Modelling populations of spiking neurons in Autistic Spectrum Disorder (ASD)

Pierre G. Gianferrara

Master of Science
Cognitive Science and Natural Language Processing
School of Informatics
University of Edinburgh
2017

(Graduation date: Thursday 30 November 2017)

Abstract

This Master's thesis introduces the computational modelling of neural data analyzed using a novel choice of mathematical model: the population-tracking model [1]. The main purpose of the model is to capture the neuronal dynamics of large populations of neurons ($N > 25$). We modelled three different types of spikes datasets: synthetic data, multielectrode array data, and calcium imaging data collected in the context of a neurobiological experiment investigating the effect of SynGap knockouts (heterozygous genotype) on neural firing behaviours. Autistic Spectrum Disorder (ASD) is a range of intellectual disorders which involves genetic mutations as well as changes in neuronal networks dynamics. How can mathematical modelling be used in order to make sense of this neurobiological disorder? This is what we attempted to understand further through the implementation of the population-tracking model. Multielectrode array modelling analyses results mainly supported the findings from past research [2]. As for calcium imaging data analyses, we were only able to gather preliminary results which supported early experimental findings [3] and suggested that there may indeed be a difference in the degree of neuronal remodelling between wild type mice and heterozygous mice.

Acknowledgements

I wish to thank my Computational Neuroscience supervisor: Dr. Matthias Hennig, for having supervised me all year long, and for having helped me understand more about the fascinating field of neural information processing, and statistical modelling applied to Neuroscience. I am also grateful for having had the opportunity to take part in such interesting research investigating synaptic plasticity and the neurobiological basis of learning.

I would also like to thank two PhD students from Dr. Hennig's lab who greatly helped me throughout the project: Joseph Cronin and Martino Sorbaro.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Pierre G. Gianferrara*)

To Odile, Paolo & Elodie Gianferrara.

Table of Contents

1 Introduction **1**

2 Neurobiological background on Autistic Spectrum Disorder and the Fragile X Syndrome **5**
2.1 Overview of long-term synaptic plasticity and homeostatic plasticity . 5
2.2 The Fragile X Syndrome: From a genetic mutation to homeostatic dysregulation 7
2.3 Experimental investigation of the Fragile X Syndrome 9

3 Neurobiological experiments and modelling methodology **11**
3.1 Brief overview of the project's methodology 11
3.2 Description of the neurobiological experiments 12
3.2.1 The multielectrode array (MEA) experiment 12
3.2.2 The synaptic plasticity experiment 13
3.3 Overview of the mathematical modelling methods 15
3.3.1 The population-tracking model 15
3.3.2 Entropy computation and the sum-of-log-normals method . . . 18
3.3.3 The Kullback-Leibler divergence 23

4 Modelling tests, results and analysis **25**
4.1 Modelling tests on synthetic data 25
4.1.1 Description of the synthetically-generated spikes matrices . . . 25
4.1.2 Synthetic data modelling over time steps 28
4.1.3 Statistical comparison of multi-neuron activity 34
4.2 Multielectrode array (MEA) data modelling 38
4.3 Calcium Imaging data modelling 42

5 Discussion and Limitations **51**

6 Conclusion **57**

A Complementary results **59**
A.1 Complementary synthetic data baseline plots 59
A.2 Complementary synthetic data time steps results 61

Bibliography **65**

Chapter 1

Introduction

[1]The main goal of this Master's project was to implement a mathematical model of neuronal population activity on calcium imaging data collected with two-photon calcium imaging in the Integrative Physiology laboratory at the University of Edinburgh. One of the latest breakthroughs in the field of Neuroscience is the development of large-scale recordings for neural activity in the brain [5]. It is now possible to record from up to thousands of neurons simultaneously. According to Hertz et al. (2013) [6], learning more about the precise computational and physiological dynamics of networks of neurons requires one to first estimate the parameters of a statistical model. Indeed, for many years, statistical physicists have attempted to capture more of the functional behaviour of neuronal networks through the laws of statistical mechanics, fitting models to experimental data collected in Neuroscience laboratories [7].

Schneidman et al. (2006) [8] published an influential paper which reported that the pairwise maximum entropy model was a very good choice of statistical model which could capture more than 90% of the neuronal networks' correlations (spikes and silences) using an energy function constrained solely on correlations in the data and firing rates of individual neurons. Importantly, the main reason which explains why the pairwise maximum entropy model became successful was that it didn't require any specification of higher-order interactions in order to describe most of the correlations of the neuronal network [8].

Another paper from Roudi et al. (2015) [9] compared two types of statistical models that were used to model neurophysiological data: generalized linear models (GLM) and maximum entropy models. In the case of GLM's, the model considers the likelihood of the recorded history rather than the set of observed spike patterns. A function $J_{ij}(\tau)$ of a time lag τ models the influence of spikes on the probability of firing of pairs of neurons (pre- and postsynaptic neurons i and j). The idea of GLM's is to predict a neuron's future spikes based on both its past spikes and the activity of neighbouring neurons (only when included in the model through couplings). On the other hand, maximum entropy models only predict neuronal spiking based on the firing behaviour of other neurons in the network, but cannot model the time history [9].

In the context of large-scale recordings, research has shown that it was often

[1]Most of this Chapter is highly inspired from my *Informatics Research Proposal* [4]

better to record neural activity with electrode arrays rather than other neuroimaging methods. Indeed, electrode arrays can record simultaneously from thousands of channels with single neuron resolution, and they can also benefit from sampling rates over 20 kHz/channel, unlike most other neuroimaging techniques [10], [11]. However, in the context of *in vivo* neurobiological experiments, other neuroimaging techniques may be preferable to microelectrode arrays, such as two-photon calcium imaging.

Two-photon calcium imaging affords a certain number of advantages which are worth mentioning. First, traditional neuroimaging techniques such as optical imaging based on intrinsic signals afford a spatial resolution of around 250-500 μm [12] whereas two-photon calcium imaging affords a spatial resolution of 10 μm [13]. Second, through the use of a fluorescent genetically-encoded calcium indicator (GCaMP6f) and two-photon laser-scanning microscopy [14], [15], it is possible to record the precise calcium levels of neurons with single-cell resolution through fluorescence measurements, meaning that every single neuron can be recorded simultaneously [16], [17]. Finally, it is also possible to record the same neurons over several days, which is crucial for the study of a process with a strong temporal component like synaptic plasticity. However, some limitations must be considered as well. One limitation is that the calcium sensors used in two-photon calcium imaging are known to be slow sensors [18] with a low sampling rate around 50 Hz, as opposed to multielectrode arrays. In order to see spikes in real time, one would require a sampling rate of 10 kHz [18].

The main problem that we were trying to address in this research is the following: how can one statistically analyze large neuronal populations? One potential solution is to use statistical modelling, which gives access to the full distribution, such that it becomes possible to compute quantities of interest, such as the relative entropies between two recordings from the same preparation. In the context of this Master's project, we mostly investigated the population tracking-model using the MATLAB programming language. The main motivation for using this technique is that other statistical models such as the pairwise maximum entropy model can only be fit to populations of about 10 neurons, and quickly becomes computationally too expensive [19].

A recent study from O'Donnell et al. (2016) [1] showed that the population-tracking model offered significant advantages over other models for the description of spiking neurons based on calcium imaging data. The main advantages of the population-tracking model over other other models were as follows: first, the model could fit large numbers of neurons approximating around 1000 neurons; second, the model used N^2 parameters (where N is the number of neurons) which were computationally cheap to fit for large N's; third, parameter estimates converged within a reasonable number of time steps; fourth, the model could yield direct estimates of pattern probabilities; fifth, the model was the only low-dimensional model that could fit the whole pattern probability distribution using a computationally tractable approximation method (see Chapter 3 for more details on this method) [1].

One of the goals of this dissertation was to implement and test the population-tracking model, both on synthetic data and real experimental data. The first part

of the Master's project focuses on baseline tests performed on data generated by a Dichotomized Gaussian (DG) model [20]. Tests were carried out across synthetic datasets and across time steps. The second part of the Master's project attempted to model already existing rat hippocampal multielectrode array data which was modelled in past research [2]. This stage of the project represents an experimental control, since we already have information about that data. Finally, in the last part of the project, we modelled novel calcium imaging spikes data from a new neurobiological experiment. Although calcium imaging data seems to have become a new standard method for the study of large populations of neurons, it is nevertheless hard to preprocess. Pre-processing usually involves denoising, deconvolution, demixing and spike train estimation using physiological models [21], [22].

A major theme in this dissertation is long-term plasticity, and its role in intellectual disorders such as Autistic Spectrum Disorder or/and the Fragile X Syndrome (FXS). The Fragile X Syndrome is an example of an intellectual disability nested in a range of mental disorders referred to as "Autism Spectrum Disorder" (ASD). FXS is particularly interesting from a Neuroscience perspective because it leads to issues at the level of the synapses, as well as impairments in synaptic and homeostatic plasticity [23], [24]. It is caused by mutations in the Fmr1 gene. The Fmr1 gene is responsible for the synthesis of the FMRP protein, which targets important mRNAs and inhibits protein synthesis. Its absence leads to excess long-term depression and a lack of regulation of key biological compounds [25]. Mutations in the Fmr1 gene usually originate from CGG repeat expansions [26]. Because FXS involves neuronal networks and homeostatic plasticity dysregulation, it is still difficult to fully understand and describe how the genetic mutation leads to disruptions in synaptic plasticity [24]. Originally, the goal was to investigate Fmr1 mutations responsible for FXS phenotypes. However, due to logistic constraints, we were not able to use experimental data related to the Fmr1 gene and had to use SynGap data instead. The SynGap data was extracted from a study from the Integrative Physiology & Neuroscience laboratory, which investigated a heterozygous genotype (gene knock-out) in another gene: the SynGap gene.

The main purpose of the neurobiological experiment was to analyze potential changes in the neuronal circuitry of wild-type mice and heterozygous SynGap knock-out (KO) mice. A second goal was to compare neuronal activity before and after Monocular Deprivation (MD). The experiment was exploratory in essence, which means that we did not have any formal hypotheses that we attempted to validate using statistics. Calcium imaging data was recorded in the visual cortex (V1) of both types of mice. The experiment itself was inspired from previous research [27] which sought to understand how MD could modify ocular dominance and changes in neuronal networks' firing activity. The idea of the monocular deprivation experiment was inspired from Wiesel & Hubel's (1963) experiments on cats [3] in which they measured single-unit recordings from the striate cortex of cats. In the context of their experiment, one of the cats' eye was blind-folded from birth for different periods of time. The main result from the experiment was that there was a shift in ocular dominance, *only* in the cats with visual deprivation. Importantly, this ocular dominance shift effect was stronger in the cats that were visually deprived for longer periods of time since birth. Although we did not have any specific hypothe-

ses in our experiment, we still expected heterozygous genotypes to prevent natural neuronal remodelling mechanisms from happening spontaneously as they would in wild-type animals, which would support past research on long-term synaptic plasticity [3].

Last but not least, an important paper from Barnes et al. (2015) [28] showed that SynGap haploinsufficiency caused neurobiological modifications related to long-term synaptic plasticity processes which were also involved in the deletion of the Fmr1 gene and lead to the onset of Autism Spectrum Disorder (ASD) symptoms. The next section provides a thorough background on long-term synaptic plasticity, homeostatic plasticity dysregulation, and Fmr1 mutations, which, despite not being the object of study of our calcium imaging experiment, are still related to SynGap mutations to a great extent.

Chapter 2

Neurobiological background on Autistic Spectrum Disorder and the Fragile X Syndrome

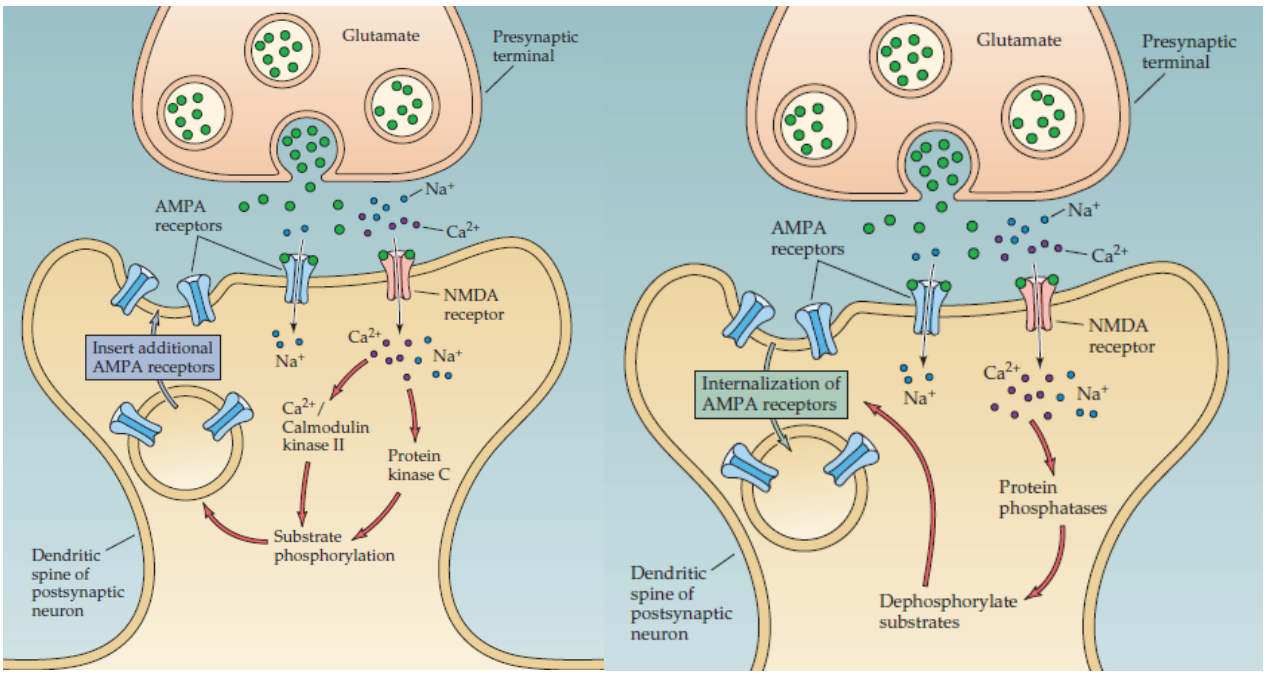
[1] The purpose of this Chapter is to introduce some key neurobiological aspects of long-term synaptic plasticity, Austistic Spectrum Disorder (ASD) and the Fragile X Syndrome (FXS). Autistic Spectrum Disorder (ASD) mostly consists of intellectual disorders which lie at the intersection between genetic mutation and changes in neuronal network dynamics [24]. Its study requires both a thorough understanding of the neurobiological mechanisms at play, and a good computational description of neuronal networks dynamics which can be achieved through mathematical modelling. In this Chapter, we will mostly focus on the neurobiological background causing the expression of the Fragile X Syndrome (FXS), in the hopes of better understanding how mathematical modelling can tell us more about synaptic plasticity disruptions in ASD.

2.1 Overview of long-term synaptic plasticity and homeostatic plasticity

One of the most important properties of the brain is its ability to shape itself over time through a complex interaction between gene expression and synaptic reinforcement through experience. Synapses, which connect neurons to one another, play a major role in this dynamic process. One striking feature of synapses is their remarkable flexibility and ability to weaken and/or strengthen over time. Two types of synaptic plasticity have been well-established over time: short-term plasticity and long-term plasticity [29], [30].

During neural development and learning, critical neurobiological events cause physiological changes at the synapses, which in turn permanently alter brain function. Two instances of long-term synaptic plasticity mechanisms are Long-term Potentiation (LTP) and Long-term Depression (LTD) [31], [32]. Both of these processes consist of complex neurobiological chain reactions. While LTP leads to synaptic

[1] This Chapter is mostly extracted from my *Informatics Research Proposal* [4]



(a) long-term potentiation (LTP) (b) long-term depression (LTD)

Figure 2.1: Long-term synaptic plasticity neurobiological mechanisms in a glutamate synapse. Figures are extracted from [32]

strengthening and activity-dependent increase in the efficacy of the synaptic transmission of electrochemical messages (as known as action potentials), LTD leads to synaptic weakening and activity-dependent decrease in the efficacy of the synaptic transmission of action potentials [32].

In the LTP neurobiological chain reaction, the glutamate neurotransmitter is first released through exocytosis into the synaptic cleft of a glutamate synapse. If the postsynaptic cell is depolarized enough, the NMDA receptor opens after glutamate binding and calcium ions enter the postsynaptic terminal. Through the activation of calcium Calmodulin Kinase II (CaMKII) and protein kinase C (PKC), new AMPA receptors are inserted into the postsynaptic spine and the synapse becomes more sensitive to glutamate neurotransmitters. This sensitization of the glutamate synapse is what leads to the more effective transmission of action potentials [32] (see Figure 2.1a).

Conversely, in the LTD neurobiological chain reaction, when glutamate is bound to the NMDA receptor and the postsynaptic terminal is depolarized, the rise of Ca^{2+} activates protein phosphatases, which results in both the internalization of postsynaptic AMPA receptors and the decrease of the glutamate synapse's sensitivity to glutamate neurotransmitters (see Figure 2.1b). It is this desensitization of the glutamate synapse that leads to the less effective transmission of action potentials. Importantly, it is the nature of the Ca^{2+} signal in the postsynaptic cell which determines whether LTP or LTD is triggered. Slow and small rises in Ca^{2+} lead to LTD whereas large and fast rises in Ca^{2+} lead to LTP [32].

One fundamental aspect of synaptic plasticity and Hebbian learning is what has been called neuronal homeostasis [34]. The main idea is that neurons modulate

levels of excitation and inhibition through regulatory mechanisms. A key concept in neuronal homeostasis is synaptic scaling: the measurement of a large number of neuronal synapses' strength through miniature Excitatory Post-Synaptic Currents (or "mEPSC") related to the spontaneous release of presynaptic vesicles. The increase of average mEPSC amplitude is called mEPSC upscaling while the decrease of average mEPSC amplitude is called mEPSC downscaling [34], [35].

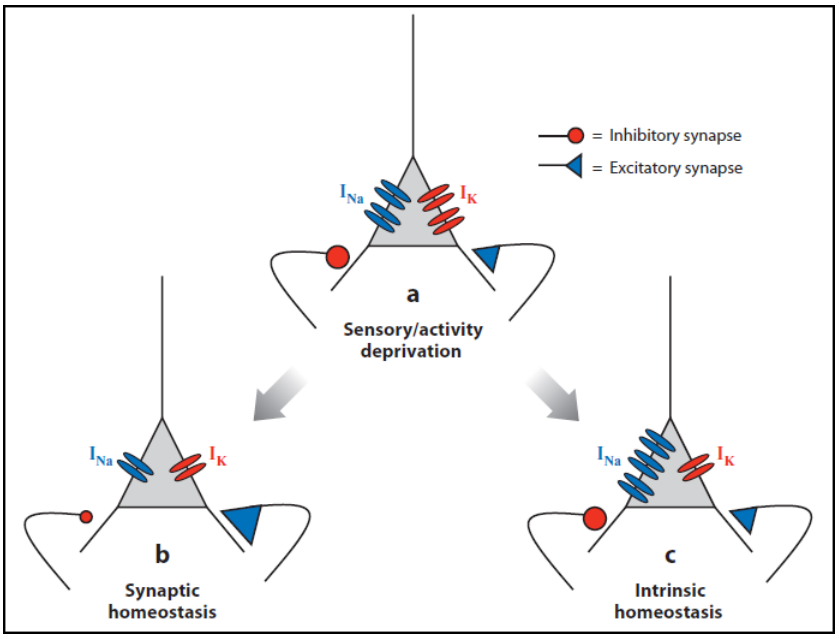


Figure 2.2: Illustration of neuronal homeostatic mechanisms. To the left, the synapse undergoes synaptic homeostasis (a to b). To the right, the synapse undergoes intrinsic homeostasis (a to c). Figure extracted from [33].

Two possible regulatory mechanisms which are believed to modulate the level of excitation of a neuron are synaptic homeostasis [36] and intrinsic homeostasis [37], [38]. On the one hand, in synaptic homeostasis, inhibitory synapses from feedback neurons are inhibited and excitatory synapses from feedback neurons are strengthened in order to scale up the excitation level of the presynaptic terminal (see Figure 2.2 $a \rightarrow b$). On the other hand, intrinsic homeostasis refers to changes in the neuron's intrinsic excitability. In other terms, the probability of the neuron to spike is modified by the re-balancing of ion channel densities [33]. On Figure 2.2 ($a \rightarrow c$), intrinsic homeostatic mechanisms cause the rebalancing of inward and outward voltage-dependent Na^+ and K^+ currents. Although these mechanisms may differ, their purpose remains the same: that is, to maintain a balance between excitation and inhibition, and ensure that neuronal circuits function normally [33].

2.2 The Fragile X Syndrome: From a genetic mutation to homeostatic dysregulation

Autistic Spectrum Disorders (ASD) are a range of intellectual disorders which cause issues at the level of the synapses [39], [40]. In particular, the Fragile X Syn-

drome is a genetic pattern which has been shown to lead to "autism" [24]. The Fragile X Syndrome (FXS) has been linked to phenotypic differences (a long, narrow face, prominent ears and flat feet), intellectual problems (mental retardation and low IQ) as well as social and communication problems (social anxiety, avoidance, language delays) [23].

FXS appears to be the result of a genetic mutation in the protein-coding Fragile X Mental Retardation 1 (Fmr1) gene on the X chromosome (cytogenetic position Xq27.3) with CGG repeat disorder [26], [41], [25]. FXS mutations most often involve an expansion of the CGG repeat located in the 5' untranslated region (UTR) of the Fmr1 gene. CGG repeats can be of different lengths. Mutations ranging from 55 to 200 CGG repeats fall into what has been called "Premutation alleles". These do not directly lead to the expression of the Fragile X Syndrome. Nevertheless, these mutations are more prone to further CGG repeat expansions than disorders involving less than 55 CGG repeats, and they also involve a higher risk of fragile X-related primary ovarian insufficiency (FXPOI) and fragile X-associated tremor/ataxia syndrome (FXTAS). When the mutation involves more than 200 CGG repeats, it is referred to as "full mutation", and directly leads to DNA methylation, chromatin condensation, histone hypoacetylation and transcriptional silencing of the Fmr1 gene [25].

One interesting property of the Fmr1 gene is that it is highly preserved across species [25]. This means that one can study an orthologue version of the human Fmr1 gene in an animal model, and expect with reasonable confidence to find the same mechanisms of Fmr1 gene expression as in the human. The most common animal models of the Fmr1 gene include the mouse and the fruit fly [25]. In the mouse, the protein-coding Fmr1 orthologue has 97% homology to the human protein-coding Fmr1 gene. Mutations in the mouse Fmr1 gene lead to phenotypes that are similar to the ones witnessed in the homo sapiens (i.e: disrupted learning and memory, higher chance of seizure onset, presence of dense and immature dendritic spines) [25]. In the fruit fly, mutations in the Fmr1 gene lead to abnormal neuronal architecture, impairments in long-term memory and issues in synaptic function and synaptic plasticity.

The Fmr1 gene is responsible for the synthesis of a protein called Fragile X Mental Retardation Protein (or FMRP). FMRP is a selective RNA-binding protein primarily expressed in neurons which plays a major role in synapses [42]. Indeed, FMRP is usually localized on postsynaptic dendritic spines and leads to translation inhibition of dendritic mRNA. It is involved in dendritic development and function, and allows the onset of key synaptic plasticity mechanisms. [25]. In the case of a mutation in the Fmr1 gene, Fmr1 transcription is silenced, which in turn prevents the translation of the gene into FMRP synthesis [43].

Because FMRP is involved in the inhibition of protein synthesis in FXS, the absence of FMRP leads to excessive protein synthesis and triggers excess AMPA receptor internalization (long-term depression or LTD) [43]. To this day, it is still unclear how FMRP precisely regulates protein synthesis in the brain at the level of neuronal networks. The neurobiological pathway that is believed to play a key role in protein synthesis regulation is the mGluR pathway (metabotropic Glutamate receptor), which activates a cascade of second messengers during LTD [23], [43]. Two such second messengers which have been hypothesized to contribute to FXS are

mTOR (mammalian target of rapamycin) and ERK (extracellular signal-related kinase). Last but not least, other biological factors ought to be considered when investigating the effect of FMRP on neuronal networks. These include phosphorylation levels of key biological components such as FMRP and ERK, as well as alternative neurobiological pathways such as the RNA interference (RNAi) pathway [25].

One important feature of intellectual disorders is that they result in disruptions at the level of networks of neurons rather than isolated genes or individual neurons. In the case of FXS, such disruptions involve impaired neuroplasticity and abnormalities in the development and function of synapses and neuronal circuits through the combination of genetic, epigenetic and environmental factors [24]. Consequently, in order to reach further insights with respect to the precise neurobiological mechanisms involved in intellectual disabilities such as the Fragile X Syndrome, one needs to think about the dynamics of neuronal networks and synaptic feedback regulation (or homeostasic plasticity), and use appropriate computational neuroscience techniques in order to capture more of the complexity of such dynamics [24].

One instance of a critical issue occurring at the level of neuronal networks is the imbalance between excitation and inhibition. In the review from Nelson & Valakh (2015) [44], the authors integrate results from a wide range of studies investigating the potential consequences of ASD's genetic mutations at the level of the synapses. Although many studies claim that ASD may be linked to an *increase* in the ratio between excitation and inhibition (hyper-excitability), as suggests results from electroencephalography (EEG) studies revealing epileptic seizures in patients with autism, other studies have cast doubt on this conclusion and have argued instead that these disorders were the result of a *decrease* in the ratio between excitation and inhibition. Results from current research remain inconclusive with respect to the precise nature of homeostatic plasticity dysregulation mechanisms in autistic spectrum disorders [44].

2.3 Experimental investigation of the Fragile X Syndrome

In their study, Osterweil et al. (2013) [45] disrupted the Ras-ERK pathway by introducing a molecule called Lovastatin, along with farnesyl thiosalicylic acid (FTS). The injection of Lovastatin is believed to inhibit the phosphorylation of Ras-GDP, which makes the ERK1/2 levelsgo down. As a result, protein synthesis can be reduced and regulated at the synapses. The authors injected different doses of lovastatin into the hippocampus and the visual cortex of both wild type and knockout mice in order to assess potential effects of lovastatin on protein synthesis levels. The researchers reported statistics on potential seizures and measured the overall excitability of the synapses, the duration of action potentials, and the levels of protein synthesis after injection. After experimental investigation, results suggested that lovastatin could indeed inhibit Ras-ERK1/2 signaling in hippocampal neurons and decrease both epileptogenesis and hyperexcitability. However, more research is needed to determine whether the beneficial effects are really due to lovastatin decreasing the Ras-ERK1/2 signalling pathway [45]. It may be the case that other proteins or modulators also play a role in this specific pathway.

One key element in the Osterweil et al. (2013) study [45] is the use of a mouse

model for the investigation of the Fmr1 gene and its effects. The use of a mouse model has indeed been supported by past research, which suggests that mice represent an adequate animal model for the study of FXS. Specifically, Huber et al. (2002) claimed that "the Fmr1 null mutant (knockout) (Fmr1-KO) mouse [...] has a behavioural phenotype consistent with fragile X syndrome" [45].

In another study from Gonçalves, Anstey, Golshani and Portera-Cailliau (2013) [46], researchers used a mouse model to understand how disruptions in neuronal networks could lead to mental disorders such as FXS. In order to do so, they used whole-cell recordings, in-vivo patch clamp recordings as well as a combination of two-photon calcium imaging and electroencephalography (EEG) to assess levels of neural activity in different behavioural states in both wild-type mice and Fmr1-Knockout mice.

Results from Gonçalves et al. (2013) [46] lead to four conclusions. First, the Fmr1 mutation may indeed cause hyper-excitability of networks of cortical neurons in a state-dependent fashion. Second, neural activity may be altered by top-down modulation effects influenced by behavioural states (such as deep sleep or wakefulness). Third, too much synchronization and unusual/very high levels of neural activity may lead to impairments and miswirings in the neural circuits, which may cause mental disorders such as FXS. This could be explained by the fact that spontaneous neural activity during brain development in critical periods is crucial for the good formation of neuronal networks. Indeed, in healthy animals, desynchronization in neural activity has been shown to give rise to sparse coding, which is computationally efficient, reinforces memory storage capacity, simplifies the processing of complex information and saves energy [47]. Finally, neuronal networks' hyper-excitability phenomenon in mutant mice was hypothesized to be caused by too much inhibition of the GABA A receptors (responsible for inhibition of neural activity), and/or by the fact that dendritic spines were already abnormally unstable during the second postnatal week [46].

Chapter 3

Neurobiological experiments and modelling methodology

3.1 Brief overview of the project's methodology

This Master's project was divided into three parts: synthetic data modelling analysis, multielectrode array data modelling analysis and calcium imaging data modelling analysis. We mainly attempted to answer the three following questions: 1) How well can our choice of mathematical model capture the data that we were given? 2) Did the connectivity change between pairs of neuronal networks? 3) What can our choice of model say about real spikes data collected in the context of neurobiological experiments?

With respect to the first question, we used a selection of four different synthetic datasets in order to test the reliability and accuracy of our model: the population-tracking model [1]. The synthetic data analysis was divided into two main stages: baseline tests and timesteps analysis. In the first stage, we described and compared the four synthetic datasets of interest, and investigated how the population-tracking model could fit its parameters to the synthetic data. The methodology that we used to generate spikes made use of the Dichotomized Gaussian (DG) model inspired from past machine learning literature [20]. The main idea is that we turned a N-dimensional gaussian random variable into a multivariate binary distribution (which is the spikes data that we want to model). Specific details regarding the synthetically-generated datasets can be found in Chapter 4.

With regards to the second question, one usually investigates the parameters of a given model, which often involve the mean firing rates and correlation properties of spikes datasets. The idea is to understand how those change from one spikes matrix to the other. Baseline pairwise comparison tests were performed across all synthetic datasets, and over time (timestep analysis). In addition, spikes data extracted from real neurobiological experiments were also analyzed and compared. The two experiments that were our object of study were the multielectrode array experiment and the synaptic plasticity experiment. All spikes data from all stages followed the same format: namely, they were all binarized matrices (only 0's and 1's) with T rows and N columns, where T referred to the different time steps, and N referred to the different neurons in the network.

Finally, the third question is the main question that we are attempting to answer in this dissertation. The population-tracking model [1] is a fairly novel statistical model (published in 2016) which could be quite useful in our attempt to make sense of binarized spikes data extracted from calcium imaging signals. We will mostly attempt to address this question in Chapter 5.

3.2 Description of the neurobiological experiments

The two main experiments that were investigated in this project were the multielectrode array experiment and the synaptic plasticity experiment. These experiments enabled us to work with and model real spikes data.

3.2.1 The multielectrode array (MEA) experiment

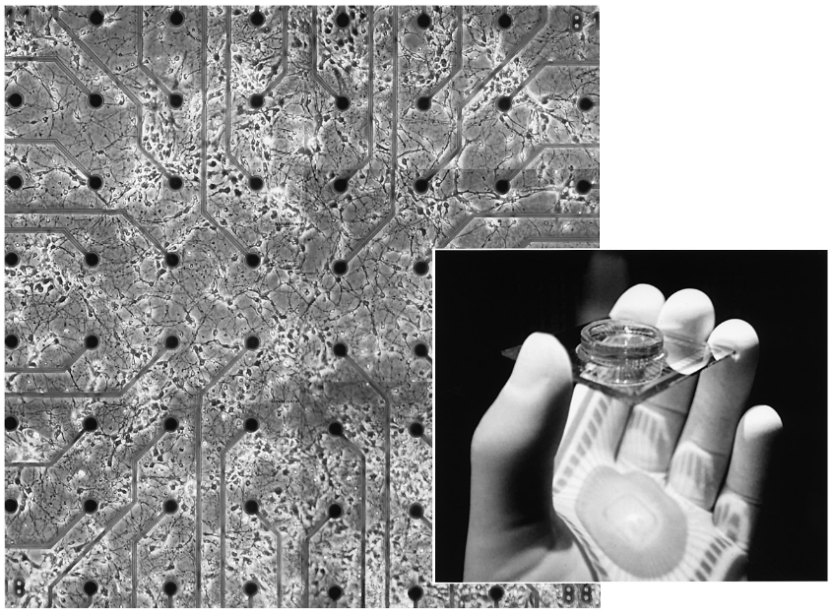


Figure 3.1: Illustration of the multielectrode array (MEA) device (this one includes 60 electrodes). Figure extracted from [48].

One of the main advantages of the multielectrode array technique is that it allows the simultaneous recording of thousands of channels. As mentioned in Potter (2001) [48], MEAs can be used for electrophysiological recordings on neural dissociated cell cultures. Figure 3.1 displays an illustration of the multielectrode array device. With regards to the MEA experiment, researchers were specifically interested in the electrophysiological recordings of dissociated hippocampal neurons from brain tissue of embryonic rats. Neural hippocampal cultures were used as the basis of the multielectrode array recordings (4,096 channel array), which measured the activity of 1,024 electrodes at a sampling rate of 7 kHz/electrode with a resolution of 12 bit per electrode. In the case of our project, it was necessary to ensure that

intercellular connections between neurons were preserved, as we were mostly interested in investigating the level of neuronal networks. The MEA experiment investigated spikes from the neural hippocampal culture over the course of a few minutes at different points in time. The collected data was presented in Panas et al. (2015) [2].

Once the electrophysiological recordings were completed, the data needed to be pre-processed in order to be turned into spike trains. Spike detection followed the methods from Muthmann et al. (2015) [49]. Weighted interpolated signals were generated in order to capture the spikes which could be recorded close to or between electrodes. All signals were further processed in order to minimize noise bias, and a support vector machine model was trained and used to classify events as true spikes or noise.

Once all spikes were detected, they needed to be clustered into chunks which represented our "neurons" of interest. Clustering was implemented using the mean shift algorithm as well as methods from the scikit-learn Python machine-learning library. In this project, we made use of the "herding spikes" methods [50] in order to carry out the spikes clustering process. We then computed the Fano factor for the spike train in each cluster. Noisy units were removed by excluding clusters with small Fano factors, and we also ignored clusters with very few spikes. The data was then binarized and binned using 10 ms time bins. In this project, we used pre-processed and clustered spikes data recorded at the following times: basal (baseline), 2 hours (H2) and 20 hours (H20).

3.2.2 The synaptic plasticity experiment

The choice of design for the synaptic plasticity experiment followed previous research from Wiesel & Hubel (1963) [3] and Rose et al. (2016) [27], which sought to better understand how monocular deprivation could lead to a change of neuronal responses, as well as modifications in functional and structural synaptic plasticity. In the context of this project, we wanted to investigate the neural activity from populations of V1 neurons (visual cortex) in two phases: the pre-monocular deprivation phase and the post-monocular deprivation phase. All subjects were adult mice. The first phase represented our baseline condition in which no monocular deprivation had occurred, and the second phase represented our experimental condition in which monocular deprivation had already happened.

Figure 3.2 A. shows an illustration of the synaptic plasticity experimental paradigm. The mouse was moving freely on a wheel while a two-photon calcium imaging device was recording a calcium signal. In the meantime, some stimuli were presented through the use of a Virtual Reality screen. Stimuli were bars tilted in four different directions: 0°, 45°, 90° and 135°. The head of the mouse was fixed in order to prevent any head movements. Figure 3.2 B shows a z-projection of the imaging stack in the visual cortex (V1) which could be visualized using a laser microscopy device. For each neuron, the fluorescence level of single neurons was recorded. A resonant scanner recorded signals with a rate of 12 kHz, and image acquisition had a rate of 40 Hz for a resolution of 600x600 pixels. At the bottom, we see the calcium signal $\Delta F(t)/F_0$ for two different neurons. Spikes in the calcium signal indicate the pres-

ence of a firing event.

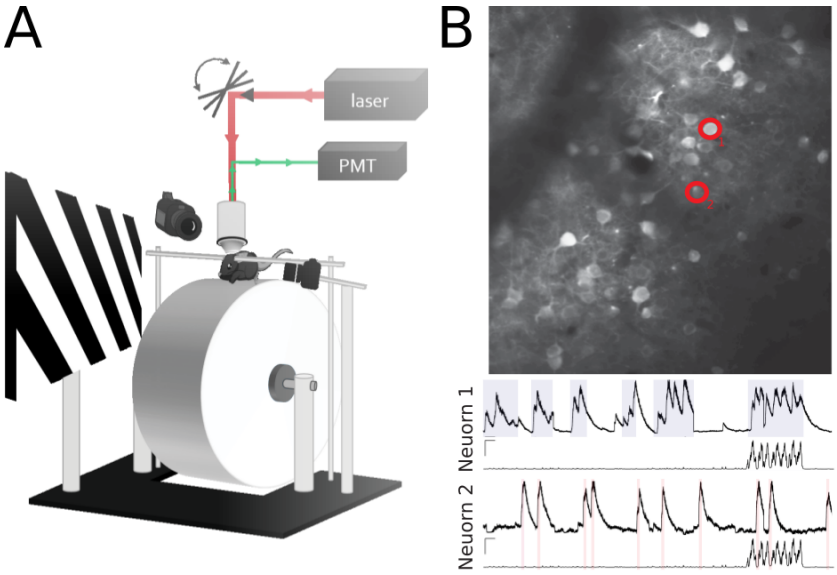


Figure 3.2: A) Illustration of the synaptic plasticity experiment. B) Recording of the two-photon calcium imaging signal and spikes estimation. *Unpublished Figure.*[1]

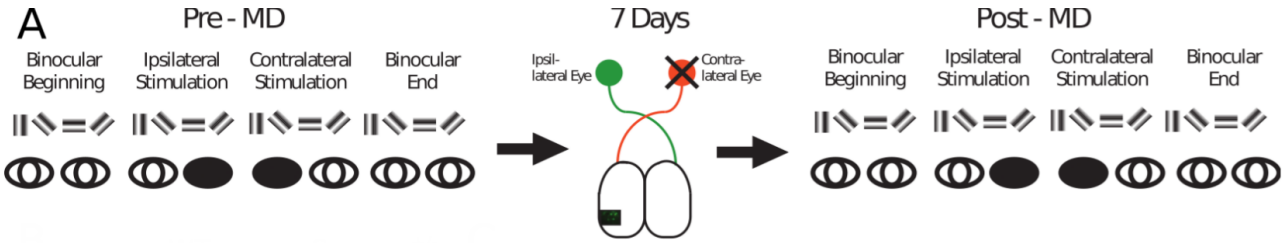


Figure 3.3: Overview of the conditions at different stages in the synaptic plasticity experiment. *Unpublished Figure.*[1]

One important question is the following: How does one turn the fluorescence calcium signal into a binary spikes train (with 0's and 1's)? This question was the object of study of Deneux et al. (2016) [22], in which researchers developed a maximum-likelihood method called "MLspike". The "MLspike" method was implemented as a pre-processing step in order to get the binary matrices that we used in this project. The goal of the method is mainly to find the most likely spike train underlying the recorded fluorescence signal. In order to do so, "MLspike" uses a physiological model of intracellular Calcium dynamics and baseline fluorescence (F_0) along with a filtering technique which runs in linear time. The maximum likelihood method used a version of the Viterbi algorithm in order to estimate the optimal spike train by maximizing the maximum a posteriori (MAP) distribution probability [22]. Using this approach, it is possible to get a reliable estimate of the firing of single neurons over time. Note that one key design choice (and potentially caveat) is the choice of

[1]This figure was provided by Dr. Rochefort from the Integrative Physiology department at the University of Edinburgh.

time bins in the extraction of spike trains. How often do we want to record binary spikes for each neuron?

Finally, Figure 3.3 shows the different conditions and the order in which the conditions followed one another. The two main genotypes that were being compared are the Wild-Type genotype and the Knock-Out genotype. Mice from the Knock-Out genotype group had one copy of the SynGap gene knocked out in the whole animal (heterozygous knockout). Measurements were recorded in the "Stimulus" condition in which the gratings were presented to the mice, and in the "Darkness" condition in which mice were immersed in darkness. The monocular deprivation experiment worked as follows. In each of the two main stages (Pre- and Post-monocular deprivation stages), measurements were recorded four times: 1) with both eyes open (Binocular pre), 2) with only the left eye open, 3) with only the right eye open, 4) with both eyes open again (Binocular post). Between the two stages, a blindfold was put on the right eye during seven days, so that the mouse could only see with the left eye.

3.3 Overview of the mathematical modelling methods

3.3.1 The population-tracking model

One of the main problems of the pairwise maximum-entropy model is that it cannot be used for *large* populations of neurons (populations above about 25 neurons) [19]. With the new technological developments in the field of Neuroscience, it is now possible to have access to large-scale recordings, such as multielectrode arrays, which can record the electrical activity from up to thousands of channels simultaneously. The implication of such breakthrough is that we now need to develop new mathematical models which can capture more of the dynamics of the neuronal networks.

According to O'Donnell et al. (2016) [1], the ideal statistical model for neural populations balances accuracy, tractability as well as usefulness. In this section, we introduce the innovative mathematical model that [1] put forward. An overview of the model is introduced on Figure 3.4. The idea is as follows: the data that is recorded in TxN matrices is assumed to behave according to an underlying probability distribution P_{true}. The goal of the mathematical model is to identify the probability distribution P_{true}. However, it is not possible to use a histogramming method with smoothing as one would usually do because the number of neurons grows exponentially. Indeed, one would expect a total number of 2^N firing patterns, as each of the N neurons can either be in the ON or OFF state. The idea of the statistical model is to estimate the parameters of a tractable model in order to get a good estimate of the probability distribution P_{model}. The model that is introduced in [1] is named the "population-tracking model" (see Figure 3.4) precisely because it is designed to be fit to data with large numbers of neurons.

The population-tracking model namely consists of 3 main parameters that are estimated from the firing matrix samples:

- ***p(k)***, a vector of N+1 entries which indicates the probability of 0 to N active

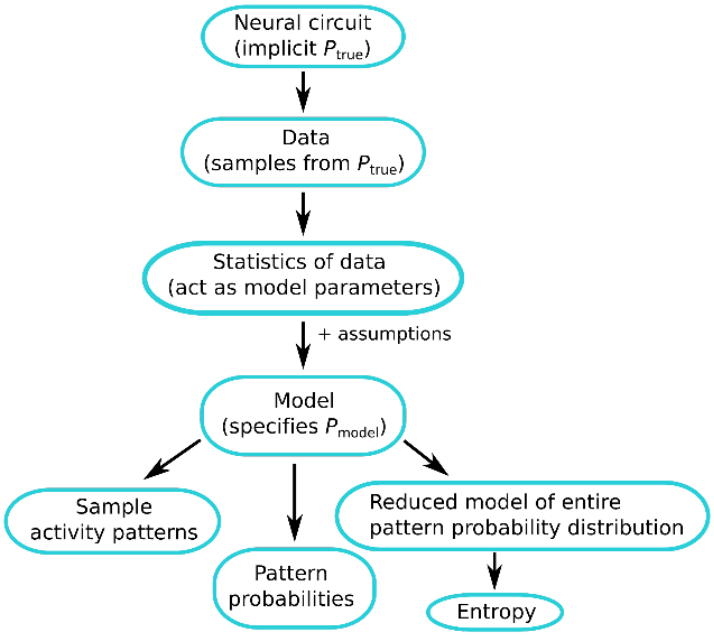


Figure 3.4: Overview of the population-tracking model. Figure extracted from [1]

neurons being active simultaneously at a timestep T. This can also be referred to as the population rate, or "high-level component" of the model.

- $\boldsymbol{p(x_i)}$, a vector of N entries which indicates the mean probability of independent firing of each of the N neurons in the neuronal network.

- $\boldsymbol{p(x_i|k)}$, a matrix of N x (N+1) entries which indicates the conditional probabilities of single neurons being active given that there are k active neurons in the neuronal network. This is also referred to as the "low-level component" of the model in [1].

We now wish to explain the implementation procedure for the parameter estimation of each of the three above probabilities. First, $\boldsymbol{p(x_i)}$ is simply obtained by taking the mean firing rate of each neuron (each column of the TxN binary spikes matrix). Second, $\boldsymbol{p(k)}$ is obtained by counting the number of total active neurons for each time step and by histogramming these counts over all values of k's. Importantly, a Dirichlet prior (conjugate to the multinomial distribution) was used as a smoothing method (in this case, alpha-smoothing) in order to account for cases in which some values of k's were never observed. The goal is to ensure that even those unobserved k's are assigned some probability rather than $p(k) = 0$. In mathematical terms, we can write:

$$\hat{p}(k,\alpha) = \frac{c_k + \alpha}{T + N.\alpha} \tag{3.1}$$

where c_k refers to the total count of words with k active neurons, T refers to the number of timesteps (rows) in the binary spikes matrix, N refers to the number of

neurons (columns) in the binary spikes matrix and α is the smoothing parameter (pseudocount $\alpha > 0$).

Finally, estimation of the $\boldsymbol{p(x_i|k)}$ parameter was performed by cycling through every value of k, finding the subset of words whose sum of active neurons matched k and counting the number of times each individual neuron was active at those timesteps. Assuming that we are referring to the words subset corresponding to a given value of k as $d_{i,k}$, then the conditional probability of independent neurons being active given k active neurons in the network corresponds to the Maximum Likelihood estimate $\hat{p}(x_i|k) = d_{i,k}/T_k$. However, since we want to prevent the case in which some values of k might be assigned a 0 probability for not having been observed in the spikes matrix, we add a regularizer in the estimation of the probability. Since x_i can be considered as a Bernoulli variable (it is either active or inactive), standard Bayesian modelling regularization techniques state that it is best to set a Beta prior distribution over each $p(x_i|K)$ (conjugate to the binomial distribution). Therefore, the posterior probability can now be written as:

$$\hat{p}(x_i|k,\beta_0,\beta_1) = \frac{d_{i,k}+\beta_1}{\beta_0+\beta_1+T_k} \tag{3.2}$$

where $d_{i,k}$ refers to the subset of words corresponding to a neuron x_i and to 'k' active neurons, T_k refers to the total number of timesteps associated to a given value of k active neurons and β_0 and β_1 refer to two beta hyperparameters. β_2 constrains the prior's mean to be equal to k/N and β_1 describes the variance of the prior by giving an indication of how much the conditional probability reflects the data (a wider variance indicates a more naive estimation). These conditions yielded a mean $\mu = k/N$ and a variance $\sigma^2 = 0.5\mu(1-\mu)$ such that:

$$\beta_1 = \frac{\mu}{\sigma^2}(\mu - \mu^2 - \sigma^2) \tag{3.3}$$

$$\beta_2 = \beta_1.\left[\frac{1}{\mu} - 1\right] \tag{3.4}$$

Note that the computational implementation of all parameter estimation methods were available on the GitHub directory which was uploaded by Cian O'Donnell (first author of [1]). A visualization of the estimation of those parameters is available in Chapter 4 (See section 4.1 for an example involving synthetic data and 4.2 for an example involving multielectrode array data).

Once all probability parameters have been successfully estimated, the model is complete. This means that we have all the necessary ingredients to compute the probability of specific words of interest (sometimes referred to as "patterns" as well). O'Donnell et al. (2016) [1] defines the probability of a pattern as:

$$p(\{x\}) = \frac{p(k)}{a_k}\left[\prod_{i=1}^{N} p(x_i|k)^{x_i}[1-p(x_i|k)]^{1-x_i}\right] \tag{3.5}$$

where $k = \sum_{i=1}^{N} x_i$ (which is effectively the mathematical definition of "k active neurons"), x_i defines a given neuron being either active ($x_i = 1$) or inactive ($x_i = 0$), N

refers to the number of neurons in the neuronal network and a_k refers to a normalization constant. The normalization constant a_k refers to the sum of probabilities of all $\binom{N}{k}$ patterns in a set $S(k)$ with k active neurons such that:

$$a_k = \sum_{\{x\} \in S(k)} \left[\prod_{i=1}^{N} p(x_i|k)^{x_i} [1 - p(x_i|k)]^{1-x_i} \right] \tag{3.6}$$

which is effectively the sum of the probabilities of *all* patterns matching a specific value of k active neurons (part of the set $S(k)$). Therefore, in the concrete implementation of the model, the computation of the normalization constant a_k actually requires the computation of a whole vector of constants $\vec{\boldsymbol{a}}$ which correspond to all possible values of k active neurons in the network. Importantly, one needs to compute the normalization constant prior to computing the probability of any given pattern of interest.

Normalization constant estimation methods include brute force enumeration (which means literally summing over all possible probabilities with k active neurons), drawing Bernoulli samples and estimating the constant based on these samples, using importance sampling, and using the sum-of-log-normals method (see below for a description). For small numbers of neurons ($N < 20$), it is best to use the brute force estimation as it is the most accurate. For $20 < N < 50$, it is reasonable to use the importance sampling method, and larger populations of neurons usually require turning the patterns into a continuous random variable, as is described below in section 3.3.2. Finally, note that the probability of patterns $p(\{x\})$ can lead to the estimation of the full probability distribution over patterns of any given TxN spikes matrix and can also be used to compute other quantities such as the entropy of the spikes data (see section 3.3.2 for more information).

One last feature of the population-tracking model (as suggests Figure 3.4) is the ability of the model to sample new spikes datasets based on the three main probability estimates $p(k)$, $p(x_i)$ and $p(x_i|k)$. The main trick from O'Donnell's sampling method is to use a cumulative sum computational method in order to generate a value of k (k active neurons) at each time step using the "high-level component" $p(k)$, and then use the "low-level component" $p(x_i|k)$ in order to determine *which* specific neurons are active at that time step given that there are k neurons active simultaneously. An example of samples comparison in the case of multielectrode array modelling can be found in Chapter 4, section 4.2.

3.3.2 Entropy computation and the sum-of-log-normals method

The Shannon entropy of a probability distribution is usually defined as follows [51]:

$$H = -\sum_{i} p(i).log_2(p(i)) \tag{3.7}$$

The entropy is continuous and it is expressed in bits (due to the use of the 'log_2'). It can be related to the physical entropy, and one can think of it as the "richness" of a given distribution [51], [52]. In the specific case of firing spikes binary matrices, one can think of the entropy as having an inverse relation with the level of correlations

in the data. This means that a lower entropy will usually be linked to *higher* levels of correlations in the neuronal networks, whereas a higher entropy will usually be linked to *lower* levels of correlations [20].

One design choice when investigating the entropy is the choice of probability of interest. More specifically, in the context of our population-tracking model implementation, we have estimated parameters corresponding to the number of active neurons in the network ($p(k)$), number of independent active neurons ($p(x_i)$), conditional probability of firing ($p(x_i|k)$) and pattern probabilities in the network ($p(\{x\})$). All represent potential choices of parameters for the computation of the entropy. One may argue that the best choice is to compute the entropy using the pattern probabilities $p(\{x\})$ for all words in the neuronal network of N neurons, as this would lead to a *full* entropy estimation which takes into account all possible words that could ever be produced in the neuronal network, such that:

$$H = -\sum_{i=1}^{2^N} p(\{x\}) log_2 p(\{x\}) \tag{3.8}$$

Although theoretically reasonable, this method cannot always be implemented computationally because of size issues. As was mentioned earlier, the number of possible words grows exponentially in the network (2^N). This makes it difficult to compute all the pattern probabilities. For instance, for a network of only 100 neurons, one would need to compute 1,267,650,600,228,229,401,496,703,205,376 (order of magnitude of 10^{31}) different pattern probabilities, which would surely take a few weeks.

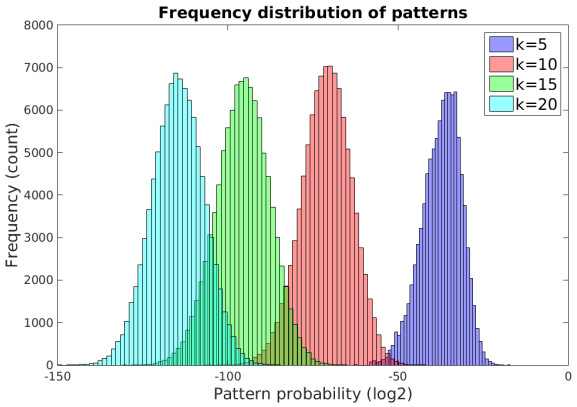
Fortunately, it is possible to compute a low-dimensional approximation of the pattern probability distribution, as mentions O'Donnell et al. (2016) [1]. Indeed, one can show that the distribution of pattern probabilities approximates the sum of log-normal distributions as k and N get larger. Such distribution makes use of the logarithm (in this case, we use the log_2 logarithm) of the pattern probability previously introduced in equation (3.5), that is:

$$log\, p(\{x\}) = log\, p(k) + \sum_{i}^{N} log[p(x_i|k)^{x_i}(1 - p(x_i|k))^{(1-x_i)}] - log\, a_k \tag{3.9}$$

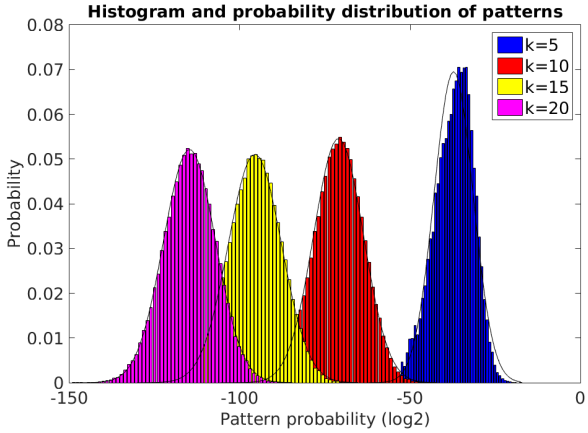
Note that x_i can either be a 1 (active) or a 0 (inactive), which means that the second term will be evaluated to $log\, p(x_i|k)$ for all active neurons, and to $log(1 - p(x_i|k))$ for all inactive neurons. Effectively, since we have a total of k active neurons in the neuronal network, we can rewrite equation (3.9) as the following:

$$log\, p(\{x\}) = log\, p(k) + \sum_{i}^{k} log\, p(x_i|k) + \sum_{j}^{N-k} log(1 - p(x_j|k)) - log\, a_k \tag{3.10}$$

Now, the main idea of the sum of log-normals approximation is to estimate some parameters which can fully account for the distribution of the two middle terms of equation (3.10) $log\, p(x_i|k)$ (active neurons with k active neurons) and $log(1 - p(x_i|k))$ (inactive neurons with k active neurons). According to the Central Limit Theorem, these probability distributions tend towards a normal distribution as N



(a) Frequency distribution of four k's



(b) Probability distributions with normal fit for all four k's

Figure 3.5: Distributions of pattern probabilities for four different k's (using MEA basal data)

gets larger. This implies that it is possible to estimate the mean and variance, which can then be used to determine the full distribution of $log\,p(\{x\})$ since we have already computed $p(k)$ and the normalization constant $\vec{a}$ when fitting the initial parameters of the model (see section 3.3.2).

The sum-of-log normals method operates as follows. First and foremost, we need to compute the initial parameter estimates as described in section 3.3.2. For 10^5 different randomly generated words (using the MATLAB function "randperm"), we first compute the log probability of each word and store it in a vector. Then, we make an initial "guess" regarding the mean and variance of the normal distributions of $log\,p(x_i|k)$ (active or 'ON') and $log(1-p(x_i|k))$ (inactive or 'OFF') using the k-means algorithms. Then, one can fit a mixture of Gaussians model using the initial "guess" of parameters, and loop through several rounds of parameter estimation using the Expectation-Maximization (EM) algorithm. After convergence of the EM algorithm, we obtain the final parameters (mean and variance) of both sum-of-log-normals distributions, which we can renormalize so that the distribution sums to 1. Finally, we can get an estimate of the probability distribution $p(\{x\})$ by taking 2, raised to the power of $log\,p(\{x\})$ (since our logarithm was of base 2). The MATLAB code with the implementation of the sum-of-log-normals method was made available by Cian O'Donnell.

On Figure 3.5, one can see the frequency distribution of all words on the left (see Figure 3.5a). Four choices of k values are investigated here: $k=5$, $k=10$, $k=15$ and $k=20$. Note that the X-axis corresponds to the logarithm of every given word's probability (equation 3.10). To the right, one can see the probability distributions of the pattern probabilities and the fitted normal distributions using the MATLAB "hist" function and the standard normal distribution mathematical formula (equation (3.11)) evaluated at each 'hist' x bin's center (see Figure 3.5a). Interestingly, for smaller values of k, we notice that the normal distribution has a narrower range than

bigger values of k.

$$x \sim \mathcal{N}(\mu, \sigma^2),\ p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \tag{3.11}$$

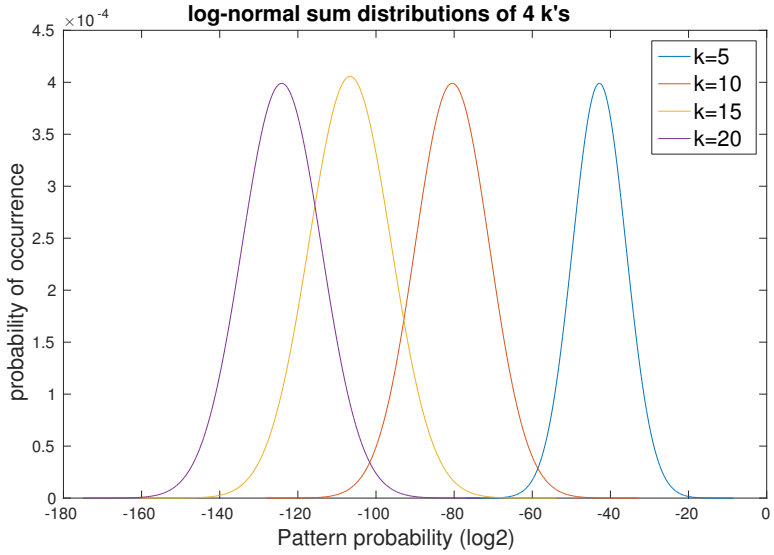


Figure 3.6: Model fits of the normal distributions of $p(\{x\})$ for the same four values of k.

Nevertheless, since the mean and variance of the normal distribution are solely computed based on the range of the x bins, these do not represent reliable estimates for our log-normal distributions. The method introduced in [1] attemps to fit two different normal distributions instead, as described above. Figure 3.6 shows the distribution of the pattern probabilities ($p(\{x\})$) after fitting the two normal distributions of ON neurons ($\mathcal{N}(\mu_{ON}(k), \sigma^2_{ON}(k))$) and OFF neurons ($\mathcal{N}(\mu_{ON}(k), \sigma^2_{ON}(k))$). For each value of k active neurons, the parameters of each distribution are given by:

$$\mu_{ON}(k) = k[logp(x|k)] \tag{3.12}$$

$$\sigma^2_{ON}(k) = k\left(\frac{N-k-1}{N-1}\right) var[logp(x|k)] \tag{3.13}$$

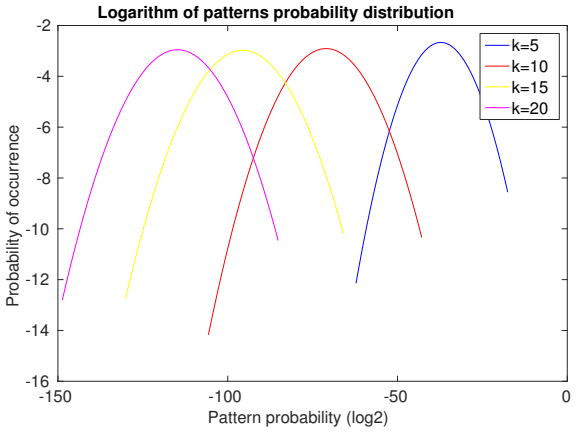
for the active (ON) neurons, and

$$\mu_{OFF}(k) = (N-k)[log(1-p(x|k))] \tag{3.14}$$

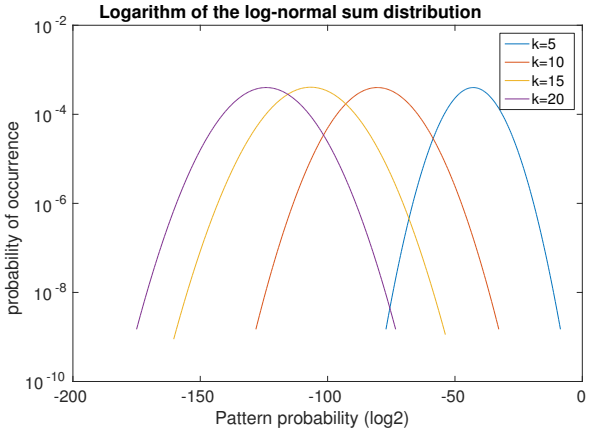
$$\sigma^2_{OFF} = (N-k)\left(\frac{k-1}{N-1}\right) var[log(1-p(x|k))] \tag{3.15}$$

for the inactive (OFF) neurons. Importantly, note that the variance terms make use of corrections due to drawing without replacement from a finite population.

Figure 3.7 displays a comparison between the logarithmic version of the frequency distribution from Figure 3.5b (see Figure 3.7a), against the model log-normal distributions (plotted on a logarithmic scale) after the Mixture-of-Gaussians fitting process (Figure 3.7b). As we can see, the Mixture-of-Gaussians process successfully approximates the probability distribution of pattern log probabilities. Do note,



(a) Logarithmic frequency distribution for 4 k's



(b) model log-normal distribution for 4 k's (log scale)

Figure 3.7: Logarithmic distributions (using MEA basal data)

however, the slight shift to the left on the X-axis (pattern probabilities) in the case of the model probabilities (Figure 3.7b). This shift is probably due to the renormalization process which turns the $log\,p(\{x\})$ parameters into a probability distribution that describes $p(\{x\})$. This shift tends to get bigger as the variance of the normal distribution increases (for greater values of k active neurons in the neuronal network).

As was mentioned previously, the computation of the sum-of-log normals is mostly useful to compute other quantities such as the population entropy of a neuronal network. O'Donnell et al. (2016) [1] decomposes the population entropy into two components, as follows:

$$H = H_k + H(p(\{x\}|k)) = H_k + H(\theta)_k \tag{3.16}$$

where $H_k = -\sum_{k=0}^{N} p(k).log_2 p(k)$ refers to the entropy of the population synchrony distribution and $H(\theta)_k = \sum_{k=0}^{N} p(k)H(\theta_k)$ is the conditional entropy of the pattern probability distribution with k active neurons. Using the probability distribution from Figure 3.6 and Figure 3.7b, we can turn the empirical probability of words given k active neurons $p(\{x\}|k)$ into the probability over patterns with k active neurons $p(\theta)_k$ where θ_k has now become a *continuous random variable*. The conditional entropy of patterns given k active neurons can thus be expressed using integration rather than summation techniques, as is shown below:

$$H(\theta_k) = \binom{N}{k} \int_0^1 p(\theta)_k . \Big[\theta_k log_2 \theta_k\Big] d\theta \tag{3.17}$$

Equation (3.16) combined with equation (3.17) represent the basis for the computation of the population entropy, which will be used in Chapter 4 in the case of the multielectrode array data and Calcium imaging data.

Last but not least, it is sometimes interesting to shuffle the rows of the TxN matrix in order to investigate the effect of temporal correlations in the neuronal net-

work using independent entropy measurements, which often represent a good control. For instance, one can compute the mean firing rate and assume homogeneity in the network, such that all neurons are assumed to have a mean firing rate of $\bar{p}(x_i)$. Alternatively, one can assume that there is heterogeneity in the neuronal network, and that each neuron has its own probability of firing $p(x_i)$. More on the matter will be introduced in Chapter 4 in the context of calcium imaging data modelling.

3.3.3 The Kullback-Leibler divergence

A traditional measure of discrepancy between two probability distributions is the Kullback-Leibler (KL) divergence [53], [54]. It is defined as follows:

$$D_{KL}(p||q) = \sum_i p(i).log_2\Big[\frac{p(i)}{q(i)}\Big] \tag{3.18}$$

where $p(i)$ and $q(i)$ represent two distinct probability distributions. The KL divergence is non-negative, $D_{KL}(p||q) \geqslant 0$, and $D_{KL}(p||q) = 0$ only when the two distributions are identical. In addition, the Kullback-Leibler divergence cannot be thought of as a distance. More specifically, it does not satisfy the symmetry criterion such that $D_{KL}(p||q) \neq D_{KL}(q||p)$. More on this will be introduced in section 4.1.

The KL divergence is usually closely related to the entropy computation. It can sometimes be thought of as a quantification of the entropy difference between different distributions. Traditionally, for small neuronal populations, one would need to perform point-by-point comparisons of probability distributions for all patterns from the two conditions being compared. This becomes an issue when one investigates populations involving large numbers of neurons (for instance, hundreds of neurons). Once again, it is possible to use another choice of probability distribution as the basis of KL divergence comparisons. For instance, one can use the parameter $p(k)$ which was estimated by the population-tracking model, and use this as the basis of the KL divergence. The discrepancy between two distributions would then be rewritten as:

$$D_{KL}(p||q) = \sum_{k=0}^{N} p(k).log_2\Big[\frac{p(k)}{q(k)}\Big] \tag{3.19}$$

where $p(k)$ and $q(k)$ refer to two different probabilities of k active neurons and N refers to the number of neurons in the neuronal network of interest.

Another choice of method involves a full distribution comparison. Instead of relying on a parameter (here, $p(k)$), it is possible to use computational tricks to speed up the divergence computation and get a reliable estimate of the point-by-point KL divergence. One such algorithm is introduced below (see Algorithm 1). Importantly, one should note that the use of for-loops often leads to very slow computational times. Therefore, although Algorithm 1 shows the gist of the KL divergence heuristic, one usually needs to use specific computational packages from Python (NumPy library) or MATLAB such as the “unique" function or/and histogramming built-in methods. More details regarding the implementation and analysis of Algorithm 1 can be found in Chapter 4.

Algorithm 1: KL-DIVERGENCE computes the "full" Kullback-Leibler divergence between two T×N spikes matrices*

Input: Two matrices of binary spikes $S1$ of size T1×N and $S2$ of size T2×N
Output: The KL divergence $D_{KL}(S1||S2)$

```
1  ns ← min(T1, T2);
2  S ← Stack(S1[1 : ns, 1 : N], S2[1 : ns, 1 : N]);
3  W ← RemoveRowRepetitions(S);
4  sort W in ascending order;
5  c1, c2, indices1, indices2 ← [];
6  W1 ← S[1 : ns, 1 : N];
7  W2 ← S[(ns + 1) : (2ns), 1 : N];
8  for w ∈ W do
9      if w ∈ W1 then
10         reference count of w from W1 in the vector c1;
11         if w ∈ W2 then
12             put a 1 in indices2;
13         else
14             put a 0 in indices2;
15     if w ∈ W2 then
16         reference count of w from W2 in the vector c2;
17         if w ∈ W1 then
18             put a 1 in indices1;
19         else
20             put a 0 in indices1;
21 p1 ← c1/Sum(c1);
22 p2 ← c2/Sum(c2);
23 KL ← D_KL(p1[indices2]||p2[indices1]);
24 return KL
```

**Note: T refers to the total number of time steps and N refers to the total number of neurons. The "full" KL divergence refers to a computation that doesn't use parameters, but rather uses the full spikes matrices*

Chapter 4

Modelling tests, results and analysis

This chapter is divided into 3 sections: synthetic data analysis, multielectrode array data analysis and calcium imaging data analysis.

4.1 Modelling tests on synthetic data

4.1.1 Description of the synthetically-generated spikes matrices

As was mentioned earlier in the dissertation, the purpose of the synthetic data is to ensure that the mathematical model of interest, here, the population-tracking model, works on datasets with specified mean firing rates and correlation coefficients (respectively the first and second moment). In this project, we followed the methods from Macke et al. (2009) [20].

First and foremost, it is important to note that the data we analyzed was in the format of binary TxN matrices, where rows refer to independent time steps, columns refer to independent neurons, and for each time step, all neurons could either be active (a '0' is in the cell) or inactive (a '1' is in the cell). In this section, we used T=1,000,000 timesteps. The binary matrices from [20] followed a Dichotomized Gaussian (DG) distribution, and could be fully accounted for by the data's mean firing rate and covariance properties. Each row represents "words", which correspond to distinct combinations of neural activity in the network. As was mentioned in Chapter 3, since each neuron can either be in the "ON" state (active) or in the "OFF" state (inactive), there are a total of 2^N possible words for a given neuronal network. For the sake of simplicity at this stage, we are not considering temporal correlations.

Sampling spikes data from a DG model operates in a two-step process. First, one needs to draw samples from a N-dimensional Gaussian random variable U. Second, a thresholding operation turns the Gaussian random variable U into a 0 or a 1, as is shown below (process extracted from [20]).

$$\left.\begin{array}{ll} X_i = 1 & \text{iff} \quad U_i > 0 \\ X_i = 0 & \text{iff} \quad U_i \leqslant 0 \end{array}\right\} \quad \text{where} \quad U \sim \mathcal{N}(\gamma, \Lambda) \tag{4.1}$$

Intuitively, we are trying to generate binary spikes data (big matrix of 0's and 1's) from a multivariate Gaussian distribution (U). The DG model takes the input pa-

rameters γ and Λ and turns them into a binary random variable X with firing rate **r** and covariance Σ. More precisely, the relationship between the parameters of X and the parameters of U are as follows, for any neurons i and j ($i \neq j$):

$$r_i = \Phi(\gamma_i) \tag{4.2}$$

$$\Sigma_{ii} = \Phi(\gamma_i)\Phi(-\gamma_i) \tag{4.3}$$

$$\Sigma_{ij} = \Psi(\gamma_i, \gamma_j, \Lambda_{ij}) \quad \text{where} \quad \Psi(\gamma_i, \gamma_j, \Lambda_{ij}) = \Phi_2(\gamma_i, \gamma_j, \Lambda_{ij}) - \Phi(\gamma_i)\Phi(\gamma_j) \tag{4.4}$$

where Φ refers to the univariate Gaussian cumulative distribution (mean 0 and variance 1) and Φ_2 refers to the cumulative distribution of a bivariate Gaussian cumulative distribution.

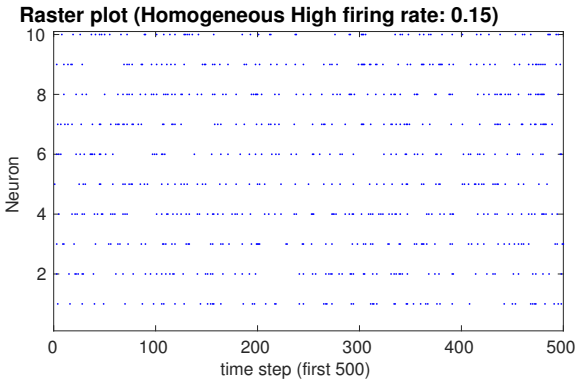
The choice of mean firing rate and correlation strength between neurons was inspired from the O'Donnell et al. (2016) [1] paper (section 2.5). The synthetically generated DG-distributed binary spikes matrices were as follows (see Table 4.1):

Table 4.1: Synthetically-generated DG spikes data properties (N=10)

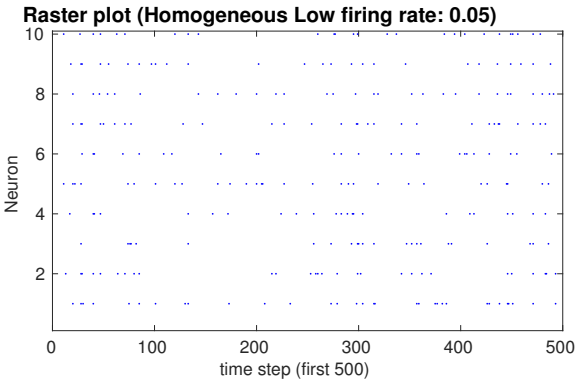
#	**Homogeneity**	**Firing rate**	**Mean rate**	**Correlation***	**Covariance***
1	Homogeneous	High	0.15	0.10	0.0128
2	Homogeneous	Low	0.05	0.10	0.0048
3	Heterogeneous	Uniform	0.20	[0.05,0.10]	[0.01,0.02]
4	Heterogeneous	Non-uniform	[0.20,0.30]	[0.04,0.10]	[0.01,0.02]

**Note: These values only apply to non-diagonal entries of the correlation and covariance matrices*

The four DG-generated datasets of binary spikes referred to in Table 4.1 will later be referred to as the *Homogeneous-High*, *Homogeneous-Low*, *Heterogeneous-Uniform* and *Heterogeneous-Non-uniform* datasets. All had 10 neurons (N=10) and 1,000,000 timesteps (T=1,000,000). Assuming that each time bin had a duration of 10ms, we could infer that the Homogeneous High dataset had an overall firing frequency of 15 Hz, the Homogeneous Low dataset had a firing frequency of 5 Hz, the Heterogeneous Uniform dataset had a firing frequency of 20 Hz and the Heterogeneous Non-uniform dataset had a firing frequency of 24 Hz. All neurons from the homogeneous datasets had the same mean firing rate (0.15 and 0.05 respectively), and all pairs of neurons had a correlation coefficient $p = 0.1$. As for the heterogeneous datasets, neurons from the *uniform* dataset had a mean firing rate of 0.20, whereas neurons from the *non-uniform* dataset had a mean firing rate which was uniformly distributed on the interval [0.20,0.30]. Both heterogeneous datasets had symmetric covariances between pairs of neurons ($\Sigma_{ij} = \Sigma_{ji}$), which were uniformly distributed on the interval [0.01,0.02]. The variance of each neuron was given by $Var_i = r_i(1 - r_i)$ for a neuron i, and the covariance between pairs of neurons followed the equation: $Cov_{ij} = p_{ij}.\sqrt{Var_i\, Var_j}$ where i and j are any given pair of neurons. In the case of the datasets 1-3 in which all neurons shared the same mean firing rate, we could use $Cov_{ij} = p_{ij}.Var_i$ since $Var_i = Var_j$. Following the above procedure, one could obtain the mean firing rates vector $\boldsymbol{r}$ and covariance matrix

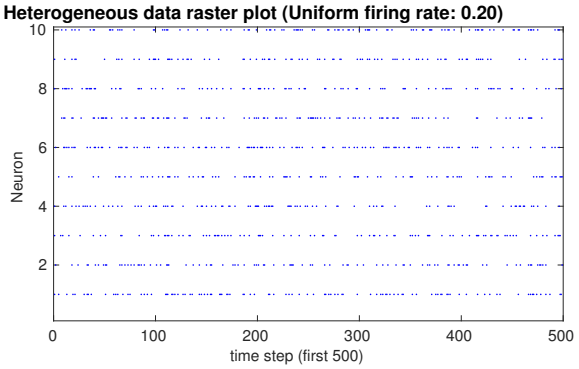


(a) Homogeneous High spikes raster plot

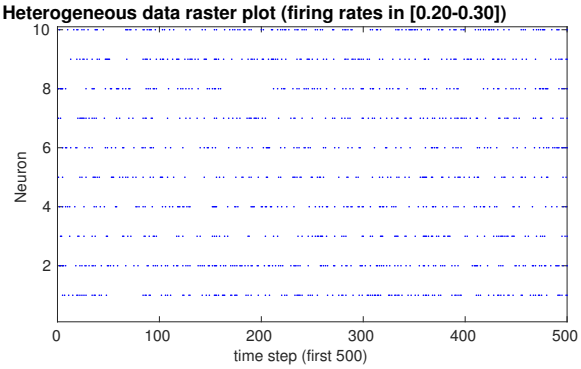


(b) Homogeneous Low spikes raster plot

Figure 4.1: Homogeneous spikes raster plots



(a) Heterogeneous Uniform spikes raster plot



(b) Heterogeneous Non-uniform spikes data

Figure 4.2: Heterogeneous spikes raster plots

Σ for each of the 4 datasets. For all sets of parameters, we generated TxN matrices with T = 1,000,000 and N = 10.

Figure 4.1a shows a raster plot of the Homogeneous High spikes dataset, and Figure 4.1b shows a raster plot of the Homogeneous Low spikes dataset. Each blue dot represents a spike corresponding to neuron i (row on graph) and time step T (column column on graph). As we can see, Figure 4.1b shows sparser spikes for each neuron, which is what we would have expected. More on the properties of the Homogeneous datasets can be found in the Appendix on Figure A.1, in which we explicitly show that the two datasets have the features that we claimed they have in Table 4.1.

Figure 4.2a shows a raster plot of the Heterogeneous Uniform spikes dataset, and Figure 4.2b shows a raster plot of the Heterogeneous Non-uniform spikes dataset. As we can see, Figure 4.2b has slightly more irregularities in its firing behaviour than Figure 4.2a which is what we would have expected. Once again, more of the properties of the Heterogeneous datasets can be found in the Appendix on Figure A.2, in which we explicitly show that the two datasets have the features that we claimed they have in Table 4.1.

Now that the four different synthetic datasets have been introduced, we are ready to test our mathematical model. As mentioned in Chapter 3, there are three main

parameters that the population-tracking model method [1] outputs: the probability of k active neurons $p(k)$, the probability of independent neurons x_i being active in the neuronal network $p(x_i)$, and the conditional probability of independent neurons being active in the network given that there are k neurons active $p(x_i|k)$.

Baseline plots of $p(k)$ and $p(x_i)$ can be found in the Appendix (see Figure A.3 and Figure A.4). They mainly show that the model has learned the right properties of the four datasets, and that the probability of k active neurons decreases for larger numbers of simultaneous k active neurons.

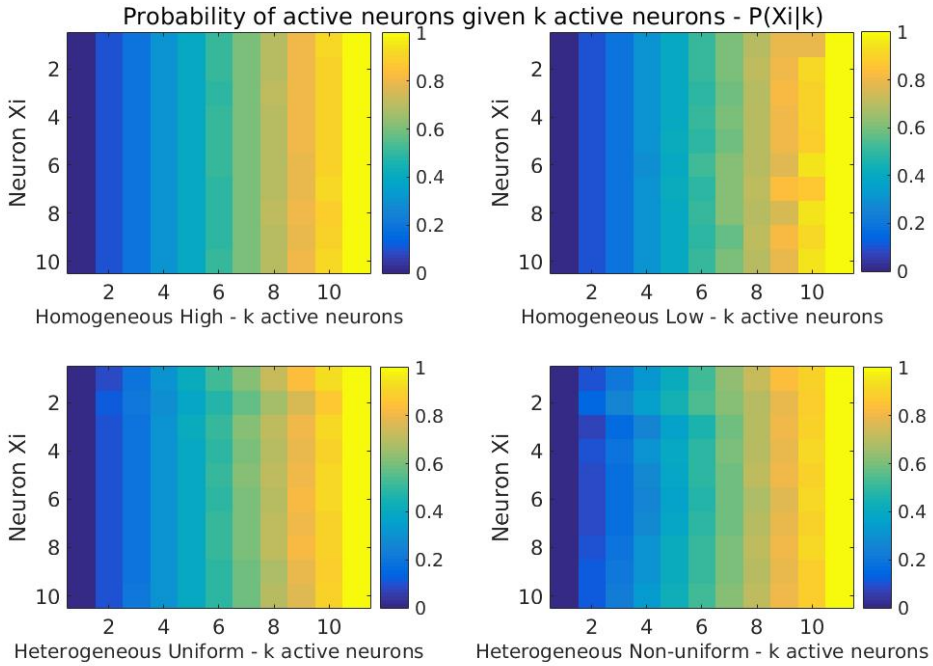


Figure 4.3: Conditional probabilities of neurons being active given k active neurons

With respect to the parameter $p(x_i|k)$, we are investigating matrices of size Nx(N+1) for each dataset on Figure 4.3. The X-axis indicates each value of k active neurons, and the Y-axis indicates the conditional probabilities of each independent neurons. As a result, we get probability values for all combinations of neurons and k active neurons. Although some slight differences exist between each dataset, we can see that the results are fairly similar across datasets. Namely, as the number of k active neurons increases, the probability of every independent neuron being active increases as well (it goes from blue to yellow). This result makes sense as we would expect neurons to be more likely to fire if there are more neurons active in the network.

4.1.2 Synthetic data modelling over time steps

As was mentioned before, the baseline synthetic data tests were all performed on a total of T=1,000,000 timesteps. However, we haven't yet described how the parameters of the model tend towards their "true" value over time steps. Figure 4.4 shows the predicted pattern probabilities as a function of true pattern probabilities

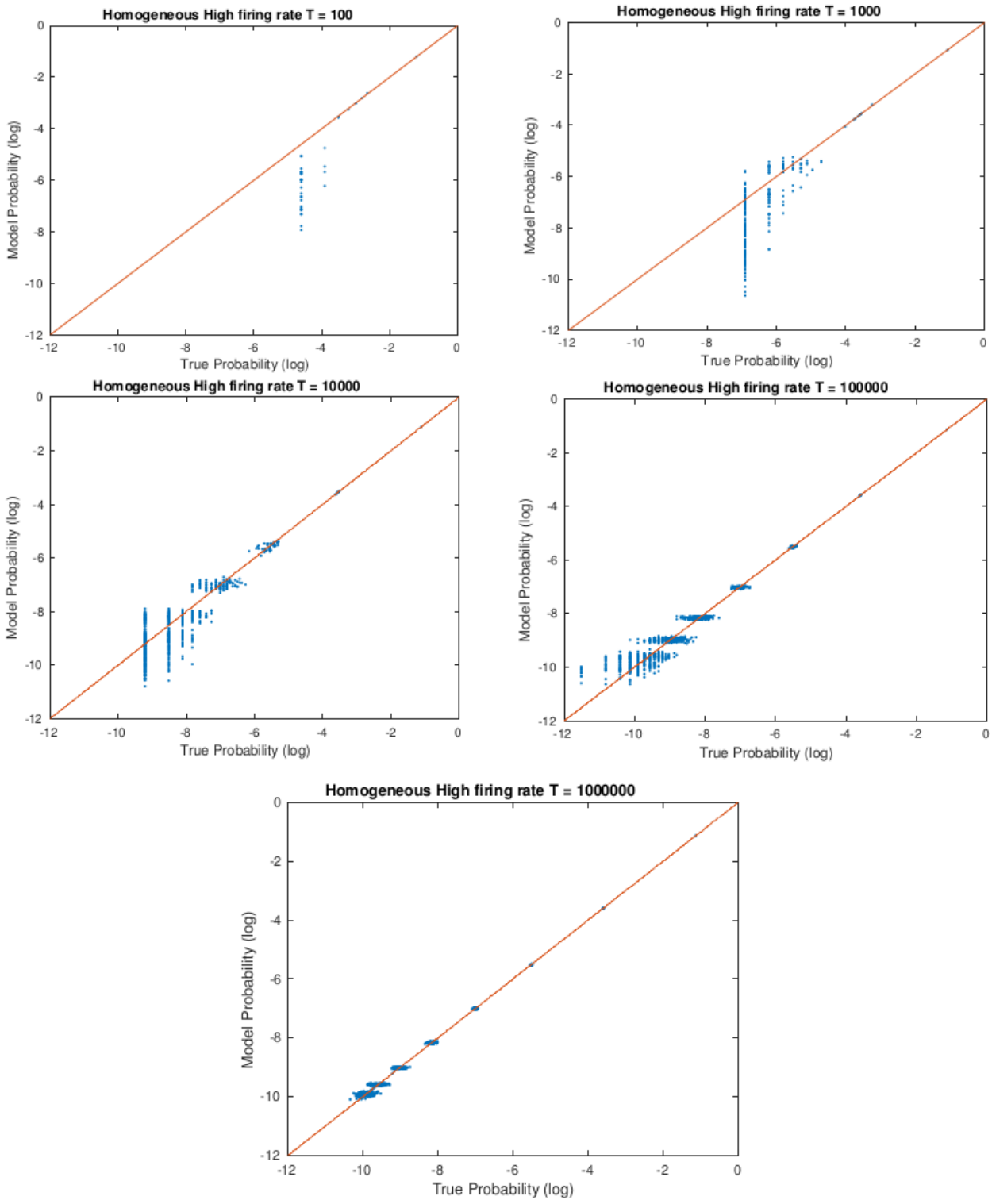


Figure 4.4: Evolution of time steps in the Homogeneous High dataset

for a population of 10 neurons in the Homogeneous High dataset. The true probability was computed empirically using a histogramming function. Since we have 10 neurons, there are $2^{10} = 1024$ different words (combinations of active/inactive neu-

rons) that are possible. For all 1024 patterns, we computed the individual empirical probability of each word using frequency counts, and we computed the *model* probability in order to compare the true probability against the model probability (see Chapter 3 for the specific description of the methods). T's ranged from T= 10^2 to T= 10^6. Note that we used a logarithmic scale (natural logarithm).

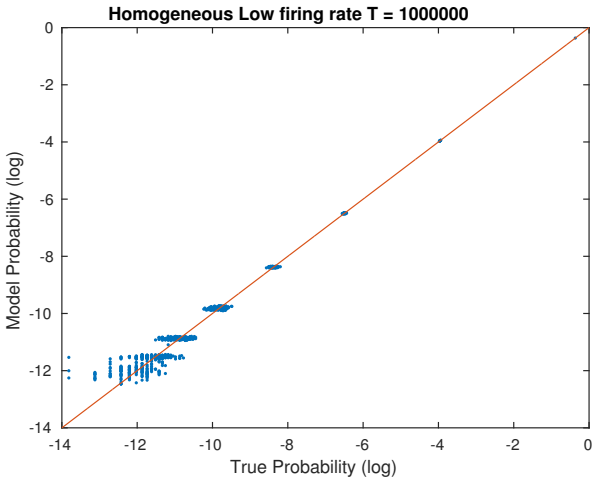
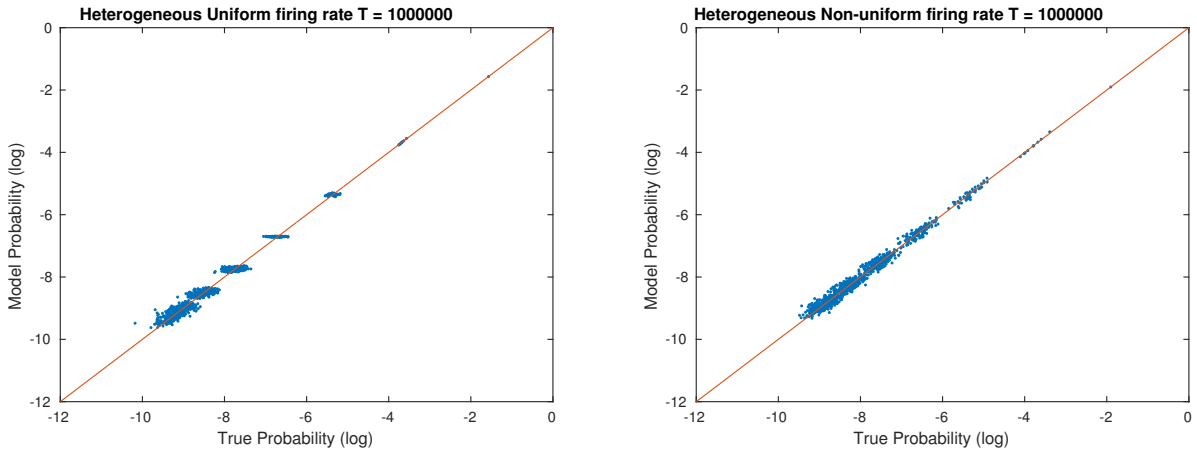


Figure 4.5: Homogeneous Low Model profile at T = 10^6

We included the full evolution of the model probability in the Homogeneous High case only on Figure 4.4. A time step analysis of the model probabilities of the three other datasets can be found in the appendix (see A.5, A.6 and A.7). Figures 4.5, 4.6a and 4.6b show the model pattern probability profile at T = 10^6 for the Homogeneous Low, Heterogeneous Uniform and Heterogeneous Non-uniform datasets respectively. Our main observation was that the model probability got closer and closer to the empirical true probability as the number of time steps increased. For T= 100, most of the predicted probability values were off the line. The more T increased in the powers of ten's, the more we could observe a clustering effect of the model probability around the identity line ($y = x$). This result suggests that having more firing spikes data (more timesteps in our matrices) does yield a better estimation of our model probability distribution. Therefore, when using the population-tracking model, one would hope to get spikes matrices that would last as long as possible in order to collect more words and get better estimates.

It is interesting to compare the spread of the model probability distribution on the identity line from each dataset type. In the case of the Homogeneous datasets (Figures 4.4 and 4.5), we observe a discretization of the model probabilities into chunks. As was mentioned earlier, these datasets have neurons which share the same mean firing rate and they also have the same covariance between each pair of neurons. On the other hand, for a dataset such as the heterogeneous dataset with non-uniform mean firing rates and pairwise covariances, we notice that the model probabilities are much more spread out over the identity line (see Figure 4.6b). As to the heterogeous data set with heterogenous pairwise covariances and uniform mean firing rates across neurons of the neuronal network, we notice that the model probabilities lie in between a fully discretized version such as the Homogeneous



(a) Heterogeneous Uniform probability at T = 10^6 (b) Heterogeneous Non-uniform probability at T = 10^6

Figure 4.6: Investigation of the heterogeneous model properties at T = 10^6

High or Low dataset and a more fluid version like the Heterogeneous Non-Uniform dataset.

We now turn to an investigation of the entropy over the timesteps for each of the different datasets. As was mentioned earlier, the entropy is a reliable measure of the "richness" of the spikes data. [55]. We first investigate the full empirical entropy in each of our synthetic datasets. We computed the empirical entropy in agreement with the methods referenced in [1]. The empirical entropy was divided into the entropy of the population synchrony distribution (using the parameter $p(k)$) and the conditional entropy of the pattern probability distribution given K, as shown below:

$$H = H_k + H(p(\{x\}|k)) \tag{4.5}$$

$$H = -\sum_{k=0}^{N} p(k).log_2\, p(k) - \sum_{k=0}^{N} p(k)\Big[\sum_{i} p(\{x\}_i|k).log_2\, p(\{x\}_i|k)\Big] \tag{4.6}$$

where $p(k)$ indicates the probability of k active neurons at a time step in the neuronal network, and $p(x)$ indicates the probability of a given word, or combination of active/inactive neurons denoted as x. According to O'Donnell et al. (2016) [1], since all neurons from homogeneous synthetic datasets have the same mean firing rates, all $\binom{N}{k}$ patterns have the same probability of occurring for any given value of k, $p(\{x\}|K = k) = p(k)/\binom{N}{k}$, which leads to the maximization of the theoretical homogeneous entropy term as follows:

$$H_{theor.} = \sum_{k=0}^{N} p(k).log_2 \frac{\binom{N}{k}}{p(k)} \tag{4.7}$$

Nevertheless, these statistical estimations based on the spikes matrices are not sufficient in and of themselves to get a reliable estimate of the underlying "True" entropy.

Table 4.2: Interpolation results in the four synthetic datasets and final entropy values

#	Homogeneity	Firing rate type	Entropy/N*	Linear fit	Polynomial fit*
1	Homogeneous	High	0.5901	0.5769	0.5892
2	Homogeneous	Low	0.2717	0.2615	0.2667
3	Heterogeneous	Uniform	0.7034	0.6874	0.6981
4	Heterogeneous	Non-uniform	0.7632	0.7391	0.7564

**Note: N=10. The polynomial fit was of degree 2. The enropy per neuron was computed at T= 10^6*

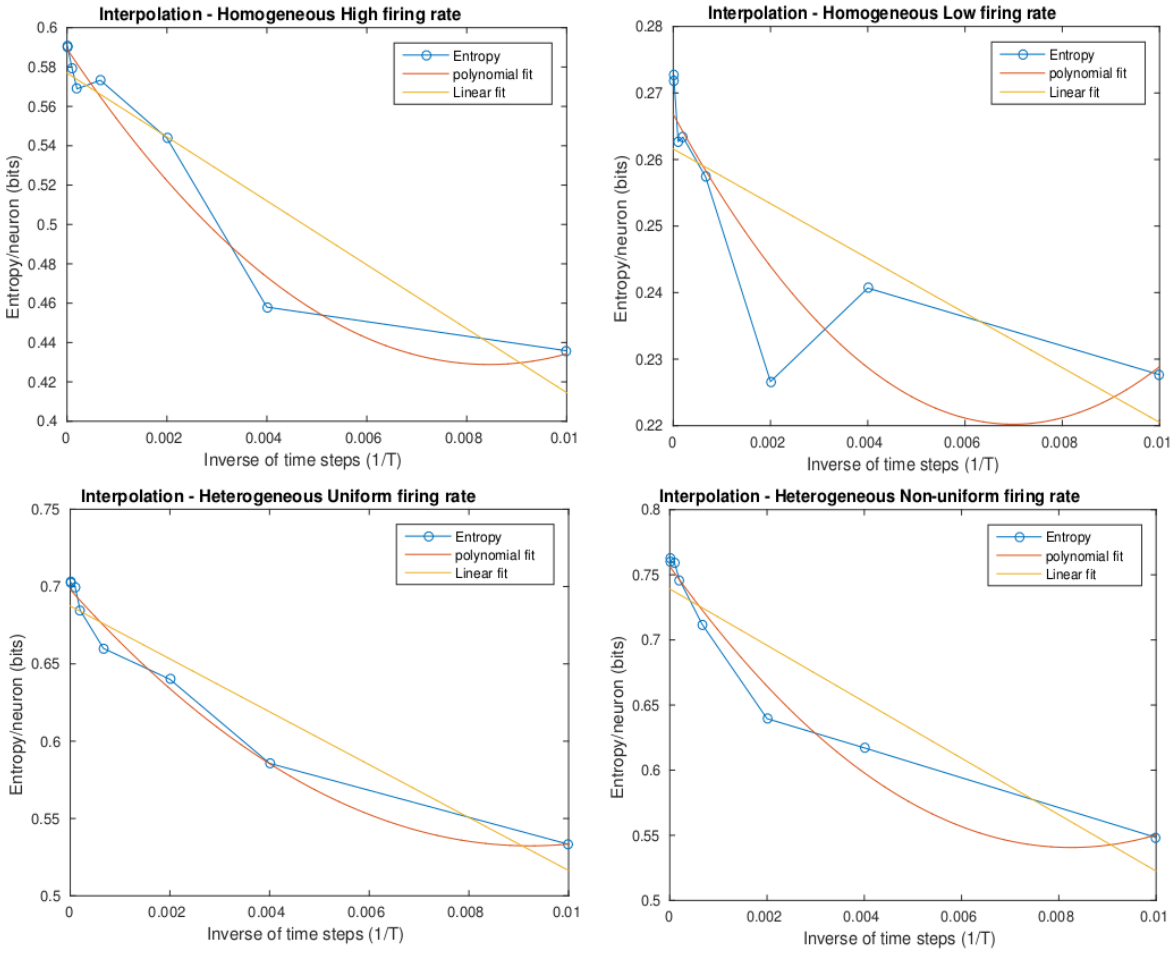


Figure 4.7: Entropy interpolation of each synthetic dataset as a function of 1/T

Unfortunately, it is difficult to know what the true entropy might be if we only base our estimates on the data that we have collected. However, it is possible to apply an interpolation method to fit our data over the time steps, and use this to extrapolate when T$\rightarrow \infty$. In order to do so, we plotted the entropy per neuron (Y-axis) against the inverse of our timesteps (1/T on the X-axis). We then fit a line and a polynomial of degree 2 to the data. Note that in order to get better fits, we recorded entropy

values at the following time steps: T = $\{100, 250, 500, 1500, 5000, 10^4, 10^5, 10^6\}$. The Entropy/N value that corresponds to $T \to \infty$ and that we want to extrapolate is the intercept of the linear and polynomial fits at X=0. Table 4.2 shows our best estimate of the Entropy per neuron and the intercept at X=0 for both linear and polynomial fits. Figure 4.7 shows a visualization of the interpolation process with both a linear and polynomial fit for each of the four synthetic datasets.

As we can see on Figure 4.7, the polynomical curves (in orange) fit the data better than the linear fits (in yellow). When looking at the results from Table 4.2, we also see that the intercepts from the polynomial fits are closer to the final value of Entropy/N at $T = 10^6$ than the intercepts from the linear fits. Therefore, the main conclusion from these interpolation tests is that the polynomial fits yield a better estimate of the "True" Entropy than the linear fits.

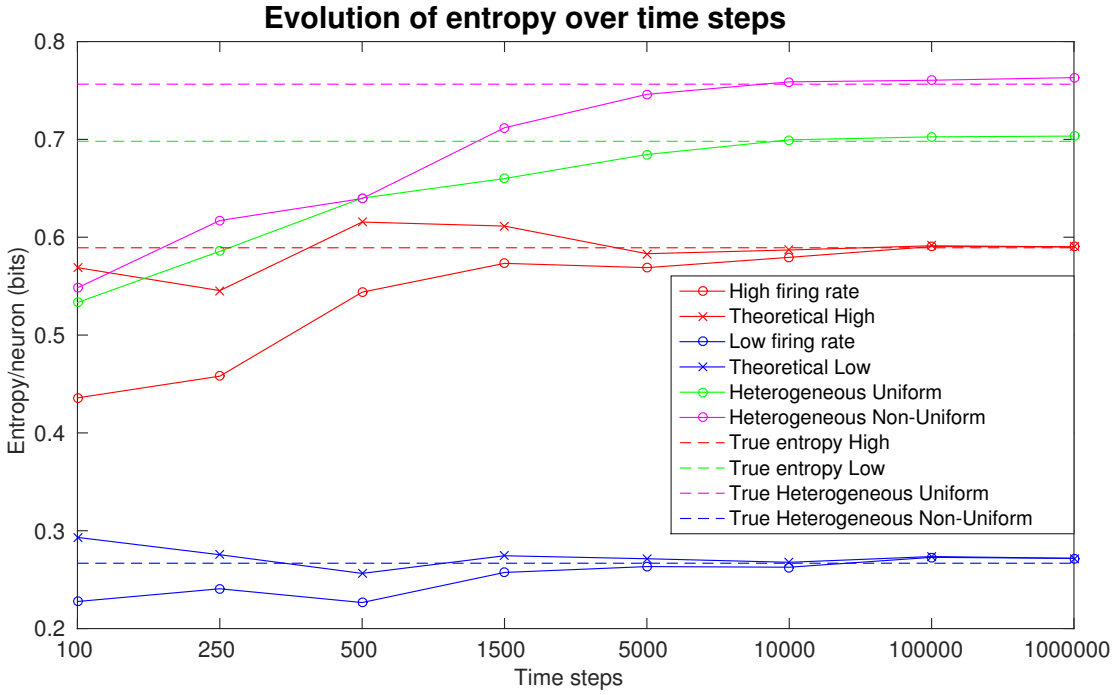


Figure 4.8: Evolution of the empirical Entropy per neuron over time steps and comparison between theoretical and empirical entropy per neuron in the Homogeneous datasets.

We first investigated the empirical entropy per neuron for all synthetic datasets. Figure 4.8 summarizes the entropy results. The dashed lines indicate the true entropy values (as introduced previously), and the circled-lines show the empirical evolution of the entropy per neuron as a function of the number of time steps. The Homogeneous High entropy results are shown in red, the Homogeneous Low entropy results are shown in blue, the Heterogeneous Uniform entropy results are shown in green and the Heterogeneous Non-uniform entropy resuls are shown in pink. The main result was that the empirical entropy per neuron converges towards its true value as a function of the time steps. A good turning point for the estimation of the entropy was T= 5,000 since the bias was inferior or equal to 0.025 for

all datasets. This means that it is usually preferable to collect about 5,000 words in order to get a reliable estimate of the entropy per neuron.

We now want to compare the empirical entropy (line with circles) with the theoretical entropy (line with crosses) in the case of our homogeneous datasets. Since the theoretical computation of the entropy per neuron could only be processed for homogeneous datasets in which the mean firing rates were the same, we only included the Homogeneous High and Homogeneous Low datasets in our comparison. Results are also shown on Figure 4.8. We saw that the theoretical approximation of the entropy value got closer to the true entropy value faster than the empirical entropy value for both the Homogeneous High dataset (in red) and the Homogeneous Low dataset (in blue). This result can be explained by the fact that the theoretical computation of the entropy per neuron (equation (4.7)) doesn't sum over all $p(\{x\}|k)$ values, but rather only considers the $p(k)$ estimate.

4.1.3 Statistical comparison of multi-neuron activity

In the context of neurobiological experiments, one is often interested in comparing different spikes datasets. One way of doing so is to compute the mathematical difference measure $D_{KL}(p||q)$ which compares the probability distribtions p(x) and q(x) from two datasets of interest. To reiterate, the Kullback-Leibler (KL) divergence is computed as follows:

$$D_{KL}(p||q) = \sum_i p(i).log_2\Big[\frac{p(i)}{q(i)}\Big] \tag{4.8}$$

In this section, we are comparing two different methods which compute the KL divergence: a p(K) (population rate) KL approximation, and a KL heuristic method. With respect to the first method, we simply assume that the probability distribution that we are interested in is the probability of k active neurons $p(k)$ that is output by the population-tracking model, such that:

$$D_{KL}(p||q) = \sum_k p(k).log_2\Big[\frac{p(k)}{q(k)}\Big] \tag{4.9}$$

As for the second method (heuristic method), we use a function that leads to a KL divergence measure by combining the two spikes matrices of interest, sorting them in increasing order, computing probability distributions of the two halves and using these probability distributions as a way to compute the KL divergence. One important point to make about this method is that it uses a pointwise comparison of the two full probability distributions of interest, and evaluates the divergence measure with relatively low time complexity due to the computational tricks it makes use of. For a more detailed description of the implementation of the heuristic KL divergence method, please see section 3.3.3.

The first point to make prior to comparing the two different KL divergence methods is that the $D_{KL}(p||q)$ divergence measure between two distributions is *not* symmetric by definition. This means that we usually don't get $D_{KL}(p||q) = D_{KL}(q||p)$. An illustration of unsymmetrized $D_{KL}(p||q)$ and $D_{KL}(q||p)$ differences is shown on Figure 4.9. As we can see on this histogram, the KL divergences $D_{KL}(p||q)$ and

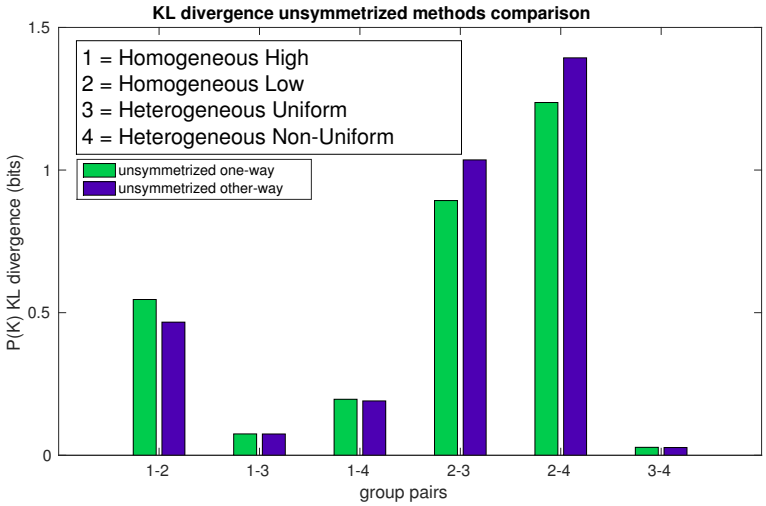
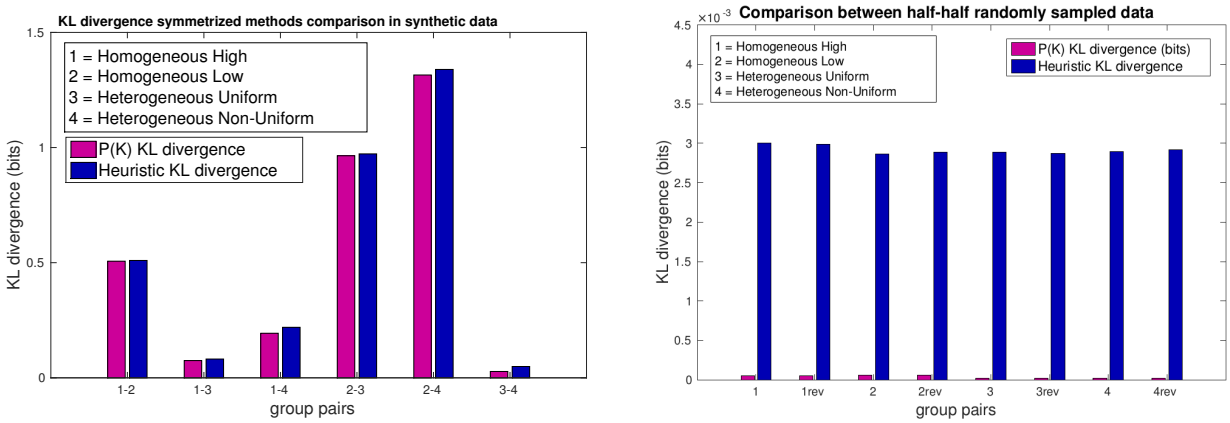


Figure 4.9: Unsymmetrized KL divergence in the P(K) approximation method



(a) Synthetic datasets KL divergence comparison

(b) Half-half random samples KL divergence

Figure 4.10: Baseline Kullback-leibler comparison

$D_{KL}(q||p)$ are often different in the pairwise comparisons of the different synthetic datasets. The difference between each KL divergence varies from 0.0003 to 0.16, which means that in some cases the divergence will be similar, and in some others it won't be. The main conclusion from these preliminary results is that we need to investigate the symmetrized version of our KL divergence of interest in order to get more reliable results.

In this section of the analysis, we wish to argue that the P(K) approximation method of the Kullback-Leibler divergence is a better estimate than the heuristic KL divergence. We first plot the symmetrized KL divergences using both the P(K) KL approximation and the heuristic KL method on Figure 4.10a. We then make a baseline comparison by randomly shuffling the rows of our synthetic datasets, splitting them into two halves, and measuring the KL divergence between the halves one-way (1,2,3 and 4) and the other-way (1_{rev},2_{rev},3_{rev} and 4_{rev}) with the P(K) approximation method and the heuristic method (see Figure 4.10b).

With respect to the results from Figure 4.10a, we see that the KL divergence esti-

mates for all pairwise synthetic datasets comparisons are quite similar for the P(K) KL divergence and the Heuristic KL divergence methd. This suggests that the parameter $p(k)$ seems to carry enough information in itself (it meets the *sufficiency* criterion as a statistic) to be used as the basis of probability distribution comparisons, since it yields the same results as another method which compares probability distributions based on the full spikes datasets. As to the results from Figure 4.10b, we notice that the baseline KL divergences between the randomly sampled half datasets halves are much smaller in the P(K) approximation KL divergence case. Indeed, baseline KL divergences that use the P(K) approximation method lie in the interval $[2.0\times10^{-5}, 5.8\times10^{-5}]$ whereas baseline KL divergences that use the heuristic method lie in the interval $[2.9\times10^{-3}, 3.0\times10^{-3}]$. As we can see, there is a difference of 2 orders of magnitude between the two baseline KL divergences, which suggests that the P(K) approximation KL divergence may indeed be a more reliable method than the heuristic method.

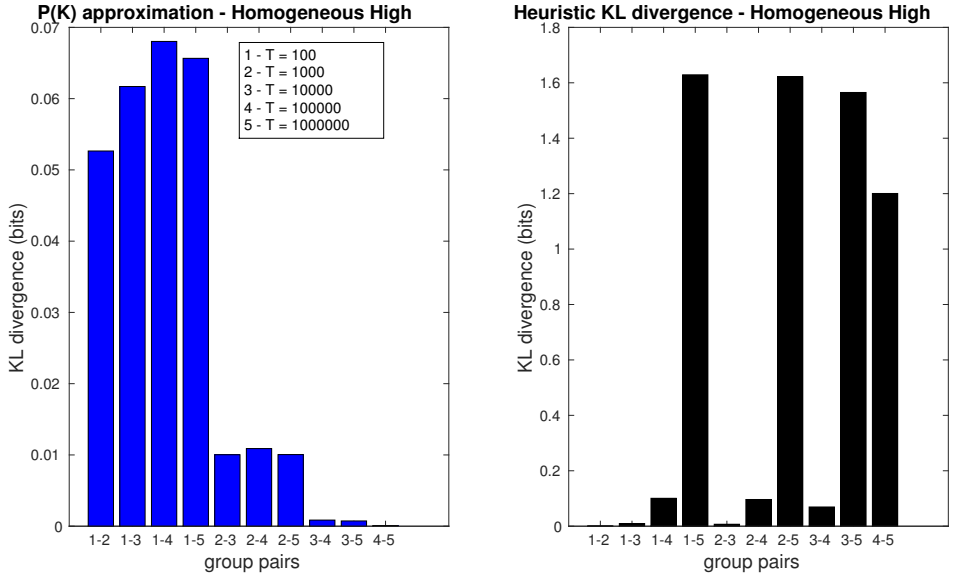


Figure 4.11: P(K) and Heuristic KL divergence comparison in the Homogeneous High dataset

We then wish to compare the two KL divergence methods in a timestep comparison. For the sake of concision, we only introduce the Homogeneous High dataset to support our point. Figure 4.11 displays all pairwise comparisons between the synthetic Homogeneous High datasets computed at the following time steps: T =$[10^2, 10^3, 10^4, 10^5, 10^6]$. As we can see, results differ greatly between the P(K) approximation method and the heuristic KL divergence method. On the one hand, the P(K) KL divergence (to the left in blue) is *bigger* for divergences including the first dataset (T = 100) than KL divergences including datasets with more timesteps (more rows, therefore, more accuracy in the estimation). On the other hand, the heuristic KL divergence (to the right in black) displays opposite results with a *smaller* divergence for divergences including the first spikes matrices with fewer time steps, than divergences including the last spikes matrices with T = 10^6. In addition, do notice the difference in the scale

of the Y-axis with a max divergence of 0.68 in the P(K) approximation case, and a max divergence of 1.63 in the heuristic case. These results are due to the fact that the heuristic method needs to combine matrices of the same size before sorting them by rows. If we compare matrices of different size (say T = 10^3 and T = 10^6), then we leave out 999,000 rows of spikes data, which represents more than 99% of the spikes data of the second dataset, and our KL divergence cannot be reliable anymore (see Algorithm 1). This problem does not happen with the P(K) approximation method because we have the same number of entries in the P(K) vector (that is: N+1) since the number of neurons in the network does not change, regardless of the number of timesteps that we are investigating. As we have seen with the entropy measurements earlier, we would expect the KL divergences to be smaller and smaller over the timesteps since we are getting more and more accurate that tend towards their true value (for instance, we expect the difference between T = 10^5 and T = 10^6 to be much smaller than the difference between T = 10^2 and T = 10^3). Therefore, the main conclusion from these results is that the symmetrized P(K) approximation KL divergence is indeed a better choice of methods.

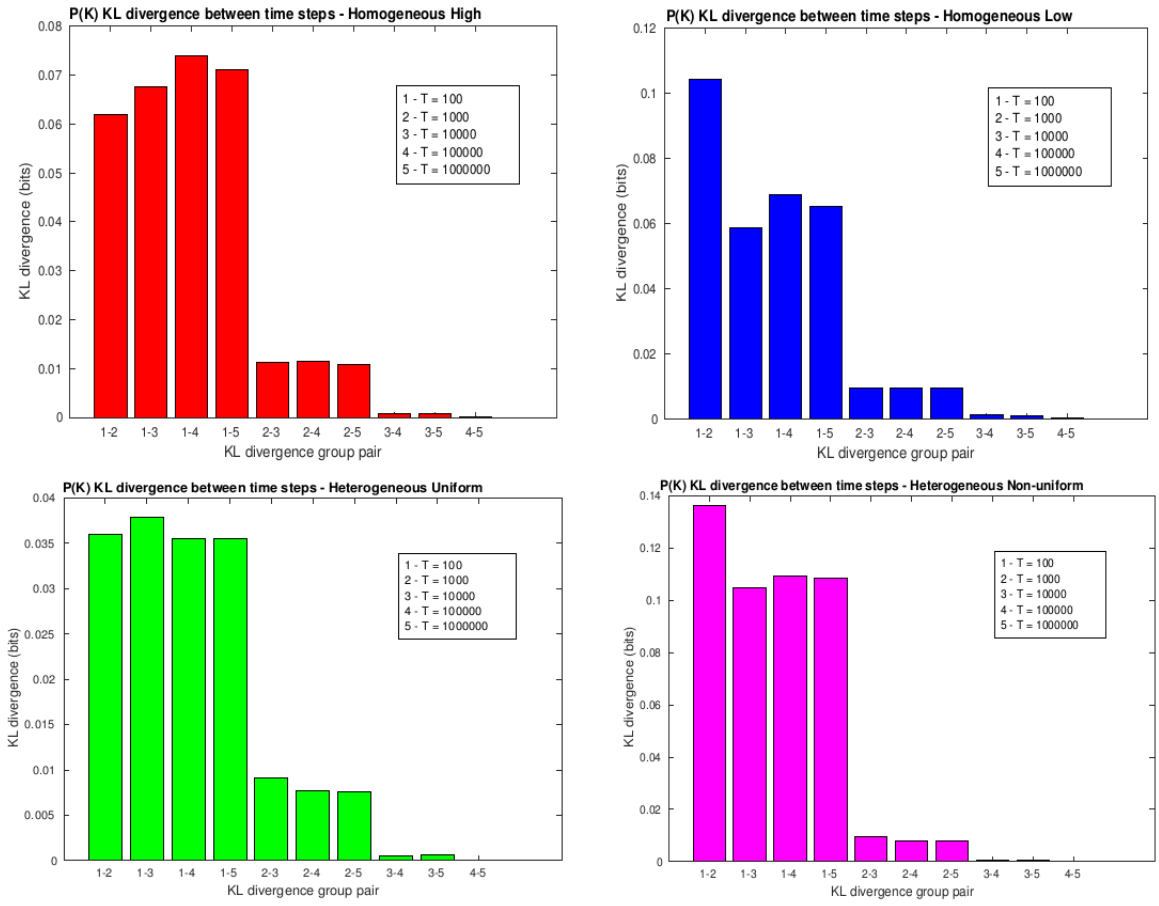


Figure 4.12: P(K) symmetrized KL divergence between spikes generated at different time steps for each synthetic dataset

Finally, using our P(K) approximation symmetrized KL divergence method, we investigated the KL divergence between spikes computed at each time step for each

synthetic dataset. Results are displayed on Figure 4.12. As we can see, results are consistent across all types of datasets. The KL divergence between datasets becomes smaller for spikes matrices with more time steps (more words included in the matrix).

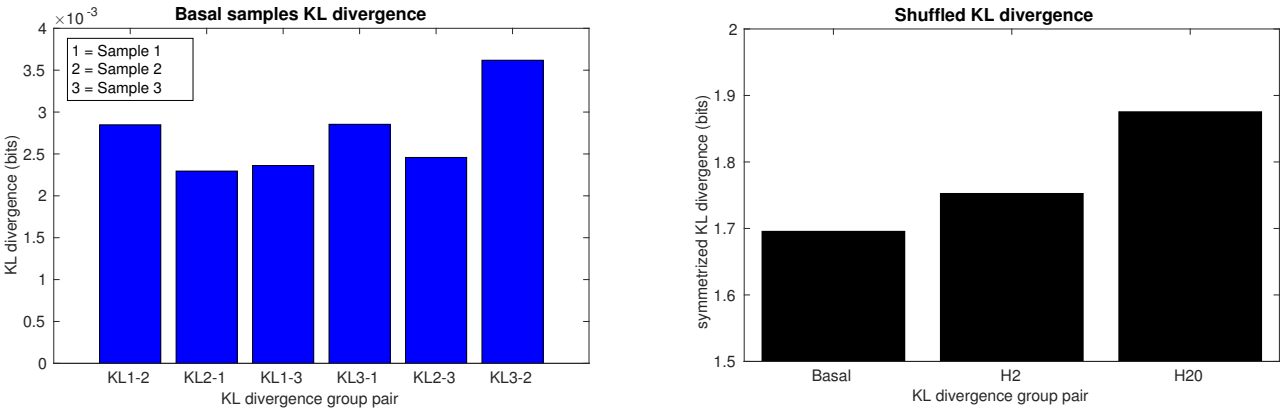
4.2 Multielectrode array (MEA) data modelling

We now wish to investigate the modelling of real data extracted from a multi-electrode array experiment, as described in Chapter 3. The three sets of data that we analyzed in this section were as follows (see Table 4.3).

Table 4.3: Multielectrode array (MEA) spikes data

#	Name	Description	Preprocessing	N	T	Recording
1	Basal	Baseline spikes	clustered	253	90000	900 s.
2	H2	Spikes after 2 hours	clustered	253	89998	900 s.
3	H20	Spikes after 20 hours	clustered	253	89999	900 s.

To reiterate, this data represented an experimental control to ensure that the population-tracking model behaved as it should on real data. Neural activity was recorded with 4,096 channels MEA from dissociated hippocampal neurons and was clustered into a neuronal network of 253 “neurons” from which we are investigating spikes behaviours. The 15 minutes long recording were divided into time bins of 10 ms (see the Methods from Chapter 3 for more information).



(a) Baseline unsymmetrized samples KL divergence

(b) Baseline KL divergence raw-shuffled

Figure 4.13: Baseline KL divergence measurements

Similarly to the methodology used in the synthetic data section of the analysis, we wished to carry out some baseline tests on the MEA datasets. Figure 4.13 reports the results from two of our MEA baseline tests. After estimation of the parameters $p(k)$, $p(x_i)$ and $p(x_i|k)$ based on the MEA spikes data matrices, one could use these probabilities to generate new samples of spikes (see Chapter 3, section 3.1 for more

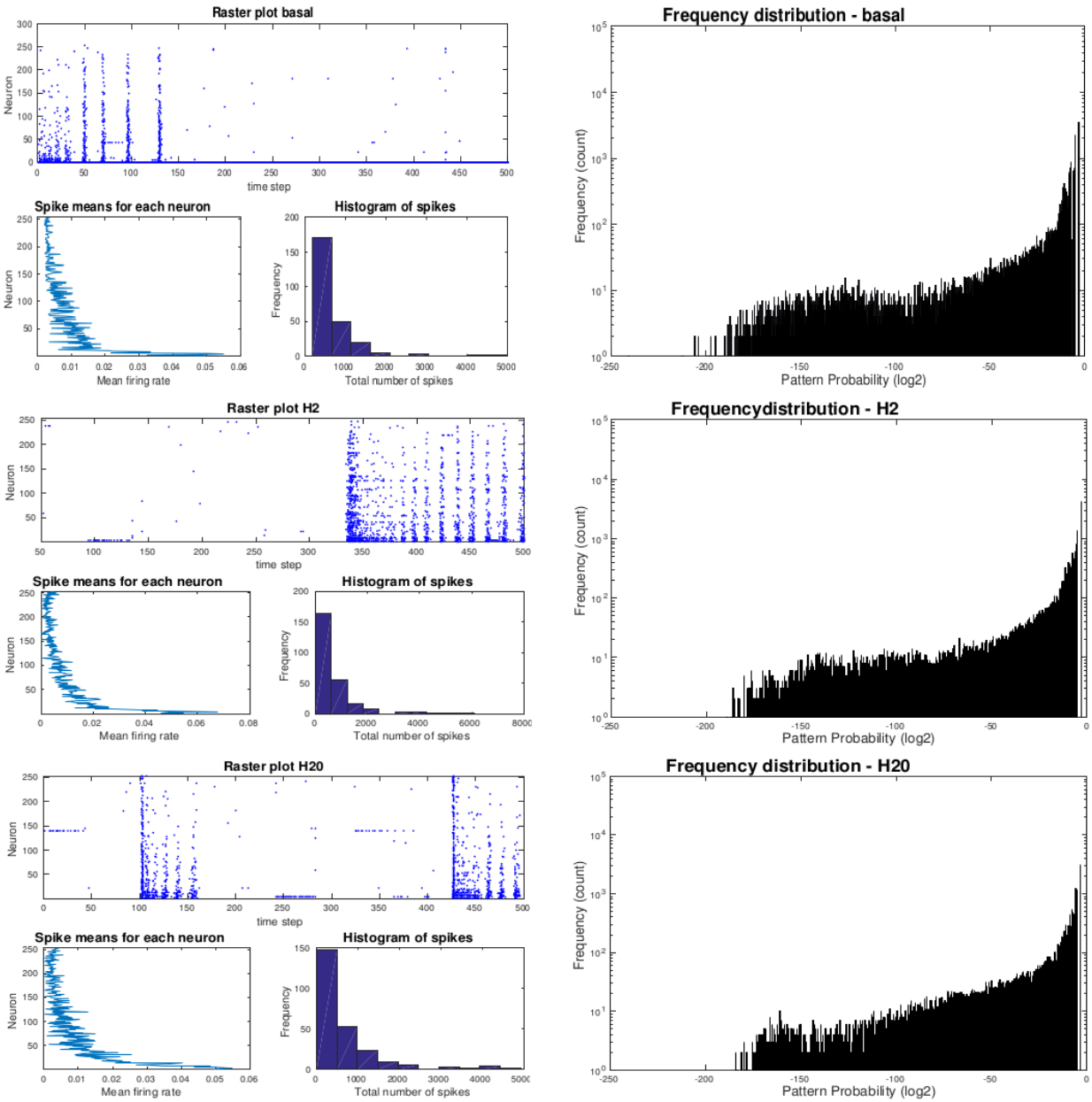


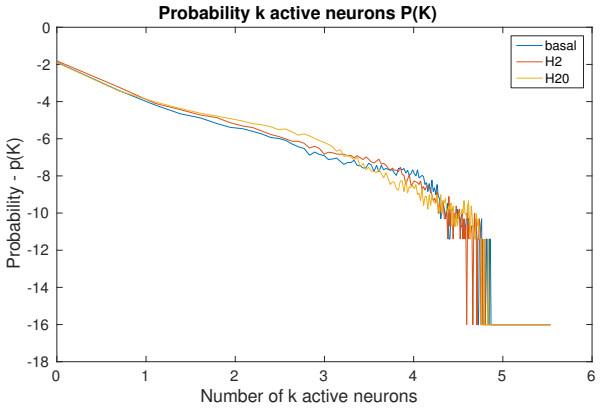
Figure 4.14: Raster plots and frequency distributions for each MEA dataset

details regarding the estimation method and sampling process). Figure 4.13a investigates the KL divergence between 3 generated samples based on the parameters from the basal MEA dataset. All samples had 10^5 timesteps (rows). The idea was that a successful parametrization of the spikes data should lead to spikes samples with a low KL divergence between them. On Figure 4.13a, we are plotting the KL-divergence in the two unsymmetrized ways ($D_{KL}(p||q)$ as well as $D_{KL}(q||p)$). The divergence measure ranged from 0.0017 to 0.0028 (10^{-3} orders of magnitude).

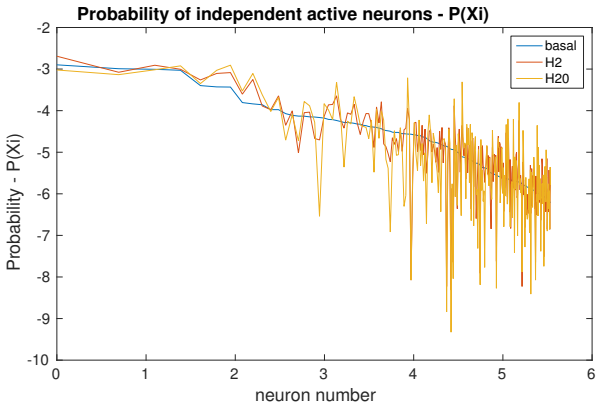
As to Figure 4.13b, we shuffled all the rows from the initial MEA datasets and compared the raw MEA datasets to the shuffled MEA datasets using the KL divergence method. Results from the comparison yielded a mean of 1.77 and a standard deviation of 0.09 across all pairwise comparisons. Overall, we observe a noticeable

difference between the raw MEA data and the shuffled MEA data, which suggests the presence of important temporal correlations in the data. Also note the increase in KL divergence from the basal dataset to the H20 dataset. This effect may suggest increasing levels of correlations between timesteps, since the shuffling lead to increasing KL divergences.

We now evaluate some properties from each MEA dataset. Figure 4.14 investigates the raster plot profiles (first 500 time steps), spike means, spike count histograms and full frequency distribution (on a logarithmic-scale) for each of our MEA spikes datasets. With regards to the spikes frequency distributions (plotted to the right on Figure 4.14), we computed the logarithm of each word ($log_2(p(\{x\}))$) and plotted them in the form of a histogram. The first main observation from the raster plots was that neurons tended to fire together rather than individually in the neuronal network. This suggests that there may be some inter-dependencies and high correlations between neurons. The second observation was that the mean firing rate for all neurons seemed to be around 0.01 which is a very low firing rate. We also notice that some of the neurons' mean firing rate tended to increase slightly as time passed by (see bottom of the spike means graphs in each condition). With respect to the histograms of the sum of spikes per word, we observe a similar profile overall, and we can see on the frequency distributions that there is a new peak in the pattern probabilities around -170 on the log scale that seems to emerge as time passes by. This potentially suggests a slow remodelling of the neuronal network over time.



(a) Probability of K active neurons (log scale)

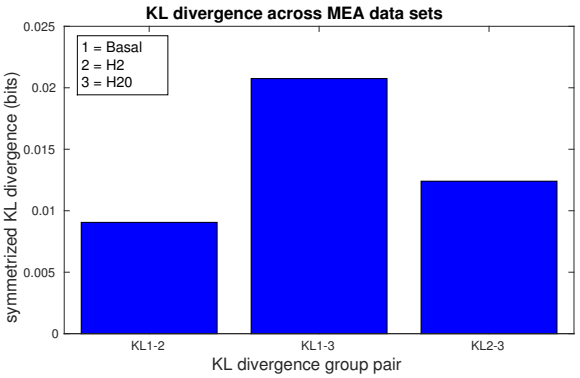


(b) Probability of independent active neurons (log scale)

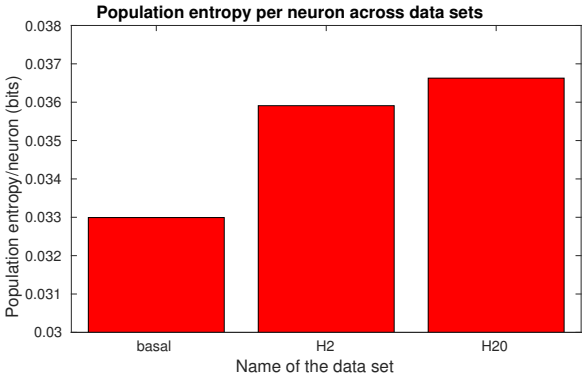
Figure 4.15: Baseline KL divergence measurements

We then want to get the parameters from the population tracking model and plot them. For the sake of concision, we only display the probability of k active neurons $P(K)$ (see Figure 4.15a) and the probability of independent active neurons $P(Xi)$ (see Figure 4.15b). Results from Figure 4.15a suggest that each MEA spikes dataset behaved in a similar fashion: that is, as the number of k active neurons increases (X-axis), the probability of k active neurons P(k) decreases. We also see a drop in the probability of k active neurons around [4.8-5.0] on the log-scale, which corresponds to [120-148] active neurons. Similarly to the synthetic data set results, this suggests that fewer active neurons are much more likely to occur than higher combinations of neurons together. Combinations of 150 and/or more active neurons don't seem

to occur in the neuronal network of interest. As to the results from Figure 4.15b, we sorted all neurons in a descending order according to the mean firing rates from the basal MEA dataset. Results from the two other MEA datasets (H2 and H20) are plotted in agreement with the initial basal firing rates, meaning that the order of neurons did not change from the basal dataset to the two other datasets (in orange and yellow). The first observation to make from Figure 4.15b is that all neurons' mean firing rates seemed to range between -6 and -3 on the log-scale, which corresponds to a probability range of [0.0025,0.0498]. A second observation to make is that neurons' independent probability of firing $p(x_i)$'s behaviour gets spikier and spikier as a function of time (i.e: there is the most spiking in the H20 dataset, followed by the H2 dataset). This suggests that some neurons changed their rate over time, and potentially that neurons with a low rate changed more than neurons with a high rate. Overall, these results represent preliminary evidence to support the hypothesis that remodelling of the neuronal network happened over time.



(a) KL divergence comparison across MEA datasets



(b) Population entropy comparison across MEA datasets

Figure 4.16: KL divergence and Population entropy comparisons across MEA datasets

We now turn to the KL divergence between MEA spikes matrices (see Figure 4.16a) and population entropy per neuron (see Figure 4.16b). First, with respect to the KL divergence graph, we used the symmetrized P(K) approximation version of the KL divergence in order to compare pairs of MEA spikes datasets, as argued previously in the case of the synthetic datasets. Results show that the KL divergence was bigger when comparing the Basal and the H20 datasets than other pairwise comparisons. This result makes sense, since we expect more remodelling to occur in the neuronal network between the baseline dataset and H20 (which is the last recording that was performed) than between the baseline and H2, or between H2 and H20. These results strongly suggest that neurons' behaviour in the dissociated embryonic hippocampal culture does change over time.

Second, with regards to the population entropy results from Figure 4.16b, we used the sum-of-log normals method in order to get the full distribution of patterns of the MEA datasets. To do so, we turned the logarithmic probability of words into a continuous random variable, and used mathematical integration techniques in order to compute the population entropy of each MEA spikes data. More information about this technique can be found in Chapter 3. As we can see, the population entropy per neuron increased as a function of time, with most of the change in entropy

happening between the basal stage and 2 hours after the start of the neurobiological experiment. The entropy of a distribution (see equation (4.5)) is closely related to the level of correlations in the neuronal networks. Therefore, this means that lower levels of population entropy imply that there are more correlations in the network, and higher levels of population entropy imply that there are less correlations in the network. The increase in the population entropy per neuron which is shown on Figure 4.16b suggests that there is a *decorrelating* process over time in the hippocampal embryonic cells. These decorrelations may imply that each cultured neuron's firing behaviour gets more and more independent as a function of time. That being said, one should keep in mind that some of these decorrelating effects may also be due to the recording itself. After completion of the first recording, the culture was taken out of the incubator and placed into the recording setup, where the temperature was different, which may have disturbed the network a little.

4.3 Calcium Imaging data modelling

Finally, we examine the calcium imaging spikes data. As mentioned in Chapter 3, we were interested in comparing the spikes matrices before (Day 1 or "D1") and after (Day 7 or "D7") monocular deprivation across genotypes (Wild-Type and Knock-Out). The spikes data were recorded with and without stimuli. The stimulus condition was referred to as the "stim" condition, and the darkness condition was referred to as the "dark" condition. Table 4.4 summarizes this information.

Table 4.4: Synaptic plasticity experiment conditions

Day	**Description**	**Stimulus condition**	**Description**
Day 1 ('D1')	Pre-Monocular Deprivation	Stimulus ('stim')	a tilted bar was shown to the animal
Day 7 ('D7')	Post-Monocular Deprivation	Darkness ('dark')	the animal was immersed in darkness

Table 4.5: Calcium imaging spikes matrices recording types and order of recording

Condition	**Left Eye**	**Right Eye**	**Order**
BinocPre	**X**	**X**	1
LeftE	**X**	-	2
RightE	-	**X**	3
BinocPost	**X**	**X**	4

As suggests the methodology introduced in Chapter 3, for each pair of conditions "Day"-"Stimulus", spikes data were recorded with both eyes open before ("Binocular anterior" or "BinocPre")and after stimulation ("Binocular posterior" or "BinocPost"), with the left eye open only ("LeftE") and with the right eye open only ("RightE").

Table 4.6: Syngap datasets information

Syngap dataset (SG)	Genotype	min. recorded N	max. recorded N
SG 259	Wild-Type	97	97
SG 288	Wild-Type	95	95
SG 252	Knock-Out	93	93
SG 291	Knock-Out	101	101
SG 63	Knock-Out	10	134

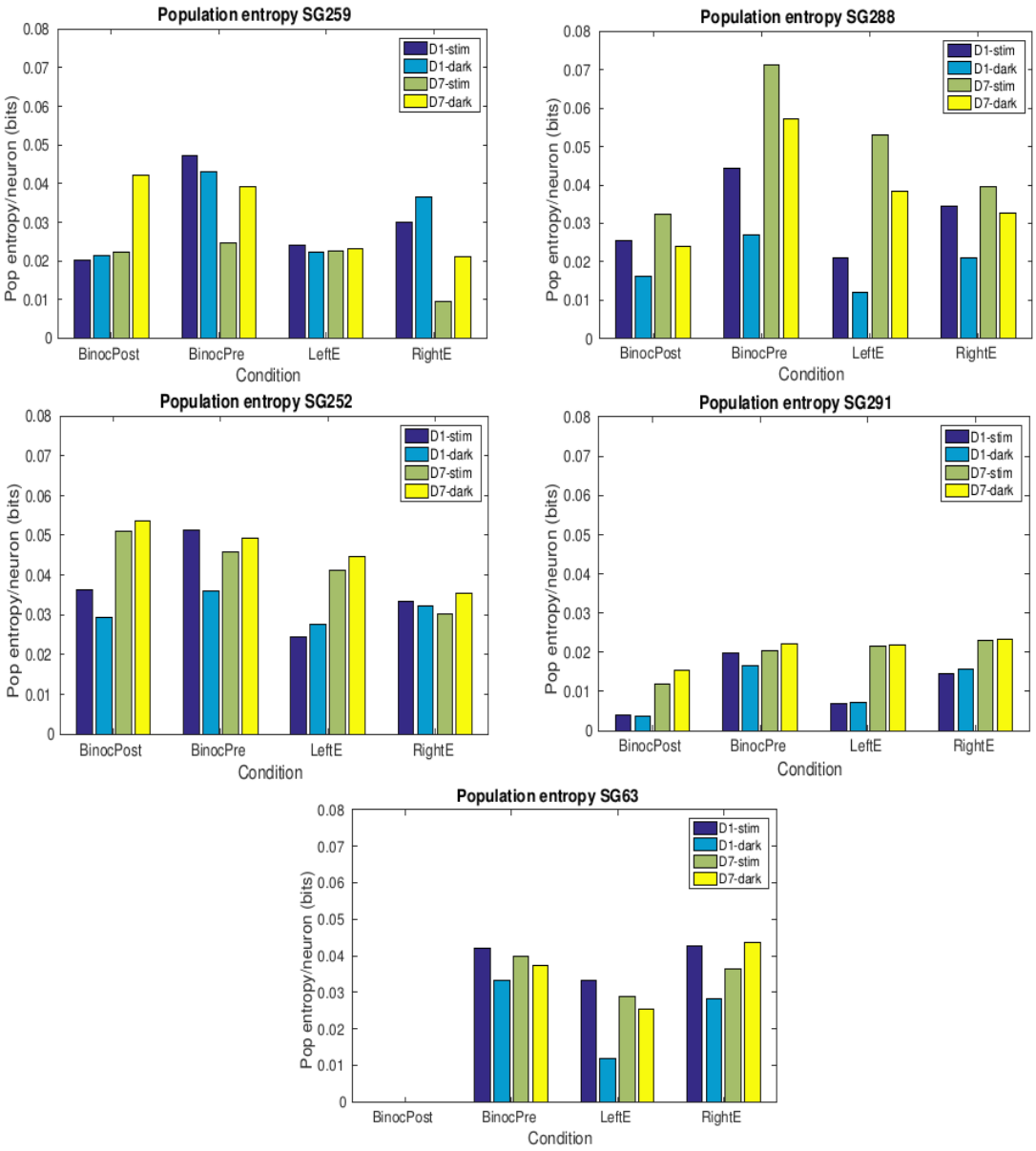


Figure 4.17: Population entropy across all conditions and Syngaps. Each condition refers to those defined in Table 4.4 and Table 4.5

Table 4.5 summarizes this information and the order in which the spikes were recorded. In the context of this Master's project, we were provided with spikes data recorded

from five different animals. Each animal was embedded in the "Wild-Type" or the "Knock-out" condition. More information about each dataset is shown in Table 4.6 For each Syngap dataset, spikes from the four conditions introduced in Table 4.5 were recorded for each pair "Day"-"Stimulus", which lead to a total of 16 recorded spikes matrices per Syngap. The population entropy of each spikes matrix was computed using the sum-of-log normals method, as introduced in Chapter 3. For the sake of completeness, we included the full list of entropy results on Figure 4.17.

As we can see, there is a great deal of variability in the entropy data. Also note the fact that the entropy per neuron of Syngap 63 BinocPost was always remarkably lower than the population entropies computed at other conditions and in other Syngap datasets. The population per entropy at all condition pairs "Day"-"Stimulus" was evaluated at 8.0725×10^{-4} bits, which is about two orders of magnitude smaller than the usual entropy per neuron. This difference can partially be explained by the fact that the Syngap 63 dataset did not use the same number of neurons for all its spike recordings (see Table 4.6). Indeed, all conditions recorded spikes at more than 100 neurons except the BinocPost condition, which only recorded 10 neurons. Because of this difference and due to the fact that we only had 2 wild-type Syngap datasets (against 3 knock-out Syngap datasets), we decided to reject SG63 from all further analyses.

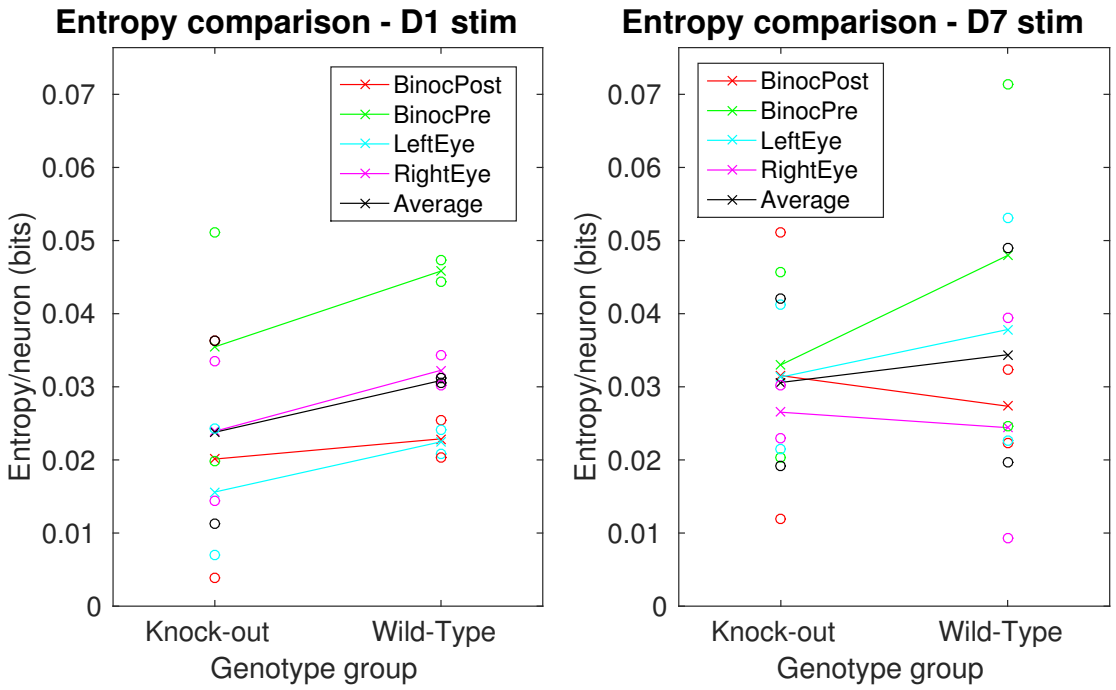


Figure 4.18: Mean entropies (crosses) across genotypes in the Stimulus condition with individual data (circles)

We now wish to group entropy results together in order to compare the two genotypes of interest: the "Knock-Out" genotype and the "Wild-Type" genotype. As suggests Table 4.6, we averaged SG 259 and SG 288 from the Wild-Type genotype, and we averaged SG 252 and SG 291 from the Knock-Out genotype. As was mentioned before, SG 63 was not included in our analyses. Figure 4.18 displays mean

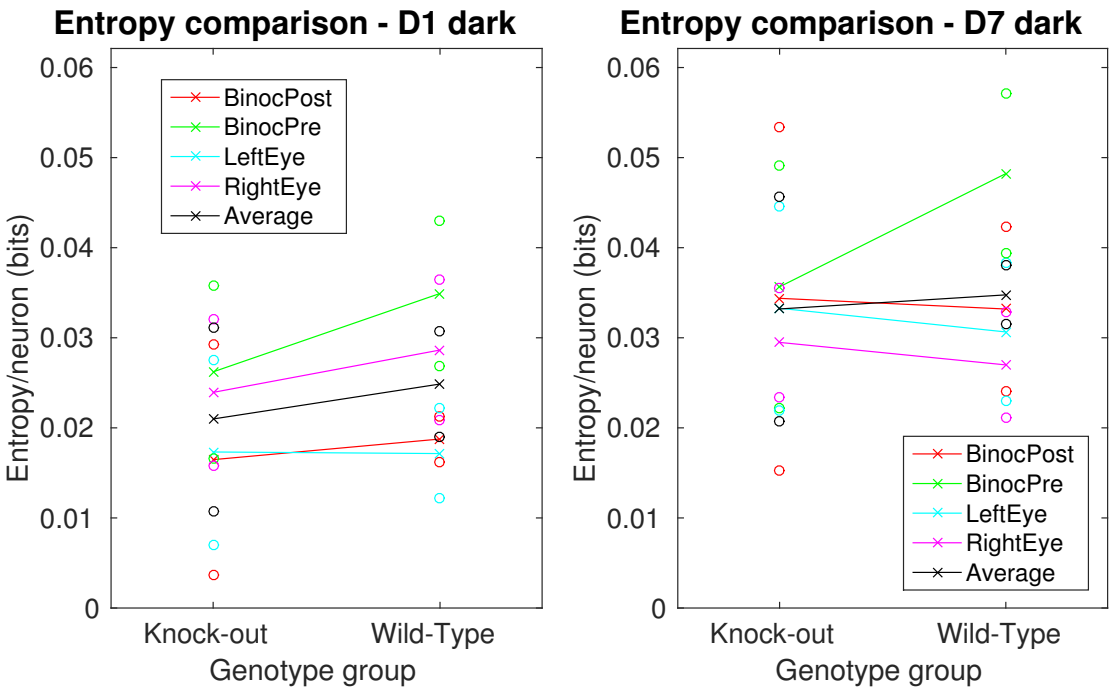
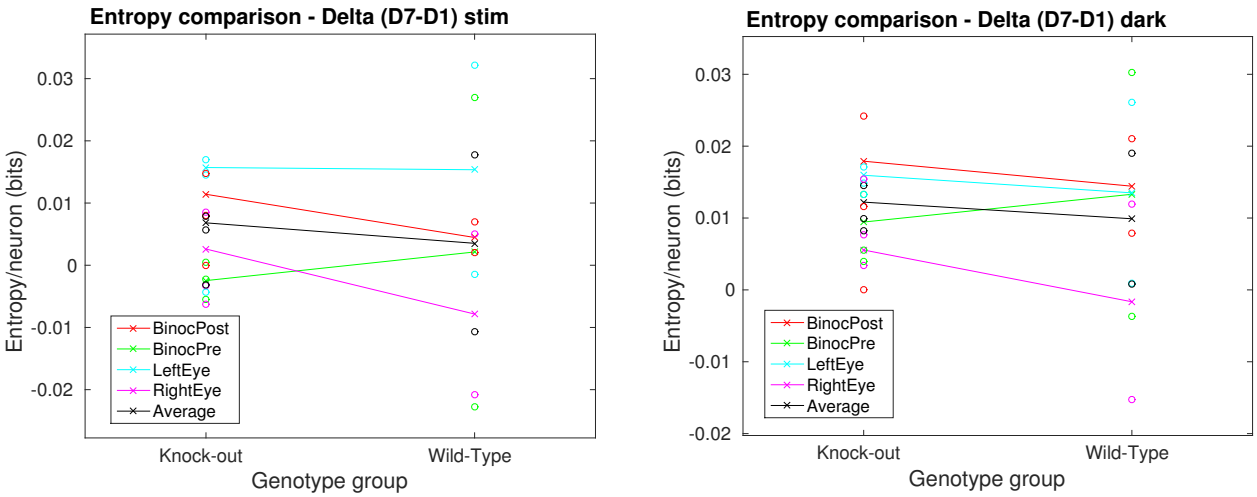


Figure 4.19: Mean entropies (crosses) across genotypes in the Darkness condition with individual data (circles)

entropies in the D1-stim and D7-stim conditions across genotypes, and Figure 4.19 displays mean entropies in the D1-dark and D7-dark conditions across genotypes. Crosses indicate the mean population entropies, and circles indicate individual animals.

The first observation was that the Wild-Type genotype had a higher entropy than the Knock-Out genotype at D1, both in the stim and the dark conditions. The second observation was that entropies at D7 (stim and dark) were overall higher than entropies at D1. This result is consistent across genotypes, both for mean entropies and for individual animals. Another result was that the Binoc Pre condition had entropies that were consistently higher than entropies from other conditions, both in the stim and the dark conditions. This suggests that at any point in time, the entropy was higher prior to the independent stimulation of either eye (see Chapter 3 for the specific methodology used). Finally, one last result was that the mean entropies seemed to uniformly shift upwards in the Wild-Type genotype, whereas mean entropies from the Knock-out genotype tend to cluster around 0.030 in the stim condition, and around 0.033 in the dark condition.

Figure 4.20 displays delta entropy results in the stim (Figure 4.20a) and dark condition (Figure 4.20b). Each delta computed the increase (positive value) or decrease (negative value) in entropy from D1 to D7. Once again, crosses indicate mean entropies and circles indicate individual animals. Overall, we see that most entropy deltas across all conditions were positive, which clearly shows that there was indeed an entropy increase from Day 1 to Day 7, consistent across the dark and stim conditions. The second result was that the variability in entropy increase was greater for the Wild-Type genotype than it was in the Knock-Out condition (individual data



(a) Delta entropy comparison - Stimulus

(b) Delta entropy comparison - Darkness

Figure 4.20: Entropy delta comparison across stimulus conditions. Positive values indicate an entropy increase from D1 to D7, whereas negative values indicate an entropy decrease.

were much more spread in the Wild-Type condition than they were in the Knock-Out condition). This suggests that the entropy tended to get fixed at a single value in the Knock-Out condition, whereas neurons from the Wild-Type condition tended to show more flexibility in their firing behaviour.

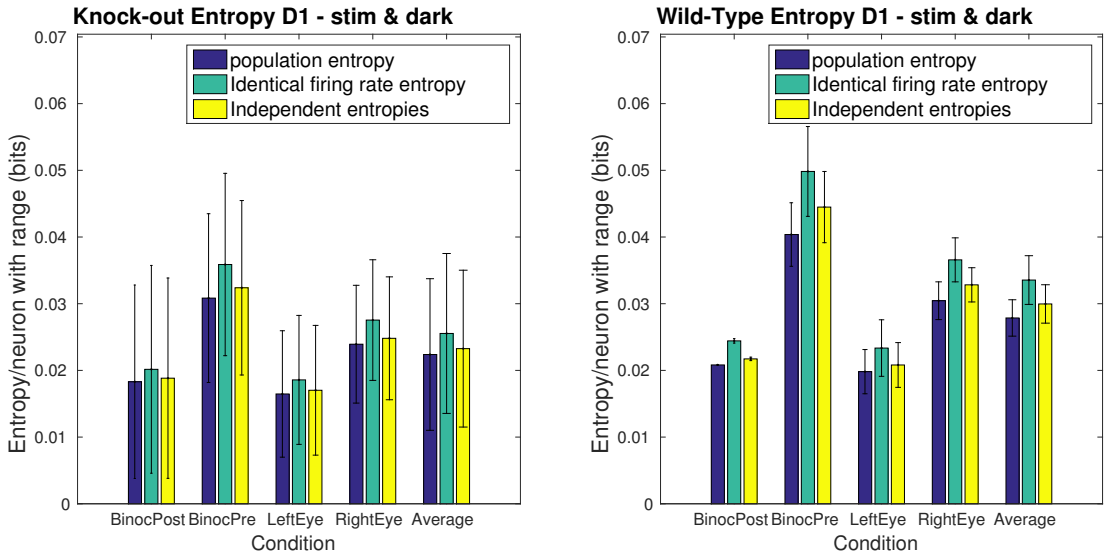


Figure 4.21: Absolute entropies with range across genotypes at Day 1. The entropy per neuron was averaged across stimulus and darkness conditions.

Figures 4.21 and 4.22 display the absolute entropy means (with range) across genotypes at Day 1 and Day 7. The idea here was to compare the population entropy per neuron computed using the sum-of-log normals method from [1] with two alternative types of entropies computed on a shuffled version of the raw binary

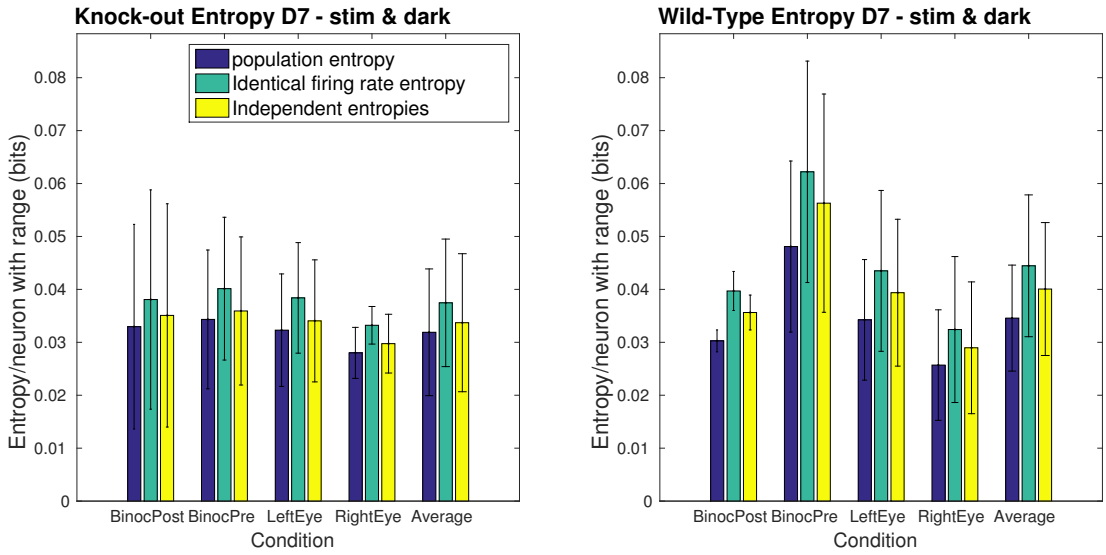


Figure 4.22: Absolute entropies with range across genotypes at Day 7. The entropy per neuron was averaged across stimulus and darkness conditions.

spikes. The first entropy type, the "identical firing rate entropy" refers to an entropy which only takes into account the mean of the vector of independent firing rates $p(x_i)$, and assumes homogeneity across all neurons. In mathematical terms, it can be rewritten as follows:

$$H_{ifr} = -N \times \left[\bar{p}(x).log_2(\bar{p}(x)) + (1-\bar{p}(x)).log_2(1-\bar{p}(x)) \right] \tag{4.10}$$

where H_{ifr} refers to the "identical firing rate" entropy, N refers to the total number of neurons in the network, and $\bar{p}(x_i)$ refers to the mean of the vector of neurons' independent firing rates $p(x_i)$ (computed by estimating parameters with the population-tracking model).

The second type of alternative entropy can be referred to as the "independent entropy" type. It quite simply computes the entropy of the neuronal network by using the parameter $p(x_i)$ and assuming full independence between all neurons. Mathematically, it can be rewritten as:

$$H_{ind} = -\sum_i \left[p(x_i).log_2(p(x_i)) + (1-p(x_i)).log_2(1-p(x_i)) \right] \tag{4.11}$$

Note that the two entropy types take into account $p(x_i)$, as well as $(1-p(x_i))$ which corresponds to the probability of a neuron x_i being inactive. The idea is that the entropy of the neuronal network needs to take account the probabilities of independent neurons being *both* active and inactive. Importantly, we computed the two alternative types of entropies on shuffled versions of the initial Calcium binary matrices that we were given. Shuffling the rows of the matrices is usually helpful in order to break many of the temporal correlations in the network. This method is often a good way of testing the effect of correlations in the raw data.

Results from Figures 4.21 and 4.22 mainly suggest that the identical firing rate entropy H_{ifr} is the highest out of the three types of entropy, followed by the independent entropy H_{ind}. We see that the population entropy computed using the

population-tracking model is always estimated as being the smallest of all. This observation is valid across all conditions, for both Day 1 and Day 7. This result can be explained by the fact that higher entropies usually indicate more independence between neurons. A key aspect of information theory is that the highest entropy is always reached in the case of a uniform distribution. Therefore, when all the neurons have the same mean firing rate, we expect them to yield the highest entropy value. This is indeed what we get on Figures 4.21 and 4.22.

As regards the independent entropy term: H_{ind}, we see that its value is always higher than the population entropy, but lower than the identical firing rate entropy H_{ifr}. This result can be explained by the fact that this entropy term only takes into account the mean firing rate of each independent neuron (represented as the vector $p(x_i)$, but does not care about the full underlying probability distribution which computes the probability of independent words {x}, which the population entropy does. Since the two other types of entropies were computed on the shuffled dataset, and were both consistently higher than the population entropy, it is safe to assume that there is an effect of correlation in the raw data.

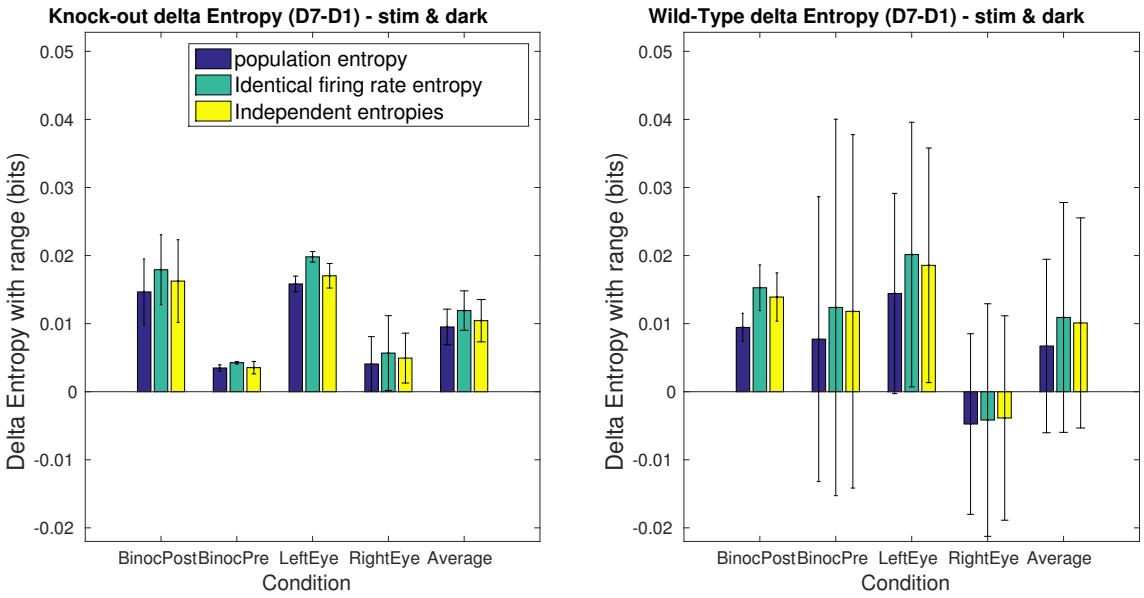


Figure 4.23: Delta entropy (increase or decrease in entropy) per neuron from Day 1 to Day 7 averaged across the stimulus and darkness conditions

A few other observations can also be made. First, we see that the absolute entropies are overall higher in the Wild-Type genotype than they are in the Knock-out genotype. This result suggests that there are more correlations in the data in the Knock-Out genotype than in the Wild-Type genotype. Second, the "BinocPre" condition, which was one of our experimental control, seems to consistently have a higher entropy than the other conditions. This result was there both at D1 and at D7. Finally, another interesting result is that the left eye condition appeared to have a *lower* entropy than the right eye condition at D1 (see Figure 4.21), but had a *higher* entropy than the right eye condition at D7 (see Figure 4.22).

Figure 4.23 investigates the delta entropy (increase or decrease) from D1 to D7

averaged across the stim and dark conditions. Both Knock-Out genotype (to the left) and Wild-Type genotype (to the right) are investigated here. Results were as follows. First, we see again that the entropy mostly increased from D1 to D7 in both genotypes, *except* for the “Right Eye" condition in the Wild-Type genotype, in which we see a mean *decrease* of entropy from D1 to D7, which suggests that there was a correlating effect in the data from D1 to D7. Another result was the variability difference across genotypes. The range information clearly shows that there was a similar entropy increase across Syngap datasets in the Knock-Out genotype, but strong variability in the entropy change in the Wild-Type genotype. Overall, there was the most entropy increase in the Left Eye condition, followed by the BinocPost condition. There was a very slight increase in entropy in the Knock-Out BinocPre condition, whereas there were much more extreme changes in entropy in the Wild-Type BinocPre condition.

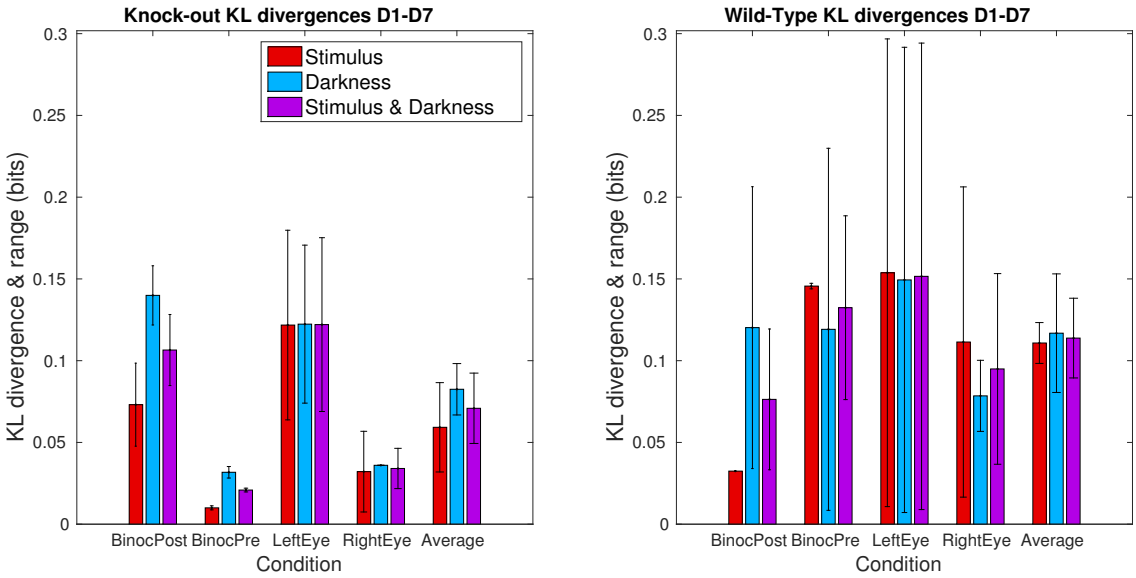


Figure 4.24: P(K) symmetrized KL divergence between Day 1 and Day 7 across genotypes

We now turn to the KL divergence results in the Calcium imaging datasets. Figure 4.24 displays the KL divergence between D1 and D7 across genotypes for each condition in the stimulus, darkness and averaged conditions. As argued previously, we decided to use the symmetrized P(K) approximation as our KL-divergence estimate (see section 4.1). The first observation was that the BinocPre and RightEye KL divergence were greater in the Wild-Type genotype than they were in the Knock-Out genotype. which suggests that there may not have been a significant neuronal remodelling in the network as one may have expected in the Right Eye. The second observation was that once again, the variability of KL divergences was much higher in the Wild-Type genotype than it was in the Knock-Out genotype (see range error bars on the graph). Finally, we see that the KL estimate was consistently higher in the darkness condition *only* in the BinocPost condition. Other results were not consistent enough to be reported here.

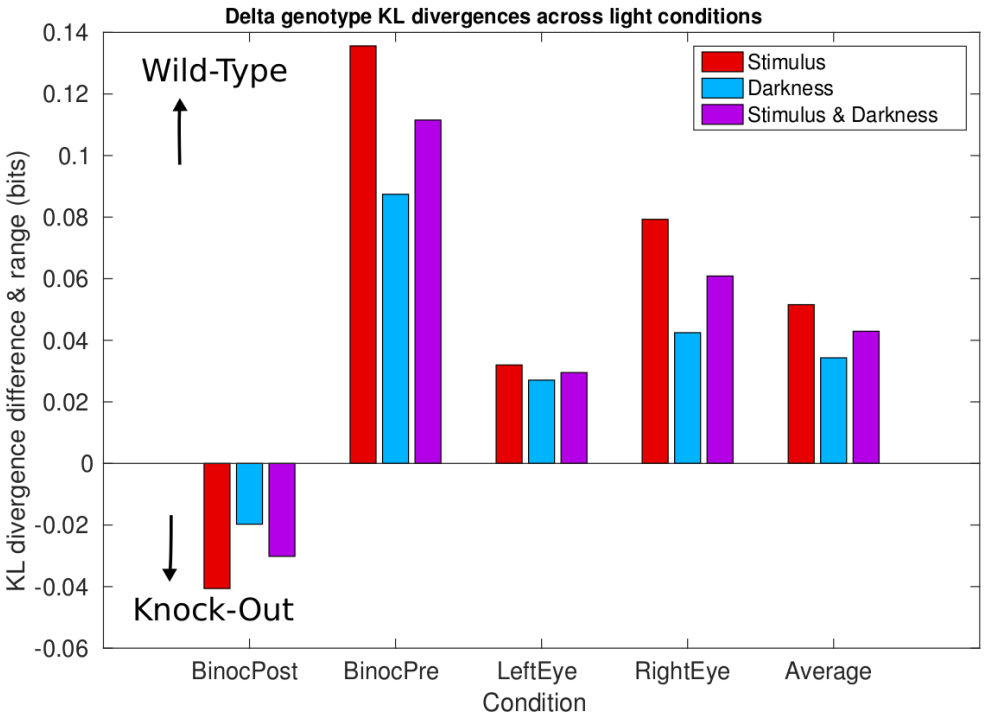


Figure 4.25: Delta KL-divergence D1-D7 across genotypes and stimulus conditions. Arrows indicate in which genotype the KL divergence was greater.

Last but not least, we investigated the delta genotype of the KL divergences in order to see how the two genotypes differed from one another. The most salient result was that KL divergences differed the most in the stimulus condition (as opposed to the darkness condition). This means that the presence of the stimulus yielded a bigger KL divergence difference across genotypes. Overall, the KL divergence was greater from D1 to D7 in the Wild-type than in the Knock-Out *except* for the BinocPost condition, in which the difference was greater in the Knock-Out genotype. Once again, we can see that the KL divergence was the greatest in the Wild-Type than in the Knock-Out genotype in both the BinocPre and RightEye conditions.

Chapter 5

Discussion and Limitations

The goal of the Master's project was to implement a mathematical model of neuronal population activity on calcium imaging data collected in the context of a neurobiological experiment. Our choice of model was the population-tracking model [1]. Data modelling was divided into three main stages. The two first stages represented controls and the third stage was more exploratory in nature, since we attempted to model new experimental data. In the first stage, we modelled data that was synthetically-generated by a Dichotomized Gaussian (DG) model [20] in order to carry out baseline tests. In the second stage, we modelled controlled experimental data from a multielectrode array experiment conducted using cultures of rat hippocampal neurons. Finally, in the third stage, we modelled calcium imaging data which were recorded in the context of a new monocular deprivation (MD) experiment performed on mice.

Results pertaining to the synthetic data baseline tests were as follows. First, regarding the main parameters of the population-tracking model, we saw that $p(k)$ tended to decrease as the number of k active neurons in the network increased. $p(x_i)$ successfully estimated the probability of firing of each neuron in the network, and $p(x_i|k)$ results suggested that there was an overall tendency for neurons to have a higher probability of firing as the number of k active neurons increased. Second, our time steps analysis revealed that as T increased, the pattern probability $p(\{x\})$ estimated by the model approached the empirical "true" probability P_{true} (computed using exhaustive histogramming methods). In addition, the empirically computed entropy also approached its "true" value (computed using an extrapolation method) as T increased. A good entropy estimate was found for T= 5,000 timesteps. Finally, in the case of homogeneous synthetic datasets, the theoretically-computed entropy was shown to get closer to the "true" entropy value faster (i.e: with a lower number of time steps) than the estimated empirical entropy.

In this dissertation, we also compared two choices of methods which estimated the Kullback-Leibler divergence between any two spikes datasets. Results from our tests indicated that the KL divergence based on the model estimate $p(k)$ was better than the full distribution heuristic computation introduced in Chapter 3 (see Algorithm 1). Three main reasons to choose the $p(k)$ approximation KL divergence method are the following. First, when estimating the divergence of two baseline synthetic datasets, the $p(k)$ method showed that it could be considered a sufficient

estimate without having to iterate over the full distributions of both datasets' words probabilities. Second, the baseline KL divergence between half-half randomly sampled synthetic datasets was lower by 2 orders of magnitude with respect to the heuristic D_{KL}. Third, the heuristic D_{KL} was not able to correctly estimate the divergence between spikes datasets with significantly different numbers of timesteps, whereas the $p(k)$ could do so, as the latter method relied on $p(k)$ which had the same dimension across datasets regardless of the number of time steps of the binary matrices being compared.

As regards multielectrode array data results, we found that there was a *decorrelating* effect over time, as suggested population entropy results. This suggests that the rat hippocampal neurons underwent neuronal remodelling as time went by in the experiment, and became more independent with respect to their firing behavior (there was more "richness" in the spikes behaviour over time). In addition, KL divergence estimates also supported this result and showed that the difference between the basal dataset (baseline) and the H20 dataset (after 20 hours) was greater than differences between other pairs of spikes datasets. This result could also be seen more explicitly on Figure 4.14, where we saw an emerging peak in the log frequency distribution for patterns with lower log_2 probability. These results are in agreement with Panas et al. (2015) [2], which showed that small subnetworks of highly active neurons accounted for the stability of neuronal networks while influencing the synaptic remodelling of most other "sloppy" neurons in the network, eventually leading to further neuronal connectivity and firing dynamics changes.

Finally, results from the modelling of calcium imaging data were as follows. We first found a higher mean entropy in the Wild-Type genotype than in the heterozygous Knock-Out genotype (with one of the SynGap genes knocked-out), which would imply higher levels of correlations between neurons in the Knock-out genotype. This suggests that neuronal remodelling in Knock-Out mice might have failed to happen as it normally would, perhaps due to the fact that one copy of the SynGap gene was knocked out (heterozygous genotype). We also noticed that there was more entropy variability (when looking at individual data) in the Wild-Type genotype than in the Knock-Out genotype, and entropy means were more clustered in the Knock-Out genotype as well.

In the context of the calcium imaging experiment, we investigated neuronal dynamics *before* (at day 1) and *after* monocular deprivation (at day 7), both with visual stimulation and in darkness. Entropy results showed that there was an overall increase in entropy from day 1 to day 7 suggesting a *decorrelating* effect over time after monocular deprivation. This suggests that neuronal remodelling lead to more independence in the neuronal network following MD. Interestingly, the right eye, which was blindfolded, demonstrated an overall *decrease* in entropy in the Wild-Type genotype, but not in the Knock-Out genotype. This is in agreement with experiments from Wiesel & Hubel (1963) [3] who showed that synapses from neuronal networks related to the blindfolded eye were weakened, whereas synapses from neuronal networks related to the *other* eye tended to get strengthened over time in a healthy animal. This effect seems to have been weaker in the animals with one of the SynGap gene knocked out, as one would have expected, which supports the experimental data analysis results (see Figure 5.1). Indeed, we notice that the

ocular dominance index decreases more in the Wild-Type genotype than it does in the Knock-Out genotype (there is a steeper decreasing slope in the Wild-Type genotype). Combined together, these results represent preliminary evidence to support our initial expectation that the heterozygous genotype may prevent spontaneous synaptic plasticity mechanisms to induce natural neuronal remodelling in networks of neurons of Knock-Out animals.

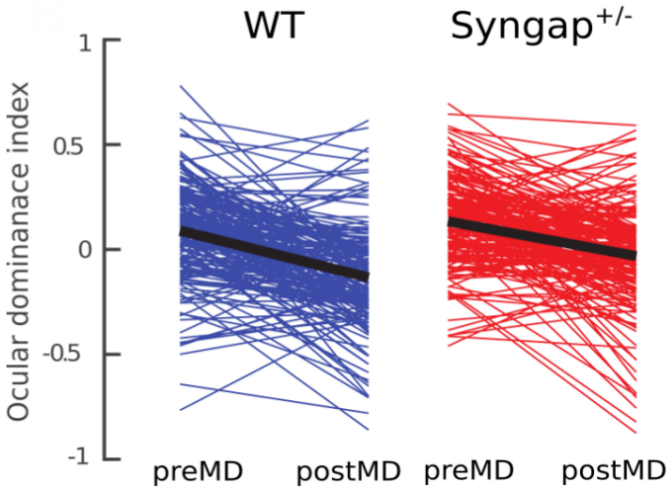


Figure 5.1: Ocular dominance index (ODI) results across genotypes in the monocular deprivation experiment. *Unpublished figure.*[1]

With respect to the KL divergence computations, we saw that more changes happened from Day 1 to Day 7 in the Wild-Type genotype overall. These results provide further evidence to support the claim that there was less neuronal remodelling in the Knock-Out genotype than there was in the Wild-Type genotype. As to the comparison of entropy computations (population entropy on raw data, and both identical firing rate entropy and independent entropy on shuffled data), our analysis showed that the identical firing rate entropy and the independent entropy were consistently higher than the population entropy. This result suggests that there was an effect of correlations in the experimental spikes datasets of interest, as shuffling the datasets seemed to be enough to increase the entropy.

As regards the comparison of the stimulus conditions, we noticed that KL divergences were greater from Day 1 to Day 7 when there was visual stimulation (as opposed to darkness), which suggests that visual stimulation induced more neuronal dynamics changes than darkness did. This result supports past findings from Pakan et al. (2016) [56], which showed that excitatory neurons from rats were more active in visual stimulation than in darkness during locomotion. Interestingly, we noticed that there were smaller D_{KL} differences between *pre*- and *post*-monocular deprivation in the right eye of the Knock Out animals than in the right eye of the Wild-Type animals. Once again, this result supports our initial hypothesis that the heterozygous genotype may prevent natural neuronal remodelling mechanisms from happening spontaneously as they would in Wild-Type animals.

[1]This figure was provided by Dr. Rochefort from the Integrative Physiology department at the University of Edinburgh.

However, the above results and interpretations should be taken with caution and a number of limitations should be considered. First, as suggests results displayed on Figure 5.2, a neurobiological data analysis of the experimental data distinguished between two different behavioural conditions: the "locomotion" condition, and the "stationary" condition. These two conditions were not differentiated in the data that we analyzed, as mice were moving *freely*. More specifically, the comparison between the "locomotion" and the "stationary" conditions was the main object of Pakan et al. (2016) [56], and their results showed that the behavioural state (the mouse is running *versus* the mouse is standing) *does* modulate cortical responses to sensory stimuli, particularly in the "visual stimulation" condition (see Figure 5.2).

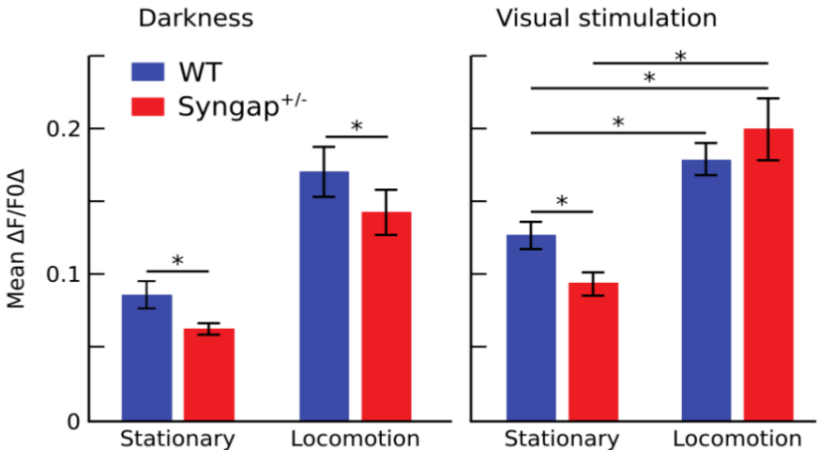


Figure 5.2: Means from experimental calcium imaging data recordings across locomotion and stationary. *Unpublished figure.*[2]

Second, one should note that our main limitation was the fact that we were only able to analyze the data from four animals (two of them were nested in the Knock-Out genotype, and the two others were nested in the Wild-Type genotype as was introduced in Table 4.6). This choice of sample size is certainly *not* enough to infer any statistical significance and may lead to biased estimations and interpretations. Moreover, we could not have a clear idea of what the true mean and standard deviation were for each genotype , since having $n = 2$ could only enable us to get a range and median estimates for the Wild-Type and the Knock-Out genotype. More data points should be collected in order to make statistically significant claims with regards to the interpretation of our results. However, the current work can still provide *preliminary* results which can give us an idea of the overall trend in entropy and Kullback-Leibler divergence comparisons.

Third, other experimental limitations ought to be taken into account as well. One is that there may have been confounding variables which biased some of our results. For example, it may be the case that fatigue impacted the firing behaviour of the V1 mice neurons, particularly in the BinocPost condition (which was the last recorded one, as explicitly referenced in Table 4.5). Another experimental limitation is that biomedically-speaking, it would be necessary to compare our experimental

[2]This figure was provided by Dr. Rochefort from the Integrative Physiology department at the University of Edinburgh.

mice to a surrogate model in order to get baseline estimations and have more reliable Kullback-Leibler divergence estimates. Finally, one last experimental concern is that we did not control for potential contamination in the calcium imaging data, which could imply significant neuronal firing behaviour differences and would lead to problematic biases in our data analysis.

Fourth, with regards to our calcium imaging data analysis, it is not clear to which degree the observed entropy changes are due to firing rates alone. It is worth reiterating some of the limitations of calcium imaging as a neuroimaging recording technique. Mainly, calcium sensors used in two-photon calcium imaging are known to be slow sensors with a low sampling rate around 50 Hz [18]. In addition, calcium imaging data pre-processing involves denoising, deconvolution, demixing and spike train estimation through the use of physiological models, as described in Chapter 3 [21], [22]. This causes a number of issues, among which the most concerning one is the fact that spike inference represents a major source of error and variability since recording conditions may vary. This implies that the spikes inference quality may differ substantially, even for the same animal.

Fifth, many pre-processing assumptions should be born in mind when inferring from binary spikes data (in the form of TxN binary spikes matrices). The most important one is the issue of time-binning. For instance, in our multielectrode array recordings, we used time bins of 10 ms. Ideally, one should investigate the role of time-binning in the estimation of population entropies and Kullback-Leibler divergences, as different choices of time bin durations may lead to significant changes in the analysis. Furthermore, with respect to the multielectrode array recordings, it is worth mentioning that the data that we analyzed in Chapter 4 was *clustered* data (see Table 4.3). Clustering was performed using herding spikes methods [49], as described in Chapter 3. Although useful in the investigation of populations of neurons because it significantly reduces the number of neurons, which in turn leads to faster computational times and easier interpretations, it also represents a potential danger since clusters of channels are assumed to describe the firing behaviours of independent neurons, which may not actually be true biologically-speaking (i.e: there may be more than one single neuron in each cluster).

We now turn to the evaluation of our choice of mathematical model. As mentioned in the introduction, the population-tracking model [1] was advantageous for many reasons. Most importantly, it could fit large populations of neurons (of many hundreds) whereas other traditional models from the Computational Neuroscience literature, such as the pairwise maximum entropy model, could only be fit to populations of about 10 neurons. Another main advantage was that parameters could be estimated with low computational time complexity (the model used N^2 parameters) and parameter estimation converged with a reasonable number of time steps. One third main advantage was that it was possible to get an approximation of the full pattern probability distribution using the low-dimensional sum-of-log-normals approximation method while being computationally tractable. According to O'Donnell et al. (2016) [1], reasons for choosing the population-tracking model over similar models such as the population-coupling model [57], [58] were that it could estimate pattern probabilities of single words of neural activity, it could estimate functions of the entire pattern probability distribution, such as the Shan-

non entropy or the Kullback-Leibler divergence, and it assumed a non-linear relationship between individual neurons' firing rate and population rate. According to the author, these advantages made the population-tracking model a more suitable choice of model in the context of neurobiological experiments.

Nevertheless, despite offering many desirable features, the model also had a few limitations, which should be addressed as well. The first and most important one was that it was often outperformed by other mathematical models for small numbers of neurons [1]. A second limitation was that it did not account for the full pairwise correlation structure, as opposed to the maximum pairwise entropy model, which could capture more than 90% of the neuronal networks' correlations [8]. Finally, three other limitations were that temporal correlations were being ignored (correlations between time steps) in the estimation of the parameters, parameters were hard to interpret from a biological perspective, and the low-dimensional sum-of-log-normals estimation of the full pattern probability distribution may have been skewed for increasing numbers of k active neurons in the network, as suggests the shift in log pattern probability displayed on Figure 3.7b.

Finally, with respect to the implementation of the Kullback-Leibler divergence in the context of cognitive neuroscience experiments, researchers have debated and often disagreed on the right use and interpretation of the KL divergence. For instance, Berkes et al. (2011) [54] made use of the KL divergence to compare stimulus-evoked (naturalistic visual stimuli) and spontaneous neural activity in the context of a visual task experiment performed on ferrets. The aim of the experiment was mainly to investigate a Bayesian statistical model of sensory cortical processing, by studying the degree of mismatch (defined as "divergence") between the average posterior (P(features|input,model)) and the prior (P(features|model)). Their main interpretation of the results was that internal cognitive models progressively adapted to the statistics of natural stimuli at the neural level. In the context of a research publication, Okun et al. (2012) [57] disagreed with this claim, and argued that their results did *not* represent evidence that a probabilistic model of the environment was learned. According to them, the results suggested instead that the two visual conditions being studied had similar statistical properties of multi-neuron firing patterns, specifically related to overall population firing rates. More research is needed to disentangle between the different uses and appropriate interpretations of Kullback-Leibler divergence estimates in the context of neurobiological experiments.

Chapter 6

Conclusion

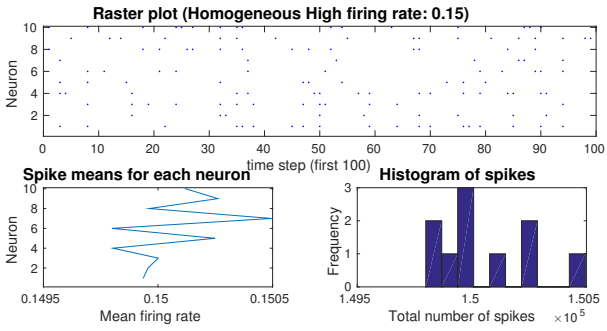
This Master's project investigated the modelling of synthetically-generated spikes data, multielectrode array data and calcium imaging data using a novel mathematical model: the population-tracking model [1]. Synthetic data baseline tests revealed that it was a reliable and fairly accurate choice of model with a number of desirable features. KL divergence methods comparisons showed that the population-rate ($p(k)$) approximation method was better than the heuristic divergence method for the modelling of large populations of neurons. Multielectrode array data analyses were in agreement with results from past research [2], and preliminary results from the calcium imaging experiment also supported past research on long-term synaptic plasticity [3], [27].

Nevertheless, with respect to the multielectrode array experiment, future research could investigate how time binning may impact the analysis of spikes datasets. It would also be of interest to compare *clustered* and *unclustered* spikes data in order to understand how clustering effects may impact data analysis in MEA experiments. As for the synaptic plasticity experiment, more animals are needed to infer about the statistical significance of neuronal remodelling effects in the context of our long-term synaptic plasticity experiment. Future research could include a surrogate model, examine potential differences between the "locomotion" and the "stationary" behavioural conditions, and study the effects of fatigue and calcium imaging data contamination, in the hope of further understanding the neuronal dynamics of Autistic Spectrum Disorder (ASD) and other fascinating genetic intellectual disabilities.

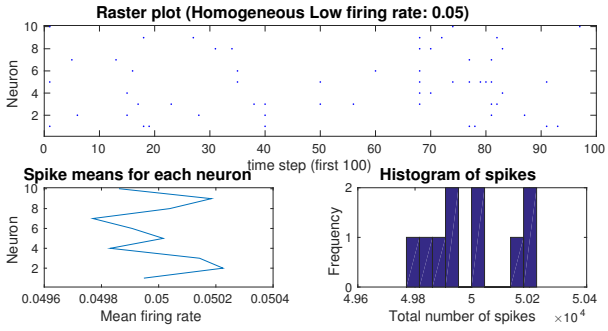
Appendix A

Complementary results

A.1 Complementary synthetic data baseline plots

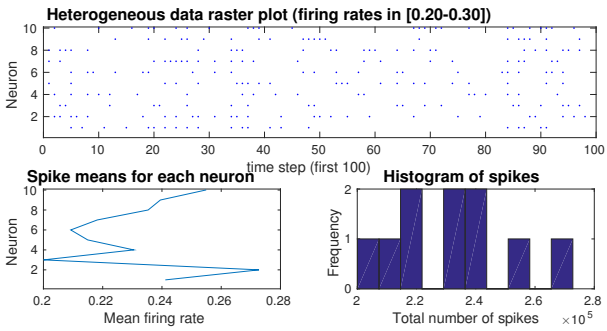


(a) Homogeneous High spikes data



(b) Homogeneous Low spikes data

Figure A.1: Homogeneous data sets properties



(a) Heterogeneous Uniform spikes data

(b) Heterogeneous Non-uniform spikes data

Figure A.2: Investigation of the heterogeneous data sets properties

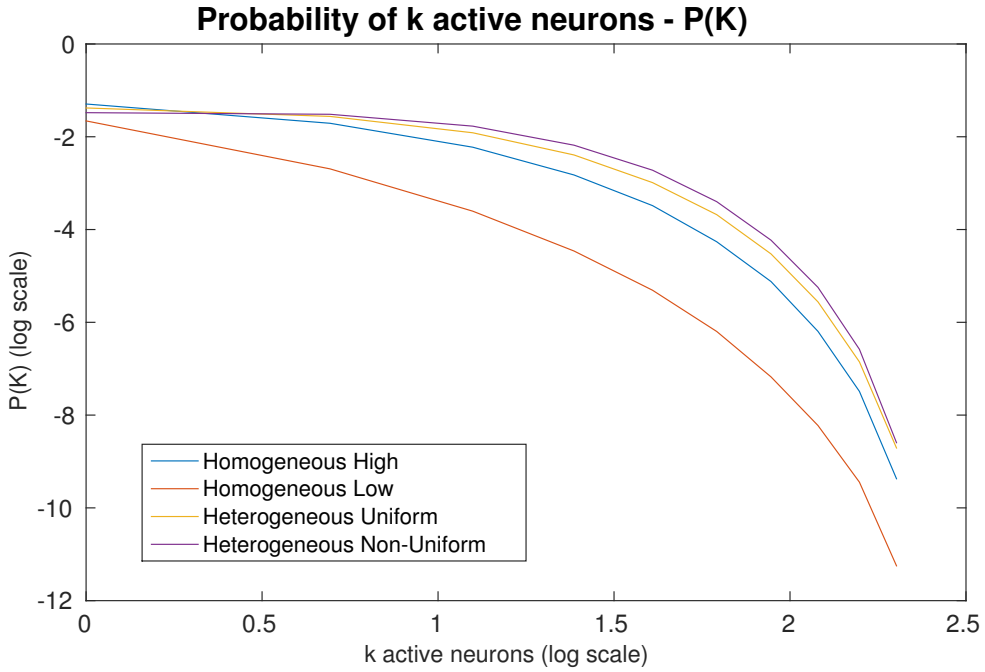


Figure A.3: Probability of k active neurons across datasets

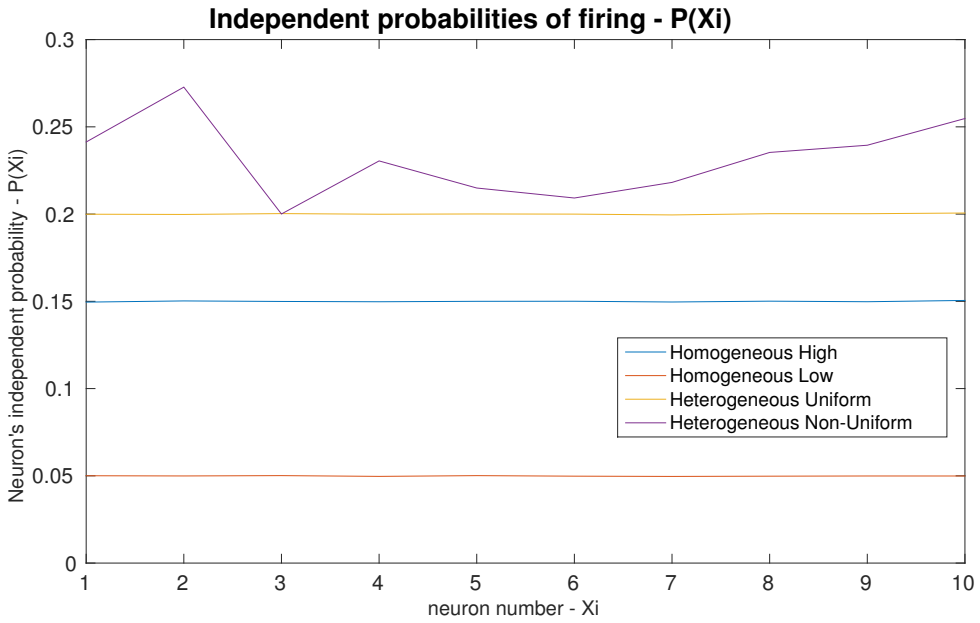


Figure A.4: Probability of independent neurons being active

A.2 Complementary synthetic data time steps results

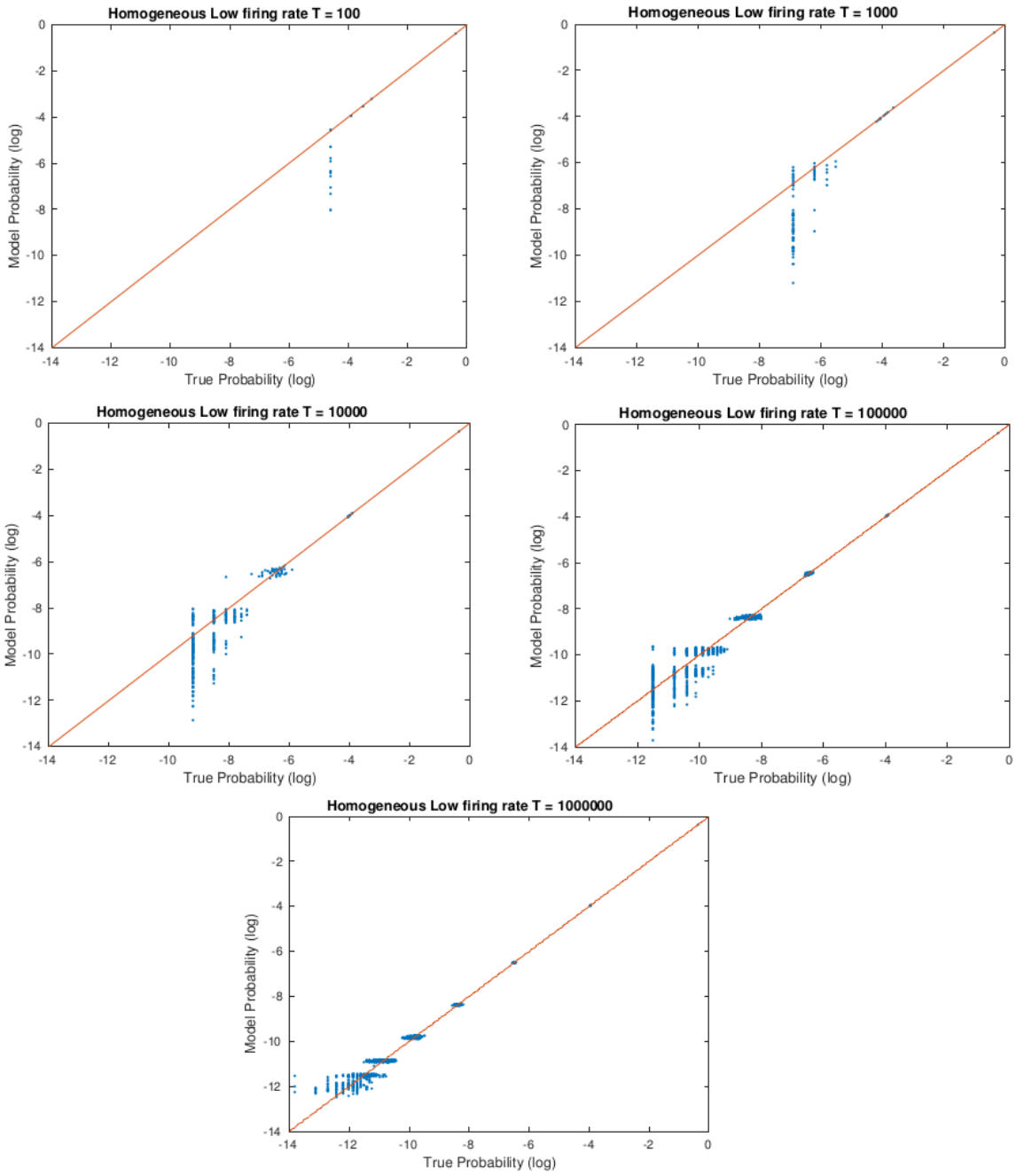


Figure A.5: Full evolution of the Homogeneous Low model probability when compared to the true probability over time steps.

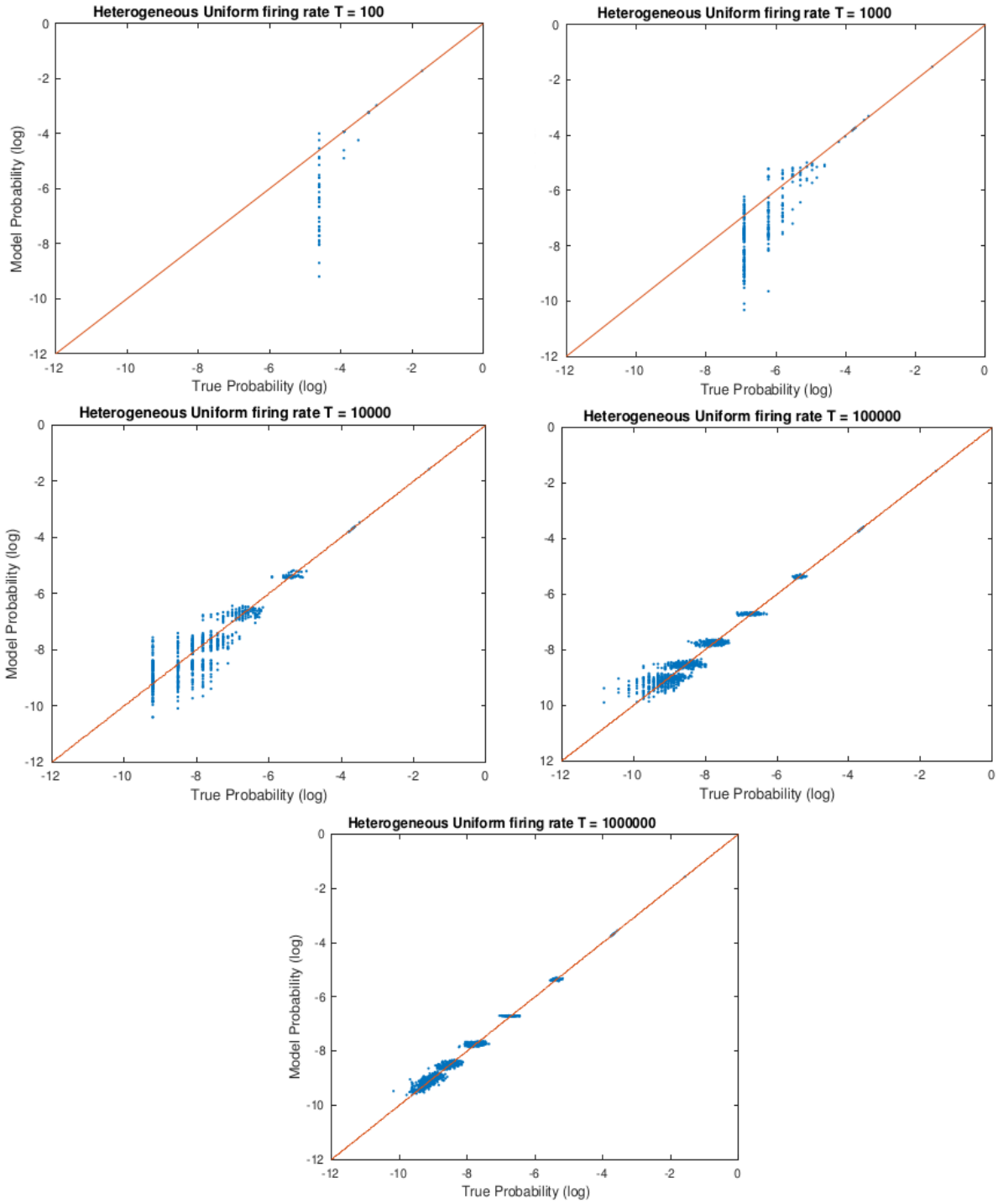


Figure A.6: Full evolution of the Heterogeneous Uniform model probability when compared to the true probability over time steps.

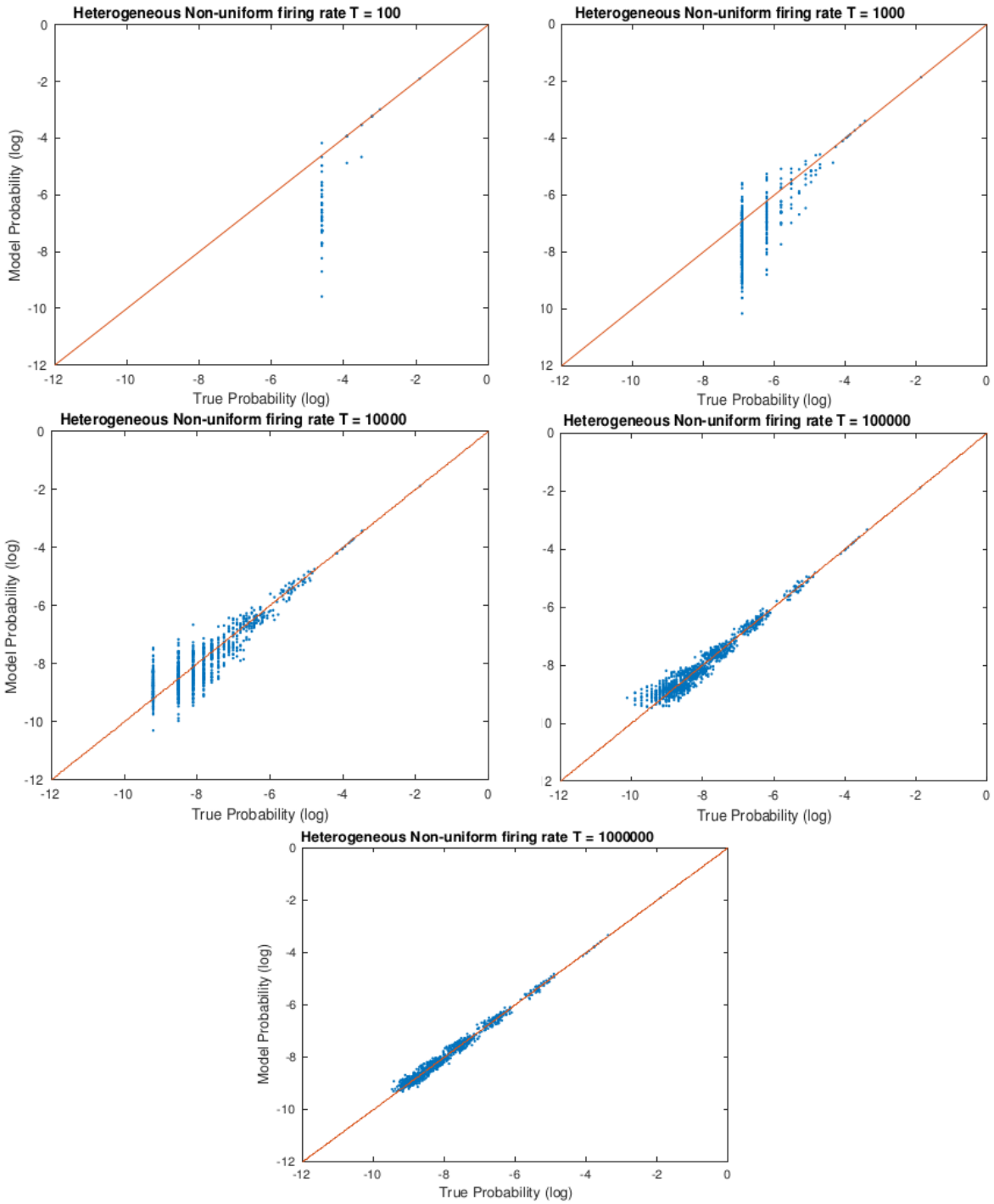


Figure A.7: Full evolution of the Heterogeneous Non-Uniform model probability when compared to the true probability over time steps.

Bibliography

[1] Cian O'Donnell, J Tiago Gonçalves, Nick Whiteley, Carlos Portera-Cailliau, and Terrence J Sejnowski. The population tracking model: A simple, scalable statistical model for neural population data. *Neural computation*, 2016.

[2] Dagmara Panas, Hayder Amin, Alessandro Maccione, Oliver Muthmann, Mark van Rossum, Luca Berdondini, and Matthias H Hennig. Sloppiness in spontaneously active neuronal networks. *Journal of Neuroscience*, 35(22):8480–8492, 2015.

[3] Torsten N Wiesel and David H Hubel. Single-cell responses in striate cortex of kittens deprived of vision in one eye. *Journal of neurophysiology*, 26(6):1003–1017, 1963.

[4] Pierre Gianferrara. Modelling populations of spiking neurons in Autistic Spectrum Disorder. *Informatics Research Proposal*, pages 1–14, 2017.

[5] György Buzsáki. Large-scale recording of neuronal ensembles. *Nature neuroscience*, 7(5):446, 2004.

[6] John Hertz, Yasser Roudi, and Joanna Tyrcha. Ising models for inferring network structure from spike data. *arXiv preprint arXiv:1106.1752*, 2011.

[7] Gašper Tkačik, Olivier Marre, Dario Amodei, Elad Schneidman, William Bialek, and Michael J Berry II. Searching for collective behavior in a large network of sensory neurons. *PLoS computational biology*, 10(1):e1003408, 2014.

[8] Elad Schneidman, Michael J Berry, Ronen Segev II, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007, 2006.

[9] Yasser Roudi, Benjamin Dunn, and John Hertz. Multi-neuronal activity and functional connectivity in cell assemblies. *Current opinion in neurobiology*, 32:38–44, 2015.

[10] Patrick J Rousche and Richard A Normann. Chronic recording capability of the Utah Intracortical Electrode Array in cat sensory cortex. *Journal of neuroscience methods*, 82(1):1–15, 1998.

[11] EM Schmidt, MJ Bak, FT Hambrecht, CV Kufta, DK O'rourke, and P Vallabhanath. Feasibility of a visual prosthesis for the blind based on intracortical micro stimulation of the visual cortex. *Brain*, 119(2):507–522, 1996.

[12] TAGA Bonhoeffer. Optical Imaging Based on Intrinsic Signals The Methodology. *Brain mapping: The methods*, pages 55–97, 1996.

[13] Hongbo Jia, Nathalie L Rochefort, Xiaowei Chen, and Arthur Konnerth. In vivo two-photon imaging of sensory-evoked dendritic calcium signals in cortical neurons. *Nature protocols*, 6(1):28, 2011.

[14] Olga Garaschuk, Ruxandra-Iulia Milos, and Arthur Konnerth. Targeted bulk-loading of fluorescent indicators for two-photon brain imaging in vivo. *Nature protocols*, 1(1):380, 2006.

[15] Olga Garaschuk, Ruxandra-Iulia Milos, Christine Grienberger, Nima Marandi, Helmuth Adelsberger, and Arthur Konnerth. Optical monitoring of brain function in vivo: from neurons to networks. *Pflügers Archiv*, 453(3):385–396, 2006.

[16] Tsai-Wen Chen, Trevor J Wardill, Yi Sun, Stefan R Pulver, Sabine L Renninger, Amy Baohan, Eric R Schreiter, Rex A Kerr, Michael B Orger, Vivek Jayaraman, et al. Ultra-sensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295, 2013.

[17] Christoph Stosiek, Olga Garaschuk, Knut Holthoff, and Arthur Konnerth. In vivo two-photon calcium imaging of neuronal networks. *Proceedings of the National Academy of Sciences*, 100(12):7319–7324, 2003.

[18] Peter TC So, Chen Y Dong, Barry R Masters, and Keith M Berland. Two-photon excitation fluorescence microscopy. *Annual review of biomedical engineering*, 2(1):399–429, 2000.

[19] Yasser Roudi, Sheila Nirenberg, and Peter E Latham. Pairwise maximum entropy models for studying large biological systems: when they can work and when they can't. *PLoS computational biology*, 5(5):e1000380, 2009.

[20] Jakob H Macke, Philipp Berens, Alexander S Ecker, Andreas S Tolias, and Matthias Bethge. Generating spike trains with specified correlation coefficients. *Neural computation*, 21(2):397–423, 2009.

[21] Eftychios A Pnevmatikakis, Daniel Soudry, Yuanjun Gao, Timothy A Machado, Josh Merel, David Pfau, Thomas Reardon, Yu Mu, Clay Lacefield, Weijian Yang, et al. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89(2):285–299, 2016.

[22] Thomas Deneux, Attila Kaszas, Gergely Szalay, Gergely Katona, Tamás Lakner, Amiram Grinvald, Balázs Rózsa, and Ivo Vanzetta. Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature communications*, 7, 2016.

[23] Kathryn B Garber, Jeannie Visootsak, and Stephen T Warren. Fragile X syndrome. *European journal of human genetics: EJHG*, 16(6):666, 2008.

[24] Matthew K Belmonte and Thomas Bourgeron. Fragile X syndrome and autism at the intersection of genetic and neural networks. *Nature neuroscience*, 9(10):1221, 2006.

[25] Michael R Santoro, Steven M Bray, and Stephen T Warren. Molecular mechanisms of fragile X syndrome: a twenty-year perspective. *Annual Review of Pathology: Mechanisms of Disease*, 7:219–245, 2012.

[26] K Snow, LK Doud, R Hagerman, RG Pergolizzi, SH Erster, and Stephen N Thibodeau. Analysis of a CGG sequence at the FMR-1 locus in fragile X families and in the general population. *American journal of human genetics*, 53(6):1217, 1993.

[27] Tobias Rose, Juliane Jaepel, Mark Hübener, and Tobias Bonhoeffer. Cell-specific restoration of stimulus preference after monocular deprivation in the visual cortex. *Science*, 352(6291):1319–1322, 2016.

[28] Stephanie A Barnes, Lasani S Wijetunge, Adam D Jackson, Danai Katsanevaki, Emily K Osterweil, Noboru H Komiyama, Seth GN Grant, Mark F Bear, U Valentin Nägerl, Peter C Kind, et al. Convergence of hippocampal pathophysiology in Syngap+/- and Fmr1-/y mice. *Journal of Neuroscience*, 35(45):15073–15081, 2015.

[29] Kimberly Gerrow and Antoine Triller. Synaptic stability and plasticity in a floating world. *Current opinion in neurobiology*, 20(5):631–639, 2010.

[30] Robert S Zucker and Wade G Regehr. Short-term synaptic plasticity. *Annual review of physiology*, 64(1):355–405, 2002.

[31] Guilherme Neves, Sam F Cooke, and Tim VP Bliss. Synaptic plasticity, memory and the hippocampus: a neural network approach to causality. *Nature reviews. Neuroscience*, 9(1):65, 2008.

[32] Dale Purves, George J Augustine, David Fitzpatrick, WC Hall, AS LaMantia, JO McNamara, and LE White. Neuroscience, 2008. *De Boeck, Sinauer, Sunderland, Mass*, 2014.

[33] Gina Turrigiano. Too many cooks? Intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement. *Annual review of neuroscience*, 34:89–103, 2011.

[34] Gina G Turrigiano and Sacha B Nelson. Homeostatic plasticity in the developing nervous system. *Nature reviews. Neuroscience*, 5(2):97, 2004.

[35] Gina Turrigiano. Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function. *Cold Spring Harbor perspectives in biology*, 4(1):a005736, 2012.

[36] Gina G Turrigiano, Kenneth R Leslie, Niraj S Desai, Lana C Rutherford, and Sacha B Nelson. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670):892, 1998.

[37] Niraj S Desai, Lana C Rutherford, and Gina G Turrigiano. Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nature neuroscience*, 2(6), 1999.

[38] Gina Turrigiano, LF Abbott, Eve Marder, et al. Activity-dependent changes in the intrinsic properties of cultured neurons. *Science-AAAS-Weekly Paper Edition-including Guide to Scientific Information*, 264(5161):974–976, 1994.

[39] Huda Y Zoghbi and Mark F Bear. Synaptic dysfunction in neurodevelopmental disorders associated with autism and intellectual disabilities. *Cold Spring Harbor perspectives in biology*, 4(3):a009886, 2012.

[40] Huda Y Zoghbi. Postnatal neurodevelopmental disorders: meeting at the synapse? *Science*, 302(5646):826–830, 2003.

[41] Annemieke JMH Verkerk, Maura Pieretti, James S Sutcliffe, Ying-Hui Fu, Derek PA Kuhl, Antonio Pizzuti, Orly Reiner, Stephen Richards, Maureen F Victoria, Fuping Zhang, et al. Identification of a gene (FMR-1) containing a CGG repeat coincident with a breakpoint cluster region exhibiting length variation in fragile X syndrome. *Cell*, 65(5):905–914, 1991.

[42] Victoria Brown, Peng Jin, Stephanie Ceman, Jennifer C Darnell, William T O'Donnell, Scott A Tenenbaum, Xiaokui Jin, Yue Feng, Keith D Wilkinson, Jack D Keene, et al. Microarray identification of FMRP-associated brain mRNAs and altered mRNA translational profiles in fragile X syndrome. *Cell*, 107(4):477–487, 2001.

[43] Peng Jin and Stephen T Warren. Understanding the molecular basis of fragile X syndrome. *Human molecular genetics*, 9(6):901–908, 2000.

[44] Sacha B Nelson and Vera Valakh. Excitatory/inhibitory balance and circuit homeostasis in autism spectrum disorders. *Neuron*, 87(4):684–698, 2015.

[45] Emily K Osterweil, Shih-Chieh Chuang, Alexander A Chubykin, Michael Sidorov, Riccardo Bianchi, Robert KS Wong, and Mark F Bear. Lovastatin corrects excess protein synthesis and prevents epileptogenesis in a mouse model of fragile X syndrome. *Neuron*, 77(2):243–250, 2013.

[46] J Tiago Gonçalves, James E Anstey, Peyman Golshani, and Carlos Portera-Cailliau. Circuit level defects in the developing neocortex of Fragile X mice. *Nature neuroscience*, 16(7):903–909, 2013.

[47] Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

[48] Steve M Potter. Distributed processing in cultured neuronal networks. *Progress in brain research*, 130:49–62, 2001.

[49] Jens-Oliver Muthmann, Hayder Amin, Evelyne Sernagor, Alessandro Maccione, Dagmara Panas, Luca Berdondini, Upinder S Bhalla, and Matthias H

Hennig. Spike detection for large neural populations using high density multielectrode arrays. *Frontiers in neuroinformatics*, 9, 2015.

[50] Gerrit Hilgen, Martino Sorbaro, Sahar Pirmoradian, Jens-Oliver Muthmann, Ibolya Edit Kepiro, Simona Ullo, Cesar Juarez Ramirez, Albert Puente Encinas, Alessandro Maccione, Luca Berdondini, et al. Unsupervised spike sorting for large-scale, high-density multielectrode arrays. *Cell reports*, 18(10):2521–2532, 2017.

[51] Claude E Shannon and Warren Weaver. The mathematical theory of information. 1949.

[52] Fred Rieke. *Spikes: exploring the neural code*. MIT press, 1999.

[53] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[54] Pietro Berkes, Gergő Orbán, Máté Lengyel, and József Fiser. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, 331(6013):83–87, 2011.

[55] John H Van Drie. The Boltzmann/Shannon entropy as a measure of correlation. *arXiv preprint math-ph/0001024*, 2000.

[56] Janelle MP Pakan, Scott C Lowe, Evelyn Dylda, Sander W Keemink, Stephen P Currie, Christopher A Coutts, and Nathalie LI Rochefort. Behavioral-state modulation of inhibition is context-dependent and cell type specific in mouse visual cortex. *Elife*, 5:e14985, 2016.

[57] Michael Okun, Pierre Yger, Stephan L Marguet, Florian Gerard-Mercier, Andrea Benucci, Steffen Katzner, Laura Busse, Matteo Carandini, and Kenneth D Harris. Population rate dynamics and multineuron firing patterns in sensory cortex. *Journal of Neuroscience*, 32(48):17108–17119, 2012.

[58] Michael Okun, Nicholas Steinmetz, Lee Cossell, M Florencia Iacaruso, Ho Ko, Péter Barthó, Tirin Moore, Sonja B Hofer, Thomas D Mrsic-Flogel, Matteo Carandini, et al. Diverse coupling of neurons to populations in sensory cortex. *Nature*, 521(7553):511, 2015.