

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

Thomas Delaney¹ and Cian O'Donnell¹

¹*School of Computer Science, Electrical and Electronic Engineering, and Engineering Mathematics, University of Bristol, Bristol, United Kingdom.*

Abstract

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 brain regions in three awake and behaving mice. Using 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 ($< 100\text{ms}$).

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 [1]. Recent findings show that spontaneous behaviours explain correlations in parts of the brain not associated with motor control [4]. 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 [6] 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 [4].

A drawback associated with the improvement in recording technology is an increase in the difficulty of analysing these data. For example, analysing the i th 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 spike count correlations between 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 Results

2.1 Average correlation size increases with increasing time bin width

First we inspected the affect of time bin width on spike count correlations. We know that using short time bins results in artificially small correlation measurements[1], 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 spike count 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 1a).

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 5.2.3). 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 1b and 1c).

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.

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 distributon of spike counts. We used the χ^2 statistic as a goodness-of-fit measure for Poisson and Gaussian (normal) distributons 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 distributon 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 1d).

87 A Poisson distributon was very likely to be a better fit for shorter time bin widths
 88 because spike counts must be non-negative, therefore any distributon of spike counts
 89 with mass distributed at or close to 0 will be skewed.

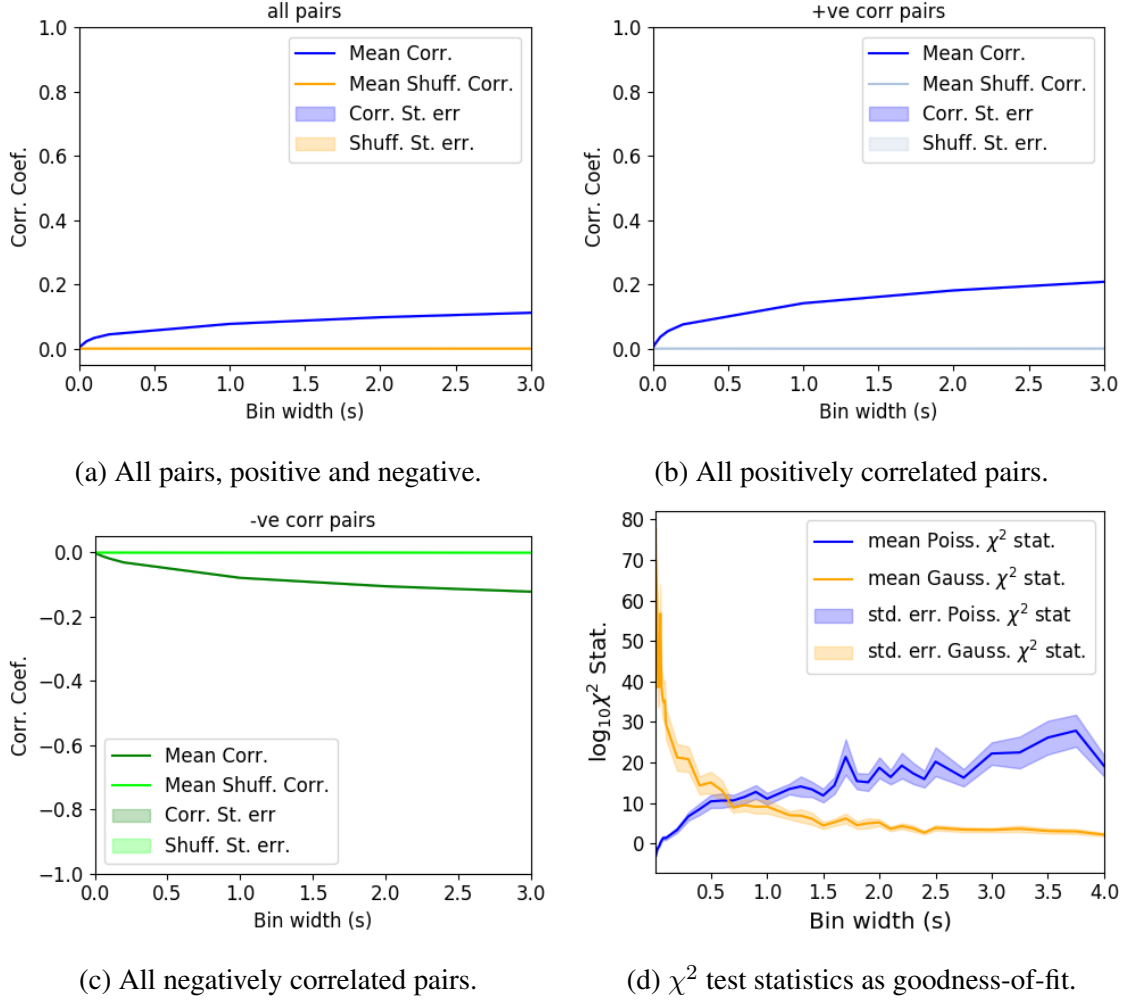


Figure 1: Mean correlation coefficients measured from pairs of 50 randomly chosen neurons. (a) All possible pairs, (b) positively correlated pairs, and (c) negatively correlated pairs. (d) Mean and standard error of χ^2 test statistics for Poisson and Gaussian distributions fitted to neuron spike counts.

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 ~ 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 cell from one region would not dominate our data. We observed that the mean intra-regional correlations were always higher than the 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). 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 spike count correlations, they found that the mean ‘within-region’ correlations were always greater than the ‘out-of-region’ correlations [4].

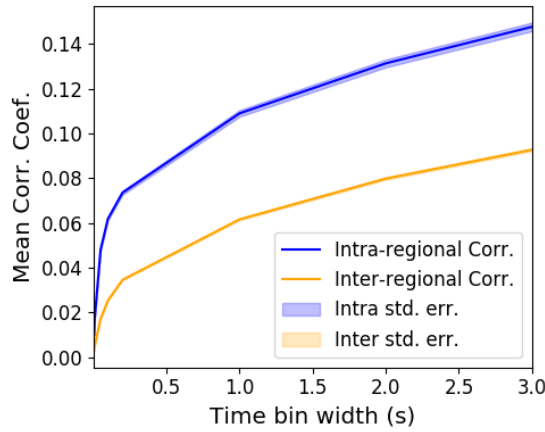
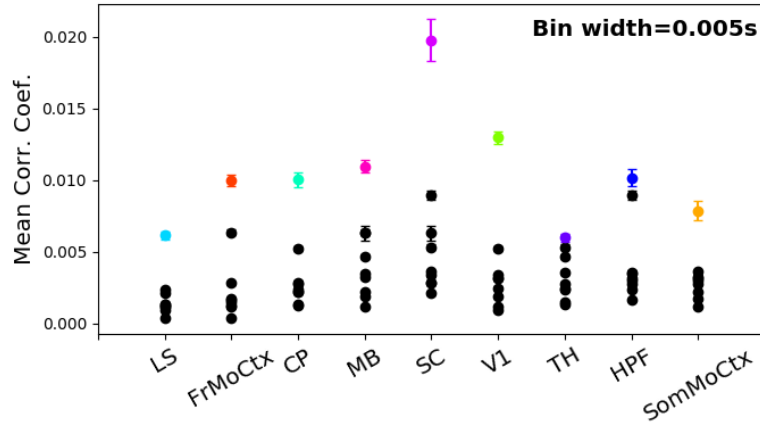
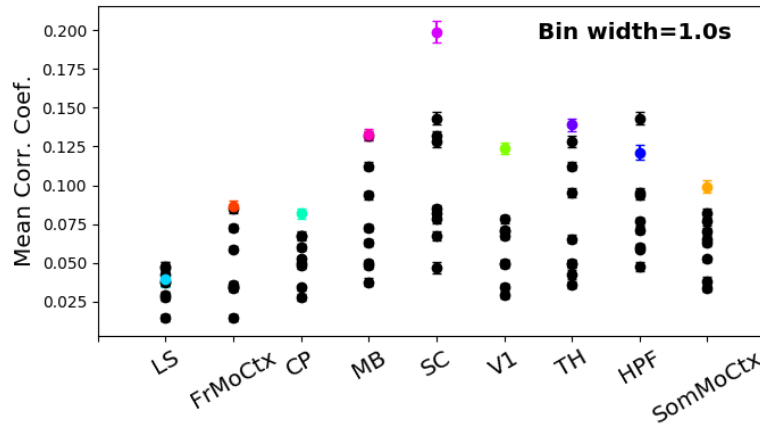


Figure 2: The mean intra-region and inter-region correlations using all possible pairs of ~ 500 neurons, spread across 9 different brain regions.

Secondly, we separated those pairs into their respective pairs, be they intra-regional, or inter-regional. 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 and most mean inter-regional correlations of the most correlated regional pairs reduced as we increased the time bin width (see figures 3a and 3b).



(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 3: 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 correlations 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.

114 Finally, to see these regional mean correlations in a bit more detail, we displayed
 115 these data using mean correlation matrix figures (see figure 4), showing the mean
 116 intra-regional correlations on the main diagonal, and the mean inter-regional corre-
 117 lations off diagonal. Comparing a version of this figure created using a short time
 118 bin width of 5ms (figure 4a) and a version using a longer time bin width of 1s (figure
 119 4b) we observed that the mean intra-regional correlations are the highest for both
 120 bin widths, but the mean correlations in some inter-regional pairs is relatively much
 121 higher when using the longer time bin width.

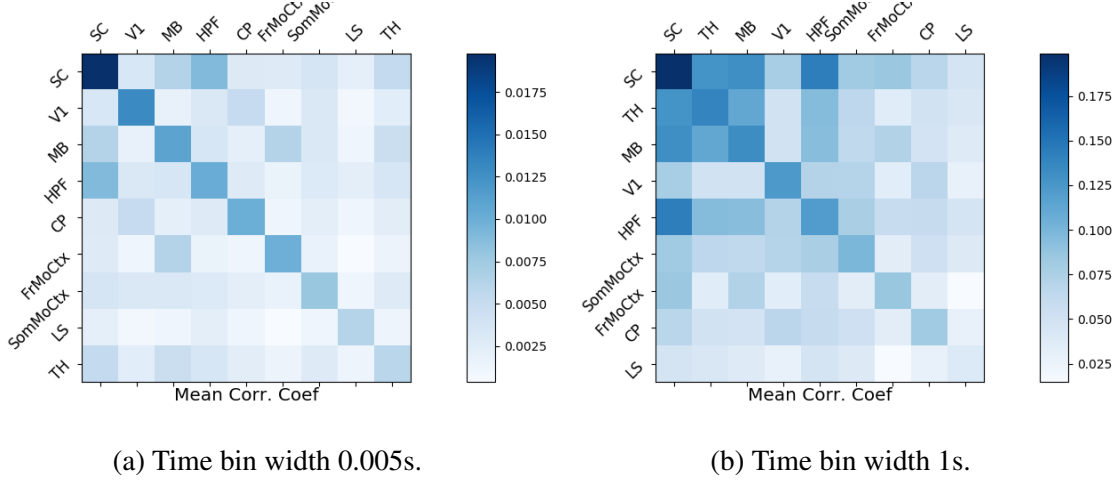


Figure 4: Mean inter-regional (main diagonal) and intra-regional (off diagonal) correlation coefficients. (a) Shows these measurements when spike times were binned using 5ms time bins. (b) Shows the same, using 1s time bins. Note that the matrices are ordered according to the main diagonal values, therefore the ordering is different in each subfigure.

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

127 2.4 Connected and k -partite structure in correlation based 128 networks reduces in dimension with increasing bin width

129 We used the correlation measurements to create weighted undirected graphs/networks
 130 where each node represents a neuron, and the weight of each edge is the pairwise cor-
 131 relation between those neurons represented by the nodes at either end of that edge.
 132 We aimed to find communities of neurons within these networks, and compare the
 133 structure of these communities to the anatomical division of those neurons. The
 134 first step of this process involved applying the ‘spectral rejection’ technique devel-
 135 oped by Humphries et al (2019)[3]. This technique compares our data network to
 136 a chosen null network model, and finds any additional structure in the data network
 137 beyond that which is captured in the null network model (if there is any such struc-
 138 ture). 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 5.5.7 and 5.5.8 respectively, and a full overview can be found in [3].

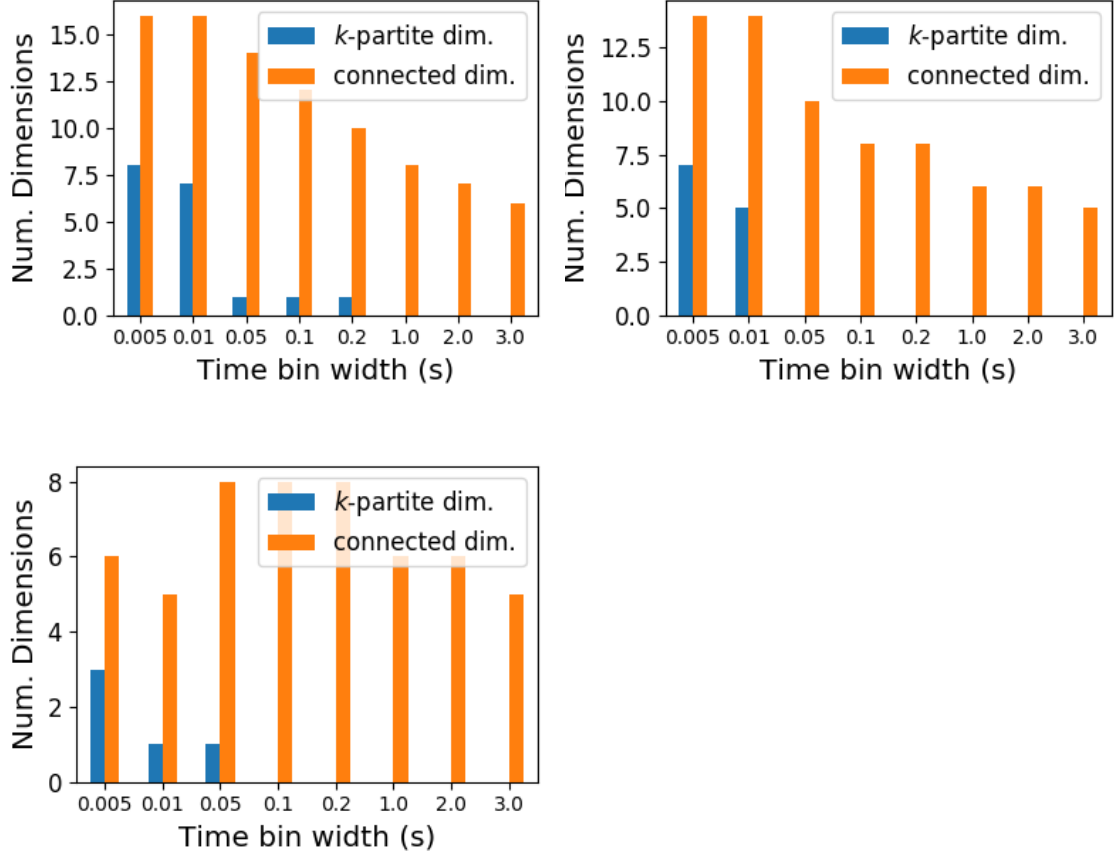


Figure 5: The number of dimensions in the k -partite and connected structure in the correlation based networks beyond the structure captured by a sparse weighted configuration null network model (see section 5.5.6), shown for different time bin widths. Note that the k -partite structure disappears for time bin width greater than 200ms for all three subjects. The dimension of the connect structure reduces with increasing bin width for 2 of the 3 subjects (top row).

We chose the sparse weighted configuration model (see section 5.5.6) 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 spike count

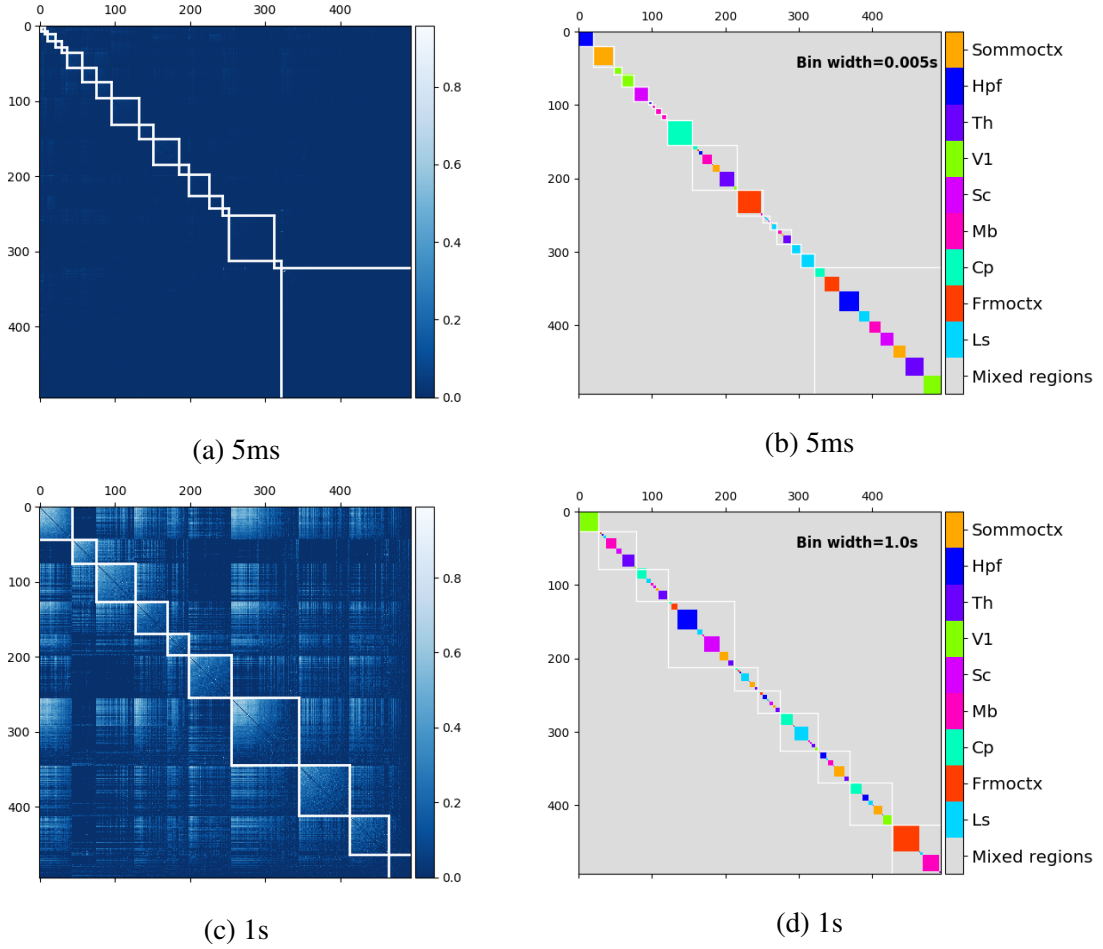


Figure 6: Test caption

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 structure reduced in two of the three mice from which we had data (see figure 5).

2.5 Detecting communities in correlation based networks

We applied the community detection procedure described in section 5.5.9 to our signal networks for our various time bin widths. As might be expected from the results described above in section 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 6).

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.

2.6 Functional communities resemble anatomical division at short timescales

2.7 Conditional correlations & signal correlations

3 Discussion

To be added.

4 Data

The data that we used in this project were collected by Nick Steinmetz and his lab members[4].

4.1 Brain regions

Neuropixels probes were used to collect extracellular recordings [6] 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 (Frmctx)
- Hippocampal formation (Hpf)

- 185 • Lateral Septum (Ls)
- 186 • Midbrain (Mb)
- 187 • Superior Colliculus (Sc)
- 188 • Somatomotor cortex (Sommocx)
- 189 • Thalamus (Th)
- 190 • Primary visual cortex (V1)

191 Readings were continuous and lasted for about 1 hour [4]. Locations of each of the
 192 probes can be seen in figure 7.

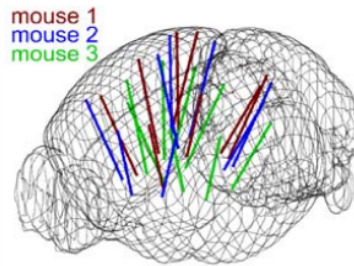


Figure 7: **Probe Locations:** The locations of the probes in each of the three mouse brains[4].

193 4.2 Video recordings

194 Video recordings of the mouse’s face were taken during the spontaneous behaviour.
 195 We had access to the top 500 principle components and top 500 eigenvectors of the
 196 processed videos. The frequency of recording was slightly less than 40Hz. Each
 197 frame contained 327×561 pixels. These principal components were used as be-
 198 havioural data. We controlled for these components when taking measurements con-
 199 ditioned on behaviour.

200 5 Methods

201 5.1 Binning data

202 We transoformed the spike timing data into binned spike count data by dividing the
 203 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.

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:

$$\rho_{XY} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (1)$$

$$= \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (2)$$

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 1, but we have samples from X and Y , Pearson’s sample correlation coefficient is defined as:

$$r_{XY} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3)$$

where $\{(x_i, y_i)\}$ for $i \in \{1, \dots, n\}$ 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.

5.2.1 Spike Count Correlation, r_{SC}

The spike count correlation (r_{SC}) of two cells is the correlation between the spike counts of those cells in response to a given stimulus condition. In this study, there was only one stimulus condition, that of no stimulus. The subjects engaged in spontaneous behaviour during recording.

5.2.2 Shuffled spike count correlations

We measured the shuffled spike count correlations between two neurons by randomly permuting one of the neuron's spike counts and measuring the spike count 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 5.5.3).

5.2.3 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.

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 4.2). Denoting the spike count of a given cell by X , and the PCs by Z_1, \dots, Z_{500} , we wanted to model X as a function of Z_1, \dots, Z_{500} in order to estimate

$$E[X|Z_1, \dots, Z_{500}] = \int_{x \in X} x P(X = x|Z_1, \dots, Z_{500}) dx \quad (4)$$

$$= \int_{x \in X} x \frac{P(X = x, Z_1, \dots, Z_{500})}{P(Z_1, \dots, Z_{500})} dx \quad (5)$$

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

241 tion of the PCs.

242 **5.3.1 Linear regression**

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

- 246 • $L1$ or 'Lasso'
- 247 • $L2$ or 'Ridge regression'
- 248 • 'Elastic net' regularisation (a linear combination of both $L1$ and $L2$ regulari-
249 sation penalties)

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

251 **5.3.2 Elastic net regularisation**

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

$$L(\lambda_1, \lambda_2, \boldsymbol{\beta}) = |\mathbf{x} - \mathbf{Z}\boldsymbol{\beta}|^2 + \lambda_2 |\boldsymbol{\beta}|_2 + \lambda_1 |\boldsymbol{\beta}|_1 \quad (6)$$

where

$$|\boldsymbol{\beta}|_2 = \sum_{j=1}^m \beta_j^2 \quad (7)$$

$$|\boldsymbol{\beta}|_1 = \sum_{j=1}^m |\beta_j| \quad (8)$$

The naïve elastic net estimator $\hat{\boldsymbol{\beta}}$ is the minimiser of the system of equations 6 [12]

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} L(\lambda_1, \lambda_2, \boldsymbol{\beta}) \quad (9)$$

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

254 As well as using the PCs, we also tried fitting the models using the raw video data
255 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.

5.3.3 Conditional covariance

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

$$\text{cov}(X, Y) = E[\text{cov}(X, Y|Z)] + \text{cov}(E[X|Z], E[Y|Z]) \quad (10)$$

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

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

$$E[\text{cov}(X, Y|Z_1, \dots, Z_{500})] = \text{cov}(X, Y) - \quad (11)$$

$$\text{cov}(E[X|Z_1, \dots, Z_{500}], E[Y|Z_1, \dots, Z_{500}]) \quad (12)$$

5.3.4 Measures of conditional correlation

As a measure of expected correlation, we measured the ‘event conditional correlation’ [13]

$$\rho_{XY|Z} = \frac{E[\text{cov}(X, Y|Z)]}{\sqrt{E[\text{var}(X|Z)]E[\text{var}(Y|Z)]}} \quad (13)$$

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’

$$\rho_{\text{signal}} = \frac{\text{cov}(E[X|Z], E[Y|Z])}{\sqrt{\text{var}(E[X|Z])\text{var}(E[Y|Z])}} \quad (14)$$

this is an actual correlation.

5.4 Information Theory

5.4.1 Entropy $H(X)$

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

$$H(X) = - \sum_{n=1}^N p_n \log_2 p_n \quad (15)$$

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 [5]. Note that any base may be used for the logarithm in equation 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, \dots, y_M , is defined as

$$H(X, Y) = - \sum_{n=1}^N \sum_{m=1}^M P(X = x_n, Y = y_m) \log_2 P(X = x_n, Y = y_m) \quad (16)$$

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 are completely dependent $H(X, Y) = H(X) = H(Y)$.

The conditional entropy of Y conditioned on X is defined as

$$H(Y|X) = - \sum_{n=1}^N \sum_{m=1}^M P(X = x_n, Y = y_m) \log_2 \frac{P(X = x_n, Y = y_m)}{P(X = x_n)} \quad (17)$$

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.

5.4.2 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 $i \in \{0, \dots, n_{\max}\}$ 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 8 to see the maximum entropy as a function of the maximum observed spike count.

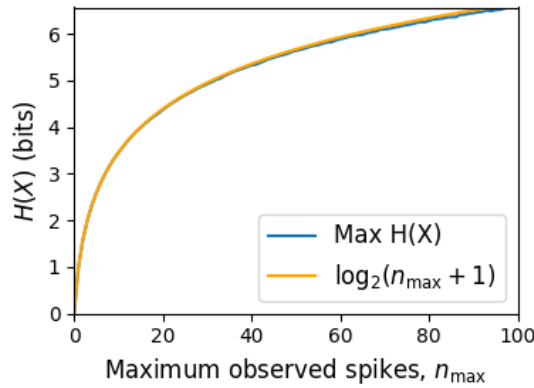


Figure 8: **Entropy Limit:** The upper limit on entropy of binned spike count data as a function of the maximum observed spike count. The orange line is the analytical maximum. The blue line is the entropy of samples with $N = 1000$ data points taken from the discrete uniform distribution.

5.4.3 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

$$I(X; Y) = H(Y) - H(Y|X) \quad (18)$$

$$= H(X) - H(X|Y) \quad (19)$$

Equation 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

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \quad (20)$$

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,

$$I(X; Y) = - \sum_{n=1}^N \sum_{m=1}^M P(X = x_n, Y = y_m) \log_2 \frac{P(X = x_n, Y = y_m)}{P(X = x_n)P(Y = y_m)} \quad (21)$$

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

$$I(X; Y) = D_{KL}(P(X, Y) || P(X)P(Y)) \quad (22)$$

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.

304 The minimum value that $I(X; Y)$ can take is 0. This occurs when the ran-
 305 dom variables X and Y are independent. Then we have $H(X|Y) = H(X)$, and
 306 $H(Y|X) = H(Y)$, which according to equation 18, gives $I(X; Y) = 0$. We
 307 also have that $H(X, Y) = H(X) + H(Y)$ in this case, which according equa-
 308 tion 20, gives $I(X; Y) = 0$. Finally, we also have $P(X, Y) = P(X)P(Y)$, which
 309 leaves us with 1 in the argument for the logarithm in equation 21, which again gives
 310 $I(X; Y) = 0$.

311 The mutual information reaches its maximum value when one of the variables
 312 X and Y is completely determined by knowing the value of the other. In that case
 313 $I(X; Y) = \min\{H(X), H(Y)\}$.

314 **5.4.4 Variation of Information $VI(X, Y)$**

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

$$VI(X; Y) = H(X) + H(Y) - 2I(X; Y) \quad (23)$$

We can rewrite this as the summation of two positive quantities

$$VI(X; Y) = [H(X) - I(X; Y)] + [H(Y) - I(X; Y)] \quad (24)$$

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

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

319 **5.4.5 Measuring entropies & mutual information**

320 In practice, we measured the mutual information between spike counts using Python
 321 and the python package `pyitlib`. We used the PT-bias correction technique to esti-
 322 mate the bias of our measurements when measuring the mutual information between
 323 the spike counts of two cells [7].

324 When measuring the mutual information between clusterings we used Python,
 325 but we used the `mutual_info_score`, `adjusted_mutual_info_score`, and

326 `normalized_mutual_info_score` functions from the `sklearn.metrics`
327 part of the `sklearn` package.

328 **5.5 Network analysis**

329 **5.5.1 Correlation networks**

330 In order to analyse functional networks created by the neurons in our ensemble, we
331 measured the spike count correlation between each pair of neurons. These mea-
332 surements induced an undirected weighted graph/network between the neurons. The
333 weight of each connection was equal to the spike count correlation between each pair
334 of neurons.

335 We followed the same procedure for spike count correlations 5.2.1, conditional
336 correlations, and signal correlations 5.3.4.

337 **5.5.2 Rectified correlations**

338 At the time of writing, the community detection method outlined in [3] could only
339 be applied to networks with positively weighted connections. But many neuron pairs
340 were negatively correlated. To apply the community detection method, we *rectified*
341 the network, by setting all the negative weights to zero.

342 We also looked for structure in the network created by negative correlations by
343 reversing the signs of the correlations, and rectifying these correlations before apply-
344 ing our network analysis.

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

348 **5.5.3 Sparsifying data networks**

349 When creating our correlation networks, we wanted to exclude any correlations that
350 could be judged to exist ‘by chance’. To do this, we measured the 5th and 95th
351 percentile of the shuffled correlations (see section 5.2.2) for the given mouse and
352 time bin width. We then set all the data correlations between these two values to
353 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 5.5.6.

5.5.4 Communities

Given some network represented by an adjacency matrix \mathbf{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 [8]. Since we are working with weighted sparse networks, we used more suitable null models, described below.

5.5.5 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 [8]. 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.

5.5.6 Sparse weighted configuration model

The *sparse weighted configuration model* is another null network model. Similar in nature to the weighted configuration model (see section 5.5.5), but the sparsity of the data network is preserved in the null network. This is achieved by sampling from a probability distribution for the creation or non-creation of each possible connection, then distributing the weight of the data network randomly in this sparse network [3]. This is the null network that we used when searching for additional structure in our data networks.

5.5.7 Spectral rejection

We made use of the spectral rejection algorithm as outlined in [3]. The spectral rejection algorithm is a method for finding structure in a network not captured by a 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

$$\mathbf{B} = \mathbf{W} - \langle \mathbf{P} \rangle \quad (25)$$

a common choice for $\langle \mathbf{P} \rangle$ in community detection is the ‘configuration model’ [8, 9]. The matrix \mathbf{B} is often called the configuration matrix, in this context we will use the term ‘deviation matrix’ as it captures the deviation of \mathbf{W} from the null model.

To test for structure in the network represented by \mathbf{W} , we examine the eigenspectrum of \mathbf{B} and compare it to the eigenspectrum of our null model. Firstly, note that since our data model doesn’t allow self loops, and is not directed, the matrix representing the network will be symmetric and positive semi-definite, and will therefore be invertible with real eigenvalues. We selected a null model with the same characteristics.

To find the eigenspectrum of the null model, we generated N samples from our null model P_1, \dots, P_N , and we measured their deviation matrices B_1, \dots, B_N . We then calculated the eigenspectrum of each of those samples. We calculated the upper bound of the null model eigenspectrum by taking the mean of the largest eigenvalues of B_1, \dots, B_N . We calculated a lower bound on the null model eigenspectrum by taking the mean of the smallest eigenvalues of B_1, \dots, B_N .

We then calculated the eigenspectrum of \mathbf{B} , our data network deviation matrix. If any of those eigenvalues lay outside of the upper or lower bounds of the null model eigenspectrum, this is evidence of additional structure not captured by the null model. If we chose the sparse weighted configuration model (see section 5.5.6) 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 \mathbf{B} lying above the upper bound of the null model eigenspectrum indicates the presence of two communities in the network [9].

5.5.8 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 columns 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, \dots, B_N . The j th eigenvector and eigenvalue of B_i gives a value for a null network's projection into the j th dimension of the additional structure vector space. The matrices B_1, \dots, 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 [3].

5.5.9 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 [9].

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 [3].

5.6 Clustering Comparison

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

$$C_k \cap C_l = \emptyset \quad (26)$$

$$\bigcup_{k=1}^K C_k = D \quad (27)$$

If we consider two clusterings, \mathcal{C} with clusters C_1, C_2, \dots, C_K and \mathcal{C}' with clusters C'_1, C'_2, \dots, 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 .

5.6.1 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

$$RI = \frac{N_{11} + N_{00}}{N_{11} + N_{00} + N_{10} + N_{01}} = \frac{N_{11} + N_{00}}{\binom{n}{2}} \quad (28)$$

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

$$ARI = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00} + N_{01})(N_{01} + N_{11}) + (N_{00} + N_{10})(N_{10} + N_{11})} \quad (29)$$

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

455 **5.6.2 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

$$P(K = k) = \frac{n_k}{n} \quad (30)$$

456 this defines a probability distribution, which makes the clustering a random variable.
 457 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 5.4 with respect to clusterings. For example, the entropy of a clustering is

$$H(\mathcal{C}) = - \sum_{k=1}^K \frac{n_k}{n} \log \frac{n_k}{n} \quad (31)$$

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

$$P(K = k, K' = k') = \frac{|C_k \cap C'_{k'}|}{n} \quad (32)$$

458 The joint distribution allows us to define the mutual information between two clus-
 459 terings, $I(\mathcal{C}; \mathcal{C}')$ [10].

460 **5.6.3 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 [11]

$$NMI_{joint} = \frac{I(\mathcal{C}; \mathcal{C}')}{H(\mathcal{C}, \mathcal{C}')} \quad (33)$$

$$NMI_{max} = \frac{I(\mathcal{C}; \mathcal{C}')}{\max\{H(\mathcal{C}), H(\mathcal{C}')\}} \quad (34)$$

$$NMI_{sum} = \frac{2I(\mathcal{C}; \mathcal{C}')}{H(\mathcal{C}) + H(\mathcal{C}')} \quad (35)$$

$$NMI_{sqr} = \frac{I(\mathcal{C}; \mathcal{C}')}{\sqrt{H(\mathcal{C})H(\mathcal{C}')}} \quad (36)$$

$$NMI_{min} = \frac{I(\mathcal{C}; \mathcal{C}')}{\min\{H(\mathcal{C}), H(\mathcal{C}')\}} \quad (37)$$

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

$$AMI_{sum} = \frac{I(\mathcal{C}; \mathcal{C}') - E\{I(\mathcal{C}; \mathcal{C}')\}}{\frac{1}{2}[H(\mathcal{C}) + H(\mathcal{C}')] - E\{I(\mathcal{C}; \mathcal{C}')\}} \quad (38)$$

461 The first term in the demoniator, taking the average of the marginal entropies, can be
462 replaced by taking the maximum, minimum, or the geometric mean [11].

463 5.6.4 Information based metrics

The variation of information between two clusterings $VI(\mathcal{C}; \mathcal{C}')$ (see section 5.4.4) is a metric on the space of clusterings [10]. That is,

$$VI(\mathcal{C}; \mathcal{C}') \geq 0 \quad (39)$$

$$VI(\mathcal{C}; \mathcal{C}') = 0 \iff \mathcal{C} = \mathcal{C}' \quad (40)$$

$$VI(\mathcal{C}; \mathcal{C}') = VI(\mathcal{C}'; \mathcal{C}) \quad (41)$$

$$VI(\mathcal{C}; \mathcal{C}'') \leq VI(\mathcal{C}; \mathcal{C}') + VI(\mathcal{C}'; \mathcal{C}'') \quad (42)$$

Another metric is the *information distance* [11]

$$D_{max} = \max\{H(\mathcal{C}), H(\mathcal{C}')\} - I(\mathcal{C}; \mathcal{C}') \quad (43)$$

Both of these can be normalised

$$NVI(\mathcal{C}; \mathcal{C}') = 1 - \frac{I(\mathcal{C}; \mathcal{C}')}{H(\mathcal{C}, \mathcal{C}')} \quad (44)$$

$$d_{max} = 1 - \frac{I(\mathcal{C}; \mathcal{C}')}{\max\{H(\mathcal{C}), H(\mathcal{C}')\}} \quad (45)$$

5.6.5 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.

References

- [1] Marlene R Cohen, John H R Maunsell, *Attention improves performance primarily by reducing interneuronal correlations*. Nature Neuroscience 12, 1594–1600, (2009)
- [2] Marlene R Cohen, Adam Kohn, *Measuring and interpreting neuronal correlations*. Nature Neuroscience 14, 811-819, (2011)
- [3] Mark D. Humphries, Javier A. Caballero, Mat Evans, Silvia Maggi, Abhinav Singh, *Spectral rejection for testing hypotheses of structure in networks*. arXiv:1901.04747, (2019)
- [4] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, Kenneth D. Harris, *Spontaneous behaviors drive multidimensional, brainwide activity*. Science 364, (2019)
- [5] C. E. Shannon, *A Mathematical Theory of Communication*, The Bell System Technical Journal, (1948)
- [6] J. J. Jun et al., *Fully integrated silicon probes for high-density recording of neural activity*. Nature 551, 232–236, (2017)
- [7] Treves, A., Panzeri, S., *The upward bias in measures of information derived from limited data samples*, Neural Computation 7, 399–407, (1995)
- [8] Bailey K. Fosdick, Daniel B. Larremore, Joel Nishimura, Johan Ugander, *Configuring random graph models with fixed degree sequences*. arXiv:1608.00607 (2017)
- [9] Mark D. Humphries, *Spike-train communities: finding groups of similar spike trains*. Journal of Neuroscience 31, 2321–2336, (2011)
- [10] Marina Meila, *Comparing clusterings — an information based distance*. Journal of Multivariate Analysis 98, 873 - 895, (2007)
- [11] Nguyen Xuan Vinh, Julien Epps, James Bailey, *Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance*. Journal of Machine Learning Research 11, 2837-2854, (2010)

- 507 [12] Hui Zou, Trevor Hastie, *Regularization and variable selection via the elastic*
508 *net*. Journal of the Royal Statistical Society: Series B (Statistical Methodology)
509 67, 301-320, (2005)
- 510 [13] Pierre-André G. Maugis, *Event conditional correlation: Or how non-linear*
511 *linear dependance can be*. arXiv:1401.1130, (2014)