable rectangular pattern of tiles $(m \times n)$ as an input to reproduce copies of

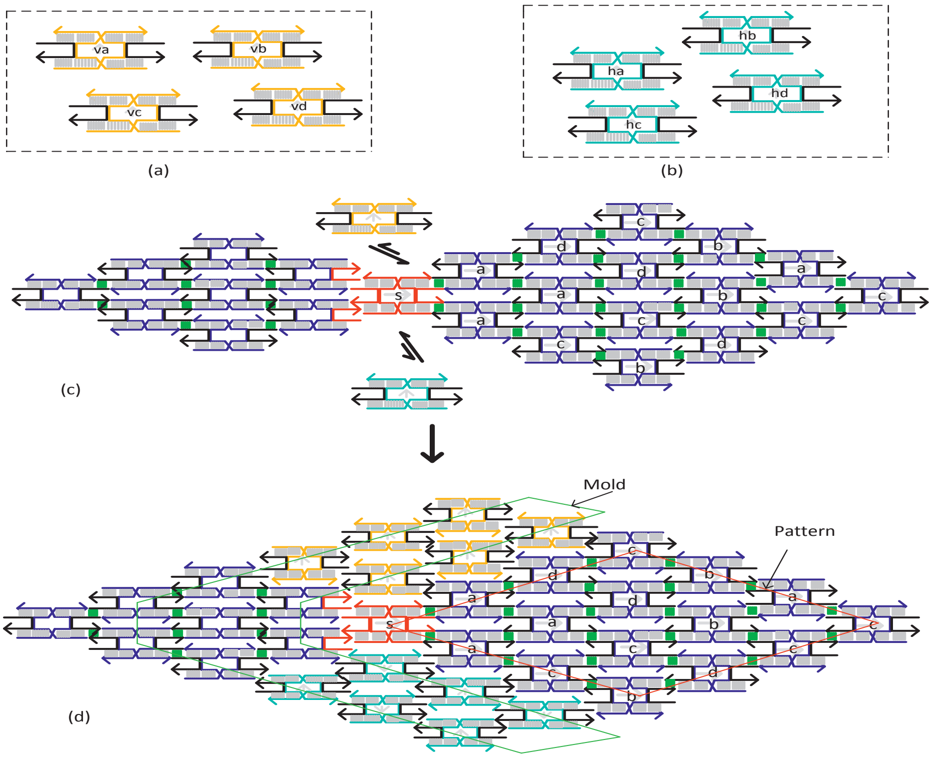


Figure 7.13: Topological view of the mold formation process. (a) Molecular tile structures of vertical mold forming tiles. Observe that the direction of binding of these tiles is vertical (shown by an arrow inside the molecular tile structure). (b) Molecular tile structures of the horizontal mold forming tiles.(c) Initiation of mold arm's formation. (d) Final mold-pattern superstructure.

the pattern, where *m* and *n* are the number of tiles forming the width and the length of the rectangular pattern, respectively. At least one pre-assembled copy of the target pattern is required to start the self-replication process.

A target pattern is initialized in two steps: First, an L-shaped random seed structure (West and South edges) of the pattern is created using an XOR tile set. A corner tile is first chosen and then the two arms (South and West edges of the L-shaped structure) grow in parallel as more tiles matching by single side join the growing seed structure. Second, using the seed structure, full pattern is formed as more tiles matching by two sides (or decided by the kinetics of the tile binding) assemble to the seed structure.

The simulator first builds a mold layer of tiles using a mold forming tile set on the outer edges of the West and South edges of the target pattern to form a combined Pattern-Mold structure (P-M). The P-M structure is dissociated into the mold and the pattern at the end of the first cycle. In the subsequent replication cycles, as illustrated schematically in Figure 7.14, the dissociated pattern and mold copies serve as seeds to drive the two pathways of the cross-coupled system shown in Figure 7.3. For the simplicity, the mold is considered as a single layer of the mold forming tiles, as shown in Figure 7.19. This assumption would not affect the dynamics of the self-replication, because mold does not disassemble after it is fully formed.

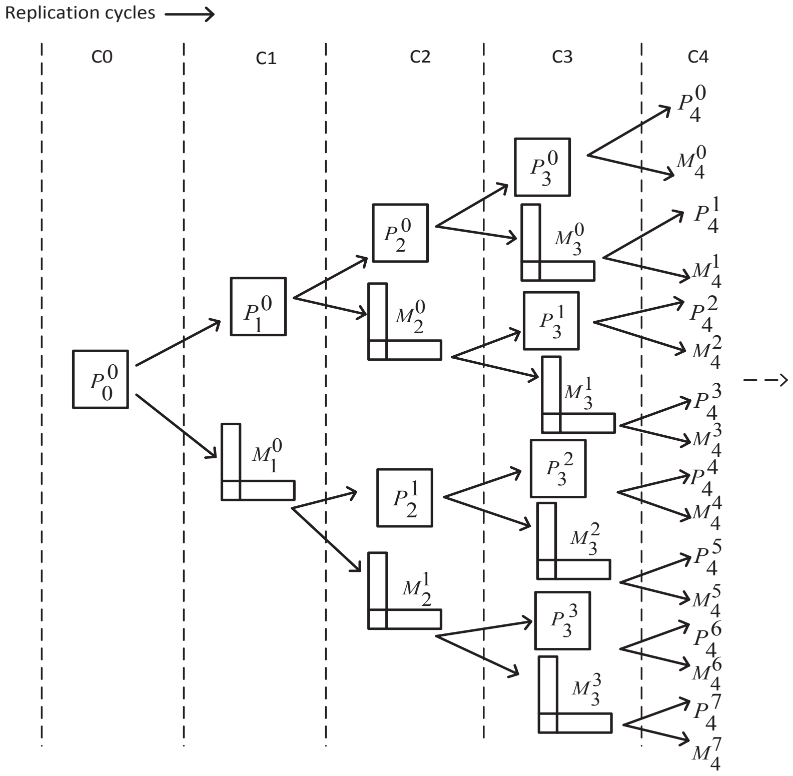


Figure 7.14: Tile pattern self-replication: unrolled cycles of cross-coupled system, shown in Figure 7.3.

In the start of the pattern self-replication, cycle C0 shown in Figure 7.14, at least one copy of the pre-assembled pattern, P_0^0, is supplied. The self-replicator produces a complex P-M, which is subsequently dissociated into new copies of P_1^0 and M_1^0 at the end of the cycle C1. In the subsequent replication cycle, C_i, 2^{i-1} copies of P and M are produced. In the C_i^{th} cycle, pattern copies are presented by P_i^j and mold copies are represented by M_i^j, where j varies from 0 to $(2^{i-1} - 1)$.

The aTPSM is based on the definitions of the aTAM model, where tiles are square units. Every tile has only two types of logical values (0/1) associated with its four edges, and glue strength of each edge is taken to be 1. The tiles are assumed to be oriented, that is, they cannot be rotated or reflected. Tiles assemble by satisfying the logical matching to their designed inputs/outputs, there is no kinetics and physics involved in the binding process of a tile in the aTPSM.

To include physico-chemical reality into the tile pattern self-replication process, the kTPSM considers assembly process of each tile as a reversible kinetic reaction, as modeled earlier in the kTAM [154]. The reversible kinetics of tile binding is governed by local assembly temperature (T), concentration of each type of constituent tile monomer, and the binding strength between two tiles. A more complete description of the kTAM appears in Section 2.4. The kTPSM uses kinetics driven tile binding for both $mold \rightarrow pattern$ and $pattern \rightarrow mold$ formation processes, described in detail in Section 7.7.2. In the kinetic model [154], tiles binding by single bond and multiple bonds have different kinetic rates of dissociation, that adds stochastic preference into the tile binding process. Due to stochastic nature of each tile binding, mismatched tiles are likely to get trapped during the assembly process. The kinetically trapped mismatched tiles are source of errors in the tile assembly processes. Also, these parameters have influence over the rate with which a assembly proceeds to forming patterns and whether the process terminates at all.

Both the aTPSM and kTPSM assume that the mold-pattern dissociation is perfect and in time, that is, in each cycle, once the formation of mold-pattern structure is complete, it dissociates into the mold and pattern structures. There is no consideration of kinetics in the dissociation of seed-mold complex.

In the following, we will illustrate: 1) the pattern forming tile set; 2) the mold forming tile set; 3) the formation of mold structure from existing pattern ; 4) the pattern formation from mold structures.

7.6.1 Pattern Forming Tile Set

To assemble a pattern in the tile self-assembly framework, a tile set and an initial configuration of tiles (seed) is required. In the absence of seed, tiles do not assemble to produce large aggregates. However, the presence of seed acts as a nucleation point where tiles join to grow the pattern. In the following, a pattern forming tile set, seed structure and pattern initialization are illustrated in brief.

A pattern forming XOR tile set is illustrated in Figure 7.15. A square tile unit having four edges (N, S, E, W) is shown in Figure 7.15(a). Input(ip)/output(op) edges and XORing logic operating implemented between them is shown in Figure 7.15(b) and Figure 7.15(c), respectively. The four tiles of XOR tile set used as Pattern forming tiles are shown in Figure 7.15(d).

Tile pattern self-replication requires at least one pre-assembled rectangular target pattern of tiles to start the self-replication process as explained in Section 7.2. Target rectangular patterns of tiles can be produced using the pattern forming XOR tile set illustrated in Figure 7.15. A number of L-shaped seed structures, as shown in Figure 7.16, can be initialized. To initialize an L-shaped seed structure, first a random tile is picked and placed at the corner of the structure, and subsequently more tiles are

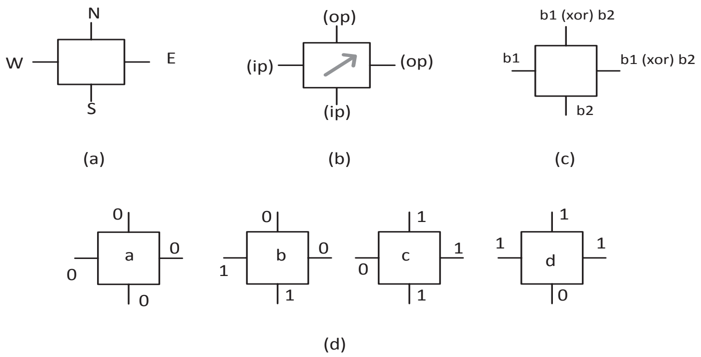


Figure 7.15: Pattern forming XOR tile set. (a) Square unit of tile. (b) Input(ip) and output(op) edges of an oriented tile unit. (c) XOR operation between inputs to derive output logic. (d) Four XOR tiles of the pattern forming tile set.

added to form the two arms of the structure. To form the horizontal arm of the L-shaped structure, tiles are selected such that logic value on the West edge of each added tile matches with the logic on the East edge of the previous tile. Similarly, to form the vertical arm, tiles are selected such that logic value on the South edge of each added tile matches with the North edge of the previous tile.

Using an L-shaped seed structure corresponding tile pattern can be formed further using the tile assembly mechanism [154]. The pattern forms by filling tiles to the seed structure, where tiles are added by matching the logic at their input pins (left and bottom) with the available sites in the growing pattern.

7.6.2 Mold Forming Tile Set

To assemble a mold structure of the given rectangular tile pattern, design of mold forming tiles is explained in Section 7.5.1. Based on the bottom and left borders of the pattern, the mold forming tiles can be divided into two sets of tiles: the horizontal mold forming tile set and vertical mold forming tile set, respectively. Mold forming tile sets implement the XOR logic, but the orientation of the tiles is different from the pattern forming tile set illustrated earlier in Figure 7.15. The horizontal mold forming tile set, Figure 7.17, uses tiles with top and right edges as inputs, and right and bottom edges as outputs. The vertical mold forming tile set, Figure 7.18, uses tiles with right and bottom edges as inputs, and left and top edges as outputs. The two tile sets form the horizontal and vertical arms of the mold, respectively, as shown in Figure 7.19.

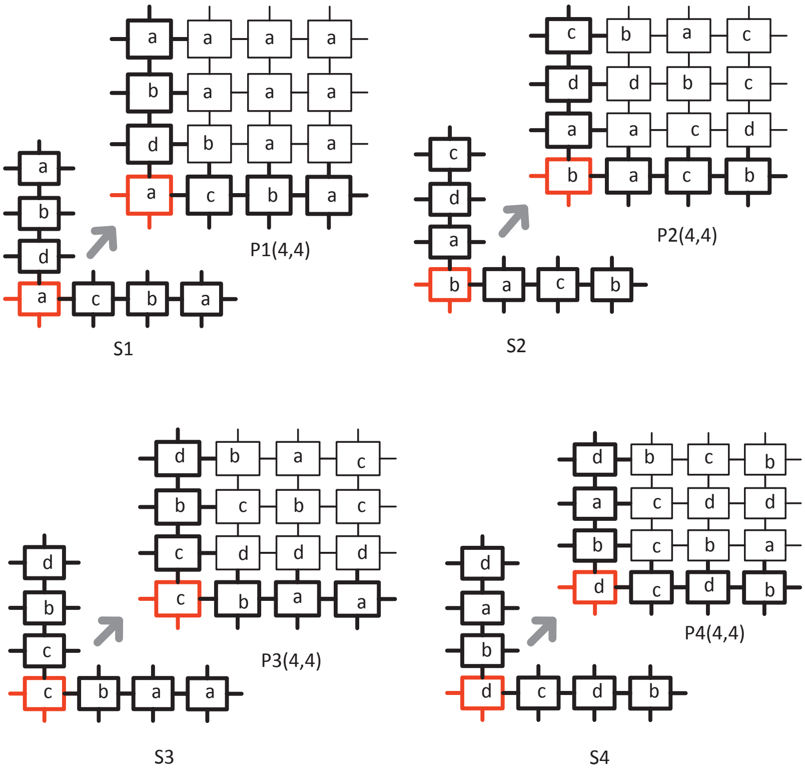


Figure 7.16: Four different L-shaped seed structures (S1, S2, S3, S4) and corresponding tile patterns (P1(4,4), P2(4,4), P3(4,4), P4(4,4),).

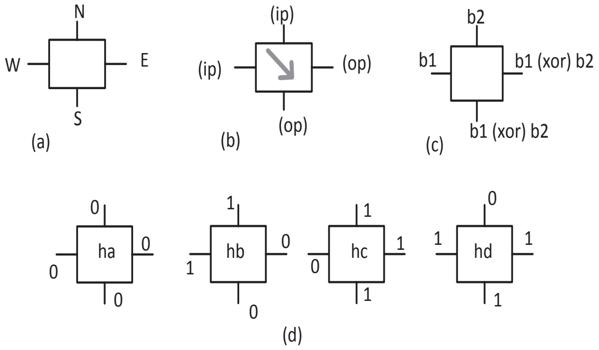


Figure 7.17: Horizontal mold forming tile set. (a) Square unit of tile. (b) Input(ip) and output(op) edges of an oriented tile unit. (c) XOR operation between inputs to derive output logic. (d) Four XOR tiles of pattern forming tile set.

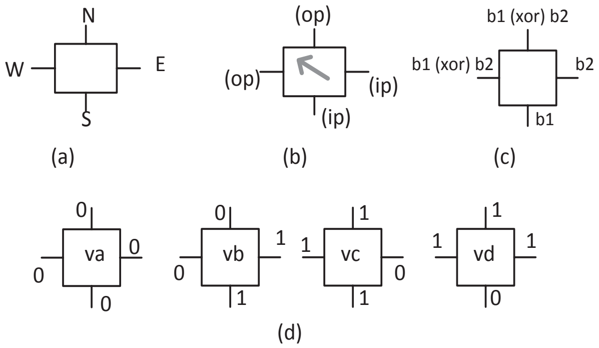


Figure 7.18: Vertical mold forming tile set. (a) Square unit of tile. (b) Input(ip) and output(op) edges of an oriented tile unit. (c) XOR operation between inputs to derive output logic. (d) Four XOR tiles of pattern forming tile set.

7.6.3 Pattern to Mold Formation

The left hand side pathway in the cross-coupled system of pattern self-replicator (shown in Figure 7.3) involves mold formation around the south-west boundary of self-replicating pattern. Mold formation , as shown in Figure 7.19, starts at the corner tile of the pattern, where mold forming tiles first bind, and subsequently the horizontal and vertical arms of the mold structure grow along the south and west edges of the pattern, respectively. Each of the horizontal and vertical edges of the mold are formed by tiles from separate tile sets: horizontal mold forming tile set and vertical mold forming tile set, as shown in Figure 7.17 and Figure 7.18, respectively. A tile in the horizontal arm of the mold attaches if its north and west edges have matching with the south edge of a tile in the pattern and east edge of the previous tile in the mold, respectively. Similarly, a tile in the vertical arm of the mold attaches if its east and south edges have matching with the west edge of a tile in the pattern and north edge of the previous tile, respectively.

7.6.4 Mold to Pattern Formation

In each cycle, assembled pattern-mold complexes dissociate into the mold and pattern structures, as explained in Section 7.4. The dissociated mold structure is used in the right hand side cycle in the cross-coupled system (shown in Figure 7.3), where the mold acts like a seed for the assembly the entire pattern, including seed. Figure 7.20 illustrates a few steps of the mold to pattern formation process. Glues on the inner side of the mold structure provide information to self-assemble the pattern using the pattern forming tiles. A tile attaches at a vacant site of the growing pattern if its south and west edges match with the north and east edges of the tiles in the vacant site, respectively. Once the filing of the rectangular pattern is complete, incoming tiles can no more attach by more than one bond, thus the pattern can not grow further: terminally assembled

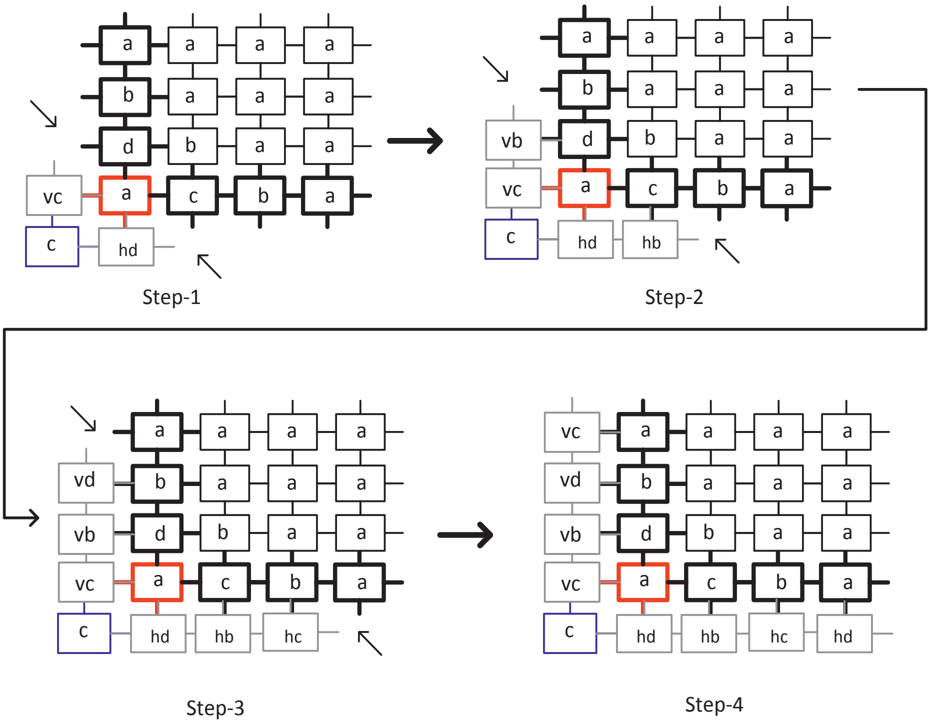


Figure 7.19: Pattern to mold formation using mold forming tile sets.

pattern.

7.7 Tile Pattern Self-replication Models

The TPSS has been developed to simulate the dynamics of tile pattern self-replication system, discussed earlier in this chapter. The TPSS simulator comes in two versions — the aTPSM and the kTPSM — that are named after the abstract (the aTAM) and kinetic (the kTAM) models of the tile self-assembly, respectively. The two models of tile pattern self-replication are described below.

A two dimensional rectangular grid of size $m \times n$, as shown in Figure 7.21, is used as a workbench of the simulator. Each cell in the grid is identified by coordinates (i, j), where $0 \leq i \leq n$ and $0 \leq j \leq m$. The conventions used to represent each of the L-shaped seed, pattern, and mold are as follows. 1) for a $m \times n$ size pattern, cell $(m, 0)$ in the grid, is used for the corner tile of the L-shaped seed of the pattern. The L-shaped seed of the pattern occupies the left most and the bottom most cells of the grid. 2) mold

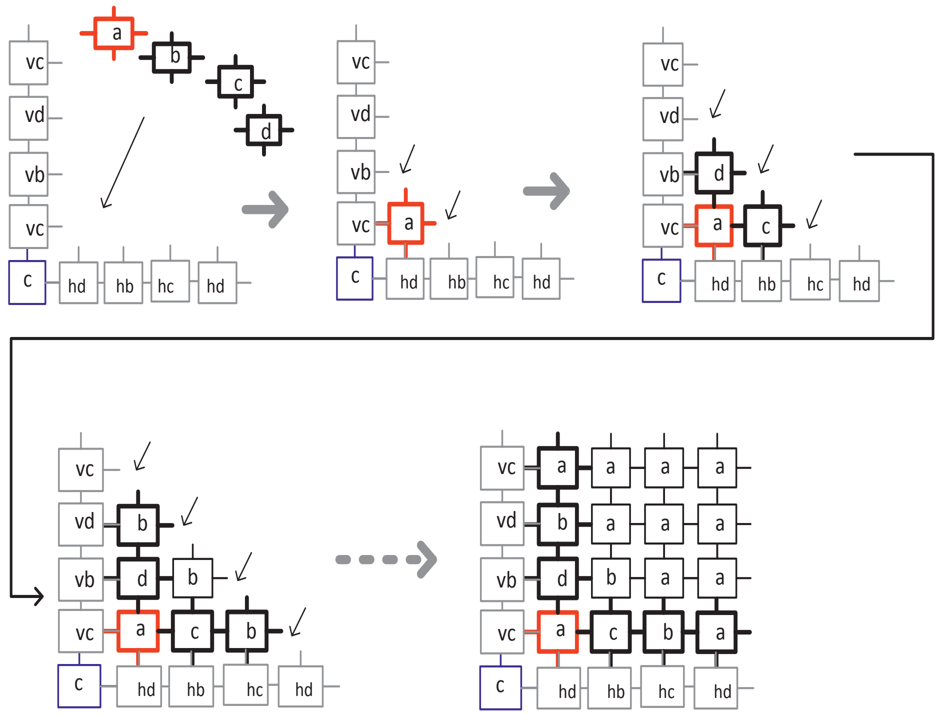


Figure 7.20: Formation of Pattern from Mold.

of a $m \times n$ pattern occupies the left most and the bottom most cells of the grid of size $(m + 1) \times (n + 1)$. 3) we used data structure to hold dynamic data corresponding to patterns and molds of self-replication cycles (shown in Figure 7.14). The data structure initializes a separate grid of size $(m + 1) \times (n + 1)$ to perform processes — $seed \rightarrow mold$ formation and $mold \rightarrow seed$ formation — for every copy of seed and mold in each cycle.

7.7.1 Abstract Tile Pattern Self-replication Model

The abstract model of tile pattern self-replication, the aTPSM, is based on the abstract tile assembly model (aTAM), explained in Section 2.4. The model includes four main steps: 1) initialization of two-dimensional rectangular pattern that is to be replicated; 2) pattern to mold formation; 3) mold to pattern formation; 4) self-replication cycles: running the steps 2 and step 3 for each copy of the pattern and the mold in parallel. In the abstract model of tile self-replication, criteria for a stable attachment of a tile is the same as in the aTAM, that is, each attaching tile must has at least τ edges matching with the tiles of the growing pattern. There is no probabilistic preference made between tiles

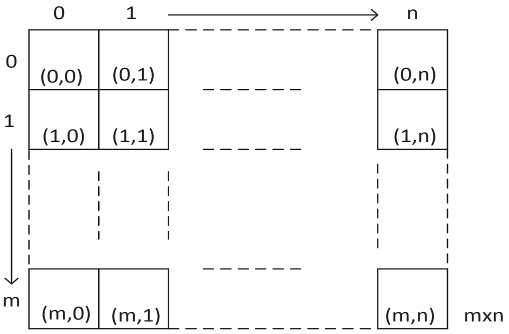


Figure 7.21: Two-dimensional grid of $m \times n$ dimension.

that attach by matching τ bonds and $< \tau$ bonds.

7.7.2 Kinetic Tile Pattern Self-replication Model

The kinetic model of tile pattern self-replication, the kTPSM, is based on the kinetic tile assembly model (the kTAM), which was used to develop the tile assembly simulator “Xgrow” [5]. While the main steps of pattern self-replication in the kTPSM are the same as aforementioned, tiles are repeatedly assembled and disassembled from the growing aggregates in both $mold \rightarrow pattern$ and $pattern \rightarrow mold$ formation processes. At each site (i, j) within the cluster of cells in the grid that would be occupied by tiles when aggregate formation is complete, the rates of association and dissociation of tiles are calculated using stochastic rules described below.

A 2-D array is used to store the arrangement of tiles in the current assembly. Initially (at time $t = 0$) the array contains all zeros to indicate empty sites, except for the cluster of sites (i,j), which are occupied by the seed. At every simulation step ($t = \Delta t, 2\Delta t...$), one of the following two events is stochastically chosen to simulate the reaction between an associating tile and an assembled aggregate: an ON-event, where a new tile is added to the array, and an OFF-event,where a tile in the assembly is removed.

On-event rate: at each time step, all *m* empty sites adjacent to the aggregate are counted to calculate the net on rate (r_{on})

$$r_{on} = mk_f e^{-G_{se}} \tag{7.1}$$

.

Off-event rate: at each time step, the off rate (r_{off}) is calculated as the following.

For all occupied sites (i, j) within the aggregate (except for the initial aggregate of the seed), the total binding strength (b_{ij}) of the tile in the occupied site is calculated. The net off rate (r_{off}) is calculated by summing up the dissociation rates of all the tiles occupying the aggregate. The net off rate is

$$r_{off} = \sum_{b} r_{off,b} \tag{7.2}$$

, where $r_{off,b} = n_b k_f e^{-bG_{se}}$ and n_b is the number of sites (i , j) having tiles attached by binding strength *b*. The kinetic rate of tile dissociation and parameters, k_f and G_{se} are described in

The kinetic rates of tile association, dissociation and the parameters, k_f, G_{mc}, G_{se} are described in Section 2.4.2.

The total rate for events of any kind is $r_{total} = r_{on} + r_{off}$, which is used to calculate both simulation time step and the type of the next event.

Simulation Time Step: time until the next event occurs, simulation time step (Δt), is chosen based on the Boltzmann distribution $P_r(\Delta t) = r_{total} e^{r_{total}} \Delta t$. Thus, Δt can be given by $\Delta t = -ln \frac{([0.0,\ 1))}{r_{total}}$.

In each simulation step, on-event or off-event is chosen based on the following stochastic criteria. An on-event is chosen with the probability, $P_r(on) = r_{on}/r_{total}$, in this case, a tile is attached to a vacant site (i, j) adjacent to the aggregate. All the adjacent sites and all tile types are equally likely to be chosen. If on-event was not chosen, an off-event occurs, and the probability that some site (i, j) within the aggregate dissociates the tile attached to it. The the probability that a site with *b* bonds dissociates is $r_{off,b}/r_{off}$, and again all such sites are equally likely. Once the event is chosen and the array is updated, all rates are calculated again to decide the next event.

To simulate the $mold \rightarrow pattern$ process, the L-shaped mold serves as seed and a 2-D array is used to store the status of pattern formation. For $pattern \rightarrow mold$ process, the seed consists of the entire pattern and the corner tile of the mold. The array used to store the status of mold formation consists of cluster of cells lying on the south most and the west most boundary of the 2-D array.

7.8 Simulation Results and Analysis

Simulations Using the aTPSM The abstract model of self-replication produces 2^{i-1} copies of the target tile pattern, where *i* is the number of self-replication cycles. For a given target pattern of size $m \times n$, producing 2^{i-1} copies requires $t_s = max(m, n) + (i - 1) \times (m + n) + i$ time-steps. The abstract model produces time stamps for each tile addition step and total time to reproduce the 2^{i-1} copies of the target tile pattern. The self-replicated copies produced using the abstract model are perfect (containing no erroneous tiles) by design.

In the context of the abstract tile pattern self-replication model, it is tempting to draw a comparison between the CA-based self-replication loops discussed in Section 3.3. In particular, the parameters, such as loop sizes and replication periods of different self-replicating loops mentioned in Table 3.1, resemble with the terms, pattern size and replication time of the abstract tile pattern self-replication model.

The closest self-replication loops to the tile pattern self-replication are Byl's loop [19] and Chou-Reggia loop [108]. The Byl's loop consists of 12 cells, and reproduces the loop in 25 time-steps. The cells in Byl's loop can be in one of the 6 possible states and there are total 43 transition rules. The Chou-Reggia loop further reduces the loops size to 5 cells, each cell can be in one of the 8 possible states, and reproduces the loop in merely 15 time-steps. Considering the abstract model of tile pattern self-replication, which uses 12 types of tiles (considering each tile to be equivalent to a state in CA concept) and four transition rules ($00 \rightarrow 00, 11 \rightarrow 00, 01 \rightarrow 11, 10 \rightarrow 11$). Tile patterns of size 4x3 and 3x2 that are similar to the sizes of Byl's loop and Chou-Reggia loop, can be reproduced in 11 and 8 time-steps, respectively. This margin in self-replication time of tile pattern self-replicator comes from the parallelism and asynchronism inherent in the tile self-assembly mechanism.

Another interesting attribute of the tile pattern self-replication is the programmability of target patterns: using the same set of tiles a larger pattern can be self-replicated by providing at least one copy of the pattern to start with. In contrast to the programmable nature of tile pattern self-replication, CA-based self-replicating loops are usually non-programmable and often implement a single dedicated functionality. Although Tempesti's loop [145] and Perrier's loop [68] introduce programmability in the self-replicating loops, large number of states (e.g., 63 states in the Perrier's loop) and increased complexity make them difficult for realization. A typical programmable self-replicating Tempesti's or Perrier's loop of size 150 cells requires ≈ 300 time-steps to reproduce itself, which is too large in comparison to the 41 time-steps needed to self-replicate a tile pattern of the similar size.

Simulations Using the kTPSM To simulate the dynamics of the kTPSM, we performed the simulation for a range of G_{mc}, G_{se} values for both a single pattern and patterns of varying sizes, as shown in Figure 7.22 and Figure 7.23, respectively.

In particular, the values of these parameters were chosen to cover the self-replication dynamics near melting ($G_{mc} \approx 2G_{se}$) transition. For a fixed G_{mc} ($G_{mc} = 16$), the self-replicator produces patterns with very few errors (1 tile out of 1000 may be faulty) for $8.4 \leq G_{se} \leq 11.4$, as can be observed from Figure 7.22. For $11.4 < G_{se} \leq 16$, errors are introduced in the self-replicating patterns. The errors grow swiftly as G_{se} approaches the G_{mc}. Simulation of self-replicating patterns of different sizes shows no significant variation in the errors near melting transition, but near $G_{mc} \approx 1.40G_{se}$

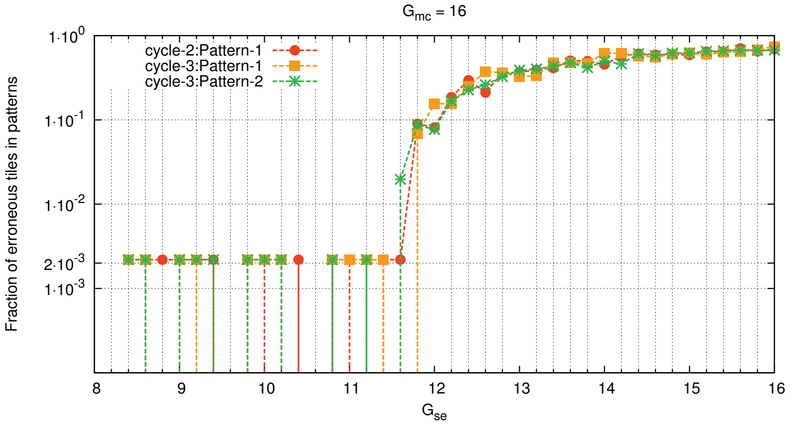


Figure 7.22: Assembly errors in self-replicating patterns produced by simulation using the kTPSM. Simulation statistics of assembly errors are collected by running the TPSS for 3 replication cycles for a 20x25 size target pattern. In these simulations, G_{mc} is fixed at 16 and G_{se} varies from 8.4 to 16 with an interval of 0.2.

fraction of erroneous tiles in a self-replicated pattern increases with its size, as shown in Figure 7.23. Errors in the patterns become very large, as G_{se} approaches the G_{mc}.

Time required and accuracy of the self-replication task in the kinetic model depends on the physico-chemical parameters (G_{mc} and G_{se}) of self-assembly. A phase diagram of the self-replication dynamics observed from the simulations using the kTPSM is illustrated in Figure 7.24. Pattern self-replicator works reliably in the region ($1.45G_{se} \leq G_{mc} \leq 1.9G_{se}$), shown in blue color. In the region $G_{mc} \leq 1.45G_{se}$, assembly errors get introduced in the self-replicating patterns (with G_{se} approaching G_{mc}, errors increase swiftly). For $G_{mc} \geq 1.90G_{se}$, the self-replicator becomes intractable, no patterns are produced.

7.9 Summary

To self-replicate 2-D rectangular patterns, we proposed a minimal self-replication system of DNA tiles under the aTAM framework. The replication mechanism is based on a cross-coupled cycle, where an L-shaped seed of the desired pattern is replicated and the remaining pattern grows in parallel. The self-replicator is implemented with the help of DX-tiles and SWitching-enabled Tiles (SWET), which form the basis of an active tile assembly process where structures are dynamically assembled and disassembled with

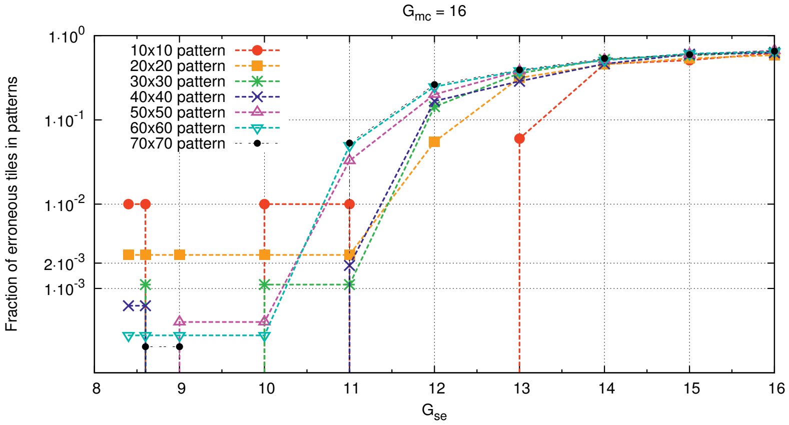


Figure 7.23: Simulation results of assembly errors in self-replicating patterns of different sizes. For each pattern size, the simulation parameters are: number of replication cycles =2; G_{mc} = 16; G_{se} is varied from 8.4 to 16.

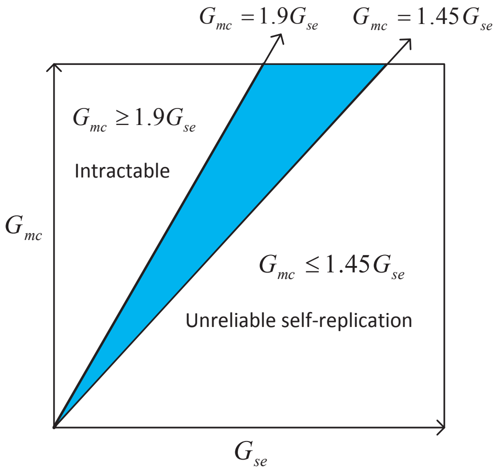


Figure 7.24: Phase diagram of tile pattern self-replicator dynamics using kinetic model. The blue area surrounded by $G_{mc} = 1.9G_{mc}$ and $G_{mc} = 1.45G_{se}$ represents the region in which a tile pattern self-replicator is both tractable and produces patterns with minimum assembly errors.

the help of an autonomous chemical oscillator that can be implemented using DNA strand displacement cascades.

We developed a simulation framework of the self-replication system of DNA-tile patterns. The self-replicator is based on a cross-coupled model, where both pattern copies and assembled mold copies serve as seeds to further assemble copies of each other. There are total three sets of tiles used in the design of the pattern self-replicator: one set of tiles self-assemble the target patterns using mold as a seed, and other two sets of tiles self-assemble molds of the patterns. Each set consists of four types of tiles, where each tile implements an XOR logic between its designated inputs and outputs.

The simulator is programmable, i.e. a user can initialize a target pattern for self-replication out of a variety of different patterns of the same size. A further programmability in the simulator can be added by extending it to include the self-replication of tile patterns that are assembled using a variety of other types of tiles, e.g. OR, AND, NAND, and XNOR tile sets. Currently, the simulator does not consider any chemical kinetics for the dissociation of pattern-mold complexes. Future work is expected to include this aspect into the self-replicator so as to further analyze its role in the dynamics of the pattern self-replication.

This work is a step forward in the direction of programmable two-dimensional nanostructure self-replication. It may be interesting to apply such self-replication for the generation of algorithmic programmable checkerboard configurations discussed in [71] (proposed design consists of a static checkerboard having an assembled array of alternating tiles). Self-replicating algorithmic programmable checkerboard arrays could produce interesting patterns together with self-assembling floating tiles.

One of the challenges in the experimental implementation of the pattern self-replication would be to prevent assembly errors [156], which are cumulative in the tile self-assembly process. Error-correction tiles, such as Enveloped Tiles and mechanism proposed in [60] can be applied to increase the reliability of self-replicated tile patterns. A reliable self-replicator with error levels not exceeding a minimum threshold may further open up new directions for investigation of fundamental principles behind reproduction and selection-driven evolution.

8 Conclusion and Future Work

8.1 Contributions

The highlights of this thesis are: 1) an error-suppressing mechanism and the design of ET for a reliable self-assembly of DNA tiles; 2) design of a system for self-replication of algorithmically self-assembled pattern of tiles. To suppress tile self-assembly errors, which are introduced due to cooperative binding of tiles, we use inspiration from nature. There are numerous examples of self-assembly in nature, where conformational switching plays a crucial role in error-free assembly or folding of macromolecules. For example, protein chaperones assist other proteins to fold correctly into their appropriate 3D structures. Although the mechanism behind the chaperones-assisted assembly is not yet well understood, the general process of protection and self-triggered deprotection can be applied for the design of dynamically controlled multi-component self-assembly systems. The error-suppression mechanism studied in this thesis uses protection assisted dynamic control to guide the self-assembly of tiles. Each tile initially stays protected, except the seed structure that is unprotected. One advantage of such protected tiles is their immunity to form spurious assemblies in the absence of a seed. Once the seed is introduced, the protected tiles can dock to the seed. During the docking, a matching protected tile is favoured against a mismatching protected tile. After successful docking, the matching protected tile releases its protection element, while its base element gets assembled with the engaged site.

We introduce a design ET that consists of two elements: an original DX DNA molecular tile, termed BT and an especially designed PT. In ET, the PT is integrated with the BT, and provides a dynamic control during local interactions of tiles. Before the self-assembly starts, all the tiles except seed structure (or a single seed tile) are introduced as ETs. However, during self-assembly, only a correctly matching tile would undergo a self-triggered activation that releases the PT and the BT gets assembled. The self-triggered activation process involves a two step upstream strand displacement by which the PT is released from the BT during the assembly process that activates the tile for a reliable downstream self-assembly reaction at the vacant site, thus enabling control over the local interactions between tiles. The two-step strand displacement includes an inter-tile bimolecular strand displacement followed by an intra-tile unimoleuclar strand displacement. The bimolecular and unimolecular strand displacement reactions are triggered as ET docks at a vacant site. The ET demonstrates effective error suppression over the tile self-assembly process. Using a widely studied example of Sierpinski pattern assembly, we demonstrate how a set of tiles can be easily converted into an equivalent set of ETs for reliable self-assembly. Thus, the approach can be easily adopted to convert any given tile set into a reliable tile set.

Designing such protected tiles using DNA structures is challenging in many ways: 1) There are limited design choices due to constraints of topologies of known DNA structures; 2) There are multiple competing requirements for a rational design of DNA sequences for such DNA structures; 3) It is difficult to establish the thermodynamic feasibility based on free energy calculations of these structures, as some of the DNA topologies involve tertiary DNA structures (DNA pseudo-knots and triplexes) whose geometries and physical parameters are not yet fully understood.

We present a design of a minimal system of self-replication for 2-D rectangular patterns within the framework of the aTAM [154]. The design adheres to simplicity and implementation feasibility in four aspects: 1) DX tiles are used; 2) all glues are of strength 1; 3) tiles do not carry signals; 4) the replication process is enzyme free. Pattern replication starts with formation of a mold structure around the “L”-shaped seed with the help of a set of SWitch-Enabled Tiles (SWET) that can be activated to switch their binding state from bound (ON) to free (OFF). Further, the assembled mold gets dissociated from the seed structure by a toehold-mediated switching control, which is cyclically triggered at precise time intervals. The dissociated mold structure grows a new copy of the “L”-shaped seed while the dissociated seed structure reiterates the process. The remaining pattern is grown on these self-replicating seed structures by supplying the system with an appropriate set of pattern forming tiles. The terms seed (“L”-shaped seed) and pattern have been interchangeably used in the rest of the article.

To study the dynamics of tile pattern self-replication, I developed a simulator in Java. Similar to the Winfree’s Xgrow simulator, which simulates the tile self-assembly process from a single tile seed, the self-replicator simulator allows to simulate the tile self-assembly process from a given L-shaped seed and XOR tiles. The simulator comes in two versions that are based on the underlying models of tile self-assembly, the aTAM and kTAM, respectively. The simulator allows to define target pattern size and desired number of replication cycles to simulate the dynamics of pattern self-replication. Currently, the simulation output can be seen in the web-version (using WAR files) of the Java IDE console. The current version also allows to run the simulation from command line using JAR files. Although the current version of the simulator allows to simulate, analyze, and visualize the pattern self-replication, a Javascript version of the simulator is under construction that will be ready soon. The simulator can serve as a useful tool for further research in the tile self-assembly and programmable tile pattern self-replication.

8.2 Future Work

8.2.1 A Few More Mechanisms of Dynamically Controlled Tiles

In this section we envision a few more designs of dynamically controlled molecular tile structures. In an integrated reaction systems of toehold-mediated strand displacement cascades and original molecular tile structures, these dynamically controlled tiles can enable a variety of features in the tile self-assembly. The dynamic control mechanisms implemented by new molecular tile structures include: tile deactivation, tile annihilation, tile reconfiguration.

Tile Deactivation

To design a complex self-assembly process using tiles, it is often desirable to suppress possible spurious interactions of tiles. To avoid undesired interactions, tiles can be dynamically deactivated for the the time being they are not required to be active. Tiles can be deactivated by protecting the protruding sticky ends of active tiles, as shown in Figure 8.1. Using an integrated dynamic reaction system of DNA molecules, such protecting strands (PS) can be produced to deactivate an active tile (AT) dynamically. Observe that the sticky ends of a deactivated tile are completely covered, thus it can not interact with other tiles, a contrast to the partially protected ETs, which could still interact to legitimate tiles. A deactivated tile can be activated back by removing its PS, and thus multiple tiles can be made to co-exit and used on demand.

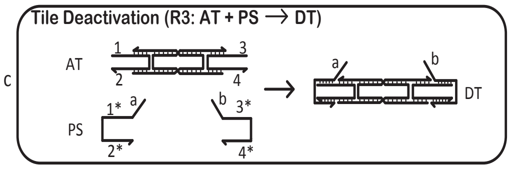


Figure 8.1: Dynamic deactivation of tile

Tile Annihilation

Tiles that are no longer needed in a reaction system can be destroyed so as to prevent any spurious interference with the system. To annihilate a tile, as shown in Figure 8.2, its sticky ends (SS) species can be removed from the tile. Using an integrated dynamic reaction system, Annihilating Strands (AS) can be produced dynamically that take away the SS and thus, destroy the tile. But the core structure of a destroyed tile can be reused to create new tile, as described below.

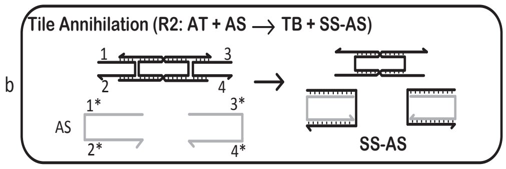


Figure 8.2: Dynamic annihilation of tile

Tile Reconfiguration

Self-assembly of original tiles is inherently static in nature, that is, tiles once assembled can not be reused or reconfigured so as to implement another system in the same medium. In a two-step process, as illustrated in Figure 8.3, tiles can be reconfigured. First, tiles are annihilated to release their backbone structures; second, a new set of SS species are introduced by an integrated dynamic reaction system. Thus, the two step process would ultimate produce the new reconfigured tile with new sticky ends.

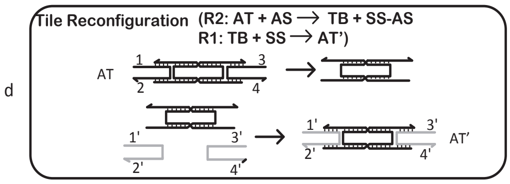


Figure 8.3: Dynamic deactivation of tile

8.2.2 An Engineering Approach for the Study of Dynamically Controlled Tile Self-assembly

Combining dynamically controlled molecular tile structures with DNA-based dynamic reaction systems introduces complex physicochemical dynamics into the assembly process. Therefore, the assembly process of dynamically controlled tiles can not be studied in the way static tiling systems are modelled. Future work for dynamically controlled tile self-assembly systems may be to develop an engineering approach for the design, implementation, and analysis of such tiling systems. A possible route to such an approach is illustrated in Figure 8.4. From bottom to top: 1) individual tile structures are designed rationally; 2) physicochemical characteristics of each building block and its assembly dynamics are studied either by experiments or using molecular dynamics simulations; 3) After detailed study of characteristics at individual component level, behavior of components is studied at the system level; 4) After studies at both individual component level and system level, realistic physicochemical parameters are extracted that can be used to study the behaviour of dynamic tiling systems implementing basic selection protocols; 5) In a long term, such approach would enable dynamically controlled self-assembly systems that may lead us to the engineering of generic tiling systems to study basic principles of selection and directed evolution at the molecular scale but perhaps in more controllable, observable, and programmable settings with more design choices!

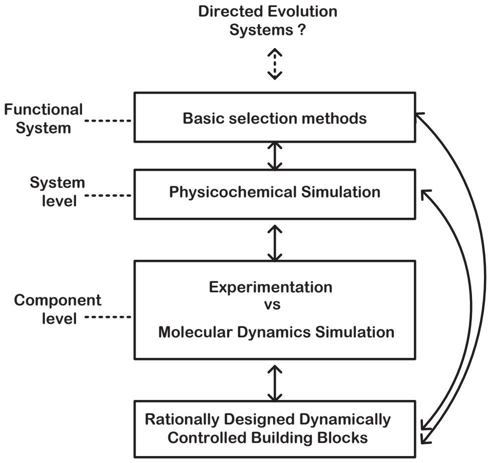

Directed Evolution Systems ?
Functional System
Basic selection methods
System level
Physicochemical Simulation
Component level
Experimentation
vs
Molecular Dynamics Simulation
Rationally Designed Dynamically Controlled Building Blocks

Figure 8.4: An engineering approach for the design and implementation of dynamically controlled tiling systems

8.2.3 Improving Reliability of Tile Pattern Self-replicator using Enveloped Tiles

Under the aTAM, tile assembly is irreversible and error-free. However, in reality, neither of these two assumptions are true. Biophysics modelling and experimental results suggest that tile assembly is a reversible process where at least two bonds (temperature 2 assembly) are required for a tile's stable assembly. Assembly error occurs when a tile binds by fewer than two bonds, and fails to detach before it gets trapped due to binding provided by other neighbouring tiles [156, 65]. This type of error can further be divided in two subtypes — growth errors [156] and facet errors [65]. Growth errors occur when a tile gets trapped at a vacant site of a growing assembly despite one of its bonds does not match correctly. Facet errors occur when a mismatched tile at a growing surface fails to detach quickly and eventually gets fixed by subsequent tiles. A second source of assembly errors is spontaneous nucleation [124, 126] where larger assemblies start growing even in the absence of the seed tile/structure.

Pattern self-replicator is prone to all these three types of assembly errors — Facet errors and growth errors occur during the self-assembly of mold and seed structures, and nucleation errors occur due to spuriously emerging undesired structures that may come under replication cycle. Moreover, due to repetitive cross-catalytic cycles these errors are multiplicative in nature. Therefore, in order to achieve a reliable self-replication in the proposed framework, maintaining error level below a minimum threshold is an important design criteria.

An immediate future work to the design of self-replicator framework may be the use of ETs. ETs are especially designed protected tiles that have already been demonstrated for growth error prevention in algorithmic tile self-assembly. Furthermore, due to protected nature of these tiles, initiating any nucleation growth in the absence of seed tile/structure would be energetically unfavourable. However, in the presence of seed, ETs joining the assembly undergo de-protection by a self-triggered activation mechanism enabling them for further binding of subsequent tiles. Therefore, ETs can also prevent nucleation errors happening in the replication process. Although it is obvious that ET may replace the DX-tiles in the proposed self-replicator design, it remains to be further explored for a quantitative analysis of error prevention and further, if an error threshold for self-replicator could be derived in the ET medium.

8.2.4 Spontaneous Emergence and Selection Using Tile Pattern Self-replicator

Let $\mathcal{P} = \{P_1, P_2, ..., P_m\}$ be the finite set of patterns, each of them being a potential subject for replication: a multiplexed pattern self-replication system. We can suppose, without loss of generality, that all the patterns have equal height, but variable lengths; otherwise we just complete the pattern using a special tile up to the desired height. We can construct a finite collection of tiles, such that by inserting within this system one (or several) of the pattern-types in $\mathcal{P}$, these structures would act as a template and derive only the replication of the chosen pattern (or patterns).

In order to construct an approximate minimal tile set that can uniquely self-assemble the self-replicating patterns in the multiplexed system, the Pattern self-Assembly Tile set Synthesis (PATS) algorithms [57] can be applied. The PATS algorithms have recently been used to determine a set of coloured tiles such that starting from an "L"-shaped bordering seed structure, the tiles would self-assemble into a given rectangular coloured pattern [57].

To apply the PATS, it would be required that all of these patterns have on their lower left corner position, a unique red tile that does not appear in any other positions inside the patterns. A joint pattern (*Master Pattern*), containing a bordered version of three patterns ($\mathcal{P} = \{P_1, P_2, P_3\}$),is shown in Figure 8.5. By applying the PATS search algorithm on $patt\mathcal{P}$ and then removing all the "grey" tiles, we can obtain the minimal PTS for the multiplexed self-replicating system. Note that in the worst case scenario, this set of tiles consists of at most as many elements as the disjoint union of all corresponding PATS solutions for each of the individual P_i patterns.

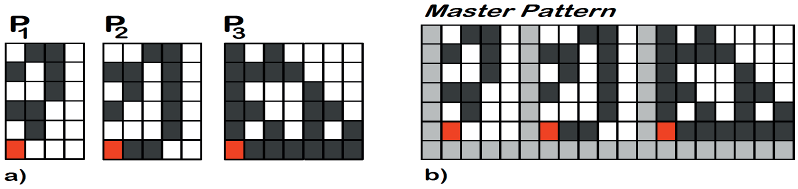


Figure 8.5: A set of patterns using PATS.(a) A set of three patterns subjected to the replication process; (b) The *master pattern* created by the concatenation of the bordered patterns.

Using such minimal PTS, interesting self-replication behaviours of tile patterns can be studied in the tile pattern self-replication framework. For example, in the multiplexed self-replication, the yield of different self-replicating patterns would depend on the concentrations of the CST(s) present in the self-replication medium. Further, the common tiles resource used in the multiplxed self-replication would be consumed more in the self-replication of the patterns that have higher self-replication rates, which may eventually dominate the self-replication space. Thus, a basic form of selection can be studied in the self-replication framework.

Bibliography

[1] cadnano. `http://cadnano.org/`.

[2] canDo. `http://cando-dna-origami.org/`.

[3] DNA sequence design. `http://www.dna.caltech.edu/DNAdesign/`.

[4] NUPACK. `http:http://www.nupack.org/`.

[5] XGROW. `http://www.dna.caltech.edu/Xgrow/`.

[6] Abel, Z., Benbernou, N., Damian, M., Demaine, E. D., Demaine, M. L., Flatland, R. Y., Kominers, S. D., and Schweller, R. T. (2010). Shape replication through self-assembly and rnase enzymes. In *SODA*, pages 1045–1064. SIAM.

[7] Abeliovich, H. (2005). An empirical extremum principle for the hill coefficient in ligand-protein interactions showing negative cooperativity. *Biophysical Journal*, 89(1):76 – 79.

[8] Abelson et al, H. (1999). Amorphous computing. Technical report, Massachusetts Institute of technology.

[9] Adamatzky, A. (2013). Slimeware: Engineering devices with slime mold. *Artificial Life*, 19(3-4):317–330.

[10] Adleman, L. (1994). Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024.

[11] Adleman, L., Cheng, Q., Goel, A., and Huang, M.-D. (2001). Running time and program size for self-assembled squares. In *Symposium on Theory of Computing (STOC), New York,*, pages 740–748.

[12] Amos, M., Axmann, I. M., Blüthgen, N., de la Cruz, F., Jaramillo, A., Rodriguez-Paton, A., and Simmel, F. (2015). Bacterial computing with engineered

populations. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373(2046).

[13] Arbib, M. A. (1966). Simple self-reproducing universal automata. *Information and Control*, 9(2):177 – 189.

[14] Barish, R. D., Rothemund, P. W. K., and Winfree, E. (2005). Two computational primitives for algorithmic self-assembly: Copying and counting. *Nano Letters*, 5(12):2586–2592.

[15] Barish, R. D., Schulman, R., Rothemund, P. W. K., and Winfree, E. (2009). An information-bearing seed for nucleating algorithmic self-assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 106(15):6054–6059.

[16] Bath, J. and Turberfield, A. J. (2007). DNA nanomachines. *Nature Nanotechnology*, 2(5):275–284.

[17] Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., and Shapiro, E. (2004). An autonomous molecular computer for logical control of gene expression. *Nature*, 429(6990):423–429.

[18] Burks, A. W., editor (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA.

[19] Byl, J. (1989). Self-reproduction in small cellular automata. *Physica D: Nonlinear Phenomena*, 34(1):295 – 299.

[20] Cairns-Smith, A. (1987). *Genetic Takeover: And the Mineral Origins of Life*. Cambridge University Press.

[21] Chakraborty, B., Jonoska, N., and Seeman, N. C. (2012). A programmable transducer self-assembled from DNA. *Chem. Sci.*, 3:168–176.

[22] Chen, H. L. and Goel, A. (2005). Error free self-assembly using error prone tiles. *DNA Computing*, 3384:62–75.

[23] Chen, H. L., Schulman, R., Goel, A., and Winfree, E. (2007a). Reducing facet nucleation during algorithmic self-assembly. *Nano Letters*, 7(9):2913–2919.

[24] Chen, H. L., Schulman, R., Goel, A., and Winfree, E. (2007b). Reducing facet nucleation during algorithmic self-assembly. *Nano Lett.*, 7(9):2913–2919.

[25] Chen, J. and Seeman, N. C. (1991). Synthesis from DNA of a molecule with the connectivity of a cube. *Nature*.

[26] Chenxiang, L., Yan, L., and Hao, Y. (2009). Designer DNA nanoarchitectures. *American Chemical Society*, 48(8):1663–1674.

[27] Chou, H.-H. and Reggia, J. A. (1997). Emergence of self-replicating structures in a cellular automata space. *Physica D: Nonlinear Phenomena*, 110(3-4):252 – 276.

[28] Chou, H.-H. and Reggia, J. A. (1998). Problem solving during artificial selection of self-replicating loops. *Physica D: Nonlinear Phenomena*, 115(3-4):293 – 312.

[29] Chrystopher, L. N. and Hertfordshire, A. A. (2002). Self-reproduction in asynchronous cellular automata. pages 201–209. MIT Press.

[30] Codd, E. F. (1968). *Cellular Automata*. Academic Press, New York, USA.

[31] Conrad, M. (1985). On design principles for a molecular computer. *Commun. ACM*, 28(5):464–480.

[32] Cragnolini, T., Derreumaux, P., and Pasquali, S. (2013). Coarse-grained simulations of RNA and DNA duplexes. *The Journal of Physical Chemistry B*, 117(27):8047–8060.

[33] Czeizler, E. and Popa, A. (2013). Synthesizing minimal tile sets for complex patterns in the framework of patterned DNA self-assembly. *Theor. Comput. Sci.*, 499:23–37.

[34] Dietz, H., Douglas, S. M., and Shih, W. M. (2009). Folding DNA into twisted and curved nanoscale shapes. *Science*, 325(5941):725–730.

[35] Dirks, R. M. and Pierce, N. A. (2003). A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry*, 24(13):1664–1677.

[36] Dirks, R. M. and Pierce, N. A. (2004). Triggered amplification by hybridization chain reaction. *Proceedings of the National Academy of Sciences of the United States of America*, 101(43):15275–15278.

[37] Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries:a review. *Artificial Life*, 7(3):225–275.

[38] Douglas, S. M., Dietz, H., Liedl, T., Hogberg, B., Graf, F., and Shih, W. M. (2009). Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*, 459(7245):414–418.

[39] Downing, K. (2001). Modular designer chemistries for artificial life. In *Genetic and Evolutionary Computation Conference*.

[40] Eisenberg, H. and Felsenfeld, G. (1967). Studies of the temperature-dependent conformation and phase separation of polyriboadenylic acid solutions at neutral ph. *Journal of Molecular Biology*, 30(1):17 – 37.

[41] Ewaschuk, R. (2006). Self-replication and self-assembly for manufacturing. *Artif. Life*, 12:2006.

[42] Feldkamp, U., Rauhe, H., and Banzhaf, W. (2003). Software tools for DNA sequence design. *Genetic Programming and Evolvable Machines*, 4(2):153–171.

[43] Field, R. J. and Niyes, R. M. (1974). Oscillations in chemical systems iv. limit cycle behavior in a model of a real chemical reaction. *Journal of Chemical Physics*, 60:1877–1884.

[44] Flint, S. (2000). *Principles of Virology: Molecular Biology, Pathogenesis, and Control*. ASM Press.

[45] Freitas, R. A. and Merkle, R. C. (2004). *Kinematic Self-replicating Machines*. Landes, Austin, TX.

[46] Frenkel, D. and Smit, B., editors (1996). *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, Inc., Orlando, FL, USA, 1st edition.

[47] Fu, T. J. and Seeman, N. C. (1993). DNA double-crossover molecules. *Biochemistry*, 32(13):3211–3220.

[48] Fujibayashi, K., Hariadi, R., Park, S. H., Winfree, E., and Murata, S. (2008). Toward reliable algorithmic self-assembly of DNA tiles: a fixed-width cellular automaton pattern. *Nano Lett*, 8(7):1791–7.

[49] Fujibayashi, K., Zhang, D. Y., Winfree, E., and Murata, S. (2009). Error suppression mechanisms for DNA tile self-assembly and their simulation. *Natural Computing*, 8(3):589–612.

[50] Gautam, V. K., Czeizler, E., Haddow, P. C., and Kuiper, M. (2014). Design of a minimal system for self-replication of rectangular patterns of DNA tiles. In Dediu, A.-H., Lozano, M., and Martín-Vide, C., editors, *Theory and Practice of Natural Computing: Third International Conference, TPNC 2014, Granada, Spain, December 9-11, 2014. Proceedings*, pages 119–133, Cham. Springer International Publishing.

[51] Gautam, V. K., Haddow, P. C., and Kuiper, M. (2013). Reliable self-assembly by self-triggered activation of enveloped DNA tiles. In Dediu, A.-H., Martín-Vide, C., Truthe, B., and Vega-Rodríguez, M. A., editors, *Theory and Practice of Natural Computing: Second International Conference, TPNC 2013, Cáceres, Spain, December 3-5, 2013, Proceedings*, pages 68–79, Berlin, Heidelberg. Springer Berlin Heidelberg.

[52] Genot, A. J., Zhang, D. Y., Bath, J., and Turberfield, A. J. (2011). Remote toehold: A mechanism for flexible control of DNA hybridization kinetics. *Journal of the American Chemical Society*, 133(7):2177–2182.

[53] Gerhardt, M., Schuster, H., and Tyson, J. J. (1990). A cellular automation model of excitable media including curvature and dispersion. *Science*, 247(4950):1563–1566.

[54] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

[55] Gonzalez, H. C., Darre, L., and Pantano, S. (2013). Transferable mixing of atomistic and coarse-grained water models. *The Journal of Physical Chemistry B*, 117(46):14438–14448.

[56] Goodman, R. P., Schaap, I. A. T., Tardin, C. F., Erben, C. M., Berry, R. M., Schmidt, C. F., and Turberfield, A. J. (2005). Rapid chiral assembly of rigid DNA building blocks for molecular nanofabrication. 310(5754):1661–1665.

[57] Göös, M., Lempiäinen, T., Czeizler, E., and Orponen, P. (2014). Search methods for tile sets in patterned DNA self-assembly. *J. Comput. Syst. Sci.*, 80(1):297–319.

[58] Griffith, S., Goldwater, D., and Jacobson, J. M. (2005). Robotics: Self-replication from random parts. *Nature*, 437:636.

[59] Grzybowski, B. A., Wilmer, C. E., Kim, J., Browne, K. P., and Bishop, K. J. M. (2009). Self-assembly: from crystals to cells. *Soft Matter*, 5(6):1110–1128.

[60] Gu, H., Chao, J., Xiao, S.-J., and Seeman, N. C. (2009). Dynamic patterning programmed by DNA tiles captured on a DNA origami substrate. *Nature nanotechnology*, 4(4):245–248.

[61] Hamada, S. and Murata, S. (2009). Substrate-assisted assembly of interconnected single-duplex DNA nanostructures. *Angewandte Chemie-International Edition*, 48(37):6820–6823.

[62] Harding, S. and Miller, J. F. (2005). Evolution in materio: Evolving logic gates in liquid crystal. In *Proc. Eur. Conf. Artif. Life (ECAL 2005),Workshop on Unconventional Computing: From cellular automata to wetware*, pages 133–149.

[63] Harrison, R. M., Romano, F., Ouldridge, T. E., Louis, A. A., and Doye, J. (2014). Coarse-grained modelling of strong DNA bending ii: Cyclization.

[64] Hinckley, D. M., Lequieu, J. P., and de Pablo, J. J. (2014). Coarse-grained modeling of DNA oligomer hybridization: Length, sequence, and salt effects. *The Journal of Chemical Physics*, 141(3).

[65] Ho-Lin, C. and Goel, A. (2005). Error free self-assembly using error prone tiles. In *In DNA Computing 10, volume 3384 of LNCS*, pages 1–11.

[66] Hoops, S. and et al. (2006). COPASI - a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074.

[67] Hutton, T. J. (2002). Evolvable self-replicating molecules in an artificial chemistry. *Artificial Life*, 8:341–356.

[68] Jean-Yves Perrier, Moshe Sipper, J. Z. (1996). Toward a viable, self-reproducing universal computer. *Physica D: Nonlinear Phenomena*, 97(4):335 – 352.

[69] Jiang, S., Yan, H., and Liu, Y. (2014). Kinetics of DNA tile dimerization. *ACS Nano*, 8(6):5826–5832.

[70] Jonoska, N., Liao, S., and Seeman, N. C. (2004). *Transducers with Programmable Input by DNA Self-assembly*, pages 219–240. Springer Berlin Heidelberg.

[71] Jonoska, N. and Seeman, N. C. (2015). Molecular ping-pong game of life on a two-dimensional DNA origami array. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373(2046).

[72] Joyce, G. F. (2002). The antiquity of RNA-based evolution. *Nature Publishing Group*, 418(0028-0836):214–221.

[73] K., R. P. W. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature*, pages 297–302.

[74] Karsenti, E. (2008). Self-organization in cell biology: a brief history. *Nat Rev Mol Cell Biol*, 9(3):255–262.

[75] Keenan, A., Schweller, R. T., and Zhong, X. (2013). Exponential replication of patterns in the signal tile assembly model. In *DNA19*, volume 8141 of *Lecture Notes in Computer Science*, pages 118–132. Springer.

[76] Korolev, N., Luo, D., Lyubartsev, A. P., and NordenskiÃűld, L. (2014). A coarse-grained DNA model parameterized from atomistic simulations by inverse monte carlo. *Polymers*, 6(6):1655–1675.

[77] Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., and Kollman, P. A. (1992). The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of Computational Chemistry*, 13(8):1011–1021.

[78] LaBean, T. H., Yan, H., Kopatsch, J., Liu, F., Winfree, E., Reif, J. H., and Seeman, N. C. (2000). Construction, analysis, ligation, and self-assembly of DNA triple crossover complexes. *Journal of the American Chemical Society*, 122(9):1848–1860.

[79] Langton, C. G. (1984). Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):135 – 144.

[80] Liu, W., Zhong, H., Wang, R., and Seeman, N. C. (2011). Crystalline two-dimensional DNA-origami arrays. *Angewandte Chemie International Edition*, 50(7082):264–267.

[81] Lohn, J. D. and Reggia, J. A. (1997). Automatic discovery of self-replicating structures in cellular automata. *IEEE Transactions on Evolutionary Computation*, 1(3):165–178.

[82] Maffeo, C., Yoo, J., Comer, J., Wells, D. B., Luan, B., and Aksimentiev, A. (2014). Close encounters with DNA. *J Phys Condens Matter*, 26:413101.

[83] Majumder, U., Labean, T. H., and Reif, J. H. (2007). Activatable tiles: Compact, robust programmable assembly and other applications. In *in DNA Computing: DNA13*, pages 15–25. LNCS.

[84] Manning, G. S. (1988). Three persistence lengths for a stiff polymer with an application to DNA B-Z junctions. *Biopolymers*, 27(10):1529–1542.

[85] Mateus, A. M., Gorfinkiel, N., and Arias, A. M. (2009). Origin and function of fluctuations in cell behaviour and the emergence of patterns. *Seminars in Cell and Developmental Biology*, 20(7):877–884.

[86] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

[87] Miller, J. F. and Downing, K. (2002). Evolution in materio: Looking beyond the silicon box. *2002 NASA/DoD Conference on Evolable Hardware, Proceedings*, pages 167–176.

[88] Minero, G. A. S., Wagler, P. F., Oughli, A. A., and McCaskill, J. S. (2015). Electronic ph switching of DNA triplex reactions. *RSC Adv.*, 5:27313–27325.

[89] Mohammed, A. M. and Schulman, R. (2013). Directing self-assembly of DNA nanotubes using programmable seeds. *Nano Letters*, 13(9):4006–4013.

[90] Morrison, L. E. and Stols, L. M. (1993). Sensitive fluorescence-based thermodynamic and kinetic measurements of DNA hybridization in solution. *Biochemistry*, 32(12):3095–3104.

[91] Ono, N. and Ikegami, T. (1999). Model of self-replicating cell capable of self-maintenance. In *Proceedings of the 5th European Conference on Advances in Artificial Life*, ECAL '99, pages 399–406, London, UK. Springer-Verlag.

[92] Ouldridge, T. E., Louis, A. A., and Doye, J. P. (2010). DNA nanotweezers studied with a coarse-grained model of DNA. *Phys Rev Lett*, 104.

[93] Ouldridge, T. E., Louis, A. A., and Doye, J. P. (2011). Structural, mechanical, and thermodynamic properties of a coarse-grained DNA model. *J Chem Phys*, 134.

[94] Ouldridge, T. E., Sulc, P., Romano, F., Doye, J. P. K., and Louis, A. A. (2013). DNA hybridization kinetics: zippering, internal displacement and sequence dependence. *Nucleic Acids Research*, 41(19):8886–8895.

[95] Padilla, J., Liu, W., and Seeman, N. C. (2012). Hierarchical self assembly of patterns from the robinson tilings: DNA tile design in an enhanced tile assembly model. *Natural Computing*, 11(2):323–338.

[96] Padilla, J. E., Patitz, M. J., Pena, R., Schweller, R. T., Seeman, N. C., Sheline, R., Summers, S. M., and Zhong, X. (2013). Asynchronous signal passing for tile self-assembly: Fuel efficient computation and efficient assembly of shapes. In *UCNC*, pages 174–185.

[97] Padilla, J. E., Sha, R., Kristiansen, M., Chen, J., Jonoska, N., and Seeman, N. C. (2015). A signal-passing DNA strand exchange mechanism for active self-assembly of DNA nanostructures. *Angew Chem Int Ed Engl*.

[98] Pankau, W.-M., Antsypovich, S., Eckardt, L., Stankiewicz, J., Monninghoff, S., Zimmermann, J., Radeva, M., and von Kiedrowski, G. (2005). *Replicable Nanoscaffolded Multifunctionality – A Chemical Perspective*. Wiley-VCH Verlag GmbH and Co. KGaA.

[99] Panyutin, I. and Hsieh, P. (1994). The kinetics of spontaneous DNA branch migration. *PNAS*, 91:2021–2025.

[100] Patzke, V. and von Kiedrowski, G. (2007). Self-replicating systems. *ARKIVOC*, pages 293–310.

[101] Paul N., J. G. (2004). Minimal self-replicating systems. *Current Opinion in Chemical Biology*, 8(6):634 – 639.

[102] Penrose, L. S. (1958). Mechanics of self-reproduction. *Annals of Human Genetics*, 23(1):59–72.

[103] Penrose, L. S. (1959). Self-reproducing machines. *Scientific American*, 200(6):105–114.

[104] Pethrick, R. A. (1988). The theory of polymer dynamics. *British Polymer Journal*, 20(3):299 – 299.

[105] Qian, L. and Winfree, E. (2011a). Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201.

[106] Qian, L. and Winfree, E. (2011b). A simple DNA gate motif for synthesizing large-scale circuits. *Journal of the Royal Society Interface*, 8(62):1281–1297.

[107] Quartin, R. S. and Wetmur, J. G. (1989). Effect of ionic strength on the hybridization of oligodeoxynucleotides with reduced charge due to methylphosphonate linkages to unmodified oligodeoxynucleotides containing the complementary sequence. *Biochemistry*, 28(3):1040–1047.

[108] Reggia, J. A., Armentrout, S. L., Chou, H.-H., and Peng, Y. (1993). Simple systems that exhibit self-directed replication. *Science*, 259(5099):1282–1287.

[109] Reif, J. H. (1999). Local parallel biomolecular computing. In *DNA Based Computers III, volume 48 of DIMACS*, pages 217–254. American Mathematical Society.

[110] Rivas, E. and Eddy, S. R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots1. *Journal of Molecular Biology*, 285(5):2053 – 2068.

[111] Romano, F., Chakraborty, D., Doye, J. P. K., Ouldridge, T. E., and Louis, A. A. (2013). Coarse-grained simulations of DNA overstretching. *Journal of Chemical Physics*, 138(8).

[112] Rothemund, P. W. K., Papadakis, N., and Winfree, E. (2004). Algorithmic self-assembly of DNA sierpinski triangles. *PLoS Biol*, 2(12):e424.

[113] Rothemund, P. W. K. and Winfree, E. (2000). The program-size complexity of self-assembled squares. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC'00, pages 459–468. ACM.

[114] Sadowski, J. P., Calvert, C. R., Zhang, D. Y., Pierce, N. A., and Yin, P. (2014). Developmental self-assembly of a DNA tetrahedron. *Acs Nano*, 8(4):3251–3259.

[115] SantaLucia, J. (1998). A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc Natl Acad Sci USA*, 95(4):1460–1465.

[116] Savelyev, A. and Papoian, G. A. (2010). Chemically accurate coarse graining of double-stranded DNA. *Proceedings of the National Academy of Sciences*, 107(47):20340–20345.

[117] Sayama, H. (1998). Introduction of structural dissolution into langton's self-reproducing loop. In *Proceedings of the Sixth International Conference on Artificial Life*, ALIFE, pages 114–122, Cambridge, MA, USA. MIT Press.

[118] Sayama, H. (1999). A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. *Artif. Life*, 5(4):343–365.

[119] Schiefer, N. and Winfree, E. (2015). Universal computation and optimal construction in the chemical reaction network-controlled tile assembly model. In *DNA Computing and Molecular Programming: 21st International Conference, DNA 21, Boston and Cambridge, MA, USA, August 17-21, 2015. Proceedings*. Springer International Publishing.

[120] Schulman, R. (2005). Self-replication and evolution of DNA crystals. In *Advances in Artificial Life: 8th European Conference (ECAL), volume LNCS 3630*, pages 734–743. Springer-Verlag.

[121] Schulman, R. and Doty, D. (2015). Designing ordered nucleic acid self-assembly processes. *Current Opinion in Structural Biology*, 31:57–63.

[122] Schulman, R. and Winfree, E. (2005a). Programmable control of nucleation for algorithmic self-assembly. In *in DNA Computing 10, Lecture Notes in Comput. Sci. 3384*, pages 319–328. Springer-Verlag.

[123] Schulman, R. and Winfree, E. (2005b). Programmable control of nucleation for algorithmic self-assembly (extended abstract). *DNA Computing*, 3384:319–328.

[124] Schulman, R. and Winfree, E. (2007a). Synthesis of crystals with a programmable kinetic barrier to nucleation. *Proceedings of the National Academy of Sciences of the United States of America*, 104(39):15236–15241.

[125] Schulman, R. and Winfree, E. (2007b). Synthesis of crystals with a programmable kinetic barrier to nucleation. *Proceedings of the National Academy of Sciences*, 104(39):15236–15241.

[126] Schulman, R. and Winfree, E. (2009). Programmable control of nucleation for algorithmic self-assembly. *Siam Journal on Computing*, 39(4):1581–1616.

[127] Schulman, R., Wright, C., and Winfree, E. (2015). Increasing redundancy exponentially reduces error rates during algorithmic self-assembly. *Acs Nano*, 9(6):5760–5771.

[128] Schulman, R., Yurke, B., and Winfree, E. (2012). Robust self-replication of combinatorial information via crystal growth and scission. *Proceedings of the National Academy of Sciences*, 109(17):6405–6410.

[129] Seelig, G., Soloveichik, D., Zhang, D. Y., and Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. *Science*, 314(5805):1585–1588.

[130] Seeman, N. C. (1982). Nucleic-acid junctions and lattices. *Journal of Theoretical Biology*, 99(2):237–247.

[131] Seeman, N. C. (1990). De novo design of sequences for nucleic acid structural engineering. *Journal of Biomolecular Structure and Dynamics*, 8(3):573–581.

[132] Seeman, N. C. (2003). DNA in a material world. *Nature*, 421(6921):427–431.

[133] Shin, J. S. and Pierce, N. A. (2004). A synthetic DNA walker for molecular transport. *Journal of the American Chemical Society*, 126(35):10834–10835.

[134] Sievers, D. and von Kiedrowski, G. (1994). Self-replication of complementary nucleotide-based oligomers. *Nature*, 369:221–224.

[135] Simmel, F. C. and Yurke, B. (2001). Using DNA to construct and power a nanoactuator. *Phys. Rev. E*, 63:041913.

[136] Simmel, F. C. and Yurke, B. (2002). A DNA-based molecular device switchable between three distinct mechanical states. *Applied Physics Letters*, 80(5):883–885.

[137] Sipper, M. (1998). Fifty years of research on self-replication: An overview. *Artif. Life*, 4(3):237–257.

[138] Smith, A., Turney, P., (corresponding, P. T., and Ewaschuk, R. (2003). Self-replicating machines in continuous space with virtual physics. *Artificial Life*, 9:21–40.

[139] Smith, S. B., Cui, Y., and Bustamante, C. (1996). Overstretching b-DNA: The elastic response of individual double-stranded and single-stranded DNA molecules. *Science*, 271(5250):795–799.

[140] Soloveichik, D., Seelig, G., and Winfree, E. (2010). DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398.

[141] Srinivas, N., Ouldridge, T. E., Šulc, P., Schaeffer, J. M., Yurke, B., Louis, A. A., Doye, J. P. K., and Winfree, E. (2013). On the biophysics and kinetics of toehold-mediated DNA strand displacement. *Nucleic Acids Research*.

[142] Stepney, S. (2008). The neglected pillar of material computation. *Physica D-Nonlinear Phenomena*, 237(9):1157–1164.

[143] Stepney, S., Clark, J., Tyrrel, A., Johnson, C., Timmis, J., Partridge, D., Adamatsky, A., and Smith, R. (2003). Journeys in non-classical computing: A grand challenge for computing research. In *Grand Challenge Report 7*. University of Edinburg.

[144] Sulc, P., Romano, F., Ouldridge, T. E., Rovigatti, L., Doye, J. P., and Louis, A. A. (2012). Sequence-dependent thermodynamics of a coarse-grained DNA model. *J Chem Phys*, 137.

[145] Tempesti, G. (1995). A new self-reproducing cellular automaton capable of construction and computation. In *in ECAL95: Proceedings of the Third European Conference on Artificial*, pages 555–563. Springer-Verlag.

[146] Torrie, G. and Valleau, J. (1977). Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187 – 199.

[147] Vichniac, G. Y. (1984). Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):96 – 116.

[148] von Kiedrowski, G. (1986). A self-replicating hexadeoxynucleotide. *Angewandte Chemie International Edition in English*, 25(10):932–935.

[149] Wang, H. (1961). Proving theorems by pattern recognition II. *Bell System Technical Journal*, 40:1–42.

[150] Whitelam, S. and Geissler, P. L. (2007). Avoiding unphysical kinetic traps in monte carlo simulations of strongly attractive particles. *The Journal of Chemical Physics*, 127(15).

[151] Whitesides, G. M. and Grzybowski, B. (2002). Self-assembly at all scales. *Science*, 295(5564):2418–2421.

[152] Whitesides, G. M., Kriebel, J. K., and Mayers, B. T. (2005). *Self-Assembly and Nanostructured Materials*, pages 217–239. Springer US, Boston, MA.

[153] Williamson, J. R. (2008). Cooperativity in macromolecular assembly. *Nat Chem Biol*, 4(8):458–465.

[154] Winfree, E. (1998). *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology Pasadena, California, USA.

[155] Winfree, E. (2000). Algorithmic self-assembly of DNA: Theoretical motivations and 2d assembly experiments. *J Biomol Struct Dyn*, 17 Suppl 1:263–70.

[156] Winfree, E. and Bekbolatov, R. (2004). Proofreading tile sets: Error correction for algorithmic self-assembly. In *In DNA Based Computers 9, volume 2943 of LNCS*, pages 126–144.

[157] Winfree, E., Liu, F., Wenzler, L. A., and Seeman, N. C. (1998). Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394(6693):539–44.

[158] Winfree, E., Yang, X., and Seeman, N. (1999). Universal computation via self-assembly of DNA: Some theory and experiments. In *DNA based computers 2*, volume 2, pages 191–213. Amer Mathematical Society.

[159] Winfree, E., Yang, X., and Seeman, N. C. (1996). Universal computation via self-assembly of DNA: Some theory and experiments. In *DNA Based Computers II, volume 44 of DIMACS*, pages 191–213. American Mathematical Society.

[160] Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media.

[161] Yan, H., LaBean, T. H., Feng, L. P., and Reif, J. H. (2003). Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proceedings of the National Academy of Sciences of the United States of America*, 100(14):8103–8108.

[162] Yan, H., Zhang, X., Shen, Z., and Seeman, N. C. (2002). A robust DNA mechanical device controlled by hybridization topology. *Nature*.

[163] Yan, J. and Marko, J. F. (2004). Localized single-stranded bubble mechanism for cyclization of short double helix DNA. *Phys. Rev. Lett.*, 93:108108.

[164] Ye, T. and Chengde, M. (2004). Molecular gears:âĂŁ a pair of DNA circles continuously rolls against each other. *Journal of the American Chemical Society*, 126(37):11410–11411.

[165] Yin, P., Choi, H. M. T., Calvert, C. R., and Pierce, N. A. (2008). Programming biomolecular self-assembly pathways. *Nature*, 451(7176):318–U4.

[166] Yin, P., Turberfield, A. J., and Reif, J. H. (2005). *Designs of Autonomous Unidirectional Walking DNA Devices*, pages 410–425. Springer Berlin Heidelberg.

[167] Yu, H., Yi, C., Haipeng, L., Alexander, E. R., and Chengde, M. (2005). Self-assembly of hexagonal DNA two-dimensional (2d) arrays. *Journal of the American Chemical Society*, 127(35):12202–12203.

[168] Yurke, B., Turberfield, A. J., Mills, A. P., Simmel, F. C., and Neumann, J. L. (2000). A DNA-fuelled molecular machine made of DNA. *Nature*, 406(6796):605–608.

[169] Zadeh, J. N., Steenberg, C., Bois, J. S., Wolfe, B. R., Pierce, M. B., Khan, A. R., Dirks, R. M., and Pierce, N. A. (2011). Nupack: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173.

[170] Zhang, D. Y. (2011). Cooperative hybridization of oligonucleotides. *Journal of the American Chemical Society*, 133(4):1077–1086.

[171] Zhang, D. Y., Hariadi, R. F., Choi, H. M., and Winfree, E. (2013). Integrating DNA strand-displacement circuitry with DNA tile self-assembly. *Nat Commun*, 4:1965.

[172] Zhang, D. Y. and Seelig, G. (2011). Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chemistry*, 3(2):103–113.

[173] Zhang, D. Y., Turberfield, A. J., Yurke, B., and Winfree, E. (2007). Engineering entropy-driven reactions and networks catalyzed by DNA. *Science*, 318(5853):1121–1125.

[174] Zhang, D. Y. and Winfree, E. (2009). Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society*, 131(47):17303–17314.

[175] Zhang, D. Y. and Winfree, E. (2010). Robustness and modularity properties of a non-covalent DNA catalytic reaction. *Nucleic Acids Research*, 38(12):4182–4197.

[176] Zhang, Y. and Seeman, N. C. (1994). Construction of a DNA-truncated octahedron. *Journal of the American Chemical Society*, 116(5):1661–1669.

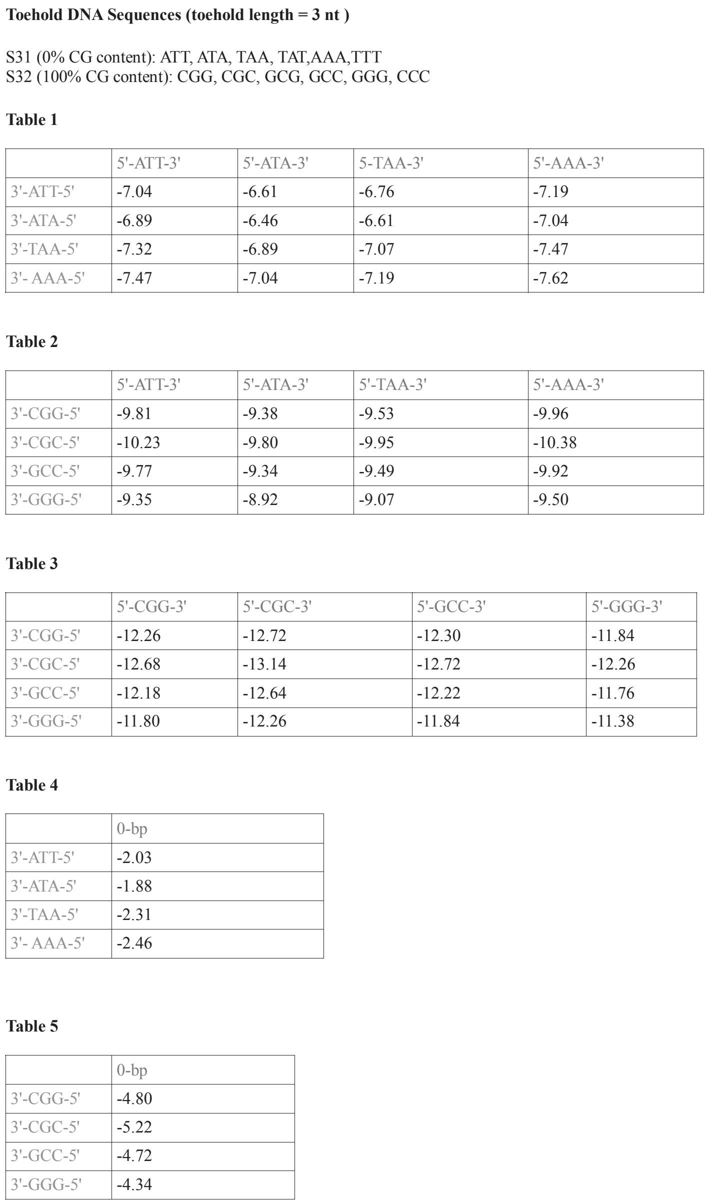
[177] Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res*, 31:3406–3415.

Appendices

A

Free Energy Calculation

A.1 Calculated DNA Toehold Binding Energies of Enveloped Tiles



Toehold DNA Sequences (toehold length = 3 nt)

S31 (0% CG content): ATT, ATA, TAA, TAT,AAA,TTT
S32 (100% CG content): CGG, CGC, GCG, GCC, GGG, CCC

Table 1

	5'-ATT-3'	5'-ATA-3'	5-TAA-3'	5'-AAA-3'
3'-ATT-5'	-7.04	-6.61	-6.76	-7.19
3'-ATA-5'	-6.89	-6.46	-6.61	-7.04
3'-TAA-5'	-7.32	-6.89	-7.07	-7.47
3'- AAA-5'	-7.47	-7.04	-7.19	-7.62

Table 2

	5'-ATT-3'	5'-ATA-3'	5'-TAA-3'	5'-AAA-3'
3'-CGG-5'	-9.81	-9.38	-9.53	-9.96
3'-CGC-5'	-10.23	-9.80	-9.95	-10.38
3'-GCC-5'	-9.77	-9.34	-9.49	-9.92
3'-GGG-5'	-9.35	-8.92	-9.07	-9.50

Table 3

	5'-CGG-3'	5'-CGC-3'	5'-GCC-3'	5'-GGG-3'
3'-CGG-5'	-12.26	-12.72	-12.30	-11.84
3'-CGC-5'	-12.68	-13.14	-12.72	-12.26
3'-GCC-5'	-12.18	-12.64	-12.22	-11.76
3'-GGG-5'	-11.80	-12.26	-11.84	-11.38

Table 4

	0-bp
3'-ATT-5'	-2.03
3'-ATA-5'	-1.88
3'-TAA-5'	-2.31
3'- AAA-5'	-2.46

Table 5

	0-bp
3'-CGG-5'	-4.80
3'-CGC-5'	-5.22
3'-GCC-5'	-4.72
3'-GGG-5'	-4.34

Table 6

	1-bp (A-T)
3'-ATT-5'	-4.85
3'-ATA-5'	-4.70
3'-TAA-5'	-5.13
3'- AAA-5'	-5.28

Table 7

	1-bp (G-C)
3'-ATT-5'	-5.13
3'-ATA-5'	-4.98
3'-TAA-5'	-5.41
3'- AAA-5'	-5.56

Table 8

	1-bp (A-T)
3'-CGG-5'	-7.62
3'-CGC-5'	-8.04
3'-GCC-5'	-7.54
3'-GGG-5'	-7.16

Table 9

	1-bp (G-C)
3'-CGG-5'	-7.90
3'-CGC-5'	-8.32
3'-GCC-5'	-7.82
3'-GGG-5'	-7.44

Table 10

	2-bp	
	AA	AT
3'-ATT-5'	-6.02	-5.88
3'-ATA-5'	-5.87	-5.73
3'-TAA-5'	-6.3	-6.16
3'- AAA-5'	-6.45	-6.31

Table 11

	2-bp	
	GG	GC
3'-ATT-5'	-7.1	-7.55
3'-ATA-5'	-6.95	-7.4
3'-TAA-5'	-7.38	-7.83
3'- AAA-5'	-7.53	-7.98

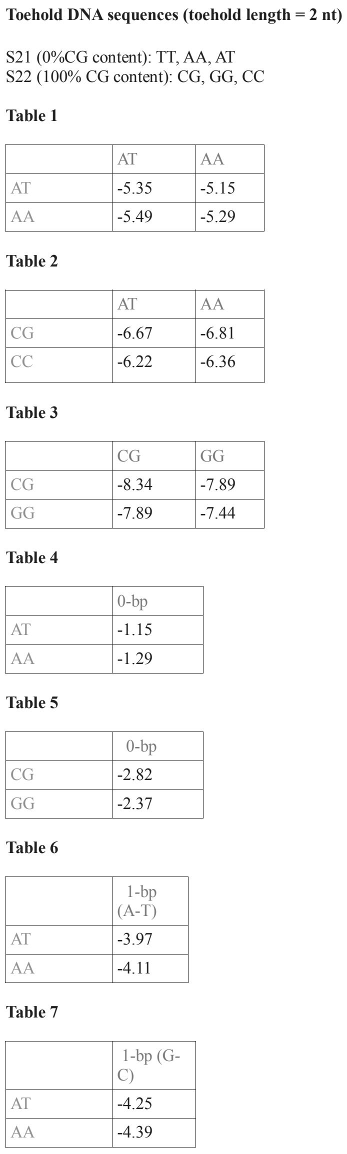
Table 12

	2-bp	
	AA	AT
3'-CGG-5'	-8.79	-8.65
3'-CGC-5'	-9.21	-9.07
3'-GCC-5'	-8.71	-8.57
3'-GGG-5'	-8.33	-8.19

Table 13

	2-bp	
	GG	GC
3'-CGG-5'	-9.87	-10.32
3'-CGC-5'	-10.29	-10.74
3'-GCC-5'	-9.79	-10.24
3'-GGG-5'	-9.41	-9.86

A.1. Calculated DNA Toehold Binding Energies of Enveloped Tiles



Toehold DNA sequences (toehold length = 2 nt)

S21 (0%CG content): TT, AA, AT
S22 (100% CG content): CG, GG, CC

Table 1

	AT	AA
AT	-5.35	-5.15
AA	-5.49	-5.29

Table 2

	AT	AA
CG	-6.67	-6.81
CC	-6.22	-6.36

Table 3

	CG	GG
CG	-8.34	-7.89
GG	-7.89	-7.44

Table 4

	0-bp
AT	-1.15
AA	-1.29

Table 5

	0-bp
CG	-2.82
GG	-2.37

Table 6

	1-bp (A-T)
AT	-3.97
AA	-4.11

Table 7

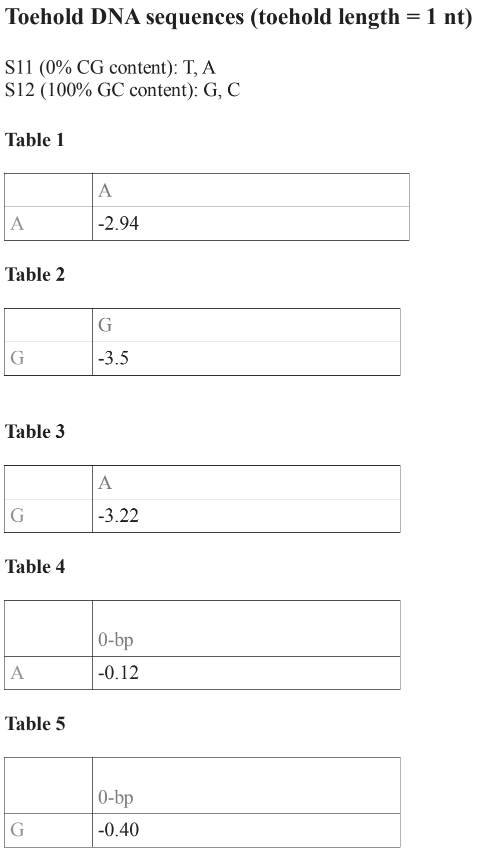
	1-bp (G-C)
AT	-4.25
AA	-4.39

Table 8

	1-bp (A-T)
CG	-5.64
GG	-5.19

Table 9

	1-bp (G-C)
CG	-5.92
GG	-5.47



Toehold DNA sequences (toehold length = 1 nt)

S11 (0% CG content): T, A
S12 (100% GC content): G, C

Table 1

	A
A	-2.94

Table 2

	G
G	-3.5

Table 3

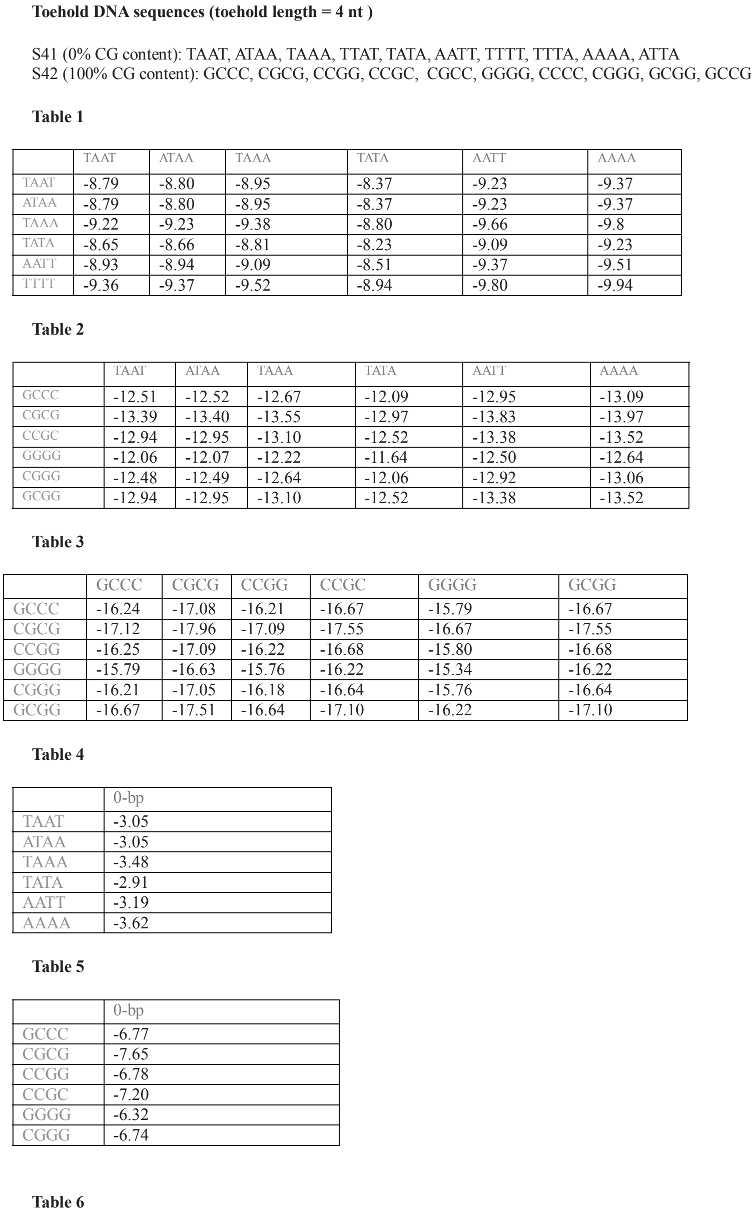
	A
G	-3.22

Table 4

	0-bp
A	-0.12

Table 5

	0-bp
G	-0.40



Toehold DNA sequences (toehold length = 4 nt)

S41 (0% CG content): TAAT, ATAA, TAAA, TTAT, TATA, AATT, TTTT, TTTA, AAAA, ATTA
S42 (100% CG content): GCCC, CGCG, CCGG, CCGC, CGCC, GGGG, CCCC, CGGG, GCGG, GCCG

Table 1

	TAAT	ATAA	TAAA	TATA	AATT	AAAA
TAAT	-8.79	-8.80	-8.95	-8.37	-9.23	-9.37
ATAA	-8.79	-8.80	-8.95	-8.37	-9.23	-9.37
TAAA	-9.22	-9.23	-9.38	-8.80	-9.66	-9.8
TATA	-8.65	-8.66	-8.81	-8.23	-9.09	-9.23
AATT	-8.93	-8.94	-9.09	-8.51	-9.37	-9.51
TTTT	-9.36	-9.37	-9.52	-8.94	-9.80	-9.94

Table 2

	TAAT	ATAA	TAAA	TATA	AATT	AAAA
GCCC	-12.51	-12.52	-12.67	-12.09	-12.95	-13.09
CGCG	-13.39	-13.40	-13.55	-12.97	-13.83	-13.97
CCGC	-12.94	-12.95	-13.10	-12.52	-13.38	-13.52
GGGG	-12.06	-12.07	-12.22	-11.64	-12.50	-12.64
CGGG	-12.48	-12.49	-12.64	-12.06	-12.92	-13.06
GCGG	-12.94	-12.95	-13.10	-12.52	-13.38	-13.52

Table 3

	GCCC	CGCG	CCGG	CCGC	GGGG	GCGG
GCCC	-16.24	-17.08	-16.21	-16.67	-15.79	-16.67
CGCG	-17.12	-17.96	-17.09	-17.55	-16.67	-17.55
CCGG	-16.25	-17.09	-16.22	-16.68	-15.80	-16.68
GGGG	-15.79	-16.63	-15.76	-16.22	-15.34	-16.22
CGGG	-16.21	-17.05	-16.18	-16.64	-15.76	-16.64
GCGG	-16.67	-17.51	-16.64	-17.10	-16.22	-17.10

Table 4

	0-bp
TAAT	-3.05
ATAA	-3.05
TAAA	-3.48
TATA	-2.91
AATT	-3.19
AAAA	-3.62

Table 5

	0-bp
GCCC	-6.77
CGCG	-7.65
CCGG	-6.78
CCGC	-7.20
GGGG	-6.32
CGGG	-6.74

Table 6

	1-bp	
	A-T	G-C
TAAT	-5.87	-6.15
ATAA	-5.87	-6.15
TAAA	-6.30	-6.58
TATA	-5.73	-6.01
AATT	-6.01	-6.29
AAAA	-6.44	-6.72

Table 7

	1-bp	
	A-T	G-C
GCCC	-9.59	-9.87
CGCG	-10.47	-10.75
CCGG	-9.6	-9.88
CCGC	-10.02	-10.30
GGGG	-9.14	-9.42
CGGG	-9.56	-9.84

Table 8

	2-bp	
	AT	AA
TAAT	-6.90	-7.04
ATAA	-6.90	-7.04
TAAA	-7.33	-7.47
TATA	-6.76	-6.90
AATT	-7.04	-7.18
AAAA	-7.47	-7.61

Table 9

	2-bp	
	GG	CG
TAAT	-8.12	-8.53
ATAA	-8.12	-8.53
TAAA	-8.55	-8.96
TATA	-7.98	-8.39
AATT	-8.26	-8.67
AAAA	-8.69	-9.10

Table 10

	2-bp	
	AT	AA
GCCC	-10.62	-10.76
CGCG	-11.50	-11.64
CCGG	-10.63	-10.77
CCGC	-11.05	-11.19
GGGG	-10.17	-10.31

CGGG	-10.59	-10.73

Table 11

	2-bp	
	GG	CG
GCCC	-12.84	-12.25
CGCG	-12.72	-13.13
CGCC	-12.27	-12.68
GGGG	-11.39	-11.80
CGGG	-11.81	-12.22
GCGG	-12.27	-12.68

Table 12

	3-bp		
	ATT	ATA	AAA
TAAT	-8.06	-7.63	-8.21
ATAA	-8.06	-7.63	-8.21
TAAA	-8.49	-8.06	-8.64
TATA	-7.92	-7.49	-8.07
AATT	-8.20	-7.77	-8.35
AAAA	-8.63	-8.20	-8.78

Table 13

	3-bp		
	ATT	ATA	AAA
GCCC	-11.78	-11.35	-11.93
CGCG	-12.66	-12.23	-12.81
CCGG	-11.79	-11.36	-11.94
CCGC	-12.21	-11.78	-12.36
GGGG	-11.33	-10.90	-11.48
CGGG	-11.75	-11.32	-11.90

Table 14

	3-bp		
	CGG	CGC	GGG
TAAT	-10.51	-10.97	-10.10
ATAA	-10.51	-10.97	-10.10
TAAA	-10.94	-11.40	-10.53
TATA	-10.37	-10.83	-9.96
AATT	-10.65	-11.11	-10.24
TTTT	-11.08	-11.54	-10.67

Table 15

	3-bp		
	CGG	CGC	GGG
GCCC	-14.23	-14.69	-13.82
CGCG	-15.11	-15.57	-14.70
CCGG	-14.24	-14.70	-13.83
CCGC	-14.66	-15.12	-14.25

GGGG	-13.78	-14.24	-13.37
CGGG	-14.20	-14.66	-13.79

B

The Tile Pattern Self-replication Simulator

B.1 Code Base

The TPSS is written in Java, and built upon NetBeans, which is a cross-platform that can be built and run on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible Java Virtual Machine (JVM). There is no requirement of any third party library to build and run the simulator.

B.2 Architecture

A screenshot of the simulator modules and overall architecture is shown in Figure B.1. The two versions of the simulator, the aTPSM and the kTPSM, can be run using modules *AbstractTPSWrapper.java* and *KineticTPSWrapper.java*, respectively.

B.3 Compiling and Running the Simulator

Download and install the NetBeans IDE in your machine. Unzip the simulator code file, and open the project file 'TPSS' a NetBeans project. Scroll down to *com.arl.chips.core* and expand it. The two java files, *AbstractTPSWrapper.java* and *KineticTPSWrapper.java*, would be visible here, if everything was done correctly. The simulation can be launched by selecting *AbstractTPSWrapper.java* for the abstract model and *KineticTPSWrapper.java* for the kinetic model. After selecting the model, go to the top of the NetBeans window, and expand the run option. Here you can compile/build and run the simultor.

User can define the number of rows, number of columns of the pattern and number of self-replication cycles. There is no limitation on the number of rows/columns and the number of iteration for which the abstract simulator can be run. However, for the kinetic model, if you choose a larger size pattern (e.g., a pattern of 100x100) and/or large number of replication cycles (e.g., 20), the simulation may lost longer. Take a coffee break and hopefully it will be complete when you come back!

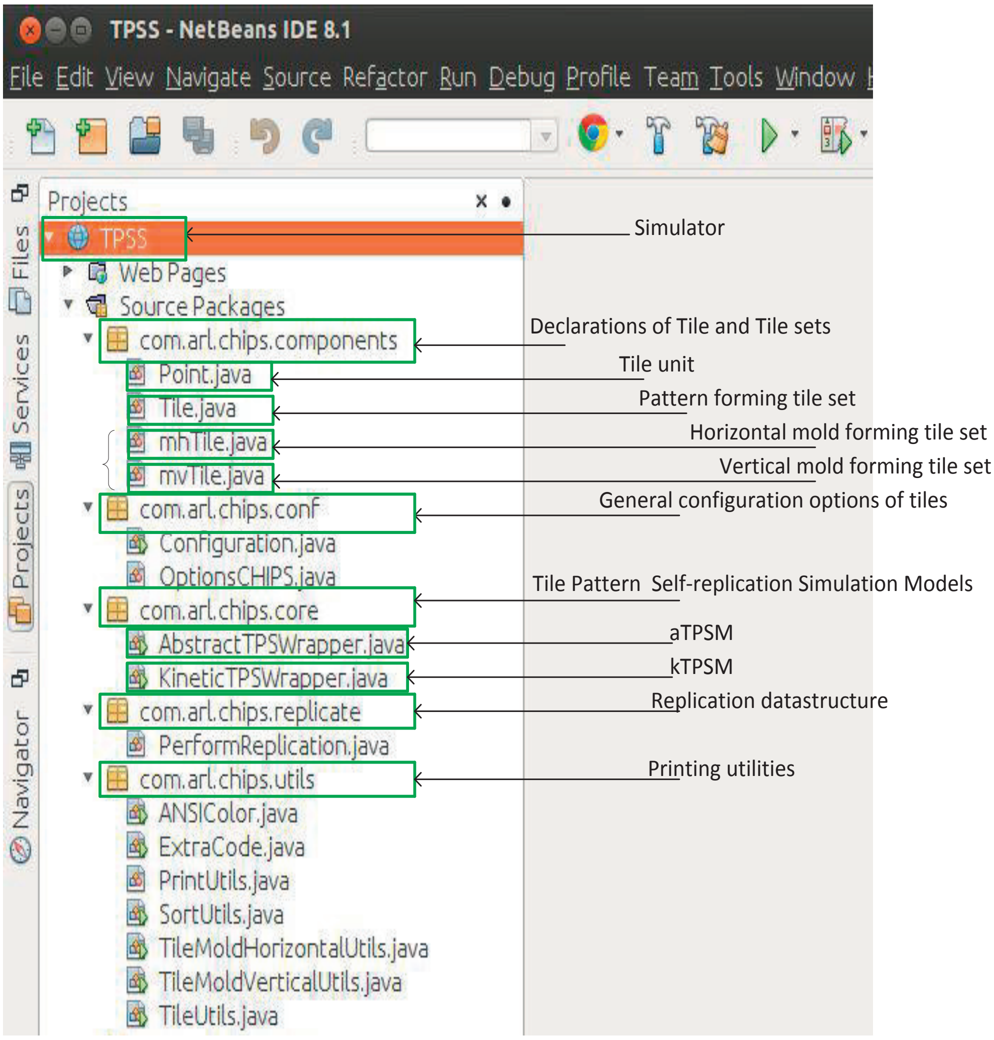
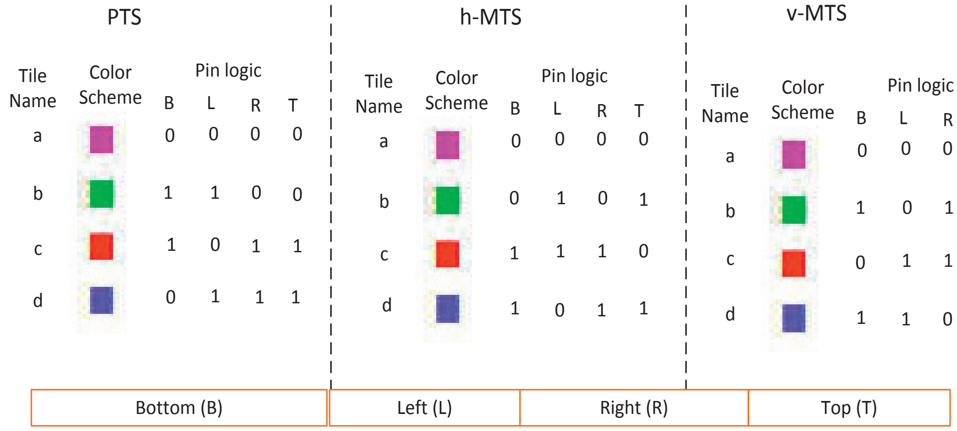


Figure B.1: Architecture of the simulator.

B.4 Simulation snapshots from the TPSS

B.4.1 How to read the snapshots?

The following simulation snapshots use a color display scheme (see Figure B.2) to represent the PTS, h-MTS, and v-MTS, previously illustrated in Figures: 7.15, 7.17, and 7.18. There are four tiles in each set, and each tile within a set has four pins: bottom (B), left(L), right(R), and top(T). Each tile has a designed set of logical values associated with its pins. Logic mapping (XORing) between the pins of different tiles is: in PTS tiles, $(B, L) \rightarrow (T, R)$; in h-MTS tiles, $(L, T) \rightarrow (B, R)$; in v-MTS tiles, $(B, R) \rightarrow (L, T)$.



PTS

Tile Name	Color Scheme	B	L	R	T
a	🟪	0	0	0	0
b	🟩	1	1	0	0
c	🟥	1	0	1	1
d	🟦	0	1	1	1

h-MTS

Tile Name	Color Scheme	B	L	R	T
a	🟪	0	0	0	0
b	🟩	0	1	0	1
c	🟥	1	1	1	0
d	🟦	1	0	1	1

v-MTS

Tile Name	Color Scheme	B	L	R	T
a	🟪	0	0	0	0
b	🟩	1	0	1	0
c	🟥	0	1	1	1
d	🟦	1	1	0	1

Pin logic
Bottom (B)
Left (L)
Right (R)
Top (T)

Figure B.2: Color display scheme in the snapshots.

An initialization of grid and L-shaped seed of the target pattern is illustrated in Figure B.3. The simulator panel first initializes a (m+1) x (n+1) grid with null (N) values to each grid position, where m and n (m = 6, n = 8 in the figure) are the rows and columns of the target pattern.

Next, a L-shaped seed of the mxn pattern is initialized, which is used further to form the target pattern. The L-shaped seed initialization starts by placing one of the four tiles from the PTS (it is tile 'b', as shown in bottom of Figure B.3) at the corner. Therefore, the subsequent tiles that can join the left and the top of the 'b' corner tile should match the logical values on their left and bottom pins, respectively. Following this simple local mapping, the L-shaped seed is initialized. Due to multiple choices for pin mappings, there can be different L-shaped seed patterns. In the TPSS, we can chose to fix a particular L-shaped seed after it is initialized.

In the simulation snapshots, we used pointers (see Figure B.4) to represent the transition between the cycles. The first cycle is represented by (0,0). The transition between the first and second cycles is denoted by $(0, 0) \Rightarrow (1, 0)$ and $(0, 0) \Rightarrow (1, 1)$, representing the patter-to-mold formation and mold-pattern formation, respectively. Similarly, the transitions between the second and third cycles are denoted by $(1, 0) \Rightarrow (2, 0)$, $(1, 0) \Rightarrow (2, 1)$,$(1, 1) \Rightarrow (2, 2)$, and $(1, 1) \Rightarrow (2, 3)$.

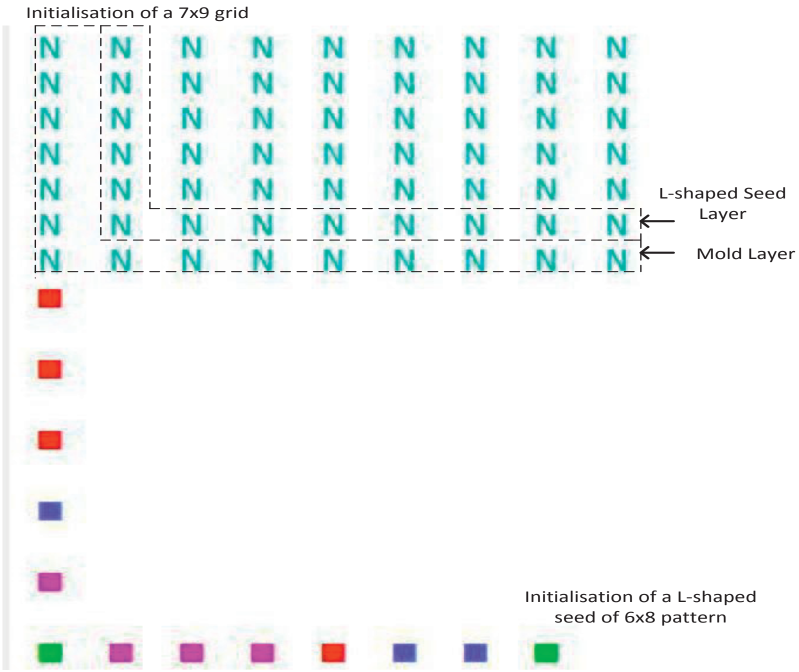

Initialisation of a 7x9 grid
L-shaped Seed Layer
Mold Layer
Initialisation of a L-shaped seed of 6x8 pattern

Figure B.3: Pointers representing transitions between the self-replication cycles.

B.4.2 The Simulation Snapshots from the aTPSM

The following snapshots have been captured from a sample run of the aTPSM. The parameters used here are: row size of the target pattern = 6; column size of the target pattern = 8; number of self-replication cycles = 3. There are two types of time stamps shown in the snapshots: 1) CTIME STAMP is used to show the progress/unit time step in the mold to pattern formation and pattern to mold formation processes within each self-replication cycle; 2) The TIME STAMP (TS) shows the total time taken to complete a given number of self-replication cycles.

Figure B.5 shows a few steps of the first self-replication cycle. Figures: B.6 – B.13 represent the second and third replication cycles.

The aTPSM completes three replication cycles of a 6x8 tiles pattern in a total 39 time steps (TS), as shown in Figure B.13(b)). The pattern-to-mold formation and mold to pattern formation require 15 time steps and 9 time steps, respectively (considering one time step for the dissociation of pattern-mold complex).

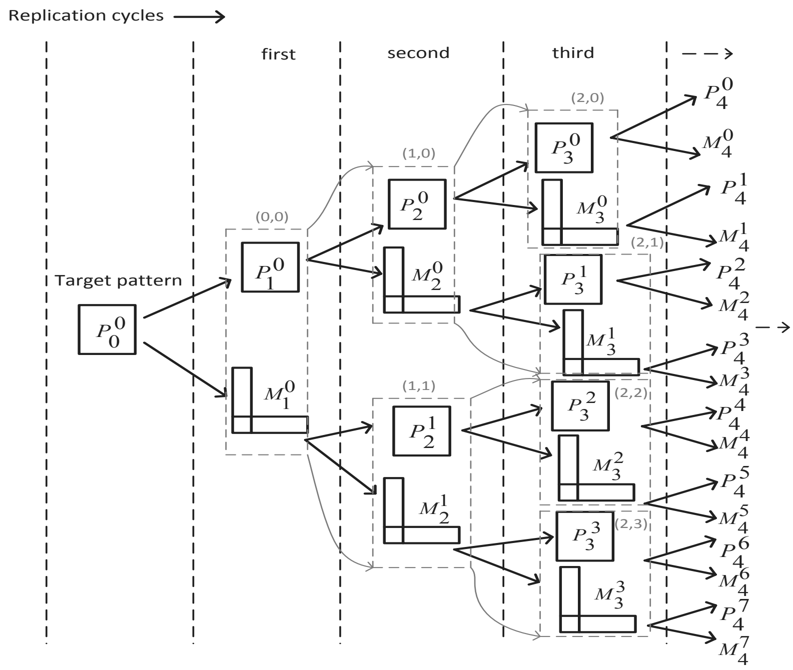


Figure B.4: Pointers representing the transitions between the self-replication cycles. Due recursive nature of the self-replication cycles, the pointers start with the first cycle.

B.4.3 The Simulation Snapshots from the kTPSM

The following snapshots have been captured from a sample run of the kTPSM. The parameters used here are: row size of the target pattern = 6; column size of the target pattern = 8; $G_{mc} = 16$; $G_{se} = 8.8$; number of self-replication cycles = 3.

The following snapshots (Figures: B.14, B.15) show only few steps of pattern-to-mold formation and mold-to-pattern formation in the kTPSM. There are several events of tile attachment/detachment in between the steps marked in these snapshots, which are not shown due to space constraint. Steps: 1-21 in Figure B.14 show the pattern-to-mold formation in the kTPSM, where tiles attach and detach based on the probabilistic preferences. Steps: 1-22 in Figure B.15 show the mold-to-pattern formation during the first replication cycle ($(0,0) \Rightarrow (1,1)$). The remaining replication cycles are similar to the cycles previously described in the aTPSM, while the processes of pattern-to-mold and mold-to-pattern formation remain the same as described here.

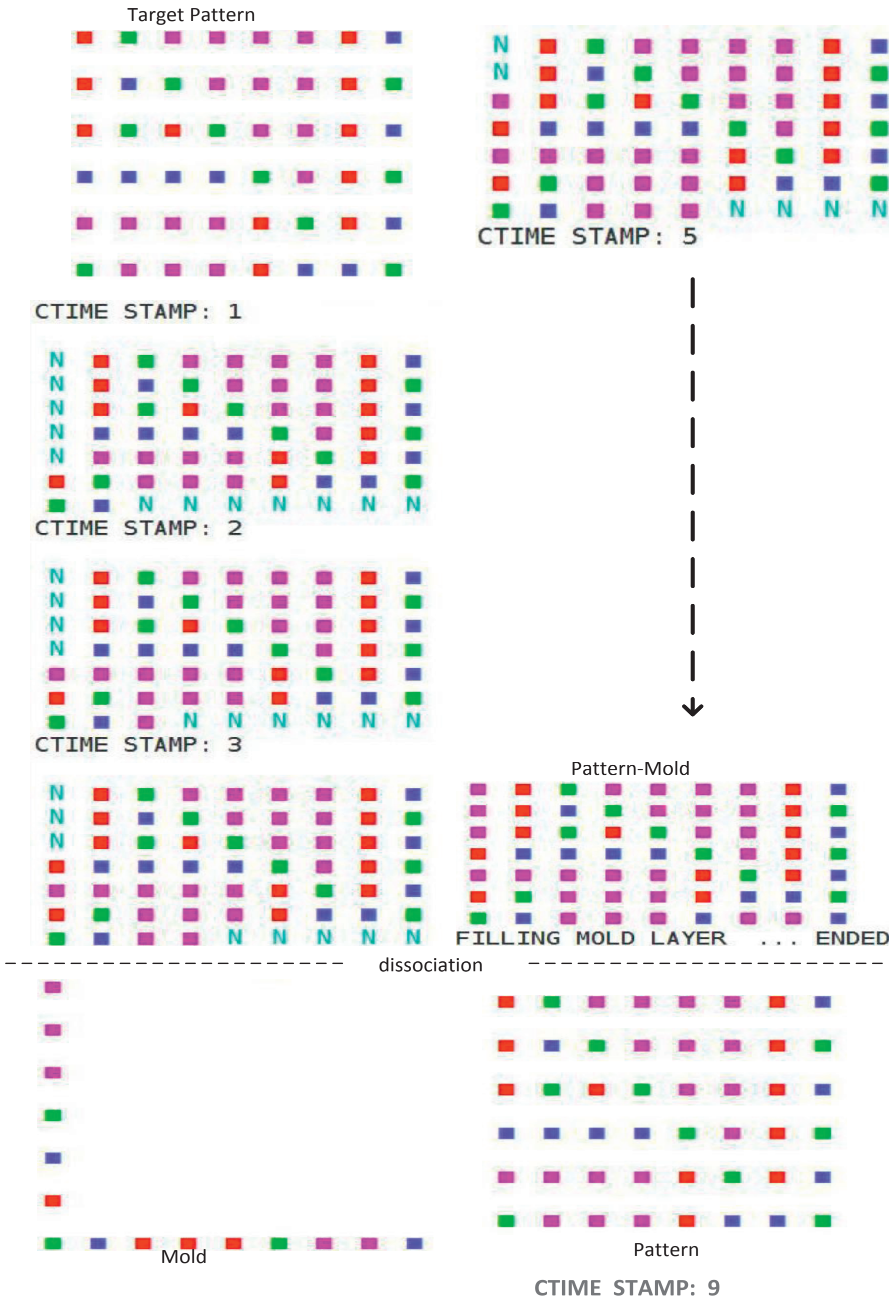
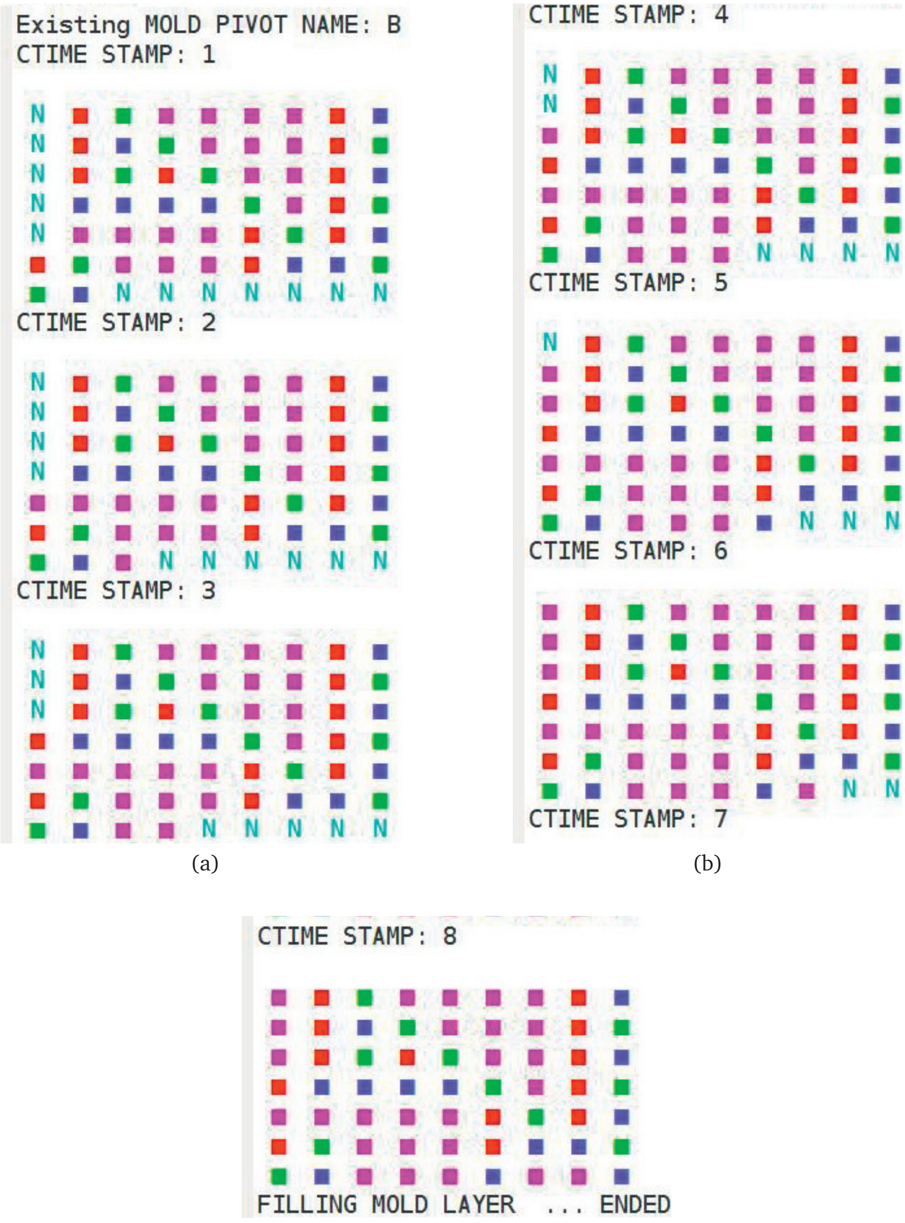


Figure B.5: First cycle of the pattern self-replication in the aTPSM: a mold is first formed and gets dissociated at the end of the first cycle.



```
Existing MOLD PIVOT NAME: B
CTIME STAMP: 1

N 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
N 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
N 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
N 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
N 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 N N N N N N N
CTIME STAMP: 2

N 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
N 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
N 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
N 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟪 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟪 N N N N N N
CTIME STAMP: 3

N 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
N 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
N 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
🟥 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟪 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟪 🟪 N N N N N
```

(a)

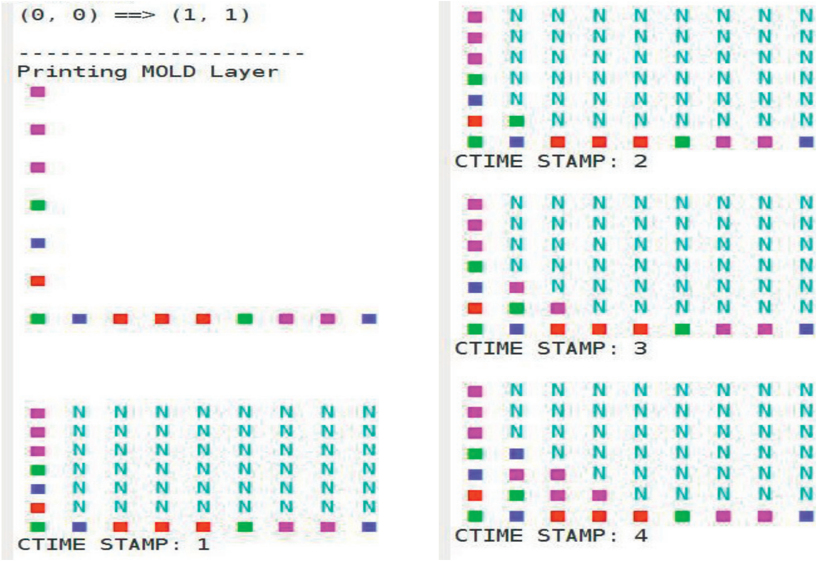
```
CTIME STAMP: 4

N 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
N 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
🟪 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
🟥 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟪 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟪 🟪 🟪 N N N N
CTIME STAMP: 5

N 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
🟪 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
🟪 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
🟥 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟪 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟪 🟪 🟪 🟦 N N N
CTIME STAMP: 6

🟪 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
🟪 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
🟪 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
🟥 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟪 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟪 🟪 🟪 🟦 🟪 N N
CTIME STAMP: 7
```

(b)

```
CTIME STAMP: 8

🟪 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
🟪 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
🟪 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
🟥 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟪 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟪 🟪 🟪 🟦 🟪 🟪 🟦
FILLING MOLD LAYER  ... ENDED
```

(c)

Figure B.6: The second cycle($(0,0) \Rightarrow (1,0)$): pattern-to-mold formation (a), (b), (c).



```
(0, 0) ==> (1, 1)

---------------------
Printing MOLD Layer
```

```
CTIME STAMP: 1
```

(a)

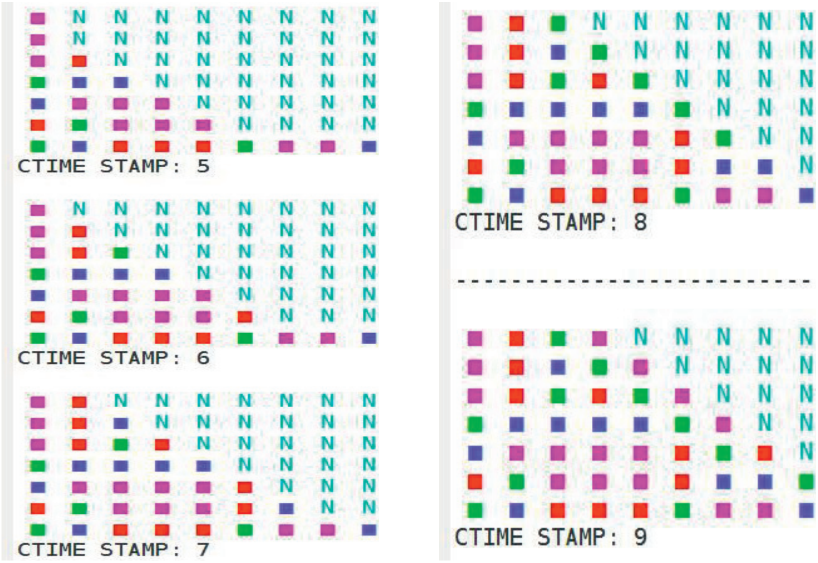
```
CTIME STAMP: 2
```

```
CTIME STAMP: 3
```

```
CTIME STAMP: 4
```

(b)

Figure B.7: The second cycle($(0,0) \Rightarrow (1,1)$): mold-to-pattern formation (a), (b).



```
CTIME STAMP: 5
```

```
CTIME STAMP: 6
```

```
CTIME STAMP: 7
```

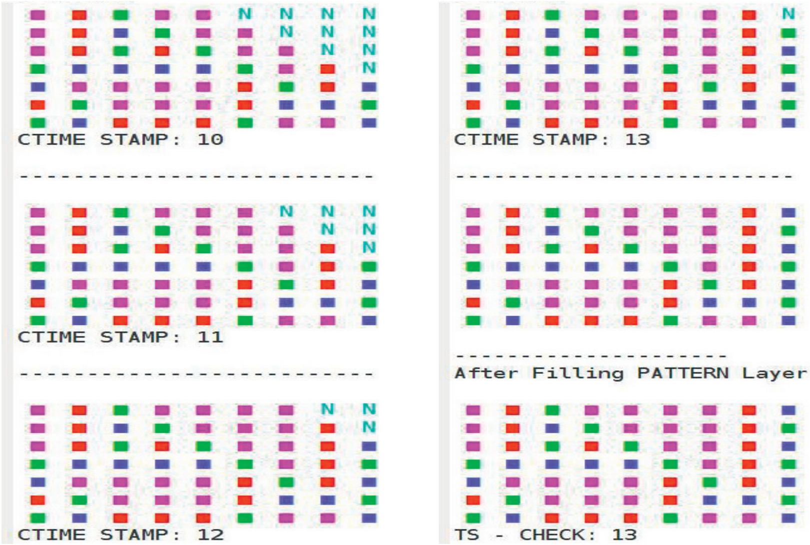
(a)

```
CTIME STAMP: 8

-------------------------
```

```
CTIME STAMP: 9
```

(b)

Figure B.8: The progress of the second cycle($(0,0) \Rightarrow (1,1)$): mold-to-pattern formation (a), (b).



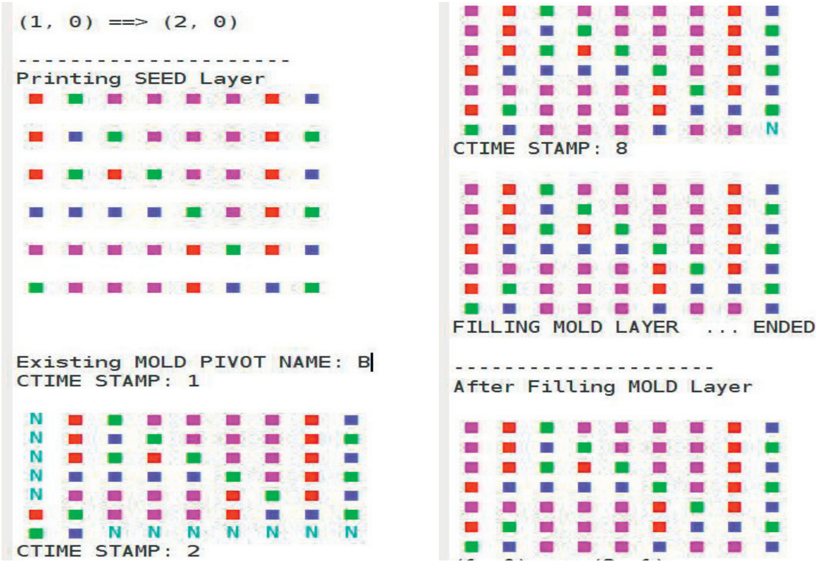
```
CTIME STAMP: 10
--------------------------
CTIME STAMP: 11
--------------------------
CTIME STAMP: 12
```

(a)

```
CTIME STAMP: 13
-------------------------
--------------------
After Filling PATTERN Layer
TS - CHECK: 13
```

(b)

Figure B.9: The second cycle $((0,0) \Rightarrow (1,1))$: (a) Mold to pattern formation is in progress. (b) Mold-to-pattern formation is complete.



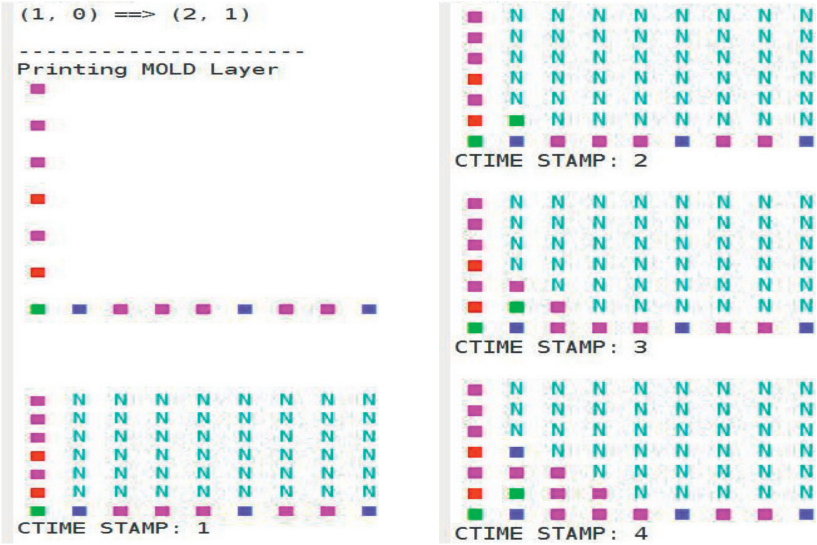
```
(1, 0) ==> (2, 0)
---------------------
Printing SEED Layer

Existing MOLD PIVOT NAME: B
CTIME STAMP: 1
CTIME STAMP: 2
```

(a)

```
CTIME STAMP: 8
FILLING MOLD LAYER  ... ENDED
--------------------
After Filling MOLD Layer
```

(b)

Figure B.10: Third replication cycle $((1,0) \Rightarrow (2,0))$: a few steps of pattern-to-mold formation (a), (b).

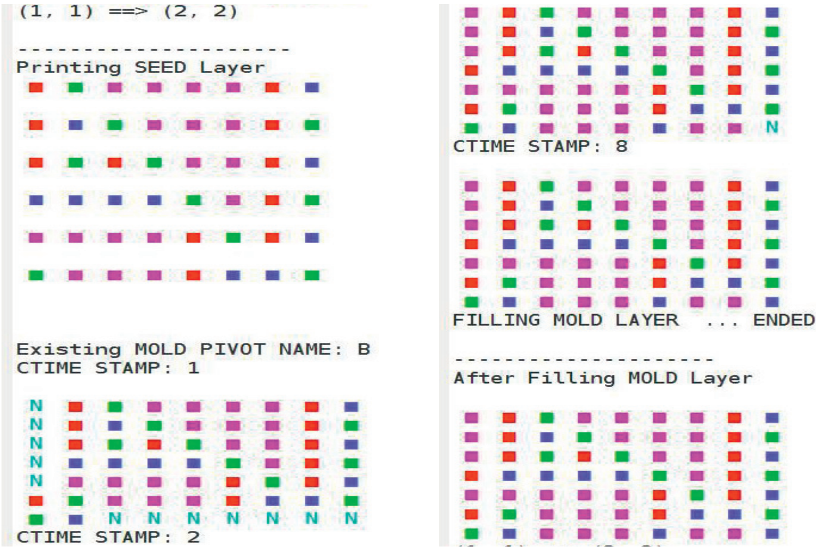


```
(1, 0) ==> (2, 1)

---------------------
Printing MOLD Layer

CTIME STAMP: 1
```

```
CTIME STAMP: 2

CTIME STAMP: 3

CTIME STAMP: 4
```

(a) (b)

Figure B.11: Third replication cycle ($(1,0) \Rightarrow (2,1)$): a few steps of mold-to-pattern formation (a),(b).

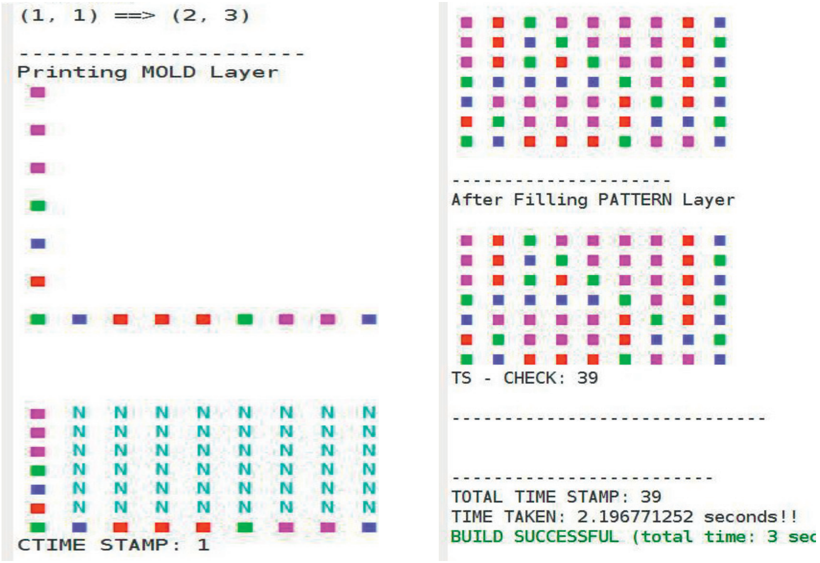


```
(1, 1) ==> (2, 2)

---------------------
Printing SEED Layer

Existing MOLD PIVOT NAME: B
CTIME STAMP: 1

CTIME STAMP: 2
```

```
CTIME STAMP: 8

FILLING MOLD LAYER ... ENDED

--------------------
After Filling MOLD Layer
```

(a) (b)

Figure B.12: Third replication cycle ($(1,1) \Rightarrow (2,2)$): pattern-to-pattern formation (a),(b).



```
(1, 1) ==> (2, 3)

---------------------
Printing MOLD Layer
🟪
🟪
🟪
🟩
🟦
🟥
🟩 🟦 🟥 🟥 🟥 🟩 🟪 🟪 🟦


🟪 N N N N N N N N
🟪 N N N N N N N N
🟪 N N N N N N N N
🟩 N N N N N N N N
🟦 N N N N N N N N
🟥 N N N N N N N N
🟩 🟦 🟥 🟥 🟥 🟩 🟪 🟪 🟦
CTIME STAMP: 1
```

(a)

```
🟪 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
🟪 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
🟪 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
🟩 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟦 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟥 🟥 🟥 🟩 🟪 🟪 🟦

--------------------
After Filling PATTERN Layer

🟪 🟥 🟩 🟪 🟪 🟪 🟪 🟥 🟦
🟪 🟥 🟦 🟩 🟪 🟪 🟪 🟥 🟩
🟪 🟥 🟩 🟥 🟩 🟪 🟪 🟥 🟦
🟩 🟦 🟦 🟦 🟦 🟩 🟪 🟥 🟩
🟦 🟪 🟪 🟪 🟪 🟥 🟩 🟥 🟦
🟥 🟩 🟪 🟪 🟪 🟥 🟦 🟦 🟩
🟩 🟦 🟥 🟥 🟥 🟩 🟪 🟪 🟦
TS - CHECK: 39

-----------------------------


------------------------
TOTAL TIME STAMP: 39
TIME TAKEN: 2.196771252 seconds!!
BUILD SUCCESSFUL (total time: 3 sec
```

(b)

Figure B.13: Third replication cycle ($(1, 1) \Rightarrow (2, 3)$): mold-to-pattern formation (a),(b).

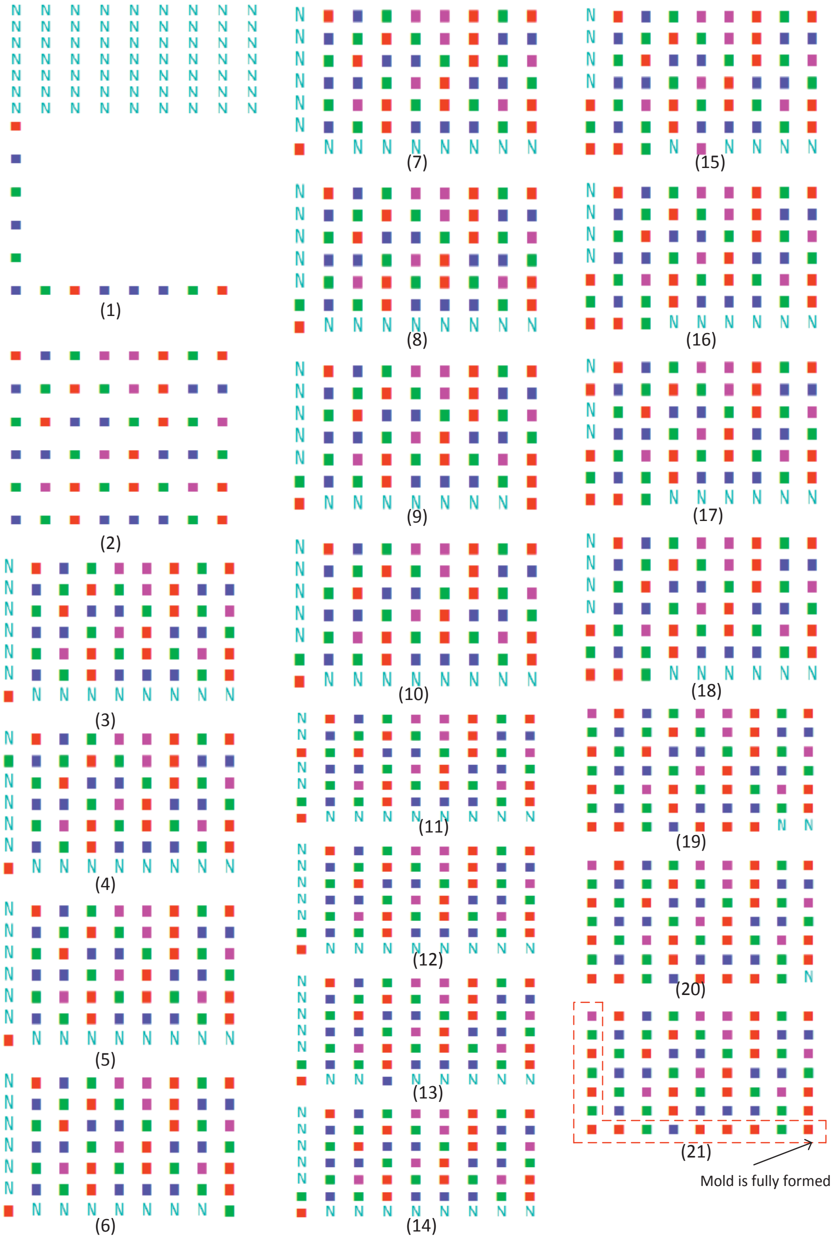
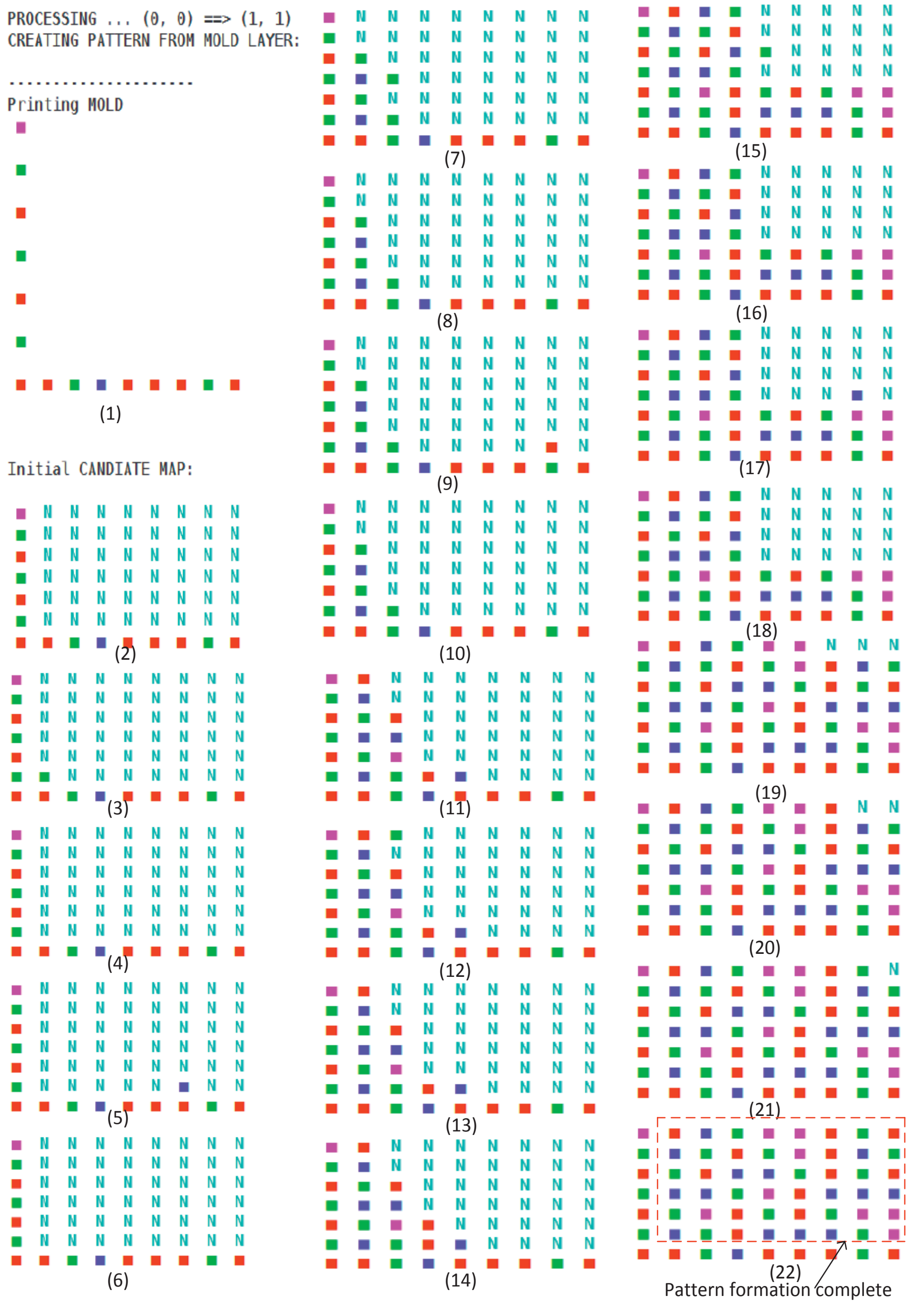


Figure B.14: Pattern-to-mold formation in the kTPSM: Steps(1-21) represent a few intermediate steps of the pattern-to-mold formation process.



```
PROCESSING ... (0, 0) ==> (1, 1)
CREATING PATTERN FROM MOLD LAYER:

......................
Printing MOLD
```

```
Initial CANDIATE MAP:
```

Figure B.15: Mold-to-pattern formation in the kTPSM: Steps(1-22) represent a few intermediate steps of the mold-to-pattern formation process.