NTNU

Norwegian University of
Science and Technology

Modelling Neuronal Activity with Jittered Generalised Linear Models

Kristian Aga

Master of Science in Physics and Mathematics
Submission date: March 2018
Supervisor: Mette Langaas, IMF
Co-supervisor: Benjamin Dunn, NTNU, Kavliinstitutt for nevrovitenskap

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This Master's thesis constitutes the final part of my masters program - Statistics for the Industrial mathematics program at the Department of Mathematical Sciences at the Norwegian University of Science and Technology (NTNU). The topic of this thesis is modelling neuronal activity with jittered generalised linear models (JGLM). The project evolved from a previous project written by Aga and Fawad (2017), where the main focus was to model neuronal activity with GLM. In particular Chapter 3 and 5 have benefited from this collaboration.

I would like to thank my supervisor Mette Langaas at the Department of Mathematical Sciences for excellent guidance and motivation in the process of writing this thesis. I would also like to thank my co-supervisor Benjamin A. Dunn at the Kavli Institute for Systems Neuroscience for including me into the fascinating world of neuroscience, without him I would not have been able to come up with the idea of JGLM. Additionally I would like to thank Buzsáki lab for providing the data presented in Oliva et al. (2016).

Kristian Aga
Trondheim, Norway
March 2018

Abstract

By measuring electrophysiological data from a rat's brain we are able to study the relations between cells, and thereby study the brain itself. In this thesis we present different statistical modelling techniques for treating neuroscience data, detecting tuning of cells, detecting connectivity between cells and thereby investigating the flow of communication in the brain. We identify head direction and spatially tuning of cells with different neuroscience modelling techniques. Most importantly we develop the new method, jittered generalised linear models (JGLM). JGLM combines the best parts of the jittered cross-correlation method and the generalised linear model into one framework, and utilises permutation test on the jittered likelihood-values to test for connectivity between cells. JGLM is useful in detecting connectivity between cells and studying information in network of neurons. Additionally we develop the tool of basis-tuning-curve, which we use for classifying the type of connection between neurons. We have discovered that interval jittering is a good jittering procedure, while basic jittering is the wrong jittering choice, for jittering both in the JGLM and the JCC framework for analysing neuronal activity.

The data analysed from the *Oliva-16* data set contains electrophysiological data from cells recorded from movement related brain areas, as well as movement data for the rat. In the end we study the neuroscience findings from investigating the data set with JGLM. Furthermore we discuss the potential of JGLM and key elements in future research on JGLM, where ground-truth-testing of JGLM is essential. Our results shows that JGLM is well suited for analysing neuronal activities.

Contents

I Neuroscience

1 Introduction

We will begin this chapter with a problem description. We will explain why this work is interesting both from a neuroscience and statistical point of view. We then present background material from neuroscience, before ending with a thesis outline.

1.1 Problem description

We are nothing without connections, therefore to identify connections correctly should be of extreme interest to us. The connectivity in the cells, does not only decide which cells talks with each other, but also constitutes the essence of our memory and decisions. The connectivity determines the selectivity of the cells, and without it our brains would be useless.

From a neuroscience point of view modelling neuronal activity is of interest because we look at connections, relations and explanatory factors for cells. First we will investigate the potential to model different covariate factors such as history effect, connectivity effect, speed, position and head direction (HD). Additionally we will look at the relation between spatially tuned cells and head direction tuned cells.

Furthermore, our work is of statistical interest, developing, investigating and using different statistical methods and analysing their general and specific usefulness. We will use the GLM framework and investigate the connectivity between cells using basis functions. Most importantly we will develop a new method, jittered generalised linear model (JGLM), which utilises jittering in a GLM framework to produce permutation test on log-likelihood-values for performing a hypothesis test. Thus, we develop a new method to detect connectivity between neurons.

1.2 The nervous system

The nervous system is the grand controller in the human body. It processes information and decides what to do with it. The peripheral nervous system is the part of the nervous system that consists of nerves and ganglia outside of the brain and the spinal cord, while the central nervous system is the brain and spinal cord. The essential role for the peripheral nervous system is to connect the central nervous system to the limbs and organs in the body.

In other terms we can consider the peripheral nervous systems primary task to relay information that have been detected in our sensory organs to the central nervous system. Its secondary but equally important task, is to relay the information from the central nervous system on what actions to take, to the limbs and organs throughout the body.

Thus, from understanding the peripheral nervous systems tasks, the central nervous system can be considered as a magic black box, where information from sensory organs and from the rest of the body is the input, and action decisions based on this information is the output. Sometimes the decision process is a fast acting process such as regulating body heat, and other times it can be a really long decision process almost getting stuck in the brain, such as thinking about philosophical questions.

As is often the case in modern days, when we talk about neuroscience we talk about the the science of the brain. The brain is considered the centre of the nervous systems in most animals, and is the most complex organ in any vertebrate's body. As such, it should be no surprise that the intricate details and behaviour of the brain is still unknown. However, some of the coarser information of the relationships and the way the brain works is known.

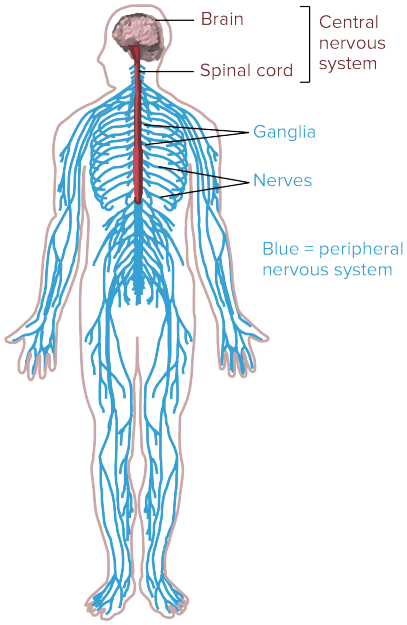

Figure 1: The human nervous system. Source: `https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse`

Another analogy for the brain is to view it as the rest of the body's external decision-centre. Small quick decision can often be decided locally throughout the body, such as removing your hand from a hot plate. However, other more complicated decision-process based on more information is being "outsourced" to the brain to think about and thereafter being solved.

1.2.1 Brain areas

Rodent and human brains The analysed data as well as the following text focuses on brain areas related to movement. Rodents are genetically similar to humans, and both mice and rats are frequently looked at in order to study the brain. Rodents have a short lifespan which have made them a preferred choice when studying behaviours over generations. Furthermore rats are bigger than most mice, thus their brains are bigger as well, and therefore easier to handle.

Compared to the human brain the rat brain is shaped differently, due to the differing shape of our heads. Furthermore the surface of the human brain (the cortex) are wrinkled while the rat brain is not. The wrinkled surface as well as the bigger size of the human brain makes humans able to have more neurons than rats, humans are therefore capable of functioning at a higher cognitive level than rats. Some brain areas like the olfactory parts are relatively bigger for the rat, as smelling is a really important tool for the rats. Humans however, have relatively bigger brain area related to language than rats. However, the functional structure of the rat brain is quite similar to that of a human. Thus, by studying the rat brain, a lot can be learned about the human brain, as well as the interactions between neurons.

Figure 2: Artistic representation of the neuronal connections in the human brain. Source: `http://www.gregadunn.com/self-reflected/self-reflected-gallery/`

Brain structure The brain can be divided in many different ways. Firstly it's common to divide it into three, the forebrain (telencephalon and diencephalon), midbrain (mesencephalon) and hindbrain (rhombencephalon). Evolutionary the hindbrain developed first thereafter the midbrain and at last the forebrain. Thus, roughly speaking the hindbrain is responsible for the most fundamental tasks related to our survival such as breathing, and blood flow. The hindbrain can be said to control necessary bodily functions that are outside conscious control. Thereafter the midbrain is tasked with the mid-level important tasks for survival such as senses and temperature regulation, which is in between conscious and sub-conscious control. At last the forebrain is tasked with more complicated tasks such as awareness of position and coordinating movement decisions. The forebrain, or the more specific sub-region of the forebrain called the cortex is in humans responsible for the majority of the processing in the brain.

Another important part of the forebrain is the hippocampus. The hippocampus and a specific part of the cortex called the motor cortex, are the main brain areas that are connected to movement related activity.

Analysing movement data is essential to better understand the brain, as movement tasks are relatively simplistic and intuitive. Additionally movement tasks are an important part of a rats decision making, and also relatively easy to measure and control. The control variables for our data set are further explained in the "Interface of the move Data" paragraph in Section 2.1.2.

The data set analysed in this paper measured data from the Cornu Ammonis areas CA1, CA2 and CA3, the dentate gyrus (DG) and the entorhinal cortex (EC) which all are believed to be related to movement tasks. The CA areas resides in the hippocampus and are related to temporary memory. The DG are thought to be related to new episodic memories and exploration of new environments, and receives most excitatory input from the cortex. Entorhinal cortex is believed to be related to the processing of the short term memory into longer memory.

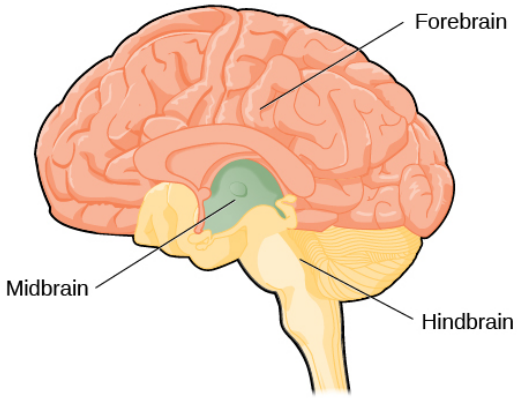

Figure 3: Human brain. Source: https://archive.cnx.org/contents/016a7e55-4c3d-49d4-bc4e-750e8f5de8cc@1/3-4-the-brain-and-spinal-cord-sw

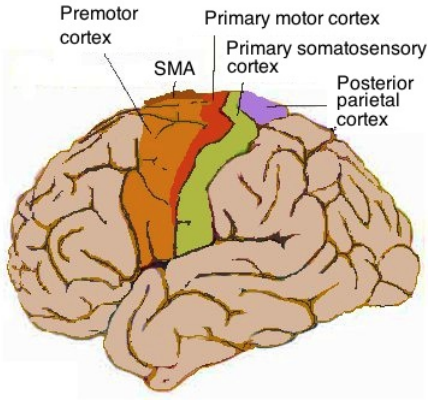

Figure 4: The human motor cortex overview. Source: https://en.wikipedia.org/wiki/Motor_cortex

1.2.2 Neurons

The main functional units of the nervous system are the neurons (nerve cells/cells). Neurons are electrically excitable cells that receives and transmits electrical and chemical signals. It is believed that neurons are the essence of communication in the brain. It has been estimated that the human brain has roughly 10^{11} neurons, each neuron may be connected up to another 10 000 other neurons. The structure of the neuron consist of the soma (the cell body) which contains the nucleus, DNA, mitochondria, ribosomes and cytoplasm as any other cell. Furthermore neurons have dendrites that projects out of the soma, the cell body, that are listeners that pick up messages (electrical or chemical signals such as electrons) from other cells, and convey their information to the cell body. Additionally neurons have axons that can either be very short or reach up to 1 meter, that are the talkers, which are the way of the cell to send messages to other cells (often other neurons). There exists a few different axon types, but most axons transmit electrons away from the cell body to other cells. Signals between neurons occur via synapses. A synapse is typically where an axon meets a dendrite. A synapse can be divided into the presynaptic cell, the sender/talker, and the postsynaptic cell, the receiver. It is possible but not likely, for two neurons to directly send and receive electrons from each other, as axons most of the time only naturally allow for electrons emitted one way.

Neurons emit electrical signals (firings). As explained in Section 1.2.3, a firing can be either excitatory or inhibitory, otherwise one such firing doesn't differ much from the other. However, the the temporal sequence of many such firings and the frequency that they are emitted can vary a lot. It's by studying such relations we will investigate the information flow in the brain in this thesis. Neurons are long lived cells that are highly specialised coming in all shapes and sizes, thus we expect the activity in different neurons to be related to different tasks as well as being related to other neurons in many different ways.

1.2.3 Action potential and neural communication

Action potential Most brain areas are constructed such that there usually is a negative electrical charge intracellularly (inside the cells/neurons), while there is a positive electrical charge extracellularly (outside of the cell). This difference in electrical charge is called the *membrane potential*. The membrane potential value is defined by the difference between inner and outer electrical charge. For a typical neuron the membrane potential is -70mV, called the *resting potential* or *resting state*. However, if a cell is exposed to a stimulus which increases the membrane potential value to above the *threshold potential* value (roughly -55mV), the membrane potential value will continue to increase and we will have an action potential. However, if the stimulus doesn't change the membrane potential to above the threshold potential we will have a failed initiation and no action potential.

A visual representation of the action potential cycle can be seen in Figure 6.

Thus, an *action potential* occurs when the negative charge inside of the cell rises compared to the outside of the cell. Thereafter it returns to the resting state. The two phases of the action potential cycle are called the depolarisation phase, and the repolarisation phase. The repolarisation phase is also called the refractory period. It's important to note that a cell is less likely to fire again during the refractory period, even though it receives a stimulus.

In the first phase, called the depolarisation phase, the membrane potential value, increases. However, the membrane potential, which is the difference in electrical charge, decreases. This reduction in the absolute value of the membrane potential is associated with a release of energy. This energy is in turn used to send (fire) a signal along the axons toward another cell.

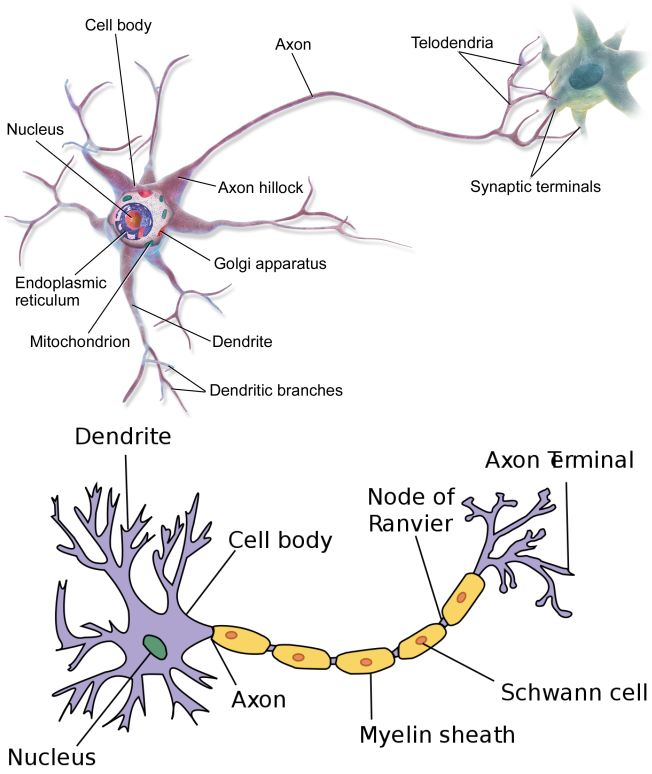

Figure 5: Two different visualisations of the anatomy of a neuron. Source: `https://en.wikipedia.org/wiki/Neuron`

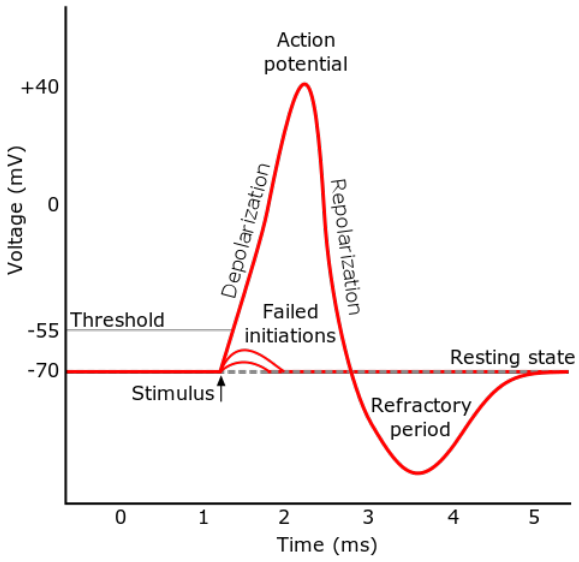

Figure 6: Action Potential Illustration. Source: `https://en.wikipedia.org/wiki/Action_potential#/media/File:Action_potential.svg` License: CC BY-SA 3.0

The second phase of the action potential is the repolarisation phase. During the refractory period the inside of the neuron is quickly charged with negative electrical charge, often making the membrane potential value a bit lower than the resting state, before it stabilises at the resting state again. As mentioned, during the refractory period a stimulus will generally not lead to another action potential. This is an important property of neural communication to note, when we try to model neural activity.

Action potential in neurons are also known as nerve impulses, spikes or firings. And the temporal sequence of action potentials generated by a neuron is often called ”spike train”. The intricacies of the action potential cycle are useful to better understand model choices and expected neuron behaviour as well as analysing our results. However, when we talk about firings, it is easier just to imagine neurons talking to each other by electrical signals ”fired” towards each-other. Thus, in this thesis we will mostly just use the word ”firings”.

As we have seen a stimulus for a neuron is typically a change in the membrane potential. A stimulus can be either excitatory or inhibitory. Whether a stimulus is excitatory or inhibitory depends on which synapses sends the stimulus. A synapse can be either an excitatory synapse or an inhibitory synapse. An excitatory synapse will send an electrical signal which will increase the electrical charge in the receiving neuron, thus making it likely or at least more likely to fire. While an inhibitory synapse will send an electrical signal which will further decrease the electrical charge in the receiving neuron, making it less likely to fire.

Neural connections Action potential signals can travel throughout the brain in many different ways. A signal from neuron A may take many different paths toward a neuron B. We can say something about how it’s likely that this path is, and thus the relation between neuron A and B, by

the temporal timing between the signals emitted from neuron A and B. The time between a firing in neuron A that causes neuron B to fire, is called the time lag. We the classify the most common underlying physical pathways, i.e. connections, by the time lag.

Common connection We can attribute the connection to a common connection when it's in the [0ms, 5ms] interval. That is, the near simultaneous firing of neuron A and B may be explained by the firing of a third neuron C, which affects them both to fire at the (almost) same time.

Direct connection The connections found in the [5ms, 20ms] interval can be assumed to be directed connections. A lag in this area may originate from neuron A and B being connected in the classical physical sense to neuron B.

Indirect connection Connections found in the [20ms, 100ms] interval can be assumed to be indirect connections. This type of connection is much like directed connection, just that the signal travels through many other neurons before it reaches neuron B.

An example of identifying connection type can be seen in Figure 24, where the black curve is outside the green interval, we are likely to have a connection. Additionally we are able to identify the type of connection by identifying the situation in the figure, at this time. The same can be done with JCC plots as seen in Figure 21.

Neuron types There are also many different types of neurons, two examples are interneurons and principle neurons. Interneurons are a broad class of neurons. They create neural circuits enabling communication between sensory or motor neurons and the central nervous system. Interneurons are mainly inhibitory neurons. Principal neurons are often long-axoned neurons which transmit information over long regions, often from one brain region to another. Principal neurons provide the pathways of communication within the nervous system. Principal neurons are mainly excitatory neurons. Studying the different types of neurons is an important tool to reveal how the information flows in the neurological network.

Neural communication There is a long ongoing discussion about the general information flow in the brain, some believe that the specific temporal timing of the firing is most important while others believe that the frequency rate is most important. Most likely the truth is somewhere in between, that both the temporal timing and the frequency can be of importance, and in varying degree for different cells.

For cortical related cells however, frequencies are presumed to be the most important factor, as decoding the fine temporal firing of the cells are complicated and haven't yielded any useful information yet. However, for song neurons in certain birds; changing the fine temporal firing rate causes the birds to sing different songs. Noteworthy is that changing the temporal tuning causes them to do the roughly same task, but at another frequency. Thus, the finer information may be tuned to temporal coding while the overall task may be tuned to the frequency.

Tuning of cells Neuronal tuning or just tuning of a cell refers to the property of the brain cell by which they selectively represent a particular type of sensory, association, motor, or cognitive information. The response of some neurons are optimally tuned to specific patterns learned through experience as stated in Sakai and Miyashita (1994). At other times however, some cells appear to

have been "hard wired" to their tuning, as it is no apparent pattern for how they have learned their tuning. Neuronal tuning can be strong and sharp, or weak and broad. Some neurons may even be tuned to several different things, such as head direction and speed. Neurons that are tuned to different signals often integrate information from different sources. Finding out more about the functionality of this integration of information is essential in better understanding the brain. Some examples of neuronal tuning are head direction tuning, movement direction tuning, grid cell tuning and spatially tuning.

- Head direction tuned cells are cells that are tuned to the direction of the head. We expect a HD tuned cell to be very active when looking in one specific direction, and then the cell activity gradually decrease as the HD deviates from this direction. Thus, if you plot the probability for firing of HD tuned cell for each angle (called a HD tuning curve), as explained in Section 2.3, we expect to see a Gaussian distributed like curve. An example Figure of the tuned activity of a HD cell can be seen in Figure 13.
- Move direction tuned cells are much like head direction tuned cells, but instead of being tuned to the head direction it's tuned to the movement direction.
- Spatially tuned cells are a class of tuned cells, that are tasked with keeping track of where in space the rat/person currently is.
- Grid cells are a sub-class of the spatially tuned cells.

Spike trains As already stated, a temporal sequence of action potentials generated by a neuron is often called a "spike train". Particularly in this thesis when we refer to a spike train from now on, we will think of it as a digital sequence of information; 1 for a spike and 0 for no spike. Thus an encoded spike train structure could look like, 001001000001000101001. In this thesis we will let spike trains be represented by a discretized m-length Bernoulli process. That is a discrete time zero-one process with outcomes in $\{0,1\}^m$.

Any such sequence can be written as $\mathbf{x} := (x_1, ..., x_m) \in \{0,1\}^m$, where x_j indicates a 1 or a 0 for the sequential position j. Furthermore each bin, x_j, will represent the event occurring during a temporal interval of the length of 1 ms.

1.2.4 Recording neuronal activity

In general the electrophysiological measurements are done by multi-electrode arrays (MEAs). MEAs are devices which contain multiple channels with many electrodes in each from which electrical activity are measured. As explained in Section 1.2.3, when a neuron fires, the extracellular (outside the cell) voltage changes. The MEAs detect this change in voltage and together with the tools of triangulation and neural oscillation identification explained in Section 2.1.1, we are able to detect individual neurons firing moments.

We can partition all MAEs into two classes, implantable and non-implantable. Using implantable MEAs we are able to conduct *in-vivo* experiments. Show example of an appropriate MAE.

In our Data set the electrophysilogical activity in the rats were measured using implanted MAEs which were high-density silicon probes. The silicon probes implanted had eight shanks with 32 recording sites on each shank.

1.3 Outline

Part 1 gives an overview of neuroscience before introducing advanced statistical tools. In chapter 1 we give an introduction to how the brain works and important concepts to understand before modelling brain activity from spike trains. In chapter 2. we describe the data and perform neuroscience analysis of the data set. We detect tuning of cells as best as we can and format the data set for later analysis.

Part 2. presents the statistical theory, with some neuroscience aspects as the method of JGLM has specifically been constructed to find connectivity between cell pairs. In chapter 3. the theory that are just related to the GLM is presented. In chapter 4. we describe the theory that are related to the jittering and JGLM. Additionally some discussion for the choices made in developing JGLM are presented.

Part 3 gives an overview of the statistical analysis of the data set, were the information from Part 1 and Part 2 is used together. Chapter 5. gives a statistical representation of the regression model. In chapter 6. the results from using our method is presented. In the final chapter, chapter 7, we present the discussion for our results and methods and the summary of our thesis.

2 Data description and neuroscience analyses

In this section we present the data set. Additionally we present the processing of the data into an easy to analyse format for our later analysis. Most importantly we present the method used to identify head direction and spatially tuned cells. Most noteworthy of these methods is the Skaggs McNaughton equation for spatial information and the Pearson correlation measure for stability.

2.1 Oliva-16 data set from the Buszaki lab

The data set used was provided by Antonio Fernández-Ruiz at the Buszaki lab and is a part of the data set used in Oliva et al. (2016). The data set contains electrophysiological information measured with two-dimensional silicon probe arrays, and recording of movement from the rat which was free to roam around a small box of roughly 46x53 cm in dimensions.

In Oliva et al. (2016) the oscillatory part of the data set was analysed. We will in this work analyse the neuron activity of identified cells.

This data set is of particular interest since neurons from brain parts where head direction tuned cells (are assumed to) reside as well as the area where grid cells and spatially tuned cells resides are recorded. This makes it possible to look at the connectivity and relation between head direction and spatially tuned neurons. The individual cells and their firing times where added to the data set at Kavli Institute in Trondheim.

2.1.1 Spike and movement data

The data contains electrophysiological information from cells from hippocampus from a rat, collected with two-dimensional silicon probe arrays. The probe measured with a frequency of $20kHz$. The data were recorded from the areas CA1 (Cornu Ammonis Field 1), CA2, CA3, DG (Dentate Gyrus) and LM (Latromedial cortex) which are sub-regions of the hippocampus.

The silicon probes implanted into the rats brain act as microphones in a huge football stadium. That is, they detect the electrophysiological activity in a area of the brain from many cells. One microphone may only detect noise from one area of the stadium, but by placing many microphones close together in a so called cluster, we may detect individual voices. Therefore the microphones are grouped together in four, which is either called cluster or tetrode. Each person (neuron) in the football stadium (brain) has a specific position, and thus a specific distance from each of the microphones (electrophysiological measuring devices), but only the microphones from the nearby cluster are close enough to detect the volume of a person's voice in the crowd. We can utilise this concept mathematically by what we call triangulation to identify certain (loud speaking) individuals in the crowd by their distance from the microphones. However, identifying specific neurons/persons from triangulation alone is potentially faulty as we may erroneously identify a person. Reasons for erroneous identification may be from symmetry properties, such that two persons are at the same distance from every microphone in the cluster, yet at different positions. This may lead to several persons being identified as one, or that one person is identified as several persons. The voice volume of a person may vary, so we may misinterpret the distance from the microphone to the person. Thus, we would like to identify individual persons not only by their distance from the microphones, but also from their individual voices. The tone of voice analogy for a neuron is called neural oscillation. Neural oscillation is that each neuron has an oscillatory activity pattern for their electrical activity, and by using information about the neural oscillatory properties of

electrophysiological measurements together with triangulation we can better identify individual neurons. This part of the processing of the data, where activities of individual single cells were identified by triangulation and neural oscillation patterns, were performed at Kavli Institute in Trondheim, and is together with the movement data the basis for this thesis.

Identification of firing times of individual cells is not perfect. Erroneous identification can still occur with both triangulation and neural oscillations as identification tools. Thus, we will in this thesis, when we compare cell pairs, only compare pairs from different tetrodes. Different tetrodes will have sufficient distance from each other, so we are certain that the cells compared are from different neurons.

The data set also consists of movement data. The movement data was recorded by placing a rod on perpendicular on the rats head, and on each end on the rod there was a LED coloured ball. The position of these two balls were identified by a camera on a frequency of $f_{\text{Move}} = \frac{1250}{32} Hz \approx 39 Hz$.

To summarise the data set consists of:

- Spikes from individual cells from many tetrodes.
- Movement for a rat with many identified cells.

2.1.2 File formats

The data set consists of two sessions. We have chosen the session called AYA9day16, as the quality of the movement data in this session was far superior to the other session called AYA9day20. Movement data quality is of importance as we would like to analyse the connectivity between spatially tuned cells and HD tuned cells, see Section 1.2.2 for explanation of tuning. In all other aspects the size and properties of the two sessions are similar.

The file formats of the firing data For the AYA9day16 data the most interesting files are the single `whl` file which contains the movement data measured, and the many `res` and `clu` files which contain the firing moments of the different cells that have been identified. "`Res`" is abbreviation of *resolution* and these files contain information about the firing times of neurons in a cluster. "`Clu`" is abbreviation of *cluster* and contains the cell number for the cells that fired in the cluster for the corresponding time in the `res` file.

Thus, the `res` and `clu` files goes together to describe the cell activity. For the session AYA9day16, there are 10 `res` files and 10 `clu` files. The set of `res3` and `clu3` describe the data from tetrode 3, the same holds for tetrode 4-12. The `res` file contains a single vector with length equal to the total number of the firings detected by the tetrode, thus the first element is the first time-point of the firing detected by this tetrode. As each tetrode is likely to detect many different cells, the cell number for a given firing time is given in the corresponding `clu` file. The `clu` file contains a vector with length one longer than the `res` file, the first element is the number of active cells detected by the tetrode. The other elements in the file are the cell numbers of the firing cells. The second element in the `clu` file corresponds to the first element in the `res` file. It is important to note that the time given in the `res` file is not in seconds or ms but in measurements from the silicon probe with the frequency of 20k Hz, thus a value in `res3` equal to 20k means that it was a firing for a cell 1 second after we started measuring the cell activity.

The file formats of the move data The `whl` file contains data stored in a 4x612351 matrix we refer to as X_{OLD} (Original-LED-Data). The 4 columns correspond to the x-coordinate & y-coordinate $(x_{\text{L}}, y_{\text{L}})$ for the left most LED ball on the rod and the x-coordinate & y-coordinate $(x_{\text{R}}, y_{\text{R}})$ for the right most ball on the rod. The row-dimensions is the time-dimension. Thus, measured with a frequency of f_{Move} we have that the total time measured with the movement camera were $\frac{612351}{f_{\text{Move}} * 60} \approx 4.35$ hours.

From the original four columns of the LED data we can construct the position, speed and head direction of the rat.

- The position of the rat's head given as the mean of the two LED balls positions. We estimate the rats head position as $(x_{\text{Head}} = \frac{x_{\text{L}}+x_{\text{R}}}{2}, y_{\text{Head}} = \frac{y_{\text{L}}+y_{\text{R}}}{2})$.
- The speed data is calculated from the position data, by the one-step central difference method.

$$v_i = \sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2} \tag{1}$$

  Where $i = 1, ..., n$ where i is the i'th measurement of time according to the `whl` file and n is the number of time measurements.

  There are other possible solutions that may yield a better estimate for the actual speed at the given time such as using more steps or a smoothing mechanism to makes the speed vector look more similar to the way we expect a rat to move. Smoothing may yield more precise data but it also include another layer of subjectivity into the data analysis, with a rather small potential gain. Thus, we leave the data as it is. For the multi-step difference method it may yield a better result. However, as the movement data vectors already contains many invalid position measurements (recorded as NaN) we would like the speed measurement to depend on as few as possible neighbours.

- The head direction (HD) data, gives the direction the rat's head is facing. This can differ from the move direction (MD) in which the rat is moving, and from the attention direction (the direction the rat is looking). From the original LED data we can calculate the HD and MD but not the attention direction. However, the tuning of cells to HD from MD has already been tested Raudies Florian et al. (2014), and it's shown that we expect cells to be better tuned to HD.

  In order to calculate the HD we first calculated the difference between the y_{L} and y_{R} and get a y_{Diff}, and we do the same for the x. By the sign of the y-diff and x-diff variable we are able to tell which Cartesian coordinate quadrant the head is looking, and given that we know this we can now use the tangent rule to find the exact angle that the rat is facing. We can do this for every time-point to get the angle at every time-point. The formula for head direction angle in radians is,

$$\theta_{Head} = \pi/2 + I(x \geq 0)\pi + \arctan\left(\frac{y_{\text{Diff}}}{x_{\text{Diff}}}\right) \tag{2}$$

  where $I(x \geq 0)$ is the indicator function which are 1 if $x \geq 0$ and 0 otherwise.

Useful to note is that the data amount in time and cells are both a lot by neuroscientific standards.

2.2 Preprocessing by binning

The reason why the binning is necessary is further explain in Section 5.1, while the binning process itself is explained below.

Binning We will in this thesis choose 1 ms bins were we only allow for either 0 or 1 firing in each bin. Thus, our binned vectors will be Bernoulli processes. Furthermore, for binning of 1 ms we have sufficient precision for the refractory period, explained in Section 1.2.3, to be estimated into our model, this is more thoroughly explained in Section 5.1.

To bin the firing data we first construct an empty binned firing vector for each cell with all values equal to 0. Each bin in this vector should correspond to the firing activity for the cell for the millisecond in the session that match its index. If the value is 0, there is no activity during this millisecond for this cell. If the value is 1 it was firing activity during this millisecond for this cell.

In order to do this we first isolate the firing for an individual cell from the firing times for all the cells in the cluster. Thus we firstly locate the index for a certain cell in a cluster, then we use this index to find the firing times for this specific cell. Since, the first element in the `clu` file is an overview element of the data, we need to take minus 1 from the index found by identifying `clu` and use it to find the time in the `res` file. The time is given in the frequency scale it was measured in, thus we have to divide it by 20 and round it up to the nearest whole millisecond. This number, is the closest millisecond for the specific cell firing, thus the new binned firing vector element with index equal to this number should have the value of 1, while the other elements in the corresponding binned firing vector should have the value zero.

We do this for all the cells, and place the vector for each cell as rows into a firing matrix. The cell number, tetrode number, and frequency for the cells are stored in separate individual vectors (`cellCounter`, `tetCounter` and `freqCounter`) with matching index to the index of the column in the firing matrix to keep a quick overview over the firing data available.

The move data is actually already binned, but with 39.1 ms bins, thus it needs a rebinning at a higher frequency than the original data. The rebinning is done by simply keeping the original position measurements, this can be done as the movement change within the 39.1 time interval is very small. Thus, the newly binned position data value for a time-point is the value of the closest original position data point in time.

As the rebinned position matrix is the basis for the construction for the HD, speed and position vectors, we do not need to do any more binning for these.

Exploring the firing data We display the information in the `res` and `clu` files by raster plots, see Figure 7 and Figure 8.

Different cells have quite different firing rates. Also the cell identification process is not without uncertainty, so it is still useful to remember that one cell may be identified as several and vice versa. Thus, if we identify raster plots that have a complete stop in firing for a period when another cell has an abrupt start, it may be the same cell.

Exploring the move data We would also like to evaluate the quality of the movement information given in the `whl` file. Firstly, for roughly 14.5% of the measurements the tracking camera was not properly able to detect the position of the LEDs, at these time points the data position is registered with a NaN-value (Not a Number).

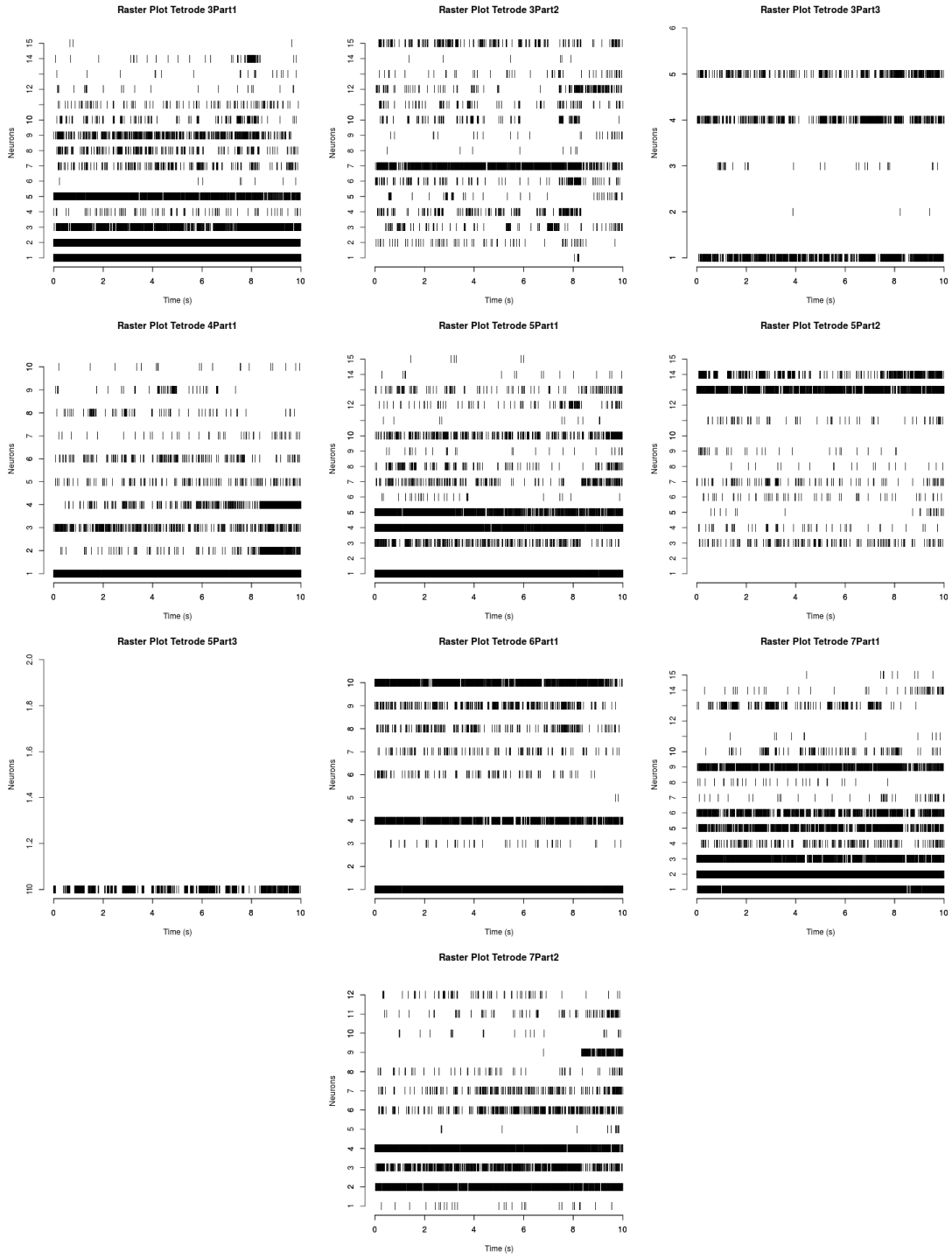

Figure 7: Raster plots for 10 first minutes, for tetrode 3-7. Part 1, up to the first 15 neurons for a tetrode, Part 2 the 16-30th neuron for a tetrode and Part 3 31-45th neuron for a tetrode.

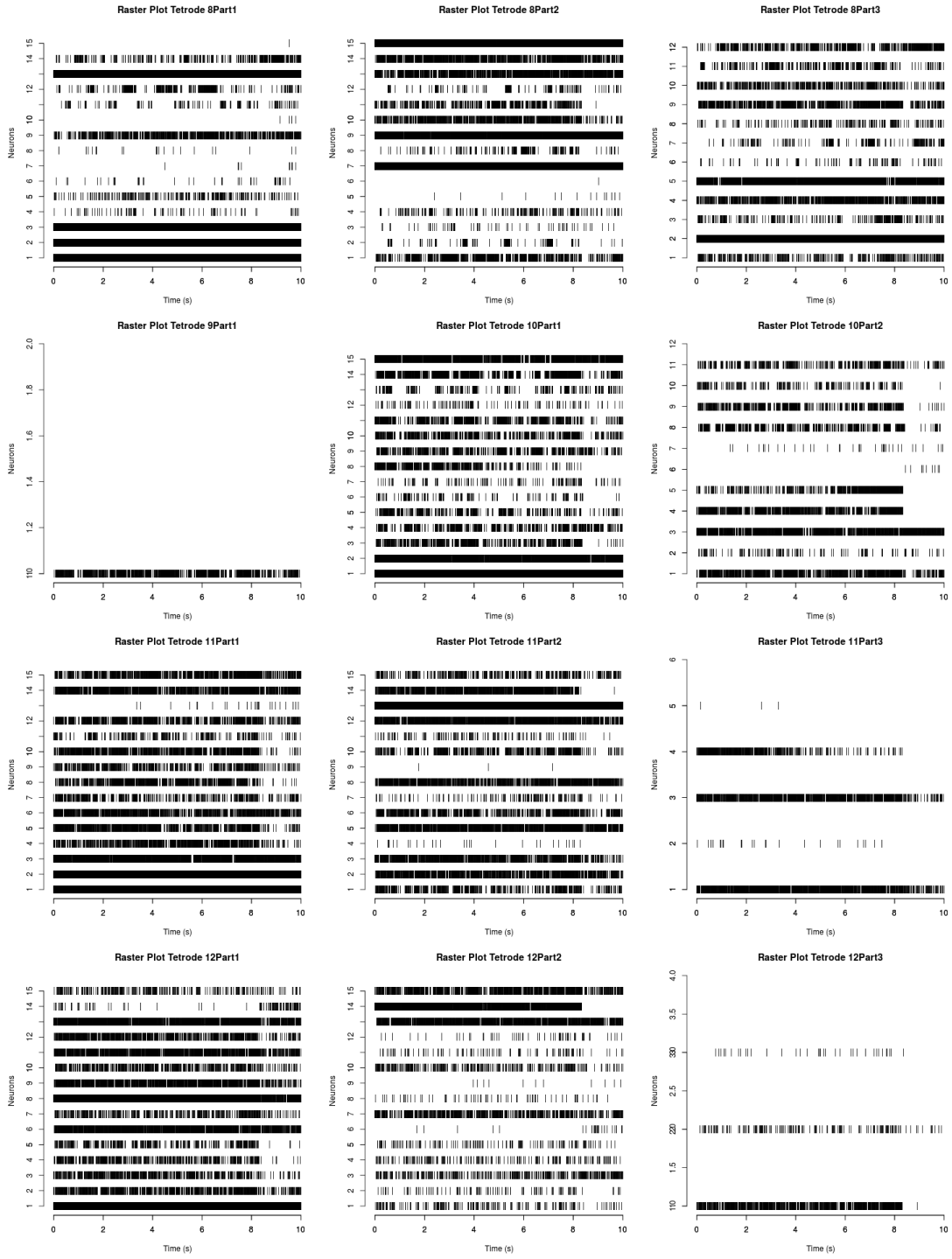

Figure 8: Raster plots for 10 first minutes, for tetrode 8-12. Part 1, up to the first 15 neurons for a tetrode, Part 2 the 16-30th neuron for a tetrode and Part 3 31-45th neuron for a tetrode.

In our case good quality of the movement data would mean that the rat is for the most parts moving with a speed of at least 5 cm/s. Also, we would like the movement to be somewhat evenly distributed throughout the box, and the movement patterns to be somewhat resembling a random walk. Such behaviour would describe a rat moving about and discovering the box, without seemingly being influenced by outside factors.

To check if our data somewhat resembles our "perfect" movement data we first plot several position/movement plots of the rat in the box, see Figure 11. From these plots we get a basic idea that the rat seems to be moving around most parts of the box, however it also seems to be spending much more time in certain areas than others. It may seem that the rat at times moves in some weird pattern-like triangles. To further investigate the movement data properly we constructed a movement video for all of the session time. Two of the videos have been uploaded to

`https://giphy.com/gifs/movement-rat-kristian-l0HUlk3ygM3ZVQcA8` and

`https://giphy.com/gifs/master-rat-kristian-3o752mjHMvTBg1mdzO`.

The first link gives the movement of the centred head position for the 0-10 and 90-120 min interval. The speed is scaled up 20 times from real life time. The second link represents the rats movement slightly faster than real life data, the red/green line represents the centred head position, while the grey line represents the right ball on the rats head. The colour of the centre-line is green when it moves above the speed threshold and red while it moves below.

Some parts of the movement data were not good according to our quality criteria. The most unsatisfactory issues with the movement data are as follow:

- The box the rat moved around in was rather small, namely 53x46cm in dimensions. Normally box environments for grid cell detection are much larger as the distance between the peaks of the grid cells are often detected to be roughly 30cm apart.
- The periods the rat is actually moving around the box is just a small part of the original data set time of over 4 hours. There are a few short periods with actual movement. Only two good longer periods of time were found. Namely the 0-10 min interval, and the 90-120 min interval. The smaller well behaved movement pockets outside these two intervals were often just lasting a few seconds. The basis functions used for connectivity and history effects, see Section 3.4, demand some element of continuity. Thus the potential benefit from adding these small pockets of well behaved movement time were deemed small.
- Even for the time in accordance with our quality criteria the rat was not moving in a very random walk like fashion, as we would like. Instead, the rat moved in repetitive movement patterns, where it did not cover the full area of the box very well.

As we have a lot of data, yet limited good movement data, we will for the main part in this thesis use the data from the time periods when the rat's movement is good. That is, from studying the movement video we choose to only look at and analyse the data from the time period of 0-10 min, and 90-120 min. This applies both to the movement data, and the firing data. This is an important step in order to improve the quality of the all movement related analysis, especially that of identifying grid cells. Identifying grid cells proved difficult and is further explained in Section 2.3.1. Only for some parts of the exploratory analysis will the whole session time be used.

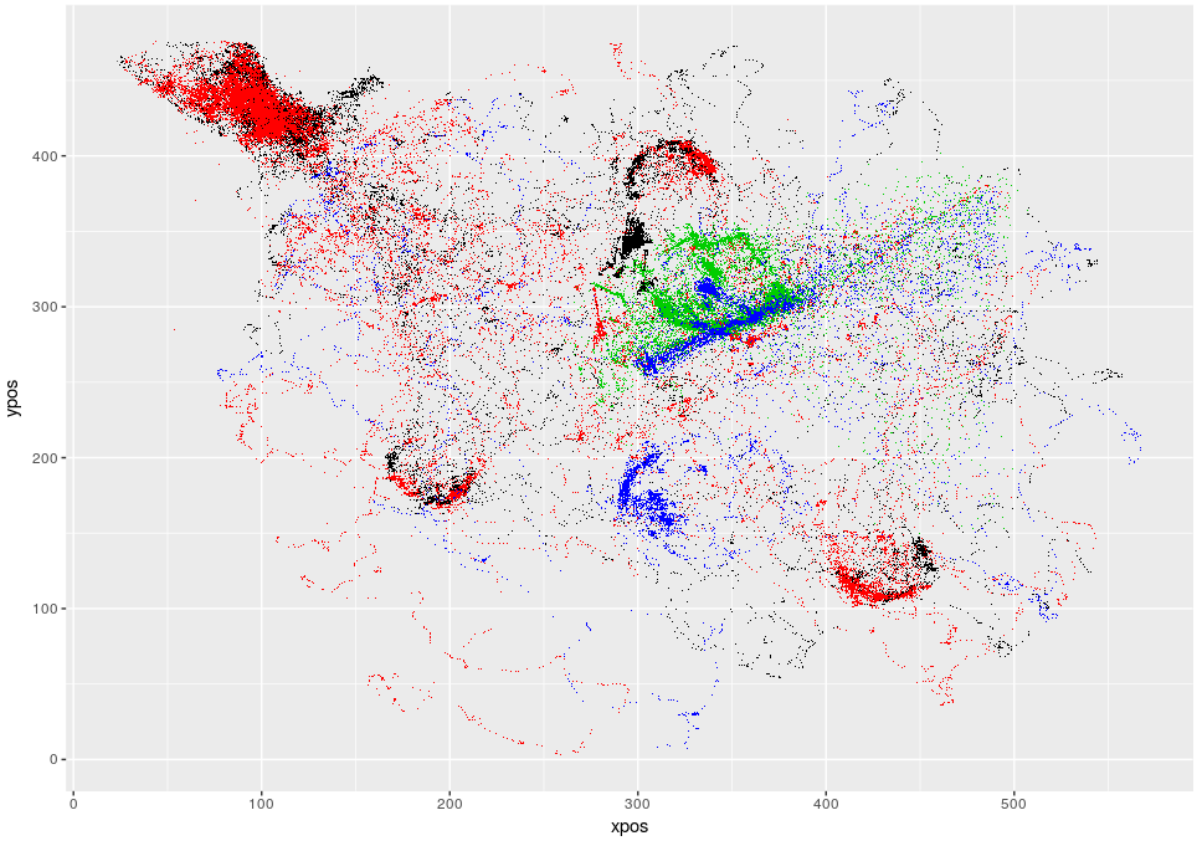

Figure 9: A point plot of the positions the rat was at. The colour changes with the time the rat was at the given position, red represents the first hour, black the second, blue the third and green the fourth hour. One point for each time measurement. We can see that the rat have been around most of the box, but spent less time close to the edges of the box.

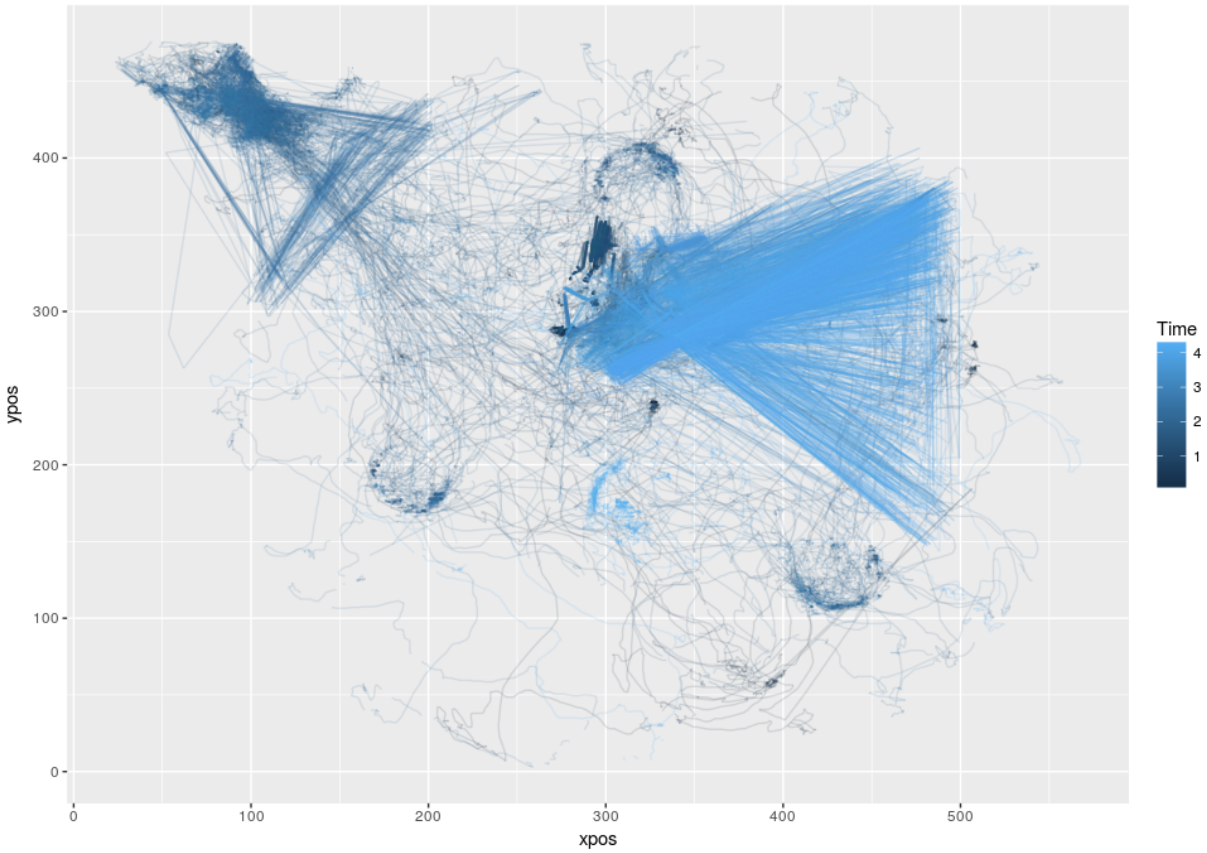

Figure 10: A plot of the rats movement. The path of the rat analysed for time span of whole session. Time is given in hours. We can see that the rat have been around most of the box, but spent less time close to the edges of the box. Furthermore it has spent its time in the box quite uniformly, expect for two repeating triangle movement patterns in the upper left corner and in the middle to the right of the box. These two confusing areas are most likely due to tracking errors, such as reflections from the wall off the LED lights placed on the rats head.

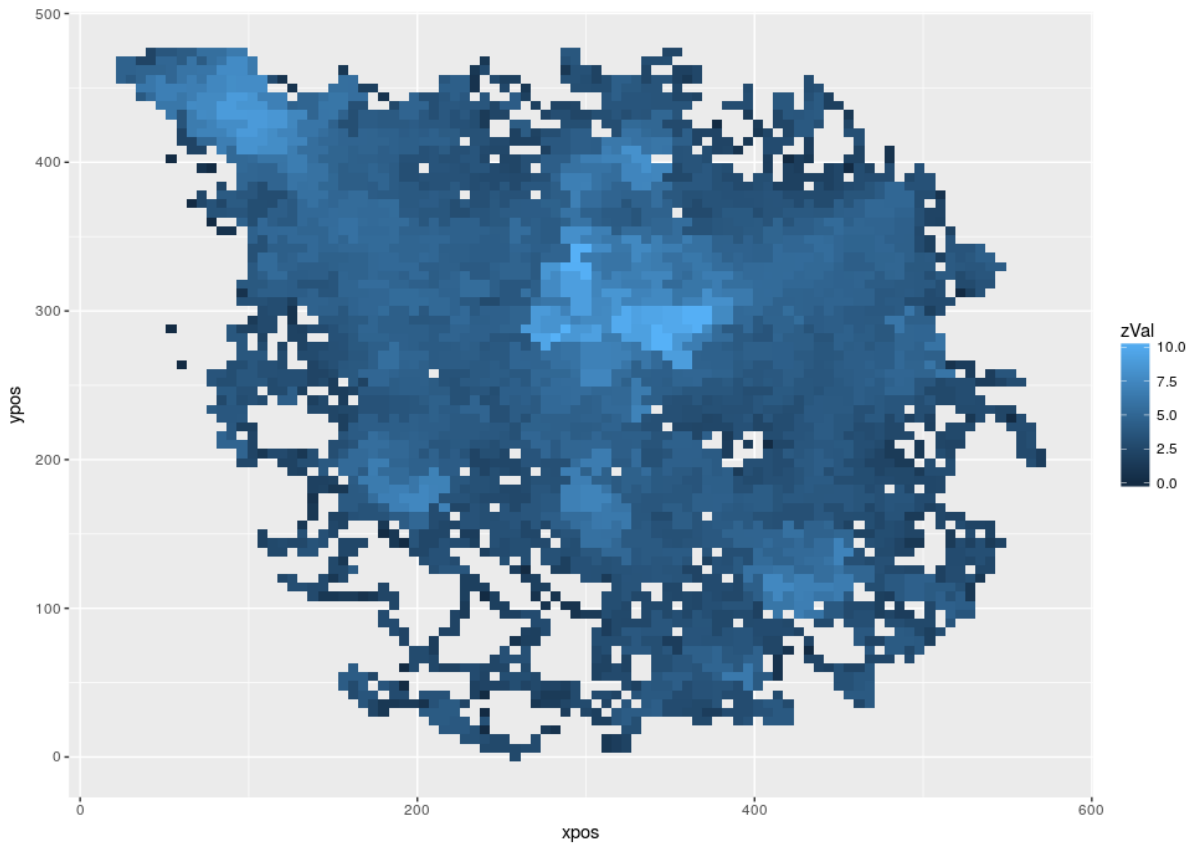

Figure 11: First plot is a heat map of where the rat spent its time in the box. The path of the rat analysed for time span of whole session. The colour/zval gives the probability for the rat spending time in a bin. The probability scale is given as the average number of times it was measured to be in a bin per second. Bins that are grey/colourless are places that rat haven't been at all.

2.3 Identification of cells

This data set has potentially measurements of neuronal activity from both HD tuned cells and spatially tuned cells. We would like to use this opportunity to further look at the connections and relations between these two types of cells. First we investigate if we are able to detect if we have any of the two cell types among our measured neurons.

To aid us in identifying HD cells we will plot HD tuning curves in Section 2.3.2, and for spatially tuned cells we will plot *rate maps* in Section 2.3.1. Additionally we will also plot the more analytic plots of *spatial information* and *stability* for HD cells and spatially tuned cells, see Section 2.3.3

Spatially tuned cells are cells that in some way are tuned to the spatial position of the rat. One specific type of spatially tuned cell are the grid cell. We would like to identify grid cells from our data, however the movement data is not good enough to detect if there are any grid cells or not. Yet the movement data is good enough to detect if the cells are spatially tuned, so we detect spatially tuned cells instead.

2.3.1 Spatially tuned cells

To help us identify spatially tuned cells we plot rate maps. Rate maps or heat maps are maps of the box where we bin the area and plot the activity in this area with a colour according to the activity level in the bin. In our case the activity level we are looking at could be position, overall firings, and probability for firing given the position.

A The heat map for position shows how much of the time the rat spends in the given bin/area. This have already been plotted in Figure 11.

B The heat map for overall firing activity shows how many times the particular cell fires when the rat is at the given area.

C The heat map for "probability for firing" shows the frequency for a specific cell to fire in a bin, given that the rat's position is in this bin. This case could be understood as the heat map of case A divided on the heat map of case B. That is, case C is the number of firings in the binned-area but adjusted for the time spent in the area, so it becomes an estimate of the probability for the cell to fire in the binned area given that it is already there.

At first rate maps were made for all of the session time, however in order to improve our prediction we later used only the shorter time periods where the movement data was known to be good, as explained in paragraph "Exploring the move data" in Section 2.1.2. Thus, the uncertainty in the position measurements of the rat were as low as possible, for the data used, and our predictive value as high as possible.

Rate map C was constructed by binning the area of the box into 18 x 18 bins, thus each bin covering an area of roughly 2,5 x 2,5 cm. Then the period the rat spent in each bin was registered. Thereafter the number of firings each cell fired in each bin were counted. Then for each cell the number of firings in a bin were divided by the time spent in the bin, in order to get the frequency for firing in a bin given the rat is there. However, this was only done for 0-10 min and 90-120 min interval, were the movement data is good, as mentioned already. Furthermore the speed of the rat needed to be above 5 cm/s to be considered good and the occupancy time spent in a bin to be above 0.5 s in order for us to look at that bin at all.

The speed criterion is included as we expect the cells that are HD and spatially tuned to only show this type of tuned behaviour when the rat is moving with a speed above a certain threshold.

We set a occupancy limit for each bin, if the time spent in a bin overall is less than 0.5 s, we do not include this bin in the rate map or other part of the analysis. This limit is included since the variance of the probability of the firing rate in a bin is too high when the occupancy is low. If the rat is only in a binned area for 0.1 ms, yet it fires once at this time, it will have an estimated firing rate of 10k Hz, while if it doesn't fire at this time the estimated firing rate would be 0 Hz.

We will mainly look at rate maps of type C, from here on this is referred to as *rate or heat maps*. Rate maps are essential in our analysis to identify spatially tuned cells.

Preferably we would like to identify if the spatially tuned cells were grid cells or not. One way to identify grid cells is by subjectively looking at the rate maps. If the rate maps have two or more distinct activity peaks for a spatial location, we can identify it as a grid cell. However, from Figure 12 (example rate map), it's difficult to clearly identify activity peaks/areas for a cell for our data. This figure may or may not show a grid cell, as the activity represented by the rate map is not uniformly distributed nor does it have several clearly distinct activity peaks. Thus, it is difficult to identify with certainty a grid cell. However, the movement data is still good enough to tell whether the cells were spatially selective or not. It's rather clear from Figure 12 that cell T3C1 is spatially tuned as the colour at different position is far from constant. However, to identify if cells are spatially tuned or not by just looking at rate maps are highly subjective. Thus, we will try to make our identification in a more rigorous analytic fashion, as we will further explain in Section 2.3.3.

2.3.2 HD tuned cells

To help us identify HD-cells we will plots a HD tuning curve (TC) for each cell. A HD TC is similar to a rate map, but now only for the angle of the head variable, instead of the 2-dimensional space variable.

For each cell we bin the angle measurements into 18 bins covering the area of 0-360 degrees. Thus, the first bin covers the area of 0-20 degrees, the next 20-40 degrees and so on. As for rate maps we can divide the construction of the HD TC into a case A, case B, and case C. In this setting the new cases can be explained briefly as follows.

A We can make an occupancy TC. This keeps track of how much time the rat looked in the direction of each HD-bin. These plots are independent of cell activity and is the same for all cells.

B We can make a firing number TC. This keeps track of how many times a single cell has fired in a HD bin.

C Finally we can make the proper HD TC, which both counts the number for case A, and case B for each bin. Then we divide the B value on the A value to get the frequency for firing in a bin for a cell given that the rat is already facing in that direction.

If the HD TC is clearly peaked, we identify it as a HD cell. However, as for rate maps this is a somewhat subjective way to identify HD cells we would also like to do so using spatial information and a stability measurement further explained in Section 2.3.3.

An example of a HD TC case C, can be seen in Figure 13.

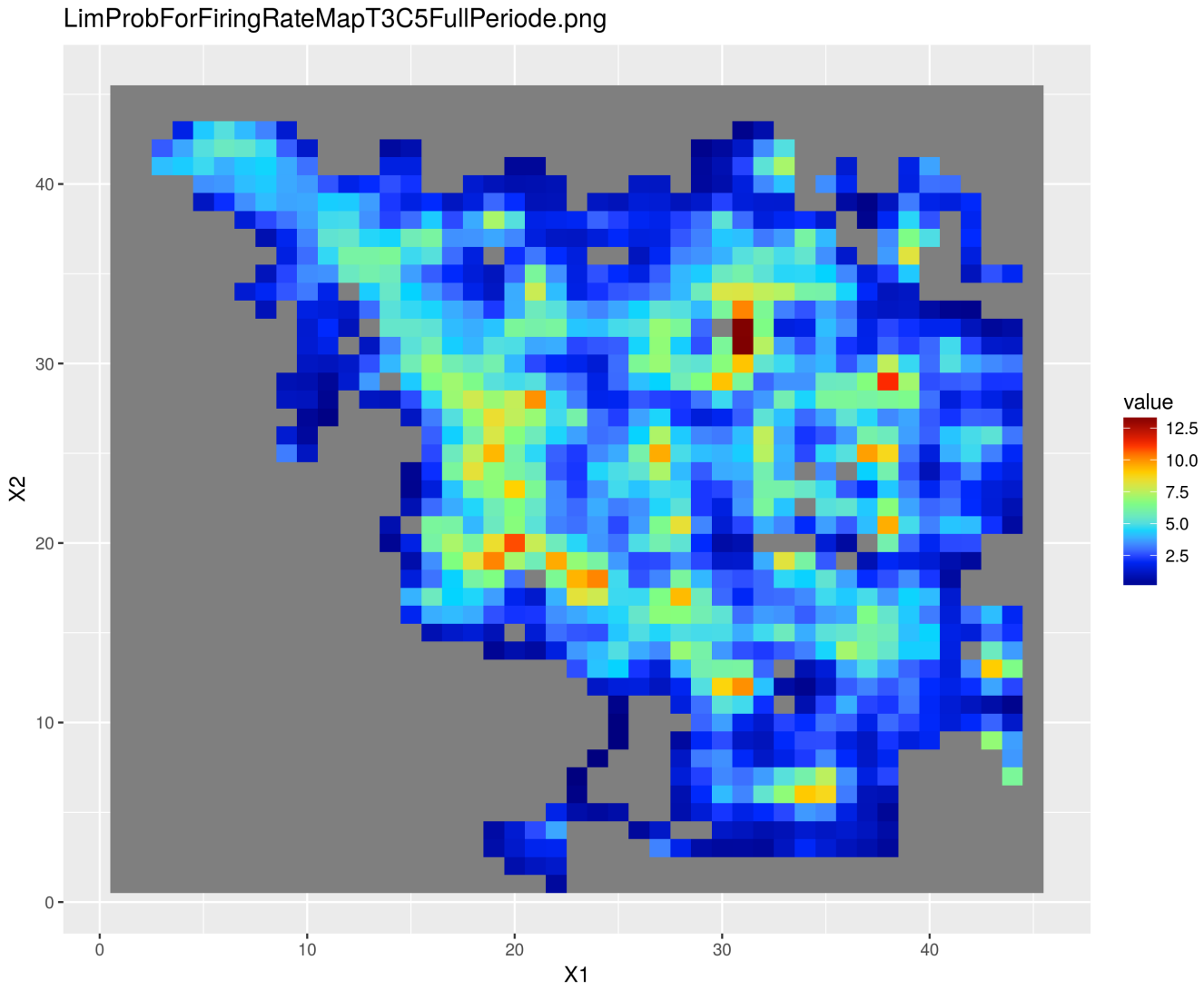

Figure 12: Heat map case C, see Section 2.3.1, for T3C5 for all the time of the session. Example for how they would all look. Tetrode 3 cell 5 is randomly chosen. The colour-coded value-scale on the left side represents the frequency in Hz for a given bin.

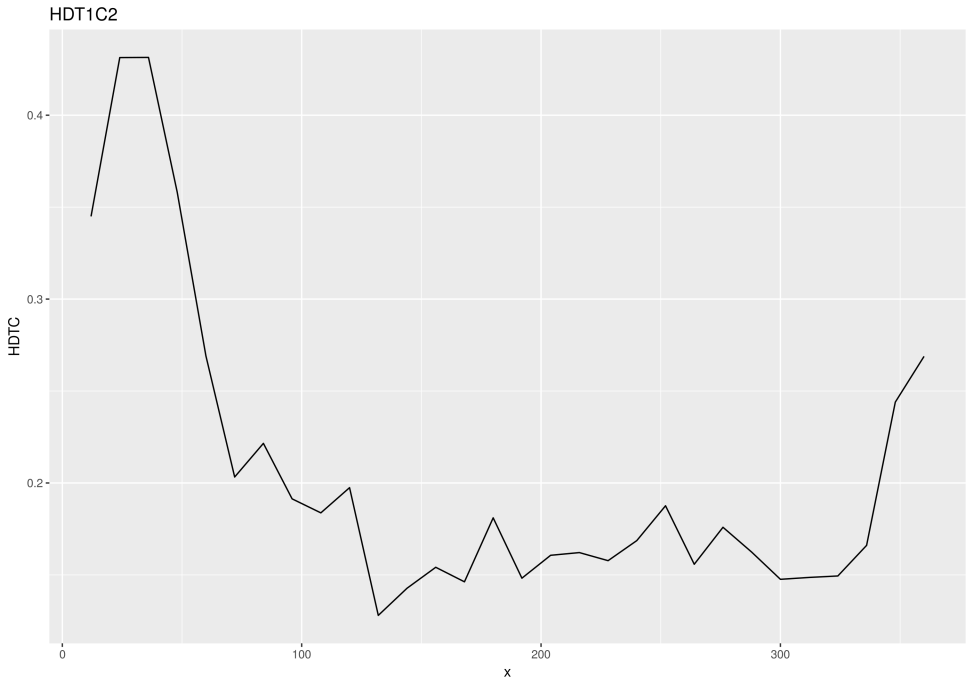

Figure 13: HD TC case C, see Section 2.3.2, for Tetrode 3 Cell 2. The x-axis gives the HD in degrees and the y-axis gives the estimated firing rate in Hz for bins of 10 degrees interval size. This cell is well tuned to HD, but got an overall low firing rate.

2.3.3 Spatial information and stability

With the aim to classify neurons as HD and spatially tuned cells, we analyse their *stability* by measuring the difference in tuning at different times and *spatial information* by Skaggs and McNaughton equation.

Stability The stability is analysed by splitting the part of the data with good movement data, namely the 0-10 min and 90-120 min, into two sub-intervals. The first sub-interval is the data from 0-10 min and 90-100 min and the second part is the data for 100-120 min. Thereafter we construct individual rate maps or HD TC respectively for ST and HD tuning, as explained in Section 2.3.1 2.3.2, for each of these two sub-intervals. After we have constructed these plots we compare the rate maps for ST to each other and the HD TC to each other, point-wise, by taking their matrix (or vector for HD) representation and finding the Pearson correlation coefficient between the matrix elements. Important to note is that the occupancy is different for the two rate map images for ST. In order for the stability analysis to be fair we only treat bins that have sufficient occupancy for both of the subsets to be analysed. We have set our sufficient occupancy to be 0.5 second spent in a bin.

The formula for the Pearson correlation coefficient is

$$\rho_{X,Y} = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}} \tag{3}$$

Where $\mathbf{X}$ and $\mathbf{Y}$ are two vectors whose i'th element are written as X_i and Y_i. The mean value of $\mathbf{X}$ are $\bar{X}$ which is defined as $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, and correspondingly so for $\bar{Y}$.

The Pearson correlation coefficient measures the correlation between the two vectors $\mathbf{X}$ and $\mathbf{Y}$. If these two vectors each represents an image, rate map or a curve. The Pearson correlation coefficient is high when there is a similar trend between the two images, and if the Pearson correlation coefficient is close to zero they appear to have no relation with each other, while if Pearson correlation coefficient is negative the two images, are opposites. What we expect and would like to see for a cell that is either tuned to space or HD is that it has a high PCC, indicating that the behaviour for the cell is similar at different times.

In our case $\mathbf{X}$ will represent either the rate map for ST, $\mathbf{X}_{ST}$, or the HD TC, $\mathbf{X}_{HD}$, for the first sub-interval period, while $\mathbf{Y}$ correspondingly represents the same for the second sub-interval period.

Thus from the general equation for stability measure by PCC (3) we get two specific equations for our case, one for ST and one for HD.

$$\rho^j_{X_{ST},Y_{ST}} = \frac{\sum_i (X^j_{i,ST} - \bar{X}^j_{ST})(Y^j_{i,ST} - \bar{Y}^j_{ST})}{\sqrt{\sum_i (X^j_{i,ST} - \bar{X}^j_{ST})^2 \sum_i (Y^j_{i,ST} - \bar{Y}^j_{ST})^2}} \tag{4}$$

$$\rho^j_{X_{HD},Y_{HD}} = \frac{\sum_i (X^j_{i,HD} - \bar{X}^j_{HD})(Y^j_{i,HD} - \bar{Y}^j_{HD})}{\sqrt{\sum_i (X^j_{i,HD} - \bar{X}^j_{HD})^2 \sum_i (Y^j_{i,HD} - \bar{Y}^j_{HD})^2}} \tag{5}$$

Here the j-notation represents that the this is done for the j'th cell. Thus, $\rho^j_{X_{ST},Y_{ST}}$ is the Pearson correlation coefficient measure for the stability between the two sub-interval rate maps

X^j_{ST} and Y^j_{ST} for cell j. And likewise we have the same for HD tuned cells and HD TC PCC $\rho^j_{X_{HD},Y_{HD}}$.

Thus, for each cell we get two PCC-values, one for the stability of the HD tuning and one for the stability of the spatially tuning of the cell. Thus, we get two vectors of size 1x250 (250 equals number of cells), $\boldsymbol{\rho}_{HD} = \left(\rho^1_{HD}, \rho^2_{HD}, ..., \rho^{250}_{HD}\right)$ and $\boldsymbol{\rho}_{ST} = \left(\rho^1_{ST}, \rho^2_{ST}, ..., \rho^{250}_{ST}\right)$ that is the stability measurement for HD and spatially tuned (ST) cells.

Spatial Information Information theory treats the cell as a communication channel where the input is the measured variable, which in our case is the rate map for position or the HD TC for a cell, and who's output is the cell's spike train. The Skaggs and McNaughton equation here presented, and derived in Skaggs et al. (1993), is an information theory approach to find systematic differences in the information content of spatial cells. Therefore, what we measure by Skaggs and McNaughton equation can often be referred to as spatial information or mutual information (the mutual information between spatial tuning and spike train). Here we will refer to it as spatial information.

In our case the equation for spatial information (SI) can be rewritten as.

$$I^j_{SM} = \sum_{r \in bins} \lambda^j(X_i) \cdot p(X_i) \cdot \log_2\left(\frac{\lambda^j}{\sum_{r \in bins} \lambda^j(X_i) \cdot p(X_i)}\right) \tag{6}$$

Where I^j_{SM} indicates the spatial information by Skaggs McNaughton equation for cell number j. $\lambda^j(X_i)$ indicates the firing frequency rate for cell j given that the rat is at position (or angular direction) X_i. $p(X_i)$ indicates the probability for the rat being at position (or facing angular direction) X_i. λ^j indicates the firing rate for cell j over all bins.

Construction of Plots When we have a stability number of the rate maps and the SI number for the mutual information between the spatial tuning of a cell and its firing we can plot stability and SI overview plot, as seen in Figure 14.

What we now have done for spatially tuned cells and rate maps can be done for HD TC and HD tuned cells. The results can also be seen in Figure 14.

The information from the plots When we look at Figure 14 we see that we identify more or less the same cells with the stability-SI-plot from HD and spatially tuned cells. A part of the reason why this is the case is likely to be because of the low quality of the move data. In order to be able to analyse the connections between the HD and spatially tuned cells, it would be preferred if the two subsets of cells with highest stability-SI-value for HD and ST cells are as different as possible. However, the data can still yield interesting results as well as help in develop interesting methodology for statistical analysis.

Using the information from Figure 14, we can choose the highest value of SI and stability for HD to identify the (most) HD tuned cells. Furthermore we can do the same for the spatially tuned cells. Thus, we get a subset of the original cells that we will further analyse. These being the HD tuned cells and the spatially tuned cells. A list of number of the identified cells can be found in Table 1.

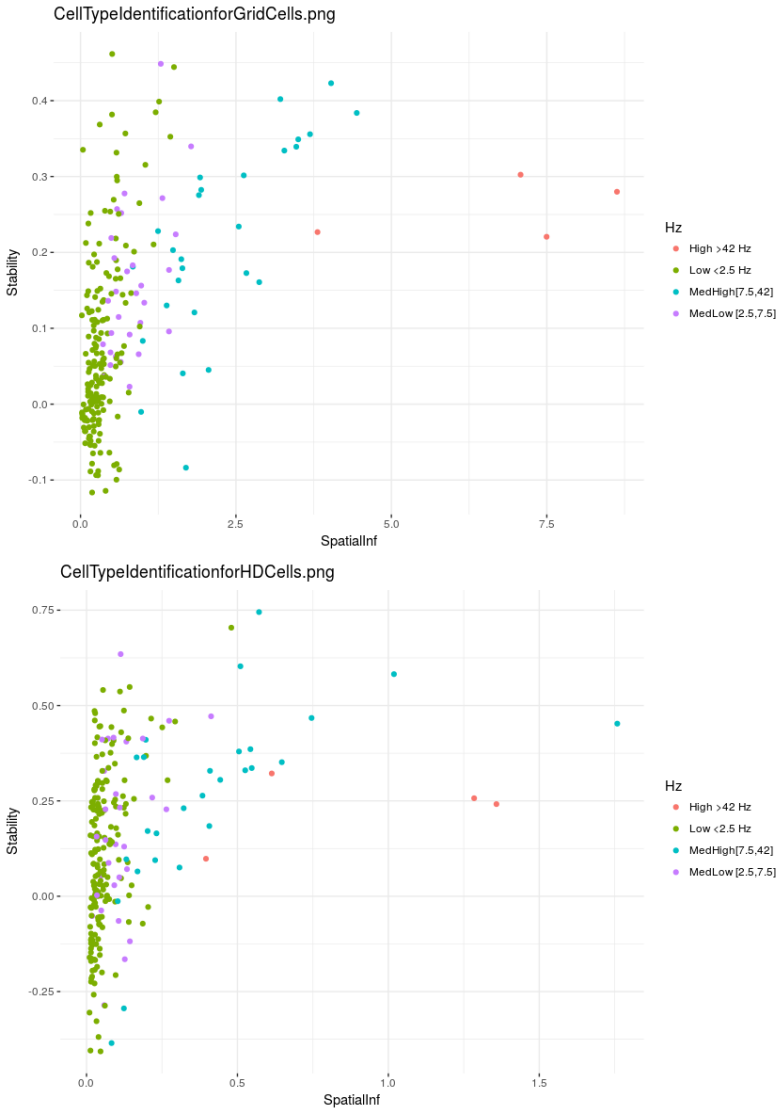

Figure 14: Here we see point plots where the spatial information is plotted again the stability value. For each cell we have calculated two HD related quantities, the stability value ρ_{HD} and the spatial information value I_{HD}, two spatially tuned quantities, the stability value ρ_{ST} and the spatial information value I_{ST} and the frequency. The two Spatially tuned quantities are represented in the first plot, and the two HD related quantities are represented in the second. Each point in each of the plots represents a cell, and its corresponding values. Additionally it has been coloured coded according to its frequency, high frequency is coded by orange, medium to high is coded by blue, medium to low is coded by purple and low frequency is coded by green. In both plots we can see that high spatial information for HD cells are positively correlated with high stability value for HD and vice versa for spatially tuning. Additionally, cells with higher firing rate have a tendency to have higher spatial information and stability values.

Tetrode	3	4	5	6	7	8	9	10	11	12
Total Cells	35	10	31	10	27	42	1	26	35	33
HD Tuned cell	2	1	1	-	-	2	-	1	1	1
Spatially Tuned cell	1	1	-	1	-	-	-	-	1	-

Table 1: Overview of the total number of cells, the identified number of HD cells and spatially tuned cells based on stability and spatial information point plot with cells with higher value than 4 in spatial information for spatially tuned cells and 0.56 for HD tuned cells.

Nr	1	2	3	4	5	6	7	8	9	10	11	12
HD	T3C2	T3C30	T4C3	-	T5C18	-	T8C3	T8C39	T10C2	-	T11C28	T12C13
St	T3C2	-	-	T4C4	-	T6C2	-	-	-	T11C2	-	-

Table 2: Overview of the exact cells that were identified as HD and ST tuned. This subset of all cells, will later be used as a subset for our analysis of connectivity between cells, discussed in Part III. Additionally the brain regions these cells comes from are described in Table 3.

Brain Area	CA1 layer	DG layer	CA2	Entorhinal cortex
Cell Number	1,2,3,4	5,6	7,8	9,10,11,12

Table 3: Table of the brain areas the cells from Table 2 resides in. The brain areas and structure are commented in Section 1.2.1.

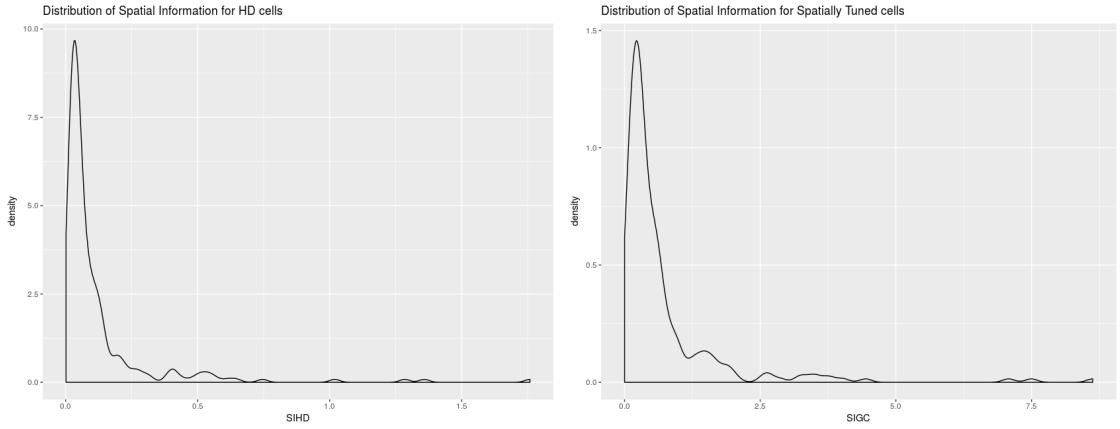

Figure 15: Density plots of the distribution of spatial information of HD and ST for all cells.

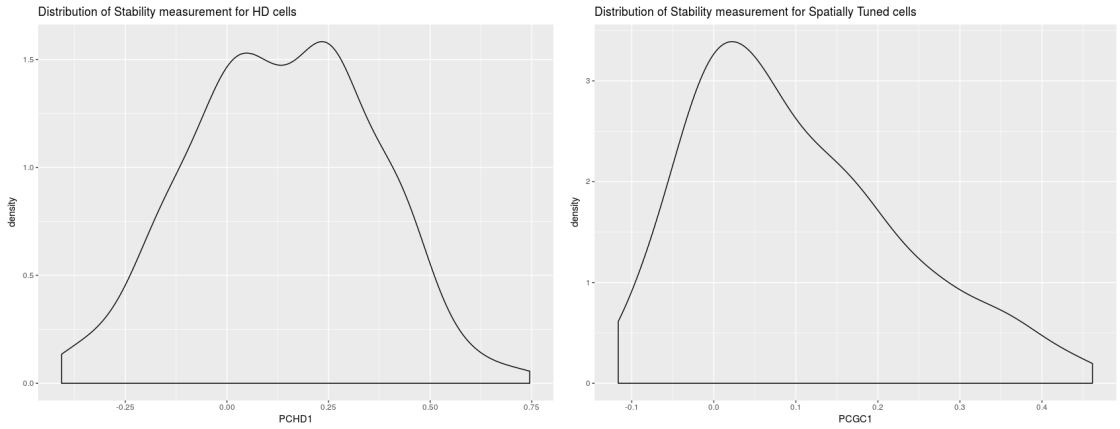

Figure 16: Density plots of the distribution of stability measurement of HD and ST for all cells.

II Statistical models and methods

3 The generalised linear model

In this section we introduce the generalised linear model (GLM), of which logistic regression is a special cases. We also present the iteratively re-weighted least squares (IWLS) algorithm, which is a method to find the maximum likelihood estimates (MLEs) of a GLM. We consider hypothesis tests for the regression parameters with the Wald test. In addition, we mention the Bonferroni correction, used in multiple hypothesis testing.

3.1 The GLM-framework

In short, the GLM consists of three elements (i) a probability distribution that is an exponential family (ii) a linear predictor η (iii) and a link function g that connects the mean of the probability distribution to the linear predictor. In the following we elaborate on each of these elements.

Consider n independent observations $y_1, ..., y_n$, where y_i is treated as a realisation of a random variable Y_i. Assume that Y_i has a probability distribution that is an exponential family, that is

$$Y_i \sim f_{Y_i}(y_i; \theta_i, \phi),$$

where θ_i is the (one dimensional) parameter of interest of the family, and ϕ is called the dispersion parameter. These parameters are essentially location and scale parameters, respectively. The probability density function can be expressed as (McCullagh and Nelder, 1989, p. 28)

$$f_{Y_i}(y_i; \theta_i, \phi) = \exp\Big(\frac{\theta_i y_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\Big), \tag{7}$$

where $a(\phi)$, $b(\theta_i)$ and $c(y_i, \phi)$ are known functions. If ϕ is known, θ_i is called the canonical parameter. Furthermore, θ_i is related to the mean and the variance of the distribution through

$$\mu_i = \mathrm{E}(Y_i) = b'(\theta_i) \tag{8}$$

$$\mathrm{Var}(Y_i) = b''(\theta_i)a(\phi). \tag{9}$$

The linear predictor can be written as

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}, \tag{10}$$

where $\boldsymbol{\eta} = (\eta_1, ..., \eta_n)^T$, $\mathbf{X}$ is a $n \times p$ design matrix and $\boldsymbol{\beta}$ is a $p \times 1$ vector of the unknown parameters. These unknown parameters can be estimated by maximising the likelihood function, as will be discussed in Section 3.2. For models with an intercept term, the first column of the design matrix $\mathbf{X}$ is a column of ones.

The link function g connects the mean of the distribution to the linear predictor by

$$\eta_i = g(\mu_i). \tag{11}$$

Whenever θ_i is a canonical parameter of (7), the link function g is the function that expresses θ_i in terms of μ_i, that is

$$\theta_i = g(\mu_i). \tag{12}$$

In this setting g is referred to as the canonical link function. It is possible to use non-canonical link functions, but then consideration should be made such that the domain of the link function matches the range of the mean of the probability distribution.

3.1.1 Binomial model for binary data

Let the n independent observations $y_1, ..., y_n$ be the numbers of *successes* in n_i independent Bernoulli trials. The result of a Bernoulli trial is a random variable Z such that (Dobson and Barnett, 2008, p. 123)

$$Z = \begin{cases} 1, & \text{with} \quad P(Z=1) = p \\ 0, & \text{with} \quad P(Z=0) = 1-p, \end{cases}$$

where $z = 1$ is considered a success. That is, each y_i is a realisation of a random variable

$$Y_i = \sum_j^{n_i} Z_{i,j},$$

where $P(Z_{i,j} = 1) = p_i$ for all j's. The distribution of Y_i is then given by

$$\begin{aligned} Y_i &\sim \text{binomial}(n_i, p_i) \\ f_{Y_i}(y_i; \theta_i, \phi) &= \binom{n_i}{y_i} p_i^{y_i} (1-p_i)^{n_i - y_i} \\ &= \exp\left(\log\left(\frac{p_i}{1-p_i} \right) y_i + n_i \log(1-p_i) + \log \binom{n_i}{y_i} \right). \end{aligned} \tag{13}$$

Comparing (13) to (7), we see that $a(\phi) = \phi = 1$, that is, θ_i is a canonical parameter. Furthermore, we have that

$$\theta_i = \log\left(\frac{p_i}{1-p_i} \right)$$

and

$$b(\theta_i) = n_i(\theta_i + \log(1 + \exp(-\theta_i))),$$

where we've used

$$p_i = (1 + \exp(-\theta_i))^{-1}$$

in the expression for $b(\theta_i)$. Hence, from (8) and (9) we get

$$\begin{aligned} \mu_i = \text{E}(Y_i) &= \frac{n_i}{1 + \exp(-\theta_i)} = n_i p_i \\ \text{Var}(Y_i) &= \frac{n_i}{(1 + \exp(-\theta_i))^2} = n_i p_i (1-p_i). \end{aligned} \tag{14}$$

And according to (12), the canonical link function is

$$\begin{aligned} g(\mu_i) &= \log\left(\frac{p_i}{1-p_i} \right) \\ &= \text{logit}(p_i) \end{aligned} \tag{15}$$

where the last step is the definition of the logit-function. Hence, combining (10) and (11) gives the model

$$\text{logit}(p_i) = \mathbf{x}_i^T \boldsymbol{\beta}.$$

3.2 Parameter estimation

As mentioned, the unknown parameters $\boldsymbol{\beta}$ in (10) can be estimated by maximising the likelihood function. These estimates are called maximum likelihood estimates (MLEs), and are denoted $\hat{\boldsymbol{\beta}}$. Calculation of the MLEs require iterative methods. In our analysis we've used the built-in function `glm` in `R` (R Core Team (2016)) to obtain an estimate for $\hat{\boldsymbol{\beta}}$. The `glm` function uses a method called iteratively re-weighted least squares (IWLS). The following is an overview of the IWLS algorithm.

3.2.1 The log-likelihood function

The log-likelihood function for a single observation y_i is given by

$$\begin{aligned}\log L_i(\theta_i, \phi; y_i) &= \log f_{Y_i}(y_i; \theta_i, \phi) \\ &= \frac{\theta_i y_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi),\end{aligned} \tag{16}$$

where we've used (7). And the log-likelihood for the set of independent observations $\mathbf{y} = (y_1, ..., y_n)^T$ is simply

$$\log L(\boldsymbol{\theta}, \phi; \mathbf{y}) = \sum_{i=1}^{n} \log L_i(\theta_i, \phi; y_i). \tag{17}$$

Note that θ_i is related to the mean of the distribution μ_i in (8), which itself is related to the linear predictor η_i in (11). And η_i is again related to the unknown parameters $\boldsymbol{\beta}$ through (10). This is relevant because it shows that the log-likelihood function in (16), and hence in (17), are functions of $\boldsymbol{\beta}$.

A formal definition of the MLE $\hat{\boldsymbol{\beta}}$ is given by (Rodríguez, 2007, Appendix A, p. 1)

$$\log L(\hat{\boldsymbol{\beta}}; \mathbf{y}) \geq \log L(\boldsymbol{\beta}; \mathbf{y}) \quad \text{for all} \quad \boldsymbol{\beta}.$$

3.2.2 Fisher's score function and the information matrix

The first derivative of the log-likelihood is called Fisher's score function, and is given by (Rodríguez, 2007, Appendix A, p. 3)

$$\mathbf{u}(\boldsymbol{\beta}) = \frac{\partial \log L(\boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\beta}}, \tag{18}$$

where $\mathbf{u}$ is a $p \times 1$ vector whenever $\boldsymbol{\beta}$ is a $p \times 1$ vector, as in (10). In general the score function can be written as a function of both the parameters $\boldsymbol{\beta}$ and the random vector $\mathbf{Y} = (Y_1, ..., Y_n)^T$, where observation y_i is considered a realisation of the random variable Y_i. Hence $\mathbf{u}(\boldsymbol{\beta}, \mathbf{Y})$ is a random vector. In that case, its mean and variance are given by (Rodríguez, 2007, Appendix A, p. 3)

$$\begin{aligned}\mathrm{E}(\mathbf{u}(\boldsymbol{\beta}, \mathbf{Y})) &= 0 \\ \mathrm{Var}(\mathbf{u}(\boldsymbol{\beta}, \mathbf{Y})) &= \mathrm{E}(\mathbf{u}(\boldsymbol{\beta}, \mathbf{Y})\mathbf{u}(\boldsymbol{\beta}, \mathbf{Y})^T) = \mathbf{I}(\boldsymbol{\beta}),\end{aligned} \tag{19}$$

where $\mathbf{I}$ is a $p \times p$ matrix, called the expected Fisher information matrix.

Since we have canonical link the log-likelihood is a concave function, thus we can find the MLE by setting the first derivative of the log-likelihood equal to zero (Rodríguez, 2007, Appendix A, p. 3). That is, we need to solve the system of equations

$$\mathbf{u}(\hat{\boldsymbol{\beta}}) = \mathbf{0}. \tag{20}$$

Using a first order Taylor series, we can expand the score function evaluated at the MLE $\hat{\boldsymbol{\beta}}$ around an arbitrary value $\boldsymbol{\beta}_0$

$$\mathbf{u}(\hat{\boldsymbol{\beta}}) \approx \mathbf{u}(\boldsymbol{\beta}_0) + \frac{\partial \mathbf{u}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\bigg|_{\boldsymbol{\beta}=\boldsymbol{\beta}_0} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0),$$

and using (20), we can solve for $\hat{\boldsymbol{\beta}}$, such that

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_0 - \mathbf{H}^{-1}(\boldsymbol{\beta}_0)\mathbf{u}(\boldsymbol{\beta}_0), \tag{21}$$

where $\mathbf{H}(\boldsymbol{\beta}) = \partial \mathbf{u}(\boldsymbol{\beta})/\partial \boldsymbol{\beta}$ is a $p \times p$ matrix called the Hessian. This expression forms the basis of an iterative technique called the Newton-Raphson method (Rodríguez, 2007, Appendix A, p. 5), where a given trial value is updated using (21) until convergence. An alternative method, known as Fisher scoring, is given by (Rodríguez, 2007, Appendix A, p. 5)

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_0 + \mathbf{I}^{-1}(\boldsymbol{\beta}_0)\mathbf{u}(\boldsymbol{\beta}_0), \tag{22}$$

where the Hessian in (21) is replaced by the information matrix $\mathbf{I}$ using (Rodríguez, 2007, Appendix A, p. 4)

$$\mathbf{I}(\boldsymbol{\beta}) = -\mathrm{E}\Big(\frac{\partial^2 \log L(\boldsymbol{\beta}; \mathbf{y})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}\Big) = -\mathrm{E}(\mathbf{H}(\boldsymbol{\beta})).$$

To find expressions for the score function $\mathbf{u}(\boldsymbol{\beta})$ and the information matrix $\mathbf{I}(\boldsymbol{\beta})$, we differentiate the log-likelihood in (16) using the chain rule (McCullagh and Nelder, 1989, p. 41)

$$\frac{\partial \log L_i}{\partial \beta_j} = \frac{\partial \log L_i}{\partial \theta_i} \frac{\mathrm{d}\theta_i}{\mathrm{d}\mu_i} \frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} \frac{\partial \eta_i}{\partial \beta_j}.$$

Using (8) and (9) we derive that $\mathrm{d}\mu_i/\mathrm{d}\theta_i = \mathrm{Var}(Y_i)/a(\phi)$, and from (10) we get that $\partial \eta_i/\partial \beta_j = x_{ij}$, such that

$$\frac{\partial \log L_i}{\partial \beta_j} = \frac{(y_i - \mu_i)}{\mathrm{Var}(Y_i)} \frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} x_{ij}.$$

This gives that each component of the score function in (18) is given by (Dobson and Barnett, 2008, p. 65)

$$u_j = \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{\mathrm{Var}(Y_i)} \frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} x_{ij}. \tag{23}$$

And using (19), we get that the elements of the information matrix $\mathbf{I}(\boldsymbol{\beta})$ are given by (Dobson and Barnett, 2008, p. 65)

$$\begin{aligned} I_{jk} &= \mathrm{E}(u_j u_k) \\ &= \mathrm{E}\Big(\sum_{i=1}^{n} \frac{(Y_i - \mu_i)}{\mathrm{Var}(Y_i)} \frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} x_{ij} \sum_{l=1}^{n} \frac{(Y_l - \mu_l)}{\mathrm{Var}(Y_l)} \frac{\mathrm{d}\mu_l}{\mathrm{d}\eta_l} x_{lk}\Big) \\ &= \sum_{i=1}^{n} \frac{\mathrm{E}(Y_i - \mu_i)^2 x_{ij} x_{ik}}{(\mathrm{Var}(Y_i))^2} \Big(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i}\Big)^2, \end{aligned}$$

since $\mathrm{E}((Y_i - \mu_i)(Y_l - \mu_l)) = 0$ for $i \neq l$ because the random variables Y_i are independent, since the observations y_i are assumed independent (Dobson and Barnett, 2008, p. 65). And by using $\mathrm{E}(Y_i - \mu_i)^2 = \mathrm{Var}(Y_i)$ we get that

$$I_{jk} = \sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{\mathrm{Var}(Y_i)} \Big(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i}\Big)^2. \tag{24}$$

Finally, we can write the information matrix as (Dobson and Barnett, 2008, p. 66)

$$\mathbf{I}(\boldsymbol{\beta}) = \mathbf{X}^T\mathbf{W}\mathbf{X}, \tag{25}$$

where we define $\mathbf{W}$ as an $n \times n$ diagonal matrix with elements

$$w_{ii} = \frac{1}{\mathrm{Var}(Y_i)} \Big(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i}\Big)^2,$$

where $\mathrm{d}\mu_i/\mathrm{d}\eta_i$ is evaluated at $\boldsymbol{\beta}$.

3.2.3 The iteratively re-weighted least squares (IWLS) algorithm

Consider now the expression in (22) as an iterative algorithm (Dobson and Barnett, 2008, p.65) such that

$$\mathbf{b}^m = \mathbf{b}^{m-1} + \mathbf{I}^{-1}(\mathbf{b}^{m-1})\mathbf{u}(\mathbf{b}^{m-1}),$$

where $\mathbf{b}^m$ is a vector of estimates of the parameters $\boldsymbol{\beta}$ at the mth iteration. Multiplying both sides of this expression with the information matrix $\mathbf{I}$, we get

$$\mathbf{I}(\mathbf{b}^{m-1})\mathbf{b}^m = \mathbf{I}(\mathbf{b}^{m-1})\mathbf{b}^{m-1} + \mathbf{u}(\mathbf{b}^{m-1}), \tag{26}$$

Using (23) and (24), we see that the right-hand side of this expression is

$$\sum_{k=1}^{p}\sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{\mathrm{Var}(Y_i)} \Big(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i}\Big)^2 b_k^{m-1} + \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{\mathrm{Var}(Y_i)} \frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} x_{ij},$$

which can be written as

$$\mathbf{X}^T\mathbf{W}\mathbf{z}, \tag{27}$$

where we've defined $\mathbf{z}$ with components

$$z_i = \sum_{k=1}^{p} x_{ik}b_k^{(m-1)} + (y_i - \mu_i)\frac{\mathrm{d}\eta_i}{\mathrm{d}\mu_i}.$$

Using (25) and (27), we can rewrite (26) as

$$\mathbf{X}^T\mathbf{W}\mathbf{X}\mathbf{b}^m = \mathbf{X}^T\mathbf{W}\mathbf{z}. \tag{28}$$

Note that for a normal linear model, the so called normal equations have the form (Dobson and Barnett, 2008, p. 89)

$$\mathbf{A}^T\mathbf{A}\mathbf{b} = \mathbf{A}^T\mathbf{y}, \tag{29}$$

where $\mathbf{A}$ is the design matrix, $\mathbf{b}$ is the estimate of the MLE (called the least square estimator) and $\mathbf{y}$ are the data. Comparing (28) and (29), we see that they have the same form, except for the weights $\mathbf{W}$ in (28). Furthermore, the equations in (28) need to be solved iteratively, contrary to (29), since both $\mathbf{z}$ and $\mathbf{W}$ are dependent on $\mathbf{b}^{(m-1)}$. Hence, the iterative method of (28) is called the iteratively re-weighted least squares (IWLS) algorithm. This algorithm is said to converge whenever the difference between $\mathbf{b}^{(m)}$ and $\mathbf{b}^{(m-1)}$ becomes small, relative to a tolerance. Then, $\mathbf{b}^{(m)}$ is taken to be an estimate of the MLE $\hat{\boldsymbol{\beta}}$ (Dobson and Barnett, 2008, p. 66).

3.3 Hypothesis testing

In this section we describe the Wald test and the likelihood ratio test, which are competitors to the permutation test. Furthermore, since we conduct a series of hypotheses tests, we mention the Bonferroni correction.

3.3.1 Wald test

Consider hypotheses of the form

$$H_0 : \boldsymbol{\beta} = \boldsymbol{\beta}_0, \tag{30}$$

where $\boldsymbol{\beta}$ is a $p \times 1$ vector of parameters and $\boldsymbol{\beta}_0$ a $p \times 1$ vector of fixed values. Such hypothesis can be tested using the Wald test, where the test statistic is (Rodríguez, 2007, Appendix A, p. 6)

$$W = (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^T \mathrm{Cov}(\hat{\boldsymbol{\beta}})^{-1} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0),$$

where $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ denotes the $p \times p$ variance-covariance matrix of the estimated parameters. The Wald test statistic has approximately in large samples a chi-squared distribution (Dobson and Barnett, 2008, p.85)

$$W \sim \chi^2(p) \qquad \text{(approx.)}, \tag{31}$$

where p is the degrees of freedom. This follows from the fact that in large samples (that is, for large values of n, the total number of observations), the MLE follows approximately a multivariate normal distribution (Rodríguez, 2007, Appendix A, p. 6)

$$\hat{\boldsymbol{\beta}} \sim N_p(\boldsymbol{\beta}, \mathrm{Cov}(\hat{\boldsymbol{\beta}})),$$

where $\mathrm{E}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$ denotes the true parameter values. The variance-covariance matrix of the MLE can be replaced by any consistent estimator, without altering the asymptotic distribution of W in (31). Particularly, the variance-covariance matrix can be estimated by

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}) = \mathbf{I}^{-1}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1}, \tag{32}$$

where we've used the expression for the information matrix in GLMs in (25). The elements of $\mathbf{W}$ are the weights obtained in the last iteration of the IWLS-algorithm in (28). Hence the Wald statistic for an MLE $\hat{\boldsymbol{\beta}}$ in a GLM-framework can be written

$$W = (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^T \mathbf{X}^T \mathbf{W} \mathbf{X} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0).$$

Such Wald tests may be used to test the significance of each estimated parameter $\hat{\beta}_j$, where the test hypothesis is

$$H_0 : \beta_j = 0.$$

In such cases where the MLE $\hat{\beta}_j$ is a scalar, it is common to take the square root of the Wald statistic (Dobson and Barnett, 2008, p. 78)

$$z = \frac{\hat{\beta}_j}{\sqrt{\text{Var}(\hat{\beta}_j)}}, \tag{33}$$

where the z-statistic has a standard normal distribution in large samples.

3.3.2 Bonferroni correction

In our analysis, we test multiple hypotheses simultaneously. This increases the overall risk of making a type I error, which is the event of rejecting H_0 when H_0 is true. For example, if we conduct a series of tests using significance level 5%, we would expect one in twenty to be significant by chance alone (Rodríguez, 2007, p. 53). A possible solution is to control the family-wise error rate (FWER), which is defined as

$$FWER = P(V > 0),$$

where V is the number of false positives among k tests. Given a significance level α, the Bonferroni correction is

$$\alpha_{\text{Bon}} = \alpha/k. \tag{34}$$

That is, we use α_{Bon} as the cutoff for each of the individual k tests. The Bonferroni correction guarantees that $FWER \leq \alpha$ (Kass et al., 2014, p. 304).

3.4 Basis function expansion

Cosine bases As in Fawad (2017) our choice for basis functions are the raised cosine bumps, given by Pillow et al. (2008) as

$$b_d(l) = \begin{cases} \frac{1}{2}\cos(a\log(l+c) - \phi_d) + \frac{1}{2}, & \text{if } a\log(l+c) \in [\phi_d - \pi, \phi_d + \pi], \\ 0, & \text{otherwise.} \end{cases} \tag{35}$$

where l represents the time lag, the ϕ_d's are comparable to the placements or peaks of the cosines which are separated by $\pi/2$, and a and c are constants. According to Pillow et al. (2008), these constants are best chosen by evaluating the auto- and cross-correlation functions of the activity of the neurons. We have chosen to use the choices of Pillow et al. (2008).

We have used $d = 1, 2, 3, 4$ to represent connectivity effects. The resulting basis functions (35) are shown in Figure 17 along with its orthonormal equivalent in Figure 18. In our analyses, we use the orthonormal cosine bases. The figures show that the orthonormal cosine basis allows for a finer temporal representation at short lags, and a coarser representation at long lags. Thus with the orthonormal bases we can more precisely measure the effects near the time of a spike, which is desirable, since more emphasis is placed on this time interval in neuroscience literature, as it can display effects such as refractory periods Kass et al. (2014).

Understanding basis functions as separate covariates Using a predefined basis functions such as the cosine basis $b_d(l)$ we can construct new model covariates $\mathbf{z}_d$ from old model covariates, such as the neighbour neuron's spike train $\mathbf{x}$. In our case the equation to construct our new model covariates is given as,

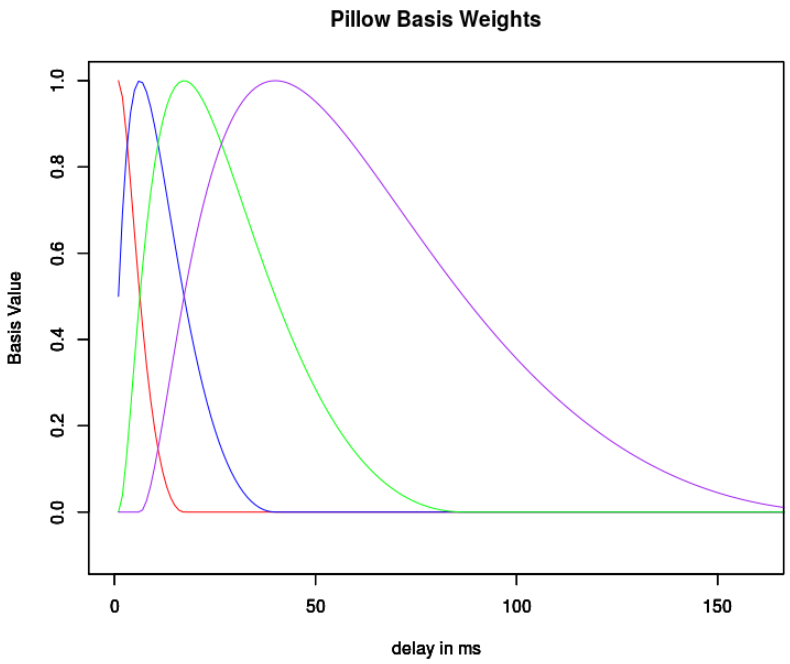

Figure 17: Illustration of the (regular) Pillow basis weights for $d = 1, 2, 3, 4$ (correspondingly red, blue, green and purple).

$$z_d(t) = \sum_{l=1}^{160} b_d(l) x(t-l) \tag{36}$$

where $z_d(t)$ is the t'th element of the model covariate constructed by the d'th basis function $\mathbf{b}_d$. $b_d(l)$ is the cosine basis element as described in Equation (35), and $x(t-l)$ is the $(t-l)$'th element of the original spike train of the neighbouring neuron. Thus, in our case for $d = 1, 2, 3, 4$, we get for new model covariates $z_1(t), z_2(t), z_3(t), z_4(t)$.

The new model covariates that are constructed with non-orthonormal Pillow basis are fairly intuitive to understand, as they are the weighted positive spike ratio in a local area. The orthonormal Pillow basis are less intuitive, as it is both negative and positive weights in a bigger and overlapping area for each model covariate. However, the orthonormal bases increase the prediction precision, and are thus preferred.

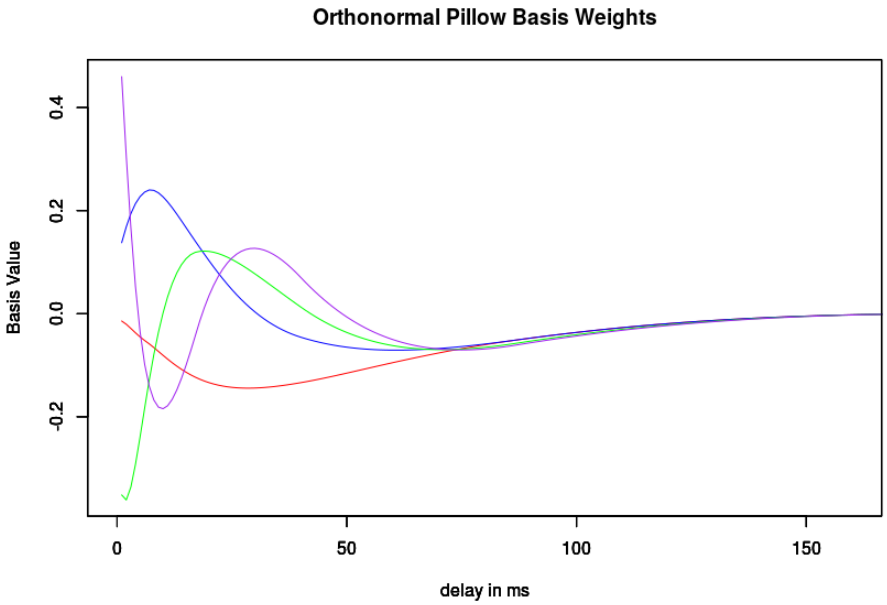

Figure 18: Illustration of the orthonormal Pillow basis weights for $d = 1, 2, 3, 4$ (correspondingly red, blue, green and purple).

4 Jitter related methods

In this section we introduce a new analytic framework we call for the jittered generalised linear model (JGLM). The JGLM is developed to analyse the connectivity between different neurons based on spike trains. Spike trains are explained in Section 1.2.3, and connectivity is explained in Section 1.2.3. In the following sections we present the permutation test, the jittering process, jittered cross-correlation, hypothesis testing related with JGLM and how to use the JGLM.

4.1 Permutation test

A permutation test is a type of non-parametric randomisation test in which the null distribution of a test statistic is estimated by permutation.

Let T be a test statistic so that large values are in favour of the alternative hypothesis. Let t_{obs} be the observed test statistic in a given experiment. We generate i.i.d. model covariate spike trains under the null hypothesis H_0, and calculate test observator T_{sim}. In principle the ideal p-value is,

$$p_\infty = P(T_{sim} \geq t_{obs})$$

but this is unknown. To compute p_∞ exactly we would need to generate an infinite number of simulated data sets (permutations). However, from Phipson and Smyth (2010), we have that p_∞ can be approximated by p_u where

$$p_u = P(T_{sim} \geq t_{obs}) = \frac{b+1}{B+1} \tag{37}$$

where b is the number of times out of B that $T_{sim} \geq t_{obs}$. This will give a valid p-value, that is $P(p_u \leq \alpha) \leq \alpha$ for all $\alpha \in [0, 1]$.

We will use the log-likelihood as our test statistic T. The GLM model fit is created from spike train data from neuron pairs. We will next discuss how random permutation of spike train data can be generated with the aid of spike jittering.

4.2 The jittering process

The basic idea of a jittering process is to disrupt the order in a sequence to investigate the importance of the order in the original sequence. We can then compare the order of the jittered sequences with the original sequence with a hypothesis test. There are many hypothesis test alternatives for testing the significance of the order of a sequence by jittering process, some of them will be explained in Section 4.2 as well as in Section 4.4.

Here we use jittering on a discretized m-length Bernoulli process. Any such sequence can be written as $\mathbf{x} := (x_1, ..., x_m) \in \{0, 1\}^m$, where x_j indicates a 1 or a 0 for the sequential position j. For neuroscience modelling $\mathbf{x}$ could represent a spike train where the value of x_j would represent a firing or no firing in time bin j.

Let $\mathbf{X} := (X_1, ..., X_m)$ be a random sequence of the same form as $\mathbf{x}$. In the neuroscience context $\mathbf{X}$ is a random spike train drawn from an Bernoulli distribution with p_j that varies for each bin j, determined by the experimental setup.

We want to make $\mathbf{X}$ into a jittered (randomised) version $\mathbf{X}^{(i)} = (X_1^{(i)}, X_2^{(i)}, ..., X_m^{(i)})$. Since the jittering process is random, many possible realisations of $\mathbf{X}^{(i)}$ exist. Each different version of $\mathbf{X}^{(i)}$ would reveal different information of the original $\mathbf{X}$, therefore to make the jittered vectors

interpretative we jitter the $\mathbf{X}$ many times. Let each $\mathbf{X}^{(i)}$ for $i = 1, 2, ..., J$ be a jittered version of the original vector $\mathbf{X}$. A common choice is to jitter $J = 1000$ times. Thereafter the results from all the jittered $\mathbf{X}^{(i)}$ can be compared to the results from the original $\mathbf{X}$.

It is assumed that the sequence $\mathbf{x}$ that we seek to jitter stems from a Bernoulli process with probability value $p_j < 0.5$ and preferably $p_j << 0.5$ for every j. Since we assume our sequences $\mathbf{x}$ to have fewer 1-values than 0-values, it will be more efficient to "jitter" or change the position of the 1-values than vice versa.

As explained in Amarasingham et al. (2011), there are many different jittering methods, two of them are presented below and are called *uniform basic jittering* and *uniform interval jittering*.

4.2.1 Uniform interval jittering

In interval jitter we partition a sequence of the form $\mathbf{x}$ into k equally-sized windows consisting of Δ number of bins each. $\Gamma_i := \{(i-1)\Delta + 1, ..., i\Delta\}$ denotes the set of bins of the i'th window. Each x_j for a bin j with value 1, is uniformly jittered in its original window. Thus, interval jittering provide local jittering that is non-centred around the spikes original position. Amarasingham et al. (2011) explain that a sequence of spike counts $N(\mathbf{x}) = (N_1(\mathbf{x}), N_2(\mathbf{x}), ..., N_l(\mathbf{x}))$ is defined as the interval-counts of spikes:

$$N_i(\mathbf{x}) = \sum_{j \in \Gamma_i} x_j$$

Let $T(\mathbf{x})$, the *test statistic*, be a fixed function that assigns a real number to a spike train.

Let $\mathbf{X} := (X_1, ..., X_m)$ be a random spike train drawn from an unknown probability distribution, determined by the experimental setup. A surrogate data set $\mathbf{X}^{(i)} := (X_1^{(i)}, X_2^{(i)}, ..., X_m^{(i)})$ is produced by sampling from the (conditional) distribution:

$$P(\mathbf{X}^{(i)} = \mathbf{x} | N(\mathbf{X}) = \mathbf{n}) = \frac{\mathbb{I}\{N(\mathbf{x}) = n\}}{\sum_{\mathbf{y} \in \{0,1\}^m} \mathbb{I}\{N(\mathbf{y}) = n\}} \tag{38}$$

(where $\mathbb{I}\{\}$ represents the indicator function. That is, $\mathbf{X}^{(i)}$ is drawn independently and uniformly from the subset of spike train outcomes (i.e., of $\{0,1\}^m$) that satisfy $N(\mathbf{X}^{(i)}) = N(\mathbf{X})$. Iterating this procedure J times, we obtain J (conditionally) independent samples $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ...\mathbf{X}^{(J)}$, drawn from the distribution given in (38). The computational procedure for sampling from (38) is straightforward: proceeding independently window-by-window, assign $N_i(\mathbf{X})$ spikes to window i by sampling from the uniform distribution on the $\binom{\Delta}{N_i(\mathbf{X})}$ possible assignments, where Δ is number of bins in a window. In other words, conditional on the number of spikes in a window we randomly assign spikes to bins.

For illustration of interval jittering procedure see Figure 19.

4.2.2 Uniform basic jittering

As explained in Amarasingham et al. (2011) basic jittering is a heuristic procedure that jitters spikes around the original spike position. In uniform basic jittering any x_j for a bin j that has the value 1, is jittered such that the new position for the 1-value is placed in x_{j^*} where $j^* \sim U[k = 1; -a, a]$ for any $a = 0, 1, 2, ...$, where $U(k = 1; -a, a)$ represents the discrete uniform distribution for any integers in the interval $[-a, a]$.

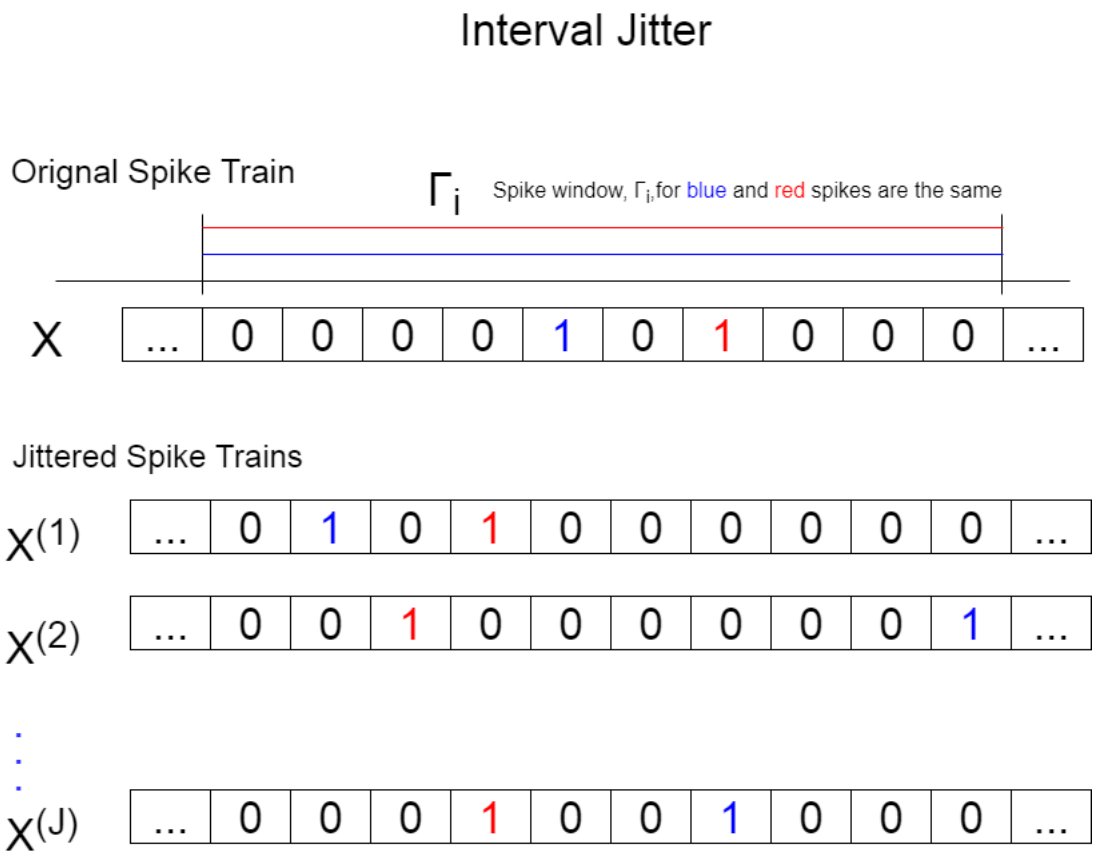

Figure 19: Illustration for the interval jittering procedure in window Γ_i for a spike train $\mathbf{X}$. Γ_i contains $N_i = 2$ firings, colour coded with blue and red in the illustration. For each jittered spike train $\mathbf{X}^{(i)}$, the red and blue firings have a randomised position within its windows Γ_i.

Alternatively we can imagine basic jittering as an interval jittering procedure were the jitter windows are centred around the spikes in the original data set. That is, instead of having non-overlapping windows as interval jittering does, we now have a jittering windows centred in every spike.

It is important to note that any basic jittering procedure provides jittering where the 1-values or so called spikes for a spike train is centred around its original position. As is further explained in Section 4.2.3, using a centred jittering procedure is not recommended.

For illustration of basic jittering procedure see Figure 20.

4.2.3 Discussion

As explained in Platkiewicz et al. (2017) interval jitter gives valid p-values, guaranteeing valid conclusions. However, such a guarantee is not available for spike-centred jitter, such as basic jitter. For the proof see the cited article, nonetheless in this section we seek to deliver a more intuitive

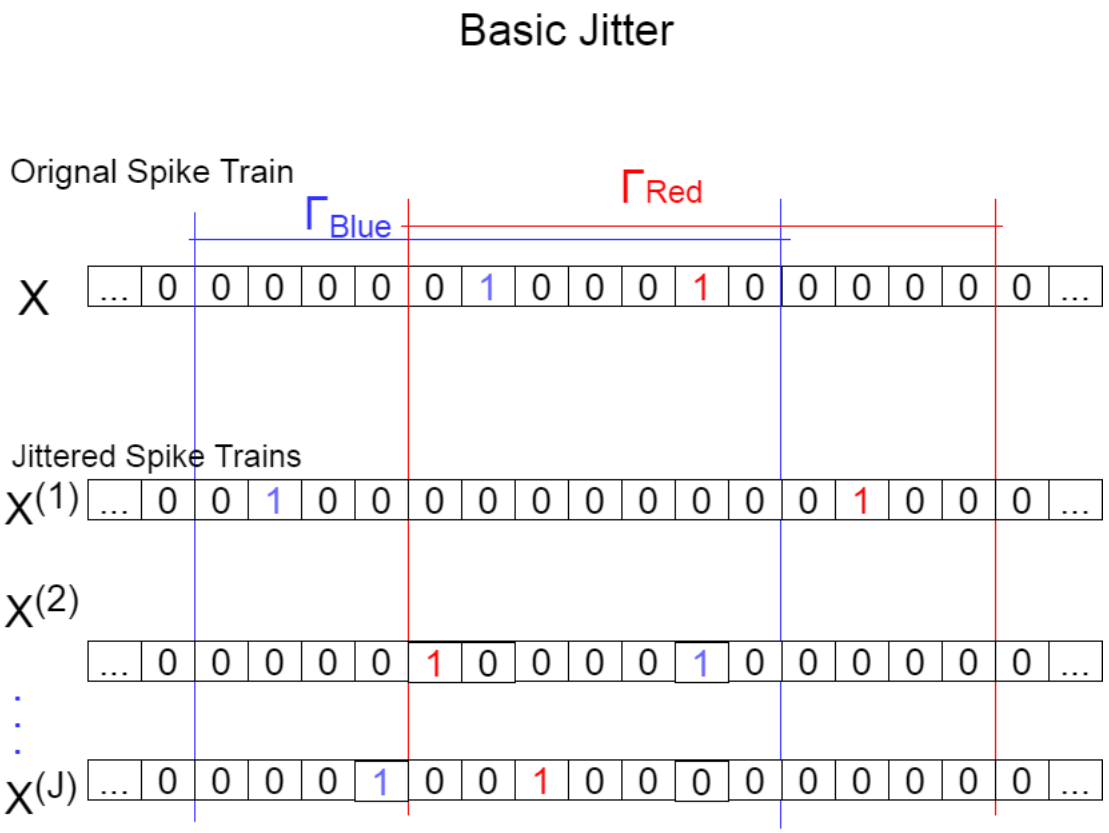

Figure 20: Illustration for the basic jittering procedure in an sub-area for a spike train $\mathbf{X}$. In this area there are two firings, which are colour coded blue and red. The "blue" firing are in each jittered version $\mathbf{X}^{(i)}$ randomly placed in the jittered-centred window Γ_{Blue}, and likewise for the "red" firing in the window Γ_{Red}.

and logical interpretation for why interval jitter is the better choice.

The idea of jittering is that we want to test for connectivity between cell pairs by their spike ordering in their spike trains (ST). We can imagine that the order in the spike trains, which makes us able to detect connectivity, are twofold: local temporal placement of spike (LTPS) and local frequency (LF). In this thesis we focus on testing connectivity due to local temporal spike placement/distribution. We do not want to test for the connectivity between cells due to the local frequency (frequency in windows) of spike trains. Thus, we want the jittering procedure to not change the local frequency in the spike trains, yet to jitter the local temporal spike placement as much as possible (completely random). In other words, we want the frequency in a window, as explained in Section 4.2.1, to be unchanged, yet the order in the window to be as random as possible. Following this logic, interval jitter would be an intuitive choice, and far better than basic jitter.

The reason why we consider interval better than basic jittering are twofold, and can be put into

words as the following.

- Interval jitter properly analyses connectivity due to LTPS, and not due to LF, while basic jitter is a mix of both.
- Not only does basic jitter isolate the effects from LTPS badly, it also hallucinates (exaggerates) connectivity due to LTPS, since the jitter spike placement are directly dependent on (centred around) the original spike placement.

If we did not want to restrict ourselves to only detect connectivity due to temporal spike placement, but also due to the frequency changes, the best jittering procedure could be to completely randomise the order of the spike train. However, by completely randomising the spike placement for all of the ST, the connectivity effects we investigate by the jittering effects are no longer local. Such a complete randomising effect can be understood as an interval jittering procedure with only one window, which covers the whole area of the spike train. However, such a procedure eliminates the effectiveness of detecting which type of connections we have between cells. As there is nothing left of the original temporal spike placement in the jittered version of the spike train, we are not able to estimate the offset in firing time between the cells in the cell pair.

At first, an apparent way to try to measure the effectiveness between the basic and interval jittering procedure may seem to be to measure in what way their results differ from the original spike train. That is, if the results from interval jittering are more different from the original spike train than the basic jittering are from the original spike train, for the same window size, we can assume that it's due to interval jittering being a better jittering procedure, as it highlights the connections between the original cell pair better. However, when we try to measure this effect we have to be very careful as the measured difference between the basic jitter and interval jitter could be different due to two different causes, which are as following.

- Two different jittering procedures may consistently produce different results due to the amount they randomise the original spike train. For instance imagine the two jittering procedure both being basic jitter, but with different window size. We expect the basic jitter with the biggest window size to have the more extreme values compared to the original spike train.
- The other reason for consistently different results between jittering methods are the way we jitter itself. That is, for two methods with the same window size such as basic jitter and interval jitter we may see different results due to basic jitter being spike-centred while interval jitter is not spike-centred.

Although we compare the results for interval and basic jitter with the same window size it may seem that the the second point is the only point influencing the consistent different results. Interval jitter with the same window size is in fact more randomised than basic jitter. This can be explained as follows; interval jittering is "changed" twice, while basic jitter is only "changed" once. That is, for interval jitter the original spike is first placed into a non-centred window, before it's jittered in this window. However, for basic jittering, the original spike is simply jittered around its original position. As a results for a window of size $\Delta = 11$ the basic jitter can move the original spike up to 5 bins (in each direction) from its original position. However, for interval jitter for a window of size $\Delta = 11$, the original spike can be moved up to 10 bins from it's original position, (if its original position is either in the first or the last bin in its corresponding window).

Thus, the isolated consistent effect for the difference between basic and interval due to the jittering procedure itself being different is difficult to measure. Thus, what at first may appear as a simple sub-problem, choosing jittering procedure, is revealed to be a very complex problem. Furthermore, the choice of jittering procedure is important for how we later can interpret our results.

4.3 Jittered cross correlation

Jittered cross-correlation (JCC) is used in neuroscience in order to investigate the connectivity between spike trains, as explained in Salt (2017). Jittered cross-correlation seeks to investigate the significance in the local temporal relationship between two discrete vectors, for example two spike trains.

For two discrete functions, that we denote by $\mathbf{Y}$ and $\mathbf{X}$, we can measure the similarity between them by discrete cross-correlation. For a real valued discrete vectors $\mathbf{Y}$ and $\mathbf{X}$, both of length L, the cross-correlation is defined as:

$$(\mathbf{Y} \star \mathbf{X})[n] \ = \sum_{m=1-n\mathrm{I}(n<0)}^{L-n\mathrm{I}(n>0)} Y[m] \ X[m+n] \tag{39}$$

where n is the displacement, also known as lag. Thus, we get a different measurement for similarity between the two discrete functions for different lags.

Given that $\mathbf{Y}$ and $\mathbf{X}$ represent two spike trains, and $\mathbf{X}$ have a tendency to fire 10 ms before $\mathbf{Y}$ fires, this is a indication of connection between them. Jittered cross-correlation tries to systematically measure this effect in order to detect connectivity between a cell pair. What is meant by connectivity in a neuroscientific context is further explained in Section 1.2.3.

As already stated in Section 4.2, there are many ways to jitter, but it is preferred to choose a non-spike-centred jittering method such as interval jittering. To test for connectivity between spike trains with JCC one of the spike trains, lets say $\mathbf{X}$ is jittered $\eta = 1000$ times. Each jittered version of $\mathbf{X}$ can be written as $\mathbf{X}^{(i)}$, where $i = 1, 2, ..., J$. Thereafter we compute the cross-correlation, see (39), for $\mathbf{Y}$ and $\mathbf{X}^{(i)}$ for every i. We compute the cross-correlation Equation 39, we choose lag from $n \in [-50, 50]$, in order to be able to detect common, direct and indirect connections as explained in Section 1.2.3.

Additionally the cross-correlation between $\mathbf{Y}$ and the original spike train $\mathbf{X}$ is computed. Thus, for the original cross-correlation, that is the cross-correlation between $\mathbf{Y}$ and $\mathbf{X}$ we have 101 cross-correlation values, each for one of the n-values, as $n \in [-50, 50]$. Thus we can write these values as a 101 element long vector $\mathbf{Z}$, where

$$\mathbf{Z} = \Big((\mathbf{Y} \star \mathbf{X})[n-50], (\mathbf{Y} \star \mathbf{X})[n-49], ..., (\mathbf{Y} \star \mathbf{X})[n+50] \Big),$$

where $\mathbf{Y}$ and $\mathbf{X}$ are the two original discrete functions we want to compare, and n is the lag between their comparison. Furthermore this can be done for every comparison between $\mathbf{Y}$ and all the jittered versions $\mathbf{X}^{(i)}$ of $\mathbf{X}$. As such we can write the jittered cross-correlation results as

$$\mathbf{Z}^{(i)} = \Big((\mathbf{Y} \star \mathbf{X}^{(i)})[n-50], (\mathbf{Y} \star \mathbf{X}^{(i)})[n-49], ..., (\mathbf{Y} \star \mathbf{X}^{(i)})[n+50] \Big)$$

Now that we both have cross-correlation results from non-jittered spikes trains $\mathbf{Z}$, and cross-correlation results from jittered spike trains $\mathbf{Z}^{(i)}$, we can compare these to see how the jittering

changes the similarity between the signals. If the difference between $\mathbf{Z}$ and one of its jittered versions $\mathbf{Z}^{(i)}$ for any i is great, we can assume that the local temporal spike relationship is of importance between the communication between the cell pair, and thus that the cell pair is connected. However, the difference in value between $\mathbf{Z}$ and any of its jittered versions $\mathbf{Z}^{(i)}$ may vary greatly between every j, thus we want to compute the mean $\bar{\mathbf{Z}}_{(i)}$, as well as the variance $\mathbf{s}^2_{\mathbf{Z}^{(i)}}$ across all of the $\eta = 1000$ of the jittered versions $\mathbf{Z}^{(i)}$.

Instead of comparing $\mathbf{Z}$ to the individual jittered results of $\mathbf{Z}^{(i)}$ we compare $\mathbf{Z}$ directly to all of the jittered results by comparing it to $\bar{\mathbf{Z}}_{(i)}$ and $\mathbf{s}^2_{\mathbf{Z}^{(i)}}$. If the difference in value at a timepoint τ between $Z(\tau)$ and $\bar{Z}_{(i)}$ is greater than that off $c \cdot s_{Z^{(i)}}(\tau)$, for a constant c that determines the quantile size, we can conclude that there is a connection between the pair due to the local temporal spike relationship. A visual representation of this can be seen in Figure 21. This procedure of using both cross-correlation and jittering to investigate connectivity between discrete functions is called jittered cross-correlation. Important to note is that the conclusion from Section 4.2 that interval jittering is a better choice than basic jittering is also valid for the JCC method.

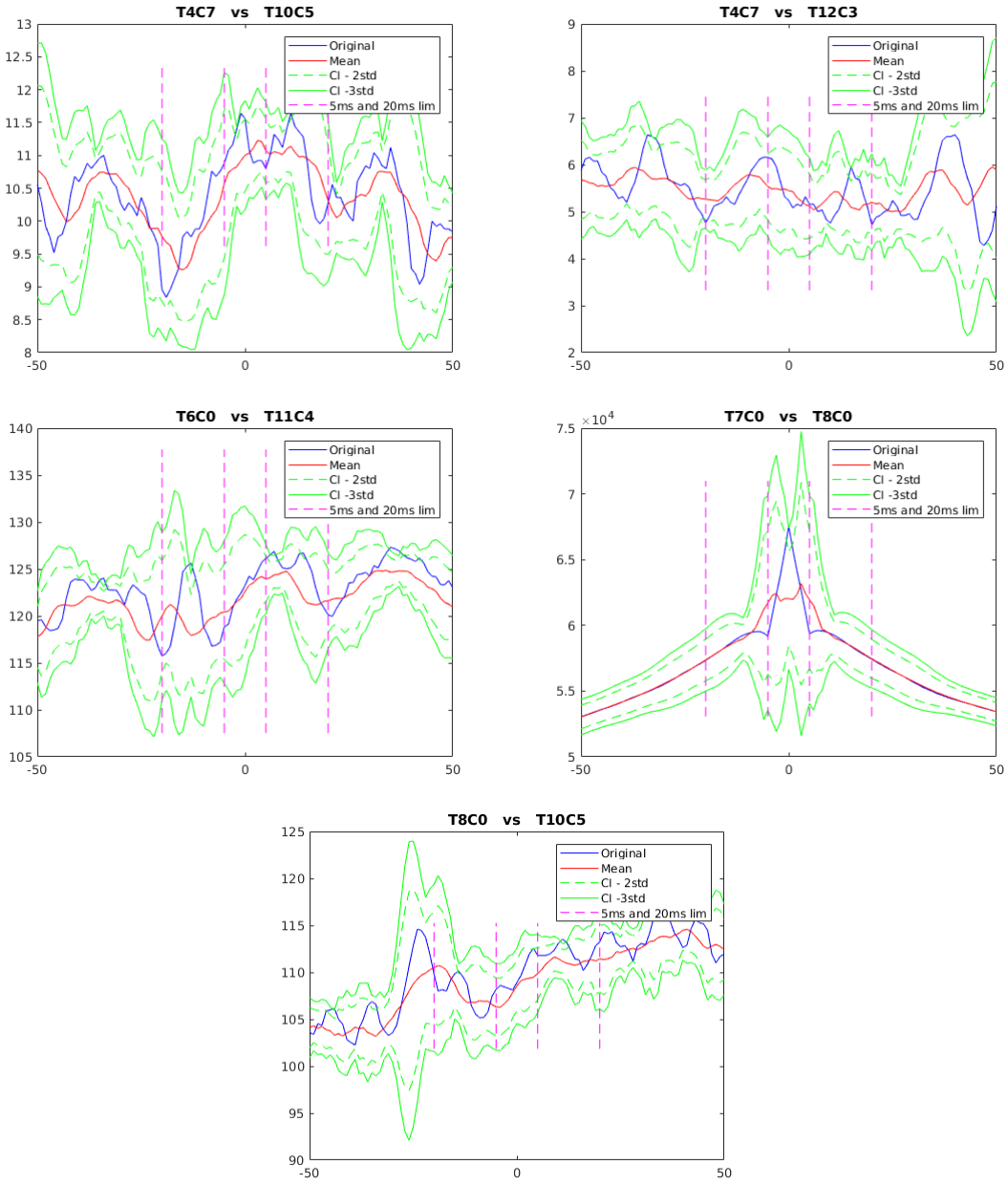

Figure 21: Visual representation of jittered cross-correlation results computed by the author, from the data set Peyrache (2015). Time lags along the x-axis, and the cross-correlation value along the y-axis.

4.4 Jittered hypothesis testing

The JGLM method requires the use of hypothesis testing in order for us to detect significant results. For any jittering procedure there are many different choices for hypothesis testing, depending on what type of data we have, what kind of jittering procedure is chosen and what kind of results we are interested in.

The hypothesis test particularly developed for JCC, explained in Section 4.3, resembles the suggested Basis TC test for JGLM in Section 4.4.2. The other choice for test of JGLM is the use of likelihood-value further explained in Section 4.4.1.

4.4.1 Test by likelihood

By jittering the model covariates and fitting a GLM to $\mathbf{X}$ we have $i = 1, ..., J$ different likelihood values $L^{(i)}$. We let $L^{(1)}, L^{(2)}, ...$ be the a sequence of i.i.d. random variables from an unknown distribution. We would like to test if our original likelihood-value L_{ORG} comes from the same distribution as any of the jittered likelihood-values $L^{(i)}$.

$H_0 : L_{ORG}$ and $L^{(i)}$ are identically distributed
$H_1 : L_{ORG}$ comes from another distribution than $L^{(i)}$ comes from

As we for this jittered likelihood test have made the same assumptions that we have for a permutation test, we can use the same test as is presented in Section 4.1. Thus (37) can in our case be rewritten as,

$$p = \frac{\sum_{i=1}^{J} \mathrm{I}(L^{(i)} \geq L_{ORG}) + 1}{J + 1} \tag{40}$$

where I() is the indicator such that $\mathrm{I}(L^{(i)} \geq L_{ORG}) = 1$ if $L^{(i)} \geq L_{ORG}$ is true, and 0 otherwise. An illustration of this test can be seen in Figure 23.

4.4.2 Basis-tuning-curve

By using basis-tuning-curve (basis-TC) we look at the difference between estimated probability for firing in the response cell based on a firing in the model covariate cell between the fitted and the jittered models. For this test we first construct the basis tuning curve which uses the fitted β-values for the Pillow basis model covariates together with the Pillow basis weights,

$$C_k = \frac{\mathrm{e}^{\eta_k}}{1 + \mathrm{e}^{\eta_\mathrm{k}}} = \frac{\mathrm{e}^{\beta_0 + \sum_{d=1}^{4} \beta_d b_d(k)}}{1 + \mathrm{e}^{\beta_0 + \sum_{d=1}^{4} \beta_d b_d(k)}} \tag{41}$$

where $b_d(k)$ is the Pillow basis weight for the d'th basis function corresponding to the k'th element, as described in Section 3.4. So $\eta_k = \beta_0 + \sum_{d=1}^{4} \beta_d b_d(k)$ is the constructed linear predictor element and C_k is the resulting estimated probability for firing in the response cell Y k ms after a firing in the paired cell X, where X only fired once in the last 160 ms (the chosen pillow basis only spans 160 ms). (The linear predictor in the regular sense is defined as $\eta(t) = \beta_0 + \sum_{d=1}^{4} \beta_d z_d(t)$.) Thus, we can construct a discretized curve, that we call the basis tuning curve, $\mathbf{C} = C_1, C_2, ..., C_{160}$.

Furthermore we can construct another such curve for each of the jittered versions, thus we get,

$$C_k^{(i)} = \frac{e^{\eta_k^{(i)}}}{1 + e^{\eta_k^{(i)}}} = \frac{e^{\beta_0^{(i)} + \sum_{d=1}^{4} \beta_d^{(i)} b_d(k)}}{1 + e^{\beta_0^{(i)} + \sum_{d=1}^{4} \beta_d^{(i)} b_d(k)}}$$

where $\beta_j^{(i)}$ is the results corresponding to β_j in Equation (41) for the results from the fitted GLM from model covariates $\mathbf{X}^{(i)}$. Thus, each jittered versions has it's own linear predictor $\eta_k^{(i)}$ and tuning curve $C_k^{(i)}$. Furthermore, we can estimate the average in each time lag k for each such curve to be,

$$\bar{C}_k = \frac{e^{\bar{\eta}_k}}{1 + e^{\bar{\eta}_k}} = \frac{e^{\bar{\beta_0}^{(i)} + \sum_{d=1}^{4} \bar{\beta_d}^{(i)} d_b(k)}}{1 + e^{\bar{\beta_0}^{(i)} + \sum_{d=1}^{4} \bar{\beta_d}^{(i)} b_d(k)}}$$

where $\bar{\beta_j}^{(i)}$ is the mean of $\beta_j{}^{(i)}$ for all i's. Additionally, we can estimate the variance, S_k^2 , of all of the linear predictors $\eta_k^{(i)}$.

$$S_k^2 = \frac{1}{J} \sum_{i=1}^{J} (\eta_k^{(i)} - \bar{\eta}_k)^2$$

We may construct a confidence interval, for the probability of Y spiking at time t as the function of the spikes in $\mathbf{X}$ for the last 160 ms before time-point t. This results in the confidence interval $[L, U]$ where the lower and upper boundary L and U are defined as,

$$L = \frac{e^{\bar{\eta}_k - cS_k}}{1 + e^{\bar{\eta}_k - cS_k}},$$
$$U = \frac{e^{\bar{\eta}_k + cS_k}}{1 + e^{\bar{\eta}_k + cS_k}},$$

furthermore c is constant determining the size of the interval. As this is a form of multiple testing, one test for each time lag k, we can argue that the constant c should be determined by Bonferroni. However, such a choice is very conservative as the different test are not independent. Thus, one of the challenges using this test is determining the interval size, which is dependent on the the correlation between neighbouring time lags k for the basis tuning curves $C_k^{(i)}$.

Furthermore, it is difficult to give an exact intuitive interpretation of the test results from this test. Thus, in this thesis the test presented in Section 4.4.1, is the preferred test for the JGLM framework.

However, the basis TC is a very useful aid into interpreting the relation between spike train pairs. An illustration of the basis TC can be seen in Figure 24.

4.4.3 Test choice

The preferred test for JGLM in this thesis is the test by likelihood, due to the various challenges with Basis TC hypothesis test. However, if we identify a cell pair to be significantly connected according to our model, we would like to investigate what type of connection we have. This can be investigated by using the Basis TC plots and check for what lag that the C_k deviates most from $\bar{C}_k$ (furthest outside the test interval $[L, U]$).

4.5 Jittered GLM

JGLM is developed as an additional tool to classic asymptotic (non-jittered) GLM and JCC in order to detect connections between cells represented as spike trains. The idea of the JGLM is to combine the strengths of both JCC and GLM into a new more precise and reliable framework in order to detect neuronal connections.

In this setting the strengths of the JCC framework is that it effectively utilises the information in the temporal sequences of spike trains by jittering. The main challenges with this is the exact choice and interpretation of the jittering on the temporal sequences.

GLM is a well developed statistical framework. However, as GLM does not require the response to be sequences of any form, the GLM does not utilise the additional temporal sequence information in spike trains. Thus the JGLM framework can be used as an alternative or an additional tool when the response are a sequence of any kind, such as a temporal sequence.

4.5.1 GLM and its limitations

GLM can be used as a tool to detect connectivity between cells when the response $\mathbf{Y}$ is the spike train from a cell and the spike train from another cell $\mathbf{X}$ is included as model covariate(s). This procedure have been investigated in Aga and Fawad (2017). At first we can imagine the covariate simply being $\mathbf{X}$ a $n \times 1$ vector representing the spike train for a single neighbouring cell, such that the linear predictor (for the GLM) can be written as

$$\eta(t) = \beta_0 + \mathrm{X}(t)\beta_1 \tag{42}$$

where $\eta(t)$ is the linear predictor for $Y(t)$, where $Y(t)$ is the t'th entry of the response spike train $\mathbf{Y}$, and β_1 a model parameter.

However, using the original spike train $\mathbf{X}$ as the only model covariate, as in (42), is not a very good model for detecting connectivity between spike train, since we are testing for a connection between the spike trains due to what happens at the exact same time only.

As explained in Section 1.2.3, we expect there to be a time lag between related events in paired spike trains. Thus, we want to transform the original spike train $\mathbf{X}$ into many model covariates $z_d(t)$, using basis function expansion as explained in Section 3.4, in such a way that we are able to better detect connectivity between cell pairs. The results is that we get more model covariates that represents the activity of $\mathbf{X}$, such that the corresponding linear predictor can be written as,

$$\eta(t) = \beta_0 + \sum_{d=1}^{4} \beta_d z_d(t) \tag{43}$$

where $z_d(t)$ for $d = 1, 2, 3, 4$ is the basis function expansion model covariates, explained in Section 3.4, that we get from expanding the original spike train $\mathbf{X}$. With the linear predictor $\eta(t)$ from (43) we can explain the connectivity for the cell pair (with the precision given by the number of basis function expansion model covariates). However, we are still able to improve the precision that we are able to detect connectivity between cell pairs with this model by including more model covariates. These new model covariates filters out "noise" from the model. By noise in this setting, we mean anything that is unrelated to the connectivity effect between the cell pair under investigation. Thus, these new covariates explains away the "connectivity" effects that are not of interest.

Adding extra model covariates The new linear predictor that we get using such a method can be written as

$$\eta(t) = \beta_0 + \sum_{d=1}^{4} \beta_d z_d(t) + \sum_{d=5}^{r} \beta_d^* z_d^*(t) \tag{44}$$

where the first part of the equation is the same as in (43), while the last part $\sum_{d=5}^{r} \beta_d z_d^*(t)$, represents any additional effect added by model covariates $z_d^*(t)$.

From Kass et al. (2014) we have that a natural division of the model covariates is; history effect, connectivity effect, external effect. Additionally we are aware that there are remaining effects that may be important that doesn't fit in either of these categories. Thus we can write our linear predictor for such a model as

$$\eta = \text{history effect} + \text{connectivity effect} + \text{external effect} + \text{other effects} \tag{45}$$

History and connectivity effects In Equation (43) we already model for connectivity effect, but only for one single neighbouring neuron, thus to model history and connectivity effect we need to extend the linear predictor as in (44). Thus, the most natural new linear predictor in our case would be,

$$\eta(t) = \beta_0 + \eta^{\text{connect}}(t) + \eta^{\text{hist}}(t) = \beta_0 + \sum_{r=1}^{R} \sum_{d=1}^{4} \beta_{d,r}^{neigh} z_{d,r}^{\text{neigh}}(t) + \sum_{d=1}^{d^{\text{self}*}} \beta_d^{\text{self}} z_d^{\text{self}}(t) \tag{46}$$

where we model connectivity effects for R neighbouring cells, where each cell has four basis function generated model covariates as in (43). Thus, $z_{d,r}^{\text{neigh}}$ is the d'th basis function for the r'th neighbouring cell and $\beta_{d,r}^{\text{neigh}}$ its corresponding parameter. Furthermore we have added an undefined $d^{\textbf{self}*}$ number of model covariates for history effect which is treated exactly the same as connectivity effect, except that the basis for the model covariates that basis function expansion is used upon is the response spike train $\mathbf{Y}$ itself. z_d^{self} represents the d'th history model covariate and β_d^{self} its corresponding parameter. Here the history effect got the same number of model covariates as any of the neighbouring cells, we may want the number of model covariates for history effect to be higher since we expect the cells own history to have a higher prediction value for the cell's own behaviour than an arbitrary neighbouring cell.

External effect In Fawad (2017) and Aga and Fawad (2017) a different type external (stimulus) effect was used than in this thesis. In these papers the stimulus varies as the trial progresses through its three stages, sample epoch, delay epoch and response epoch. A simple approach to incorporate the effects of these periods is to include the trial time as a covariate. Such trial time may have non-linear effects on the firing rate of a neuron. Thus such stimulus can have both a linear and a non-linear effect, leading Fawad (2017) to suggest the linear predictor

$$\eta_j(t_i) = \alpha_{0j} + \sum_{d=1}^{D} \gamma_{dj} t_i^d, \tag{47}$$

were the polynomial is represented with a set of orthogonal bases $\{P_1(t_i), ..., P_D(t_i)\}$ called Legendre polynomials. Each $P_k(t_i)$ is a polynomial of degree k, and can be expressed using Rodriguez formula Rodríguez (2007).

$$P_k(t) = \frac{1}{2^k k!} \frac{d^k}{dt^k} (t^2 - 1)^k \tag{48}$$

Thus, resulting in

$$\eta_j(t_i) = \alpha_{0j} + \sum_{d=1}^{D} \gamma_{dj} P_d(t_i). \tag{49}$$

However, as our data set, and thus our stimulus/external effects are different, we would need different model covariates for external effects. The external effects in Aga and Fawad (2017) have very few natural choices for model covariates as it is a sequential order of three discrete levels. However, in this thesis we have many continuous 1-D external effects, such as speed and HD, and we have a continuous 2-D external effects, position, which gives us many additional reasonable model covariates to include into this GLM model.

The 1-D effects can be modelled by dividing them into time bins such as the response spike train, here the t'th element in the speed model covariate $X_{Speed}(t)$ can simply be the speed of the rat at the given millisecond. Such a modelling allows for detecting a linear trend between the speed of the rat and the firing rate in the response neuron. Potentially we can model the external effect of the speed with more than one model covariate to allow for detecting non-linear effects between the speed and the response.

Furthermore, to model HD we would recommend to divide HD into angle bins as done in Section 2.3.2. Thus, we get a model covariate for each angle bin i, thus $X_{HD(i)}(t)$ represents the model covariate for the t'th element in time for the i'th angle bin. Here a natural choice is to represent $X_{HD(i)}(t)$ with a 1 if the rat had its head in the direction of the i'th angel bin in the time represented by the t'th element, and 0 otherwise.

Additionally we got even more model covariates for the position as it's a 2-D effect. There are two apparent natural ways to model this external effect

- The non-overlapping discrete case. Divide the box into non-overlapping squares where each one of them are represented by a model covariate. If the rat is in the given position at a given time, the elements value is 1, otherwise 0. This suggestions is the 2-D analogy to the suggestion for HD modelling.
- The overlapping weighted case. In an effort to reduce the number of model covariates without reducing the modelling precision we introduce the overlapping weighted case. In this case we suggest to introduce a number of equidistant focal points that span the box, each which measures the distance of the rat from the given point at each time period.

Additionally we may want to add some basis function expansion, in the same way as we do for connectivity effect, to also evaluate the effects with a delay in time.

Summary This model is considered good for detecting connectivity between spike train. In order for this method to be even better we need to add more model covariates. Although the methodology and idea of this is simple, we run into computational and practical challenges doing so. Thus, in

practice we have to set a limit for how many model covariates the model can include, and thus how precise it can be.

With JGLM we seek to bypass this problem with more directly utilising the temporal information in the spike train sequences, which we consider the essential part in being able to detect connectivity between cells.

4.5.2 JGLM framework

In short the JGLM framework can be summarised as.

1) We want to fit a GLM to a pair of spike trains to see whether they are connected. This can be done as explained in Section 4.5.1. Once the model is fitted, we can evaluate the likelihood to see whether the chosen model covariates make for a good fit between the cells.

2) We deem the previous point, not a good enough test for connectivity, as detecting a good fit in this way, i.e. connectivity may easily be confounded with other external factors.

3) We consider a variation of permutation test for the fitted GLM results, particularly developed for pairs of spike trains, to be a good enough test for detecting connectivity between cells. The permutation process is explained in Section 4.2.1, and the testing is explained in Section 4.4.1.

4) We want to test many cell pairs, thus we may want to adjust for multiple testing. In our case we leave the p-value at $p = 0.05$ but one can adjust the p-values with the Bonferroni correction. The p-value adjusted with Bonferroni is $p \leq \alpha_{LOC} = \frac{0.05}{m}$, where m is the number of cell pairs we are comparing.

5) If we detect connectivity between two neurons, we classify the connection type by using the basis tuning curve, as explained in Section 4.4.2.

A more philosophical discussion of the JGLM framework can be seen in Section 7.

III Statistical analyses

5 Regression model

In this chapter define the response variables and the model covariates used in the JGLM model in Sections 5.1 and 5.2, respectively.

5.1 Spike train as response variable

The activity of a cell is recorded as a spike train. Simply put, a cell can be in either of two states, a resting state or an active state. The point in time when a cell transitions from a resting state to an active state is called a *firing*. When analysing neural data, the rate at which a cell fires, and the proportion of time it fires, is of interest. An important concept in neurophysiology is that cells respond to a stimulus or contribute to an action by increasing their firing rate (Kass et al., 2014, p. 563). Formally the firing rate at time t is defined as

$$FR(t|x_t) = \lim_{\Delta t \to 0} \frac{\mathrm{E}(\text{number of spikes in } (t, t+\Delta t)|x_t)}{\Delta t}, \tag{50}$$

where x_t can incorporate any experimental conditions, any history effects of previous spikes or even activity of neighbouring cells. The numerator in (50) is the expected number of spikes in a time interval of length Δt.

Consider an observed spike train $s_1, s_2, ..., s_m$ over a time interval $(0, T]$, where the s_i's are the times at which a spike occurs. Such a spike train is modelled as a point processes $S_1, S_2, ...$ on $(0, \infty)$ (Kass et al., 2014, p. 564). To analyse a point process within the framework of the GLM, we need to discretise the spike times (Kass et al., 2014, p. 568). A way towards discretisation is the counting process representation of the point process, $N(t)$. The function $N(t)$ counts the total number of spikes that have occurred up until and including time t. Next, we divide the finite observed time interval $(0, T]$ into n bins of equal length $\Delta t = T/n$. The number of spikes in bin i can now be counted as $\Delta N_i = N(t_i) - N(t_{i-1})$, where $t_i = i \cdot \Delta t$. The set $\{\Delta N_i; i = 1, ..., n\}$ is called the discrete increments, and it is displayed in Figure 22 along with the spike times and the counting process. Note that both the point process $S_1, S_2, ...$ and the counting process $N(t)$ are stochastic processes in continuous time, while the discrete increments $\{\Delta N_i; i = 1, ..., n\}$ are a sequence of random variables in discrete time.

Let $Y_i = \Delta N_i$. If we choose a small enough bin size Δt, it is unlikely that there will be more than one spike occurrence in a single bin (Kass et al., 2014, p. 568). That is, $P(Y_i > 1) \approx 0$. Hence we have that

$$Y_i \sim \text{Bernoulli}(p_i), \tag{51}$$

where $p_i = P(Y_i = 1)$. This is the case illustrated in Figure 22. Furthermore, since $\mathrm{E}(Y_i) = p_i$ we can rewrite the firing rate defined in (50) as

$$FR(t|x_t) = \lim_{\Delta t \to 0} \frac{P(\text{spike in } (t, t+\Delta t)|x_t)}{\Delta t}.$$

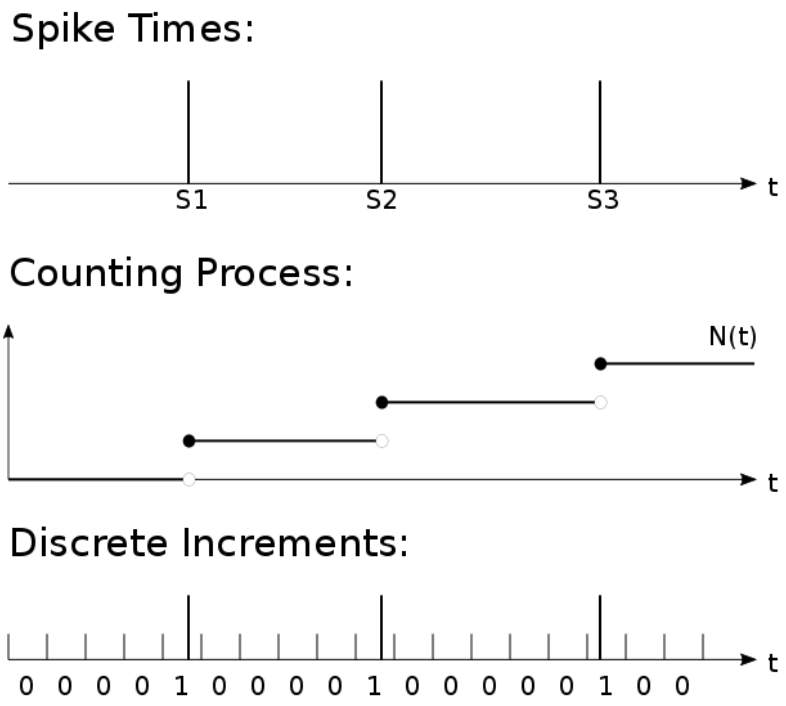

Figure 22: The spike times $S_1, S_2, ...$ are random variables, each representing the point in time when a spike occurs. $N(t)$ is a cumulative count of spikes that have occurred up until and including time t. The discrete increments ΔN_i count the number of spikes in bin i. This figure is taken from Kass et al., 2014, p. 567, with permission from Springer.

5.1.1 Joint PDF approximations

The joint pdf of a sequence $S_1, ..., S_M$ from an inhomogeneous Poisson process, with intensity function $\lambda(t)$, over an interval $(0, T]$ is given by (Kass et al., 2014, p. 574)

$$f_{S_1,...,S_M}(s_1, ..., s_m) = \exp\Big(- \int_0^T \lambda(t)dt \Big) \prod_{i=1}^{m} \lambda(s_i). \tag{52}$$

The reason for assuming non-stationarity is because the p_i's in (51) are assumed to vary between bins.

Next, consider a binary sequence $Y_1, ..., Y_m$ constructed in the same manner as in Section 5.1,

over the same time interval $(0, T]$. The joint pdf of this binary sequence can be derived as

$$\begin{aligned} f_{Y_1,...,Y_m}(y_1, ..., y_m) &= \prod_{i=1}^{m} f_{Y_i}(y_i) \\ &= \prod_{i=1}^{m} p_i^{y_i}(1-p_i)^{1-y_i}, \end{aligned}$$

where we've assumed that the Y_i's are independent, and used (51) in the last step. Now according to (Kass et al., 2014, p. 575), as $\Delta t \to 0$ we have that

$$\frac{1}{(\Delta t)^m} \prod_{i=1}^{m} p_i^{y_i}(1-p_i)^{1-y_i} \to \exp\Big(-\int_0^T \lambda(t)dt\Big) \prod_{i=1}^{m} \lambda(s_i), \tag{53}$$

where we've defined

$$p_i = \lambda_i \Delta t, \tag{54}$$

where $\lambda_i = \lambda(t_i)$, and t_i is the midpoint of bin i. Note that we've scaled each $f_{Y_i}(y_i)$ in (53) by $1/\Delta t$, to avoid that $p_i \to 0$ when $\Delta t \to 0$, where p_i is defined in (54).

5.2 Connectivity as model covariates

In this work the model covariates are the pairwise history effect from the subset of cells chosen in Section 2.3.3 presented in Table 1. Each spike train of these cells have been modelled by cosine bumps basis functions as explained in Section 3.4, into four new covariates, weighted as represented in Figure 18.

5.2.1 Other model covariates

Additional covariates, as discussed in Section 4.5.1 may be history effect and external effect. History effect which can be modelled in the same fashion as connectivity effect, just that the in this case the cell is paired with itself. We have in this work only modelled connectivity effects between cell pairs, that is one response cell $\mathbf{Y}$ is modelled by one neighbouring cell $\mathbf{X}$ and its basis function expansion model covariates.

In this work there are two main challenges with using external effects as model covariates.

- Number of covariates. Challenges with many covariates as mentioned in Section 4.5.1
- Quality of movement data. Explained in Section 2.2

6 Data analyses

In this chapter we present the part of our analyses were tools from Part II are used. All analyses was done on the time period 1 (0-10 first minutes), see Section 2 for further description, with a p-value off 5%, see Section 4.5.2 for discussion, on the subset of cells described in Section 2.3.3. Our main goal is to estimate connectivity between cells. In Section 6.1 and 6.2 we will in detail look at three specific cell pairs. In Section 6.3 we will look at the connectivity's for all the cell pairs in the subset in order to investigate the information in the network.

We have decided to treat the data in the programming language R Core Team (2016), packages that was used which is due credit are 'ggplot2' Wickham (2009) used for graphics and 'corrplot' Wei and Simko (2017) used for plotting Figure 25. Additionally 'STAR' Pouzat (2012) was used for spike train representation and 'smoother' Hamilton (2015) used for constructing movement video.

6.1 Finding connections

First we use the model following the description in Section 5 to find connections between cell pair from the subset of HD and spatially tuned cells analysed in Section 2.3.3, overview of which cell tag number belongs to exactly which cell can be seen in Table 2. The connections between these cells have been found by the permutation test on the likelihood values generated by JGLM with interval jittering as discussed in Section 4.1 and Section 4.5.2. In the analysis done the number of jittering was set to 1000. In this and the next section we look further into the one way communication for three cell pairs

A, information flow from cell 10 to 12 (from T11C2 to T12C13), see Table 2, was identified as significant with both interval and basic jittering.

B, information flow from cell 7 to 2 (from T8C3 to T3C30), was identified as significant with basic but not with interval jittering.

C, information flow from cell 3 to 12 (from T4C3 to T12C13), was identified as significant with interval but not with basic jittering.

From Figure 26 and 27 we have that JGLM with basic jittering produces many more significant connections than interval jittering. Case C is in fact the only connection identified where interval is significant but basic is not, but both the situation in case A and B, where either just Basic or both basic and interval is significant is quite common.

The distribution of log-likelihood values from JGLM procedure with interval jittering and the original log-likelihood value are illustrated for all the three cases in Figure 23. Thus from Figure 23, the p-value that indicate whether there is a connection or not between the cell pair is illustrated. For case A and C we have that the original log-likelihood is in the rightmost 5 percent quantile of the jittered log-likelihood density, but this is not the case for cell pair B.

6.2 In depth analysis of specific connections

To study the three cell pair cases in more depth we analyse them with non-jittered GLM and Basis-TC as well.

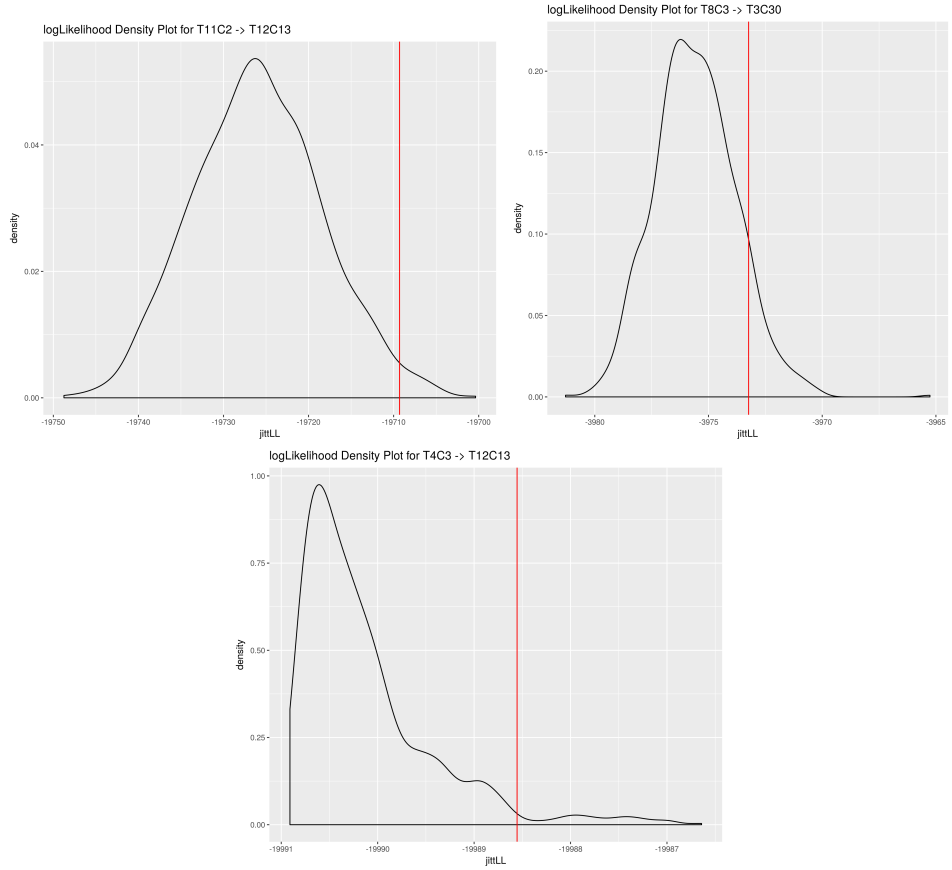

Figure 23: From the p-values we identify one of the significant connections. This significance is illustrated in the plot, were the black curve represents the density plot for all the 1000 jittered log-likelihood-values, while the red line represents the original log-likelihood-value.

Analysis with non-jittered-GLM. For analysis with non-jittered-GLM we get results presented in Table 4. If we were to only test for connection between cell pairs with non-jittered-GLM with the given model covariates, the reasonable choice seems to classify any cell pair with at least one significant model covariate as significantly connected. However, we expect most cell pairs to have at least one significant model covariate, even though they are not connected, as we consider the non-jittered-GLM test to test for more than just the connectivity between cells. An overview over connections based on this type of classification can be seen in Figure 28, and we see that in fact close to all cells are classified as connected with this type of test.

Furthermore we want to look at the specific results for the three cases in Table 4. For case A we have that all four model covariates are significant, for case B that three of four model covariates are significant, while we have zero significant model covariates for case C.

It is as expected that case A is well connected according to non-jittered-GLM as it is both connected according to JGLM with basic and interval jittering. Furthermore it is natural that case

ModelCov	10-12 (T11C2-T12C13)	7-2 (T8C3-T3C30)	3-12 (T4C3-T12C13)
1	$p < 10^{-16}$ *	$5.28\ 10^{-6}$ *	0.8839
2	$p < 10^{-16}$ *	$6.08\ 10^{-5}$ *	0.9444
3	$p < 10^{-16}$ *	0.47402	0.3789
4	0.0104 *	0.00923 *	0.0554

Table 4: P-values for the fitted model covariates for connection from 12 to 3 and from 8 to 2, for non-jittered GLM. We can see that both of them have at least one significant model covariate, i.e. using non-jittered GLM as this, we would classify both connections as significant. 10-12 significant for I1 B1. 7-2 significant for B1 but no I1. 3-12 only that is significant for I1 but not for B1.

B is also well connected due to non-jittered-GLM, but not quite as much as case A, as it is connected according to JGLM with basic but not with interval jittering. However, more surprisingly is it that case C has no significant model covariates. Case C is in some sense an odd case, since both basic and non-jittered GLM are expected to exaggerate connections, yet they do not indicate connection for case C, but interval does indicate connection, which we do not expect to exaggerate connections. It is thus tempting to think of case C as an false positive by the interval JGLM test, however there is no way to know for sure with this data set. However if it in fact were a false positive there are three reasonable ways to deal with this

- Leave it as it is, as there will always be false positives.
- Change the p-value cut-off to take into account multiple testing.
- Lastly, as in case C, the connection passed the interval JGLM test, but not the non-jittered-GLM, we may consider double testing, both with interval JGLM and non-jittered-GLM. This is not standard procedure to double test, but the argument for would be that interval JGLM and non-jitt-GLM test tests for different features for the connection, and that it may be reasonable to demand for the cell pair to have a certain threshold for both features to be classified as significant.

However, the fact that the results from case C is somewhat surprising doesn't mean that anything is classified incorrectly, thus before changing anything this hypothesis should be tested on bigger scale with a "ground-truth" data set, where the truth of the connections are known.

The idea of ground-truth testing is additionally of interest for testing the whole JGLM framework, more thoroughly discussed in Section 7.

Analysis with basis TC Once we have found connections as described in Section 6.1, we are interested in analysing the particular connection in further detail. One important tool in doing so is the use of the Basis TC described in Section 4.4.2. For the three specific cases illustrated in Figure 23 we get the following Basis TC plots shown in Figure 24. There are many potential ways to use the basis-TC plots to identify connections, here we have chosen to let the first time the original basis-TC curve are outside the 2 standard deviation confidence interval of the mean curve decide the connection type.

From these plots we can see the following.

A - Indicates a clear excitatory common connection.

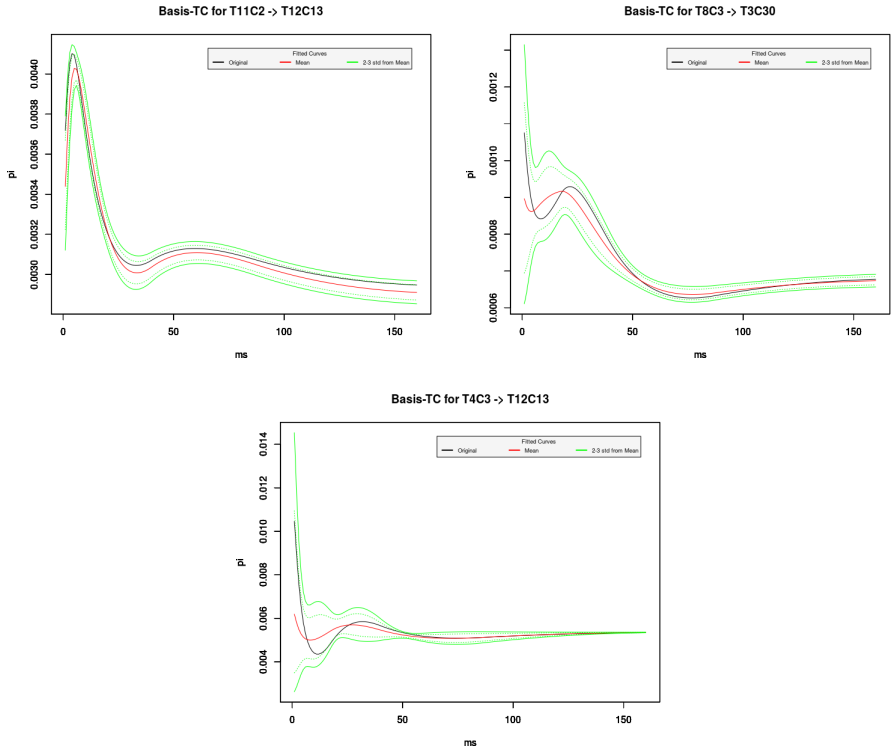

Figure 24: We want to investigate the type of connection by basis-tuning-curve. The black line represents the original basis tuning curve, the red line the mean of all the basis tuning curves from the jittered results and the green line 2 (dotted) and 3 standard deviations from the red line. The upper left picture is case A, see Section 6.1, the upper right is case B and the bottom picture is case C. Both case A and C indicates a common excitatory connection between the cells, as the black line is outside the 2 standard deviation interval at the 0-5 ms lag. Case B indicates no connection as we would expect, as it is not significant for interval JGLM test.

B - No clear indication off connection.

C - Indicates a excitatory common connection (but not as strongly as in case A).

As expected case A got the most clear indication for type of connection from the Basis TC plot. However, B shows no connection, and C shows the same type off connection as A but with weaker significance. Although, we could argue that case B is more significantly connected than case C, the results make sense as the basis-TC plots plotted are based on interval jittering. Thus, if the interval JGLM p-value is significant, we would expect basis-TC based on interval jittering to indicate a connection and vice versa. Thus, it is superfluous to use basis-TC to classify if we have a connection when we have test by likelihood explained in Section 4.4.1, however basis-TC plots are very useful to gain insight into the type of connection and whether the connections is excitatory or inhibitory, see Section 1.2.3.

Alternative basis-TC use As mentioned there are many alternative choices for identifying connection types with basis-TC. In the given framework, when we use basis-TC when we have already classified if there are a connection or not. Thus, it could be argued that we do not really need a confidence interval, as every connection has a connection type.

Given the framework we are using, when we are using basis-TC we have already classified the connection, but not which type. The connection however will always have a type, yet we will always have a degree of uncertainty in classifying the type. Thus, choosing the size of the confidence interval or classification method could be done in many ways. One alternative choice is instead of looking at the first time the original curve is significantly different than the mean curve, we could identify the position where the original curve is the most different from the mean curve. These two criteria often coincides, especially with the basis functions chosen (where the weighted value of the basis function decreases with temporal lag).

Of additional interest is that the basis-TC oscillates in time between indicating an excitatory and inhibitory effect. This is quite common as the temporal timing of the spikes are of importance in the spike train. For case A, this means that for a single spike in T11C2 at lag 0 increases the chance for a spike in T12C13, while a spike at lag 20 in T11C2 actually decreases the chance of a spike in T12C3 slightly.

Basis-TC as a generalisation Furthermore it should be noted that the basis-TC plots only show the variation of the probability for firing according to our model for a single spike in the sender-cell in the 0-160ms interval. Thus it does not show what happens when the cell spikes more than once in the 0-160ms interval. For two spikes in the plotted interval there would be no single basis-TC curve, but rather an ensemble of 159 different curves, one for each of the different temporal combinations of spikes in the interval. Thus, the basis-TC is a generalisation of the representation of the influence spikes in cells have on other cells. Yet, it is assumed, from our understanding of cells from Section 1, that the singular spike version of basis-TC presents a good picture for how the multiple spike versions of the basis-TC look like. That is to say that two spikes at temporal timings that individually increases the firing rate in the pair cell, would additionally increase the firing rate probability for the neighbour cell when fired in the same interval, (and not change the behaviour completely i.e. inhibit firing rate).

6.3 Network of neurons

We have in Section 6.1 and 6.2 looked in depth at the connection between three directed cell pairs. We now want to look at the connections between all the cells (from the subset given in Table 2) simultaneously. The connections are still identified with p-value at 5 percent. Figure 26 illustrates these connections for interval JGLM, Figure 27 for basic JGLM, and Figure 28 for non-jittered-GLM test. In addition to these plots which represents significant connections between cells, we have Figure 25 with p-values for connections, identified by the test by likelihood see Section 4.4.1 for both basic and interval jittering.

Different type of network identified From these figures we can see that basic JGLM indicates more connections than interval JGLM. This coincides well with what was reasoned in Platkiewicz et al. (2017), that basic jittering can exaggerate the number of connections one gets. Thus, from what was presented earlier in Section 4.2, interval JGLM (or any JGLM with non-centred jittering) should be preferred over basic JGLM (or any JGLM with centred jittering).

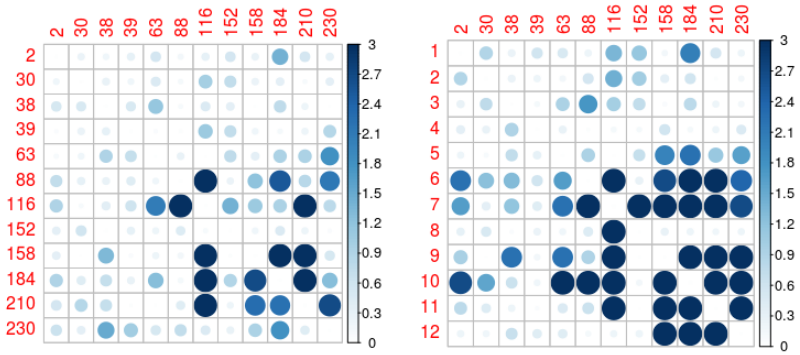

Figure 25: Correlation plot between connections from the test by likelihood, here element (1,2) represents a connection from cell 2 to cell 1. The value in the y-column represents the negative 10-logarithm value of the p-value from the test by likelihood defined in Section 4.4.1. The leftmost plot represents the value for this procedure by interval jittering, while the rightmost is for basic jittering. Thus, we can see that overall basic jittering indicates more connections than interval jittering.

Furthermore non-jittered-GLM indicates that nearly all cell pairs have a significant connection, even more than basic JGLM. As discussed in Section 6.2 we did expect our version of non-jittered-GLM to exaggerate the number of connections. Yet it remains an open question whether this version of non-jittered-GLM test is useful for a sort of double test as discussed in Section 6.2.

Neuroscience analysis on tuned cells From Table 2, we have that neuron 1, is both identified as HD tuned and ST, neurons 4,6 and 10 as ST and the rest are identified as HD. From Figure 26 we see that the ST tuned neurons receive more information from the other cells in the subset than they relay. Furthermore we can think of ST tuned cells as having a more complicated task than HD cells, as there are more ways to map the 2-D space, than there are different head directions. Thus, it is natural that HD cells communicate more with ST cells than vice versa.

Neuroscience analysis on cells from different brain areas The cells in the chosen subset resides in the brain areas described in Table 3. The subset of cells is too small to precisely say anything about the flow of information between different brain areas. But if we were to say anything

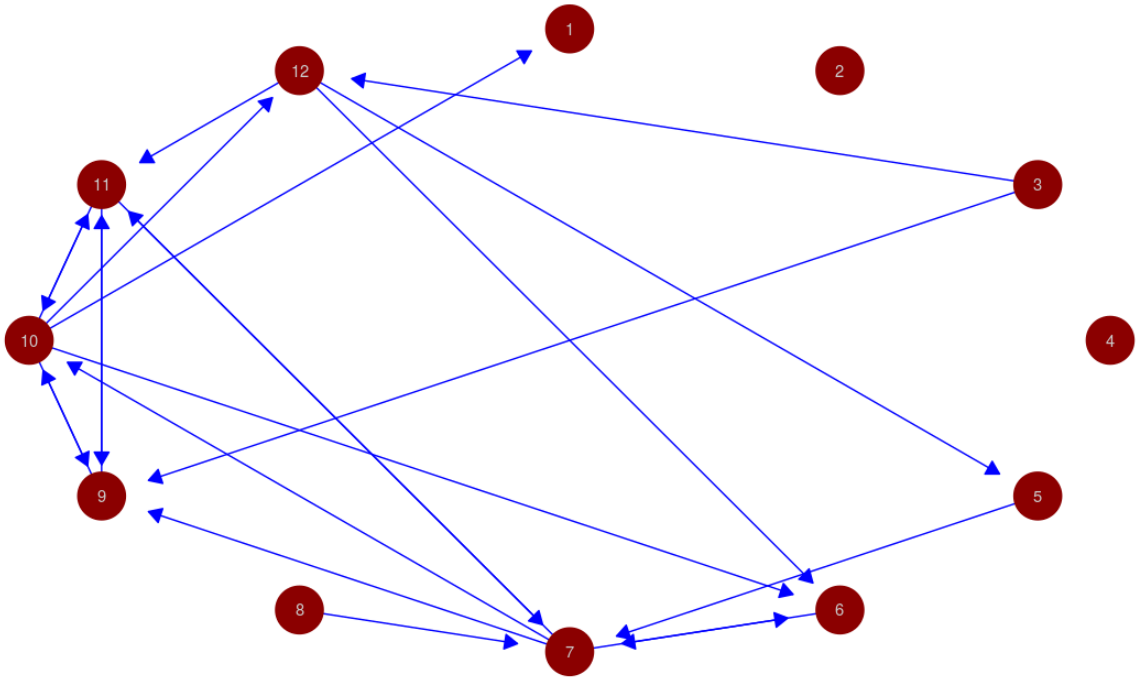

Figure 26: Significant connection representation of the network of neurons for JGLM with interval jittering.

from Figure 26, we would say that cells in the CA1 region have few significant connections compared to our other cells, and are only connected with EC-cells, mostly sending information towards them. The two cells from DG only receives information from the EC cells, which corresponds well with what is known about DG see Section 1.2.1, and communicate some with the two cells from CA2. Furthermore the EC cells got a high connection rate between other cells in the EC, in addition to communicating quite a bit with other cells.

It appears that the CA region is tasked to relay and receive information from EC. While the EC-cells in addition to relay and receive towards CA-cells regions, are heavily tasked with internal communication with other EC-cells.

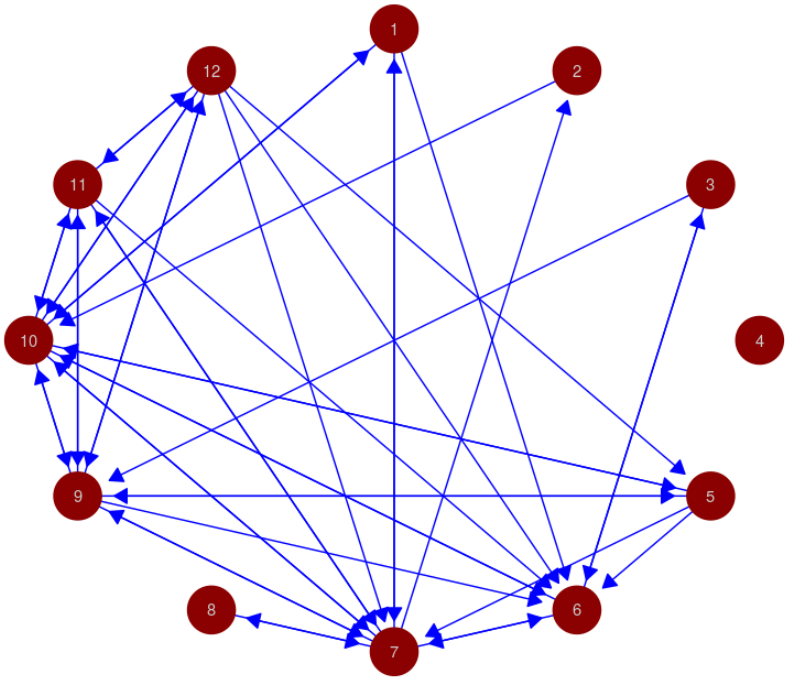

Figure 27: Significant connection representation of the network of neurons for JGLM with basic jittering.

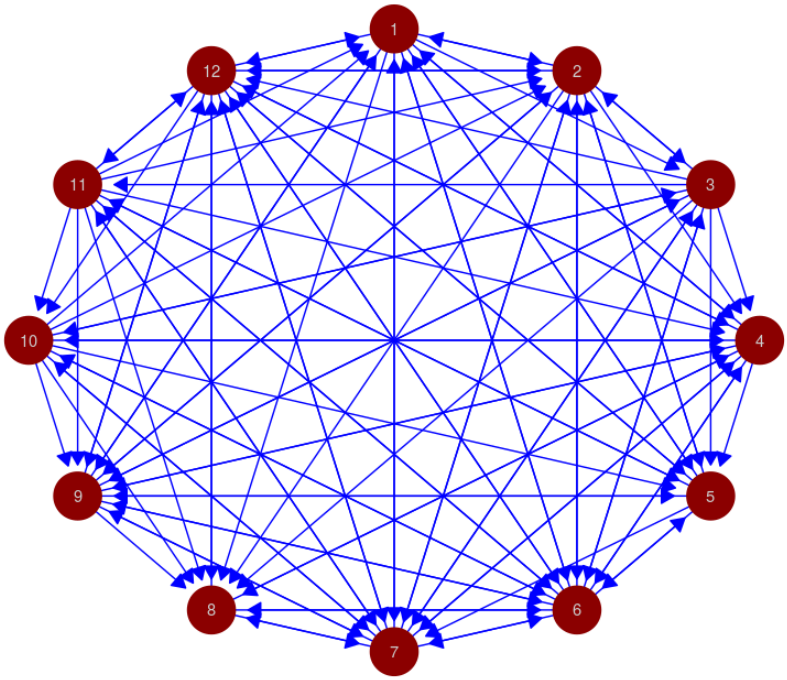

Figure 28: Connections from regular GLM by Wald test as explained in Section 6.2. Nearly all cells have significant connections for one or more of the 4 model covariates. Thus, testing for connectivity between neural networks with a GLM with few model covariates gives bad results. It is apparent that using JGLM with few model covariates gives much better results than using GLM with few model covariates.

7 Discussion and conclusion

In discussion and conclusion we will in Section 7.1 present a summary of the most important parts of our thesis. Thereafter for a particular summary of the challenges in the data analysis, see Section 7.2, for discussion about further development of JGLM see Section 7.3 and lastly to see the conclusion of this thesis see Section 7.4. As we have developed a new method, a lot of the motivation and discussion is written throughout the thesis. Therefore, it is recommended to consult the references in this section for a complete discussion and understanding of the choices made.

7.1 Summary

The original idea of the thesis was to analyse the information flow between HD and spatially tuned cells, however as discussed in Section 2.3, there were some challenges to identify tuning of cells in this data set, particularly for identifying cells of spatially tuning due to the quality of the movement data.

Thus, the natural focus in this thesis shifted towards analysing the neuronal network (and the tuning of the cells we were able to detect) as best as possible. The use of GLM have already been extensively used and described in Aga and Fawad (2017) and Fawad (2017). Thus, it was then wanted to compare the use of JCC and GLM for detecting connections between cell pairs. Yet, from this idea it arose a natural question of what the best part of JCC is and what the best part of the GLM was, and why either one of them would be better than the other. The pros and cons of each method have been discussed in Section 4.

The idea of the author was that, the most important part of the JCC was the jittering, while the most important part of the GLM framework was the rigid proper statistical idea and framework behind it. Thus, the next idea was to combine the best part of the GLM and the JCC into one, which resulted into the JGLM framework. From the results in Section 6 we see that JGLM appears to be a good model for detecting connectivity between cell pairs, but that further testing and maybe modification of the JGLM as noted in Section 7.3 is needed. In Section 4.2 we discuss why interval jittering fits the JGLM and JCC framework very well, while basic jittering does not. Additionally the development of the basis-tuning-curve, which is very useful for classifying the type of a connection between neurons, is explained in Section 4.4.2. Furthermore, as commented in Section 6.3, from our limited test sample it appears that HD tuned neurons talk more to ST neurons than vice versa.

7.2 Challenges in the data analysis

As described in Section 2 one of the main challenges in this work was the data science analysis of the cell identification. Unfortunately the movement part of the data was not sufficient for precisely classifying tuning of cells (particularly that of spatially tuning and grid cell tuning). Thus, as mentioned in Section 7.1, the focus in the thesis shifted towards developing the JGLM framework.

Thus the next main challenge in the thesis was to develop the JGLM framework and managing all the choices that came with it. The jittering method was discussed in Section 4.2. It was chosen to jitter the covariates together instead of jittering the response. Additionally we chose to jitter before constructing the basis function covariates. These choices are subject to further study and also commented in Section 7.3.

In Section 6 the results from the analysis are presented, and gives us an indication of the the effectiveness of the JGLM framework. Yet in neuroscience it is though to know whether a method for detecting connections between cell pair is any good at all as we do not truly know how the brain works, see Section 1. By toady's measuring devices it is seemingly impossible to classify this with absolutely certainty, yet using the method on a ground-truth data set holds very good promise, see Section 7.3.

7.3 Future work

The data set In this work we have analysed a subset of all cells for a subset of the total session time with JGLM. Thus, there are still a lot of information about the connectivity and flow of information in the neuronal network. For future work it is recommended to analyse all cells, for all times. It is also of interest to analyse only the times where the movement is good separately from when the movement is not so good, and cells with significant tuning with other cells with significant tuning and so on. Thus, there are also many interesting combinations of subsets of the data to be analysed. Additionally it is of interest to test the methodology on other data sets.

JGLM model choices There are many choices for developing JGLM further, some of the important choices to consider is the following.

- We have the choice of which type of jittering procedure to use. As stated in Section 6 we now know that interval jittering is preferred over basic jittering procedure. We have chosen interval jitter, as it is an intuitive easy-to-use jittering procedure that is commonly used.
- We have the choice of whether to jitter the covariates together or separately. We have here chosen to jitter the model covariates together, as their common temporal effect then are allowed to remain, but yet we remove the exact temporal timing of the model covariates. This is an intuitive viable way to set up the JGLM as we consider it important to let the common timing remain. We believe that to jitter the covariates separately unnecessarily removes too much of the original signal, and creates an entirely new relation between the spike trains, even for non-connected cell pairs. Thus, we believe our choice to be the best choice, however, as we do not perfectly know how cells communicate, changing this feature may allow us to test for a different communication pattern.

Although we believe to have chosen the most intuitive and best jittering procedure, the mentioned model choices are still worthy of further investigation and testing.

Ground-truth testing The most important part of the future work is to better test if the JGLM is a better method for detecting connectivity between cell pairs than JCC or GLM. Unfortunately we are not able to measure electrophysiological data from many cells and know their true connections to other cells. However, we can replace the current data set with a ground-truth data set, where the actual connections between the cell pairs are known. Such a ground truth data set will need to be constructed from the textbook example of how, we believe, the cell works. Thus, given that the textbook example of cell communication is to be relied upon, we are able to perfectly test and find the best method.

Additional model covariates In Section 4.5.1, the idea of JGLM were summarised as a simpler way to model connectivity between cell pairs, than the GLM alternative which requires many model covariates. Additionally we would like to suggest the use of JGLM with many model covariates to increase the model precision even further. Section 4.5.1 presents how we can add extra model covariates. When the number of model covariates become many it is of interest to include some factor into the model that restrict the excessive computing time due to the extra model covariates. One such tool is the use of lasso regularisation used in Fawad (2017), where the least influencing model covariates are essentially left out of the model.

Thus the JGLM framework provides two ideas. The simpler way to more effectively detect connectivity between cell pairs, as we have done. Additionally it provides the possibility for a framework with better detection possibilities than the regular GLM, where we include as many covariates as possible. This deviates from the original idea where we want to investigate JGLM as GLM requires longer computational time, which JGLM bypasses. However, the idea presented that JGLM utilises the temporal information in the spike train more effective than the GLM framework is still valid, and what makes JGLM interesting.

Multiple testing We have presented Bonferroni as an example for multiple testing, but not made use of it. Additionally we have the potential of using the false discovery rate (FDR) method of Benjamin-Hochberg. Both of these methods could have been utilised in an effort to adjust for finding too many connections.

Alternative use The JGLM also holds potential for revealing any type of information in a neural network, and not just only information about connectivity between cell pairs.

7.4 Conclusion

We have implemented a method for identifying tuning of cells which appear to be good. However, we require higher quality spatial data to be able to better identify tuning of cells.

Furthermore we have developed what we have chosen to call basis-TC plot, which is excellent for classifying the type of connections between cell pairs, see Section 4.4.2 and 6.2. Additionally we have discovered that interval jittering is better than basic jittering both in the JGLM and the JCC framework for analysing neuronal activity.

As discussed in Section 4.5 we have made a method called JGLM which appears to be a great new tool for detecting connectivity between cell pairs. The most important part of developing it further is to test whether or in which case this method is better than the previous methods, this can be done as described in Section 7.3.

References

Aga, K. and H. Fawad (2017). Modelling neuronal activity with generalized linear models.

Amarasingham, A., M. T. Harrison, N. G. Hatsopoulos, and G. Stuart (2011, oct). Conditional modeling and the jitter method of spike resampling. *Journal of Neurophysiology 107*(2), 517–531.

Dobson, A. J. and A. G. Barnett (2008). *An Introduction to Generalized Linear Models, Third Edition*. Texts in Statistical Science. Boca Raton, FL: Chapman & Hall/CRC Press.

Fawad, H. (2017). Modelling neuronal activity using lasso regularized logistic regression.

Hamilton, N. (2015). *smoother: Functions Relating to the Smoothing of Numerical Data*. R package version 1.1.

Kass, R. E., U. T. Eden, and E. N. Brown (2014). *Point Processes*, pp. 563–603. New York, NY: Springer New York.

McCullagh, P. and J. Nelder (1989). *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Oliva, A., A. Fernández-Ruiz, G. Buzsáki, and A. Berényi (2016). Role of hippocampal {CA2} region in triggering sharp-wave ripples. *Neuron 91*(6), 1342 – 1355.

Peyrache, A., B. G. (2015). Extracellular recordings from anterior lateral motor cortex (alm) neurons of adult mice performing a tactile decision behavior. crcns.org. `http://crcns.org/data-sets/thalamus/th-1`.

Phipson, B. and G. K. Smyth (2010). Permutation P-values Should Never Be Zero: Calculating Exact P-values When Permutations Are Randomly Drawn. *Statistical Applications in Genetics and Molecular Biology 9*(1).

Pillow, J. W., J. S. L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli (2008). Spatio-temporal correlations and visual signaling in a complete neuronal population. *Nature* (454 (7206), .), 995–999.

Platkiewicz, J., E. Stark, and A. Amarasingham (2017). Spike-Centered Jitter Can Mistake Temporal Structure. *Neural Computation 29*(3), 783–803. doi: 10.1162/NECO_a_00927.

Pouzat, C. (2012). *STAR: Spike Train Analysis with R*. R package version 0.3-7.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Raudies Florian, Brandon Mark P, Chapman G William, and Hasselmo Michael E (2014, nov). Head direction is coded more strongly than movement direction in a population of entorhinal neurons. *Brain research 1621*, 355–367.

Rodríguez, G. (2007). Lecture notes on generalized linear models. `http://data.princeton.edu/wws509/notes/`.

Sakai, K. and Y. Miyashita (1994). Neuronal tuning to learned complex forms in vision. *Neuron-Report 91*, 1342 – 1355.

Salt, T. E. (2017, nov). Modern Techniques in Neuroscience Research, edited by Uwe Windhorst and Johansson. *Trends in Neurosciences 23*(5), 231. doi: 10.1016/S0166-2236(00)01578-2.

Skaggs, W. E., B. L. McNaughton, and K. M. Gothard (1993). An Information-Theoretic Approach to Deciphering the Hippocampal Code. In S. J. Hanson, J. D. Cowan, and C. L. Giles (Eds.), *Advances in Neural Information Processing Systems 5*, pp. 1030–1037. Morgan-Kaufmann.

Wei, T. and V. Simko (2017). *R package "corrplot": Visualization of a Correlation Matrix*. (Version 0.84).

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Appendix A R-code

In the following appendix, the code that is essential for our JGLM analysis for identifying connectivity between cell pairs is included.

A.1 Cosine Bases

The following R-code is taken from Fawad (2017) which replicated it from the Matlab-code that accompanied Pillow et al. (2008), which can be downloaded at `http://pillowlab.princeton.edu/code_GLM.html`.

```
getBasis = function(nBases, binSize){
  b=binSize*nBases
  peaks=c(binSize, binSize*10*nBases)

  #nonlinearity for stretching x axis (and its inverse)
  nlin= function(x){log(x+1e-20)}
  invnl=function(x){exp(x)-1e-20}

  #Generate basis of raised cosines
  yrange= nlin(peaks+b)
  db=diff(yrange)/(nBases-1)
  centers=seq(yrange[1], yrange[2],db)
  maxt = invnl(yrange[2]+2*db)-b
  iht=seq(binSize,maxt,binSize)
  nt=length(iht)

  raisedCosineBasis = function(x,b,db){
    (cos(max(-pi,min(pi,(x-b)*pi/db/2)))+1)/2   # c in thesis is replaced with b in
        script (since c in R is taken)
  }

  ihbasis =matrix(NA, nrow=nt, ncol=nBases)
  for( i in seq(1,nt)){
    for(j in seq(1,length(centers))){
      ihbasis [i,j]= raisedCosineBasis(nlin(iht+b)[i],centers[j],db)
    }
  }


  #orthogonal bases
  library(pracma)
  ihbas=orth(ihbasis)

  return(list(bas=ihbasis, bas_orth=ihbas, tau_N=maxt))
}
```

A.2 Firing Matrices

The following code is used to construct easy-to-access spike trains for all the 250 cells in the data set. It is divided into three time periods, the first period representing 0-10 min, the second period 90-100 min and the third period 100-120min. All the time in these periods are for when we have good movement data for the rat's movement.

```
### This dcript creates binned firing matrices for the interesting times.
```

```
### creates bigY1All, first 10minutes for all neurons. bigY2All 90-100 mins and
    bigY3All 100-120mins.
### cellCounter and tetCounter. for equal indeces as bigYAll it gives the cell and
    tetrode Number.
### also hzCounter for easy overview of activity.
library(ggplot2)

moveFreq=1250/32
extraBins=1000/moveFreq

minStart1=0
minEnd1=10
minInt1=minEnd1-minStart1
binsNeeded1=1000*60*minInt1
posBinStart1=round(1000*60*minStart1)+1
posBinEnd1=round(1000*60*minEnd1)

minStart2=90
minEnd2=100
minInt2=minEnd2-minStart2
binsNeeded2=1000*60*minInt2
posBinStart2=round(1000*60*minStart2)+1
posBinEnd2=round(1000*60*minEnd2)

minStart3=100
minEnd3=120
minInt3=minEnd3-minStart3
binsNeeded3=1000*60*minInt3
posBinStart3=round(1000*60*minStart3)+1
posBinEnd3=round(1000*60*minEnd3)

bigY1All=bigY2All=bigY3All=integer(0)
hzCounter=cellCounter=tetCounter=integer(0)
###Access every res <-> clu pair
for(tet in 3:12){

  tmpclu=read.table(paste("/home/shomed/k/kristag/Desktop/AntonioData/AYA9day16/
      AYA9day16.clu.",tet,sep=""))
  res=read.table(paste("/home/shomed/k/kristag/Desktop/AntonioData/AYA9day16/
      AYA9day16.res.",tet,sep=""))

  cellN=tmpclu$V1[1]
  clu=data.frame(V1=tmpclu$V1[-1])
  bigY1=matrix(0,binsNeeded1,cellN)
  bigY2=matrix(0,binsNeeded2,cellN)
  bigY3=matrix(0,binsNeeded3,cellN)


  ####Forandre tid her.
  resind1=which(res>(20*1000*60*minStart1) & res<(20*1000*60*minEnd1)) #look at
      first 10minutes.   #look at 90:120 minutes.
  res1=res$V1[resind1]
  clu1=clu$V1[resind1]

  resind2=which(res>(20*1000*60*minStart2) & res<(20*1000*60*minEnd2)) #look at
      first 10minutes.   #look at 90:120 minutes.
  res2=res$V1[resind2]
  clu2=clu$V1[resind2]
```

```
54
55    resind3=which(res>(20*1000*60*minStart3) & res<(20*1000*60*minEnd3)) #look at
          first 10minutes.   #look at 90:120 minutes.
56    res3=res$V1[resind3]
57    clu3=clu$V1[resind3]
58
59    ## this res-clu pair gets binned and stored in bigY.
60    for(j in 0:(cellN-1)){
61      bin1=ceiling(res1[which(clu1==j)]/20)
62      bin1=bin1-1000*60*minStart1
63      bigY1[bin1,(j+1)]=1
64
65
66      bin2=ceiling(res2[which(clu2==j)]/20)
67      bin2=bin2-1000*60*minStart2
68      bigY2[bin2,(j+1)]=1
69
70      bin3=ceiling(res3[which(clu3==j)]/20)
71      bin3=bin3-1000*60*minStart3
72      bigY3[bin3,(j+1)]=1
73
74
75      tetCounter=c(tetCounter,tet)
76      cellCounter=c(cellCounter,j)  #consider +1 indexing.
77      hzCounter=c(hzCounter,(length(bin1)+length(bin2)+length(bin3))*1000/(60*1000*10*
            4))  #for this time selection only
78
79    }
80    bigY1All=cbind(bigY1All,bigY1)
81    bigY2All=cbind(bigY2All,bigY2)
82    bigY3All=cbind(bigY3All,bigY3)
83  }
```

A.3 JGLM network identification

The following code is for identifying p-values with permutation test on likelihood-values with JGLM with interval jittering.

```
1  ### pConnectValue script. to find the p-value for connection between cells. (
       appendix 3)
2
3  setwd("/home/shomed/k/kristag/Desktop/MasterOppgave/WriteData")
4
5
6  source("getBasis.R")
7  library(pracma)
8  library(MASS)
9
10
11
12 jitterNr=1000
13 cellNr=c(2,30,38, 39,63,88,116,152,158,184,210,230) #The cells with highest Spatial
       Information detected.
14
15 N=length(cellNr)
16
17 ###set store vector/matrix to 0 in case it wasnt already.
```

```
18  pVal=integer(0)
19  pValMat=matrix(0,N,N)
20
21  ### model covariates parameter to match with getBasis()
22  nBases_connectivity=4
23  binSize=0.001
24  basOrth=PB=getBasis(nBases_connectivity,binSize)$bas_orth
25
26  ntp=dim(bigY1)[1] #6 or 12
27  ###While loop that get p-value for all cell pairs possible from cellNr. One-way
        connections only. Reverse cellNr and redo for a complete cell-pair p-value.
28  I=0
29  J=1
30  while(I<(N-1)){
31    I=I+1
32    J=I
33    while(J < N){
34      J=J+1
35
36      cellNr1=cellNr[I]
37      cellNr2=cellNr[J]
38
39      y=bigY1All[,cellNr1]
40      x=bigY1All[,cellNr2]
41
42      ####construct covariates for the GLM-fit.
43      basisCovariates=matrix(0,length(x),nBases_connectivity)
44      for(k in 1:nBases_connectivity){
45        tmp=convolve(c(0,x),rev(basOrth[1:160,k]),type="open")[2:(dim(bigY1All)[1]+1)]
46        basisCovariates[,k]=tmp
47      }
48
49      fit=glm(y[-c(1:160)]~basisCovariates[-c(1:160),],family=binomial(link=logit))
50
51      orgLL=logLik(fit)
52      orgCoeff=fit$coefficients
53
54      #make jittered GLM-fits. Store the coeffficient and LL temporary in
            intervalCoeff and intervalLL
55      #store the PermTest p-val from the evaluation of all LL-values for a cellpair in
            pValMat, which gets updated and repeatedly stored on drive for each
            cellpair.
56      intervalCoeff=integer(0)
57      intervalLL=integer(0)
58      set.seed(123)
59      for(i in 1:jitterNr){
60
61        jitteredX=matrix(0,length(x),1)
62
63        origFirInd=(which(x==1))
64        origFirInd=origFirInd[(origFirInd>5) & (origFirInd<(ntp-5))]
65        origFirInd=floor(origFirInd/11)*11    #Add this line to do Interval Jittering
              in stead of Basic Jittering.
66
67        jitteredFactor=sample(x=0:11,size=length(origFirInd),replace = TRUE)
68        jitteredIndex=origFirInd+jitteredFactor
69        jitteredX[jitteredIndex]=1
70
```

```
71          jitteredbC=matrix(0,length(jitteredX),nBases_connectivity)
72          for(k in 1:nBases_connectivity){
73            tmp=convolve(c(0,jitteredX),rev(basOrth[1:160,k]),type="open")[2:(ntp+1)] #
                  ntp+1 pga c(0,x)
74            jitteredbC[,k]=tmp
75          }
76
77
78          tmpfit=glm(y[-c(1:160)]~jitteredbC[-c(1:160),],family=binomial(link=logit))
79          intervalLL=cbind(intervalLL,logLik(tmpfit))
80
81
82        }
83
84        tmp=(sum(intervalLL>orgLL)+1)/(jitterNr+1)
85        pVal=cbind(pVal,tmp)
86        pValMat[I,J]=tmp
87        write.matrix(x=pValMat,file="firstMatrix",sep="␣")
88      }
89    }
90    pVal1=pVal
91
92
93    #intervalCoeff, intervalLL, orgCoeff,orgLL
94    df=data.frame(jittLL=t(intervalLL))
```