CELL ASSEMBLY DETECTION IN LOW FIRING-RATE SPIKE TRAIN DATA

Phan Minh Duc Truong
truong@smu.edu

Follow this and additional works at: https://scholar.smu.edu/hum_sci_mathematics_etds

Part of the Applied Statistics Commons, Data Science Commons, and the Other Applied Mathematics Commons

Recommended Citation
https://scholar.smu.edu/hum_sci_mathematics_etds/8

This Thesis is brought to you for free and open access by the Mathematics at SMU Scholar. It has been accepted for inclusion in Mathematics Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.
CELL ASSEMBLY DETECTION IN LOW FIRING-RATE SPIKE TRAIN DATA

Approved by:

Dr. Andrea Barreiro
Associate Professor of Mathematics

Dr. Alejandro Aceves
Professor of Mathematics

Dr. Kathryn Hedrick
Assistant Professor of Mathematics

Dr. Wei Xu
Assistant Professor of Neuroscience
CELL ASSEMBLY DETECTION IN LOW FIRING-RATE SPIKE TRAIN DATA

A Dissertation Presented to the Graduate Faculty of the

Dedman College

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Computational and Applied Mathematics

by

Duc P. Truong

B.A., Economics, California State University
M.A., Economics, Cornell University
M.A., Applied Mathematics, California State University Fullerton

August 4, 2020
Cell assemblies, defined as groups of neurons forming temporal spike coordination, are thought to be fundamental units supporting major cognitive functions. However, detecting cell assemblies is challenging since they can occur at a range of time scales and with a range of precisions, from synchronous spikes to co-variations in firing rate. In this dissertation, we use a recently published cell assembly detection (CAD) algorithm that is capable of detecting assemblies at a range of time scales and precisions [48]. We first showed that the CAD method can be applied to sparser spike train data than what have previously been reported. This allows us to apply the method to calcium imaging data of neuronal activity in the CA1 region of the hippocampus, a brain region critical for encoding and generalizing contextual memories, during contextual fear conditioning training and tests. We found that CA1 hippocampus plays a role in encoding and retrieving contextual memories. In particular, there exists a group of neurons whose exploratory activities predict the animal’s ability to distinguish different contexts. Moreover, the mechanisms for processing contextual information are different between two genetically distinct strains of mice that are included in the experiments. Lastly, as inspired by experimental findings, we extend the CAD method to extract multiscale assemblies whose activities happen at different time scales.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>xiii</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1. Review of Cell Assembly Detection Methods</td>
<td>3</td>
</tr>
<tr>
<td>1.1.1. Linear Decomposition Based Methods</td>
<td>3</td>
</tr>
<tr>
<td>1.1.2. Convolutional Methods for Temporal Patterns</td>
<td>5</td>
</tr>
<tr>
<td>1.1.3. Statistical Pairwise Correlation Based Methods</td>
<td>6</td>
</tr>
<tr>
<td>1.2. Overview of dissertation</td>
<td>7</td>
</tr>
<tr>
<td>2. Extending the Range of Validity for a Cell Assembly Detection Algorithm</td>
<td>9</td>
</tr>
<tr>
<td>2.1. Introduction</td>
<td>9</td>
</tr>
<tr>
<td>2.2. Construction of the Test Statistic for Pairwise Correlation</td>
<td>11</td>
</tr>
<tr>
<td>2.2.1. The lower bound on the expected joint spike count</td>
<td>12</td>
</tr>
<tr>
<td>2.2.2. Adjustment for non-stationarity</td>
<td>13</td>
</tr>
<tr>
<td>2.2.3. The test statistic (Q_l)</td>
<td>14</td>
</tr>
<tr>
<td>2.2.4. Multiple testing correction</td>
<td>15</td>
</tr>
<tr>
<td>2.3. Distribution of Test Statistic (Q_l)</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1. Simulating the distribution of the test statistic (Q_l)</td>
<td>15</td>
</tr>
<tr>
<td>2.3.2. The distribution of (Q_l) at different values of (E[#_{AB,l}])</td>
<td>16</td>
</tr>
<tr>
<td>2.4. Quantifying the Goodness of Fit using Power Law</td>
<td>19</td>
</tr>
<tr>
<td>2.4.1. Power law for continuous distributions</td>
<td>20</td>
</tr>
<tr>
<td>2.4.2. Fitting a power law to the tail of (Q_l) and F distribution</td>
<td>20</td>
</tr>
</tbody>
</table>
2.4.3. Error analysis across multiple scenarios ........................................... 25
2.4.4. Adjustment for degree of freedom ....................................................... 26
2.4.5. Summary ............................................................................................ 29
2.5. Comparison of original and extended methods on calcium imaging data .... 31
2.6. Discussion ............................................................................................ 33
2.7. Appendix ............................................................................................... 37
   2.7.1. Hypergeometric distribution of joint binary subprocesses .............. 37
   2.7.2. Hypothesis testing ........................................................................... 37
3. Application to Fear Conditioning Data ....................................................... 42
   3.1. Introduction ......................................................................................... 42
   3.2. Fear Conditioning Experiment ........................................................... 44
      3.2.1. Experiment procedure ................................................................. 44
      3.2.2. Data collection ........................................................................... 46
   3.3. Single Cell Analysis ........................................................................... 46
      3.3.1. Neuron classification ................................................................. 46
      3.3.2. Which neurons encode contextual information? ....................... 50
      3.3.3. More correlations from fully recorded animals ......................... 51
   3.4. Pair Assembly Analysis for Separate Epochs ...................................... 55
      3.4.1. Non-synchronous pair ratios and freezing behaviors ............... 57
      3.4.2. Decomposition of bin-lag histogram ......................................... 59
      3.4.3. Pair assembly activities and freezing episodes ....................... 63
   3.5. Pair Assembly Analysis for Combined Epochs .................................... 64
      3.5.1. Synchronous versus nonsynchronous pairs .............................. 66
      3.5.2. Cell type containing pair assemblies ......................................... 70
      3.5.3. More specific classification for pair assemblies ....................... 74
   3.6. Assembly Analysis ............................................................................ 77
3.6.1. Detecting cell assemblies .................................................. 77
3.6.2. DR Assembly Enrichment in B6C Strain ................................. 80
3.6.3. Correlations between Firing Rate and Freezing Behaviors ........... 80
3.6.4. Spatial coherency of cell assemblies ..................................... 82
3.7. Summary Results and Discussion ............................................ 82
  3.7.1. DR cell activities are associated with contextual memories ........ 83
  3.7.2. Synchronous vs non-synchronous pairs ................................. 84
  3.7.3. Assembly spatial coherency .............................................. 85
3.8. Appendix ................................................................. 86
  3.8.1. Neuron classification algorithm ........................................... 86
4. Multiple Time Scale Assemblies Detection ................................. 90
  4.1. Introduction ........................................................................ 90
  4.2. New Pruning Step ................................................................ 91
    4.2.1. Identifying multiple time scale assemblies ......................... 92
    4.2.2. Collecting assemblies activities ...................................... 93
  4.3. Synthetic Spike Train with Multiscale Assemblies ..................... 93
    4.3.1. Generating Spike Train Population .................................... 93
    4.3.2. Embedding Spiking Pattern ........................................... 94
  4.4. Preliminary Result on Synthetic Data .................................... 94
  4.5. Discussion ......................................................................... 97
5. Conclusion ............................................................................. 98
  5.1. Summary ........................................................................... 98
  5.2. Future Direction ............................................................... 99
BIBLIOGRAPHY ........................................................................ 101
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Calcium imaging data example - pair distribution over expected joint spike count</td>
</tr>
<tr>
<td>2.2</td>
<td>Binary process construction</td>
</tr>
<tr>
<td>2.3</td>
<td>PP-plots of simulated data using suggested parameters</td>
</tr>
<tr>
<td>2.4</td>
<td>PP-plots of simulated data with different firing rates</td>
</tr>
<tr>
<td>2.5</td>
<td>Power law behavior of $Q_l$ and F distribution’s tails</td>
</tr>
<tr>
<td>2.6</td>
<td>Fitting power law to F distributions</td>
</tr>
<tr>
<td>2.7</td>
<td>Relationship of ExpA and ExpAB</td>
</tr>
<tr>
<td>2.8</td>
<td>Fitted parameters of power law of $Q_l$</td>
</tr>
<tr>
<td>2.9</td>
<td>Estimated error predict observed error</td>
</tr>
<tr>
<td>2.10</td>
<td>Estimated errors vs observed errors across multiple scenarios</td>
</tr>
<tr>
<td>2.11</td>
<td>Power law plots from a stationary independent simulation</td>
</tr>
<tr>
<td>2.12</td>
<td>Optimal denominator degree of freedom $\hat{d}_2$ for F distribution</td>
</tr>
<tr>
<td>2.13</td>
<td>Regression of $\hat{d}_2$</td>
</tr>
<tr>
<td>2.14</td>
<td>Improved error from adjusting degree of freedom for F distribution</td>
</tr>
<tr>
<td>2.15</td>
<td>Number of pais comparison between original and extended methods</td>
</tr>
<tr>
<td>2.16</td>
<td>Compare bin-lag histogram of pair assemblies</td>
</tr>
<tr>
<td>2.17</td>
<td>Method comparison for assembly detection</td>
</tr>
<tr>
<td>2.18</td>
<td>Assembly raster plot</td>
</tr>
<tr>
<td>2.19</td>
<td>The pdf of t distribution at different degree of freedom $\nu$</td>
</tr>
<tr>
<td>2.20</td>
<td>One-tail and two-tail hyphothesis testing</td>
</tr>
</tbody>
</table>
3.26 Regression of combined epoch pair ratios................................. 67
3.27 Firing rate of combined epoch pair assembly groups........................ 68
3.28 Regression of non-synchronous pairs activities ............................ 69
3.29 Observed and approximated pair ratio ........................................ 71
3.30 Pair ratios by cell types ............................................................. 72
3.31 DR containing pair assembly activities correlation ....................... 72
3.32 R containing pair assembly activities correlation ........................... 73
3.33 Labeled pair ratio ................................................................. 74
3.34 Labeled pair ratio p-value ........................................................ 75
3.35 R-DR firing rate correlations in B6J ............................................. 76
3.36 DR-{C,S} firing rate correlation in B6J .......................................... 76
3.37 Assembly size distribution .......................................................... 78
3.38 Observed and approximated cell containing assembly ratio ............ 80
3.39 DR containing assemblies firing rate correlation ............................. 81
3.40 Assembly spatial coherency ........................................................ 83
3.41 Significant event response .......................................................... 88
3.42 Histogram of total spikes in 10s before- to after-shock from Xu Lab Animals. .......................................................... 88
3.43 Comparison of simulated and observed threshold ........................... 89
4.1 Identify multiple time scale assemblies ............................................. 92
4.2 Synthetic spike train with multiscale assemblies .............................. 95
4.3 Performance comparison of pruning steps ....................................... 96
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Mean p-value of cell type containing pair ratio</td>
<td>70</td>
</tr>
<tr>
<td>3.2</td>
<td>Number of detected assemblies before and after pruning</td>
<td>79</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor, Professor Andrea Barreiro, for all of her valuable teaching, support, encouragement, and inspiration. Thanks for patently guiding me through the research, for making me a better researcher, and for introducing me to such a fascinating field.

I would like to thank Professor Wei Xu for his guidance and support through the fear conditioning project from which I have learned a lot about neuroscience. I also wish to thank Jun Guo for being a good collaborator and a good friend.

I would like to thank Professor Daniel Reynolds for being a role model to me as a passionate and dedicated teacher. I would like to thank people in SMU Math department for their teaching, support and interesting conversations, including Alejandro Aceves, Kathryn Hedrick, Johannes Tausch, Benno Rumpf, Thomas Hagstrom, Yunkai Zhou, Weihua Geng, Amnon Meir, Brandy Stigler and Liz Berry.

I am grateful to my family for their understanding and support, and to friends who have been with me through my Ph.D. time. Finally, I am grateful to my better half Thao Nguyen for her endless patience and unconditional love, and my son Oliver for keeping me responsible and mindful.
Chapter 1
Introduction

The goal of studying the brain is to map the activities of billions of neurons to behaviors. For a long time, neuroscientists have struggled to figure out what those neurons are doing while animals, including humans, are performing different tasks. Those tasks can be either observable such as sleeping, reaching their arms, or finding their resources; or more intangible such as forming and retrieving memories, making decisions or navigating their paths. Even though there are many challenging opening questions, most agree that many complex cognitive processes cannot be performed by single cells separately, but rather by a dynamic collaboration between a group of neurons. This concept was first proposed by Donald Hebb in 1949 as a cell assembly [17]. By his definition, a cell assembly is a network of neurons being activated repeatedly under a certain mental process. This concept has played a major role in neuroscience ever since.

Recent advanced recording techniques such as multi-electrode or calcium imaging allow us to keep track of the activities of a large population of neurons. These high dimensional spike train datasets provide insight into the brain in the form of the statistical structure of neuronal population activity. More particularly, these datasets have opened up new opportunities to study the role of neuronal spatio-temporal coordination, and in fact different types of neural activity organizations into cell assemblies have been observed from experiments. First of all, various forms of spike correlation, a proxy for cell assembly, at different levels of precision have been observed in different areas of the brain: visual [52], auditory [50], somatosensory [54], and frontal [1] cortices. These assemblies have also been observed in zebrafish optic tectum [46, 55]. Moreover, other works have shown that cell assemblies play a crucial role in neural computation [41, 49]. Formation of cell assemblies involves specific interactions between
interneurons and pyramidal cells is also found in [4]. Furthermore, some experimental studies have suggested that cortical spiking activity may harbor motifs with more complex structure [20,61].

Cell assemblies are also found in hippocampus, an area known to be crucial for learning and memory. For example, a brain structure in hippocampus is shown to be strongly implicated in facilitating long term memory [32,51]. Another finding is that during sleep, neural patterns of activity appearing in the immediately previous awake experience are replayed [21,24,32,60]. Besides, cell assembly replays can happen at different time scales, or can be reversed under certain conditions [10,12,38]. This observation further complicates the task of detecting cell assemblies in spike train population. Overall, these findings motivate the need to associate disparate assemblies into a cognitive unit.

Given that the role of cell assemblies has been emphasized in many experimental findings, and with the availability of high dimensional neuronal spike train datasets, there is a need for data analysis tools to extract cell assemblies information from the neuronal activities. One approach is to look at spike train data and identify groups of neurons that form co-activity (or co-firing) patterns. We realize that this is only a necessary condition for a group of neurons to be an assembly. Beside forming co-firing patterns, some scientists argue that participating neurons in an assembly need to have strong internal synaptic connections [13]. For clarity, here we only use the term assembly as a group of neurons with co-activity patterns. One might think that these groups of neurons are qualified candidates for being a true cell assembly.

However, studying cell assemblies is challenging for several reasons. First, there is no universally accepted definition; mathematical formulations include different type of patterns, from synchronized spiking to temporal rate changes. Moreover, patterns could happen at an arbitrary time lag and temporal precision, and can also be compressed to a finer time resolution [38] or be replayed in reverse [10,12]. All of these factors make it hard to develop a tool which could detect all type of assemblies. The recent cell assembly detection (CAD) method of Russo and Durstewitz [48] addresses most of these issues, and is capable of
detecting cell assemblies at multiple time scales with arbitrary lag. However, to maintain the validity of the method, the expected co-firing rate between neurons needs to pass a certain threshold. This is not always true, especially for low sampling rate data such as calcium imaging data. Therefore, in chapter 2, we address this limitation and extend the range of validity of the cell assembly detection method for it to be more applicable to low firing rate spike train data. As a real-world application in chapter 3, the extended method is then applied to calcium imaging data from Xu lab’s fear conditioning experiments and used to gain insight into the memory encoding and retrieving process. Lastly, in chapter 4, we propose a new pruning step to extend the capability of the method to detect cell assemblies whose activities happen at different time scales.

In the next section, I will give a review of some cell assembly detection methods, just for the purpose of comparing with Russo’s method. The last section will be an overview of the main results in this dissertation.

1.1. Review of Cell Assembly Detection Methods

1.1.1. Linear Decomposition Based Methods

Extracting assemblies from spike count data can be formed as a clustering problem in which strongly correlated neurons, defined by a correlation measure, are clustered together. One common approach to solve clustering problems is to decompose the correlation matrix of the spike counts using principal components analysis (PCA) [35,36]. Notice that in order to correctly implement PCA, the spike count of each neuron needs to be normalized by z-score transformation; in this sense PCA is equivalently the eigendecomposition of the covariance matrix. The uncorrelated eigenvectors (or principal components) are the directions of the highest variances, which are measured by the eigenvalues. The elements in a principal component represents the weights of those elements in that component.

As shown in [26], in some cases, PCA might extract components which mix patterns of different assemblies. To avoid this, independent component analysis (ICA) can be applied.
Basically, ICA rotates components to get a new set of vectors which are independent; in the sense that they share the least amount of mutual information. Equivalently, we can think of ICA as a method to extract statistically independent non-Gaussian components from a neuronal population signal [23]. Another way to improve the quality of principal components is assembly vector (AV), which is also introduced in [26].

1.1.1.1. Determine the number of assemblies

In order to determine the number of assemblies, the idea is to find the eigenvalues which are statistically large. There are two approaches to address this. The first approach is to use the Marchenko-Pastur distribution assuming the normalized covariance matrix is a normal random matrix. The second approach is to shuffle the matrix many times to generate its eigenvalues distribution.

**Marchenko-Pastur distribution approach**  Eigenvalues extracted from PCA measure the strength of correlation between neurons in that component. Therefore, one way to determine the number of assemblies is to count eigenvalues that are significantly large, given a distribution of eigenvalues.

The eigenvalues of the correlation matrix of a normal random matrix with statistically independent rows are shown to follow Marchenko-Pastur (MP) distribution [31]

\[
p(\lambda) = \frac{q}{2\pi\sigma^2} \frac{\sqrt{(\lambda_{\text{max}} - \lambda)(\lambda - \lambda_{\text{min}})}}{\lambda}
\]

where \( q = N_{\text{column}}/N_{\text{row}} \), \( \sigma = 1 \) because the data is z-score transformed, and \( \lambda_{\text{max}} = \sigma^2(1 \pm \sqrt{(1/q)})^2 \)

Under the assumptions that (1) the matrix is large, (2) the matrix rows are statistically independent, (3) matrix entries are derived from Gaussian distributions, the probability of finding eigenvalues outside the interval \([\lambda_{\text{min}}, \lambda_{\text{max}}]\) is zero. Therefore, any component whose eigenvalue is greater than \( \lambda_{\text{max}} \) can be considered an assembly. Moreover, \( \lambda_{\text{max}} \) has been shown to be a robust threshold for smaller matrices [26, 40].
**Shuffling approach**  One issue with using MP distribution to identify the assembly is that the data from correlation matrix usually is not drawn from Gaussian distribution, thus the theoretical threshold $\lambda_{max}$ might not be appropriate. Instead of being approximated by Machenko-Pastur distribution, the eigenvalue distribution can be generated from shuffling spike count data. There are two options to shuffle the data: shuffle the bins of each neuron randomly, or circular-shift the spike counts. As shown in [27], under their experimental settings, the eigenvalue distribution generated by circular-shifted spike counts matches closely with the Marchenko-Pastur distribution. However, since experimental spike counts might have different statistical properties, both distributions should be considered in the analysis. Moreover, circular-shifting data might preserve more statistical dependencies in the original data.

1.1.1.2. *Estimating assembly activities*

After identifying assemblies from the principal components, the assemblies activities, or activated time, can be computed by projecting the neuronal population spike count onto the components. This will show the signal strength of the assembly in each time bin.

1.1.1.3. *Limitation*

The first limitation of these methods is that they analyze the correlation matrix of spike counts, which only captures the synchronous co-activities. Even though zero-lag coactivation has been shown to provide rich information about the neuronal network [33, 45], and that zero-lag coactivation is an easier target for statistical pattern recognition methods, the methods certainly are not able to detect assemblies with more complicated activity patterns which are observed in experimental data.

The second limitation of these methods is that they do not address the nonstationarity which is common in neuroscience spike train data. This could lead to falsely identify cell assemblies.
1.1.2. Convolutional Methods for Temporal Patterns

As an extension of linear methods to detect assemblies with more arbitrary spiking patterns, a new algorithm based on sparse convolutional nonnegative matrix factorization (NMF) has been proposed by [34]. In this method, the spiking pattern is represented by a matrix, instead of a vector as in PCA-based methods. This will allow to capture more complex spiking patterns which happens over a window of multiple time bins. The spiking pattern matrix is then convoluted with the activity vector to reconstruct the original data. Since the method is a convolutional nonnegative matrix factorization with $l_0$ and $l_1$ regularization, its factors are sensitive to the number of latent factors being chosen for the model. Unlike PCA, the set of latent factors for each dimension number is totally different. Therefore, without taking this into account when setting the latent dimension, the extracted factors might give a wrong information about the assemblies. This is one limitation of the method, and further extension is needed to improve its reliability. Another similar method is [30] which is also based on convolutional NMF.

1.1.3. Statistical Pairwise Correlation Based Methods

Another approach to detecting assemblies is to start from finding pairs of neurons whose activities are significantly correlated. Then larger assemblies can be formed by merging those significant pairs together. To detect non-synchronous sequential patterns, many methods use statistical models that capture the significant cross-correlation [14, 48, 56, 58]. The CAD method studied here can be thought as a method in this class. However, limitations of these methods is that it might be computationally expensive to detect assemblies with a large number of neurons, and have difficulties to deal with background activities [42]. The output of these methods will be a large set of small-size assemblies, while the previous methods will extract a small number of large-size assemblies.
1.2. Overview of dissertation

Having given a general background about experimental findings of cell assembly replays, including ones observed from multiple timescales, and having review different models to detect cell assemblies from spike train data, now I give an overview of the following chapters of the thesis.

Chapter 2 improves the range of validity for the CAD method to be more applicable to low firing rate data. Originally, the CAD method only assesses the correlation between two spike counts whose expected joint spike count, or the level of co-activities, meets a preset threshold. The authors’ explanation is that below this threshold, the constructed test statistic is no longer well approximated by the theoretical F distribution, which implies that the correlation testing is not accurate in this range. As we tried to apply this method to our calcium imaging data, this threshold becomes problematic. The majority of pairs in our neuronal population data did not meet the threshold since the time resolution of calcium imaging data is low (≈ 10Hz), and neuron activities in hippocampus are known to be sparse. Therefore, in the attempt of utilizing this method for calcium imaging data sets, we assess the difference between the distribution for synthetic data and the theoretical F distribution. We found an extended range of the threshold in which the error is acceptable, in the sense that the p-value of significant correlations are small. Moreover, we also propose a model to approximate the error, the difference between two distributions, as a function of expected joint spike count and the size of neuron population. In conclusion, we claim that the original threshold can be lowered without losing the validity of the CAD method.

Chapter 3 shows the results of applying the CAD method to data from fear conditioning experiments. We show that CA1 hippocampus neuronal activities are associated with contextual memory learning and retrieval. Different correlations between neuronal activities and freezing behaviors are found at different level of analyses: single cell, pair assembly and full assembly analysis. Those correlations also indicate that the mechanism for processing contextual information are different between two genetically different strains of mice. Moreover, the results from pair assembly analysis indicates that synchronous and non-synchronous pairs
are associated with freezing behaviors in different ways. Lastly, analysis of neuron physical location shows that cells participating in an assembly tends to say closer spatially.

Chapter 4 proposes a new pruning process for the CAD method to detect multiscale cell assemblies whose spiking patterns are replayed at different timescales. We only show preliminary result in which the proposed pruning step outperforms the original step on two synthetic datasets with embedded multiscale assemblies. In future work, the method will be tested on more complexed multiscale spiking pattern and applied to real-world datasets.
Chapter 2
Extending the Range of Validity for a Cell Assembly Detection Algorithm

2.1. Introduction

The recent published cell assembly detection (CAD) method of Russo and Durstewitz [48] has addressed many current issues of finding spiking patterns from data, and is capable of detecting cell assemblies at multiple time scales with arbitrary lag. Specifically, the algorithm seeks to find cell pairs that are unusually correlated; i.e. that violate the null hypothesis of independence. To do so, a test statistic for two independent spike trains is constructed. This test statistic is nonstationarity adjusted and well approximated by the F-distribution (Appendix - Theorem 2.3). The nonstationarity correction feature gives this algorithm an advantage when compared with other methods such as cross-correlations [5], PCA [26], ICA [27], or sparse convolutional coding [34].

However, in order to ensure that the test statistic follows the F-distribution sufficiently well, a lower bound is imposed on the quantity called Expected Joint Spike Count, which measures the average co-spike counts (or co-firing rate) between a pair of units, or between a unit and an assembly. Even though this assumption is easily satisfied by high-sampling-rate data, such as data recorded from multi-electrode arrays (≈ 20kHz), low-sampling-rate calcium imaging data (≈ 10Hz) usually does not pass this threshold. For example, Figure 2.1 shows the pair distribution over expected joint spike count of a B6J mouse’s spike train inferred from calcium imaging data of fear conditioning experiments. With the suggested threshold at 5 from the original implementation, a majority of pairs do not pass the threshold. As a consequence, there are many pairs will end up not being tested, and potentially significant assemblies will be lost.
Figure 2.1: Pair distribution over expected joint spike count of a B6J mouse’s spike train data. Using the threshold suggested by the original method, many pairs do not pass the threshold, and will not be tested.

Therefore, as an effort to make this method more applicable to low-sampling-rate spike train data, we first analyze the test reliability of using F-distribution to approximate the test statistic distribution when expected joint spike count is smaller than the threshold. Particularly, we use a power law to quantify the approximation error at the distribution tails. Our results show that threshold can be lowered to 2 without losing the statistical significance. Moreover, we also propose a guideline to achieve a better testing reliability by adjusting the denominator degree of freedom of the F distribution. With this guideline, the threshold can be safely lowered to 1.

The structure of this chapter is as follows. In the section 2.2, we show how to construct the test statistic for a pair of spike counts. In the section 2.3, we simulate the distribution of the test statistic under different stationarity, independence, and firing rate conditions. Section 2.4 is the main section showing how to quantify the error of using F-distribution to approximate the test statistic’s distribution, and also showing the guideline to use the
extended method in the region of small expected joint spike count. The comparison between the original and the extended test statistic on real datasets is in Section 2.5. Section 2.6 is our discussion.

2.2. Construction of the Test Statistic for Pairwise Correlation

The backbone of this algorithm is a test statistic that assesses pairwise correlation for a pair of spike trains. In this section, we will describe how the test statistic is constructed, how it is adjusted for nonstationarity and corrected for multiple testing, and where the lower bound on expected joint spike count is imposed.

A given pair of spike trains are binned using a specified bin width to get a pair of spike counts \( \{A, B\} \). For a range of all considered lags \( l \in \{-l_{max}, ..., l_{max}\} \), the algorithm tests whether the joint spike count \( #_{AB,l} \) of two cells \( A \) and \( B \) at lag \( l \) is significantly greater than what would have been expected under the null hypothesis \( (H_0) \) of independence of the two spike counts. For clarity, the joint spike count \( #_{AB,l} \) is calculated by counting the co-occurrences of spikes in cell \( A \) and in cell \( B \) \( l \) bins later. Since the expected spike counts largely do not depend on lag, the highest joint spike count is the most surprising one. Therefore, only the highest joint spike count \( #_{AB,l} = \max_l #_{AB,l} \) is tested.

After that, each spike count \( c \in \{A, B\} \) is split into a set of \( m = 1..M_c \) binary subprocess \( #_{m,c,l} \), where \( M_c = \max(c) \), as shown in Figure 2.2. The \( m \) binary process is defined to be 1 for all time bins in which the spike count is greater than or equal to \( m \). Then for two cells \( A \) and \( B \), the total joint count \( #_{AB,l} \) is given by:

\[
#_{AB,l} = \sum_{m=1}^{M} #_{m,AB,l} \quad \text{where} \quad M = \min(M_A, M_B)
\]

Notice that since each of the \( M \) subprocesses is binary, \( #_{m,AB,l} \) follows a hypergeometric distribution under the null hypothesis \( H_0 \) of stationary independent activity (see Appendix 2.7.1 for further explanation). Also, \( M \) binary processes are not independent.
Figure 2.2: The spike count in this example is split into a set of three binary subprocesses, since the maximum spike counts in a bin is three. The $m^{th}$ binary process is defined as 1 in all bins for which the spike count is greater than or equal to $m$. - Adapted from [48]

Therefore, the expected joint spike count and variance of $\#_{AB,l}$ under $H_0$ can be computed using the following formulas:

$$\mu_{AB,l} = E[\#_{AB,l}] = \sum_{m=1}^{M} \frac{\#_A \#_B^{m}}{T - l}$$

(2.1)

$$\sigma^2_{AB,l} = E[(\#_{AB,l} - E[\#_{AB,l}])^2] = \sum_{m=1}^{M} \frac{\#_A^{m} \#_B^{m} (\tilde{T} - \#_A^{m})(\tilde{T} - \#_B^{m})}{T^2(T - 1)} +$$

$$2 \sum_{m=1}^{M-1} \sum_{\gamma > m}^{M} \frac{\#_A^{\gamma} \#_B^{\gamma} (\tilde{T} - \#_A^{m})(\tilde{T} - \#_B^{m})}{T^2(T - 1)} \text{ with } \tilde{T} = T - l$$

(2.2)

2.2.1. The lower bound on the expected joint spike count

As mentioned above, to ensure that the test statistic follows the F-distribution sufficiently well, a lower bound is imposed on the expected joint spike count, i.e. $E[\#_{AB,l}] \geq 5$. If this constraint is not satisfied, the cell pair will no longer be considered. However, as we will show later, low sampling rate spike trains such as ones from calcium imaging recording hardly meet this constraint, which causes many pairs to be discarded. Therefore, relaxing this constraint will make the algorithm more applicable for such datasets. On the other
hand, if the constraint on $E[\#_{AB,i}]$ is met, the algorithm will move to the next step for nonstationarity correction.

2.2.2. Adjustment for non-stationarity

The approximation of the variance in (2.2) strongly depends on an assumption of stationarity: that is, that the statistical characteristics of the two spike trains are unchanging in time. This is not usually the case for neuroscience data. During most experiments, both internal and external stimuli are constantly presented to the brain which result in firing rate fluctuations on different time scales. For example, during the fear conditioning experiment we discuss in Chapter 3, animals received audio and foot-shock stimulation, and were exposed to different visual environments. Hence the spike trains from these experiments are presumably nonstationary. Consequently, the variance in (2.2) will underestimate the true variation of the data [48]. For that reason, a treatment for non-stationarity is necessary.

Generally, two common methods for non-stationary time series are bootstrap-based techniques [39] and sliding window analyses [15]. However, these two methods have some limitations which make them not appropriate to be used in this algorithm. The bootstrap-based techniques are computationally expensive, since for each non-stationarity adjusting step, 100-1000 bootstrap replications need to be repeated, and the whole algorithm usually calls this step tens of thousands times. The sliding window method, on the other hand, uses only small fragments of the data for estimation in each window, which can lead to weak statistical power.

Therefore, in this algorithm [48], a different approach was proposed for the non-stationarity issue, which is defined as

$$\#_{ABBA,\bar{t}} = \#_{AB,\bar{t}} - \#_{AB,\bar{-t}} \tag{2.3}$$

In the case of nonstationarity, such as common rate fluctuation in a pair of independent spike counts, some spike co-occurrences caused by the rate changes might incorrectly contribute to the assembly activity at the finer timescale (bin width). Besides, the nonstationarity has the same effect on both spike counts, and thus on both $\#_{AB,\bar{t}}$ and $\#_{AB,\bar{-t}}$. Therefore,
the adjustment will remove the non-stationarity on slower time scale (≥ \( \bar{l} \Delta \)), where \( \Delta \) is the considered bin width). For the strictly synchronous case (\( \bar{l} = 0 \)), \( \#_{ABBA,i} = \#_{AB,i} - \#_{AB,i}^* \)
where \( l^* \neq 0 \) (\( l^* \) is set to 2 in the algorithm).

Under the null hypothesis, we have

\[
\mu_{ABBA,i} \equiv E[\#_{ABBA,i}] = E[\#_{AB,i}] - E[\#_{AB,-i}] = 0 \\
\sigma^2_{ABBA,i} \equiv E[(\#_{ABBA,i} - E[\#_{ABBA,i}])^2] = 2\sigma^2_{AB,i} - 2\text{cov}(\#_{AB,i}, \#_{AB,-i})
\] (2.4)

\[\] (2.5)

2.2.2.1. Stationarity segmentation for stronger nonstationarity

Moreover, as described in the original method, if the process is expected to be strongly nonstationary, another remedy similar to the idea in [43], can be applied to get a better approximation of the variance. First, the spike counts are divided into \( C \) segments of \( k \) time bins. The local, segment-wise variance estimates are then combined to approximate global variance \( \hat{\sigma}^2_{ABBA,i} \) in Eq. (2.5) with

\[
\hat{\sigma}^2_{AB,i} = \text{var}(\#_{AB,i}) = \sum_{c=1}^{C} \text{var}(\#_{cAB,i}) + 2\sum_{c=1}^{C-1} \sum_{\zeta>c} \text{cov}(\#_{cAB,i}, \#_{\zeta AB,i})
\] (2.6)

\[
\text{cov}(\#_{AB,i}, \#_{AB,-i}) = \sum_{c=1}^{C} \text{cov}(\#_{cAB,i}, \#_{cAB,-i}) + 2\sum_{c=1}^{C-1} \sum_{\zeta>c} \text{cov}(\#_{cAB,i}, \#_{\zeta AB,-i})
\] (2.7)

\[
\text{cov}(\#_{cAB,i}, \#_{cAB,-i}) = \sum_{m=1}^{M} \frac{\#_{cA}^m \#_{cB}^m (k - \#_{cA}^m)(k - \#_{cB}^m)}{k^2(k-1)} + \\
2\sum_{m=1}^{M-1} \sum_{\gamma>m} \frac{\#_{cA}^\gamma \#_{cB}^\gamma (k - \#_{cA}^\gamma)(k - \#_{cB}^\gamma)}{k^2(k-1)}
\] (2.8)

2.2.3. The test statistic \( Q_{\bar{l}} \)

After having all parameters under the null hypothesis \( H_0 \), the test statistic is constructed for the hypothesis testing (See Appendix 2.7.2 for a brief explanation of hypothesis testing).
The approximately F-distributed test statistic is defined:

\[ Q_{\bar{l}} = \left( \frac{\#_{ABBA,\bar{l}} - \mu_{ABBA,\bar{l}}}{\hat{\sigma}^2_{ABBA,\bar{l}}} \right)^2 \sim F_{1,v} \]

where \( v \) is set to be \( 2(T - \bar{l})M - 1 \) for large sample size, or to be \( T - \bar{l} \) for small sample size \( (T < 50) \). The difference between using two values of \( v \) is neglectable when \( T > 400 \) [48].

2.2.4. Multiple testing correction

Finally, since multiple tests are performed, the chance of having significant assemblies increases, which leads to incorrectly reject the null hypotheses. To compensate for this increase, all the \( \alpha \)-levels are Bonferroni-corrected for the number of tested pairs, which means that the larger the population size, the smaller the corrected \( \alpha \). For example, if we run the test for a population of 100 cells, or equivalently 4,950 pairs, and the Type-I error \( \alpha \) is set to 5\%, the Bonferroni-corrected \( \alpha \) (\( \alpha_{bc} \)) will be

\[ \alpha_{bc} = \frac{\alpha}{\text{number of pairs} \times \text{number of tested lag}} = \frac{0.05}{4,950 \times 21} \approx 1e^{-6} \] (2.9)

A less stringent alternative for Bonferroni correction is Holm method [18], which is too computationally expensive and memory consuming to implement in this algorithm.

2.3. Distribution of Test Statistic \( Q_{\bar{l}} \)

In this section we construct the distribution of the test statistic \( Q_{\bar{l}} \). We do this by simulating a large number of spike train pairs, and computing their \( Q_{\bar{l}} \). The process of simulating a pair of spike train is similarly done in [48]. Different conditions of stationarity and independence, and different values of \( E[\#_{AB,\bar{l}}] \) are considered. The simulated distributions of \( Q_{\bar{l}} \) are then compared with corresponding F-distribution using the probability-probability plot (hereafter, pp-plot).

2.3.1. Simulating the distribution of the test statistic \( Q_{\bar{l}} \)

A pair of independent spike trains was drawn from a Bernoulli process with probabilities \( \pi_{\text{low}}(\pi_{\bar{l}}) \) (these are called low-rate states). In order to add nonstationary, we mixed \( m \)
high-rate states into the spike trains (Bernoulli process with $\pi_{\text{high}}(\pi_h)$). The higher value of $m$ will create higher level of nonstationarity. Moreover, we also have scenarios in which we generated coherent pairs where the high-rate states of two spike-trains are synchronously embedded.

A pair of spike counts is generated as follows:

- The length of spike-trains is $T = 10^4$ (number of elementary time bins).
- $m$ high-rate states of duration $L$ are mixed with low-rate states.
- High-rate states are synchronously or independently generated between two spike count time series.
- For each elementary time bin, spikes were drawn from a Bernoulli process with probabilities $\pi_{\text{low}}(\pi_l)$ and $\pi_{\text{high}}(\pi_h) = 5\pi_l$.

Notice that by construction, $E[\#_{AB,l}]$ is a function of $T$, $m$, $L$, $\pi_l$ and $\pi_h$. Therefore, by changing these variables, we can set the desired value for $E[\#_{AB,l}]$.

For each scenario, given a set of parameters $(m, L, \pi_l, \pi_h)$, $2 \times 10^7$ pairs of spike-counts were simulated and their test statistics $Q_l$ were computed, and thus generating a distribution. Other parameters used to construct the test statistic were the value of lag $l = 5$ and the length of segment $k = 100$ bins.

2.3.2. The distribution of $Q_l$ at different values of $E[\#_{AB,l}]$

Here we compare the distribution of $Q_l$ with the F-distribution using pp-plot. The goodness of fit between $Q_l$ and F-distribution is equivalently shown as how well the pp-plot follows the 45-degree line.

We first run the simulations using parameters suggested by [48]. Figure 2.3 shows that the $Q_l$ distribution matches the corresponding F distribution quite well (another case where there is no high-rate region in the spike trains is also added). However, in these cases, $E[\#_{AB,l}]$ are very large, which is not our range of interest for $E[\#_{AB,l}]$. We want to analyze the goodness of fit of the F distribution when $E[\#_{AB,l}]$ is at most 5. Therefore, the firing
rates $\pi_l$ and $\pi_h$ are adjusted to reduce $E[\#_{AB,l}]$. As shown in Figure 2.4, as $E[\#_{AB,l}]$ became smaller, the distribution of $Q_l$ started to deviate from the F distribution. More specifically, the simulated distributions start fluctuating around the F-distribution.

Nonetheless, we speculated that the goodness of fit is still good at the right tail, which is the region where the pairwise correlation is tested. By looking at the tail area of $1e-4$ of simulations with small $E[\#_{AB,l}]$, we can see that the tails of both distributions are approximately described by power-law (Figure 2.5). This observation gives us the idea of using a power law distribution to approximate the tails and to quantify the difference between them.
Figure 2.3: PP-plots of simulated data using suggested parameters from [48]. The approximations are very good but $E[\#_{AB,l}]$ (meanExpAB in the figure) are very large. Right figure ($L=1, m=1$) represents the case of no high-rate region, i.e. there is no nonstationarity.

Figure 2.4: PP-plots of simulated data with different firing rates. The approximation is getting worse when $E[\#_{AB,l}]$ becomes smaller.
2.4. Quantifying the Goodness of Fit using Power Law

As the tails of both $Q_l$ and the F distributions have power-law behavior, as shown in Figure 2.5, we use a power law approximation to assess the goodness of fit between the tail of the empirical $Q_l$ and the corresponding F-distribution. The basic idea is that we will fit a power law to the tail of both the $Q_l$ and F-distribution, and then compare their best-fit models. Because the density function of a power law is so easy to integrate, we can easily construct a formula to estimate the error incurred by using an F distribution instead of the true test statistic $Q_l$ in the independence hypothesis test. We note that this process introduces an additional additional source of error: the approximation error from fitting a power law to the two distributions. We will show that these errors are relatively small and that the difference between two best-fit power laws actually mimics the observed error.
2.4.1. Power law for continuous distributions

Given parameters $x_{\text{min}}$, the lower bound for power-law behavior, and $\theta > 1$, a scaling parameter, the power law distribution for a continuous random variable $x$ is defined as

$$p(x) = \frac{\theta - 1}{x_{\text{min}}} \left( \frac{x}{x_{\text{min}}} \right)^{-\theta} \quad \forall x \geq x_{\text{min}}$$

**Estimating the scaling parameter $\theta$** Assuming $x_{\text{min}}$ is known, we can derive the maximum likelihood estimator (MLE) of the scaling parameter from the data $\{x_i\}_{i=1}^n$ [7].

$$\hat{\theta} = 1 + n \left[ \sum_{i=1}^n \ln \frac{x_i}{x_{\text{min}}} \right]^{-1}$$

**Estimating the lower bound $x_{\text{min}}$** In order to estimate $\theta$, we first need an accurate estimation of $x_{\text{min}}$. We use the approach proposed by [8], which is to set $x_{\text{min}}$ to minimize the distance between the measured data’s probability distribution and the best-fit power-law model. The distance between two distributions is measured using the Kolmogorov-Smirnov (KS) statistic, which is the maximum distance between the cumulative distribution functions (CDFs) of the data and the fitted model:

$$D_{KS}(x_{\text{min}}) = \max_{x \geq x_{\text{min}}} |S(x) - P(x)|$$

where $S(x)$ is the CDF of the data with value at least $x_{\text{min}}$, and $P(x)$ is the CDF of the best-fit power-law model for such data. That is,

$$x_{\text{min}} = \arg \min_{x_{\text{min}}} D_{KS}(x_{\text{min}}^*)$$

2.4.2. Fitting a power law to the tail of $Q_\ell$ and F distribution

For each simulation, we only keep the data whose test statistics are greater than the $x_0$, $(1 - 10^{-4})$ quantile of the corresponding F distribution. We choose to truncate the data at that threshold since it is large enough to cover our testing region of interest, and because the tail of the observed distributions seems to have power law behavior.
Given the truncated data from both distributions, we set the power law distribution parameter $x_{\text{min}} = x_0$ and compute the maximum likelihood estimators for the power law scaling parameters $\theta_Q$ and $\theta_F$.

2.4.2.1. Power law for the F distribution with different denominator degrees of freedom

We consider a range of denominator degree of freedom $d_2 = [50 : 10 : 100, 200 : 50 : 1000]$ for the corresponding $F(1,d_2)$ distribution (see Appendix 2.3). For each value of $d_2$, 100 simulations of the following are implemented:

- A sample $X$ with sample size $10^8$ is drawn from $F(1,d_2)$.
- Data is truncated at $x_0 = \text{CDF}_{F}^{-1}(1 - 10^{-4}, 1, d_2)$,
- A power law is fitted with $x_{\text{min}} = x_0$, and the MLE scaling parameter is computed.

From all simulations for one value of $d_2$, we get a distribution of the scaling parameter. Figure 2.6 shows the mean and one-standard-deviation band of the MLE scaling parameter across denominator degree of freedom $d_2$, and the approximation error at the tail calculated by KS distance. The scaling parameter tends to increase with $d_2$, with a narrow fluctuation. Moreover, the approximation error is fluctuating in the range of $[1 - 3] \times 10^{-6}$, which is about 1-3 percent of the tail area. We conclude that the F-distribution can be accurately approximated by a power law.

2.4.2.2. Defining error measures

Since we want to show when the F distribution can be used in the place of $Q_I$ for the correlation test, we only measure how much larger the tail area of $Q_I$ is in comparison to the tail area of the corresponding F distribution. In the case that $Q_I$ has a smaller tail, the F distribution is still usable without having larger type I error. Therefore, the observed error in tail areas between $Q_I$ and F distributions is defined as

$$Err(Q_I, F) = \max \left\{ \max_{x \geq x_0} \left[ (1 - CDF_{Q_I}(x)) - (1 - CDF_{F}(x)) \right] ; 0 \right\} \quad (2.10)$$
Figure 2.6: Scaling parameters and error of fitting power law to F distribution with different denominator degree of freedom. *(Blue using the left y-axis)*- scaling parameters with one-standard-deviation band. *(Red using the right y-axis)*- Approximation error (and one-standard-deviation band) of using power law to model the F distribution’s tail.

and the estimated error, the error in tail areas between $Q_i$’s power law (PLQ) and F’s power law (PLF) is defined as

$$Err(PLQ, PLF) = \max_{x \geq x_0} \left[ s_Q \left( \frac{x}{x_0} \right)^{1-\theta_Q} - s_F \left( \frac{x}{x_0} \right)^{1-\theta_F} \right]$$

where $s_{Q/F} = 1 - CDF_{Q/F}(x)$ are normalizing factors. This formula comes from integrating the tail of a power law, as

$$\int_x^\infty \frac{\theta - 1}{x_0} \left( \frac{y}{x_0} \right)^{-\theta} dy = \left( \frac{x}{x_0} \right)^{1-\theta}$$
2.4.2.3. Stationary independent scenario

Here we describe the result from one simulation scenario in which all spike trains are stationary and independently generated. In this scenario, there is no high-rate period in the spike trains, and the parameters are \( T = 1e4, \ m = L = 0 \). For a given set of parameter \((T, m, L, \pi_l)\), the expected spike count of a simulated spike train, or \( ExpA \), can be derived as

\[
ExpA = mL\pi_h + (T - ml)\pi_l = mL(5\pi_l) + (T - ml)\pi_l = (5mL + T - mL)\pi_l = (T + 4mL)\pi_l
\]

Figure 2.7: One-to-one relationship between \( ExpA \) vs mean value of \( E[\#_{AB,l}] \) (mean(ExpAB)).

Notice that for each value of \( ExpA \), a simulation will generate a distribution of \( E[\#_{AB,l}] \) (or ExpAB). As shown in Figure 2.7, the correspondence between \( ExpA \) and the mean of \( ExpAB \) (mean(ExpAB)) is one-to-one. Thus we can use these two variables interchangeably. Using \( ExpA \) is more convenient for simulation, but demonstrating results in mean(ExpAB) is more relevant to what we want to show.

For each value of \( ExpA \), multiple simulations are run using the process in section 2.3.1. Each simulation obtains a distribution of \( Q_l \), whose tail is fitted by a power law similar to the above process for F distribution. We choose the same \( x_{min} = CDF_F^{-1}(1 - 10^{-4}, 1, d_2) \), since we want to compare how the tail areas of both distribution are different at a certain
threshold. The tail area can be interpreted as the Type I error for the hypothesis test that we are using. Therefore, in order to compare with the power law model from F distribution, the power law of $Q_\ell$ distribution needs to be rescaled. The normalizing factor for $Q_\ell$ is set to be equal to the proportion (or probability) of $Q_\ell \geq x_{\min}$.

Figure 2.8 shows how the estimated power law scaling parameter and normalizing factor change with $\text{mean}(\text{ExpAB})$. While the scaling parameter seems to converge to the scaling parameter of corresponding F distribution, the normalizing factor does not. However, it does not seem to affect the error approximation. As shown in Figure 2.9, the observed error is well captured by the estimated error, implying that we can model the observed error between the tails of $Q_\ell$ and the F distribution using their best-fit power law approximations. In the region of small $\text{mean}(\text{ExpAB}) \leq 6$, the error starts increasing as the value of $\text{ExpA}$ gets smaller, and more rapidly when $1 \leq \text{mean}(\text{ExpAB}) \leq 2$. Overall, the error is relatively small (at most 2% of the tail area) for $\text{mean}(\text{ExpAB}) \geq 2$, but quite large for $1 \leq \text{mean}(\text{ExpAB}) \leq 2$.

Figure 2.8: Fitted parameters of power law of $Q_\ell$. Scaling parameter and normalizing factor for power law fits to the test statistic $Q_\ell$, for spike trains with independent high-rate periods and parameters $(T, m, L) = (1e - 4, 1, 1)$. Top: As the mean expected joint spike count $\text{mean}(\text{ExpAB})$ increases, the scaling parameter converges to the power law scaling parameter for the corresponding F distribution. Bottom: The normalizing factor seems to stay close to the normalizing factor of the power law from F distribution, but does not converge.
2.4.3. Error analysis across multiple scenarios

We now simulate data with various proportion of high-rate states in the spike train, and in which two spike trains have high-rate periods either independently or synchronously generated. Figure 2.10 shows the observed and estimated errors from multiple scenarios. For example, the name ‘mL10L5’ means that the scenario has independently chosen high-rate periods (‘m’), that 10 percent of the elementary bins are high-rate states (‘L10’), and that each high-rate segment has a length of 5 bins (‘L5’). Similarly, ‘smL10˙L5’ indicates the synchronous version of that scenario, i.e. in which high-rate periods are synchronous between the two spike trains.

The errors in the region of $meanExpAB \leq 6$ behave similarly across scenarios, and start decreasing as $meanExpAB$ gets larger. However, when $1 \leq mean(ExpAB) \leq 2$, the error increases more rapidly, indicating that the test statistic $Q$ is not well approximated by the $F$ distribution in this interval. For clarity, higher error will associate with higher number of
false positive pairs. Thus we suggest setting the new threshold at $ExpAB \geq 2$ (instead of $ExpAB \geq 5$ originally) to ensure the reliability of the correlation test. Furthermore, we will show next that we can also adjust the denominator degree of freedom for the F distribution to have a better goodness of fit, which improves the test reliability. With this adjustment, the threshold can be safely lowered further to 1.

### 2.4.4. Adjustment for degree of freedom

As shown in Figure 2.10, the error of using the F distribution to approximate the test statistic distribution $Q_l$ starts getting larger when $ExpAB$ gets smaller. Since we can approximate the tails of both $F(1, d_2)$ and $Q_l$ distribution using power laws (PLF and PLQ),
we can use those power laws to further investigate the observed error’s behavior.

Figure 2.11: Power law approximation of $Q_l$ and F distribution at $meanExpAB \leq 6$ from stationary independent simulations. The scaling parameters of PLQ is smaller than PLF.

By looking at their plots in log-log scale from different simulations, we notice that when $ExpAB \leq 6$, the scaling parameters of PLQ is smaller than PLF (Figure 2.11 shows an example of the power law plots at different $meanExpAB$ values from one simulation). This equivalently means that the tail of PLQ is heavier than the tail of PLF. From Figure 2.6, we observe that the PLF scaling parameter and the denominator degree of freedom of the F distribution ($d_2$) have a one-to-one relationship. These facts suggest that for a given PLQ, we can adjust $d_2$, which affects both the scaling parameter and normalizing factor of PLF, to optimize the estimated error.

So for each simulation, the optimizing procedure is as follows:

- Given a value of $d_2$, we can compute the scaling parameter for PLF, as in Figure 2.6.
- The normalizing factor can be computed as the tail area from $x_0 = F^{-1}(1e - 4, 1, d_0)$ where $d_0$ is the original suggested degree of freedom from [48].
- Then the optimal $\hat{d}_2 = \arg\min_{d_2} Err(PLF(d_2), PLQ)$
Applying this procedure for all simulations of different scenarios to estimate the optimal $\hat{d}_2$ (Figure 2.12), the result shows that when $meanExpAB \leq 6$, the error between PLF and PLQ is minimized at denominator d.f. $\hat{d}_2 < d_0$ (in these simulations, the originally suggested $d_0 = 995$). Moreover, there is a monotonically decreasing relationship between $meanExpAB$ and $\hat{d}_2$ in this region. The minimized error is significantly dropped to the scale of $10^{-10}$ for $1 \leq meanExpAB$. The meaning of this result is that when $meanExpAB \leq 6$, the corresponding F distribution with the suggested degree of freedom from [48] is not the best-fit approximation of the $Q_l$ distribution. Rather, we should use a smaller denominator degree of freedom for the F distribution to achieve a better approximation, and thus a better testing reliability. Furthermore, this adjustment can be inferred from $meanExpAB$.

Figure 2.12: Optimal denominator degree of freedom $\hat{d}_2$ of simulations from different scenarios. The smaller degree of freedom for F distribution is a better fit to approximate $Q_l$ distribution (originally suggested $d_0 = 995$). There is a trend between $\hat{d}_2$ and $meanExpAB$ when $meanExpAB \leq 6$ that can help to recommend the degree of freedom based on $meanExpAB$. The estimated errors decrease to the range of $10^{-10}$ for $meanExpAB \geq 1$, which are two orders of magnitude smaller than using originally suggested $d_0$.

Therefore, to develop an adjustment guideline, we first model the relationship between $meanExpAB$ and $\hat{d}_2$ by fitting a line into the data when $mean(ExpAB) \leq 6$. However, instead of fitting the line exactly for $meanExpAB$ vs $\hat{d}_2$, we fit the line between $meanExpAB$ vs $\frac{d_0}{\hat{d}_2}$. Our reasoning is that the ratio $\frac{d_0}{\hat{d}_2}$ measures how $\hat{d}_2$ changes with respect to the
original suggested $d_0$, and by modelling this relative change, this model can be generalized to other testing scenarios with different values of $d_0$. Furthermore, the trend switches between $1 \leq \text{meanExpAB} \leq 2$ to a faster rate of change. Therefore, we actually fit two lines to the data (Equation 2.11). The regressions show that (1) the optimal degree of freedom $\hat{d}_2$ is similar to the original $d_0$ when $\text{meanExpAB}$ is around 6, (2) when $2 \leq \text{meanExpAB} \leq 6$, the rate of decreasing is about 2.5 times, and (3) when $1 \leq \text{meanExpAB} \leq 2$, the ratio of $\frac{d_0}{\hat{d}_2}$ decreases much faster with the rate about 8.5.

$$\frac{d_0}{\hat{d}_2} = a \times \text{mean(ExpAB)} + b$$

$a \approx -2.5$ with p-value $=1e-20$ ($2 \leq \text{meanExpAB} \leq 6$) \hspace{2cm} (2.11)

$a \approx -8.4$ with p-value $=7e-14$ ($1 \leq \text{meanExpAB} \leq 2$)

From these regressions, we suggest a guideline to adjust the degree of freedom for F distribution to get a better approximation for $Q_{\bar{l}}$ distribution.

$$\left\{ \begin{array}{l}
\text{new } d_2 = \frac{d_0}{2 \times (6 - \text{ExpAB})} \quad \text{for } 1 \leq \text{ExpAB} \leq 5.5 \\
\text{new } d_2 = d_0 \quad \text{for } \text{ExpAB} \geq 5.5
\end{array} \right. $$

(2.12)

The reason we choose this rule 2.12 is that (1) it is easy to recall while still improving the approximation (2 instead of the slope 2.5), and (2) it forms a lower bound for all simulations that we have (Figure 2.13). Next we test this rule to see if it actually improves the observed error, the difference in the tails between $Q_{\bar{l}}$ and our suggested F distribution. The result shows an improvement in the goodness of fit compared to the original observed error (Figure 2.14), suggesting that this rule is favorable compared to the original one.

2.4.5. Summary

In this section 2.4, we use a power law distribution to approximate the tails of the test statistic $Q_{\bar{l}}$ distribution, and its F-distribution-based approximation. From analyzing their power law distributions ($PLQ$ and $PLF$), we show that the observed error (the difference in tail areas of F and $Q_{\bar{l}}$ distributions) is well captured by the estimated error (the difference
Figure 2.13: The regression of \textit{meanExpAB} and $d_0/d_2$. Two pieces of regression show different rates of change for two intervals of \textit{meanExpAB}. \textit{Green curve}- The suggested adjustment for the F distribution’s degree of freedom to have a better approximation of the $Q_{\bar{l}}$ distribution.

in tail areas of \textit{PLF} and \textit{PLQ}). Since the error is relatively small in the sense of not losing the statistical significance of the test when $2 \leq \text{meanExpAB} \leq 6$, we suggest that the method can be safely used with the threshold for \textit{ExpAB} being lowered to 2. Furthermore, also from the analysis of the power law approximations, we proposed an adjustment for denominator degree of freedom for the F distribution to achieve a better approximation for $Q_{\bar{l}}$ test statistic’s distribution. Overall, this extension allows the cell assembly detection method to be applied to low firing rate spike train data. As we will show in the next section 2.5, for low firing rate data, the extended method will be able to perform the correlation test on more data, and also detect more assemblies, compared to the original method.
Figure 2.14: By adjusting the denominator degree of freedom when $\text{mean}(ExpAB) \leq 6$, the goodness of fit between $F$ and $Q_l$ distributions is improved.

### 2.5. Comparison of original and extended methods on calcium imaging data

Here we compare the performance of the original method (threshold on $ExpAB$ is set at 5) to that of the extended methods (threshold is set at 1 with and without degree of freedom adjustment). We denote them T5, T1 and T1a respectively. We include the T1 method in the comparison, not because we suggest it (as we have shown that the error is unreasonably high when $1 \leq ExpAB \leq 2$), but because we want to see what the false-positive rate would be compared to the suggested extended method T1a.

We apply the three algorithms to three spike train populations inferred from calcium imaging data of a B6C mouse participating in fear conditioning experiment. The three data sets come from the same mouse, but at three different periods in time (more details about data and experiments are given in the next chapter).

We first detect only correlated pairs (i.e. we do not perform the agglomerative clustering algorithm [48], to fuse pairs into larger assemblies). The results show that the number of pairs being tested is significantly increased in T1 and T1a compared to the original method.
Figure 2.15: Comparison between original and extended methods in detecting pair assemblies from a real calcium imaging dataset. The extended method T1 (without adjusting degree of freedom) and T1a (with adjusting degree of freedom) test more pairs and detect more significant pairs than the original method T5. The difference in significant pairs between T1 and T1a reflects false positives from using less stringent threshold in T1.

For example: in Epoch 1, only 15% of pairs are evaluated using the more stringent threshold of T5, while 60% of pairs are evaluated in T1 and T1a (left panel in Fig. 2.15). Similarly, the number of significant pairs also significantly increases (right panel in Fig. 2.15).

Moreover, we notice that the number of detected significant pairs in T1 is higher than in T1a. The reason for this is that, adjusting (or more accurately, reducing) the denominator degree of freedom for the F distribution will make the tail heavier, which equivalently will increase the threshold for the hypothesis test. In other words, for a given type-I error, the testing threshold for the test statistic will be higher in the T1a method than in the T1 method. However, since we have shown that the approximation error is higher in T1, especially when $1 \leq ExpAB \leq 2$, the difference in detected significant pairs between T1 and
T1a methods could be understood as the increased false-positive rate incurred by using the less stringent threshold of T1.

Next we look at how the methods are different in capturing the 2D histogram of significant pairs over binwidth and lag. Recall that when applying the method to different bin widths (or timescales), different significant pairs might be detected. Each assembly also has a characteristic time lag at which its co-activity occurs. Together this information defines a histogram of significant pairs over bin width and lag. As shown in [48], the bin-lag histogram, or the structure of significant pairs, could be quite dependent on an animal’s behavioral state: for example, between exploration and task-focusing.

Figure 2.16 shows that, compared to the T5 method, the bin-lag histogram from T1 and T1a are different. For example, in the second and third epochs, the T1 and T1a methods identify a large number of synchronous (lag 0) assemblies that go undetected by T5. Furthermore, the histograms from T1a seem to be cleaner (or more concentrated) than the ones from T1.

Next we apply the three methods to detect higher-order assemblies from the same data. In particular, we combine the three correlation test with the last two steps of the original CAD method ([48]). Compared to T5, T1 and T1a methods are able to detect more assemblies. The default pruning process is used to only keep the largest assemblies across bin widths. The distributions of assemblies over assembly sizes also show that T1 and T1a methods tend to detect higher-order assemblies (Figure 2.17). Finally, we show the raster plot of the first 20 detected assemblies from the B6C mouse to visualize the output of the algorithm (Figure 2.18). The plot shows neuron activities and highlighted assembly activities over three epochs. Each row represents activities of one neuron. Each assembly is represented by a different color.

2.6. Discussion

In this chapter, we extended the pairwise correlation test of the cell assembly detection method in [48] to be more applicable to sparser, or lower firing rate, spike train data than
what had been previously reported. This type of data is usually seen in calcium imaging recording. The key component that prevents the original algorithm from running the correlation test on a pair of spike train is the threshold imposed on their expected join spike count ($E[\#_{AB,l}]$), a quantity used to construct the test statistic. From our real-world calcium imaging dataset, we observed that most of the spike train pairs do not pass this threshold. Therefore, using the original threshold will potentially result in undetected assemblies, losing information from the data. Our extended method alleviates this limitation, on average doubling the number of detected cell assemblies from the data without an increase in false positives.
Figure 2.17: Method comparison for assembly detection. (Left figure) Assembly counts from three methods. T1 and T1a methods detect more assemblies compared to T5. (Right figure) Distribution of assemblies over assembly sizes. T1 and T1a methods tend to detect assemblies with larger number of cells compared to T5.

Figure 2.18: Raster plot of a B6C mouse. Each row represents activities of a neuron. Each assembly activities are highlighted by a different color.
We did this by first using the power law distribution to approximate the tail of the test statistic distribution and its corresponding F-distribution, and then using the power law approximations to estimate the tail area difference between the two distributions (observed error). The result showed that the power law approximations well capture the observed error. Then by analyzing the power law approximations, we showed that in the region of small expected joint spike count \( E[\#_{AB,l}] \leq 6 \), there exists a better degree of freedom than originally suggested for the F distribution. Therefore, we suggest to adjust the F distribution’s degree of freedom in this region to have a sufficiently good approximation for the test statistic \( Q_l \). With this adjustment, the threshold can be safely lowered from 5 to 1 without losing the test reliability. We then showed that the extended method tests and detects more assemblies compared to the original one, which indicated that the extended method is more suitable for low firing-rate spike train data. This result provides a more powerful data analysis tool for the common calcium imaging data to detect cell assemblies which are believed to be for neural processes.

**A potential extension:** A limitation of this pairwise correlation test is that the test statistic is constructed using only information from the optimal lag. Given the correlation is significant at that lag, useful information about assembly at other time lags might be wastefully discarded. By considering the co-activities that happen at multiple lags, one may be able to detect assemblies whose firing pattern is formed at multiple time scales. This type of multiple time scales co-activity patterns has been observed in experimental data [37]. Adding this feature into the algorithm is an interesting but challenging future work that we wish to investigate.
2.7. Appendix

Here we provide details on some probability concepts we refer to throughout Chapter 1.

2.7.1. Hypergeometric distribution of joint binary subprocesses

The hypergeometric distribution is a discrete probability distribution that describes the probability of \( k \) successes in \( n \) draws, without replacement, from a finite population \( N \) that contains exactly \( K \) objects with that feature. A random variable \( X \) follows the hypergeometric distribution if the probability mass function is given by

\[
p_X(k) = Pr(X = k) = \binom{K}{k} \binom{N-K}{n-k} \binom{N}{n}
\]

Now suppose that we have two independent stationary binary subprocesses \( A \) and \( B \), whose total spike counts \( \#_A = c_A \) and \( \#_B = c_B \) are fixed by the observed data. The length of spike train is \( N \). It is shown in [16] that the distribution of \( \#_{AB,l} \), a joint spike count of \( A \) and \( B \) with lag \( l \), conditional on specific realizations of spike counts, is described by the hypergeometric distribution:

\[
Pr(\#_{AB,l} = k | \#_A = c_A, \#_B = c_B) = \binom{c_A}{k} \binom{N-c_A}{c_B-k} \binom{N}{c_B}
\]

2.7.2. Hypothesis testing

Hypothesis testing is a statistical method for assessing the probability of a hypothesis. We will illustrate the process of hypothesis testing by an example of testing for a mean value.

Suppose that we have a data sample \( x_i, i \in \{1, ..., N\} \), which is assumed to be independently drawn from an identical distribution (i.i.d.). We first need to form a null hypothesis \( H_0 \) about the mean, i.e. the population mean \( \mu \) is \( \mu_0 \). An alternative hypothesis \( H_1 \) would be that the population mean \( \mu \) does not equal \( \mu_0 \). Based on the sample data, we want to test if the null hypothesis \( H_0 \) is true.

Next we need to set criteria for making the decision, i.e. when to accept/reject the null hypothesis. Assuming the null hypothesis is true, the level of significance, shows the
likelihood of a particular value of the sample mean. If the likelihood is too small, or it is too unlikely to get that sample mean, we will reject the null hypothesis. By setting the threshold $\alpha$ for the level of significance, we have set the criteria for our test. The value of $\alpha$ is usually set at 5%. The threshold $\alpha$ is also called Type I error, because it is the probability of rejecting a null hypothesis when it is actually true.

The next step is to compute the level of significance for the observed sample data. For this purpose, we use the test statistic. Specifically, in this example, where we test the hypothesis about the population mean, with an unknown population variance, a t-test will be used.

**Definition 2.1** [Student’s t-distribution] If $Z \sim \mathcal{N}(0, 1)$ and $U \sim \chi^2_\nu$, and $Z$ and $U$ are independent, then the distribution of $Z/\sqrt{U/\nu}$ is called the t distribution with $\nu$ degrees of freedom. The probability density function (pdf) of the Student’s t-distribution with degree of freedom $\nu$ is given by

$$f(x) = \frac{\Gamma\left(\frac{\nu + 1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu + 1}{2}}$$

where $\Gamma$ is the gamma function.

**Theorem 2.1 (t-statistic)** Suppose that $x_i, i \in \{1, ..., N\}$ is independently and identically distributed from a normal distribution with mean $\mu_0$. The sample mean $\bar{x}$ and sample variance $s^2$ are computed as $\bar{x} = \sum_{i=1}^{N} x_i/N$, $s^2 = \sum_{i=1}^{N} (X_i - \bar{X})^2/(N - 1)$.

Then the t-statistic $(\bar{x} - \mu_0)s/\sqrt{N}$, follows $t_{N-1}$ Student’s t-distribution with degree of freedom $N-1$.

**Remark 1** The test statistic can be rewritten as follows:

$$\frac{\bar{x} - \mu_0}{s/\sqrt{N}} = \frac{\sqrt{N}(\bar{x} - \mu_0)/\sigma}{\sqrt{s^2/\sigma^2}}$$

Notice that if $X \sim \mathcal{N}(\mu_0, \sigma^2)$, $(N - 1)s^2/\sigma^2$ are proved to follow $\chi^2_{N-1}$, the chi-square distribution with $N - 1$ degrees of freedom [44]. Then by the Definition 2.1, the test statistic follows $t_{n-1}$.

In the case that $X$ does not follow a normal distribution, the numerator $\sqrt{N}(\bar{x} - \mu_0)/\sigma$ is asymptotically normal by Central Limit Theorem (Theorem 2.2 - CLT). How well it follows a normal distribution depends on the distribution of $X$ and the sample size. However, there is no CLT analog for sample variance. Therefore, we cannot theoretically prove that the test statistic is t-distributed without normality.
Figure 2.19: The pdf of t distribution at different degree of freedom $\nu$.

**Theorem 2.2 (Central Limit Theorem)** Let $\{X_1,\ldots,X_N\}$ is a sequence of i.i.d. random variables with $E(X_i) = \mu$ and $\text{Var}(X_i) = \sigma^2 \leq \infty$. Then as $n$ approaches infinity, the random variable $\sqrt{N}(\bar{X} - \mu)$ converges in distribution to a normal $\mathcal{N}(0,\sigma^2)$

$$\sqrt{N}(\bar{X} - \mu) \xrightarrow{d} \mathcal{N}(0,\sigma^2)$$

Using the above definition and theorems, the t-test can be implemented as follows

- Given sample mean $\bar{x}$ and sample variance $s^2$, we can compute the t-value

  $$t = \frac{\bar{x} - \mu_0}{s/\sqrt{N}}$$

- Using the pdf, we can compute p-value, the likelihood of the t-value, or equivalently the level of significance of the sample mean, given that the population mean is $\mu_0$.

- How we compare the p-value with $\alpha$, and make decision depends on the alternative hypothesis $H_1$.

  If $H_1 : \mu \neq \mu_0$, then we will have a two-tailed t-test (Figure 2.20). Therefore, if p-value $\leq \alpha/2$, we accept the null hypothesis, and reject otherwise.

  If $H_1 : \mu > \mu_0$ or $H_1 : \mu < \mu_0$, then we will have a one-tailed t-test (Figure 2.20). Therefore, if p-value $\leq \alpha$, we accept the null hypothesis, and reject otherwise.
There is another distribution, F-distribution, which can also be used for hypothesis testing. The following theorem relates F-distribution to t-distribution.

**Definition 2.2 [F-distribution]** For real $x > 0$, the probability density function (pdf) of the F distribution with degree of freedom $d_1$ and $d_2$ is given by

$$f(x; d_1, d_2) = \frac{\sqrt{(d_1 x)^{d_1} d_2 / (d_1 x + d_2)^{d_1 + d_2}}}{x B(d_1/2, d_2/2)}$$

where $B$ is the beta function.

**Theorem 2.3 (F-distribution [44])** If $X \sim t_n$, then $X^2 \sim F_{1,n}$.

Now we can show that the test statistic developed in Section 1.2 approximately follows the F-distribution. Given $\#_{ABBA,i}$, $\mu_{ABBA,i}$, $\bar{\sigma}_{ABBA,i}$, by Theorem 2.1, we have approximately

$$\frac{\#_{ABBA,i} - \mu_{ABBA,i}}{\bar{\sigma}_{ABBA,i}} \sim t_\nu$$

where $\nu = 2(T - \bar{t})M - 1$

Then by Theorem 2.3,

$$Q_i = \left(\frac{\#_{ABBA,i} - \mu_{ABBA,i}}{\bar{\sigma}_{ABBA,i}}\right)^2 \sim F_{1,\nu}$$

The F-test can be implemented as same as t-test. (Figure 2.21)

Notice that $\#_{ABBA,i}$ has $2(T - \bar{t})M$ terms, whose distribution is unknown, so as discussed above, the numerator of the t-statistic is only asymptotically normal, and we cannot prove that the denominator is chi-squared. However, as shown in [48] and in Section 2.4 using...
simulation, the test statistic is accurately captured by the F-distribution in the case of high expected joint spike counts.

Figure 2.21: Rejection region of F-test
Chapter 3
Application to Fear Conditioning Data

3.1. Introduction

During the fear conditioning experiment, the animals are tested on encoding and retrieving memories for a familiar context, and on generalize the encoded information to another novel context. While the animals are performing the tasks, the calcium imaging technique is used to record the activities of hundreds of neurons in CA1 hippocampus, a brain region known for being associated with memory and learning. The datasets are from two strains of mice (B6C and B6J). We observed that there are differences in term of freezing level, a proxy for memory recall and generalization, across animals. This indicates the differences in recognizing familiar contexts, and in distinguishing between contexts. Therefore, the goal of the analysis is to associate the neuronal activities with the freezing behavior, and understand how the neuronal population encodes, recalls and generalizes the information. Generally, these tasks can be performed at the level of a single neuron, pairwise correlations or higher order correlations between neurons. Moreover, these correlations could happen at different time scales, and in various forms: synchronous spikes, sequentially coordinated spikes, co-variations in firing rate, etc.

Following that line of reasoning, we analyze the data at the levels: single cell, pair assembly and full assembly analysis. For the assembly analysis, the extended CAD method is used. Specifically in this study, we want to address two questions: (1) how does the hippocampus encode, retrieve and generalize contextual memories? and (2) since the behavior of two strains are different, what are the corresponding differences in neuronal activity?
We first show that there exists a group of neurons whose activities during the exploration period can predict how well the animal distinguishes between familiar and novel context. This correlation is detected at different level of analysis, at the single cell level for B6C and at the pair assembly level for B6J. Moreover, the correlation trend is opposite between two strains, which implies that the processing mechanism for contextual memories between two strains are different. Another finding about this special group of neurons is that, only in B6C, they are significantly enriched in pair and full assemblies. Overall, one implication from these analyses is that different sets of information about the data are extracted at different levels.

We also found that synchronous and non-synchronous pairs assemblies are important and involved in different ways. While non-synchronous pairs seems to associated with novel contexts, synchronous pairs have much stronger correlation with freezing episodes in testing contexts. Besides, this correlation is higher in B6J.

Lastly, we found that neurons participating in an assembly tends to stay closer to each other compared to what we would predict by chance. This supports the hypothesis that those neurons in the detected assemblies might be internally connected.

This chapter is organized as follows. First, we describe the experiment and how the data was collected in Section 3.2. Then in Section 3.3, results of single cell analysis are shown. The results of pair assembly analysis are shown in two Sections. Section 3.4 is the analysis for pair assemblies detected in each sessions separately, while Section 3.5 is the results from pair assemblies detected from combining data from all sessions. Next the results for full assembly analysis is shown in Section 3.6. Lastly, in Section 3.7, all results will be highlighted, connected and interpreted as evidences supporting for different main points. Their implications will also be discussed.
3.2. Fear Conditioning Experiment

The fear conditioning experiment has been conducted by Xu Lab at UT Southwestern Medical Center. In this experiment, calcium imaging technique is used to record the neuronal activity from two strains of mice (B6C and B6J) at the CA1 region of the hippocampus during contextual fear conditioning training and tests.

3.2.1. Experiment procedure

As shown in Fig 3.1, a lens was implanted into the animal’s skull, and two months later, a mini-endoscope was installed. This setting allows the animal to move freely around the environment. Moreover, the mini-endoscope enables us to monitor neuronal activities from about 200 CA1 neurons per animal over days.

Each animal participates in three sessions (Fig 3.2). The first session is training, in which the animal is put into a novel context. During the first two minutes, the animal explores the environment. After that, the animal will be alerted by a tone and then stimulated by a two-second electric foot-shock three times, 30 seconds apart. The idea is that in the first two minutes, the animal will encode information about the environment, and by giving them the shocks, those information or memories will be consolidated.

The second session is context test, in which the animal is put into the same environment as in training session, but this time, there is no shock being presented to the animal, and they just explores for 5 minutes. In this session, the animal hopefully will start recalling the memories about this environment when it encounters contextual information.

Figure 3.1: Recording Equipment Installation.
Figure 3.2: An animal participates in three sessions. In the training session, the animal explores a context and receives stimuli. In the context test, the animal revisits the original context. In the altered context test, the animal visits a novel context.

The third session is altered context test, in which the animal is put in a different environment. There are two questions we want to answer during this session. The first question is whether the animal can distinguish this context from the original context. Equivalently in term of neuronal activity, is the activity during the context test different from the activity during the altered context test? The second question is about whether the animal can generalize the contextual memories they collected in the original context to recognize this altered context, and if yes, at which level they can do it.

As shown in Fig 3.2, the altered context has some differences compared to the original context. The metal bars for giving the electric shocks have been removed, and the wall decoration is also changed. Other than those, the altered context is the same as the original context. Therefore, we would expect to see some level of memory generalization.
3.2.2. Data collection

The calcium signal video, or a stack of images were captured at the frequency of about 10.5 Hz. Fig 3.3 shows the image of one single frame. The recorded videos from 14 B6J mice and 11 B6C mice were then processed by *Constrained Nonnegative Matrix Factorization for microEndoscopic data* (CNMF-E) [62]. The method first applied the matrix factorization technique to get the calcium signal traces. The idea is that from all the images of one session, the algorithm identifies and separates the unchanged background from each image. The remains of each image, or the foreground, show the calcium signal intensity and the location of neurons at that frame. Finally, the neuron locations from all frames were uses to create their contours (Fig 3.3). After that, the algorithm deconvolutes the calcium traces to infer calcium events, which is the spike train data (Fig 3.4).

3.3. Single Cell Analysis

Given that the spike train data was already inferred from calcium-imaging recordings, we first looked for the difference in neuronal activities between two strains of mice. The average activity across all neurons within one strain showed that CA1 neurons in B6J mice were immediately activated after shocks, while those in B6C experienced a delayed response (Fig 3.5).

3.3.1. Neuron classification

![Image of calcium intensity image and neuron contours](image)

Figure 3.3: *left- An example of a calcium intensity image; right- A plot of neuron contours identified by CNMF-E method.*
Figure 3.4: Constrained nonnegative matrix factorization for micro-Endoscopic data method (CNMF-E). This method identifies neurons location and infers calcium events from calcium imaging data.

Figure 3.5: The average activity of all neurons within each strain. CA1 neurons in B6J mice show an immediate activation after shocks, whereas those in B6C have a delayed response.
The average activities from each strain suggested that the next step would be to look at
the group of neurons in each animal which strongly shows the shock-responding character-
istics. We did this by classifying neurons based on their responsiveness to the shocks.

First, we used the method described in [29]. For each neuron, and for each shock, the
activity from a before-shock time window $X_b$ and from an after-shock time window $X_f$ were
chosen. The firing rate difference between these periods was compared with the shuffled dis-
tribution to see if the response is significantly higher than it would be observed by chance.
Afterward, we showed that the shuffled distribution is equivalent to a hypergeometric distri-
bution. This helps to reduce the computational cost of the shuffling process (see Appendix
3.8.1 for the discussion about the shuffled algorithm).

The neuron population was divided into five mutually exclusive groups (R, DR, C, S and
A group). The first group, responsive group (R group), contained significantly responsive
neurons of at least one shock, when the after-shock time window was set to 10 second after
the shocks. The second group, delayed responsive group (DR group), contained significant
delayed responsive neurons of at least on shock, when the after-shock time window was set
from 10 second to 30 second after the shocks. Before-shock time window was always chosen
as 10 seconds before the shock. If a neuron was both responsive and delayed-responsive for
the same shock, it was placed into combine group (C group) (Figure 3.6). The silent group
(S group) contained neurons that were not responsive or delayed-responsive to any shocks.
The A group contains neurons that were responsive to one shock, but delayed responsive
to a different shock. We think this group is rather special and requires its own analysis to
understand their responses. These neurons might encode different information that can be
further investigated.

Moreover, we checked the average pairwise distance within each group and compare with
the neuron population. The result suggests that there is no spatial aggregation within each
group (Figure 3.7).
Figure 3.6: Time windows for identifying different groups of neurons.

Figure 3.7: *left-* The contours of estimated neurons by CNMF-E. *right-* There is no spatial aggregation within each group of neurons.
3.3.2. Which neurons encode contextual information?

After clustering neuron populations into different groups (R, DR, C, S groups), the firing rates, or calcium events per second, of each group during interested time windows were calculated. The firing rate of a group is the average of neurons firing rate in that group during a time window. Moreover, we also separated activities into active and freezing time (or frame). A freezing frame is a frame in which the animal does not have any movement. This is assessed by the recording software, which compared the movement frame by frame.

We then compared the firing rate between three sessions, and see that the neuron with delayed response in DR and C groups are relatively more active in both training and context sessions, but not in altered context session. This is confirmed by the t-test between two groups with the significant threshold of 0.001 (Figure 3.8). Moreover, the firing rate of those neurons during the freezing period, in which the animal does not move, of the context session is also significantly higher (Figure 3.9). This suggests that those neurons participate in encoding and retrieving the contextual information.

Another result related to the delayed responsive neurons is that the activity of these neurons (DRC group neurons) during exploration in training predicts memory specificity. This correlation only exists in the group of B6C animals. For clarity, the exploration in training is the first two-minute period of the training session. Memory specificity is measured by the discrimination index, which is defined by

\[ DisIndex = \frac{Fr_C - Fr_A}{Fr_C + Fr_A} \]

where \( Fr_C \) and \( Fr_A \) is the animal freezing time in the context session, and in the altered context session, respectively. Since \( Fr_C \geq Fr_A \) in all animals, we always have \( 0 \leq DisIndex \leq 1 \). The value 0 means the animal cannot distinguish between the original context and altered context, while the value 1 means that the animal can. As shown in Figure 3.10, the firing rate during the training exploration is positively correlated with the discriminant index, which indicates that if the delayed response neurons are relatively more active during the exploration, their freezing level in two contexts are more separable.
3.3.2.1. Memory specificity vs. generalization

Intuitively, we can think that higher discrimination index means more contextual information is picked up by the animal, or higher memory specificity. However, this does not necessarily mean better learning ability. Decoding too many details about the environment will reduce the chance of recognize similar contexts in the future, which will cause low learning efficiency. On the other hand, if too few details are encoded, the animal cannot distinguish between different contexts. This is called *over-generalization*, and has been linked to diseases such as PTSD [22].

3.3.3. More correlations from fully recorded animals

In order to analyze activities of different groups of neuron in context and altered context, here we only focus in 18 fully recorded animals (8 B6C and 10 B6J animals).
Figure 3.9: Neurons with delayed response are active during memory retrieval, i.e. freezing in context session. The symbol *** indicates a significant t-test between two groups with p-value < 0.001.

Figure 3.10: The correlation between discriminant index and firing rate of DRC neurons in the training exploration is significant in B6C, but not in B6J. Each dot represents one animal.

We first look at the firing rate of DR cells whose firing rate in exploration is correlated with discrimination index as we found above. We found that, only in B6C strain, DR cells firing rate in exploration, training context, context test, and altered context are all positively correlated with discrimination index (Figure 3.11). This means if DR cells are more active in any epoch, the animal distinguish the contexts better. More interestingly, in the plot between discrimination index and DR cells firing rate in context test (bottom-middle plot of Figure 3.11), we notice that the correlation trends are different between B6C and B6J strain. Even though it is only an insignificant trend in B6J (p-value = 0.069), this might indicate that DR cells in both strains play different role in memory retrieval process.
Figure 3.11: DR cell firing rate vs freezing behaviors. (1) Higher firing rate of DR cells in any epoch is associated with higher discrimination index. (2) (Bottom-middle plot) Even though it is only an insignificant trend, there is a difference between how DR cells firing rate of both strains in context test are associated with discrimination index.

Next we look at the firing rate of R cells, the group of cells that immediately response to the shocks in training. Again we only found correlations in B6C strain (Figure 3.12). The R cell firing rate in training and altered context are positively correlated with discrimination index, while the firing rate in altered is negatively correlated with altered freezing. For the C group, we only found the correlation between their firing rate in altered context and discrimination index, again only in B6C (Figure 3.13).

Since the firing rate of both R and DR cells in B6C shows some correlations with freezing behaviors, we went further and checked the firing rate of all neurons in different epochs (Figure 3.14). Only in B6C, we found that the firing rate of all neurons in context test and altered context are positively correlated with discrimination index. Moreover, they are also negatively correlated with freezing level in context test and altered context. These correla-
Figure 3.12: R cell firing rate vs freezing behaviors. Correlations are found only in B6C. (1) Firing rate of training and altered context are positively correlated with discrimination index, and (2) firing rate in altered context is negatively correlated with freezing in altered context.

Figure 3.13: C cell firing rate correlation. B6C C cell firing rate in altered context is positively correlated with discrimination index.

tions indicate that the B6C animals encoded contextual information in training context, and that neurons in hippocampus CA1 in B6C actually participate in the process of encoding, decoding contextual memories. Overall, the correlations also show that DR cells in B6C play an important role in encoding information in the exploration epoch, and that DR cells in B6J might play a different role in this process. More data points are needed to validate the role of B6J DR cells.
Figure 3.14: Correlations of all cell firing rate and freezing behaviors. In B6C, all cell firing rate in context and altered are negatively correlated with freezing level and positively correlated with discrimination index. These correlations indicate that Hippocampus CA1 neurons involve in the process of encoding and retrieving contextual memories.

3.4. Pair Assembly Analysis for Separate Epochs

In this section, we analyzed the pairwise correlations of the neuronal populations by using the test statistic developed in the previous chapter. For each animal, we detected significantly correlated pairs of neurons in each epoch separately. A pruning step is applied in the way such that if a pair is detected at multiple bin width, only the one with the smallest p-value is kept and others are discarded. After that, we constructed the bin-lag histogram of the significant pairs’ distribution. By observing the average bin-lag histogram of each strain, as shown in Figure 3.15(A), we notice that there are fewer significant pairs in the exploration session (the first 2 minutes of the training epoch), compared to the other three sessions. This difference is significantly confirmed by t-test. This observation is similar to
the finding of [48], in which they also detected significant pairs in different brain regions (ACC, CA1 and EC) when the animals perform either the exploration task, exploring the linear track, or the delayed alternation task. They found that during the latter, the much larger and richer structures of significant pairs are found, and that the pair structures are focusing more in higher lag and larger bin sizes.

Figure 3.15: (A) Average bin-lag histogram within strains. (B) Marginal distribution of significant pairs across bin widths. (C) Percentage of significant pairs over all tested pairs in each epochs.

Moreover, B6J has more significant pairs than B6C (Figure 3.15(C)). In term of the bin-lag structures of the pair distribution, the *synchronous pairs*, pairs whose lag is zero, is dominant in B6J, while oppositely, the pair structure in B6C is mixed between synchronous and non-synchronous pairs. We also speculated that when the animals are introduced to a novel context (exploration, training and altered context), the assemblies at higher lags
and larger bin widths appeared more often. This leads us to take a loo into the population of synchronous pairs and non-synchronous pairs separately. As shown in Figure 3.16, the percentage of synchronous pairs is higher in B6J, but the percentage of non-synchronous pairs are comparable in both strains. The ratio of non-synchronous pairs over significant pairs are higher in novel contexts (exploration, training and altered context), but drops in the context test. This confirmed our above speculation about non-synchronous pairs appear more often in the context with novelty. Next we will show some evidences to show that non-synchronous pairs involve in encoding the novel contextual information.

Figure 3.16: Synchronous vs non-synchronous pairs as percentage of all pairs. **Left**- B6J has higher percentage of synchronous pairs than B6C. **Middle**- Both strains have comparable percentage of non-synchronous pairs. **Right**- Nonsynchronous pairs as a percentage of significant pairs. The ratio is higher in the context with novelty (exploration, training and altered context) compared to a familiar context (context test).

### 3.4.1. Non-synchronous pair ratios and freezing behaviors

Here we show statistically significant linear regressions between non-synchronous pair ratios and freezing behaviors (context freeze, altered freeze and discrimination index). Firstly, as shown in Figure 3.17, the B6J’s non-synchronous pair ratio in altered context is positively correlated with context freeze. If a B6J animal freezes more in the context test, their spike train in altered context observed more non-synchronous pairs. Moreover, the B6C’s non-synchronous pair ratio in context test is negatively correlated with altered freeze, and positively correlated with discrimination index. These show if a B6C animal has more non-
synchronous pair, they will freeze less in altered and also distinguish the contexts better. All these regressions agree that with more non-synchronous pairs, the animal seems to better encode contextual information.

Figure 3.17: Correlation between non-synchronous pair ratio and freezing behaviors. (B6J) An animal freezing more in context test observes more non-synchronous pairs in altered context. (B6C) An animal with more non-synchronous pairs in context freezes less in altered context, and has higher discrimination index. All significant correlations indicate that an animal with more non-synchronous pair seems to better encode contextual information.

Secondly, we also found significant correlations between non-synchronous pair as a ratio of significant pairs and freezing level in the context test (Figure 3.18). Particularly, a B6J animal, with more non-synchronous pair relatively to synchronous pair, freeze more in context test. This indicates non-synchronous pairs are related to the learning process.

Figure 3.18: B6J animals with more non-synchronous pairs as a ratio of significant pairs freeze more in the context test. This indicates non-synchronous pairs are related to the learning process.
3.4.2. Decomposition of bin-lag histogram

From observing different patterns of the averaged bin-lag histogram between strains and across epochs, we utilize a tensor decomposition method called Nonnegative Canonical Polyadic Decomposition (NCPD) [6], combined with a latent dimension determination method [57], to decompose the bin-lag histogram of significant pairs from all animals within one strain. Particularly, for each of four epochs, and for a strain, we have a tensor whose dimensions are 11 lags $\times$ 7 bin widths $\times$ number of animals (10 B6J and 8 B6C). The NCPD then decomposed the histogram tensor into $k$ rank-one tensors, where the latent dimension $k$ is determined based on the given tensor (Figure 3.19). Each rank-one tensor is an outer product of three vectors in lag, bin and animal dimension. Then the outer product of the lag vector and the bin vector is a histogram pattern, and the animal vector will show how strongly that pattern expresses in each animals.

![Figure 3.19: Nonnegative canonical polyadic decomposition](image)

Figure 3.19: Nonnegative canonical polyadic decomposition for bin-lag histogram: each factor is a rank-one tensor. The outer product of the lag vector and the bin vector in a factor is a histogram pattern. The animal vector shows the pattern expression across animals.

3.4.2.1. Histogram pattern analysis

The NCPD factors extracted from histogram tensors of both strains are shown in Figure 3.20 (B6J) and Figure 3.21 (B6C). Since we focus on learning about synchronous pairs vs.
non-synchronous pairs, we label each factor either synchronous or non-synchronous pattern based on which type of lag the histogram contains. For example, we name the first factor in exploration as E1s as it contains only synchronous pairs. We specify the name and the label above each histogram pattern as either s (synchronous) or ns (non-synchronous). We also do not analyze factors which is strongly expressed in one animal, since that factor is not representative for the strain.

3.4.2.2. Regression between NCPD factors and freezing behaviors

The expression level of each histogram pattern in both strains will be regressed against freezing level in context test and altered context, and discrimination index. All significant correlations are collected in Figure 3.22. Interestingly, all the correlations are from non-synchronous pairs. Similar to the result in the pair ratios, a B6J animal, freezing more in context test, has a stronger expression from a non-synchronous pattern (A3ns) in altered context. Besides, a B6C animal with stronger expression of a non-synchronous histogram pattern in training (T3ns) would have a higher discrimination index. Finally, also in B6C animals, stronger expression from a non-synchronous pattern in context test (C2ns) leads to less freezing in altered context and higher discrimination index.

Again, these correlations from NCPD factor analysis indicate that more non-synchronous pairs will help the animals learn and distinguish the context better.
Figure 3.20: NCPD factors of B6J in all four epochs. The factors are labeled either as s (synchronous) or ns (non-synchronous) based on their histogram patterns. Each pattern has an expression vector specifying how strongly that pattern is expressed in the animal within a strain.
Figure 3.21: NCPD factors of B6C in all four epochs.
Figure 3.22: Regression of NCPD factors expression level and freezing behavior. All correlated factors are non-synchronous factors. Especially, in B6C, the expression level of one non-synchronous histogram pattern in training is positively correlated with discrimination index.

3.4.3. Pair assembly activities and freezing episodes

To further assess the difference between synchronous and non-synchronous pairs, we utilize the total correlation coefficient [19] to measure the dependencies between a neuron/pair assembly activities with the freezing episode in context test and altered context. The total correlation measure captures dependencies between two processes over long timescales, although this measure might also capture the dependencies over reasonably short timescales [3, 47].

For each of the two epochs, the total correlation between each pair assembly activities and freezing episodes was computed. The distribution of the total correlation from pair assemblies was then compared with the one from neurons. In the context test and only in B6J, total correlation of pair assemblies is significantly higher than the total correlation of neurons. However, in both strains, the total correlation of synchronous pairs is significantly higher than of nonsynchronous pairs. While the total correlation of synchronous pairs are comparable between two strains, the total correlation of non-synchronous pairs in B6J is significantly higher than in B6C (Figure 3.23). This indicates that the non-synchronous pairs cause the difference between B6C and B6J.
In altered context, the overall total correlation is lower than in context test. Besides, the level of the total correlation in B6J is significantly higher than in B6C, and also the total correlation of synchronous pairs is higher than of non-synchronous pairs in both strains (Figure 3.24).

In summary, these correlations indicate that overall, (1) the synchronous pairs’ activities are strongly correlated with the freezing episodes, and (2) the pair assembly activities in B6J are more associated with the freezing episodes compared to B6C.

![Figure 3.23](image-url): Total correlation distribution of cells and pair assemblies in context test. The total correlation of B6J pair assemblies is significantly higher than the total correlation of neurons. Moreover the total correlation between synchronous pairs and context freezing episodes are much stronger than of non-synchronous pairs.

3.5. Pair Assembly Analysis for Combined Epochs

In this section, we extract pair assemblies from spike train combined from all epochs (training, context test, and altered context). This analysis only applies for 18 fully-recorded
Figure 3.24: Total correlation distributions of pair assemblies in altered context. The total correlation level in altered context is lower than in context test. Similar to context test, the total correlation of synchronous pairs is also higher than of non-synchronous pairs.

animals. For each animal, a set of pair assemblies is obtained. If a pair is detected at multiple bin width (time scale), a pruning step is applied to only keep the pair with the lowest p-value. Therefore, there is not double count in the pair population. We will then cluster the assembly population in a different way to find the ones related to freezing behaviors. Especially, we look at how different type of cells (R,DR,C,and S cells) participate in the pair assembly population.
3.5.1. Synchronous versus nonsynchronous pairs

3.5.1.1. Pair assembly ratios analysis

Firstly, by looking at the bin-lag histogram from both strain (Figure 3.25), we found that B6J has significantly more synchronous pairs than B6C, and both strains have comparable ratios of significant pairs and non-synchronous pairs. However, as a ratio of significant pairs, B6C has significantly more non-synchronous pairs. From here, we found that (1) in B6C, having more synchronous pairs is associated with less freezing in the context test, and (2) in B6J, higher ratio of non-synchronous pair over significant pairs is associated with more freezing in the context test (Figure 3.26).

![Figure 3.25: Average bin-lag histogram for combined epoch pair assemblies from both strains. B6J significantly has higher number of synchronous pairs, while B6C significantly has higher number of non-synchronous pairs as a ratio of significant pairs.](image)

3.5.1.2. Pair assembly activity analysis

Since the activities of these pair assemblies are extracted from spike train combined from three epochs, we can analyze their activities as we did in the single cell analysis. However, here we focus on synchronous and non-synchronous pair assemblies. We started by looking at the firing rate of each groups in different epochs (exploration, training, context test and altered context). Figure 3.27 shows that the firing rate of synchronous pairs are comparable across four epochs. This is true for both strains. More interestingly, the firing rate of the non-synchronous pairs is significantly lower in context test when compared to the firing rates
Figure 3.26: Regression between combined epoch pair ratios vs freezing behavior. (1) The B6C significant pair ratio is positively correlated with discrimination index. (2) Having more synchronous pairs is associated with less freezing in the context test in both strain. (3) B6J - Higher ratio of non-synchronous pair over significant pairs is associated with more freezing in the context test.

in exploration, training and altered context. This difference is confirmed by t-test, with an exception of the difference between firing rate of B6C non-synchronous pairs in context test and in altered context (p-value=0.07). This indicates that the non-synchronous pairs are more active in exploration, training and altered context which are novel contexts.

Moreover, we found that (1) in both strains, if the pair assemblies are more active in context, the animal will freeze less in context, (2) the firing rate of the B6C non-synchronous pairs in training is positively correlated with context test freezing and (3) the firing rate of the B6C non-synchronous pairs in context test is positively correlated with discrimination index (Figure 3.28). These show that the non-synchronous pair assemblies actually participate in recall memories and distinguish between contexts.
Figure 3.27: Firing rate of synchronous pairs and non-synchronous pairs. The firing rate from synchronous pairs are comparable across epochs in both strain. The firing rate of non-synchronous pairs is significantly lower in the context test.
Figure 3.28: Regression between non-synchronous pairs activities and freezing behaviors. 
Left- The firing rate of the non-synchronous pairs in training is positively correlated with context test freezing. 
Right- The firing rate of the non-synchronous pairs in context test is positively correlated with discrimination index.
3.5.2. Cell type containing pair assemblies

3.5.2.1. Cell participation in pair assemblies

Here we analyze how different types of cells from single-cell analysis participate into pair assemblies. First, given a number of different cell types in the neuron population, we ask if any type of cells is more actively participating into pair assemblies. This is done by simulating many pair population and computing the ratio of different cell type containing pairs. From these, the distributions of the ratio are constructed, and the p-value for each pair type's ratio are computed. Figure 3.29 shows that the observed ratio of DR containing pairs is consistently larger than the mean ratio from simulated distribution across all animals in B6C. This means the DR cells in B6J is enriched in pair assemblies compared to other cell type. Moreover, the ratio of DR containing pairs is also significantly higher than others in B6C. Furthermore, the mean p-value within each strain also confirms this assessment (Table 3.1). Lastly, the participating ratio of DR cells in non-synchronous pairs are significantly higher, while in B6J strain it is R cells in synchronous pairs (Figure 3.30).

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>R</th>
<th>DR</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>B6C</td>
<td>0.85</td>
<td>0.54</td>
<td>0.065</td>
<td>0.3361</td>
</tr>
<tr>
<td>B6J</td>
<td>0.78</td>
<td>0.43</td>
<td>0.48</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 3.1: Mean p-value of cell type containing pair ratio. The small p-value of DR containing pair ratio in B6C indicates that DR containing pairs are enriched in B6C.

3.5.2.2. Cell containing pair activities

By associating DR containing pair activities with freezing behaviors, we found that their firing rate of both strain in the context test are correlated with discrimination index in an opposite way (Figure 3.31(A)). This was similarly observed in the behavior of DR single-cell analysis, and might hints a different behavior between DR cells of two strains. Secondly, only in B6J, the firing rate of DR synchronous pairs in training is positively correlated with
context freeze, while the altered firing rate is negatively correlated with discrimination index (Figure 3.31(B)). These are new correlations which we did not see at the single-cell analysis, which suggest that DR cells do participate in processing contextual memories. Thirdly, we found a new correlation in B6C between the firing rate of DR pair in context with context freeze. The other two are with altered freeze negatively and discrimination index positively.

Next we look at the activities of R containing pair assemblies. First, firing rate of R pairs in the context test is negatively correlated with context freeze in B6J. Second, their firing rate in context is negatively correlated with context freeze, and of altered is correlated with altered freeze positively and discrimination index negatively. This behavior is opposite to the R cells of B6C. For B6C, the firing rate of R containing pairs in context is associated with context freeze, altered freeze and discrimination index.
Figure 3.30: Pair ratios by cell types. DR cells in B6C actively participate into non-synchronous pairs. R cells in B6J actively participate into synchronous pairs. The brackets indicate significant t-test.

Figure 3.31: DR containing pair assembly activities correlation. (A) Correlation of DR containing pair. (B) Correlation of DR synchronous containing pairs. (C) Correlation of DR non-synchronous pairs.
Figure 3.32: R containing pair assembly activities correlation. (A) Correlation of R containing pair. (B) Correlation of R synchronous containing pairs. (C) Correlation of R non-synchronous pairs.
3.5.3. More specific classification for pair assemblies

We can be more specific and cluster the pairs by the type of the two participating neurons. We label them by the type of their cells (SS, RDR, etc...). By computing the p-values of having a pair type in a strain as we did above for cell containing pairs, even though there is not a significant p-values, we notice that DRC and SDR in B6C, and SS in B6J groups have the lowest p-values. Next we will analyze activities of each labeled group.

![Figure 3.33: Labeled pairs as ratios of significant pairs. We notice the differences between strains in DRC, RC, SDR and SR groups.](image)

3.5.3.1. Labeled pair activities

By associating each group’s firing rate with freezing behaviors, we first found that, in B6J, the firing rate of R-DR group in exploration is positively correlated with context freeze, and the firing rate in training is correlated with context freezing (+), altered freezing (+), and discrimination index (-) (Figure 3.35).

Furthermore, when we looked at the B6J DR-{C,S} pairs, that contain a DR neurons and either a C or a S neurons, we saw their firing rate in exploration is negatively associated with discrimination index.
Figure 3.34: mean p-values of having a labeled pair in a strain give a population of different cell types.

From B6C strain, we found correlations from R-R pairs. Particularly, their firing rate in exploration is positively correlated with discrimination index, and the firing rate in training is negatively correlated with context freeze. There is no analogous correlations at single cell analysis for B6C R neurons.
Figure 3.35: R-DR firing rate correlations in B6J. Firing rate in exploration is positively correlated with context freeze. Firing rate in training is positively correlated with context and altered freeze, and negatively correlated with discrimination index. *Red - B6J, Blue - B6C*.

Figure 3.36: DR-{C,S} firing rate correlation in B6J *Red*. Their firing rate in exploration is negatively correlated with discrimination index.
3.6. Assembly Analysis

In this section, the full cell assembly detection method (including pairwise correlation test, agglomerating scheme, and pruning process) is used to detect assemblies from all fully recorded animals. We notice that most of detected assemblies are pair assemblies, which shows that pairwise correlation capture most of the information. Similar to pair analysis, we classified the assemblies based on the cell type of assembly’s elements. We found that in B6C, given the profile of animals cell type, the probability of having an assembly containing DR cells is significantly higher than would be predicted by chance. Furthermore, there are some correlations between the firing rate of B6J DR containing assembly. Finally, we show that neurons participating in an assembly tend to stay closer to each other.

3.6.1. Detecting cell assemblies

Here we use the full cell assembly detection method which includes the extended pairwise correlation test, an agglomerate scheme, and a pruning process to keep assemblies with the smallest p-value. The method is applied on 18 fully recorded animals. For each animal, the spike train includes the training, context, and the first five minutes of altered sessions. The spike train is then converted to spike stamp, which is the time stamp when the spikes happen, to be compatible with the algorithm. This can be done since we know the sampling rate of all recordings, which is about 10 Hz. Finally, the value of parameters are as follows:

- **BinSizes** = [0.1, 0.25, 0.4, 0.65, 0.8, 1.5, 2.3, 2.8, 3.1]

- **MaxLags** = 10 - the maximum lag to find assembly

- **α** = 0.05 - uncorrected Type-I error

- The threshold for expected joint spike count is \( ExpThres = 1 \) with F distribution’s degree of freedom adjustment

After assemblies from all bin sizes were detected, a pruning step was applied to keep only the assemblies with the smallest p-values. Thus, for each animal, we have a population of
assemblies with their activities. Assembly activities are the spike counts at the frames where the co-spiking patterns happen.

By looking at the distribution of assembly size across all animals, we found that about 90 percent of assemblies are pair assemblies, and only 10 percent of assemblies have the size of at least 3 neurons (Figure).

![Figure 3.37: Assembly size distribution. About 90 percent of assemblies are pair assemblies.](image)

### 3.6.1.1. Some remarks about the pruning process:

As mentioned in [48], all the pruning steps suggested by the authors are only for the purpose of visualization. Some available options for the pruning process is the smallest p-value, the largest assemblies, or the highest pattern occurrence. Table 3.2 shows the number of detected assemblies before and after the pruning process. Doing so will result in losing information, especially activation time for those assemblies whose spiking patterns happens at different time scales, which is commonly observed in neuronal spike train data [9,12,38]. Therefore, a better pruning process which can capture assembly activities at different time scales is an interesting and feasible extension to this algorithm.
<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>After Pruning</th>
<th>Before Pruning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20160416-4B6J</td>
<td>19</td>
<td>99</td>
</tr>
<tr>
<td>2</td>
<td>20160613-2B6C</td>
<td>71</td>
<td>247</td>
</tr>
<tr>
<td>3</td>
<td>20160914-1B6J</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>20161012-2B6C</td>
<td>100</td>
<td>320</td>
</tr>
<tr>
<td>5</td>
<td>20161025-1B6J</td>
<td>55</td>
<td>220</td>
</tr>
<tr>
<td>6</td>
<td>20170120-2B6C</td>
<td>15</td>
<td>57</td>
</tr>
<tr>
<td>7</td>
<td>20170309-2B6C</td>
<td>16</td>
<td>52</td>
</tr>
<tr>
<td>8</td>
<td>20170317-3B6C</td>
<td>83</td>
<td>363</td>
</tr>
<tr>
<td>9</td>
<td>20170320-2B6C</td>
<td>38</td>
<td>139</td>
</tr>
<tr>
<td>10</td>
<td>20170411-2B6J</td>
<td>49</td>
<td>151</td>
</tr>
<tr>
<td>11</td>
<td>20170601-1B6J</td>
<td>42</td>
<td>128</td>
</tr>
<tr>
<td>12</td>
<td>20170613-2B6J</td>
<td>91</td>
<td>249</td>
</tr>
<tr>
<td>13</td>
<td>20170905-1B6J</td>
<td>198</td>
<td>469</td>
</tr>
<tr>
<td>14</td>
<td>20170928-2B6J</td>
<td>21</td>
<td>111</td>
</tr>
<tr>
<td>15</td>
<td>20180320-2B6J</td>
<td>31</td>
<td>143</td>
</tr>
<tr>
<td>16</td>
<td>20180425-1B6J</td>
<td>107</td>
<td>221</td>
</tr>
<tr>
<td>17</td>
<td>20180522-1B6C</td>
<td>115</td>
<td>661</td>
</tr>
<tr>
<td>18</td>
<td>20180523-1B6C</td>
<td>50</td>
<td>236</td>
</tr>
</tbody>
</table>

Table 3.2: Number of detected assemblies before and after pruning. The pruning process uses the smallest p-value option.
3.6.2. DR Assembly Enrichment in B6C Strain

Similar to pair analysis, we also found that DR cells in B6C are significantly more active in participating into assemblies.

Figure 3.38: Observed and approximated cell containing assembly ratio. DR cells in B6C are significantly more active in participating into assemblies

3.6.3. Correlations between Firing Rate and Freezing Behaviors

Here we show some findings about the correlations between the firing rate of specific cell type containing assemblies and freezing behaviors. Only in B6J animals, we found that the activities of DR containing assemblies in exploration and training are associated with freezing behaviors. Particularly, their firing rate in exploration has a positive trend with altered freezing and a negative trend with discrimination index, and their training firing rate is significantly correlated with those two behaviors. These correlations are what we saw in the labeled pair analysis (DR-{S,C} and DR-R pairs). Moreover, the correlation is stronger. This might indicate that the DR containing assemblies are more fine-tuned to freezing behaviors compared to DR containing pairs.
Figure 3.39: DR containing assemblies firing rate correlation.
3.6.4. Spatial coherency of cell assemblies

Another question we investigated is how the physical positions of cells participating in assemblies are different from what would be predicted by chance from the whole population. First, a distance metric for each assembly is defined as follows

\[ D = \frac{1}{N} \sum_{n=1}^{N} \| x_i - \bar{x} \|_F \]

where \( x_i \) is 2D-coordinate of unit \( i \) in the computed assembly, and \( \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \) is the centroid of the assembly. The distance metric is the mean of distance from all assembly units to the assembly centroid, representing the compactness of the assembly.

3.6.4.1. Generating Distribution of the Distance Metric

In order to determine whether the n-cell assembly are more spatially compact than randomly chosen groups of n cells, we generated 1,000 random n-cell groups from the true population of neurons and computed their distance metric \( D \). This empirical distribution is compared with the distribution of \( D \) from all assemblies.

Figure 3.40 shows that the distribution from assemblies is shifting to the left relatively from the simulated distribution, indicating that cell assemblies are more spatially coherent than we would predict by chance. The difference is confirmed by significant t-test with p-value = 1e-60. This result implies that neurons which are relatively closer in space have a higher chance to form an assembly. This evidence strengthen the hypothesis that those neurons that form a spiking pattern together might be internally connected.

3.7. Summary Results and Discussion

From the above sections, we show the methodology and findings from four different analyses: single cell, pair assembly from separate epochs, pair assembly from combined epochs, and full assembly from combined epochs. In this section, we summarize results from different analyses and discuss their implications. Overall, we found that (1) CA1 hippocampus
neurons, especially DR neurons, involve in processing contextual information, and the processing mechanisms are different between strains B6C and B6J, (2) The synchronous and non-synchronous pairs are both associated with freezing behaviors but in different ways, and (3) neurons participating in assembly are spatially coherent.

3.7.1. DR cell activities are associated with contextual memories

First, we found some correlations between DR neurons and freezing behaviors. For clarification, DR neurons are neurons that have a delayed response to at least one shock in the training session. At the single cell analysis, only in B6C, the firing rate of DR neurons in the exploration period is positively correlated with discrimination index (Figure 3.11). We could not find any association between neuronal activities and freezing behaviors in B6J.

However, from pair assembly analysis for combined epochs, we found that in B6J, the firing rate of DR containing pairs in training is positively correlated with context freezing (Figure 3.31). Moreover, with a more specific classification, we found that the firing rate
of R-DR pairs in exploration is positively correlated with context freeze, and the firing rate in training is negatively correlated with discrimination index (Figure 3.35). Also in B6J, the firing rate of DR-\{C,S\} in exploration is negatively correlated with discrimination index (Figure 3.36). The main point here is that, even though there are correlations between neuron activities in exploration and discrimination index from both strains, the correlation trends are opposite. This indicates that the mechanism for processing contextual information between two strains are different.

Another finding about DR neurons in B6C is that they are significantly more active in participating into pair assemblies, especially non-synchronous pairs (Figure 3.30). This fact is also observed at the full assembly level (Figure 3.38) in B6C. There is no enrichment of any cell type in assemblies for B6J.

Lastly, different level of analyses do provide different information about the data. For example, we could not find any correlations in B6J strain with the single cell analysis. Only with the pair/assembly analysis, we can actually associate the neuron activities with freezing and discrimination index.

3.7.2. Synchronous vs non-synchronous pairs

We also observe some special characteristics when splitting the pair population into synchronous and non-synchronous pairs. Firstly, from separate epoch pair analysis, there is a smaller number of pairs in exploration epoch, compared to other contexts. This observation was seen in [48] when the authors analyzed correlated pairs from CA1 population during the exploratory task and delayed alternation task.

Secondly, there are more non-synchronous pairs in novel contexts; i.e. exploration, training and altered context. Moreover, from combined epoch pair analysis, we also found that while the firing rate of synchronous pairs are comparable across epochs, the firing rate of non-synchronous pairs is significantly lower in the context test, and higher in other contexts. These indicate that non-synchronous pairs might associate with novel information.
Thirdly, there are some correlations between non-synchronous/synchronous pairs with freezing behaviors. In term of pair ratio, The non-synchronous pair ratio in context test is positively correlated with discrimination index, and the ratio of non-synchronous pair over significant pair is positively correlated with context freeze. We also found one non-synchronous NCPD histogram pattern in training whose expression level is positively correlated with discrimination index. From the combined epoch pair analysis, we also see that higher non-synchronous pair ratio is associated with higher context freezing. In term of pair activities, the firing rate of non-synchronous pairs in training is positively correlated with context freeze.

Lastly, synchronous pairs have stronger correlations with freezing episodes in both context test and in altered context, and that the correlation with freezing episodes is higher in B6J. This indicate that synchronous pairs also play a role in memory retrieval and context generalization.

3.7.3. Assembly spatial coherency

The last finding when analyzing the physical location of neurons is that the elements in an assembly is significantly spatially more coherent than we would predict by chance (Figure 3.40). This strengthens the hypothesis that the detected assemblies from spike train data might be actually neuroscience assemblies.
3.8. Appendix

3.8.1. Neuron classification algorithm

One question we want to address when observing the activity of a neuron around a shock is whether it is responsive to that shock. One way to answer this is to compare the difference in the firing rate of the recorded activity with the shuffled distribution. Here we describe the algorithm (more details in supplemental materials of [29]).

Suppose $X_b$ and $X_f$ are the spike trains of the neuron in a before-shock and after-shock time window, respectively. The difference in observed firing rate is defined by

$$D_{obs} = mean(X_f) - mean(X_b)$$

The shuffled distribution is computed as follows:
For each iteration out of 10,000 iterations
• The spike train $[X_b, X_f]$ is shuffled
• The difference between before-shock and after-shock firing rate is computed for the shuffled spike train.

After all shuffles, we will have the distribution of the firing rate difference. The difference $D_{obs}$ which at least equals the threshold $D_{th} = 95^{th}$ percentile of the simulated distribution, $D_{obs} \geq D_{th}$, is considered significant. It turns out that whether to include $95^{th}$ percentile in the significance region has an important effect on the output of the algorithm, especially when the total spike count is very small. This is explained by the next simulation.

3.8.1.1. Validity of classification scheme for small spike counts

Suppose we have the following scenario. The time windows for both before-shock and after-shock are 100 frames (this is the case where the R group is detected). The scenario in which other groups are identified can be validated similarly. Given a total number of spikes in both time windows, we generate the simulation of the difference number $D$, and calculate
the threshold at 95\textsuperscript{th} percentile and \(D_M\) the maximum value that \(D\) can take (Figure 3.41). Since the total spike counts of this interval from Xu Lab data are rather small (fluctuating around 20-30 spikes, see Figure 3.42), we only generate simulations for scenarios for total spikes from 1 to 40 spikes.

As shown in Figure 3.43, when the number of spikes is not greater than 4, the \(D_M\) equals to the 95\textsuperscript{th} percentile threshold of the simulated distribution. Therefore, if we do not include the 95\textsuperscript{th} percentile of the simulated distribution, those scenarios are never chosen, even though we believe them to be biologically significant. In these cases, there is no spike before shocks, and all the spikes are after-shock, which should be considered as shock responsive. Moreover, neuronal activity in hippocampus is believed to be sparse, therefore even a few after-shock spikes are meaningful. For those reasons, the 95\textsuperscript{th} percentile is included in the significance region of the algorithm.

3.8.1.2. The shuffled distribution is a Hypergeometric distribution

Here we show that the shuffled distribution of the firing rate difference \(D\) actually follows a hypergeometric distribution. Generally, suppose that the number of frames for before-shock and after-shock time window are \(n_b\) and \(n_f\), respectively. As before, the spike trains are \(X_b\) and \(X_f\). The probability of having \(k\) spikes in the before-shock time window is equivalent to the probability of having \(k\) successes in \(n_b\) draws, without replacement, from a finite population \(N = n_b + n_f\) that contains \(K = (\text{sum}(X_b) + \text{sum}(X_f))\) objects (or spikes in this case). This is exactly the hypergeometric distribution. Mathematically, if the total spike count before-shock is \(k\), then the total spike count after-shock is \(K - k\), which gives \(D = K - 2k\). The probability of this event is calculated as

\[
P(D = \{K - 2k\}) = P_X(k) \text{ where } X \sim \text{Hypergeometric}(N, K, n_b)
\]

In conclusion, we just showed that the shuffling process can be replaced by using a hypergeometric distribution which produces a more computationally efficient algorithm.
Figure 3.41: *top-left:* The observed calcium event, *bottom-left:* generating multiple shuffled spike train, *right:* Right-tail 95th percentile threshold for statistical significance.

Figure 3.42: Histogram of total spikes in 10s before- to after-shock from Xu Lab Animals.
Figure 3.43: (Black dot) Maximum difference value from the simulated distribution. Notice that due to small number of iterations, $D_M$ line is not linear as it should be - (Blue curve) 95$^{th}$ percentile threshold from the simulated distribution.
4.1. Introduction

To detect assemblies from a spike train population binned at a specific time scale, the original method of [48] includes three steps: a pairwise correlation test of two spike trains, an agglomeration scheme to merge significant pairs to larger assemblies, and a pruning process to select representative assemblies for assembly clusters. As mentioned in the original paper, the main purpose of having this pruning step is to provide better visualization and interpretability for the final output assemblies.

More specifically, the original pruning process has two options. The first option is to discard across multiple time scale any assembly which is a subset of another, so that only the largest assemblies are kept. Another option is to cluster all assemblies and choose the best assembly to represent each cluster. The distance measure used here for clustering is the cosine distance between the element vectors of the relevant assemblies. Assemblies in a cluster will have pairwise distance smaller than a certain threshold (this parameter is the user’s choice, and the default value in the algorithm is 0.3). Then the representative assembly of a cluster is the one with the smallest p-value, defined for the pairwise correlation test statistic in Chapter 2.

However, one would realize that these pruning step might cause missing information from discarding useful assemblies. This is certainly true if the goal of data analysis is to extract assemblies for which replay happens at multiple time scales. This interesting phenomenon has been observed from different experiments, and shown to encode important information. For example, in [38], the authors showed that the spiking patterns of a group of place cells are
replayed at a finer time scale, so called compressed replays. Other studies also showed that hippocampus replay for encoding behaviorally related information happens at a compress timescale \[9,12,24\]. Another example is that a spiking pattern can be replayed forward and reverse, as mentioned in \[2,10,12,37\]. These findings show that the reactivation of a cell assembly is not a temporally precise pattern. Any method designed to detect precise spiking pattern, such as ones mentioned in Chapter 1, will fail to capture relevant information for these assemblies. Therefore, a useful extension for the method would be a more appropriate pruning process that might help to extract more informative assemblies from neuronal spike train population.

One related class of method to detect imprecise replays from spike train data is the template-based (or template matching) method. The idea is that a firing template is created based on behavioral information and is used to compare with other activities. Given a firing template, these methods can detect the multiple time scale assemblies by using either (1) scaling factor analysis to quantify an imprecise match \[24,28\], or (2) rank-ordered correlation to only assess the firing order of assembly elements \[10,12,25\], or (3) Bayesian approach \[9,11\]. However, one limitation of these methods is that not all experiment has information to construct a firing template; fear conditioning data in Chapter 3 is an example.

In this chapter, we propose an algorithm to extract assemblies whose spiking patterns are replayed at different time scales, or \textit{multiple time scale assemblies (MTS)}. This development includes how to identify those multiple time scale assemblies from the pre-pruned assembly output, and how to collect the assemblies’ activities. The new pruning step is shown to outperform the original method on synthetic data sets.

4.2. New Pruning Step

The output of the CAD method, before pruning, can be summarized by an assignment matrix (Figure 4.1). Each column of the matrix shows the elements of one assembly. The color of each element shows its lag relative to the lag-zero element in the assembly. The color bar below the matrix shows the time scale at which the corresponding assembly is detected.
As shown in the example in Figure 4.1, there are 12 detected assemblies at different time scale, and many of them have the same set of units, i.e. there are two groups of units: unit 2-6, and unit 7-11. If we apply the original pruning step to this set of assemblies, we will get two assemblies as the output: assembly 3 for the group of unit 2-6, and assembly 4 for the group of unit 7-11. Other assemblies will be discarded and we lose information from other time scales. Therefore, here we suggest a new pruning step to identify multiple time scale assemblies and to collect their activities.

Figure 4.1: Identify multiple time scale assemblies. There are two MTS assemblies in this assignment matrix. The first assembly contains unit 2-6, the second one contains unit 7-11. They are detected at different time scales.

4.2.1. Identifying multiple time scale assemblies

Given a set of detected assemblies at different time scales, we can identify multiple time scale assemblies as follows:

- If an assembly is detected at multiple time scale, it can be flagged as a MTS assembly. Other assemblies whose elements are a subset of the assembly will be discarded.
- In a more complicated scenario, if multiple assemblies detected at different time scale have a common subset, that subset is also a MTS assembly.

After identifying the elements of an MTS assembly, we will keep track a list of the time scales it being detected and their elements’ lag. For example, the group of unit 2 to 6 is a MTS assembly since it is detected five times at different time scale, and the lag between elements is also different. Therefore, we will keep track on all five time scales and the lag information. These will be used to extract the assembly activities at those five different time scales.

4.2.2. Collecting assemblies activities

The activities of a MTS assembly can be extracted by following the list of the time scales and its lags. Particularly, we will extract activities from the larger time scales first, and only extract additional activities from the smaller time scales, to avoid double counting.

Notice that this pruning step is only to identify MTS assembly. For getting a full set of assemblies from a spike train population, we suggest using the original pruning process first, and then replace appropriate assemblies with its multiple time scale version. However, depending on the goal of data analysis, the set of MTS assemblies alone might already provide interesting information about the neuronal population.

4.3. Synthetic Spike Train with Multiscale Assemblies

4.3.1. Generating Spike Train Population

We first test the performance of the new pruning process on synthetic spike train data. The generative model is adapted from [48]. The synthetic spike train population is generated by an inhomogeneous Poisson process, in other words, inter-spike-intervals are drawn from an exponential distribution.
The firing rate $\lambda_{it}$ of neuron $i$ at time $t$ was governed by an underlying stable first-order autoregressive process. The coefficient matrix $D$ is constructed such that the autoregressive process is stationary, i.e. the spectral radium of $D$ is less than 1.

$$s_{t+1} = Ds_t + \epsilon_t, \quad \text{where } \epsilon_t \sim \mathcal{N}(0, \sigma_s^2 I)$$

The firing rate $\lambda_t$ is generated by pushing $s_t$ through a sigmoid nonlinearity

$$\lambda_t = \left(1 + erf \left(\frac{s_t - \bar{s}}{\sigma_s}\right)\right) \bar{\lambda}$$

where $erf$ is the error function, and $\bar{\lambda}$ is a constant mean vector. Finally, a constant delay $\tau$ is added to each inter-spike-interval as a refractory period. Finally, parameters value are $D = 0.9I$, $\bar{\lambda} = 5$, $\tau = 0.15$, $v = 0.2$, and $\sigma_s = 0.01$

4.3.2. Embedding Spiking Pattern

The next step is to embed multiscale assemblies into the synthetic spike train. Particularly, two spiking patterns with different time scales (1 second (blue pattern) and 0.2 second (green pattern)) are embedded (Figure 4.2-middle plot). For clarity, the time scale of a spiking pattern is the length of an activation of the pattern. Then we generate datasets for two scenarios: (1) two block scenario in which each block of data contains one type of spiking patterns, and (2) alternating block scenarios in which each spiking pattern has two blocks and they are alternatingly embedded into the data. Moreover, the datasets have 10 units, and only the first 5 units have spiking patterns.

4.4. Preliminary Result on Synthetic Data

The first and second steps of the CAD algorithm is applied to the binned spike train at multiple bin width. The pairwise correlation test is the extended one in Chapter 2. However, as the firing rate is rather high in these scenarios, it is not different from using the original correlation test. The tested bin widths are $[0.015 0.025 0.04 0.06 0.085 0.15 0.25 0.4 0.6 0.85 1]$ seconds.
The set of extracted assemblies before the pruning step is shown as an assignment matrix in Figure 4.3. The algorithm successfully extracts multiple assemblies from the first five units as expected. Furthermore, the outputting assembly from the original pruning step indicates that the first five units form an assembly, and by the construction of the method, the fourth assembly (the fourth column in the assignment matrix) is the one being selected. As a result, the extracted assembly activation time is only from the fast timescale (green) spiking pattern. The information extracted from the original algorithm totally misses the activities from the slow timescale pattern (Activity from Original Pruning in Figure 4.3).

On the other hand, the proposed pruning step flags the first five units as an multiple time scale assembly. Specifically, the information from the assembly number 4,5,6,7,8 and 11 will be used to extract the assembly’s activities (or activation time). Consequently, the algorithm can fully recover the activities from both fast and slow timescales, as shown in Activity from Multiscale Pruning in Figure 4.3.
Figure 4.3: Performance comparison between the original and the proposed pruning steps. The proposed pruning step can capture activities from two time scales, while the original one can only extract the activities from the faster time scale.
4.5. Discussion

The result on synthetic datasets shows that the proposed pruning step can help to extract information that otherwise will be discarded by the original pruning process. Certainly how much useful this new step is depends on the goal of data analysis. As there are evidences from neuroscience experiments showing that assemblies can experience replays at different time scales, this tool might become handy for detecting those assemblies.

For limitations of this study, we think that these two testing scenarios are rather simple, and that in the real spike train data, assembly activities might be more complicated and dynamic. Another limitation is that by using this algorithm to detect cell assemblies, the model assumes that the assembly’s activities are already significant at each time scale. Another scenario can be envisioned in which the assembly activities are not a rare event at any time scale alone, but when combined, the co-activities from different time scales actually show a significant correlation between assembly elements. Addressing this possibility requires constructing a test statistic which accounts for events from different time scales.

For future work, we will evaluate the performance of the method with more complex multiscale variety of the spiking patterns. Secondly, this method needs to be tested on a real datasets to prove its efficacy. For example, place cells are known to form a sequential spikes while the animal moves on a track, but those sequential spikes are compressed to a finer time resolution when they are planning their path \[38\]. The algorithm can be used on this data to detect those events.
Chapter 5

Conclusion

5.1. Summary

Advanced recording techniques which can monitor activities of hundreds of neurons are creating opportunities to observe and understand how the brain functions. At the same time, the complex high dimensional data requires more advanced workflow to process the data and extract meaningful information from it. Particularly, by observing activities from neurons simultaneously, the neuroscientists now can pay attention to how the neurons coordinate their activities into assemblies to perform certain cognitive processes. However, the task of detecting assemblies from high dimensional spike train data has been challenging because of various complex patterns those assemblies can form. As a contribution to the broad active study in cell assemblies, this thesis proposes a method appropriate to low firing-rate data (Chapter 2), and another method to detect assemblies whose replays happen at different timescales (Chapter 4). The method is then applied to calcium imaging data from fear conditioning project to study how the hippocampus involves in forming, retrieving, and generalizing contextual information (Chapter 3).

The pairwise correlation test for two spike trains in [48] originally is not applicable if their co-firing rate is too low. In chapter 2, we showed that the test validity can be maintained in the region of low co-firing rate. We did this by first using a power law distribution to quantify to error of the test, and then proposing a better approximating distribution for the test statistic. This is shown to produce a lower error compared to the original method. In term of performance on the real-world data sets, we show that the new method is able to
test more pairs of spike train, and detect more correlated pairs. The implication is that more relevant information are extracted from the data.

The analysis of fear conditioning data is shown in Chapter 3. We found that there exists a special group of neurons (DR group), which experience delayed response to the training shocks. In fact, this group is associated with the animals’ behaviors in both strain of mice. Moreover, since the correlations between DR group and freezing behaviors are different between two strains, the mechanisms for processing information of hippocampus CA1 are different. In B6J mice, even though we could not find any correlation in the single cell analysis, we did find some with the help of cell assembly detection method. Moreover, we found that synchronous and non-synchronous pair assemblies are associated with the behaviors in different ways. Lastly, from the full assembly analysis, we found that neurons participating into an assembly tend to stay closer than we would predict by chance. Overall, all findings show that analyzing cell assemblies is an useful approach to understand neuronal activities.

In Chapter 4, inspired by some observations in neuroscience about cell assemblies whose replays happen at different time scales, we proposed an algorithm to automate the process of detecting such assemblies. We showed that the proposed method outperforms the original method on two synthetic datasets. As another future work, the algorithm needs to be assesses for its performance on real-world data. Furthermore, it also needs to be developed into a full scheme to detect both uni- and multi- timescale assemblies. This will provide a complete and easy-to-implement tool for neuroscientists.

5.2. Future Direction

Given that non-synchronous assemblies participate in processing contextual information in Chapter 3. I think that developing tools to detect such assemblies will be proved useful in the future. However, they are challenging as non-synchronous patterns are complex and varied. In fact, there are still very few methods which are able to detect non-synchronous/temporal patterns. As I learn from this study, all we need to build a such
method is (1) to have a measure for co-activities between two spike trains, and this measure needs to be capable of capturing the lagged coincidence, and (2) the distribution of that measure is available either by statistical based construction or by shuffling simulations. Other spike distances might be good candidates are Victor-Purpura distance [59], edit similarity distance and precise temporal correlation [53].


