Investigating, implementing, and creating methods for analysing large neuronal ensembles

---

Author:
Thomas J. DELANEY

Supervisors:
Dr. Cian O'DONNELL
Dr. Michael C. ASHBY

ii

July 2, 2020

Declaration of Authorship

I, Thomas J. DELANEY, declare that this thesis titled, "Investigating, implementing, and creating methods for analysing large neuronal ensembles" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.

- Where I have consulted the published work of others, this is always clearly attributed.

- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

- I have acknowledged all main sources of help.

- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF BRISTOL

Abstract

Engineering

Department of Computer Science

Doctor of Philosophy

Investigating, implementing, and creating methods for analysing large neuronal ensembles

by Thomas J. DELANEY

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too. . .

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

xii

List of Figures

List of Tables

367 *For/Dedicated to/To my...*

368 **Chapter 1**

369 Sensitivity of the 370 spikes-to-fluorescence transform to 371 calcium indicator and neuron 372 properties

UNIVERSITY OF BRISTOL

Abstract

Engineering

Department of Computer Science

Doctor of Philosophy

Investigating, implementing, and creating methods for analysing large neuronal ensembles

by Thomas J. DELANEY

Fluorescent calcium indicators such as GCaMP are widely used to monitor neuronal activity. However the relationship between the fluorescence signal and the underlying action potential firing is poorly understood. This lack of knowledge makes it difficult for experimenters to decide between different indicator variants for a given application. We addressed this problem by studying a basic biophysical model of calcium dynamics in neuronal soma. We fit the model parameters to publicly available data where GCaMP6s fluorescence and whole-cell electrophysiological recordings were made simultaneously in the same single neurons. We systematically varied the model's parameters to characterise the sensitivity of spike train inference algorithms to the calcium indicator's main biophysical properties: binding rate, dissociation rate, and molecular concentration. This model should have two potential uses: experimental researchers may use it to help them select the optimal indicator for their desired experiment; and computational researchers may use it to generate simulated data to aid design of spike inference algorithms.

1.1   Introduction

Although fluorescent calcium indicators such as GCaMP are widely used to monitor neuronal activity, the relationship between the fluorescence signal and the underlying action potential firing is imperfect. The fluorescence signal has a low signal-to-noise ratio, and most indicators' kinetics are slow relative to the millisecond-timescale dynamics of the membrane voltage (example in Figure 1). This makes spike inference difficult. Furthermore, the effects of the indicator and cell properties on the fluorescence signal are unknown. For example, genetically encoded indicators can accumulate within neurons over weeks and months (Chen et al., 2013). Studies using calcium-sensitive fluorescent dyes have shown that indicator concentration has substantial effects on the spike-to-fluorescence relationship (Maravall et al., 2000). Therefore spike rates inferred from GCaMP fluorescence signals may give misleading results if comparing across imaging sessions. More generally, the poor understanding of the spike-to-fluorescence transform means experimenters may not know whether to trust the outputs of spike train inference methods in any given application.

Spike trains are usually inferred from the time series of intensity values of one pixel of the fluorescence image, where the pixel is located at the cell's soma. The problems of identfying these pixels, and inferring spikes from their time series can solved separately or together. When attempting to infer spikes, the fluorescence trace is modelled as a linear combination of calcium concentration dynamics, a baseline calcium concentration, and some Gaussian noise. The calcium concentration dynamics are modelled as an autoregressive process of degree 1 or 2 with a pulse input corresponding to the spike train, or the number of spikes fired in a time step. The model includes no dynamics for the fluorescent indicator itself. Furthermore, in order to make this model into an easily solvable linear programming problem the number of spikes fired in a timestep is not restricted to non-negative integers but to arbitrary non-negative values (Vogelstein et al., 2010; Pnevmatikakis et al., 2016; Friedrich and Paninski, 2016; Pnevmatikakis et al., 2013; Pnevmatikakis et al., 2014). More biologically inspired spike inference models do exist (Deneux et al., 2016), but their fundamentals are very similar. In this work, we investigated the effect of changing dynamics and buffer concentrations on the accuracy of the inference algorithms based on these models.

The aim of this project was to model the fluorescence traces produced by a fluorescent calcium indicator in a neuron soma resulting from a specific spike train, given calcium indicator parameters such as binding rate, dissociation rate, and molecular concentration. Such
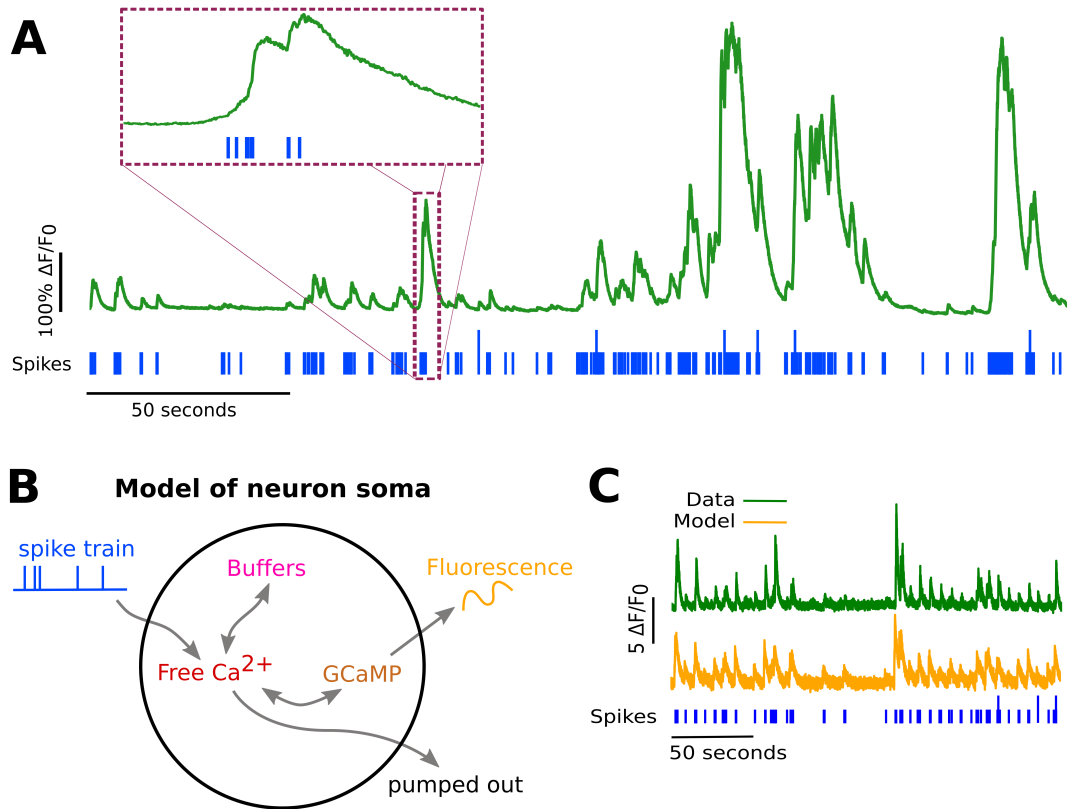
FIGURE 1.1:
A: Example spike train (blue) and the corresponding GCaMP6s fluorescence trace (green), data replotted from (Berens et al., 2018). Inset shows zoomed section of traces to highlight slow decay of GCaMP6s fluorescence relative to spike time intervals.
B: Schematic diagram of the neuron calcium and GCaMP computational model.
C: Good visual match of data fluorescence trace (green) and model simulated fluorescence (orange) in response to an identical spike train (blue).

426   a model would allow benchmarking of various spike inference algorithms, and enable under-

427   standing of how indicator characteristics affect the quality of spike train inference.

428   The model we developed consisted of free calcium, fluorescent indicator molecules, and

429   mobile and immobile endogenous calcium buffers. The indicator molecules which were

430   bound to a calcium molecule could be either excited, i.e. able to release a photon, or relaxed.

431   In order to reproduce the noise inherent in the data collection, we modelled the release of

432   photons from the excited indicator bound calcium as a stochastic process.

433   The fluorescence traces produced by the simulation were calibrated to reproduce the

434   signal-to-noise ratio observed in experimental data. Previously published spike inference

435   algorithms were then used to infer spike trains from the experimental fluorescence traces and

436   the modelled fluorescence traces. The parameters of the model were then varied in order to

437   determine the effect on the system dynamics and the effects on spike inference.

1.2   Results

1.2.1   A biophysical computational model can generate accurate fluorescence traces from spike trains

To study the relationship between action potential firing and calcium fluorescence, we built a computational model of calcium dynamics in a neuronal soma. The model consisted of four dynamic variables: the concentration of free calcium, two types of endogenous buffer, and the calcium-sensitive fluorescent indicator. Each of the buffers and the indicator could independently bind and unbind with calcium. These reactions were modelled as

$$[X][Ca^{2+}] \underset{k_{Xb}}{\overset{k_{Xf}}{\rightleftharpoons}} [XCa]$$
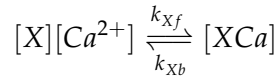
where X is the buffer concentration and Ca^{2+} is the calcium concentration. Each species could therefore exist in two states: either bound with calcium or unbound. To model the imaging process, we also added a third, excited state to the indicator. When in the calcium-bound state, the indicator could be converted to an excited state, corresponding to the absorption of a photon. The rate of this excitation process could be interpreted as the intensity of the light illuminating the sample. Once excited, the species decayed back to the unexcited state at a fixed rate, corresponding to the spontaneous emission of photons. The total emitted fluorescence signal was interpreted as proportional to this de-excitation flux. To represent experimental noise in the photon capture process, we drew a random number of captured photons at each time step from a binomial distribution, parameterised by a number p that corresponds to the mean fraction of released photons that are captured.

The model had 17 parameters in total describing the molecules' concentrations and reaction rates (Methods). We set 13 of these parameters to values from the literature. The remaining 4 parameter values we fit to publicly-available data (Berens et al., 2018), briefly explained as follows (see Methods for full details). Single neurons from acute rat cortical slices expressing GCaMP6f were imaged with two-photon microscopy while the membrane potentials of the somata of the same neurons were simultaneously recorded via whole-cell patch clamp electrophysiology. In this dataset, the electrical recordings give unambiguous information about neurons' spike times. To do the parameter fitting, we feed these spike trains as inputs to the computational model. After running, the model returns a simulated fluorescence trace. We aimed to find the model parameter values that give the best match between this simulated fluorescence trace and the real fluorescence time series recorded in

the corresponding neuron. To do this we used a suite of optimisation procedures to jointly fit both the real neuron's fluorescence time series and power spectrum, which capture complementary information about the spikes-to-fluorescence mapping (Methods). We performed the fitting procedure independently for each of the 20 neurons in the spikefinder dataset (http://spikefinder.org). After fitting, the model produced realistic-looking fluorescence time series (Figure 1.1).

1.2.2 Spike inference algorithms perform similarly on real data compared with time series simulated from the model

Researchers often pass the fluorescence time series through a spike inference tool before performing further statistical analyses. These spike inference algorithms take the fluorescence trace as input and attempt to estimate the neuronal spike train that triggered them (Vogelstein et al., 2010; Pnevmatikakis et al., 2016; Friedrich and Paninski, 2016; Pnevmatikakis et al., 2013; Pnevmatikakis et al., 2014; Deneux et al., 2016). Part of our motivation for building this model was to allow us to ask the question: how do the properties of the cell and the calcium indicator affect the quality of spike inference? In order to trust the conclusions from our model, we should first be confident that spike inference from our simulated fluorescence traces is similar to that from the real data. To test this we passed each of the simulated fluorescence traces through three previously published spike inference algorithms, quantified their performance against the ground-truth electrophysiology data, repeated the procedure for the real calcium fluorescence time series, and compared the accuracy of the inference processes in all cases. The *true positive rate*, also known as the *recall*, the *sensitivity*, or the *probability of detection* of spike inference varied across the three inference algorithms we tried (p value and statistical test here). The constrained non-negative matrix deconvolution algorithm (Pnevmatikakis et al., 2016) (CNMD algorithm) correctly detected approximately 45% of the true spikes, the OASIS algorithm (Friedrich and Paninski, 2016) correctly detected approximately 35% of the true spikes, and the ML spike algorithm (Deneux et al., 2016) correctly detected approximately 15% of the true spikes (see figure 1.2). Notably, for two of the three inference algorithms, the quality of inference was also fairly consistent for individual spike trains, not just the group means ($p > 0.05$, paired t-test). This demonstrates that the models were generating fluorescence time series that were similarly difficult to decode as the real data, in ways that were not specific to any one inference algorithm. This is evidence that the models captured real aspects of the spikes-to-fluorescence transform.
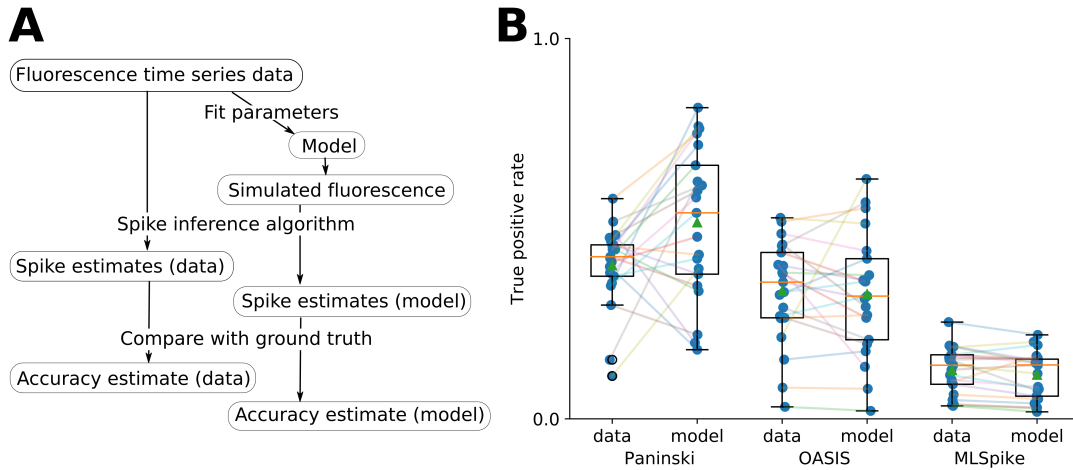
FIGURE 1.2:
A: Workflow to compare spike inference for real versus simulated fluorescence data.
B: True positive rates achieved by three different spike inference algorithms when applied to observed spike trains, and simulated spike trains. Data points overlaid as blue circles. The performance is similar from real and simulated data for each of the algorithms.

1.2.3 Relative effects of various buffers to the fluorescence signal

One of the benefits of computational models over laboratory experiments is that we can observe all the variables in the simulation to gain insight into the system's dynamics, which can be difficult to do in the lab. We plotted the concentrations of the various species over time for a version of the model fit to one data set, in response to the same train of spikes used for fitting (figure 1.3). Figure 1.3a shows the absolute values of the species concentrations, summed. Consistent with experimental estimates (Maravall et al., 2000), only a small fraction ($\sim$ 0.1%) of calcium is free and unbound to any buffer. Of the bound calcium, the vast majority, ($\sim$ 96%) is bound to the GCaMP indicator. The two types of endogenous buffer are bound to the remaining calcium ($\sim$ 4%). An influx of calcium from a single spike adds very little to the total calcium, in relative terms (red line in Figure 3a).

When calcium entered the model neuron it was rapidly buffered (Bartol et al., 2015). However the relative fractions of which buffer molecules bound to the influxed calcium was dynamic, and changed over time . Figure 1.3 (b-f) shows the time course of the various species over time in response to a calcium influx event from a single action potential. Crucially, the indicator $[BCa]$ competed with the endogenous buffers $[ImCa]$ and $[ECa]$ – all three bind calcium on similar timescales. This implies that the timecourse and amplitude of the $[BCa]$ variable will also depend on the binding rates and availabilities of the endogenous buffers. For example if we decreased the concentration of an endogenous buffer, we might
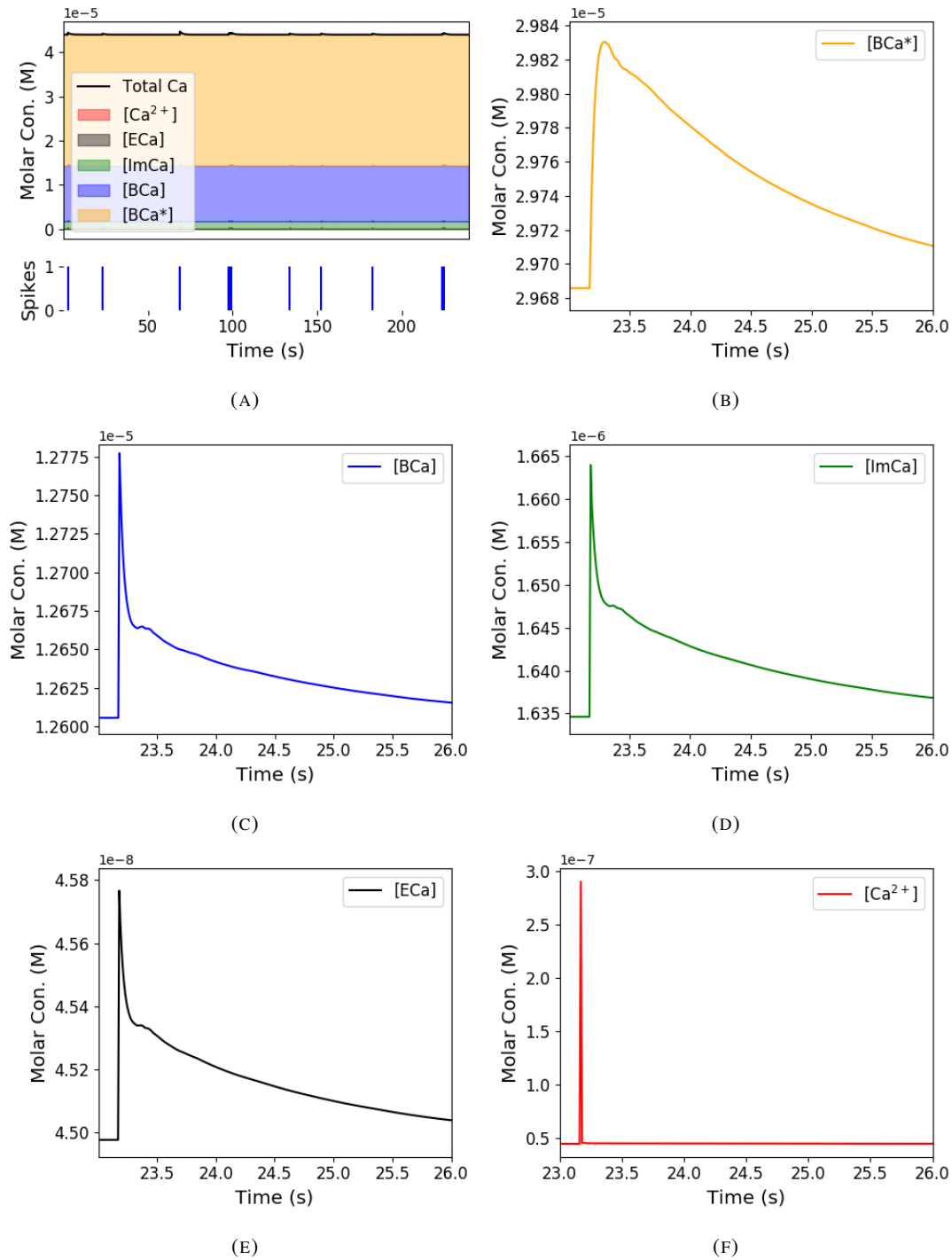
FIGURE 1.3: **Calcium Buffering Dynamics** (a) The proportions of bound and free calcium concentrations within a cell, with the associated spike train. (b)-(f) The dynamics of the concentration of (b) excited indicator bound calcium, (c) indicator bound calcium, (d) immobile endogenous buffer bound calcium, (e) mobile endogenous buffer bound calcium, and (f) free calcium in response to an action potential at ∼23.2s.

expect both a faster rise time and greater peak amplitude of the $[BCa]$ signal in response to a calcium influx event. The slowest component of the decay had a similar time constant for $[BCa]$, $[ImCa]$ and $[ECa]$, which in turn matched the $[Ca]$ extrusion time constant in our model ($\sim 6.29 \times 10^{-22} \text{Ms}^{-1}$). This implies that the buffers and the indicator had reached a dynamic equilibrium and were jointly tracking the free calcium concentration as calcium was slowly extruded from the cell.

Interestingly the excited bound calcium species ($[BCa^*]$) showed a qualitatively different timecourse in response to a calcium influx event. This concentration is subject to the added 'excitation and release' dynamic, where a certain proportion of the concentration absorbs the energy from an incoming photon and goes into an 'excited state' at each time step. A certain proportion of the concentration releases a photon and reverts to a 'relaxed state' at each timestep also. This means that the excited bound calcium lags behind the bound calcium trace. We could think of the excited bound calcium trace as a low pass filtered version of the bound calcium trace.

1.2.4 Spike inference accuracy is sensitive to indicator properties, and likely varies within and between cells

The above results imply that the fluorescence signal depends on the relative properties of both GCaMP and the endogenous buffers. We next used the model to directly ask how sensitive spike inference was to these components. We focused on three key parameters that likely vary from cell to cell and experiment to experiment: GCaMP binding kinetics, GCaMP concentration, and endogenous buffer concentration.

Several variants of GCaMP itself have been made that differ in calcium binding kinetics, baseline fluorescence, fluorescence efficiency, and other factors. For example, GCaMP6f has a decay time constant of ~ 1s, while GCaMP6s has a decay time constant of ~ 2s (Chen et al., 2013). Here we asked how these differences in binding kinetics affect spike inference. We jointly varied the calcium binding and unbinding rates of the indicator by the same factor over a range from 100-fold slower to 100-fold faster from the fitted values, and simulated the fluorescence response for each of the parameter settings in response to the same spike trains as before (figure 1.4). Notably this manipulation does not affect the indicators affinity, and therefore would not affect steady-state responses to prolonged changes in calcium. Instead it is likely to affect its sensitivity to the spike train dynamics. We computed two summary measures from the simulated fluorescence traces: the signal-to-noise ratio for a single spike (Methods, section 1.4.6), and the accuracy of spike inference for each of the spike trains. We

observed a reduction in the signal-to-noise ratio and the spike inference quality when we set the binding and unbinding rates were set to one hundreth of their fitted values, and to one tenth of their fitted values. When we increased the value of both binding rates, we observed no change in these measurements. The reduction in both rates lead to smaller increases in fluorescence in response to an action potential and a longer decay time (figure 1.4a), this caused the reduction in signal-to-noise ratio. As both rates were increased, the change in $\Delta F/F_0$ in response to an action potential increased and the decay time decreased slightly, but the fluorescence trace created by these values was very similar to the trace created by the fitted values.

Second, the overall concentrations of GCaMP often varies from cell to cell. For example different cells, even of the same type in the same tissue, can express different levels of GCaMP, due to proximity to the infection site, or the cell becoming 'nuclear-filled' (Tian et al., 2009; Chen et al., 2013). Also, GCaMP is often used for longitudinal experiments where the same cells are re-imaged across multiple days or weeks. However since GCaMP expression typically ramps up over time (Chen et al., 2013), the accuracy of spike inference may differ across multiple longitudinal recordings in the same cell. We addressed this by varying the concentration of calcium indicator in the model, simulating spike trains and measuring signal-to-noise ratio and spike inference accuracy on the resulting fluorescence traces. Both increasing and decreasing the concentration of the indicator had effects on the fluorescence trace, signal-to-noise ratio, and spike inference. The signal-to-noise ratio and spike inference quality decreased with decreased indicator concentration, and both showed a decrease when the indicator concentration was increased to 100 times it's fitted value (figure 1.5). The signal-to-noise ratio showed an increase when the indicator concentration was increased to 10 times it's fitted value, but there was no corresponding change in the spike inference quality. The decrease in indicator concentration caused a reduction in the increase in $\Delta F/F_0$ in response to an action potential, and an increase in the decay time of this increase (figure 1.5a). The increase in indicator concentration had the opposite effect, it casued an increase in the change in $\Delta F/F_0$ in response to an action potential, and a decrease in the decay time.

Third, the concentration and types of endogenous calcium buffers also vary from neuron to neuron, both within and between cell types (Bartol et al., 2015; Maravall et al., 2000; Neher and Augustine, 1992). Since the calcium buffer capacity of neurons is high, around 50-70 (Lee et al., 2000) in excitatory hippocampal pyramidal cells, around 100-250 (Lee et al., 2000) in inhibitory hippocampal pyramidal cells, and 900-200 in Purkinje cells (depending on the age of the subject), these endogenous buffers compete with GCaMP for binding

(A)

(B)

(C)
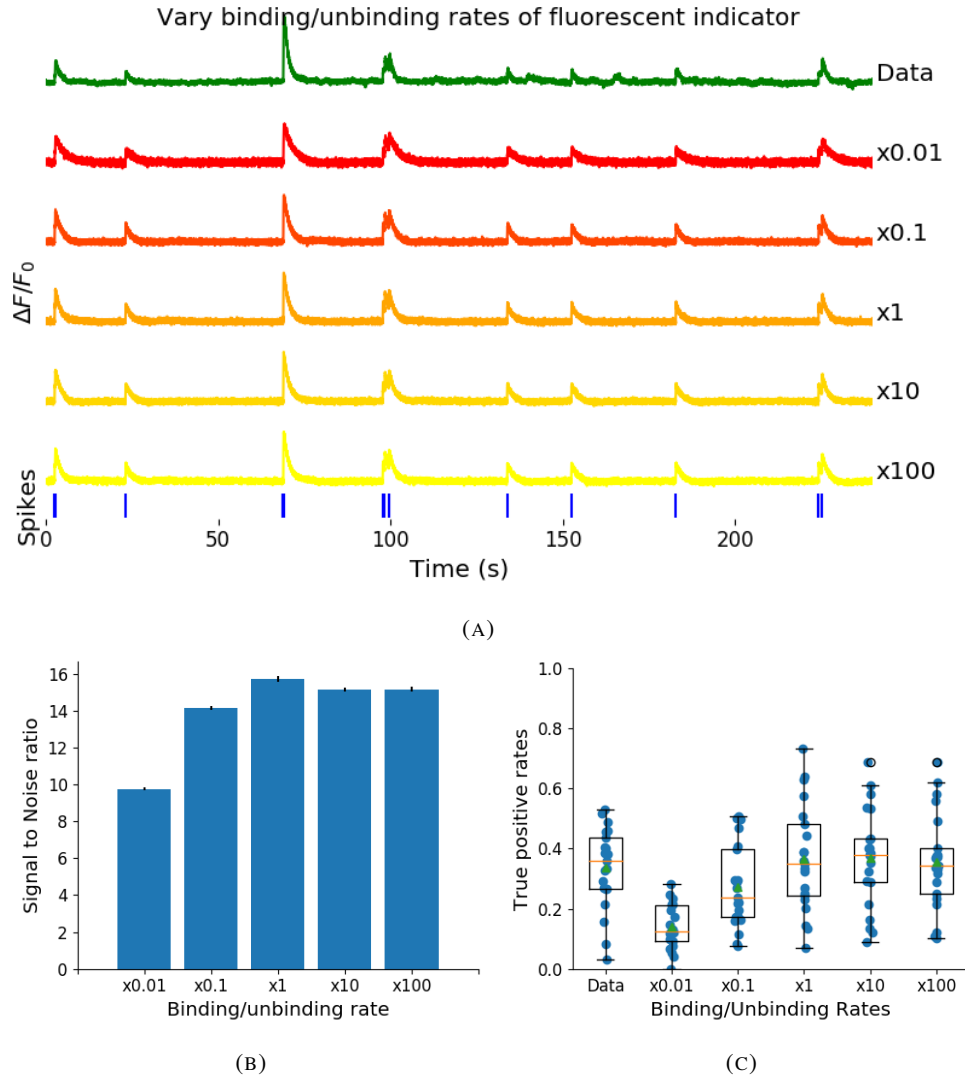
FIGURE 1.4: (a) An example trace for each of the five pairs of values used for the binding and unbinding rates of the fluorescent calcium indicator. (b) The signal-to-noise ratio of the modelled fluorescence traces using each of the four perturbed values, and the experimental value. The SNRs for the two pairs with values lower than the experimental value are lower than the experimental pair or the higher value pairs. (c) The true-positive rates of the deconvolution algorithm's predictions when inferring from the observed data, and inferring from modelled traces using the perturbed and experimental values.

(A)

(B)

(C)
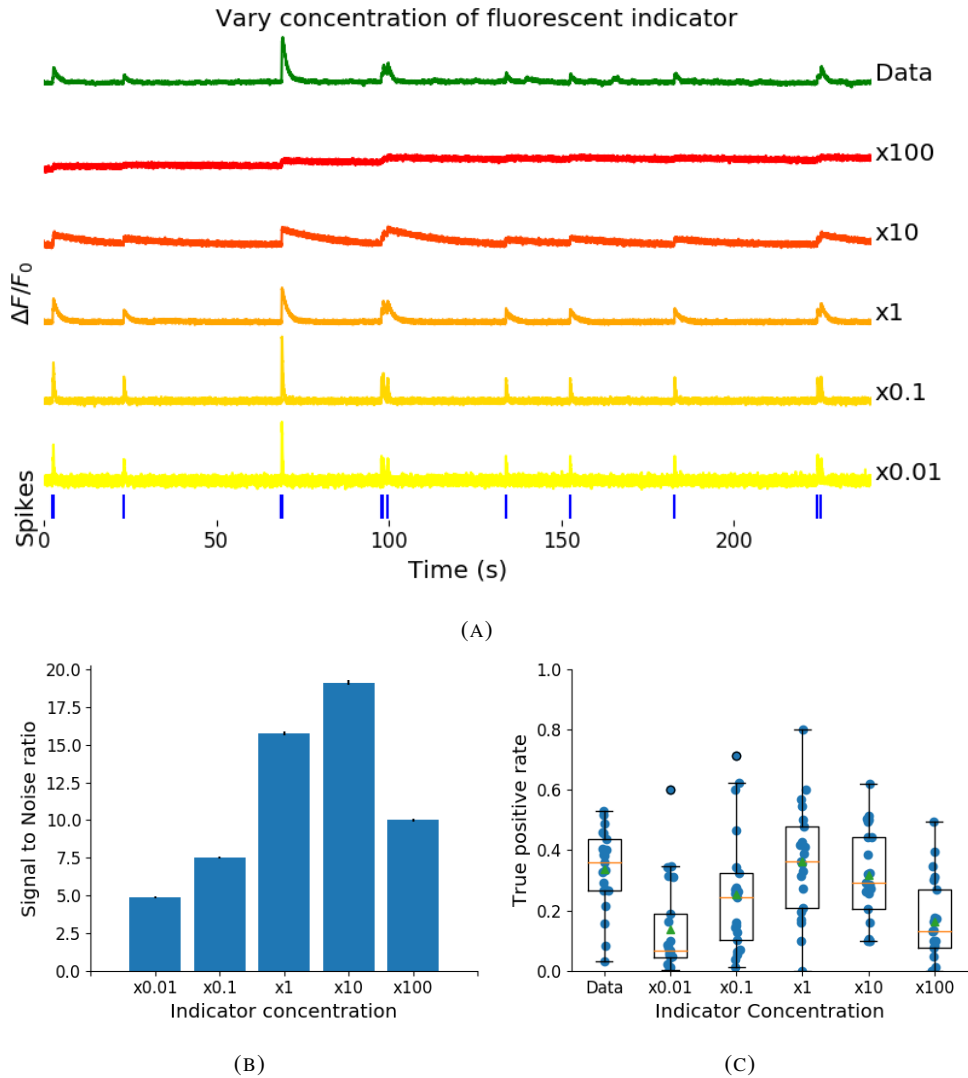
FIGURE 1.5: (a) An example trace for each of the five perturbed values for the concentration of fluorescent calcium indicator. The top two traces are produced by the lower perturbed values, the middle trace is produced by the experimental value, and the lowest two traces are produced when using the higher perturbed values. (b) The signal-to-noise ratio of the modelled fluorescence traces using each of the four perturbed values, and the experimental value. Extreme perturbations of the concentration either above or below the experimental level lowered the SNR. (c) The true-positive rates of the deconvolution algorithm's predictions when inferring from the observed data, and inferring from modelled traces using the perturbed and experimental values. We found that the algorithms performs equally badly on the two most extreme values, and performs equally well on the experimental value, and the next higher perturbed value.

to calcium, and variations in endogenous buffer concentration may affect GCaMP signal and therefore spike inference. To address this we varied the concentration of the endogenous buffer in the model neuron over five orders of magnitude from 0.8 to 8000 μM, simulated calcium fluorescence traces in response to the same set of spike trains, and performed spike inference on the resulting fluorescence time series. Increasing the endogenous buffer concentration had a substantial effect on the GCaMP fluorescence signal, both decreasing its amplitude and slowing its kinetics (figure 1.6(a)). This corresponded with a decrease in both single-spike signal-to-noise ratio (figure 1.6(b)) and spike inference accuracy (figure 1.6(c)). In contrast, decreasing endogenous buffer capacity from the fitted value had little effect on either the GCaMP signal or spike inference (figure 1.6).

1.2.5   Single spike inference accuracy drops for high firing rates, but firing rate itself can be estimated from mean fluorescence amplitude

The fluorescence signal recorded from neurons using calcium indicators is typically much slower than changes in membrane potential for two reasons: first, because the calcium and the indicator have slow binding and unbinding kinetics, the signal is a low-pass filtered version of the membrane potential. Second, neuronal two-photon imaging experiments are often performed in scanning mode, which limits their frame rate to $\sim$ 10Hz or slower. This implies that multiple spike events that occur close in time might be difficult to resolve from a calcium indicator time series. Many cells, especially several types of inhibitory interneurons, fire tonically at rates higher than 10Hz. We used the model to test whether spike inference accuracy depended on the neuron's firing frequency by driving the cell with spike trains sampled from a Poisson processes of varying frequency. We simulated a variable firing rate using an Ornstein-Uhlenbeck process, and simulated the spike trains using a Poisson distribution with its rate taken from this process. Because of the high frequency firing rate of these spike trains, we using the accuracy as the measure of spike inference quality. We simulated 30 spike trains at average firing rate of 1, 5, and 10Hz, and measured the spike inference quality of all these traces. Spike inference accuracy decreased with increasing firing rate, for up to 10Hz Poisson spike trains (figure 1.8(left)). Although, the accuracy remained above 90% for each of the three frequencies. We also plotted the average $\Delta F/F_0$ as a function of stimulation firing rate. We found that it increased monotonically as a function of firing rate (figure 1.8(right)).

We expected lower spike inference quality as the average spiking frequency increased. Since the fluorescence trace, in some sense, is a low pass filtered version of the spike train, a

(A)

(B)

(C)
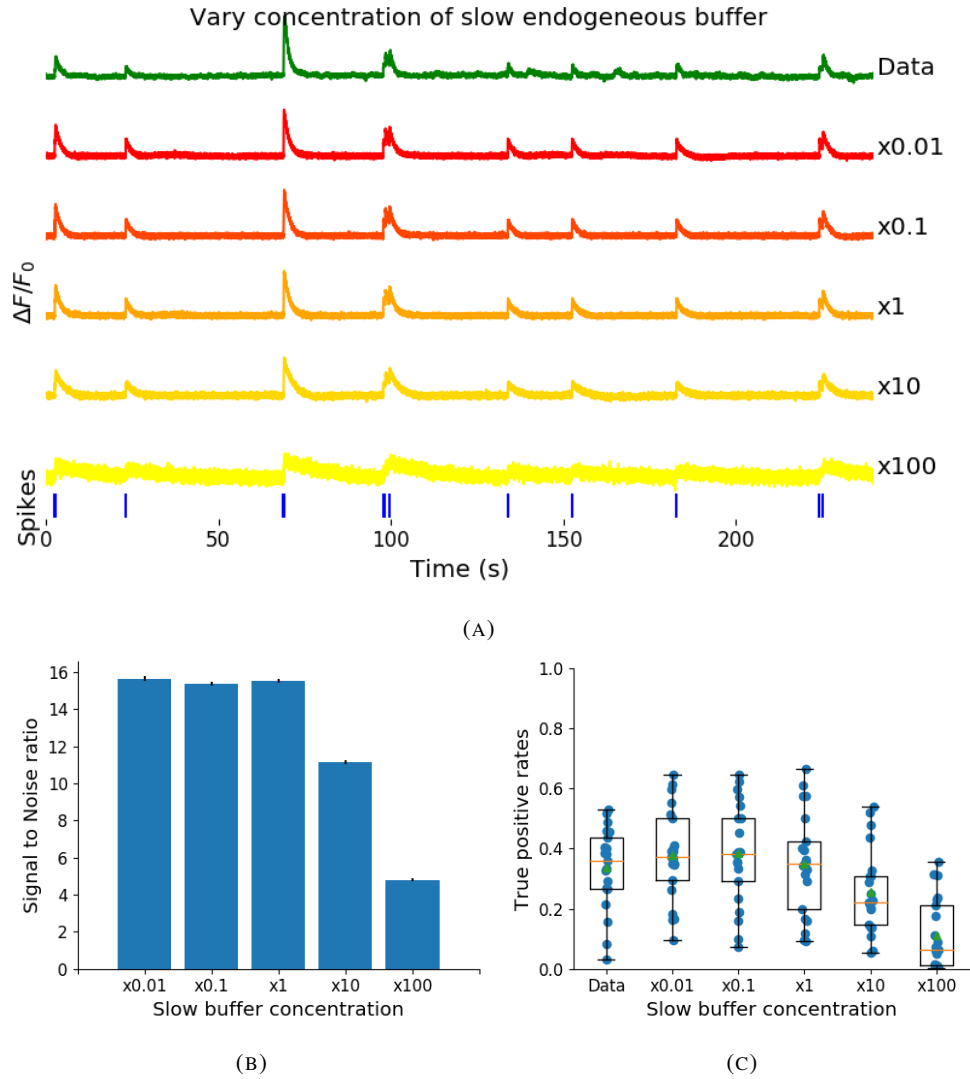
FIGURE 1.6: (a) An example trace for each of the five perturbed values for the concentration of immobile endogenous buffer. (b) The signal-to-noise ratio of the modelled fluorescence traces using each of the four perturbed values, and the experimental value. The lower values for the immobile buffer produce the same SNR as the experimental value. But the higher perturbed values produce fluorescence traces with a lower SNR. (c) The true-positive rates of the deconvolution algorithm's predictions when inferring from the observed data, and inferring from modelled traces using the perturbed and experimental values.

614 tightly packed groups of spikes will be more difficult to infer than isolated spikes. However,

615 the increasing amplitude of the fluorescence trace with increasing frequency suggests that

616 some spike inference algorithm could be developed based on this amplitude.
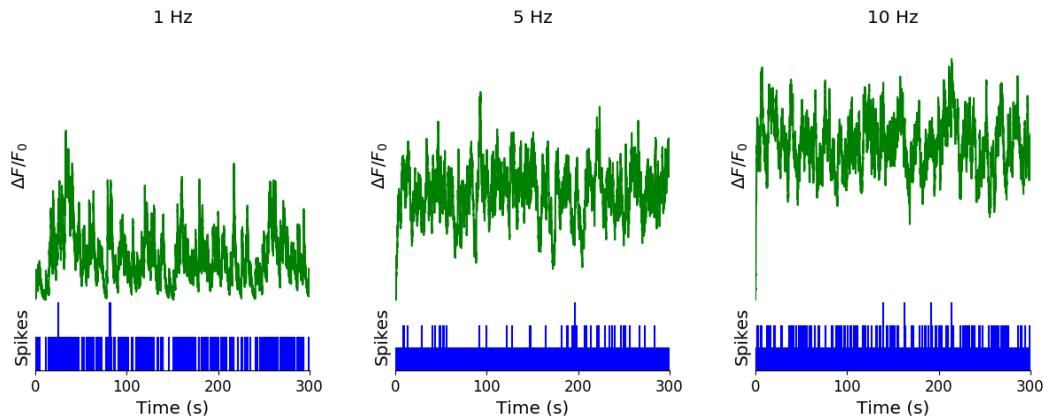
FIGURE 1.7: **Simulating fluorescence traces at different firing rates** Example modelled traces created using simulated spike trains with a mean firing rate of 1Hz (left column), 5Hz (middle column), and 10Hz (right column). Note the difference in amplitude with different mean firing rates.
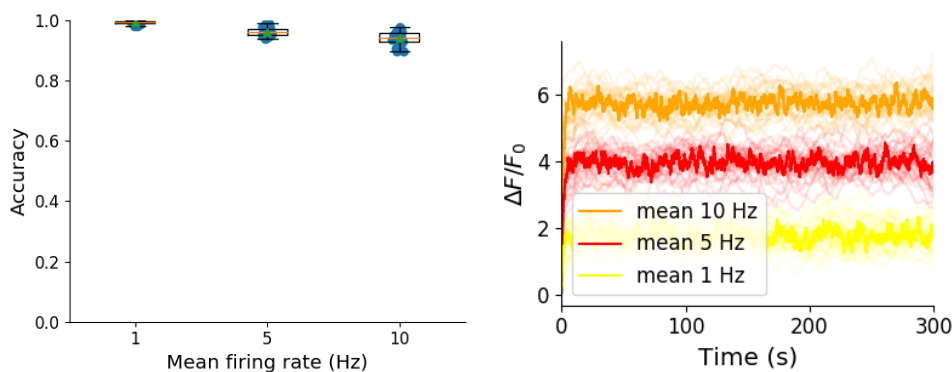
FIGURE 1.8: **Inference quality and $\Delta F/F_0$ vs Firing rate** (left) The spike inference accuracy when applied to 30 traces created using simulated spike trains with mean firing rates of 1, 5, and 10 Hz. (right) The mean $\Delta F/F_0$ across those 30 traces for each frequency.

1.3 Discussion

618 We designed a biophysical model for the changes in free calcium and bound calcium con-

619 centrations within a cell soma with a fluorescent calcium indicator. We used this model to

620 model the fluorescence trace resulting from a spike train in this cell. We fit the free parame-

621 ters of the model by matching the power spectrum and amplitude of fluorescence traces with

622 simultaneously measured spike trains. We inferred spikes from real fluorescence traces and

modelled fluorescence traces, and measured the quality of the spike inference in both cases. We found that the spike inference quality was similar in both cases. We perturbed the concentration of the calcium buffers in the model, and the binding/unbinding rates of those buffers in the model, and measured the effect on the signal-to-noise ratio (SNR) of the modelled fluorescence traces and the spike inference quality.

For the fluorescent calcium indicator, we found that any large perturbation away from the experimental value led to a reduction in SNR, and spike inference quality. For the binding/unbinding rates, we kept the ratio of these rates constant, but altered their values in parallel. The lower values caused a reduction in SNR, and a reduction in spike inference quality. For the endogenous buffer concentration, an increase above the experimental value caused a reduction in SNR and spike inference quality.

Although the model produced visually similar time series to the real data, there were a few aspects it did not capture. First, the real data featured some low-frequency components that did not appear related to the spike events. These were not captured by the models we used in this study, but could be added in future by adding a suitable low-frequency term to the resulting time series. Second, the real data seemed to have some nonlinearities not captured in the model, for example the response to two nearby spikes was greater than expected from the linear sum of two single spikes. This may be due to the co-operative binding of Calmodulin to calcium, which gives calmodulin a supra-linear sensitivity to calcium concentration. The model, in contrast behaved much more linearly, but could be extended in future to include such nonlinearities. Third, in the real data the fluorescence peak amplitude seemed to vary from spike to spike, even for well-isolated spike events. However in our model we assumed each spike lead to the same fixed-amplitude injection of calcium to the cell, leading to much greater regularity in fluorescence peak amplitudes. This variability could be added in future versions of the model by making the injected calcium peak a random variable. Fourth, we modelled the soma as a single compartment, but in reality there is likely a non-uniform spatial profile of calcium concentration. This may matter because some endogenous buffers might access calcium right as it influxes from the extracellular space, whereas the majority of the fluorescence signal is more likely coming from the bulk of the cytoplasm. Future models could attempt to model these spatial dependencies to assess whether they affect the overall spike inference procedure.

As well as the optimised parameters, the model has 14 fixed parameters than can be changed to simulate different types of calcium indicators. This model could be used to test the theoretical performance of proposed new types of calcium indicator. The model could

also be used by developers of spike inference algorithms to test the effects of changing calcium indicator parameters on spike inference, or to test the affects of changing spiking characteristics on spike inference. For example, high firing rate vs low firing rate, or bursting vs no bursting. Given the increasing amplitude of the fluorescence trace with increasing mean firing rate, it would be possible to build a spike inference algorithm on this principle at least in part.

Our model has already been used as a tool by our colleagues, for simulating fluorescence traces in response to cells that can fire with a continuous rate between 10 and 20Hz, but do not always do so. Our colleagues found that a combination of the amplitude and the variance of the simulated fluorescence trace was the best indicator of firing rate. For example, when a cell was not firing, the amplitude and variance of the fluorescence trace was relatively low. When the cell fired with a low firing rate $\sim$ 1Hz, the mean amplitude was still low but the variance of the fluorescence trace was high, and for high firing rate $\sim 10 - 20$Hz, the fluorescence amplitude was high, and the variance was low. In this way, our model may be useful for investigating firing rates underlying real fluorescence traces in response to cells which can fire in these rage ranges.

A recent paper by Greenberg et al (2018) described a biophysical model for spike train inference called the 'Sequential binding model'. Similar to our model, this model included parameters for two types of endogenous buffer. But this model also included dynamics for calcium binding to and unbinding from these endogenous buffers. Furthermore, this model included dynamics for calcium binding to and unbinding from the four binding sites present on a GCaMPs6 molecule. In the accuracy measurements specified in that paper, this model performed better than the MLspike algorithm, which is also partially a biophysically model, and it performed better than the constrained non-negative deconvolution algorithm. The sequential binding model also biophysically interpretable parameters, and its fitted parameters for quantites such as buffering capacity and calcium influx upon action potential firing fall in line with experimental values (Greenberg et al., 2018). Biophysical models like this appear to be the way forward for spike inference algorithms.

1.4   Methods

1.4.1   Calcium dynamics model

We wrote a biophysical model of the calcium dynamics within a cell soma. When a neuron fires an action potential, voltage-dependent calcium ion-channels open up that allow a

689 current of Ca^{2+} to flow into the neuron (Koch, 1999). The increase in the free calcium ion

690 concentration inside of the cell, along with changes in the concentration of potassium and

691 sodium, causes the change in cell membrane potential, which must be depolarised. The de-

692 polarising process consists of free calcium ions leaving the cell through open ion channels,

693 or binding to molecules within the cell called buffers, or calcium storage by organelles such

694 as the endoplasmic reticulum. A diagram illustrating the cell, its channels, and its buffers

695 can be seen in figure 1.1A. There are several different types of calcium buffer, each with

696 different dynamics and different concentrations within different types of excitable cell. The

697 fluorescent calcium indicator is another calcium buffer, with the useful property that when it

698 is bound to a calcium ion, the bound molecule may become excited by a photon and release

699 a photon in return. This is what creates the fluorescence. After the action potential has taken

700 place, the free calcium concentration within the cell will return to a baseline level (Maravall

701 et al., 2000).

702   We modelled the the dynamics of five molecular concentrations,

703   • Free calcium ion concentration, $[Ca^{2+}]$

704   • Fluorescent indicator bound calcium, $[BCa]$

705   • Endogenous mobile buffer bound calcium, $[ECa]$

706   • Endogenous immobile buffer bound calcium, $[ImCa]$

707   • Excited buffered calcium, $[BCa^*]$

The term 'buffering' refers to free calcium ions coming into contact with buffer molecules and binding together. Diagrammatically:

$$[X][Ca^{2+}] \underset{k_{Xb}}{\overset{k_{Xf}}{\rightleftharpoons}} [XCa]$$
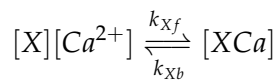
708 where $[X]$ represents any buffer molecule, and k_{X_f} and k_{X_b} represent the binding and un-

709 binding (dissociation) rates in units of per molar concentration per second ($\mathrm{M}^{-1}\,\mathrm{s}^{-1}$) and per

710 second (s^{-1}) respectively. The speed of this chemical reaction is determined by the binding

711 and unbinding rates.

712   There are a number different endogenous buffers in any neuron. Which buffers are

713 present, and the buffers' concentrations vary from cell to cell. In order to capture the ef-

714 fects of mobile and immobile endogenous buffers without introducing several parameters,

715 they were modelled as two buffers. One representing mobile buffers and the other represent-

716 ing immobile buffers. Each with their own binding and unbinding rates.

The fluorescent calcium indicator behaves similarly to the other calcium buffers. The calcium is buffered by the indicator in the same way. But an indicator bound calcium molecule can become excited by absorbing the energy from a photon. An excited indicator bound calcium molecule can then release a photon to go back to its 'relaxed' state.

$$[B][Ca^{2+}] \underset{k_{Bb}}{\overset{k_{Bf}}{\rightleftharpoons}} [BCa] \rightsquigarrow [BCa^*]$$
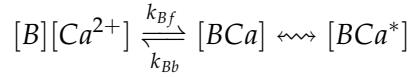
₇₁₇ The released photons are captured by a photon collector. This gives us the fluorescence trace.

₇₁₈ Ignoring the baseline level of free calcium in a neuron, the system of equations we used

₇₁₉ to model all of these interactions is as follows:

$$\begin{aligned}
\frac{d[Ca^{2+}]}{dt} =& k_{Bb}[BCa] + k_{Eb}[ECa] + k_{Imb}[ImCa] \\
& - k_{Bf}[B][Ca^{2+}] - k_{Ef}[E][Ca^{2+}] - k_{Imf}[Im][Ca^{2+}] \\
& + \beta([Ca_0^{2+}] - [Ca^{2+}])
\end{aligned} \tag{1.1}$$

$$\frac{d[BCa]}{dt} = k_{Bf}[B][Ca^{2+}] - k_{Bb}[BCa] + r[BCa^*] - e[BCa] \tag{1.2}$$

$$\frac{d[ECa]}{dt} = k_{Ef}[E][Ca^{2+}] - k_{Eb}[ECa] \tag{1.3}$$

$$\frac{d[ImCa]}{dt} = k_{Imf}[Im][Ca^{2+}] - k_{Imb}[ImCa] \tag{1.4}$$

$$\frac{d[BCa^*]}{dt} = \eta[BCa] - r[BCa^*] \tag{1.5}$$

₇₂₀ where $[Ca_0^{2+}]$ is the baseline calcium concentration within the cell soma, β is a rate defining

₇₂₁ how quickly free calcium enters or leaves the cell in the absence of an action potential, η

₇₂₂ is the excitation rate for indicator bound calcium, r is the photon release rate for the excited

₇₂₃ indicator bound calcium, and f and b are used to indicate the forward and backward rates

₇₂₄ for chemical reactions respectively. The excitation rate defines the proportion of indicator

₇₂₅ bound calcium that becomes excited at each time step. The photon release rate defines the

₇₂₆ proportion of excited indicator bound calcium that releases a photon and returns to its relaxed

₇₂₇ state at each time step. An action potential is modelled as a discontinuous increase in the free

₇₂₈ calcium concentration to an appropriate value (Maravall et al., 2000).

₇₂₉ Note that each of the three pairs of binding and unbinding terms in the first equation has a

₇₃₀ corresponding pair in one of the subsequent three equations. Binding removes a free calcium

₇₃₁ molecule and adds a bound calcium molecule, and unbinding does the opposite.

₇₃₂ When using this model to simulate a fluorescence trace, the system of equations above are

first solved over a period of 25s without action potentials. This lets each of the five tracked chemical concentrations reach their steady state. Then we use the given spike train and the parameters to model the fluorescence trace.

Note that since the model has no spatial component, the mobile and immobile buffers only differ in their binding and unbinding rates.

Photon release & capture

We used a simple model for the photon release. The number of photons released at each time step was controlled by the number of excited indicator bound calcium molecules in the cell and a parameter called the 'release rate'. The release rate is an optimised free parameter of the model.

As for the photon capture, in two-photon excitation microscopy the photons scattered by the fluorescent indicator get scattered in all directions. Therefore the number of photons detected is stochastic. This made the process for capturing photons the natural source of noise in the system. The number of photons captured, and therefore the intensity of the fluorescence, is modelled using a binomial distribution. The number of photons released was used as the number of trials. The probability of success, or 'capture rate' was a free parameter of the model that we optimised.

1.4.2   Parameter optimisation

The free parameters of the model are as follows:

Calcium rate, β Controls how quickly the concentration of free calcium will be driven to the baseline concentration.

Capture rate, p The average proportion of photons captured by the photon detector.

Excitation rate, η The number of indicator bound calcium molecules that become excited by photon bombardment at each time step.

Release rate, r The number of excited indicator bound calcium molecules that release a photon at each time step.

To optimise the free parameters given a fluorescence trace, we applied the following procedure:

1. The frequency power spectrum of the trace was measured.

2. The power spectrum was smoothed using a boxcar smoother (aka. sliding average, box smoother).

3. The log of the smoothed power spectrum was measured.

4. Use the model to create a modelled fluorescence trace.

5. Apply steps 1, 2, and 3 to the modelled fluorescence trace.

6. Calculate the root mean squared difference between the log power of the actual fluorescence trace, and the log power of the modelled fluorescence trace.

7. Calculate the root mean squared difference between the actual fluorescence trace and the modelled fluorescence trace.

8. Use an optimisation algorithm to reapply this process, attempting to minimize the sum of the two root mean squared differences at each iteration.

Using the root mean squared difference of the log power spectra as part of the objective function forces the model to match the noise frequency of the actual fluorescence. Using the root mean squared difference of the traces themselves forces the model to match the amplitude of the fluorescence trace more accurately.

In order to minimise the objective function, a suite of meta-heuristic optimisation (aka. black-box optimisation) algorithms were implemented on each of the traces in the dataset. These methods were chosen because they don't require a gradient for the objective function (gradient-free) and they are particularly useful for minimising stochastic objective functions like the one we used here. The free parameters were optimised for each individual fluorescence trace. The most successful method for each trace was recorded. The method that was most often successful was probabilistic descent, and the second most successful method was generating set search. Both of these methods are examples of pattern search. These two methods were the best optimisers on about 75% of the traces in the dataset.

Although this optimisation procedure minimises the value of the optimisation function, the value never reaches zero for a number of reasons. Firstly, the fluorescence traces carry low frequency fluctuations that cannot be captured by the model. Secondly, the model assumes that the process of calcium binding to the fluorescent indicator is linear in time (see equation 1), but there are more complicated dynamics involved here. Fluorescent calcium indicators are often built upon the calcium binding protein called 'calmodulin'. This protein has four calcium binding sites. These sites are locally split into two pairs. Each pair has a different

Parameter	Description	Value	Source
baseline	The baseline concentration of free calcium within the cell soma	4.5×10^{-8}M	(Maravall et al., 2000)
cell radius	The radius of the soma (assumed to be spherical)	10^{-5}M	(Fiala and Harris, 1999)
endogenous	The concentration of endogenous mobile buffer within the cell soma	10^{-4}M	(Faas et al., 2011)
frequency	The frequency at which the spike trains are sampled.	100Hz	
immobile	The concentration of endogenous immobile buffer within the cell soma	7.87×10^{-5}M	(Bartol et al., 2015)
indicator	The concentration of fluorescent indicator within the cell soma	10^{-4}M	(Maravall et al., 2000)
k_{Bb}	The unbinding rate of the fluorescent calcium indicator	160s^{-1}	(Bartol et al., 2015)
k_{Bf}	The binding rate of the fluorescent calcium indicator	$7.77 \times 108\text{s}^{-1}\text{M}^{-1}$	(Bartol et al., 2015)
k_{Eb}	The unbinding rate of the endogenous mobile buffer	10^4s^{-1}	(Bartol et al., 2015)
k_{ef}	The binding rate of the endogenous mobile buffer	$10^8\text{s}^{-1}\text{M}^{-1}$	(Bartol et al., 2015)
k_{Imb}	The unbinding rate of the endogenous immobile buffer	524s^{-1}	(Bartol et al., 2015)
k_{Imf}	The binding rate of the endogenous immobile buffer	$2.47 \times 10^8\text{s}^{-1}\text{M}^{-1}$	(Bartol et al., 2015)
peak	The increase in free calcium concentration within the cell induced by an action potential	2.9×10^{-7}M	(Maravall et al., 2000)

TABLE 1.1: **Fixed parameters** A table of the parameters fixed before optimising the model. The values of these parameters could be changed to model different fluorescent calcium indicators.

affinity for calcium, and the affinity of the binding sites is affected by the occupancy of the other binding sites (Kilhoffer et al., 1992). So the calcium to calcium indicator binding process is non-linear, but the model does not take this into account.

Fixed parameters

As well as the optimised parameters mentioned in section 1.4.2, the model also has thirteen fixed parameters. Please see table 1.1 for details of these parameters and their values. In an application of the model, these parameters can be changed in order to model any given fluorescent calcium indicator.

1.4.3 Julia

The programming language used to write and execute the model was 'Julia'. Julia is a dynamic programming language designed for technical computing. Julia was designed specifically to provide a convenient high-level dynamic language similar to MATLAB, or Python, with improved performance. Julia's type system and Julia's direct interfaces with C and Fortran allow this aim to be achieved (Bezanson et al., 2012). The Julia version of the 'Sundials' package for ODE solving was used to solve the system of equations above. The `BlackBoxOptim.jl` package for Julia was used to perform the optimisation.

1.4.4 Spike inference

We used spike inference algorithms to compare the quality of spike inference using the modelled traces to the quality of spike inference using the observed traces. We also used the spike inference algorithms to assess the effect of parameter perturbation on the spike inference. Three algorithms were used:

Constrained non-negative deconvolution algorithm (aka Pnevmatikakis algorithm) This algorithm uses a constrained version of non-negative Weiner deconvolution to infer a calcium signal and a 'spiking activity signal' from the fluorescence trace (Vogelstein et al., 2010; Pnevmatikakis et al., 2016). The spiking activity signal is a non-negative vector of real numbers reflecting the cell's activity rather than an actual spike train. We inferred a spike train by choosing an optimised threshold for the spiking activity signal. Whenever the spiking activity signal exceeded that threshold, an action potential was inferred. The threshold was optimised by minimising the difference between the number of spikes observed and the number of spikes predicted.

ML-Spike algorithm This algorithm uses a generalised version of the Viterbi algorithm to return the spike train that maximises the likelihood of producing the given fluorescence trace. The Viterbi algorithm is an algorithm for estimating the most likely sequence of hidden states resulting in a sequence of observed states in a discrete-time finite-state Markov process (**forney**). In this case, each hidden state is defined by the presence or absence of an action potential, and each observed state is the value of the fluorescence trace at each time step. This algorithm assumes that the concentration of calcium within the cell will decay to a drifting baseline, rather than a fixed baseline (Deneux et al., 2016).

Online Active Set method to Infer Spikes (OASIS) This algorithm is once again based on an auto-regressive model of the fluorescence trace, but can be generalised to any order. The algorithm itself is a generalisation of the pool adjacent violators algorithm (PAVA) that is used in isotonic regression. The OASIS algorithm works through the fluorescence trace from beginning to end, this combined with the speed of the algorithm means that it could be used for real-time online spike inference (Friedrich and Paninski, 2016). Given a fluorescence trace, the algorithm will return the most likely spike train and an inferred denoised fluorescence signal.

In order to quantify the quality of spike inference for a given algorithm, we ran that algorithm on all of the fluorescence traces in dataset number eight of the spike finder datasets. Then we measured some binary classification measures on the results. These measures included

- Accuracy

- True positive rate (aka recall, sensitivity, hit rate)

- True negative rate (aka specificity)

- Precision

- Negative predicted value

- False negative rate (aka miss rate)

- False positive rate (aka fall-out)

- False discovery rate

- False omission rate

In making these measurements, we allowed a tolerance of two subsequent time bins for spike prediction. For example, the spike train data is a vector of 0s and 1s, with one element for each time bin. A '0' denotes inactivity, a '1' denotes the presence of at least one action potential. The inferred spike trains produced by the spike inference algorithms take the same form. In our analysis, if a spike appeared in the inferred spike train up to two time frames after a spike in the observed spike train, that spike was considered correctly inferred i.e. a true positive. However, once a spike in the inferred spike train was matched to a spike from the observed spike train, the inferred spike could not be matched to another observed spike.To illustrate, if two spikes were inferred in the two time bins following an isolated observed

spike, the first inferred spike was considered correctly inferred, but the second inferred spike was considered incorrectly inferred, i.e. a false positive.

The most useful measure was the true positive rate. This is because the spiking is sparse and this measurement is sensitive to the number of spikes observed and inferred, but is not affected by the true negative or false negative rates. After optimising the parameters for each fluorescence trace we measured the spike inference quality for the observed fluorescence traces, and compared this to the spike inference quality for the modelled traces.

When measuring the spike inference quality for higher frequency spike train $(1 - 10\text{Hz})$, we used the accuracy as our binary classification measure. At these frequencies the variance of the fluorescence trace was much higher than for sparser spiking regimes, therefore we wanted to take into account the number of false negatives inferred by the algorithm.

Comparing spike inference quality

In order to compare spike inference quality we had to use methods for comparing samples. When comparing the true positive rate distributions arising from two different datasets, or two different algorithms on the same dataset, we compared the distributions using a paired t-test.

1.4.5 Perturbation analysis

In order to measure the sensitivity of spike inference to changes in a given model parameter, we perturbed the parameter and compared the quality of spike inference with the perturbed parameters to the quality of spike inference with the experimental or optimised parameters. In order to maximise the possibility of observing a difference due to the perturbation, we perturbed the chosen parameter by a relatively large amount. For example, the experimental value for the molar concentration of the fluorescent indicator within the cell was 10^{-4}M (Maravall et al., 2000). The perturbed values used for this parameter were 10^{-2}M, 10^{-3}M, 10^{-5}M, and 10^{-6}M. The quality of the inference was compared by measuring the true positive rate for each perturbed value and using a t-test to compare the distributions of the results.

This analysis was performed firstly without any optimisation of the free parameters for use with the perturbed parameters. Then the analysis was performed after the optimised parameters for each perturbed value were calculated.

1.4.6 Signal-to-noise ratio

To assess the effect of perturbation on the modelled traces, we measured and compared the signal to noise ratio (SNR) on each of the modelled traces. We calculated the SNR as the peak change in fluorescence divided by the standard deviation of the baseline fluctuation of the fluorescence trace (Tada et al., 2014). We measured these values by running the model on a spike train consisting a long period of inactivity followed by one action potential. We ran the model on this spike train one hundred times. We then measured the mean change in fluorescence and standard deviation of baseline activity across the one hundred modelled fluorescence traces, and calculated the SNR.

1.4.7 Data sources

All of the data used in this project was sourced from the 'Spike Finder' project (spikefinder.codeneuro.org). The data consisted of a collection of datasets with simultaneously measured fluorescence traces and action potentials (Berens et al., 2018).

Chapter 2

Functional networks expand across anatomical boundaries as correlation time-scale increases

UNIVERSITY OF BRISTOL

Abstract

Engineering

Department of Computer Science

Doctor of Philosophy

Investigating, implementing, and creating methods for analysing large neuronal ensembles

by Thomas J. DELANEY

Decades of research has established that correlations play a crucial role in representing sensory information. One drawback associated with the recent improvement in recording technology and consequent large datasets is the difficulty in analysing higher order correlations in large neuronal ensembles. One benefit of these datasets that has not yet been explored is the opportunity to compare correlations within anatomical regions to correlations across anatomical regions. In this work, we measured correlations between neurons residing in nine different brains regions in three awake and behaving mice. Using the these correlation measurements, we created weighted undirected graph networks and applied network science methods to detect functional communities in our neural ensembles. We compared these functional communities to their anatomical distribution. We repeated the analysis, using different timescales for our correlation measurements, and found that functional communities were more likely to be dominated by neurons from a single brain region at shorter timescales (< 100ms).

2.1   Introduction

Decades of research has established that correlations play a crucial role in representing sensory information. For example, the onset of visual attention has been shown to have a greater affect on the correlations in the macaque V4 region than on the firing rates in that region (Cohen and Maunsell, 2009). Recent findings show that spontaneous behaviours explain correlations in parts of the brain not associated with motor control (Stringer et al., 2019). In order to understand the brain, we must understand the interactions between neurons.

Because of limitations in recording technology almost all research has explored correlations between neurons within a given brain region. Relatively little is known about correlations between neurons in different brain regions. However, the recent development of 'Neuropixels' probes (Jun et al., 2017) has allowed extracellular voltage measurements to be collected from multiple brain regions simultaneously routinely, and in much larger numbers than traditional methods. In this project we used a publicly-available Neuropixels dataset to analyse correlations between different brain regions (Stringer et al., 2019).

A drawback associated with the improvement in recording technology is an increase in the difficulty of analysing these data. For example, analysing the ith order interactions of N neurons generally requires estimation of N^i parameters. A number that becomes astronomical for large N. New methods are required for analysing these new large datasets. We attempted to address this requirement in this piece of research by applying a cutting-edge network science community detection method to neural data.

Another unexplored area of research is the changes in cell interactions at different timescales. Most studies focus on quantifying interactions at a given timescale. But neurons may interact differently, or may interact with different neurons, at different timescales. Here we explore correlated communities of neurons at different timescales.

In this work, we measured correlations between binned spike counts from neurons from nine different regions of the mouse brain. Using these correlation measurements, we created weighted undirected graphs or networks with each node representing a neuron. We used newly invented network methods to detect communities in these networks. We compared these detected communities to the anatomical division of the cells. We also measured the conditional correlation and signal correlation of between cell spike counts, conditioning on the subjects' behaviour. We repeated the network analysis using these measurements. We also repeated these analyses using different timescales for measuring the correlations.

2.2 Results

Note that in the following text, we refer to the correlation coefficient between two sequences of spike counts from two different cells as the *total correlation*. We refer to the correlation between spike counts in response to a certain stimulus as the *spike count correlation*, and we refer to the correlation between mean or expected responses to different stimuli as the *signal correlation*. These definitions correspond to those given in Cohen et al (2011) (Cohen and Kohn, 2011).

2.2.1 Average correlation size increases with increasing time bin width

First we inspected the affect of time bin width on total correlations. We know that using short time bins results in artificially small correlation measurements (Cohen and Kohn, 2011), so we expected to see an increase in correlation amplitude with increasing time bin width. That is exactly what we observed. Taking 50 cells at random, we calculated the total correlation between every possible pair of these cells, using different time bin widths ranging from 0.005s to 3s. We found that the longer the time bin width, the greater the correlations (see figure 2.2a).
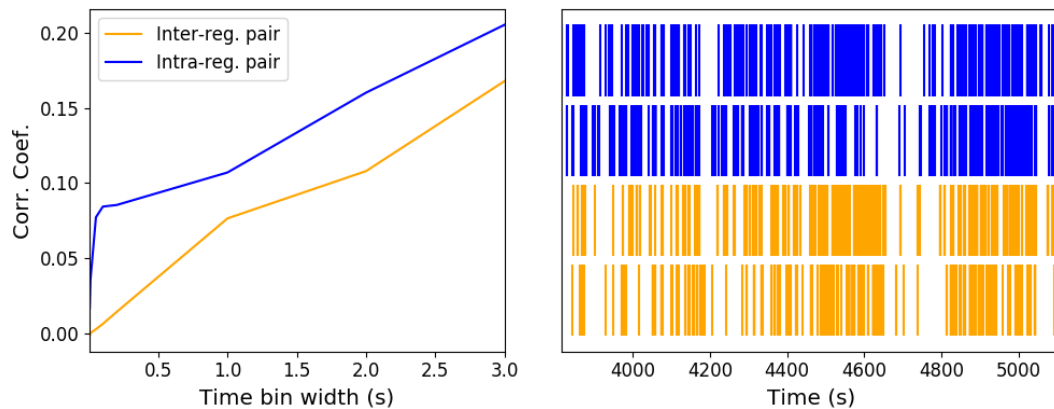
FIGURE 2.1: (left) An examples of the correlation coefficients between two different pairs of cells, one where both cells are in the same brain region (intra-regional pair), and one where both cells are in different brain regions (inter-regional pair). The correlation coefficients have been measured using different time bin widths, ranging from 5ms to 3s. Note the increasing amplitude of the correlations with increasing bin width. (right) A raster plot showing the spike times of each pair of cells.

We also separated the positively correlated pairs from the negatively correlated pairs using the mean correlation of each pair across all bin widths (see section 2.5.2). We found that the positively correlated pairs become more positively correlated with increasing time bin

width, and the negatively correlated pairs become more negatively correlated with increasing time bin width (see figures 2.2b and 2.2c).

When taking the mean across all pairs, the positively correlated pairs dominate in terms of both number of pairs, and amplitude of correlations. Therefore the mean across all pairs is positive.

These results were observed in each of the three mouse subjects from which we had data.

For the sake of illustration, I have included two figures showing the correlation coefficients of two pairs of neurons, and a raster plot of the spikes of those cells (see figure 2.1).

2.2.2 Goodness-of-fit for Poisson and Gaussian distributions across increasing time bin widths

We wanted to investigate if the width of the time bin used to bin spike times into spike counts had an effect on the distribution of spike counts. We used the χ^2 statistic as a goodness-of-fit measure for Poisson and Gaussian (normal) distributions to the spike count of 100 randomly chosen neurons for a number of bin widths ranging from 0.01s to 4s. For the χ^2 statistic, the higher the value, the worse the fit. As you might expect, we found that that a Poisson distribution is the best fit for shorter time bins less than 0.7s in length. Then a Gaussian distribution is a better fit for time bins greater than 0.7s in length (see figure 2.2d).

We expected a Poisson distribution to be a better fit for shorter time bin widths because spike counts must be non-negative, therefore any distribution of spike counts with mass distributed at or close to 0 will be skewed. The distribution of spike counts is more likely to be distributed close to 0 when the time bin widths used to bin spike times into spikes counts are small relative to the amount of time it takes for a neuron to fire an action potential ($\sim$ 1ms in the case of non-burst firing neurons).

2.2.3 Differences between and inter- and intra- regional correlations decrease with increasing bin width

We investigated the differences in distribution between inter-regional correlations, i.e. correlations between neurons in different brain regions, and intra-regional correlations, i.e. correlations between neurons in the same brain region.

Firstly, we investigated these quantities for all possible pairs of $\sim$ 500 neurons taken from across all the 9 brain regions from which we had data. We distributed these neurons as evenly as possible across all of the regions, so that cells from one region would not dominate our data. We observed that the mean intra-regional correlations were always higher than the

(A) All pairs, positive and negative.

(B) All positively correlated pairs.

(C) All negatively correlated pairs.

(D) χ^2 test statistics as goodness-of-fit.
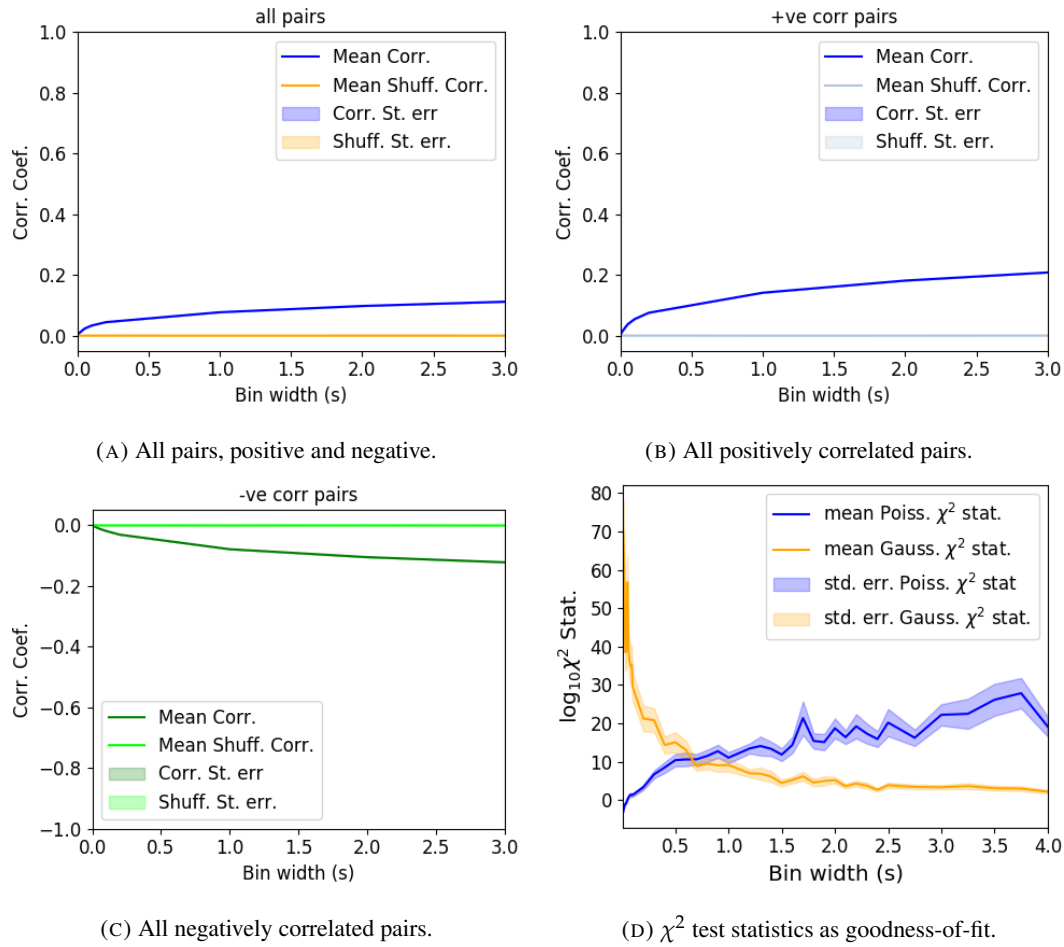
FIGURE 2.2: Mean correlation coefficients measured from pairs of 50 randomly chosen neurons. (a) All possible pairs, (b) positively correlated pairs, and (c) negatively correlated pairs. (d) Mean and standard error of χ^2 test statistics for Poisson and Gaussian distributions fitted to neuron spike counts.

mean inter-regional correlations for every value of time bin width used. We also observed that as the time bin width increased these mean correlations increased and the difference between the mean inter-regional and intra-regional correlations grew (see figure 2.3). Stringer et al. (2019) had a similar finding using the same data. They used only one value for the time bin width, 1.2s. Using this time bin width to bin spike times and measure total correlations, they found that the mean 'within-region' correlations were always greater than the 'out-of-region' correlations (Stringer et al., 2019).

Examples of the correlations of one intra-regional pair and one inter-regional pair can be seen in figure 2.1.

Secondly, we separated those pairs into intra-regional and inter-regional groups. We noted that the mean intra-regional correlations for a given region tended to be higher than the mean inter-regional correlations involving cells from that region. However, in contrast with our previous result, we noted that the difference between the mean intra-regional correlations
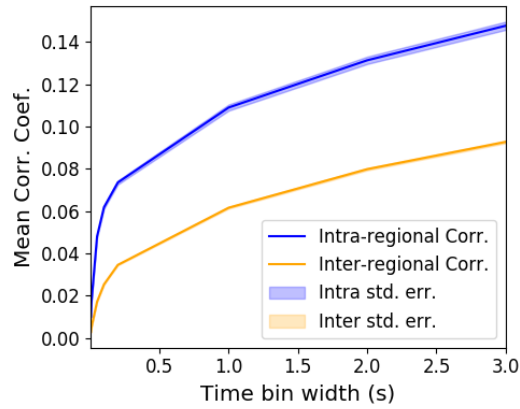
FIGURE 2.3: The mean intra-region and inter-region correlations using all possible pairs of ~ 500 neurons, spread across 9 different brain regions.

1023   and most highly correlated inter-regional correlations reduced as we increased the time bin

1024   width (see figures 2.4a and 2.4b).
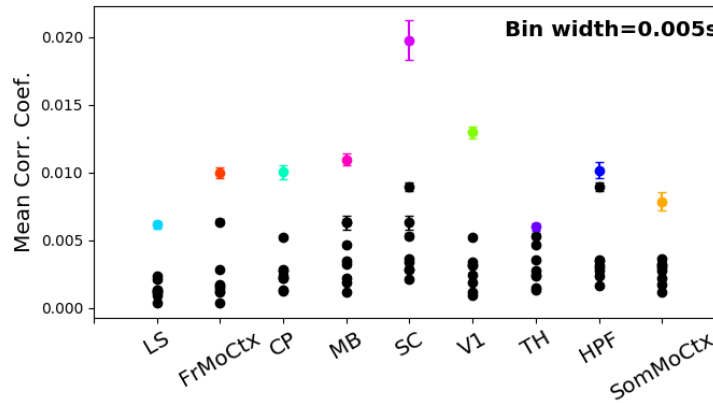
(A) Mean inter-regional and intra-regional correlations using a time bin width of 5ms.
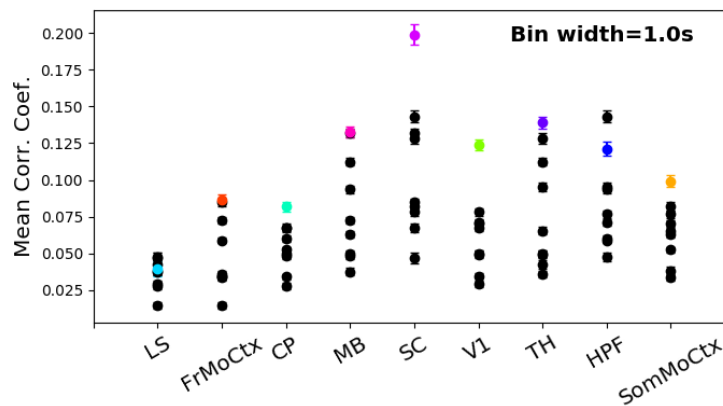
(B) Mean inter-regional and intra-regional correlations using a time bin width of 1s.

FIGURE 2.4: The mean intra-regional correlations (coloured dots) and mean inter-regional correlations (black dots) for a given region, indicated on the x-axis, for different time bin widths. Each black dot represents the mean inter-regional correalations between the region indicated on the x-axis and one other region. (a) shows these measurements when we used a time bin width of 5ms. (b) shows these measurements when we used a time bin width of 1s. Note that the difference between the mean inter-regional correlations and mean intra-regional correlations is smaller for 1s bins.

Finally, to see these regional mean correlations in a bit more detail, we displayed these data in a matrix of mean correlations (see figure 2.5), showing the mean intra-regional correlations on the main diagonal, and the mean inter-regional correlations off diagonal. Comparing a version of this figure created using a short time bin width of 5ms (figure 2.5a) and a version using a longer time bin width of 1s (figure 2.5b) we observed that the mean intra-regional correlations are always relatively high in comparison to the mean inter-regional correlations, but the mean correlations in some inter-regional pairs are relatively much higher when using the longer time bin width.

(A) Time bin width 0.005s.   (B) Time bin width 1s.
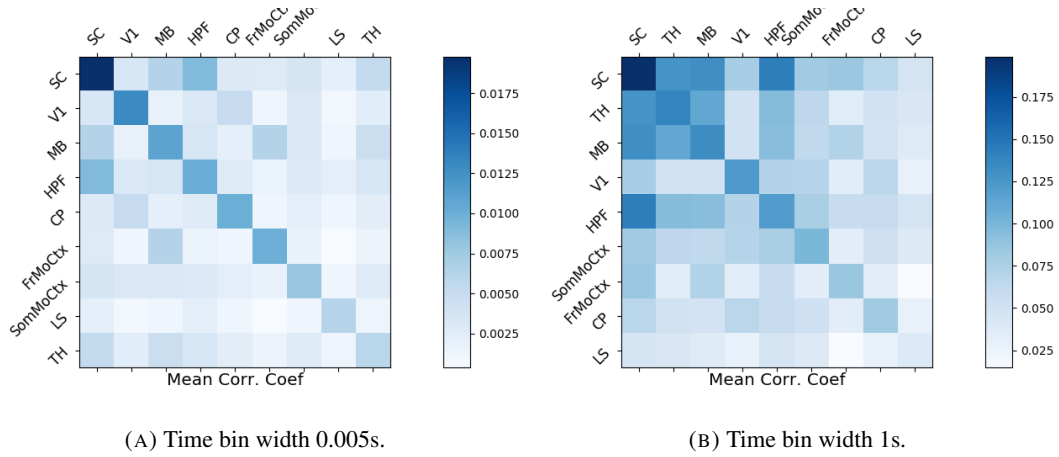
FIGURE 2.5: Mean inter-regional (main diagonal) and intra-regional (off diagonal) correlation coefficients. (a) Shows these measurements when spike times were binned using 5ms time bins. (b) Shows the same, using 1s time bins. Note that the matrices are ordered according to the main diagonal values, therefore the ordering is different in each subfigure.

These results were consistent across the three mouse subjects. But, the relative maginitudes of the mean intra-regional and inter-regional correlations were not consistent. For example, the region with the highest mean intra-regional correlations when using 1s bin widths for subject one is the superior colliculus (SC), but for subject two it is the midbrain (MB).

2.2.4   Connected and k-partite structure in correlation based networks reduces in dimension with increasing bin width

We used the correlation measurements to create weighted undirected graphs/networks where each node represents a neuron, and the weight of each edge is the pairwise correlation between those neurons represented by the nodes at either end of that edge. We aimed to find communities of neurons within these networks, and compare the structure of these communities to the anatomical division of those neurons. The first step of this process involved applying the 'spectral rejection' technique developed by Humphries et al (2019) (Humphries et al., 2019). This technique compares our data network to a chosen null network model, and finds any additional structure in the data network beyond that which is captured in the null network model (if there is any such structure). By comparing the eigenspectrum of the data network to the eigenspectrum of many samples from the null network model, this technique allows us to estimate the dimensionality of the additional structure in the data network, and gives us a basis for that vector space. The technique also finds which nodes contribute to this additional structure, and divides our data network into signal and noise networks. The details

of spectral rejection and node rejection can be found in sections 2.5.5 and 2.5.5 respectively, and a full overview can be found in (Humphries et al., 2019).
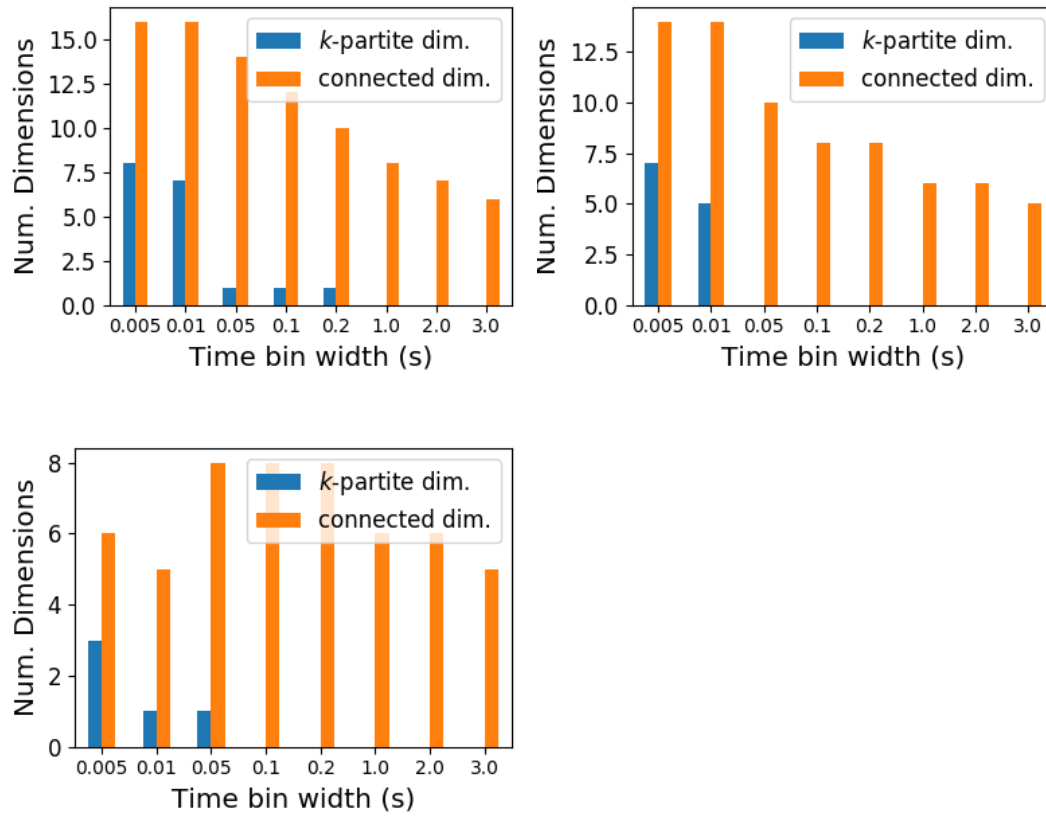
FIGURE 2.6: The number of dimensions in the *k*-partite and connected structure in the correlation based networks beyond the structure captured by a sparse weighted configuration null network model (see section 2.5.5), shown for different time bin widths. Note that the *k*-partite structure disappears for time bin width greater than 200ms for all three subjects. The dimension of the connected structure reduces with increasing bin width for 2 of the 3 subjects (top row).

We chose the sparse weighted configuration model (see section 2.5.5) as our null network model. This model matches the sparsity and the total weight of the original network but distributes the weight at random across the sparse network.

We applied the spectral rejection method to our networks based on total correlations using different values for the time bin width. We observed that for smaller time bin widths, our data networks had both *k*-partite structure, and community structure. As the width of the time bin increased, we found that the *k*-partite structure disappeared from our data networks, and the dimension of the community strcuture reduced in two of the three mice from which we had data (see figure 2.6).

(A) 5ms

(B) 5ms

(C) 1s

(D) 1s
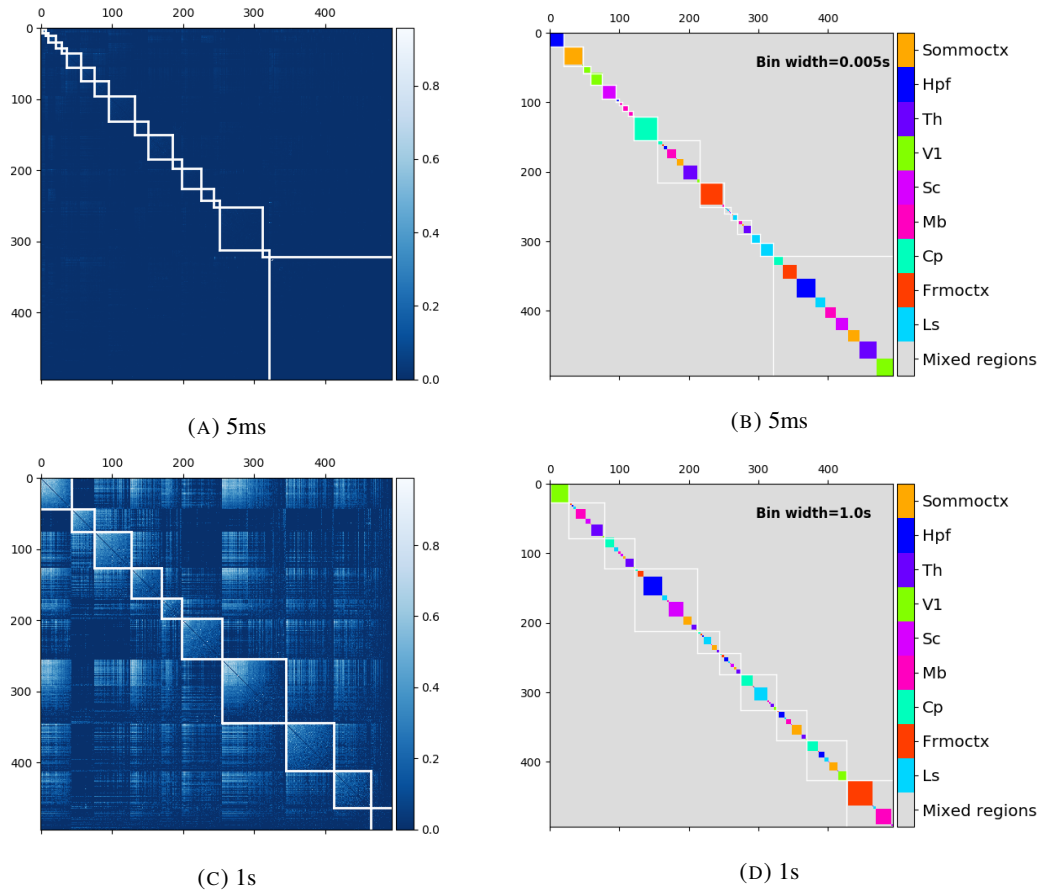
FIGURE 2.7: (a) and (c) Correlation matrices with detected communities indicated by white lines. Each off main diagonal entry in the matrix represents a pair of neurons. Those entries within a white square indicate that both of neurons are in the same community as detected by our community detection procedure. Matrices shown are for 5ms and 1s time bin widths respectively. (b) and (d) Matrices showing the anatomical distribution of pairs along with their community membership. Entries where both cells are in the same reigon are given a colour indicated on the colour bar. Entries where cells are in different regions are given the grey colour also indicated on the colour bar.

2.2.5   Detecting communities in correlation based networks

We applied the community detection procedure described in section 2.5.5 to our signal networks for our various time bin widths. As might be expected from the results described above in section 2.2.4, we detected a greater number of smaller communities at shorter time bin widths, and a smaller number of larger communities for longer time bin widths (see figure 2.7).

We also noticed that at short time bin widths the communities detected tended to be dominated by cells from one region. Whereas communities existing in networks created using wider time bin widths tended to contain cells from many different brain regions. More on this in the next section.

2.2.6 Functional communities resemble anatomical division at short timescales

In order to quantify the similarity of the communities detected to the anatomical division of the cells. We treated both the anatomical division and the communities as clusterings of these cells. We then used measures for quantifying the difference or similarity between clusterings to quantify the difference or similarity between the detected communities and the anatomical division. Details of these measures can be found in section 2.5.6 or in (Vinh, Epps, and Bailey, 2010).

We used two different types of measures for clustering comparison; information based measures (see section 2.5.6) and pair counting based measures (see section 2.5.6). We include one example of each in figure 2.8.

The variation of information is the information based measure included in figure 2.8a. This measure forms a metric on the space of clusterings. The larger the value for the variation of information, the more different the clusterings.

The adjusted Rand index is the pair counting based measure included in figure 2.8b. In contrast with the variation of information, the adjusted Rand index is a normalised similarity measure. The adjusted Rand index takes value 1 when the clusterings are identical, and takes value 0 when the clusterings are no more similar than chance.

(A) Variation of information

(B) Adjusted Rand index

FIGURE 2.8: (a) The variation of information is a measure of distance between clusterings. The distance between the anatomical 'clustering' and community detection 'clustering' increases with increasing time bin width. (b) The adjusted Rand index is a normalised similarity measure between clusterings. The anatomical and community detection clusterings become less similar as the time bin width increases.

Both measures indicated that the detected communities and the anatomical division of the cells were more similar when we used shorter time bins widths (see figure 2.8). This indicates that correlated behaviour in neuronal ensembles is more restricted to individual brain regions

at short timescales (< 250ms), and the correlated activty spreads out across brain regions over longer time scales.

2.2.7   Conditional correlations & signal correlations

In light of the excellent research of Stringer et al (2019) showing that spontaneous behaviours can drive activity in neuronal ensembles across the visual cortex and midbrain (Stringer et al., 2019), we decided to control for the mouse's behaviour when performing our analyses. My second supervisor, Dr Michael Ashby, suggested that our community detection process may be detecting communities across multiple brain regions at longer time scales due to aggregating neuronal activity driven by several spontaneous behaviours occuring during the time interval covered by a given time bin. A time bin of 1s, for example, could contain a spike count where those spikes were driven by different spontaneous behaviours. We aimed to investigate this possibility by applying our community detection analysis to conditional correlation measures.

We used the top 500 principal components of a video of the mouse's face as a measure of the mouse's behaviour (see section 2.4.2). We modelled the spike counts as a linear combination of the principal components using linear regression with ElasticNet regularisation (see section 2.5.3). Using this model, we quantified the expected spike count given the mouse's behaviour $E[X|Z_1, \ldots, Z_{500}]$.

We used these expected values to measure $\mathrm{cov}(E[X|Z], E[Y|Z])$, and we used that value, the covariance $\mathrm{cov}(X, Y)$, and the *law of total covariance* (see section 2.5.3) to measure $E[\mathrm{cov}(X, Y|Z)]$. Here X and Y represent spike counts from individual cells, and Z is shorthand for the 500 principal components mentioned above. The two components of the covariance, $\mathrm{cov}(E[X|Z], E[Y|Z])$ and $E[\mathrm{cov}(X, Y|Z)]$, represent a 'signal covariance' and expected value of a 'spike count covariance' respectively, analogous to the signal correlation and spike count correlation (Cohen and Kohn, 2011).

We examined the means of these components for different values of the time bin width (see figure 2.9). We observed a consistent increase in $E[\mathrm{cov}(X, Y|Z)]$ as the time bin width increased. But we saw different trends for $\mathrm{cov}(E[X|Z], E[Y|Z])$ for each mouse.

Using $\mathrm{cov}(E[X|Z], E[Y|Z])$ we measured the signal correlation, ρ_{signal}, and using $E[\mathrm{cov}(X, Y|Z)]$ we measured the event conditional correlation, $\rho_{X,Y|Z}$ (see section 2.5.3 for more details). We saw a consistent increase in $\rho_{X,Y|Z}$ as the time bin width increased, this corresponds to the result for $E[\mathrm{cov}(X, Y|Z)]$. We observed different trends for ρ_{signal} for each mouse, this corresponds to the result for $\mathrm{cov}(E[X|Z], E[Y|Z])$.
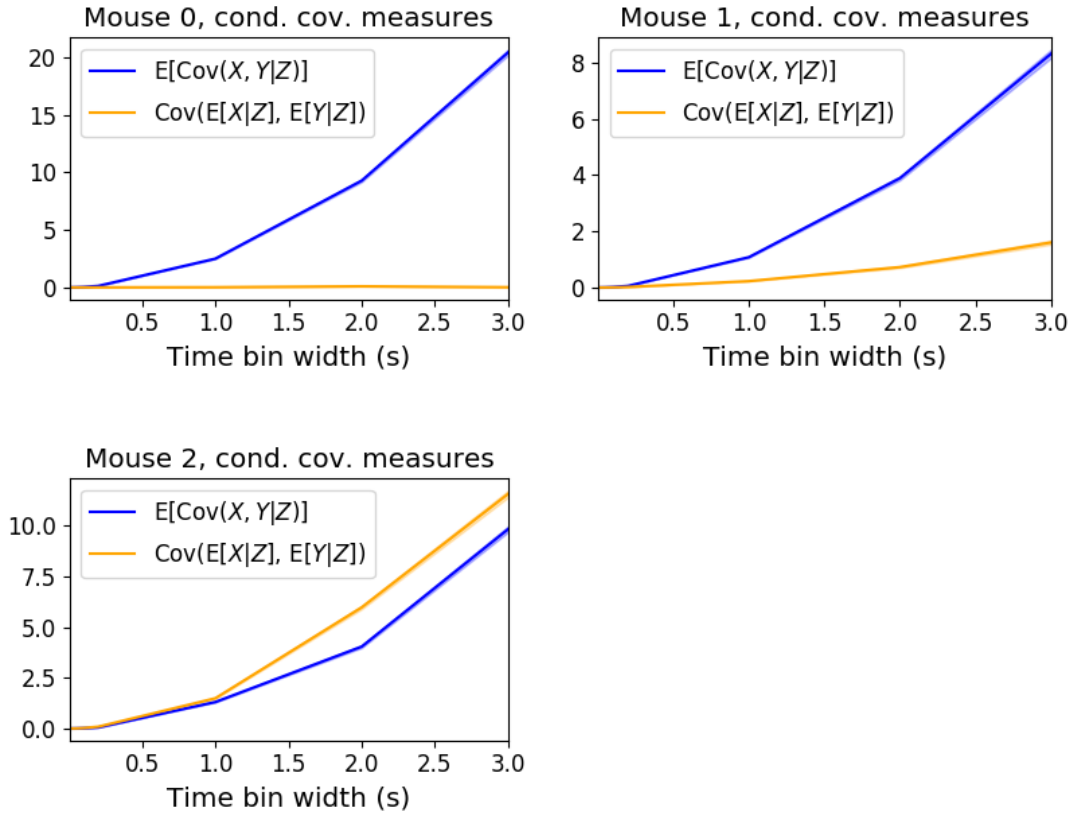
FIGURE 2.9: Comparing the components of the spike count covariance across different values for the time bin width. We observed a consistent increase in $E[\text{cov}(X, Y|Z)]$ as the time bin width increased. But we saw different trends for $\text{cov}(E[X|Z], E[Y|Z])$ for each mouse.

We applied our network noise rejection and community detection process to networks based on the spike count correlations $\rho_{X,Y|Z}$ and the signal correlations ρ_{signal}. We noted that the community detection on $\rho_{X,Y|Z}$ behaved similarly to the community detection on the total correlation. We can see this in figures 2.11a and 2.11b. At very short time bin widths, we detect more communities, and those communities often contain cells from one brain region only. At longer time bin widths, we detect fewer communities, and those communities tend to contain cells from multiple brain regions. When we examine the distance between (or similarity between) the anatomical division of the cells, and the detected communities we notice that the two clusterings are more similar at shorter time bin widths (see figure 2.12).

When we applied the network noise rejection and community detection process to the networks based on the signal correlations ρ_{signal} we found the number of communities we detected reduced with increasing time bin width. But the number of communities detected was less than that for the total correlations or the spike count correlations. The communities detected always tended to contain cells from multiple regions at both short and long
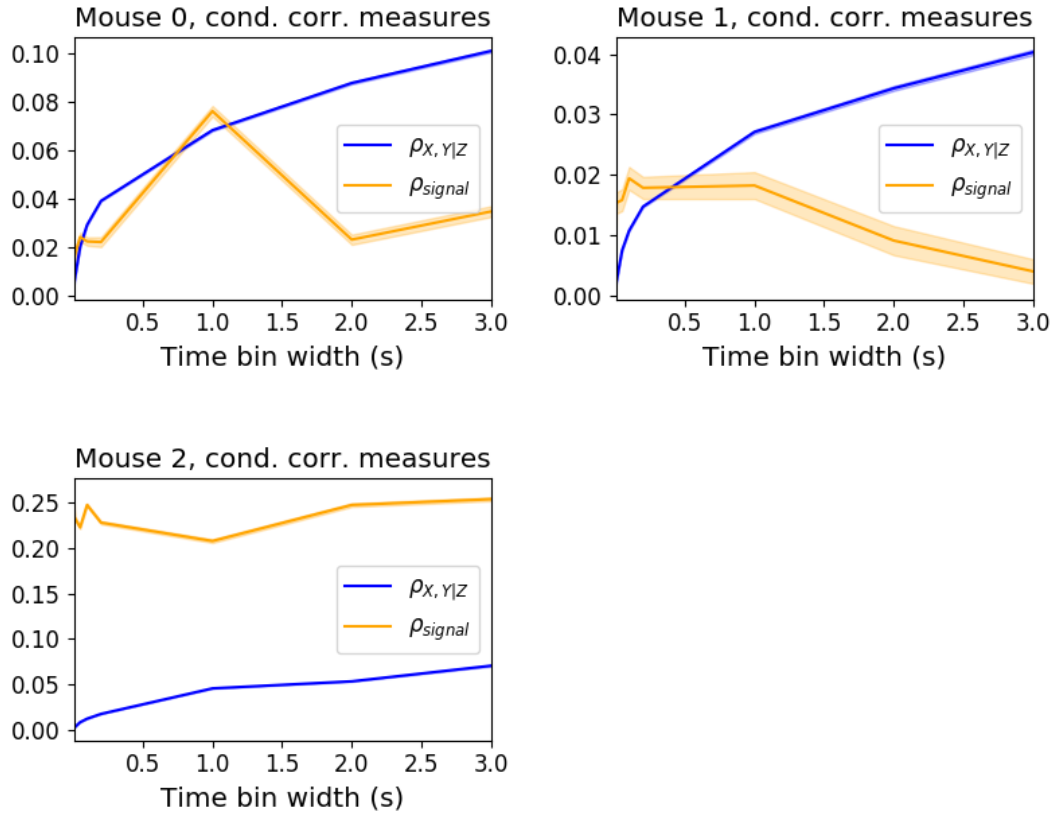
FIGURE 2.10: Comparing the components of the total spike count covari-
ance across different values for the time bin width. We saw a consistent
increase in $\rho_{X,Y|Z}$ as the time bin width increased in all three subjects. But
we saw different trends for in ρ_{signal} for each of the subjects.

timescales (see figures 2.11c and 2.11d). The communities detected bore very little relation
to the anatomical division of the cells. The adjusted Rand index between the community
clustering and the anatomical 'clustering' is close to zero for every time bin width (see figure
2.13b). This indicates that the similarity between the clusterings is close to chance. We did
observe a slight downward trend in the variation of information with increasing bin width
(see figure 2.13a), but this is more likely due to a decrease in the number of communities
detected rather than any relationship with anatomy.

We also observed that the network noise rejection process rejected some of the cells
when applied to the network based on the signal correlations. This means that those cells
did not contribute to the additional structure of the network beyond that captured by the
sparse weighted configuration model. This is why the matrices in figures 2.11c and 2.11d are
smaller than their analogues in figures 2.11a and 2.11b.
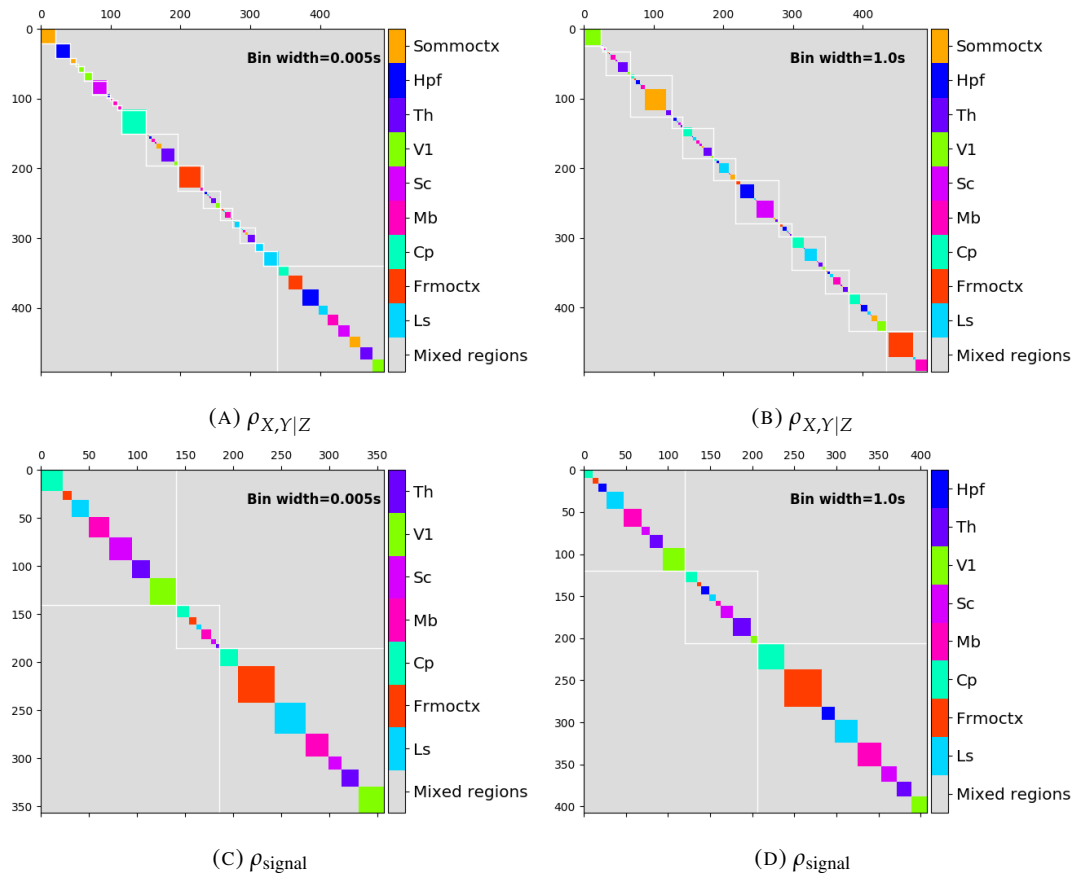
FIGURE 2.11: Matrices showing the regional membership of pairs by colour, and the communities in which those pairs lie. (a-b) Detected communities and regional membership matrix for network based on rectified spike count correlation $\rho_{X,Y|Z}$, using time bin widths of 0.005s and 1s respectively. (c-d) Detected communities and regional membership matrix for network based on rectified signal correlation ρ_{signal}, using time bin widths of 0.005s and 1s respectively.

2.2.8 Absolute correlations and negative rectified correlations

At the moment, the network noise rejection protocol can only be applied to weighted undirected graphs with non-negative weights. This meant that we had to rectify our correlated networks before applying the network noise rejection and community detection process. We wanted to investigate what would happen if instead of rectifying the correlations, we used the absolute value, or reversed the signs of the correlations and then rectified.

When we used the absolute value of the correlations, we found very similar results to those shown above for the rectified total correlations and the rectified spike count correlations. The only exception being that we detected more communities. This could indicate that we detected both positively and negatively correlated communities, but we haven't done any further investigation so we cannot say for sure.

When we used the sign reversed rectified correlated networks, we tended to find fewer

(A) $\rho_{X,Y|Z}$ Variation of information.   (B) $\rho_{X,Y|Z}$ Adjusted Rand Index.
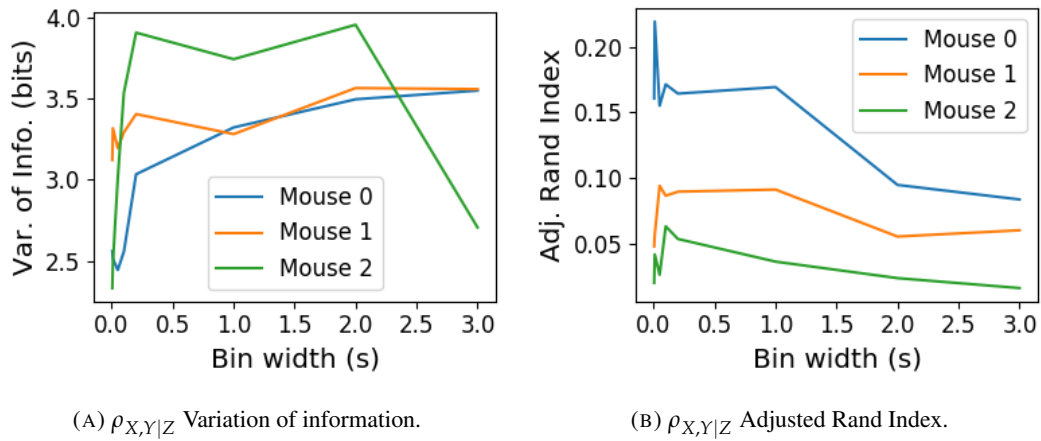
FIGURE 2.12: Distance and similarity measures between the anatomical division of the neurons, and the communities detected 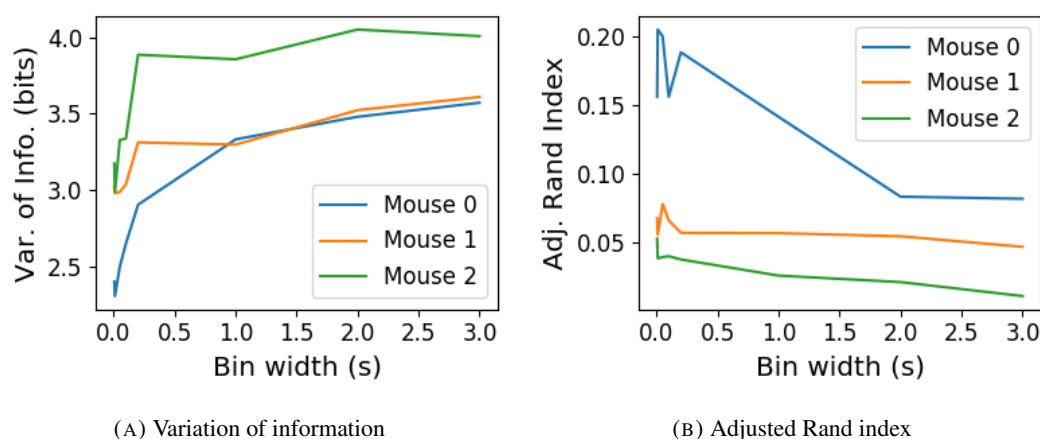in the network based on the spike count correlations $\rho_{X,Y|Z}$. (a) The variation of information is a 'distance' measure between clusterings. The distance between the anatomical 'clustering' and the community clustering increases as the time bin width increases. (b) The adjusted Rand index is a similarity measure between clusterings. The detected communities become less similar to the anatomical division of the cells as the time bin width increases.

communities and often found no signal network after applying network noise rejection. This indicates that there was not much structure in the negatively correlated networks beyond that captured by the sparse weighted configuration model.

2.3   Discussion

It is well established that the brain uses correlated behaviour in neuronal ensembles to represent the information taken in through sensation (Cohen and Maunsell, 2009; Litwin-Kumar, Chacron, and Doiron, 2012; deCharms and Merzenich, 1996). However, most studies that examine the nature of these correlations in-vivo, study an ensemble of cells from only one brain region (Cohen and Kohn, 2011). Furthermore, recent results have shown that behaviour can drive correlated activity in multiple brain regions, including those not normally associated with motor control (Stringer et al., 2019). In this study, we utilised one of the newly recorded large datasets containing electrophysiological recordings from multiple brain regions simultaneously. We investigated correlated behaviour in these different brain regions and we investigated correlated behaviour between neurons in different regions, during spontaneous behaviour.

A number of studies have found that the timescale of correlated behaviour induced by a stimulus can be modulated by the stimulus structure and behavioural context. For example,

(A) ρ_{signal} Variation of information.

(B) ρ_{signal} Adjusted Rand Index.
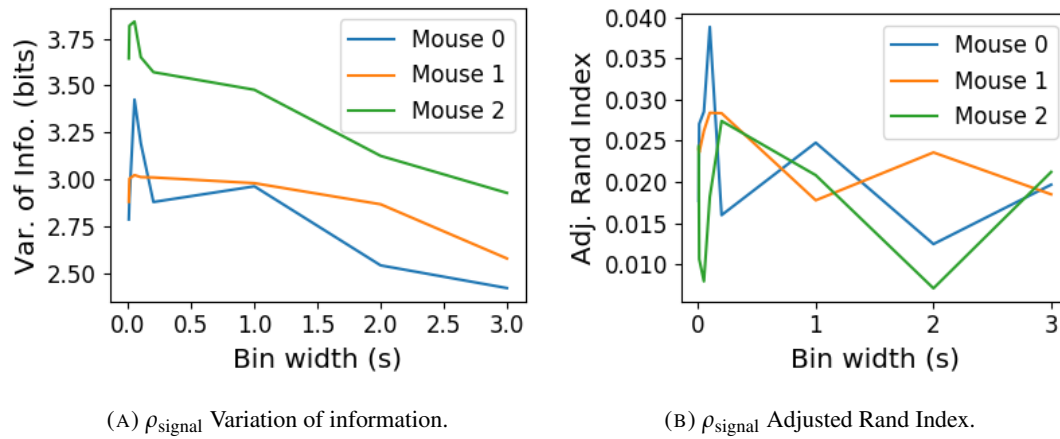
FIGURE 2.13: Distance and similarity measures between the anatomical division of the neurons, and the communities detected in the network based on the signal correlations ρ_{signal}. (a) The variation of information is a 'distance' measure between clusterings. The distance between the anatomical 'clustering' and the community clustering increases as the time bin width increases. (b) The adjusted Rand index is a similarity measure between clusterings. The detected communities become less similar to the anatomical division of the cells as the time bin width increases.

the spike train correlations between cells in weakly electric fish are modulated by the spatial extent of the stimulus (Litwin-Kumar, Chacron, and Doiron, 2012), and neurons in the marmoset primary auditory cortex modulate their spike timing (and therefore correlation) in response to stimulus features without modulating their firing rate (deCharms and Merzenich, 1996). Furthermore, the width of the time bins over which spike counts are measured has been shown to have an effect on the magnitude of those correlations (Cohen and Kohn, 2011). Despite this, very little research has been done comparing correlation measures from the same dataset at different timescales. We investigated this by varying the time bin width used to bin spike times into spike counts from as short as 5ms up to 3s.

In order to further investigate the effect of these correlations at different timescales, we regarded our neuronal ensemble as a weighted undirected graph, where each neuron is represented by a node, and the weight on each edge is the correlation between the neurons connected by that edge. We then applied a novel clustering method from network science (Humphries et al., 2019) to identify communities in these networks. These networks, and the community detection process, were completely agnostic of anatomical division of the cells in our ensemble. When we compared the detected communities with the anatomical division of the cells using distance and similarity measures for clusterings, we found that the detected communities were more similar to the anatomical division and shorter timescales. That is, when we used a wider time bin to count spikes, and computed pairwise correlations with these spike counts, the correlated communities tended to exist within anatomical regions at

shorter timescales, and tended to span anatomical regions at longer timescales. This could reflect localised functional correlations at short time scales rippling outwards across brain regions at longer timescales.

We acknowledged that the region spanning correlated communities that we detected at longer time scales could exist due to collating activity driven by distinct spontaneous activities. In order to account for this, we modelled the spike counts as a linear function of the top 500 principal components of a video of the mouse's face filmed simultaneously with the electrophysiological readings. We applied our network noise rejection and community detection process to the weighted undirected networks formed by the spike count correlations and the signal correlations that we calculated using our model. For the spike count correlation networks, we found much the same results as for the total correlations as described above. For the signal correlations, the communities detected in these networks bore little relation to the anatomical division of the cells.

There is a lot of room for further investigation based on this research. For a start, the data that we used here were collected from nine different regions in the mouse brain, but none of these regions were part of the somatosensory cortex. Given that a mouse experiences so much of its environment through its sense of smell, some data from this region would be interesting to investigate. On the same theme, the mice in the experiment from which the data were collected were headfixed and placed on a rotating ball, but were otherwise behaving spontaneously. Had these mice been exposed to a visual, aural, or olfactory stimulus, we could have examined the responses of the cells in the brain regions corresponding to vision, hearing, and olfaction, and compared these responses to the responses from the other brain regions. Furthermore, we could have investigated the interaction between the sets of responses.

Another space for further investigation is the community detection. The algorithm that we used here never detects overlapping communities. But functional communities could indeed have overlaps. Clustering methods that detect overlapping clusters do exist (Baadel, Thabtah, and Lu, 2016). Applying one of those algorithms could yield some interesting results. Also, the community detection algorithm that we used here cannot process graphs with negative weights, this forced us to separate positive and negative correlations before applying our network noise rejection and community detections process, or use the absolute value of our correlations. A community detection algorithm that can work on weighted undirected graphs with negative weights could yield some interesting results here.

2.4 Data

The data that we used in this project were collected by Nick Steinmetz and his lab members (Stringer et al., 2019).

2.4.1 Brain regions

Neuropixels probes were used to collect extracellular recordings (Jun et al., 2017) from three different mice. The mice were awake, headfixed, and engaging in spontaneous behaviour. The mice were of different sexes and different ages. One mouse was 'wild-type', the others were mutants. Details as follows:

1. male, wild type, P73.

2. female, TetO-GCaMP6s, Camk2a-tTa, P113

3. male, Ai32, Pvalb-Cre, P99

Eight probes were used to collect readings from 2296, 2668, and 1462 cells respectively. Data were collected from nine brain regions in each mouse:

- Caudate Putamen (CP)

- Frontal Motor Cortex (Frmoctx)

- Hippocampal formation (Hpf)

- Lateral Septum (Ls)

- Midbrain (Mb)

- Superior Colliculus (Sc)

- Somatomotor cortex (Sommoctx)

- Thalamus (Th)

- Primary visual cortex (V1)

Readings were continuous and lasted for about 1 hour (Stringer et al., 2019). Locations of each of the probes can be seen in figure 2.14.
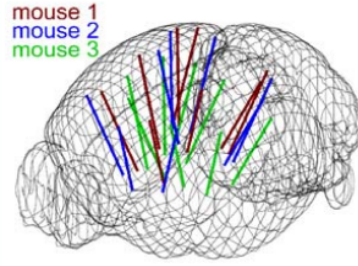
FIGURE 2.14: **Probe Locations:** The locations of the probes in each of the
three mouse brains (Stringer et al., 2019).

2.4.2   Video recordings

Video recordings of the mouse's face were taken during the spontaneous behaviour. We had access to the top 500 principle components and top 500 eigenvectors of the processed videos. The frequency of recording was slightly less than 40Hz. Each frame contained 327×561 pixels. These principal components were used as behavioural data. We controlled for these components when taking measurements conditioned on behaviour.

2.5   Methods

2.5.1   Binning data

We transoformed the spike timing data into binned spike count data by dividing the experimental period into time bins and counting the spikes fired by each cell within the time period covered by each of those bins. The data were divided into time bins of various widths ranging from 0.01s to 4s.

If the total length of the recording period was not an integer multiple of the time bin width, we cut off the remaining time at the end of the recording period. This period was at most 3.99s. This is far less than the total recording time of around 1 hour. So, this detail would not affect our results.

2.5.2   Correlation coefficients

We calculated Pearson's correlation coefficient for pairs of spike counts from pairs of neurons. For jointly distributed random variables X and Y, Pearson's correlation coefficient is

defined as:

$$\rho_{XY} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

$$= \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{2.2}$$

where E denotes the expected value, μ denotes the mean, and σ denotes the standard deviation. The correlation coefficient is a normalised measure of the covariance. It can take values between 1 (completely correlated) and -1 (completely anti-correlated). Two independent variables will have a correlation coefficient of 0. But, having 0 correlation does not imply independence.

If we do not know the means and standard deviations required for equation 2.1, but we have samples from X and Y, Pearson's sample correlation coefficient is defined as:

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.3}$$

where $\{(x_i, y_i)\}$ for $i \in \{1, \ldots, n\}$ are the paired samples from X and Y, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ are the sample means.

In practice we used the python function `scipy.stats.pearsonr` to calculate the correlation coefficients.

Total correlations, r_{SC}

The total correlation (r_{SC}) of two cells is the correlation between the spike counts of those cells in response to a given stimulus condition.

Shuffled total correlations

We measured the shuffled total correlations between two neurons by randomly permuting one of the neuron's spike counts and measuring the total correlations. These shuffled correlations were useful when measuring the effect of time bin width on correlations, and when deciding which correlations should be preserved when creating correlation networks (see section 2.5.5).

Separating Correlations & Anti-correlations

In order to compare the effect of bin width on measures of negative r_{SC} (anti-correlation) and positive r_{SC} separately, we had to separate correlated and anti-correlated pairs. To do this, we

simply measured the mean r_{SC}, taking the mean across all the bin widths. If this quantity was positive or zero we regarded the pair as positively correlated. If this quantity was negative we regarded the pair as anti-correlated.

2.5.3 Conditioning on behavioural data

Our behavioural data consisted of the top 500 principal components (PCs) of a processed video recording of the mouse's face (see section 2.4.2). Denoting the spike count of a given cell by X, and the PCs by $Z_1, \ldots, Z_{500}$, we wanted to model X as a function of $Z_1, \ldots, Z_{500}$ in order to estimate

$$E[X|Z_1, \ldots, Z_{500}] = \int_{x \in X} x P(X = x | Z_1, \ldots, Z_{500}) dx \qquad (2.4)$$

$$= \int_{x \in X} x \frac{P(X = x, Z_1, \ldots, Z_{500})}{P(Z_1, \ldots, Z_{500})} dx \qquad (2.5)$$

Given the 500 components, a naïve estimation of $P(Z_1, \ldots, Z_{500})$ or $P(X, Z_1, \ldots, Z_{500})$ by histogramming was impossible. Therefore we modelled X as a linear combination of the PCs.

Linear regression

We modelled the spike count of a given cell, X, as a linear combination of the PCs of the video of the mouse's face, $\mathbf{Z} = Z_1, \ldots, Z_{500}$. We tried three different types of regularization

- $L1$ or 'Lasso'

- $L2$ or 'Ridge regression'

- 'Elastic net' regularisation (a linear combination of both $L1$ and $L2$ regularisation penalities)

The elastic net regularisation performed the best, so we stuck with that.

Elastic net regularisation

Suppose we wish to model n observations of a random variable X, $\mathbf{x} = (x_1, \cdots, x_n)$ using n instances of m predictors $\mathbf{Z} = (Z_1, \cdots, Z_m)$. The naïve elastic net criterion is

$$L(\lambda_1, \lambda_2, \boldsymbol{\beta}) = |\mathbf{x} - \mathbf{Z}\boldsymbol{\beta}|^2 + \lambda_2 |\boldsymbol{\beta}|_2 + \lambda_1 |\boldsymbol{\beta}|_1 \qquad (2.6)$$

where

$$|\boldsymbol{\beta}|_2 = \sum_{j=1}^{m} \beta_j^2 \tag{2.7}$$

$$|\boldsymbol{\beta}|_1 = \sum_{j=1}^{m} |\beta_j| \tag{2.8}$$

The naïve elastic net estimator $\hat{\boldsymbol{\beta}}$ is the minimiser of the system of equations 2.6 (Zou and Hastie, 2005)

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} L(\lambda_1, \lambda_2, \boldsymbol{\beta}) \tag{2.9}$$

We implemented the model using the `ElasticNetCV` method of Python's `sklearn.linear_models` package.

As well as using the PCs, we also tried fitting the models using the raw video data reconstructed from the PCs and eigenvectors. These models performed worse than those using the PCs. We expected this because each representation contains the same amount of information, but the raw video representation spreads this information across many more components. This requires more parameter fitting, but given the same information.

Conditional covariance

We calculated the expected value of the conditional covariance using the law of total covariance.

$$\mathrm{cov}(X, Y) = E[\mathrm{cov}(X, Y|Z)] + \mathrm{cov}(E[X|Z], E[Y|Z]) \tag{2.10}$$

where these expected values are calculated with respect to the distribution of Z as a random variable.

The law of total covariance breaks the covariance into two components. The first component $E[\mathrm{cov}(X, Y|Z)]$ is the expected value, under the distribution of Z, of the conditional covariance $\mathrm{cov}(X, Y|Z)$. This covariance could be interpreted as the unnormalised version of what Cohen et al. (2011) call the spike count correlation (Cohen and Kohn, 2011), aka. the noise correlation. In particular, this is the covariance of the spike counts in response to repeated presentation of identical stimuli.

The second component is analogous to what Cohn et al. (2011) call the *signal correlation* (Cohen and Kohn, 2011). In particular, $\mathrm{cov}(E[X|Z], E[Y|Z])$ is the covariance between

spike counts in response to different stimuli.

Using our linear model, we calculated $E[X|Z_1, \ldots, Z_{500}]$ for each cell X. Then we proceeded to calculate

$$E[\text{cov}(X, Y|Z_1, \ldots, Z_{500})] = \text{cov}(X, Y) - \tag{2.11}$$

$$\text{cov}(E[X|Z_1, \ldots, Z_{500}], E[Y|Z_1, \ldots, Z_{500}]) \tag{2.12}$$

Measures of conditional correlation

As a measure of expected correlation, we measured the 'event conditional correlation' (Maugis, 2014)

$$\rho_{XY|Z} = \frac{E[\text{cov}(X, Y|Z)]}{\sqrt{E[\text{var}(X|Z)]E[\text{var}(Y|Z)]}} \tag{2.13}$$

Although this is not an actual correlation, it is an intuitive analogue to the correlation as a normalised version of the covariance.

For comparison, we also measured the 'signal correlation'

$$\rho_{\text{signal}} = \frac{\text{cov}(E[X|Z], E[Y|Z])}{\sqrt{\text{var}(E[X|Z])\text{var}(E[Y|Z])}} \tag{2.14}$$

this is an actual correlation.

2.5.4   Information Theory

Entropy $H(X)$

The entropy of a random variable X, with outcomes $x_1, \ldots, x_N$, and corresponding probabilities $p_1, \ldots, p_N$ is defined as

$$H(X) = -\sum_{n=1}^{N} p_n \log_2 p_n \tag{2.15}$$

This quantity is also known as the information entropy or the 'surprise'. It measures the amount of uncertainty in a random variable. For example, a variable with a probability of 1 for one outcome, and 0 for all other outcomes will have 0 bits entropy, because it contains no uncertainty. But a variable with a uniform distribution will have maximal entropy as it is the least predictable. This quantity is analogous to the entropy of a physical system (Shannon,

1948). Note that any base may be used for the logarithm in equation 2.15, but using base 2 means that the quantity will be measured in 'bits'.

The joint entropy of two jointly distributed random variables X and Y, where Y has outcomes $y_1, \ldots, y_M$, is defined as

$$H(X,Y) = -\sum_{n=1}^{N} \sum_{m=1}^{M} P(X = x_n, Y = y_m) \log_2 P(X = x_n, Y = y_m) \qquad (2.16)$$

If X and Y are independent then $H(X,Y) = H(X) + H(Y)$. Otherwise $H(X,Y) < H(X) + H(Y)$. When X and Y and completely dependent $H(X,Y) = H(X) = H(Y)$.

The conditional entropy of Y conditioned on X is defined as

$$H(Y|X) = -\sum_{n=1}^{N} \sum_{m=1}^{M} P(X = x_n, Y = y_m) \log_2 \frac{P(X = x_n, Y = y_m)}{P(X = x_n)} \qquad (2.17)$$

When X and Y are independent $H(Y|X) = H(Y)$. Intuitively, we learn nothing of Y by knowing X, so Y is equally uncertain whether we know X or not. If Y is totally dependent on X, then the fraction in the logarithm is 1, which gives $H(Y|X) = 0$.

These entropy measures are the basis of the mutual information measure.

Maximum entropy limit

When spiking data is binned into spike counts there is an upper limit on the entropy of these data. The maximum entropy discrete distribution is the discrete uniform distribution. A random variable with this distribution will take values from some finite set with equal probabilities. Binned spike count data will take values between 0 and some maximum observed spike count $n_{\max}$. A neuron with responses that maximises entropy will take these values with equal probability, i.e. if $i \in \{0, \ldots, n_{\max}\}$ then $P(X = i) = \frac{1}{n_{\max}+1}$. The entropy of this neuron will be

$$\begin{aligned}
H(X) &= -\sum_{i=0}^{n_{\max}} P(X = i) \log_2 P(X = i) \\
&= -\sum_{i=0}^{n_{\max}} \frac{1}{n_{\max} + 1} \log_2 \left(\frac{1}{n_{\max} + 1} \right) \\
&= -\log_2 \left(\frac{1}{n_{\max} + 1} \right) \\
&= \log_2 (n_{\max} + 1)
\end{aligned}$$

Therefore, the maximum entropy of the binned spike counts of a neuron is $\log_2 (n_{max} + 1)$. Of course, it would be very unusual for a neuron to fire in accordance with the discrete uniform distribution. Most measurements of entropy taken on binned spiking data will be much lower than the maximum. See figure 2.15 to see the maximum entropy as a function of the maximum observed spike count.
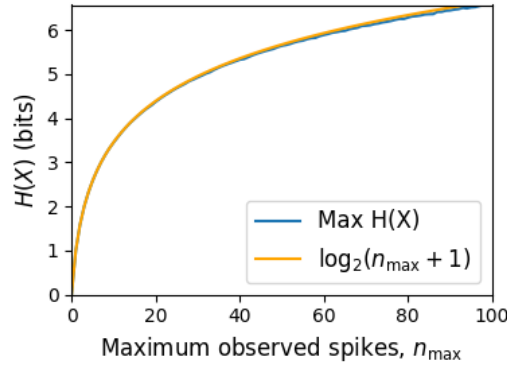
FIGURE 2.15: **Entropy Limit:** The upper limit on entropy of binned spike count data as a function of the maximum observed spike count. The orange line is the analytical maximum. The blue line is the entropy of samples with $N = 1000$ data points taken from the discrete uniform distribution.

Mutual Information $I(X;Y)$

The mutual information can be defined mathematically in a number of ways, all of which are equivalent. These definitions illustrate the different ways of interpreting the mutual information.

For two jointly distributed random variables X and Y, the mutual information $I(X;Y)$ is defined as

$$I(X;Y) = H(Y) - H(Y|X) \tag{2.18}$$
$$= H(X) - H(X|Y) \tag{2.19}$$

Equation 2.18 fits with the following intuition: The mutual information between X and Y is the reduction in uncertainty about X gained by knowing Y, or vice versa. We could also say the mutual information is the amount of information gained about X by knowing Y, or vice versa.

Another useful entropy based definition for the mutual information is

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{2.20}$$

This definition is useful because it does not require the calculation of conditional probabilities.

The mutual information can also be defined in terms of marginal, joint, and conditional distributions. For example,

$$I(X;Y) = -\sum_{n=1}^{N} \sum_{m=1}^{M} P(X = x_n, Y = y_m) \log_2 \frac{P(X = x_n, Y = y_m)}{P(X = x_n)P(Y = y_m)} \qquad (2.21)$$

Notice that this can be rewritten as a Kullback–Leibler divergence.

$$I(X;Y) = D_{KL}(P(X,Y)||P(X)P(Y)) \qquad (2.22)$$

So, we can also think of the mutual information as a measure of the difference between the joint distribution of X and Y, and the product of their marginal distributions. Since the product of the marginal distributions is the joint distribution for independent variables, we can think of the mutual information as a measure of the variables' dependence on one another.

The minimum value that $I(X;Y)$ can take is 0. This occurs when the random variables X and Y are independent. Then we have $H(X|Y) = H(X)$, and $H(Y|X) = H(Y)$, which according to equation 2.18, gives $I(X;Y) = 0$. We also have that $H(X,Y) = H(X) + H(Y)$ in this case, which according equation 2.20, gives $I(X;Y) = 0$. Finally, we also have $P(X,Y) = P(X)P(Y)$, which leaves us with 1 in the argument for the logarithm in equation 2.21, which again gives $I(X;Y) = 0$.

The mutual information reaches its maximum value when one of the variables X and Y is completely determined by knowing the value of the other. In that case $I(X;Y) = \min\{H(X), H(Y)\}$.

Variation of Information $VI(X,Y)$

The variation of information is another information theoretical quantity based on the mutual information. It is defined as

$$VI(X;Y) = H(X) + H(Y) - 2I(X;Y) \qquad (2.23)$$

We can rewrite this as the summation of two positive quantities

$$VI(X;Y) = [H(X) - I(X;Y)] + [H(Y) - I(X;Y)] \qquad (2.24)$$

In English, the variation of information is the summation of the uncertainty in the random variables X and Y excluding the uncertainty shared by those variables.

This measure will become more relevant when we go on to talk about clusterings because $VI(X;Y)$ forms a metric on the space of clusterings.

Measuring entropies & mutual information

In practice, we measured the mutual information between spike counts using Python and the python package `pyitlib`. We used the PT-bias correction technique to estimate the bias of our measurements when measuring the mutual information between the spike counts of two cells (Treves and Panzeri, 1995).

When measuring the mutual information between clusterings we used Python, but we used the `mutual_info_score`, `adjusted_mutual_info_score`, and `normalized_mutual_info_score` functions from the `sklearn.metrics` part of the `sklearn` package.

2.5.5   Network analysis

Correlation networks

In order to analyse functional networks created by the neurons in our ensemble, we measured the total correlation between each pair of neurons. These measurements induced an undirected weighted graph/network between the neurons. The weight of each connection was equal to the total correlation between each pair of neurons.

We followed the same procedure for total correlations 2.5.2, spike count correlations, and signal correlations 2.5.3.

Rectified correlations

At the time of writing, the community detection method outlined in (Humphries et al., 2019) could only be applied to networks with positively weighted connections. But many neuron pairs were negatively correlated. To apply the community detection method, we *rectified* the network, by setting all the negative weights to zero.

We also looked for structure in the network created by negative correlations by reversing the signs of the correlations, and rectifying these correlations before applying our network analysis.

Finally, we used the absolute value of the correlations as the weights for the graph/network. By doing this, we hoped to identify both correlated and anti-correlated functional communities of neurons.

Sparsifying data networks

When creating our correlation networks, we wanted to exclude any correlations that could be judged to exist 'by chance'. To do this, we measured the 5th and 95th percentile of the shuffled correlations (see section 2.5.2) for the given mouse and time bin width. We then set all the data correlations between these two values to 0. This excluded any 'chance' correlations from our network, and created a sparser network. This allowed us to make use of the 'sparse weighted configuration model' as described in section 2.5.5.

Communities

Given some network represented by an adjacency matrix **A**, a community within that network is defined as a collection of nodes where the number of connections within these nodes is higher than the expected number of connections between these nodes. In order to quantify the 'expected' number of connections, we need a model of expected networks. This is analogous to a 'null model' in traditional hypothesis testing. We test the hypothesis that our data network departs from the null network model to a statistically significant degree. For undirected unweighted networks, the canonical model of a null network is the configuration model (Fosdick et al., 2016). Since we are working with weighted sparse networks, we used more suitable null models, described below.

Weighted configuration model

The *weighted configuration model* is a canonical null network model for weighted networks. Given some data network, the weighted configuration model null network will preserve the degree sequence and weight sequence of each node in the data network. But the edges will be distributed randomly (Fosdick et al., 2016). Any structure in the data network beyond its degree sequence and weight sequence will not be captured in the weighted configuration model. So, this model can be used in testing the hypothesis that this extra structure exists.

Sparse weighted configuration model

The *sparse weighted configuration model* is another null network model. Similar in nature to the weighted configuration model (see section 2.5.5), but the sparsity of the data network is

1439 preserved in the null network. This is achieved by sampling from a probability distribution

1440 for the creation or non-creation of each possible connection, then distributing the weight of

1441 the data network randomly in this sparse network (Humphries et al., 2019). This is the null

1442 network that we used when searching for additional strcuture in our data networks.

Spectral rejection

1444 We made use of the spectral rejection algorithm as outlined in (Humphries et al., 2019). The

1445 spectral rejection algorithm is a method for finding structure in a network not captured by a

1446 supposed null model, if such structure exists.

To describe the method, we denote our data network matrix $\mathbf{W}$, we denote the expected

network of our null network model as $\langle \mathbf{P} \rangle$. Then the departure of our data network from the

null network can be described by the matrix

$$\mathbf{B} = \mathbf{W} - \langle \mathbf{P} \rangle \tag{2.25}$$

1447 a common choice for $\langle \mathbf{P} \rangle$ in community detection is the 'configuration model' (Fosdick et

1448 al., 2016; Humphries, 2011). The matrix $\mathbf{B}$ is often called the configuration matrix, in this

1449 context we will use the term 'deviation matrix' as it captures the deviation of $\mathbf{W}$ from the

1450 null model.

1451 To test for structure in the network represented by $\mathbf{W}$, we examine the eigenspectrum of $\mathbf{B}$

1452 and compare it to the eigenspectrum of our null model. Firstly, note that since our data model

1453 doesn't allow self loops, and is not directed, the matrix representing the network will be

1454 symmetric and positive semi-definite, and will therefore be invertible with real eigenvalues.

1455 We selected a null model with the same characteristics.

1456 To find the eigenspectrum of the null model, we generated N samples from our null

1457 model $P_1, \ldots, P_N$, and we measured their deviation matrices $B_1, \ldots, B_N$. We then calculated

1458 the eigenspectrum of each of those samples. We calculated the upper bound of the null model

1459 eigenspectrum by taking the mean of the largest eigenvalues of $B_1, \ldots, B_N$. We calculated a

1460 lower bound on the null model eigenspectrum by taking the mean of the smallest eigenvalues

1461 of $B_1, \ldots, B_N$.

1462 We then calculated the eigenspectrum of $\mathbf{B}$, our data network deviation matrix. If any of

1463 those eigenvalues lay outside of the upper or lower bounds of the null model eigenspectrum,

1464 this is evidence of additional structure not captured by the null model. If we chose the sparse

1465 weighted configuration model (see section 2.5.5) as our null network model, then eigenvalues

lying below the lower bound indicate k-partite structure in the network. For example, if one eigenvalue lay below the lower bound, this would indicate some bipartite structure in the data network. If any eigenvalues lay above the upper bound of the null model eigenspectrum, this is evidence of community structure in the data network. For example, one eigenvalue of **B** lying above the upper bound of the null model eigenspectrum indicates the presence of two communities in the network (Humphries, 2011).

Node rejection

If there are d data eigenvalues lying outside of the null network eigenspectrum, the d eigenvectors corresponding to these eigenvalues will form a vector space. If we project the nodes of our network into this vector space, by projecting either rows or colmns of the data matrix, we can see how strongly each node contributes to the vector space. Nodes that contribute strongly to the additional structure will project far away from the origin, nodes that do not contribute to the additional structure will project close to the origin. We want to use this information to discard those nodes that do not contribute.

We can test whether a node projects *far* away from the origin or *close* to the origin using the eigenvalues and eigenvectors of $B_1, \ldots, B_N$. The jth eigenvector and eigenvalue of B_i gives a value for a null network's projection into the jth dimension of the additional structure vector space. The matrices $B_1, \ldots, B_N$ give N projections into that dimension. These projections are a distribution of the null networks' projections. If the data node's projection exceeds that of the null network projections this node is judged to project *far* from the origin, and therefore contribute to the additional structure. Otherwise, the node is judged to project *close* to the origin, and is therefore rejected (Humphries et al., 2019).

Community detection

Another application for this d dimensional space is community detection. We first project all of the nodes into this d-dimensional space, then perform the clustering in this space. The clustering and community detection procedure is described in (Humphries, 2011).

In practice, the procedure is carried out n times (we chose $n = 100$ times), this returns n clusterings. We resolve these n clusterings to one final clustering using *consensus clustering*. We used the consensus clustering method that uses an explicit null model for the consensus matrix, as outlined in (Humphries et al., 2019).

2.5.6   Clustering Comparison

A clustering $\mathcal{C}$ is a partition of a set D into sets $C_1, C_2, \ldots, C_K$, called clusters, that satisfy the following for all $k, l \in \{1, \ldots, K\}$:

$$C_k \cap C_l = \varnothing \tag{2.26}$$

$$\bigcup_{k=1}^{K} C_k = D \tag{2.27}$$

If we consider two clusterings, $\mathcal{C}$ with clusters $C_1, C_2, \ldots, C_K$ and $\mathcal{C}'$ with clusters $C_1', C_2', \ldots, C_K'$. There are a number of measurements we can use to compare $\mathcal{C}$ and $\mathcal{C}'$. In the following, the number of elements in D is denoted by n, and the number of elements in cluster C_k is n_k.

Adjusted Rand Index

The *adjusted Rand Index* is a normalised similarity measure for clusterings based on pair counting.

If we consider the clusterings $\mathcal{C}$ and $\mathcal{C}'$, and denote

- the number of pairs in the same cluster in $\mathcal{C}$ and $\mathcal{C}'$ by N_{11}

- the number of pairs in different clusters in $\mathcal{C}$ and $\mathcal{C}'$ by N_{00}

- the number of pairs in the same cluster in $\mathcal{C}$ and different clusters in $\mathcal{C}'$ by N_{10}

- the number of pairs in different clusters in $\mathcal{C}$ and the same cluster in $\mathcal{C}'$ by N_{01}

then the *Rand Index* is defined as

$$RI = \frac{N_{11} + N_{00}}{N_{11} + N_{00} + N_{10} + N_{01}} = \frac{N_{11} + N_{00}}{\binom{n}{2}} \tag{2.28}$$

The Rand Index is 1 when the clusterings are identical, and 0 when the clusterings are completely different.

The *adjusted Rand Index* intends on correcting the Rand Index for chance matching pairs. This is defined as

$$ARI = \frac{2\left(N_{00}N_{11} - N_{01}N_{10}\right)}{\left(N_{00} + N_{01}\right)\left(N_{01} + N_{11}\right) + \left(N_{00} + N_{10}\right)\left(N_{10} + N_{11}\right)} \tag{2.29}$$

The adjusted Rand Index is 1 when the clusterings are identical, and 0 when the Rand Index is equal to its expected value.

Clusterings as random variables

If we take any random element of D, the probability that this element is in cluster C_k of clustering $\mathcal{C}$ is

$$P(K = k) = \frac{n_k}{n} \tag{2.30}$$

this defines a probability distribution, which makes the clustering a random variable. Any clustering can be considered as a random variable this way.

This means that we can measure any of the information theoretic quantities defined in section 2.5.4 with respect to clusterings. For example, the entropy of a clustering is

$$H(\mathcal{C}) = - \sum_{k=1}^{K} \frac{n_k}{n} \log \frac{n_k}{n} \tag{2.31}$$

If we have two clusterings, the joint probability distribution of these clusterings is defined as

$$P(K = k, K' = k') = \frac{|C_k \cap C'_{k'}|}{n} \tag{2.32}$$

The joint distribution allows us to define the mutual information between two clusterings, $I(\mathcal{C}; \mathcal{C}')$ (Meilă, 2007).

Information based similarity measures

The mutual information between two clusterings is a similarity measure, with $I(\mathcal{C}; \mathcal{C}') = 0$ if $\mathcal{C}$ and $\mathcal{C}'$ are completely different, and $I(\mathcal{C}; \mathcal{C}') = H(\mathcal{C}) = H(\mathcal{C}')$ if $\mathcal{C}$ and $\mathcal{C}'$ are identical. This can be normalised in a number of different ways to make more similarity measures (Vinh, Epps, and Bailey, 2010)

$$NMI_{joint} = \frac{I(\mathcal{C}; \mathcal{C}')}{H(\mathcal{C}, \mathcal{C}')} \tag{2.33}$$

$$NMI_{max} = \frac{I(\mathcal{C}; \mathcal{C}')}{\max\{H(\mathcal{C}), H(\mathcal{C}')\}} \tag{2.34}$$

$$NMI_{sum} = \frac{2I(\mathcal{C}; \mathcal{C}')}{H(\mathcal{C}) + H(\mathcal{C}')} \tag{2.35}$$

$$NMI_{sqrt} = \frac{I(\mathcal{C}; \mathcal{C}')}{\sqrt{H(\mathcal{C})H(\mathcal{C}')}} \tag{2.36}$$

$$NMI_{min} = \frac{I(\mathcal{C}; \mathcal{C}')}{\min\{H(\mathcal{C}), H(\mathcal{C}')\}} \tag{2.37}$$

We can control for chance similarities between the two clusterings by measuring the *adjusted mutual information* between the clusterings. This is defined as

$$AMI_{sum} = \frac{I(\mathcal{C};\mathcal{C}') - E\{I(\mathcal{C};\mathcal{C}')\}}{\frac{1}{2}\left[H(\mathcal{C}) + H(\mathcal{C}')\right] - E\{I(\mathcal{C};\mathcal{C}')\}} \tag{2.38}$$

The first term in the demoniator, taking the average of the marginal entropies, can be replaced by taking the maximum, minimum, or the geometric mean (Vinh, Epps, and Bailey, 2010).

Information based metrics

The variation of information between two clusterings $VI(\mathcal{C};\mathcal{C}')$ (see section 2.5.4) is a metric on the space of clusterings (Meilă, 2007). That is,

$$VI(\mathcal{C};\mathcal{C}') \geq 0 \tag{2.39}$$

$$VI(\mathcal{C};\mathcal{C}') = 0 \iff \mathcal{C} = \mathcal{C}' \tag{2.40}$$

$$VI(\mathcal{C};\mathcal{C}') = VI(\mathcal{C}';\mathcal{C}) \tag{2.41}$$

$$VI(\mathcal{C};\mathcal{C}'') \leq VI(\mathcal{C};\mathcal{C}') + VI(\mathcal{C}';\mathcal{C}'') \tag{2.42}$$

Another metric is the *information distance* (Vinh, Epps, and Bailey, 2010)

$$D_{max} = \max\{H(\mathcal{C}), H(\mathcal{C}')\} - I(\mathcal{C};\mathcal{C}') \tag{2.43}$$

Both of these can be normalised

$$NVI(\mathcal{C};\mathcal{C}') = 1 - \frac{I(\mathcal{C};\mathcal{C}')}{H(\mathcal{C},\mathcal{C}')} \tag{2.44}$$

$$d_{max} = 1 - \frac{I(\mathcal{C};\mathcal{C}')}{\max\{H(\mathcal{C}), H(\mathcal{C}')\}} \tag{2.45}$$

Comparing detected communities and anatomical divisions

In order to quantify the difference or similarity between the communities detected in our correlation network and the anatomical classification of the cells in that network, we considered the communities and the anatomical regions as clusters in two different clusterings, $\mathcal{C}_{comm}$ and $\mathcal{C}_{anat}$, respectively. We then measured the similarity between the clusterings using the mutual information, the adjusted mutual information, and the normalised mutual information. We measured the difference between, or the distance between, the clusterings using the

variation of information, the normalised variation of information, and the normalised information distance. We also measured the difference between the clusterings using the adjusted Rand Index, just to use a non-information based measure.

We took all of these measures for communities detected using different time bin widths. This gave us an idea of the effect of time bin width on correlation networks in neural ensembles relative to anatomical regions within those ensembles.

Chapter 3

A simple two parameter distribution for modelling neuronal activity and capturing neuronal association

UNIVERSITY OF BRISTOL

Abstract

Engineering

Department of Computer Science

Doctor of Philosophy

Investigating, implementing, and creating methods for analysing large neuronal ensembles

by Thomas J. DELANEY

Recent developments in electrophysiological technology have lead to an increase in the size of e-phys datasets. Consequently, there is a requirement for new analysis techniques that can make use of these new datasets, while remaining easy to use in practice. In this work, we fit some one or two parameter probability distributions to spiking data collected from a mouse exposed to visual stimuli. We show that a little known probability distribution entitled the Conway-Maxwell-binomial distribution is a suitable model for the number of active neurons in a neuronal ensemble at any given moment. This distribution fits these data better than binomial or beta-binomial distributions. It also captures the correlated activity in the primary visual cortex induced by stimulus onset more effectively than simply measuring the correlations, at short timescales (< 10ms).

3.1 Introduction

Recent advances in electrophysiological technology, such as 'Neuropixels' probes (Jun et al., 2017) have allowed extracellular voltage measurements to be collected from multiple brain regions simultaneously routinely, and in larger numbers than traditional methods. These larger datasets require innovative methods to extract information from the data in a reasonable amount of time, 'reasonable' being subjective in this case.

Theoretically, all the information in an electrophysiological dataset with n neurons could be captured by calculating the probability distribution for every possible spiking pattern. This would require defining a random variable with 2^n possible values, a task that quickly becomes impossible as n increases. Attempts at approximating this random variable often involve measuring pairwise or higher order correlations (Schneidman et al., 2006; Flach, 2013; Ganmor, Segev, and Schneidman, 2011). But pairwise correlations may not be enough to characterise neural activity (Tkačik et al., 2014).

Measuring higher order correlations becomes computationally impractical quite quickly also (the number of 'three neuron correlations' to measure scales with $\binom{n}{3}$). In this paper, we dispense with measuring correlations directly, and attempt to characterise correlated behaviour by measuring 'association'; a more general concept that includes correlation.

In this work, we examined the ability of simple distributions to model the number of active (spiking) neurons in a neuronal ensemble at any given timepoint. We compared a little-known distribution named the Conway-Maxwell-binomial distribution to the binomial distribution and the beta-binomial distribution. Similar, to the binomial, and beta-binomial, the Conway-Maxwell-binomial distribution is a probability distribution of the number of successes in a series of Bernoulli trials, but allows over- and under-dispersion relative to the standard binomial distribution. This distribution should therefore be a good candidate for our purposes. We found that Conway-Maxwell-binomial distribution was usually the best candidate of the three that we examined.

We also observed some interesting changes in the number of active neurons at stimulus onset in the primary visual cortex and hippocampus. We observed some changes in this activity throughout the stimulus presentation in the thalamus.

We found that fitting a Conway-Maxwell-binomial distribution was a better method of capturing association between neurons than mearing the spike count correlation for the short time bins that we used (< 10ms).

Finally, we replicated the findings made by Churchland et al. (2010) relating to a reduction in neural variability at stimulus onset in the macaque cortical regions, but for readings taken from the mouse primary visual cortex.

3.2 Results

We defined a neuron as *active* in a time bin if it fires at least one spike during the time interval covered by that bin. We measured the number of active neurons in the primary visual cortex of a mouse in 1ms bins across 160 trials of a moving bar visual stimulus. We then slid a 100ms window across these 1ms bins taking mesurements, and fitting distributions along the way. We did the same for neurons in the thalamus, hippocampus, striatum, and motor cortex. We repeated the analysis for 5ms time bins with 40 bin windows, and 10ms time bins with 40 bin windows.

3.2.1 Increases in mean number of active neurons and variance in number of active neurons at stimulus onset in some regions

We measured the average number of active neurons, and the variance of the number of active neurons in a 100ms sliding window starting 500ms before stimulus onset until 1000ms after stimulus onset. We found differences in the response across regions. There were no observed changes in response to the stimulus in the motor cortex or the striatum. The changes in the other regions are detailed below.

Primary visual cortex

We found a transient increase in both the average and variance of the number of active neurons at stimulus onset, followed by a fall to pre-stimulus levels, followed by another transient increase (see figure 3.1). The oscillation in both of these measurements appear to reflect the frequency of the stimulus. We see a similar but lower amplitude oscillation at the end of the stimulus presentation.

(A) Moving average.

(B) Moving variance.

FIGURE 3.1: The (a) average and (b) variance of the number of active neurons, measured using a sliding window 100ms wide, split into 100 bins. The midpoint of the time interval for each window is used as the timepoint (x-axis point) for the measurements using that window. The grey shaded area indicates the presence of a visual stimulus. The opaque line is an average across the 160 trials that included a visual stimulus of any kind. We can see a transient increase in the average number of active neurons and the variance of this number, followed by a fluctuation and another increase.

Hippocampus

In the hippocampus we observed a transient increase in the average number of active neurons and in the variance of the number of active neurons at stimulus onset (see figure 3.2). The increase lasted about 125ms, and the subsequent fall to baseline took the a similar amount of time.

(A) Moving average.

(B) Moving variance.
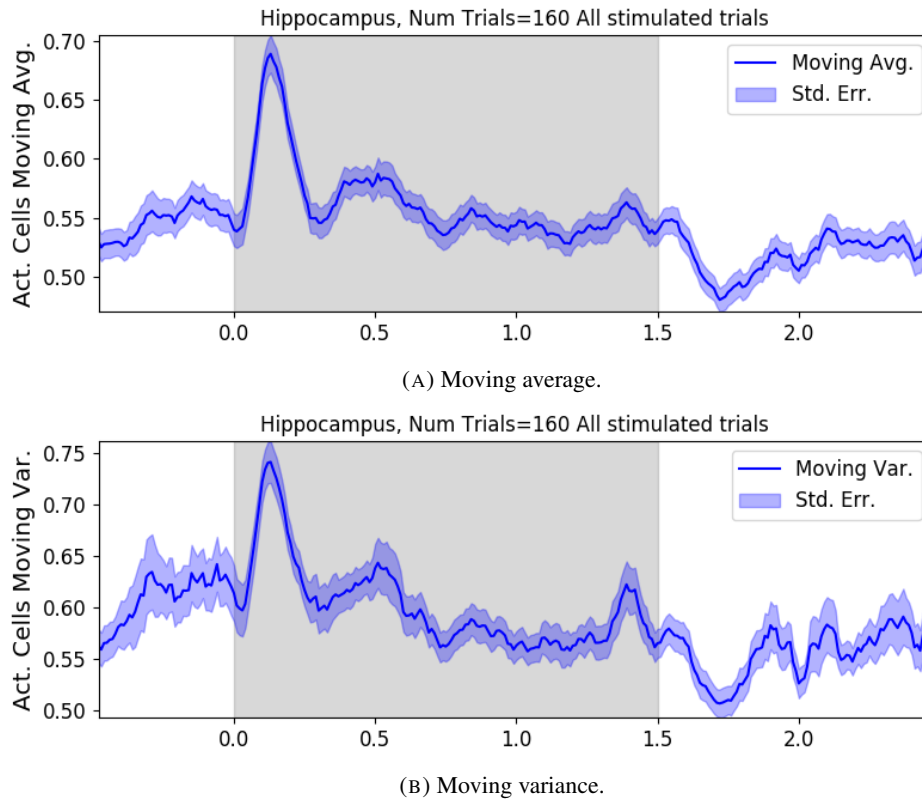
FIGURE 3.2: The (a) average and (b) variance of the number of active neurons, measured using a sliding window 100ms wide, split into 100 bins. The midpoint of the time interval for each window is used as the timepoint (x-axis point) for the measurements using that window. The grey shaded area indicates the presence of a visual stimulus. The opaque line is an average across the 160 trials that included a visual stimulus of any kind. We can see a transient increase in the average number of active neurons and the variance of this number.

Thalamus

In the thalamus we observed a transient increase in the both the average and variance of the number of active neurons on stimulus onset, followed by a fall to pre-stimulus levels, followed by a sustained increase until the stimulus presentation ends.
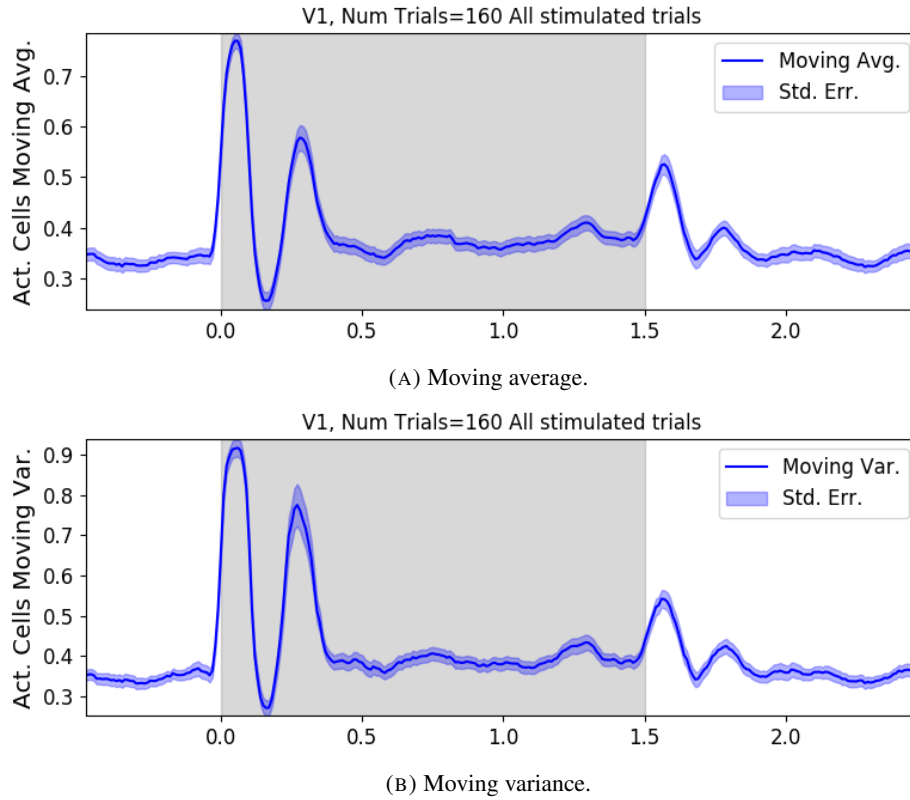
(A) Moving average.

(B) Moving variance.
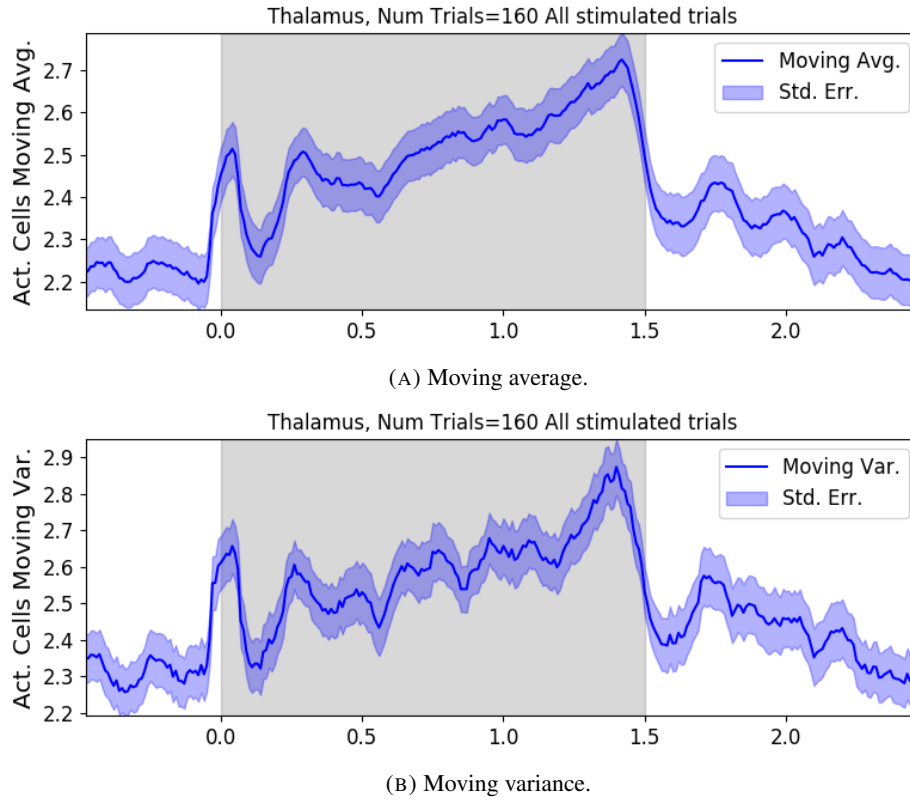
FIGURE 3.3: The (a) average and (b) variance of the number of active neurons, measured using a sliding window 100ms wide, split into 100 bins. The midpoint of the time interval for each window is used as the timepoint (x-axis point) for the measurements using that window. The grey shaded area indicates the presence of a visual stimulus. The opaque line is an average across the 160 trials that included a visual stimulus of any kind.

3.2.2 Conway-Maxwell-binomial distribution is usually a better fit than binomial or beta-binomial

After fitting a binomial, a beta-binomial, and a Conway-Maxwell-binomial (COMb) distribution to each window for each bin width, and each region, we assessed the goodness-of-fit of each distribution by calculating the log-likelihood of each fitted distribution using the associated sample. We measured the proportion of samples for which each distribution was the best fit, for each bin width value and each region.

We found that the COMb distribution was the best fit for most of the samples regardless of bin width or region. The bin width had an effect on the number of samples for which the COMb distribution was the best fit. For a quick summary of the results see table 3.1. For a bin width of 1ms, the COMb distribution was the best fit for over 90% of samples, the beta-binomial distribution was the best fit for less than 10% of samples, and the binomial distribution was the best fit for less that 1% of samples, across regions. For 5ms bins, the COMb distribution was the best fit for $70 - 80\%$ of samples, the beta-binomial distribution

Bin Width (ms)	Binomial	Beta-binomial	COMb
1ms	$< 1\%$	$< 10\%$	$> 90\%$
5ms	$< 0.1\%$	$20 - 30\%$	$70 - 80\%$
10ms	$< 0.1\%$	$20 - 47\%$	$53 - 80\%$

TABLE 3.1: Proportion of samples for which each distribution was the best fit, grouped by bin width. The COMb distribution is the best fit most of the time.

was the best fit for $20 - 30\%$ of the samples, and again the binomial distribution was the best fit for less that 1% of samples, across regions. Finally, for 10ms bins, the COMb distribution was the best fit for $53 - 80\%$ of samples, the beta-binomial distribution was the best fit for $20 - 47\%$ of the samples, and the binomial distribution was the best fit for less that 0.1% of samples, across regions.

(A) Example of fitted distributions.

(B) Proportion of best fit
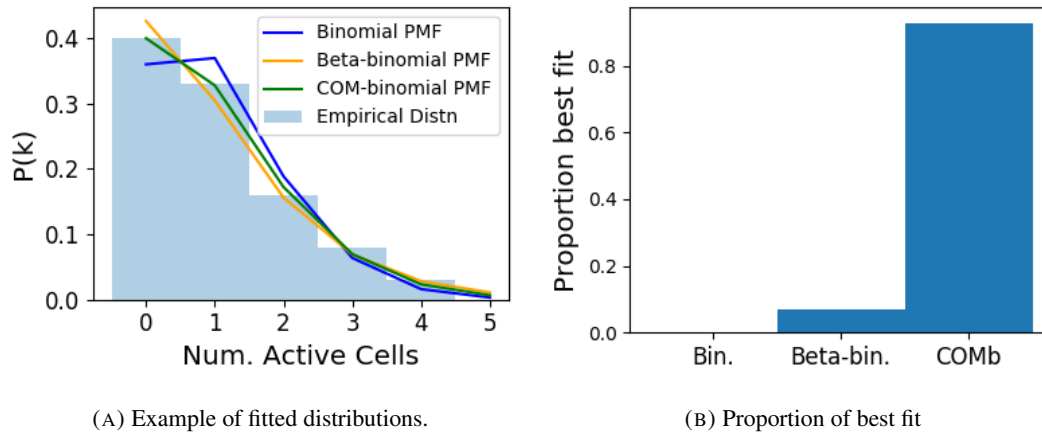
FIGURE 3.4: (a) An example of the binomial, beta-binomial, and Conway-Maxwell-binomial distributions fitted to a sample of neural activity. The Conway-Maxwell-binomial distribution is the best fit in this case. The histogram shows the empirical distribution of the sample. The probability mass function of each distribution is indicated by a different coloured line. (b) Across all samples in all trials, the proportion of samples for which each fitted distribution was the the best fit. The Conway-Maxwell-binomial distribution was the best fit for 93% of the samples taken from V1 using a bin width of 1ms.

3.2.3 Conway-Maxwell-binomial distribution captures changes in association at stimulus onset

We fit a Conway-Maxwell-binomial (COMb) distribution to the number of active neurons in the 1ms time bins in a 100ms sliding window. We also measured the correlation coefficient between the spike counts of all possible pairs of neurons, and took the average of these coefficients. We did this for all the trials with a visual stimulus. We observed a reduction in the COMb distribution's ν parameter at stimulus onset from around 1 to between 0 and 1 (see figure 3.5a). A value of ν less than 1 indicates positive association between the neurons (see section 3.5.5). We might expect to see this positive association reflected in the correlation coefficients, but this is not the case. We see no change in the time series of average correlation measures at stimulus onset.

This may be due to the very short time bin we used in this case. We know that using small time bins can artificially reduce correlation measurements (**cohen**). In this case, fitting the COMb distribution may be a useful way to measure association in a neuronal ensemble over very short timescales (< 10ms).

(A) COMb ν parameter.

(B) Average correlation coefficient.
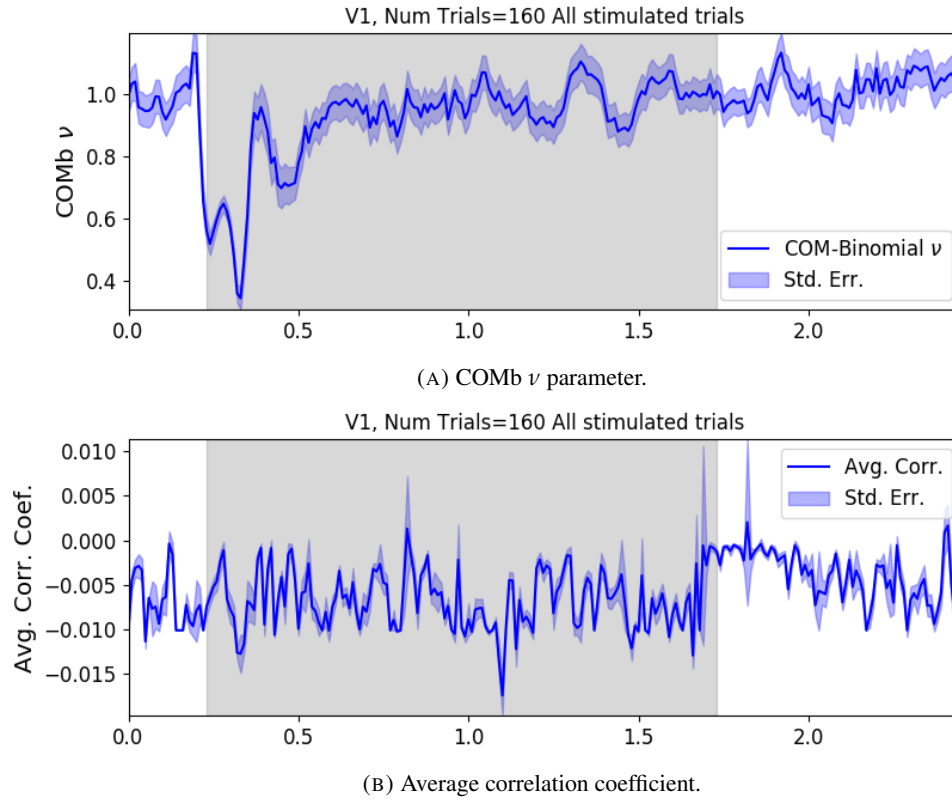
FIGURE 3.5: (a) We fit a Conway-Maxwell-binomial distribution to the number of active neurons in 1ms time bins of a 100ms sliding window. We did this for all trials with a visual stimulus and took the average across those trials. We see a transient drop in value for the distribution's ν parameter at stimulus onset. This shows an increase in association between the neurons. (b) We measured the correlation coefficient between the spike counts of all possible pairs of neurons in the same sliding window. The took the average of those coefficients. We also did this for every visually stimulated trial, and took the average across trials. The increase in association is not reflected with an increase in average correlation.

3.2.4   Replicating stimulus related quenching of neural variability

Churchland et al. (2010) inspected the effect of a stimulus on neural variability. One of the measures of neural variability that they employed was the Fano factor of the spike counts of individual cells (see section 3.5.4). They found a reduction in neural variability as measured by the Fano factor in various cortical areas in a macaque at the onset of various visual stimuli, or a juice reward (Churchland et al., 2010).

We measured the fano factor of the spike count of each cell in each brain region, during each trial. We measured the mean and standard error of these Fano factors from 500ms before stimulus onset until 1000ms after stimulus end. For the primary visual cortex, we found a transient reduction in the Fano factor immediately after stimulus onset. We used a Mann-Whitney U test to check that the Fano factors measured in a window starting at stimulus onset and ending 100ms later were significantly lower than the factors measured in a window ending at stimulus onset ($p < 0.001$, see figure 3.6a). We did not get this statistically significant result in any other region.

Our findings agree with those of Churchland et al. for the primary visual cortex. However Churchland also found a reduction in the Fano factor in the dorsal premotor cortex (PMd) at stimulus onset. Our measurements from the mouse motor cortex show no change at stimulus onset (see figure 3.6b). This could indicate some difference in the functionality of the motor cortex in a macaque and the motor cortex of a mouse.

(A) Primary visual cortex.   (B) Motor cortex.
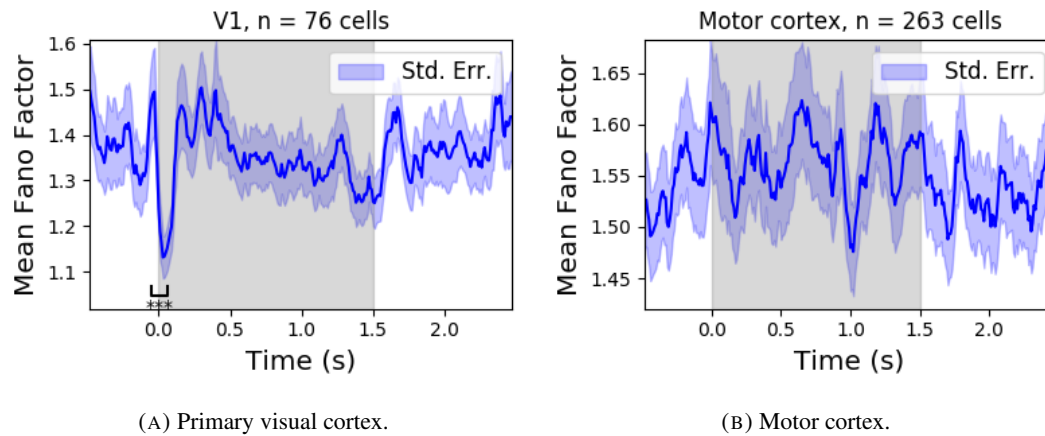
FIGURE 3.6: (a) The mean Fano factor of the spike counts of the cells in the primary visual cortex. Means were taken across cells first, then across trials. There was a significant decrease in the Fano factors immediately after stimulus onset. (b) The mean Fano factor of the spike counts of the cells in the motor cortex. No significant change in measurements at any point.

3.3 Discussion

Our aim in this research was to develop a new statistical method for analysing the activity of a neuronal ensemble at very short timescales. We wanted our method to use information taken from the whole ensemble, but we also wanted the method to be quick and easy to implement. It is likely that analysis methods with these characteristics will become valuable as electrophysiological datasets include readings from more cells over longer time periods. In this case, we used the number of active, or spiking, neurons in a very short time bin (< 10ms) as a measure of ensemble activity.

First of all, we showed that there were changes in behaviour to model at these very short time scales, in some of the brain regions from which we had recordings. Using such short time bins can produce artificially small spike count correlation measurements (**cohen**). Overcoming this limitation was one of our objectives for our new method.

We showed that a little known probability distribution, the Conway-Maxwell-binomial distribution, was a good choice compared to the better known options. Furthermore, we showed that the positively associated behaviour between neurons in the primary visual cortex could be captured by fitting a Conway-Maxwell-binomial distribution, but was not captured by the more standard approach of measuring the spike count correlation.

This work could be just a first step in creating analysis methods based on the Conway-Maxwell-binomial distribution, or similar statistical models. One way to extend the method would be to pair it up with the 'Population Tracking model' (O'Donnell et al., 2017). This

would give us a model that gives a probability of firing for each cell, and therefore probabilities for full spiking pattern, without adding a hugh number of parameters to fit.

A more complex way to extend the model would be to fit a Conway-Maxwell-binomial distribution to data recorded from multiple brain regions simultaneously, with a different fit for each region, then to analyse the temporal relationship between the fitted parameters of each region. This may give some results relating to the timescales in which information is processed in different brain regions.

TODO: mention application to unsorted data mention online applications

3.4 Data

We used data collected by Nick Steinmetz and his lab 'CortexLab at UCL' (Steinmetz, Carandini, and Harris, 2019). The data can be found online [1] and are free to use for research purposes.

Two 'Phase3' Neuropixels (Jun et al., 2017) electrode arrays were inserted into the brain of an awake, head-fixed mouse for about an hour and a half. These electrode arrays recorded 384 channels of neural data each at 30kHz and less than 7μV RMS noise levels. The sites are densely spaced in a 'continuous tetrode'-like arrangement, and a whole array records from a 3.8mm span of the brain. One array recorded from visual cortex, hippocampus, and thalamus, the other array recorded from motor cortex and striatum. The data were spike-sorted automatically by Kilosort and manually by Nick Steinmetz using Phy. In total 831 well-isolated individual neurons were identified.

3.4.1 Experimental protocol

The mouse was shown a visual stimulus on three monitors placed around the mouse at right angles to each other, covering about ±135 degrees azimuth and ±35 degrees elevation.

The stimulus consisted of sine-wave modulated full-field drifting gratings of 16 drift directions ($0°, 22.5°, \ldots, 337.5°$) with 2Hz temporal frequency and 0.08 cycles/degree spatial frequency displayed for 2 seconds plus a blank condition. Each of these 17 conditions were presented 10 times in a random order across 170 different trials. There were therefore 160 trials with a drifting-grating visual stimulus present, and 10 trials with a blank stimulus.

---

[1] http://data.cortexlab.net/dualPhase3/

3.5 Methods

3.5.1 Binning data

We converted the spike times for each cell into spike counts by putting the spike times into time bins of a given 'width' (in milliseconds). We used time bins of 1ms, 5ms, and 10ms. We used different time bin widths to assess the impact of choosing a bin width.

3.5.2 Number of *active* neurons

To count the number of active neurons in each neuronal ensemble, we split the time interval for each trial into bins of a given width. We counted the number of spikes fired by each cell in each bin. If a cell fired *at least* one spike in a given bin, we regarded that cell as active in that bin. We recorded the number of active cells in every bin, and for the purposes of further anaylsis, we recorded each cell's individual spike counts.

It should be noted that when we used a bin width of 1ms, the maximum number of spikes in any bin was 1. For the wider time bins, some bins had spike counts greater than 1. Consequently when using a bin width of 1ms, the number of active neurons and the total spike count of a given bin were identical. But for wider bin widths, the total spike count was greater than the number of active neurons.

So for the 1ms bin width, the activity of a neuron and the number of spikes fired by that neuron in any bin can be modelled as a Bernoulli variable. But for wider time bins, only the activity can be modelled in this way.

3.5.3 Moving windows for measurements

When taking measurements (e.g. moving average over the number of active neurons) or fitting distributions (eg. the beta binomial distribution) we slid a window containing a certain number of bins across the data, and made our measurements at each window position. For example, when analysing 1ms bin data, we used a window containing 100 bins, and we slid the window across the time interval for each trial moving 10 bins at a time. So that for 3060ms of data, we made 296 measurements.

For the 5ms bin width data, we used windows containing 40 bins, and slid the window 2 bins at a time when taking measurements.

For the 10ms bin width data, we used windows containing 40 bins, and slid the window 1 bin at a time when taking measurements (see table 3.2 for concise details).

Bin width (ms)	Window size (bins)	Window size (ms)	Windows per trial
1ms	100	100ms	296
5ms	40	200ms	286
10ms	40	400ms	266

TABLE 3.2: Details of the different bin width and anaylsis window sizes used when binning spike times, and analysing those data.

By continuing to use windows containing 40 bins, we retained statistical power but sacrificed the number of measurements taken.

There was an interval between each trial with a grey image in place of the moving of the moving bar stimulus. This interval varied in time. But we included some of this interval when recording the data for each trial. We started recording the number of active neurons, and the number of spikes from each neuron from 530ms before each trial until 1030ms after each trial. This way, we could see the change in our measurements at the onset of a stimulus and the end of stimulus presentation.

As mentioned in section 3.5.2, we recorded the number of active neurons in each bin, and the spike count for each neuron in each bin. The actual measurements we took using these data in each window were as follows:

Moving average The average number of active cells in each window.

Moving variance The variance of the number of active cells in each window.

Average correlation We measured the correlation between the spike counts of each pair of cells in the ensemble, and took the average of these measurements.

Binomial p We fitted a binomial distribution to the data in each window and recorded the fitted probability of success, p in each case.

Beta-binomial α, β We fitted a beta-binomial distribution to the data in each window, and recorded the values of the fitted shape parameters, α and β, of each distribution.

Conway-Maxwell-binomial distribution p, ν We fitted a Conway-Maxwell-binomial distribution to the data in each window, and recorded the fitted values of p and ν for each distribution.

Log-likelihoods We also recorded the log-likelihood of each of the fitted distributions for each window.

3.5.4 Fano factor

The *Fano factor* of a random variable is defined as the ratio of the variable's variance to its mean.

$$F = \frac{\sigma^2}{\mu} \tag{3.1}$$

We measured the Fano factor of the spike count of a given cell by measuring the mean and variance of the spike count across trials, and taking the ratio of those two quantities. When calculated in this way the Fano factor can be used as a measure of neural variability that controls for changes in the firing rate. This is similar to the calculation used in (Churchland et al., 2010).

3.5.5 Probability Distributions suitable for modelling ensemble activity

We present here three different probability distributions that could be suitable to model the number of active neurons in an ensemble. Each distribution has the set $\{0, \ldots, n\}$ as its support, where n is the number of neurons in the ensemble. These are simple distributions with either two or three parameters each. However, we regard n as known when using these distributions for modelling, so in effect each distribution has either one or two free parameters.

Association

Association between random variables is similar to the correlation between random variables but is more general in concept. The correlation is a measure of association; and association doesn't have a mathematical definition like correlation does. Essentially, the association between two random variables is their tendency to take the same or similar values. Positively associated variables tend to take the same value, and negatively associated variables tend to take different values. In this research, we work with probability distributions of the number of successes in a set of Bernoulli trials. These Bernoulli variables may or may not be associated.

A probability distribution over the number of successes in n Bernoulli trials, where the Bernoulli variables may be associated, could constitute a good model for the number of active neurons in an ensemble of n neurons.

Binomial distribution

The binomial distribution is a two parameter discrete probability distribution that can be thought of as a probability distribution the number of successes from n independent Bernoulli

trials, each with the same probability of success. The parameters of the binomial distribution are n the number of trials, and $0 \leq p \leq 1$, the probability of success for each of these trials. A random variable with the binomial distribution can take values from $\{0, \ldots, n\}$. The probability mass function of the distribution is

$$P(k; n, p) = \binom{n}{k} p^k (1 - p)^{n-k} \tag{3.2}$$

As a model for the activity of a neuronal ensemble, the main problem with the binomial distribution is that it treats each neuron, represented as a Bernoulli trial, as independent. It is well know that neurons are not independent, and that correlated behaviour between neurons is vital for representing sensory information (**cohen**). The binomial distribution falls short in this regard, but it is useful as performance benchmark when assessing the performance of other models.

Beta-binomial distribution

The beta distribution is the conjugate distribution of the binomial distribution. The beta-binomial distribution is the combination of the beta distribution and the binomial distribution, in that the probability of success for the binomial distribution is sampled from the beta distribution. This allows the beta-binomial distribution to capture some over dispersion relative to the binomial distribution.

The beta-binomial distribution is a three parameter distribution, n the number of Bernoulli trials, and $\alpha \in \mathbb{R}_{>0}$ and $\beta \in \mathbb{R}_{>0}$ the shape parameters of the beta distribution. The probability mass function for the beta-binomial distribution is

$$P(k; n, \alpha, \beta) = \binom{n}{k} \frac{B(k + \alpha, n - k + \beta)}{B(\alpha, \beta)} \tag{3.3}$$

where $B(\alpha, \beta)$ is the beta function.

This probability distribution can be reparametrised in a number of ways. One of which defines new parameters π and ρ by

$$\pi = \frac{\alpha}{\alpha + \beta} \tag{3.4}$$

$$\rho = \frac{1}{\alpha + \beta + 1} \tag{3.5}$$

This reparametrisation is useful because π acts as a location parameter analogous to the p parameter of a binomial distribution. A value of $\rho > 0$ indicates over-dispersion relative to a binomial distribution.

As a model for the activity of a neuronal ensemble, the beta-binomial distribution is more suitable than a binomial distribution because the over-dispersion of the beta-binomial distribution can be used to model positive association between the neurons. An extreme example of this over-dispersion/positive association can be seen in figure 3.7b. In this figure, the neurons are positively associated and so tend to take the same value, consequently the probability mass of the beta-binomial distribution builds up close to $k = 0$ and $k = n$. It is worth noting that the location parameter for each distribution has the same value, $p = \pi = 0.5$.

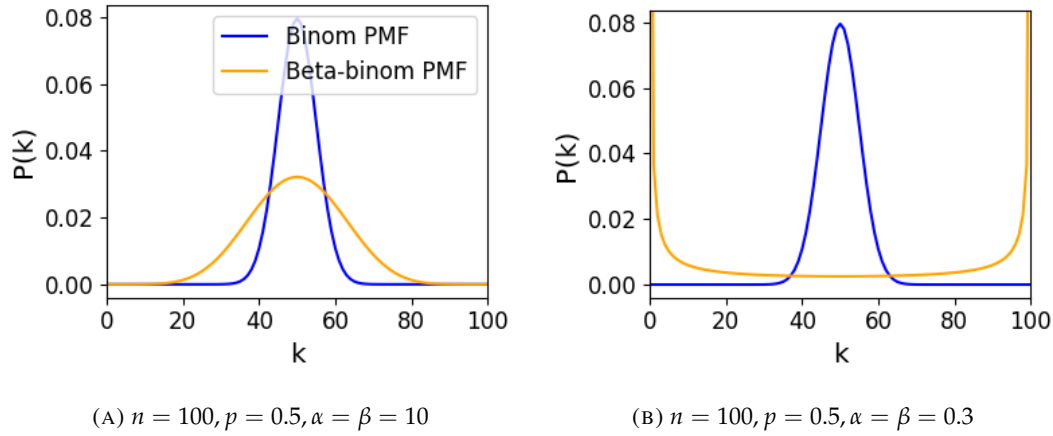(A) $n = 100, p = 0.5, \alpha = \beta = 10$ (B) $n = 100, p = 0.5, \alpha = \beta = 0.3$

FIGURE 3.7: Figures showing the over-dispersion possible for a beta-binomial distribution relative to a binomial distribution. Parameters are shown in the captions.

Conway-Maxwell-binomial distribution

The Conway-Maxwell-binomial distribution (COMb distribution) is a three parameter generalisation of the binomial distribution that allows for over dispersion and under dispersion relative to the binomial distribution. The parameters of the distribution are n the number of Bernoulli trials, and two shape parameters $0 \leq p \leq 1$, and $\nu \in \mathbb{R}$.

The probability mass function of the COMb distribution is

$$P(k; n, p, \nu) = \frac{1}{S(n, p, \nu)} \binom{n}{k}^{\nu} p^k (1-p)^{n-k} \tag{3.6}$$

ν	**Relative dispersion**	**Associaton between neurons/variables**
< 1	over	positive
1	none	none
> 1	under	negative

TABLE 3.3: Relative dispersion of the COMb distribution, and association between Bernoulli variables as represented by the value of the ν parameter.

where

$$S(n, p, \nu) = \sum_{j=0}^{n} \binom{n}{k}^{\nu} p^j (1-p)^{n-j} \tag{3.7}$$

1830 The only difference between this PMF and the PMF for the standard binomial is the intro-

1831 duction of ν and the consequent introduction of the normalising function $S(n, p, \nu)$.

Indeed, if $\nu = 1$ the COMb distribution is identical to the binomial distribution with the same values for n and p. We can see in figure 3.8d that the KL-divergence $D_{KL}(P_{COMb}(n, p, \nu)||P_{Bin}(n, p)) = 0$ along the line where $\nu = 1$. The analytical expression for the divergence is

$$D_{KL}(P_{COMb}(k; n, p, \nu)||P_{Bin}(k; n, p)) = (\nu - 1) E_{P_{COMb}(k; n, p, \nu)} \left[\log \binom{n}{k} \right] \tag{3.8}$$

$$- \log S(n, p, \nu) \tag{3.9}$$

1832 At $\nu = 1$, we have $S(n, p, 1)$ which is just the sum over the binomial PMF, so $S(n, p, 1) = 1$

1833 and therefore $D_{KL}(P_{COMb}(n, p, \nu)||P_{Bin}(n, p)) = 0$.

1834 If $\nu < 1$ the COMb distribution will exhibit over-dispersion relative to the binomial

1835 distribution. If $p = 0.5$ and $\nu = 0$ the COMb distribution is the discrete uniform distribution,

1836 and if $\nu < 0$ the mass of the COMb distribution will tend to build up near $k = 0$ and $k = n$.

1837 This over-dispersion represents positive association in the Bernoulli variables. An example

1838 of this over-dispersion can be seen in figure 3.8b.

1839 If $\nu > 1$ the COMb distribution will exhibit under-dispersion relative to the binomial

1840 distribution. The larger the value of ν the more probability mass will build up at $n/2$ for even

1841 n, or at $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ for odd n. This under-dispersion represents negative association

1842 in the Bernoulli variables. An example of this under-dispersion can be seen in figure 3.8a.

1843 It should be noted that the p parameter of the COMb distribution does not correspond to

1844 the mean of the distribution, as is the case for the binomial p parameter, and beta-binomial

1845 π parameter. That is, the COMb p parameter is not a location parameter. An illustration of

1846 this can be seen in figure 3.8c. This is because an interaction between the p and ν parameters

1847 skews the mean. There is no analytical expression for the mean of the COMb distribution.

1848   Since the COMb distribution has the potential to capture positive and negative associatons

1849   between the neurons/Bernoulli variables, it should be an excellent candidate for modelling

1850   the number of active neurons in a neuronal ensemble.

1851   We wrote a dedicated Python package to enable easy creation and fitting of COMb dis-

1852   tribution objects. The format of the package imitates the format of other distribution objects

1853   from the `scipy.stats` Python package. The COMb package can be found here:

1854   https://github.com/thomasjdelaney/Conway_Maxwell_Binomial_Distribution

(A) $n = 100, p = 0.5, \alpha = \beta = 10, \nu = 2.5$

(B) $n = 100, p = 0.5, \alpha = \beta = 0.3, \nu = 0.1$

(C) $n = 100, p = 0.4, \alpha = 10, \beta = 15, \nu = 0.1$

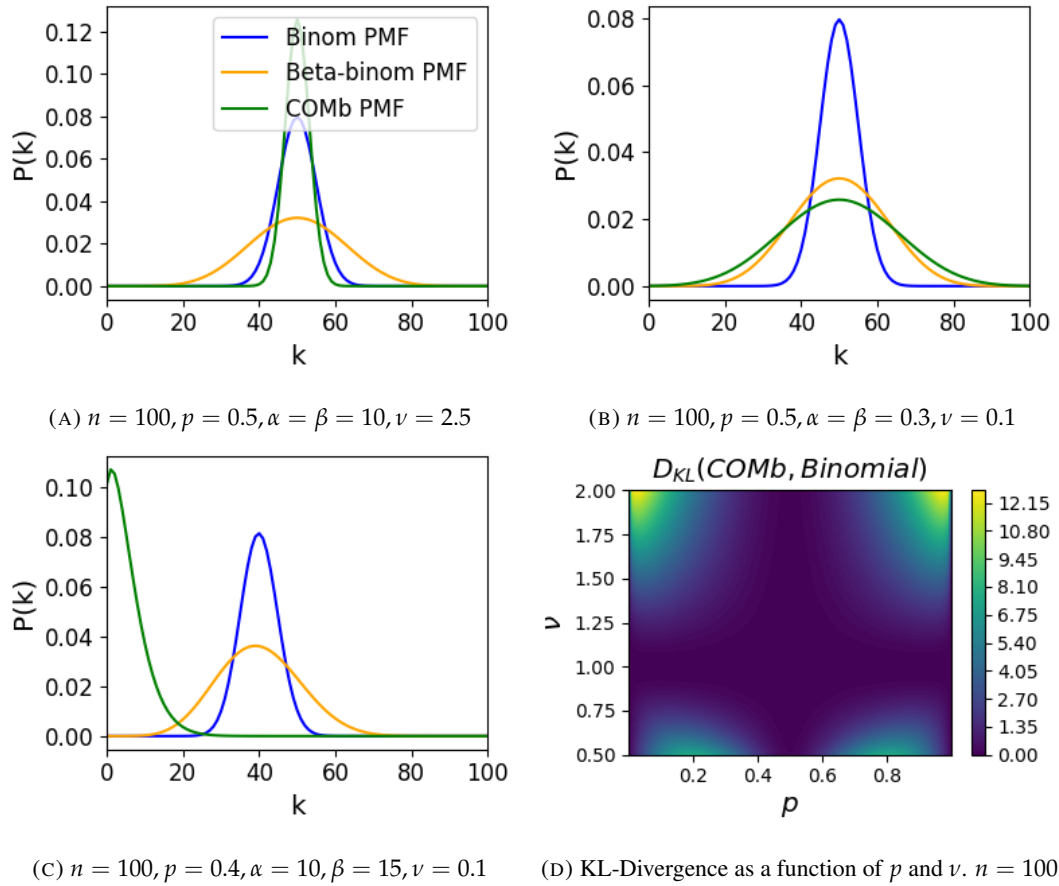(D) KL-Divergence as a function of p and ν. $n = 100$.

FIGURE 3.8: Figures showing (a) the over-dispersion and (b) under-dispersion permitted by the COMb distribution relative to a binomial distribution. (c) illustrates that the p parameter of the COMb distribution does not correspond to the mean of the distribution, as it does for the binomial and beta-binomial distributions. (d) shows a heatmap for the value of the Kullback-Liebler divergence between the COMb distribution and the standard binomial distribution with same value for n, as a function of p and ν. Parameters are shown in the captions.

3.5.6 Fitting

We fitted binomial, beta-binomial, and Conway-Maxwell-binomial (COMb) distributions to
the neural activity in each of the overlapping windows covering each trial. To fit the distribu-
tions we minimised the appropriate negative log likelihood function using the data from the
window.

There is an analytical solution for maximum likelihood estimate of the binomial distribu-
tion's p parameter.

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{N} k_i \tag{3.10}$$

We minimised the negative log likelihood function of the beta-binomial distribution nu-
merically. We calculated the negative log likelihood for a sample directly, by taking the sum
of the log of the probability mass function for each value in the sample. We minimised the
negation of that function using the `minimise` function of the `scipy.optimize` Python
package.

The log likelihood function of the COMb distribution given some sample
$\{k_1, \ldots, k_N\}$ is

$$\ell(p, \nu | k_1, \ldots, k_N) = N \left[n \log(1 - p) - \log S(n, p, \nu) \right] \tag{3.11}$$

$$+ \log \frac{p}{1 - p} \sum_{i=1}^{N} k_i \tag{3.12}$$

$$+ \nu \sum_{i=1}^{N} \log \binom{n}{k_i} \tag{3.13}$$

We minimised the negation of this function using numerical methods. More specifically, we
used the `minimise` function of the `scipy.optimize` Python package.

3.5.7 Goodness-of-fit

After fitting, we measured the goodness-of-fit of each model/distribution with their log like-
lihood. We calculated this directly using the `logpmf` functions of the distribution objects in
Python.

Chapter 4

Studies with practical limitations & negative results

Abstract

Doctor of Philosophy

Investigating, implementing, and creating methods for analysing large neuronal ensembles

by Thomas J. DELANEY

Here I will present some details on research topics that I started, but that unfortunately did not lead anywhere useful. There are two pieces of research, based on two papers. Each paper is related to the overall theme of my PhD of analysing and modelling behaviours of populations of neurons. The first part is based on a model of parallel spike trains including higher order interactions by Shimazaki et al (2012). The second part is based on a multiscale model for making inferences on hierarchical data.

4.1   Dynamic state space model of pairwise and higher order neuronal correlations

In their paper Shimazaki et al (2012) aimed to model spike trains from populations of neurons in parallel, with pairwise and higher order dynamic interactions between the trains. They modelled the spike trains as multi-variate binary processes using a log-linear model, and they extracted spike interaction parameters using a Bayesian filter/EM-algorithm. They developed a goodness-of-fit measure for the model to test if including these higher order correlations is necessary for an accurate model. Their measure was based on the Bayes factor but they also assessed the suitability of higher order models using the AIC and BIC. So the increase in the number of parameters associated with fitting higher order interactions was taken into account. They tested the performance of the model on synthetic data with known higher order correlations. They used the model to look for higher order correlations in data from awake behaving animals. They use the model to demonstrate dynamic appearance of higher order correlations in the monkey motor cortex**shimazaki**.

We used the available Python repository to implement the model, and we successfully worked through the tutorial provided. But we found that the model did not scale well to larger populations. We attempted to fit the model to a population of 10 neurons and found we didn't manage to finish the process. Since, our goal was to find a model to scale to hundreds or thousands of neurons, we decided that this model was no longer worth pursuing.

4.2   A multiscale model for hierarchical data applied to neuronal data

In their paper Kolacayk et al (2001) developed a framework for a modelling hierarchically aggregated data, and making inferences based on a model arising from this framework. They assumed that a hierarchical aggregation existed on the data in question, where each element at each level of the hierarchy had some associated measurements, an associated mean process, which was the expected value of these measurements. They also assumed that the measurements of each parent were equal to the sum of the measurements from all of its children. They showed that these assumptions gave rise to a relationship between parent and child measurements across all levels of the hierarchy, where the product of the likelihood of the parameters of the lowest level of the hierarchy can be expressed as products of conditional likelihoods of the elements of higher levels of the hierarchy**kolacayk**.

They gave examples of these expressions for measurements sampled from Gaussian distributions, and Poisson distributions, and showed the definitions of the hierarchical parameters which reparametrise the distribution of these data taking the hierarchy into account. They go on to suggest prior distributions for this multiscale model, and integrate these priors to give posterior distributions for the measurements from each element at each level in the hierarchy, and expressions for the MAP estimated parameters of each the associated processes**kolacayk**.

We implemented their model in Python by creating some synthetic data from Poisson distributions, and defining a hierarchy by agglomerating these data. We calculated the MAP estimates using our knowledge of the hierarchy, and using the expressions given in the paper. We found that the MAP estimates were far less accurate than would be achieved by simply ignoring the hierarchy during estimation, and using a maximum likelihood approach. After that, we decided to move on.

Bibliography

Baadel, S., F. Thabtah, and J. Lu (2016). "Overlapping clustering: A review". In: *2016 SAI Computing Conference (SAI)*, pp. 233–237. DOI: 10.1109/SAI.2016.7555988.

Bartol, Thomas M. et al. (2015). "Computational reconstitution of spine calcium transients from individual proteins". In: *Frontiers in Synaptic Neuroscience* 7, p. 17. ISSN: 1663-3563. DOI: 10.3389/fnsyn.2015.00017. URL: https://www.frontiersin.org/article/10.3389/fnsyn.2015.00017.

Berens, Philipp et al. (May 2018). "Community-based benchmarking improves spike rate inference from two-photon calcium imaging data". In: *PLOS Computational Biology* 14.5, pp. 1–13. DOI: 10.1371/journal.pcbi.1006157. URL: https://doi.org/10.1371/journal.pcbi.1006157.

Bezanson, Jeff et al. (Sept. 2012). "Julia: A Fast Dynamic Language for Technical Computing". In: *MIT*.

Chen, Tsai-Wen et al. (July 2013). "Ultrasensitive fluorescent proteins for imaging neuronal activity". In: *Nature* 499, 295–300. DOI: 10.1038/nature12354. URL: https://doi.org/10.1038/nature12354.

Churchland, Mark M. et al. (2010). "Stimulus onset quenches neural variability: a widespread cortical phenomenon". eng. In: *Nature neuroscience* 13.3. 20173745[pmid], pp. 369–378. ISSN: 1546-1726. DOI: 10.1038/nn.2501. URL: https://pubmed.ncbi.nlm.nih.gov/20173745.

Cohen, Marlene R. and Adam Kohn (2011). "Measuring and interpreting neuronal correlations". In: *Nature Neuroscience* 14.7, pp. 811–819. ISSN: 1546-1726. DOI: 10.1038/nn.2842. URL: https://doi.org/10.1038/nn.2842.

Cohen, Marlene R. and John H. R. Maunsell (2009). "Attention improves performance primarily by reducing interneuronal correlations". eng. In: *Nature neuroscience* 12.12. 19915566[pmid], pp. 1594–1600. ISSN: 1546-1726. DOI: 10.1038/nn.2439. URL: https://pubmed.ncbi.nlm.nih.gov/19915566.

deCharms, R. Christopher and Michael M. Merzenich (1996). "Primary cortical representation of sounds by the coordination of action-potential timing". In: *Nature* 381.6583,

pp. 610–613. ISSN: 1476-4687. DOI: `10.1038/381610a0`. URL: `https://doi.org/10.1038/381610a0`.

Deneux, Thomas et al. (July 2016). "Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo". In: *Nature Communications* 7.1. DOI: `10.1038/ncomms12190`. URL: `https://doi.org/10.1038/ncomms12190`.

Faas, G. C. et al. (Mar. 2011). "Calmodulin as a direct detector of Ca2+ signals". In: *Nature neuroscience* 14.3, pp. 301–304. DOI: `10.1038/nn.2746`. URL: `https://dx.doi.org/10.1038/nn.2746`.

Fiala, John C. and Kristen M. Harris (1999). *Dendrite Structure*.

Flach, Boris (Sept. 2013). "A Class of Random Fields on Complete Graphs with Tractable Partition Function". In: *IEEE transactions on pattern analysis and machine intelligence* 35, pp. 2304–6. DOI: `10.1109/TPAMI.2013.99`.

Fosdick, Bailey et al. (Aug. 2016). "Configuring Random Graph Models with Fixed Degree Sequences". In: *SIAM Review* 60. DOI: `10.1137/16M1087175`.

Friedrich, Johannes and Liam Paninski (2016). "Fast Active Set Methods for Online Spike Inference from Calcium Imaging". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., pp. 1984–1992. URL: `http://papers.nips.cc/paper/6505-fast-active-set-methods-for-online-spike-inference-from-calcium-imaging.pdf`.

Ganmor, Elad, Ronen Segev, and Elad Schneidman (2011). "Sparse low-order interaction network underlies a highly correlated and learnable neural population code". In: *Proceedings of the National Academy of Sciences* 108.23, pp. 9679–9684. ISSN: 0027-8424. DOI: `10.1073/pnas.1019641108`. eprint: `https://www.pnas.org/content/108/23/9679.full.pdf`. URL: `https://www.pnas.org/content/108/23/9679`.

Greenberg, David et al. (Nov. 2018). "Accurate action potential inference from a calcium sensor protein through biophysical modeling:" in: DOI: `10.1101/479055`.

Humphries, Mark et al. (Jan. 2019). "Spectral rejection for testing hypotheses of structure in networks". In:

Humphries, Mark D. (2011). "Spike-Train Communities: Finding Groups of Similar Spike Trains". In: *Journal of Neuroscience* 31.6, pp. 2321–2336. ISSN: 0270-6474. DOI: `10.1523/JNEUROSCI.2853-10.2011`. eprint: `https://www.jneurosci.org/`

content/31/6/2321.full.pdf. URL: https://www.jneurosci.org/
content/31/6/2321.

Jun, James J. et al. (2017). "Fully integrated silicon probes for high-density recording of
neural activity". In: *Nature* 551.7679, pp. 232–236. ISSN: 1476-4687. DOI: 10.1038/
nature24636. URL: https://doi.org/10.1038/nature24636.

Kilhoffer, Marie-Claude et al. (Sept. 1992). "Use of Engineered Proteins With Internal Tryp-
tophan Reporter Groups and Pertubation Techniques to Probe the Mechanism of Ligand-
Protein Interactions: Investigation of the Mechanism of Calcium Binding to Calmod-
ulin". In: *Biochemistry* 31.34, pp. 8098–8106. DOI: 10.1021/bi00149a046. URL:
https://doi.org/10.1021/bi00149a046.

Koch, Christoff (1999). *Biophysics of Computation: Information Processing in Single Neu-
rons*. Oxford University Press. ISBN: ISBN 0-19-510491-9.

Lee, Suk-HO et al. (July 2000). "Differences in Ca2+ buffering properties between excitatory
and inhibitory hippocampal neurons from the rat". In: *The Journal of Physiology* 525.
DOI: 10.1111/j.1469-7793.2000.t01-3-00405.x.

Litwin-Kumar, Ashok, Maurice Chacron, and Brent Doiron (Sept. 2012). "The Spatial Struc-
ture of Stimuli Shapes the Timescale of Correlations in Population Spiking Activity".
In: *PLoS computational biology* 8, e1002667. DOI: 10.1371/journal.pcbi.
1002667.

Maravall, M et al. (May 2000). "Estimating intracellular calcium concentrations and buffer-
ing without wavelength ratioing". In: *Biophysical Journal* 78.5, pp. 2655–2667. DOI:
10.1016/S0006-3495(00)76809-3. URL: https://doi.org/10.1016/
s0006-3495(00)76809-3.

Maugis, Pa (Jan. 2014). "Event Conditional Correlation: Or How Non-Linear Linear Depen-
dence Can Be". In:

Meilă, Marina (2007). "Comparing clusterings—an information based distance". In: *Journal
of Multivariate Analysis* 98.5, pp. 873 –895. ISSN: 0047-259X. DOI: https://doi.
org/10.1016/j.jmva.2006.11.013. URL: http://www.sciencedirect.
com/science/article/pii/S0047259X06002016.

Neher, E. and G. J. Augustine (1992). "Calcium gradients and buffers in bovine chromaffin
cells". eng. In: *The Journal of physiology* 450. 1331424[pmid], pp. 273–301. ISSN: 0022-
3751. DOI: 10.1113/jphysiol.1992.sp019127. URL: https://pubmed.
ncbi.nlm.nih.gov/1331424.

O'Donnell, Cian et al. (Jan. 2017). "The population tracking model: a simple, scalable statistical model for neural population data". English. In: *Neural Computation* 29.1, pp. 50–93. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00910.

Pnevmatikakis, E.A. et al. (Jan. 2016). "Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data". In: *Neuron* 89.2, pp. 285–299. DOI: 10.1016/j.neuron.2015.11.037. URL: https://doi.org/10.1016/j.neuron.2015.11.037.

Pnevmatikakis, Eftychios et al. (Nov. 2013). "Bayesian spike inference from calcium imaging data". In: *Conference Record - Asilomar Conference on Signals, Systems and Computers*. DOI: 10.1109/ACSSC.2013.6810293.

Pnevmatikakis, Eftychios et al. (Sept. 2014). "A structured matrix factorization framework for large scale calcium imaging data analysis". In:

Schneidman, Elad et al. (2006). "Weak pairwise correlations imply strongly correlated network states in a neural population". eng. In: *Nature* 440.7087. 16625187[pmid], pp. 1007–1012. ISSN: 1476-4687. DOI: 10.1038/nature04701. URL: https://pubmed.ncbi.nlm.nih.gov/16625187.

Shannon, C. E. (1948). "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x.

Steinmetz, Nick, Matteo Carandini, and Kenneth D. Harris (2019). *"Single Phase3" and "Dual Phase3" Neuropixels Datasets*. DOI: 10.6084/m9.figshare.7666892.v2. URL: https://figshare.com/articles/dataset/_Single_Phase3_Neuropixels_Dataset/7666892/2.

Stringer, Carsen et al. (2019). "Spontaneous behaviors drive multidimensional, brainwide activity". In: *Science* 364.6437. ISSN: 0036-8075. DOI: 10.1126/science.aav7893. eprint: https://science.sciencemag.org/content/364/6437/eaav7893.full.pdf. URL: https://science.sciencemag.org/content/364/6437/eaav7893.

Tada, Mayumi et al. (2014). "A highly sensitive fluorescent indicator dye for calcium imaging of neural activity in vitro and in vivo". In: *European Journal of Neuroscience* 39.11, pp. 1720–1728. DOI: 10.1111/ejn.12476. eprint: https://onlinelibrary.

wiley.com/doi/pdf/10.1111/ejn.12476. URL: https://onlinelibrary.
wiley.com/doi/abs/10.1111/ejn.12476.

Tian, Lin et al. (2009). "Imaging neural activity in worms, flies and mice with improved
GCaMP calcium indicators". eng. In: *Nature methods* 6.12. 19898485[pmid], pp. 875–
881. ISSN: 1548-7105. DOI: 10.1038/nmeth.1398. URL: https://pubmed.
ncbi.nlm.nih.gov/19898485.

Tkačik, Gašper et al. (2014). "Searching for collective behavior in a large network of sen-
sory neurons". eng. In: *PLoS computational biology* 10.1. 24391485[pmid], e1003408–
e1003408. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003408. URL:
https://pubmed.ncbi.nlm.nih.gov/24391485.

Treves, Alessandro and Stefano Panzeri (1995). "The Upward Bias in Measures of Informa-
tion Derived from Limited Data Samples". In: *Neural Computation* 7.2, pp. 399–407.
DOI: 10.1162/neco.1995.7.2.399. eprint: https://doi.org/10.1162/
neco.1995.7.2.399. URL: https://doi.org/10.1162/neco.1995.7.
2.399.

Vinh, Nguyen Xuan, Julien Epps, and James Bailey (Dec. 2010). "Information Theoretic
Measures for Clusterings Comparison: Variants, Properties, Normalization and Correc-
tion for Chance". In: *J. Mach. Learn. Res.* 11, 2837–2854. ISSN: 1532-4435.

Vogelstein, Joshua T. et al. (Oct. 2010). "Fast nonnegative deconvolution for spike train infer-
ence from population calcium imaging". In: *Journal of neurophysiology* 104.6, 295–300.
DOI: 10.1152/jn.01073.2009. URL: https://dx.doi.org/10.1152%
2Fjn.01073.2009.

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the Elastic
Net". In: *Journal of the Royal Statistical Society, Series B* 67, pp. 301–320.