Investigating Information Processing Within The Brain Using Multi-Electrode Array (MEA) Electrophysiology Data

Paul Michael Horton

Submitted for the degree of DPhil

University of Sussex

September 2010

Declaration

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Signature:

Acknowledgements

I would like to thank, firstly, my supervisor Professor Jianfeng Feng for his support and direction throughout my studies.

Secondly, I would like to thank Dr Andrew Philippides for his help and support especially for reading through and providing invaluable comments on improving this thesis.

Thirdly, I would like to thank Professor Keith Kendrick and Dr Ali Nicol from the Babraham Institute for their help and support especially for providing me with the biological data used in this thesis.

Finally, I would like to thank my family, especially my parents Jean and Mick, my girlfriend Laura Hughes, my sister Julie Owen, my brother in law Nigel, my nephew and niece Max and Chloe, and my friends Greg Wilson, Daniel Earl, Robert Earl and Lesley Warren for their encouragement and support. In addition, I would like to thank Dr Simon Durrant for being a good friend throughout my time studying for the DPhil.

University of Sussex

Paul Michael Horton

Submitted for the Degree of DPhil

Investigating Information Processing Within The Brain Using Multi-Electrode Array (MEA) Electrophysiology Data

Summary

How a stimulus, such as an odour, is represented in the brain is one of the main questions in neuroscience. It is becoming clearer that information is encoded by a population of neurons, but, how the spiking activity of a population of neurons conveys this information is unknown. Several population coding hypotheses have formulated over the years, and therefore, to obtain a more definitive answer as to how a population of neurons represents stimulus information we need to test, i.e. support or falsify, each of the hypotheses. One way of addressing these hypotheses is to record and analyse the activity of multiple individual neurons from the brain of a test subject when a stimulus is, and is not, presented. With the advent of multi electrode arrays (MEA) we can now record such activity. However, before we can investigate/test the population coding hypotheses using such recordings, we need to determine the number of neurons recorded by the MEA and their spiking activity, after spike detection, using an automatic spike sorting algorithm (we refer to the spiking activity of the neurons extracted from the MEA recordings as MEA sorted data). While there are many automatic spike sorting methods available, they have limitations. In addition, we are lacking methods to test/investigate the population coding hypotheses in detail using the MEA sorted data. That is, methods that show whether neurons respond in a hypothesised way and, if they do, shows how the stimulus is represented within the recorded area. Thus, in this thesis, we were motivated to, firstly, develop a new automatic spike sorting method, which avoids the limitations of other methods. We validated our method using simulated and biological data. In addition, we found our method can perform better than other standard methods. We next focused on the population rate coding hypothesis (i.e. the hypothesis that information is conveyed in the number of spikes fired by a population of neurons within a relevant time period). More specifically, we developed a method for testing/investigating the population rate coding hypothesis using the MEA sorted data. That is, a method that uses the multi variate analysis of variance (MANOVA) test, where we modified its output, to show the most responsive subareas within the recorded area. We validated this using simulated and biological data. Finally, we investigated whether noise correlation between neurons (i.e. correlations in the trial to trial variability of the response of neurons to the same stimulus) in a rat's olfactory bulb can affect the amount of information a population rate code conveys about a set of stimuli. We found that noise correlation between neurons was predominately positive, which, ultimately, reduced the amount of information a population containing >45 neurons could convey about the stimuli by ∼30%.

Contents

List of Figures

Chapter 1

Introduction

After over a century of neurophysiological research we still do not understand the principle by which a stimulus, such as an odour, an image or a sound is represented by neurons in the brain (Albright et al., 2000; Dayan et al., 2001; Feng, 2004). It is generally accepted that:

1. neurons are the basic elements that process information. There are many neurons in the brain and each are connected to several others. For example, in the human brain there are 10^{12} neurons and each neuron is connected with approximately a thousand others (Braitenberg et al., 1998).

2. within the connected neuronal network, information is transmitted from one neuron to another through action potentials, the so called spikes (Adrian, 1928; Hodgkin et al., 1939).

However, how stimulus information is represented by these neuronal networks in the brain is still the subject of research.

1.1 The Neural Code

To investigate how neurons represent stimulus information in the brain, neurophysiologists would typically record the spiking activity of a neuron in the brain of a test subject, when a stimulus is, and is not, presented. They would then examine the relationship between the stimulus and the spiking activity of the neuron recorded, i.e. they would analyse the difference between the spiking activity of the neuron when

the stimulus was, and was not, presented. The first investigations made detailed analyses of response profiles of single cells in isolation (e.g. Adrian 1926; Baker et al., 1981; Britten et al., 1992; Hubel et al., 1959; Logothetis et al., 1996; Maunsell et al., 1983; Perett et al., 1982; Perez et al., 1992; Rolls 1984; Yamane 1988). These studies showed that neurons changed their frequency of spikes (commonly known as the spiking rate) in the presence of external sensory stimuli. For example, (Baker et al., 1981; Britten et al., 1992) showed that the spiking rates of neurons in the visual area (V5) of a monkey brain increased with specific motion directions. Other studies showed that neurons fired precisely timed spikes after stimulus onset. For example, (Buracas et al., 1998) showed that neurons in the cortical area MT fired particular spike sequences (i.e. fired spikes at exactly the same times after stimulus onset on different trials) in response to a changing visual motion (velocity) stimulus. These results have led to several hypotheses. The main ones are:

1. the single cell coding hypothesis, which assumes that a neuron encodes all of the information about a specific stimulus and each neuron encodes information about a stimulus different from the other neurons (Barlow 1972; Desimone et al., 1984; Gross 1994; Gross 2002; Hahnloser et al., 2002; Kreiman et al., 2000a; Kreiman et al., 2000b; Kreiman et al., 2002). In other words, this hypothesis assumes that a single neuron responds to a specific stimulus, and varies its response according to the stimulus information presented. The main hypotheses as to how a single neuron conveys, i.e. transmits, information are:

   (a) the rate coding hypothesis, which assumes that information is carried in the spiking rate of a neuron, i.e. in the number of spikes they fire within a relevant time window (Adrian 1928; Baker and Lemon, 2000; Barlow, 1972; Bialek et al., 1991; Bialek and Rieke, 1992; Britten et al., 1992; Brody, 1999; Desimone et al., 1984; Oram et al., 1999; Shadlen et al., 1994; Shadlen et al., 1998; Rieke et al., 1997), and/or.

   (b) the temporal coding hypothesis, which assumes that information is carried in the spike sequence a neuron produces, i.e. in the precise times they fire spikes (Bair 1999; Bhandawat et al., 2007; Butts et al., 2007; Cariani 2001; Carr 1986; Eggermont 1990; Kreiman 2004; Mainen et al., 1995; Lestienne 2001; Rieke et al., 1999).

2. the population coding hypothesis, which assumes that a neuronal population encodes all the information about a stimulus and that each neuron encodes information about more than one stimulus (Averbeck et al., 2006; Barlow et al., 1964; Rolls et al,. 1997). In other words, this hypothesis assumes that many neurons respond to a stimulus and the response of each neuron varies according to the stimulus information presented. The main hypotheses as to how a neuronal population transmits information are:

   (a) the population rate coding hypothesis, which assumes that information is carried in the spiking rates of each neuron in an ensemble (Averbeck et al., 2006; Barlow et al., 1964; Georgopoulos et al., 1986; Hubel et al., 1959; Maunsell et al., 1983). In addition, the capacity of a population rate code, i.e. how much information the code conveys, is thought to be dependent on the relationships (correlations) between the spiking rates of the neurons in the ensemble. More specifically, as:

      i. the brain is noisy, (i.e. a neuron's response to the same stimulus on different experiments (trials) tends to vary) (Stein et al., 2005) and

      ii. noise is predominately positively correlated (i.e. the spiking rates of neurons, generally, increase and decrease together in response to the same stimulus on different trials[1]).

      it is thought that noise correlation may increase or decrease the capacity of a population rate code (Averbeck et al., 2006; Quian Quiroga et al., 2009). That is, noise correlation may determine whether neurons contribute independent, synergistic or redundant information to the ensemble.

   (b) the population temporal coding hypothesis, which assumes that information is carried in the precise times each neuron in an ensemble fires spikes, and/or, in the relationships (correlations) between the precise times multiple neurons in an ensemble fire spikes (Abeles 1991; Bair 1999; Hebb 1949; Gray et al., 1995; Krahe et al., 2002; Lestienne 2001; Panzeri et al.,

[1]Many empirical studies have shown that, in the brain, the noise correlation coefficient is ~ 0.2. For examples, see (Averbeck et al., 2006; Kohn et al., 2005; Reich et al., 2001).

2001b; Reich et al., 2001b). For example, information may be contained in the sychronised firing of two or more neurons and/or in the timing of spikes with respect to ongoing brain oscillations (Singer 1993).

The most appealing characteristic of the single-cell coding hypothesis is its simplicity. However, the most appealing characteristics of the population coding hypotheses is that they overcome several limitations of the single-cell coding hypothesis (Buzsaki 2004; Brown et al., 2004; Pouget et al., 2000). More specifically:

1. a population code has greater coding potential. That is, a population encodes stimuli combinatorially allowing a significantly larger number of stimuli to be represented. For example, if the brain used 500 binary neurons, the brain could represent 2^{500} faces. However, if the brain used single cell coding, the brain could only represent 500 faces, i.e. one face per neuron.

2. a population code is more robust to dying neurons. That is, if many neurons encode information about a particular stimulus item, a neuron lost would not greatly affect the representation of the stimulus. For example, if one neuron died out of the 500 neurons that represented a person's grandmother, the majority of the representation is still there, and therefore, the person could still recognise their grandmother.

3. a population code has the potential of processing information faster than a single cell. For example, if 20 spikes signal the presence of the colour red, it would be faster for 20 neurons to fire 1 spike at the same time than a neuron firing 20 consecutive spikes.

4. a population code has the potential of representing stimulus information more accurately (i.e. provide more information) in a noisy environment than a single cell (Averbeck et al., 2006; Ermentrout et al., 2008; Faisal et al., 2008). In other words, as the response of a neuron is typically noisy, as discussed earlier, it may be difficult for the brain to determine what information one neuron is conveying. However, the uncertainty about the information being conveyed due to neuronal variability can reduce, i.e. the accuracy of the stimulus information can increase, with the number of neurons pooled.

1.2 Determining The Population Code

Due to the appealing points of the population coding hypotheses, studies are now shifting from analysing/recording the responses of a single neuron to a stimulus, (i.e. analysing the changes in the spiking activity of a single neuron when a stimulus was, and was not, presented) to, analysing/recording the responses of multiple neurons (i.e. analysing the changes in the spiking activity of multiple neurons when a stimulus was, and was not, presented). Investigations into how populations of neurons convey (transmit) stimulus information are being performed at:

1. the single cell level of the brain by recording and analysing the spiking activity of multiple single neurons (e.g. Rolls et al., 1997). We note that this type of investigation does not involve recording the simultaneous spiking activity of multiple neurons.

2. the gross structural level of the brain by:

   (a) using brain imaging techniques, which are indirect markers of brain electrical activity, including:

      i. single photon emission computed tomography (SPECT) (Amen et al., 2000) and functional magnetic resonance imaging (fMRI) (Belliveau et al., 1991), which record changes in blood flow.

      ii. Positron emission tomography (PET) (Young et al., 1999), which record changes in metabolic activity and would show the area(s) of dominant brain activity.

   (b) recording and analysing the integrated spiking activity, i.e. the extracellular field potential, of a number of neurons in the brain. The extracellular field potential can be recorded on a small scale (i.e. from a few neurons), which is referred to as the local field potential, or, on a much larger scale (e.g. from millions of neurons) using electroencephalography (EEG) (Creutzfeldt et al 1966; Niedermeyer et al., 2004; Nunez et al., 1981) or magnetoencephalography (MEG) (Hamalainen et al., 1993) techniques. Such recordings can be used to determine whether information is conveyed in the synchronized activity of neurons, i.e. in the frequency,

amplitude and phase of neural oscillations (Buszaki et al., 2006; Fries et al., 2001; Huxter et al., 2003; Juergens et al 1999; Mitzdorf et al 1987; Quian Quiroga et al., 2009; Rutishauser et al., 2010; Siapas et al., 1998; Sirota et al., 2003).

However, even though some progress towards understanding how populations of neurons represent stimulus information is being achieved using the above techniques, they have a few limitations. Firstly, it is not easy to address, using the data recorded at the single cell level, the holistic issues of how large ensembles of neurons can integrate information both spatially and temporally (Buszaki et al., 2004). Secondly, it is not easy to determine, using the data recorded at the gross structural level, the responses of individual neurons or the information individual neurons convey in an ensemble (Buszaki et al., 2004). Furthermore, brain imaging techniques may fail to capture the temporal dynamics of the brain activity (Buszaki et al., 2004; Legatt et al., 1980).

A promising approach to provide an understanding of how a population of neurons conveys stimulus information at the level of multiple cell-cell interactions is to record and analyse the simultaneous output (action potentials, spikes) from large numbers of individual cells within a defined system (brain area). This type of study has been difficult to achieve due to the lack of appropriate hardware to simultaneously record the spiking (electrical) activity of multiple individual neurons. However, a small number of groups have now developed the necessary recording hardware, i.e. the multi-electrode array, and software tools for data acquisition. These are available commercially (e.g. Plexon Inc. and Cyberkinetics Inc., USA) and have been applied to an expanding number of studies of neural activity, in both perceptual (Maynard et al., 1999) and cognitive (Johnson et al., 2003) systems, and also in the development of implantable neuroprosthetic devices (Patterson et al., 2004; Warwick et al., 2003). However, before we can interrogate the spiking activity of the neurons recorded by the electrodes, we need to perform three steps. Firstly, we need to extract the part of the electrophysiological signal recorded by each electrode that contains the spiking activity of the individual neurons, which we refer to as the spike signal. We then need to detect and extract the spikes from the spike signal using a spike detection algorithm. We then need to determine the number of neurons recorded

and the spikes they fired from the spikes extracted using an automatic spike sorting algorithm. Thus, once a number of neurons and their spiking activity is identified from the MEA recordings (we refer to the extracted spiking activity of individual neurons as MEA sorted data) we can interrogate it. In other words, we can use it to test/investigate the population coding hypotheses.

1.3 Aims And Outcomes Of This Thesis

There is little doubt that much of the information processing power of the brain resides in the activities of co-operating and competing networks of neurons (Brown et al., 2004; Buzsaki, 2004; Quian Quiroga et al., 2009). Thus, to achieve a more definitive answer as to how information is represented in the brain we need to test/investigate, i.e. support or reject, each of the population coding hypotheses. In other words, we need to determine whether neurons respond to stimuli in the hypothesised ways, what, and how much, information such responses convey about the stimuli, and address any questions related to the hypotheses. In addition, while there are many automatic spike sorting methods available that can be used with multi-electrode array (MEA) recordings, they have limitations (we discuss these in chapter 2). In addition, we are lacking methods to test/investigate the population coding hypotheses in detail using the MEA sorted data. That is, methods that show whether neurons respond in a hypothesised way and, if they do, shows how the stimulus is represented within the recorded area. Thus, in this thesis, we:

1. were motivated to develop a new automatic spike sorting method. Our method consists of using machine learning algorithms, which we extended, and, avoids the limitations of other methods (we discuss this in more detail in chapter 2). We show that our method is capable of extracting the correct number of neurons and their spiking activity from biological and simulated datasets, and, can perform similarly, and in some cases better than, other standard automatic methods. We note that this algorithm was used in the work of (Nicol et al., 2005; Tate et al., 2006). We discuss our method and results in chapter 5.

2. focus on one of the main population coding hypotheses: population rate coding. That is, we:

(a) were motivated to develop a method for testing/investigating the population rate coding hypothesis using MEA sorted data. More specifically, a method that, from MEA sorted data recorded before and during stimulus presentation, provides a detailed statistical analysis of the stimulus evoked changes in the spiking rates of the neurons, and shows, the responsive, and most responsive, subareas of neurons (i.e. the subareas exhibiting the most significant change in spiking rates) within the MEA recorded area. We refer to these subareas as hot spots.

We require such methods so we can easily address questions related to the hypothesis such as:

i. does the whole recorded area respond to the stimulus, or, does one or many small subareas respond. In other words, is the stimulus represented spatially in the responses of small groups of neurons, or, is it represented in the responses of a large group of neurons.

ii. do neurons respond to a stimulus with the same or different spiking rates.

Our method consists of using a multi-variate analysis of variance (MANOVA) test, where we modified its output, to show the hot spots within the recorded area. We show that our method can identify hot spots within a neuronal area using simulated and biological data. We note that this method was used in the work of (Nicol et al., 2005). We discuss our method and results in chapter 6.

(b) investigated whether relationships (i.e. noise correlation) between the spiking rates of neurons could affect (i.e. increase or decrease) the amount of information a population rate code conveys. We addressed this question using the responses of 236 neurons recorded from a rat's olfactory bulb when a set of stimuli was presented. We found that noise correlation can affect the amount of information a population rate code conveys, where the effect is dependent on the tuning similarity of the neurons, and the number of neurons pooled. In addition, we found that noise correlation, ultimately, reduced the amount of information a population containing > 45 neurons could convey about the stimuli by $\sim 30\%$. We

discuss our investigation and results in chapter 7.

1.4 Sections Of The Thesis

The thesis is divided into the following sections:

- In chapter 2-4 we extend our introduction, i.e. we discuss our aims in more detail and other work that has addressed them. More specifically:

    1. in chapter 2 we introduce the neuron, how they generate spikes, and, the multi-electrode array (MEA). We also introduce the problems MEA electrophysiological recordings present, and, discuss in detail, other solutions to the spike sorting problem, their limitations, and introduce our method.
    2. in chapter 3 we introduce the rat olfactory bulb. We introduce this as we tested our methods and investigated the population rate coding hypothesis using spiking data recorded from a rat's olfactory bulb.
    3. in chapter 4 we briefly describe the methods that can be used to investigate/test the population rate coding hypothesis using MEA sorted data. We then introduce our statistical analysis tool. Finally, we discuss studies that have investigated the effects of noise correlation on the amount of information population rate codes convey. We then briefly discuss our investigation.

- In chapter 5 we discuss our proposed automatic spike sorting method.

- In chapter 6 we discuss our proposed method for investigating/testing the population rate coding hypothesis using MEA sorted data, i.e. detecting hot spots within a recorded area.

- In chapter 7 we address the question: can noise correlation between neurons affect the amount of information a population rate code conveys, using the spiking activity of the multiple individual neurons recorded by a multi-electrode array from a rat's olfactory bulb.

- In chapter 8 we discuss all of our results.

Chapter 2

Introduction Part 2: Neuronal Signals and Multi-Electrode Array Electrophysiological Recordings

Multi-electrode arrays (MEA) are currently one of the best tools for recording the simultaneous spiking activity of multiple single cells. However, before we can interrogate the spiking activity of the neurons recorded we need to perform three steps. Firstly, we need to distinguish between the two components that comprise the electrophysiological signal recorded by each electrode on the array. More specifically, we need to extract the part of the signal containing the spiking activity of the individual cells, which we refer to as the spike signal. The other part of the signal contains the integrated spiking activity of the neurons, i.e. the local field potential ('LFP'). Secondly, the number of neurons and their spiking activity contained in the spike signal need to be identified. This is a two step process. Firstly, we need to detect and extract the spikes (spike shapes) from the spike signal. This is referred to as the spike-detection stage. We then need to determine the number of neurons recorded and their spiking activity from the spike shapes. This is referred to as the spike sorting stage.

In this chapter we discuss the structure of the neuron, how they generate spikes and the above stages in more detail. We also discuss current spike sorting methods, their limitations, and we introduce our spike sorting method.

2.1 Properties Of Neurons

A neuron is comprised of three components, the dendrites, the cell body (soma), and the axon. We show this, i.e. the structure of a typical neuron, in fig. 2.1. A spike is transmitted via the axon, and a neuron receives spikes from other neurons via the dendrites. The axon of a neuron terminates at the synapse, i.e. the junction between two neurons. A spike (action potential) is generated when the cell's membrane potential (i.e. the difference in the voltage between the cell's interior and exterior) is sufficiently raised above a threshold level from its resting potential (which is usually in the range of -70mV to -80mV). The membrane potential of a cell arises due to the ion concentrations on the intracellular and extracellular sides of the membrane, and changes when the ion channels embedded in the membrane open and close. That is, the membrane potential increases (which is referred to as depolarisation) when the channels allow positively charged ions into the cell, such as sodium (Na^+) and calcium (Ca^{2+}). In addition, the membrane potential decreases (which is referred to as hyperpolarisation) when the channels allow into the cell negatively charged ions such as chloride (Cl^-) and/or allow out positively charged ions such as potassium (K^+). The channels open and close in response to voltage changes and to both internal and external signals. For example, the voltage transient of an action potential generated by a pre synaptic neuron opens the ion channels of the postsynaptic neuron. An action potential is roughly a 100mV fluctuation in the electrical potential across the cell's membrane that lasts about 1ms (Kandel et al., 1991).

2.2 Recording Neuronal Activity Using Multi Electrode Arrays (MEA)

Multi-electrode arrays are extracellular recording devices. In other words, extracellular recording devices record electrical activity (action potentials) generated by neurons adjacent to the electrode tip[1]. A multi-electrode array (MEA) records an

[1]Electrical activity generated by neurons can be recorded using intracellular recording devices. These devices measure the voltage across the membrane of a cell and require cells to be impaled by an electrode. A disadvantage of recording spiking activity in this manner is that the recording

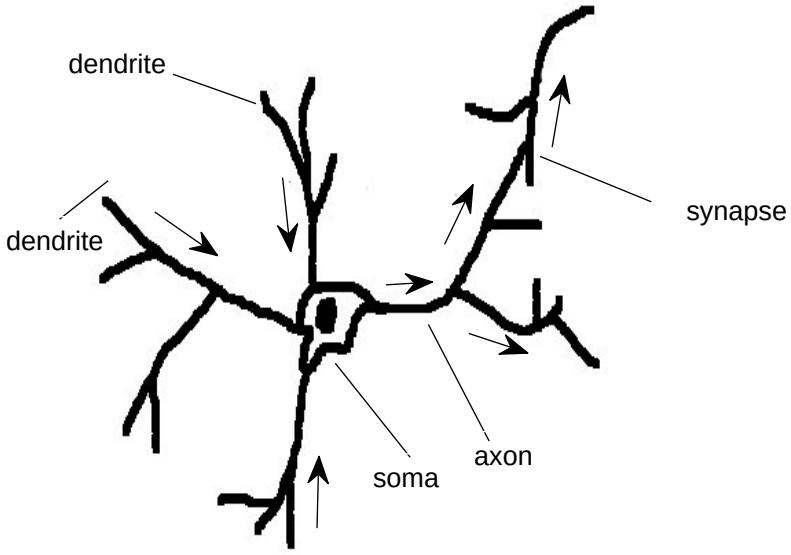


Figure 2.1: A simplified picture of a neuron, showing a neuron's main components: axon, soma and dendrites. The picture also shows the synapse, i.e. the junction, between two neurons.

area of neural activity by using several electrodes, where the electrodes are organised in a grid like manner (see fig. 2.2)[2]. Each electrode, therefore, records the spiking activity of a subset of the whole neuronal area. The electrodes are sufficiently spaced out on the MEA (typically by $350\mu m$, see fig. 2.2) so that each records the activity of a different group of neurons. Typically, electrodes on an MEA have an impedance in the range of 30-400 kΩ. Furthermore, the diameter of the electrodes are typically in the range of 3 to 10 μm (Harris et al., 2000) so that they record the activities of individual neurons. That is, if the electrode tips are too large, the activity of individual neurons cannot be distinguished. However, larger electrode tips can be used to record the local field potential (i.e. the integrated activity of the surrounding neurons).

Thus, by using an MEA to record a neural area when a stimulus is presented, we can determine whether the whole area or subsets of the area responded to the stimulus presentation.

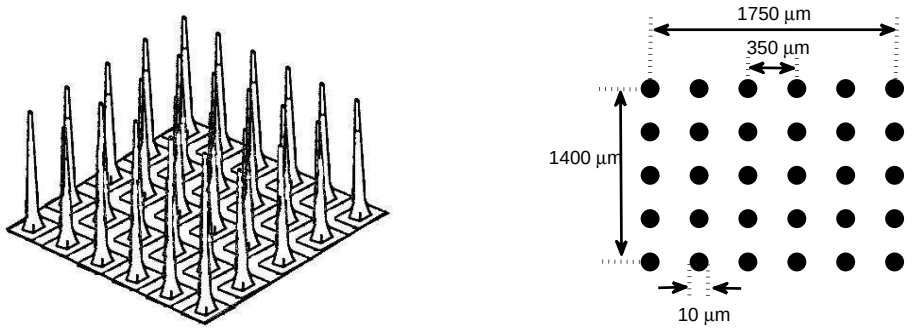
The dataset created by an MEA can be very large. As most individual experimental sessions last several hours or more, MEAs can produce datasets of hundreds of GBytes.

2.3 Local Field Potential And Multi-Unit Activity

The spiking activity of individual neurons and the local field potential (LFP) can be recorded by the same electrode. Therefore, the two signals must be distinguished. The spiking activity of individual neurons is a very fast signal on a timescale of one millisecond with signal components that range up to a few kHz, and, LFP is a slow signal with signal components between 1 and 300 Hz. Thus, the slower and faster parts of the signal can be extracted from the recorded signal using two band

device may damage the cell, and therefore, affect the neuron's true spiking activity.

[2]We note that a reference electrode is also placed in the brain, away from the neuronal area of interest. The recording from the reference electrode is used to amplify the signal, and minimise the noise, recorded by the electrodes on the MEA. That is, the amplified signal is the signal difference between the reference electrode, and the recording electrode, amplified. Thus, the amplification process amplifies the signal but minimises the amplification of the noise contained in the recordings.



(a) (b)

Figure 2.2: **Panel (a):** Shows a multi-electrode array (MEA). **Panel (b):** shows a schematic of a typical MEA. The schematic shows the size of an MEA, the distance between, and the size of, the electrodes.

pass filters. The part of the signal containing the spikes from individual neurons, which we refer to as the spike signal, can be extracted using a band pass filter with a bandwidth between 300 Hz and 5 kHz. We note the LFP signal can be extracted using a band pass filter with a bandwidth between 1 Hz and 300 Hz.

In the rest of this chapter we discuss issues related to analysing the spiking activity of individual neurons, and therefore, we will not further discuss issues related to analysing the local field potential (LFP). However, if the reader is interested in LFPs we refer the reader to (Brown 2004; Busaki et al., 2004).

2.4 Spike Detection And Spike Sorting

The spike signal extracted from the electrode recordings represent the simultaneous electrical (spiking) activity of an unknown number of neurons. Therefore, before we can interrogate the spiking activity of the neurons recorded, we need to determine the number of neurons recorded and the spikes they fired from the spike signal. This process is performed in 2 steps. The first step is called spike detection and the second step is called spike sorting.

2.4.1 Spike Detection

Spike detection refers to the process of identifying the spike events or action potentials from the voltage trace (spike signal) recorded by an electrode. This is typically achieved by setting upper and lower thresholds, thus, if the amplitude of the signal is in between the two thresholds it is considered noise. Noise can refer to the background firing of other neurons or fluctuations induced artificially by external electrical equipment etc. If the amplitude of the signal recorded is higher than the upper threshold or lower than the lower threshold then that part of the signal is considered a spike (action potential) fired by a neuron. These thresholds can be set manually by visually inspecting the voltage trace, or, automatically by, for example, setting the thresholds as a multiple of the standard deviation of the signal (Bergman et al., 1992; Pouzat et al., 2002; Quian Quiroga et al. 2004; Schmidt 1984; Shahid et al., 2010).

Thus, once we have detected spikes in the signal we need to extract the spike shapes. That is, we extract sections of the signal from sampling point $t - x$ to $t + y$, where t is the point in time when the signal crossed the threshold and x and y is the number of sampling points before and after t respectively. We refer to these extracted sections of the signal as spike shapes. We extract spike shapes from the recordings so we can use them in the spike sorting process. We discuss this process in the next section.

2.4.2 Spike Sorting

Spike sorting refers to the process of determining, from the spike shapes extracted in the spike detection stage, the number of neurons recorded and which spikes belong to each neuron. In other words, given that:

1. the spikes fired by a neuron tend to be of a particular shape, and different from the spike shapes fired by other neurons[3].

2. whenever a neuron fires a spike, the spike shape recorded is rarely exactly

[3]Factors influencing variation in spike shape, i.e. the reasons why spike shapes emitted by neurons are different, range from the purely physical (e.g. position of the electrode relative to the neuron) to the physiological, (e.g. the ion-channel composition of the neuron (Gold et al., 2006)).

the same (i.e. the spike shape recorded from a neuron is usually randomly distorted) (reasons discussed below).

the spike sorting process refers to grouping similar spike shapes together, where the spike shapes in one group are different from the ones belonging to other groups. Therefore, the groups formed correspond to the spiking activity of individual neurons.

The spike sorting process consists of 2 stages: feature extraction and clustering. The feature extraction stage refers to the process of reducing the dimensionality of the spike shape data. That is, the process of converting the spike shapes into a smaller set of components that describe the spike shapes most distinctive features. Thus, whether separable clusters form in the feature space is dependent on the features chosen. The clustering stage refers to the grouping of the spikes based on the feature components derived from the feature extraction stage. Thus, whether the correct number of clusters (neurons) is identified and whether each spike is classified to the correct cluster (i.e. whether each spike is associated to the correct neuron) is dependent on both the feature extraction and clustering method used. The feature extraction stage is implemented to reduce the computational time of the clustering process. That is, clustering data with few dimensions is usually faster than clustering data with many. In addition, the process may also eliminate inputs dominated by noise, thus, improving cluster separation (Quian Quiroga et al., 2004). Furthermore, if one wants to determine the clusters manually, then, to see the clusters visually, each spike waveform has to be plotted using 2 or 3 feature components.

Several spike sorting techniques exist. These include manual (which require the user to perform both stages of the spike sorting process) and automatic methods (i.e. methods that perform both stages of the spike sorting process automatically): (Biffi et al., 2010; Chandra et al., 1997; Delescluse et al., 2006; Fee et al., 1996; Harris et al., 2002 (Klustakwik Program); Hazan et al., 2004 (Kluster software); Hillel et al., 2006; Hulata et al., 2000; Hulata et al., 2002; Letelier et al., 2000; Lipa et al., 2004 (BubbleClust Software); Pouzat et al., 2002; Pouzat 2004; Quian Quiroga et al., 2004 (Waveclus Software); Redishi et al., 2005 (Mclust software); Shoham et al., 2003; Takahashi et al., 2003; Takahashi et al., 2005; Thakur et al., 2007; Vargas et al., 2007; Wehr et al., 1999; Zhang et al., 2004; Other spike sorting

software: Spike2 (Cambridge Electronic Design Ltd., UK)). For other methods see (Brown et al., 2004; Buzsaki et al., 2004; Lewicki, 1998). However, none of these methods supersedes the rest as each has limitations. We discuss the standard feature extraction and clustering methods used in the majority of the work above, below.

Typically, automatic methods perform better than manual methods as they are less labour intensive and are not affected by human biases (Wood et al., 2004). That is, as MEAs can record hundreds of thousands of mixed spike shapes it can be difficult and time consuming for a user to:

1. select distinctive spike shape features by hand.

2. group the data into clusters. That is, determine, by hand, the number of clusters and their boundaries in the feature space. This process is made more difficult when clusters overlap and the spike data is represented in a high dimensional feature space.

Thus, due to the problems with manual methods, many of the available spike sorting methods are automatic. Below, we discuss the main automatic feature extraction and clustering methods.

Automatic Feature Extraction Methods

As mentioned earlier, the first step of spike sorting is to extract the most distinctive features of the spike shapes. Thus, if the features extracted distinguish between the various classes of spike shape, they will reveal separable clusters in a low dimensional feature space. One proposed method is to extract the peak to-peak amplitude or width of the spike shapes. However, (Quian Quiroga et al., 2004) showed that, generally, using such features are not optimal for differentiating between spike shapes. Other methods include principal component analysis (Abeles et al., 1977; Lewicki, 1998), and wavelets (Hulata et al., 2002; Letelier et al., 2000; Quian Quiroga et al., 2004). Within the principal components analysis (PCA) framework, a set of orthogonal eigenvectors of the covariance matrix of the spike shapes is estimated. Then, each spike is completely represented by a sum of the principal component vectors with the corresponding scale factors, the so called scores. The scores extracted, which we refer to as the principal components, are used as spike features for sorting.

In practice, the use of the first two or three scores are used, as they account for the largest variance in the spike shapes. Wavelet transform provides a time-scale decomposition of the spike shapes.

Using the principal components (scores) as the set of features may fail to separate spike classes. This is because PCA selects the directions of maximum variance of the data, which, are not necessarily the directions of best separation (Pavlov et al., 2007; Quian Quiroga et al., 2004). In addition, these components may not reflect the significant distinctions between spike shapes that appear on small time scales (Pavlov et al., 2007; Quian Quiroga et al., 2004). Wavelets can outperform PCA only when the wavelet pre-processing procedure is properly tuned to the spike waveform data by the user. That is, when optimal pre-defined criteria is selected by the user (Pavlov et al., 2007). Selections include: how the wavelets coefficients are calculated (i.e. the mother wavelet used), the level of decomposition (i.e. the time scales we should extract the wavelets at), and, which, and the number of, wavelets coefficients should be used in the clustering process. However, achieving a properly tuned waveform pre-processing procedure is difficult, as typically:

1. we would not know the optimal values (selections) to use, i.e. which criteria best differentiates the different classes of waveform shapes in the dataset.

2. we cannot tune (optimise) the parameter values (i.e. run the wavelet pre-procesing method with a clustering method multiple times with different values until a particular result is achieved) as we would not know the optimal result (i.e. we would not know the number of clusters the features should form).

Thus, due to these problems, wavelet parameter values are typically guessed, and as a consequence, the features extracted from the spike shapes may not form separable clusters in the features space as they do not describe the most distinctive features of the spike shapes (Pavlov et al., 2007).

Automatic Clustering Methods

The clustering stage refers to the grouping of the spike shapes based on the feature components derived from the pre-processing stage. That is, we need to identify the number of clusters in the feature space and assign each spike to one of the clusters.

There are many automatic, i.e. unsupervised, clustering methods used for spike sorting. These include k-means (Lloyd et al., 1982), self-organising maps (Kohonen 1984), super-parametric clustering (used in the spike sorting tool Waveclus (Quian Quiroga et al., 2004)), the Classification Expectation-Maximisation (CEM) algorithm (Celeux et al., 1994) (used in the spike sorting tool Klustakwik (Harris et al., 2000), hierarchical clustering (Fee et al., 1996b) and the k-nearest neighbour algorithm (Cover et al., 1967). While they can perform well, they may not always determine the correct number of clusters in the feature space. In addition, they may misclassify large numbers of spikes, i.e. classify spikes to the wrong clusters, and/or leave many spikes unclassified. This is because some methods are based on assumptions about the distribution and variance of the clusters in the feature space (Harris 2003; Pouzat et al., 2002), which, for some datasets, may be incorrect. For example, the clustering method Klustakwik (Harris et al., 2000), which uses the CEM algorithm (Celeux et al., 1994), is based on the assumption that clusters fit a Gaussian distribution. However, there are several issues that make the assumption of Gaussian clusters not plausible. These are:

1. electrodes move during recording.
2. the shape of the spike waveform emitted by a neuron varies due to bursting.
3. overlapping spikes may occur, i.e. when two neuron fire in synchrony or with a slight delay.
4. non-stationary background noise.
5. the spikes shapes extracted in the spike detection stage are misalligned.
6. correlations between local field potential and spikes.

Thus, methods that assume Gaussian distributions can over-cluster by fitting several multivariate Gaussians (Harris 2003). To avoid this problem some methods allow the user to restrict the number of clusters the method forms. However, choosing an optimal number is difficult, as, typically, we would not know the number of clusters in the feature space. Thus, specifying a value too small or too large may still result in the method clustering the data into too few or too many groups.

Other clustering methods avoid the Gaussian assumption by identifying and grouping together points[4] that reside within the dense areas of the feature space. However, while they can work well, they may need user involvement to achieve a high/better performance. That is, the method may require optimal parameter values. For example, the Super-paramagnetic clustering (SPC) (Quian Quiroga et al., 2004) requires a threshold value that determines which spikes are grouped together. In other words, the method groups together points that are close given that the local density is larger than a chosen value. Thus, if a suboptimal value is used, the method may form too many or too few clusters, and/or, leave many spikes unclassified. Also, for example, the Kohonen's self organising map (SOM) (Kohonen., et al 1984) and K-means (Lloyd et al., 1982) clustering methods, which are both standard unsupervised learning algorithms, need to know the exact number of clusters in the feature space. Thus, if the number used is not correct, the methods will form too few or too many clusters. In practice, choosing optimal values for any method to increase its performance is difficult, and, we cannot optimise such values (i.e. we cannot run the method multiple times with different values until a particular result is achieved) as we would not know the optimal result, (i.e. we would not know which spikes should be grouped together).

2.4.3 New Approach For Spike Sorting

Motivated by the above limitations of current spike sorting methods, we developed a new automatic spike sorting method. That is, an automatic (unsupervised) spike sorting method that:

1. does not require the user to specify optimal parameter values to achieve a high performance.

2. includes a clustering method that is not based on assumptions about the variance and the distribution of the clusters.

Our proposed method uses a novel pre-processing method in conjunction with a novel self-organising map-based algorithm. We discuss our spike sorting method in detail in chapter 5.

[4]we refer to a point as a spike shape in the feature space.

Chapter 3

Introduction Part 2: The Olfactory Bulb

In this thesis we addressed questions and tested our methods on the spiking activity recorded from a rat's olfactory bulb. These recordings were acquired by the neuroscience department of the Babraham Institute in Cambridge using multi-electrode arrays. In this section we introduce the olfactory bulb and the experimental procedure used.

3.1 The Olfactory Bulb

In the mammalian olfactory system, much of the encoding of odor information takes place in the olfactory bulb (OB) (Kendrick et al., 1997; Desmaisons et al., 1999; Bracci et al., 2003; Debarbieux et al., 2003; Egger et al., 2003; Neville et al., 2003; Pinato et al.,2003; Uchida et al., 2003). Olfactory receptor neurons project to glomeruli in the OB, where they synapse with efferent mitral cells (MCs) (see fig. 3.1). The quality of a particular odour is believed to be encoded in the specific combination of glomeruli activated by that stimulus (Johnson et al, 2002; Spors et al., 2002). It is thought that the mitral cells process the input signals arriving from the glomeruli and transmit the information about the odor to multiple other parts of the olfactory system in the brain (Friedrich et al., 2001; Rinberg 2006; Fantana et al., 2008; Soucy 2009; Dong et al., 2009; Mandairon et al., 2009), where the signals may be processed to form a synthesized olfactory perception.

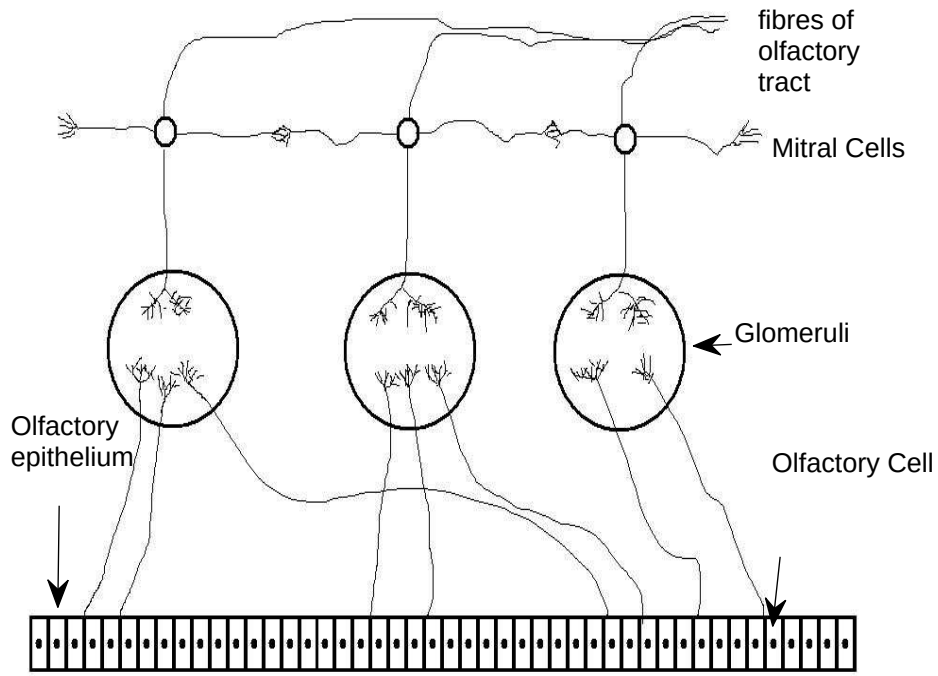


Figure 3.1: Structure of the olfactory bulb

3.2 Experimental Procedure

Neuronal activity was sampled from electrodes on a microelectrode array (MEA) positioned in the olfactory bulb (OB). Microelectrode arrays were comprised of 30 electrodes (6x5, where the distance between each electrode was $350\mu m$). Craniotomy and removal of the left eye permitted lateral access to the OB through the left orbit. During surgery and recordings, humidified air was supplied to the rat through a mask over the nose. A thermistor was used to record air temperature in the mask and to monitor breathing. Shortly ($\sim$30s) before delivery of an odour stimulus, the air to the mask was switched to dry air. Odours, which were the vapours of different solutions, were carried in nitrogen gas (odourless), and delivered to the rat for 10s via the mask. 24 solutions were made. Each solution was made from one of the following compounds: Amyl acetate, M-Butanol, Pinene, DL-Camphor, Linalool, Cineol, at a concentration of either $6.4 * 10^{-8}M$ (moles of solute (dissolved substance) per litre of solution), $2.7 * 10^{-7}M, 1.4 * 10^{-7}M, 6.4 * 10^{-8}M$. Thus, 6 odours (corresponding to the 6 compounds) at 4 concentrations were presented to

the rat 3 times. Onset of odour delivery was precisely timed to mid-expiration. Thus, delivery of the odour to the nasal epithelia would have commenced from the start of the subsequent inhalation (Nicol et al., 2003). Neuronal activity was recorded for a period spanning 10s before odour onset to 10s after odour offset. After the recording process, we identified the number of neurons recorded by each electrode and their corresponding spiking activity using the spike detection method (discussed below) and our spike sorting method discussed in chapter 5. Spikes were detected from all channels when the recorded signal crossed a triggering threshold. This threshold level was set to two times the background noise level, i.e. two times the root mean square (RMS) of the signal.

Chapter 4

Introduction Part 3: Tools To Investigate The Population Rate Coding Hypothesis Using MEA Data, And, The Effects Of Noise Correlation On Population Rate Codes

To obtain a more definitive answer as to how the spiking activity of a population of neurons conveys stimulus information in the brain, we need to test/investigate each of the population coding hypotheses. To investigate/test these hypotheses at the level of multiple cell-cell interactions in the brain, neurophysiologists would typically record, using a multi electrode array (MEA), the spiking activity of neurons in the brain of a test subject, when a stimulus is, and is not, presented. The spiking activity of the individual neurons recorded are then extracted from the recordings (we refer to the datasets containing the extracted neurons and their spiking activity as MEA sorted data) using a spike detection and spike sorting algorithm. The MEA sorted datasets would then be analysed. That is, they would be used to determine whether neurons respond to stimuli in the hypothesised ways, what, and how much information the recorded responses convey about the stimuli,

and address any questions related to the hypotheses. In this chapter/thesis we focus on the population rate coding hypothesis. This hypothesis assumes that stimulus information is conveyed/transmitted in the spiking rates of each neuron, i.e. in the number of spikes fired by each neuron within a relevant time period, in an ensemble. This hypothesis also assumes that the response of each neuron varies according to the stimuli presented (i.e. each neuron contributes information about more than one stimulus) (Averbeck et al., 2006; Barlow et al., 1964; Georgopoulos et al., 1986; Hubel et al., 1959; Maunsell et al., 1983; Mehring et al., 2003; Pouget et al., 2003; Reich et al., 2001; Quian Quiroga et al., 2005; Quian Quiroga et al., 2009; Rolls et al., 1997; Stopfer et al., 2003). In addition, the capacity of a population rate code, i.e. how much information the code conveys, is thought to be dependent on the relationships between the spiking rates of the neurons in the population. More specifically, as:

1. the brain is noisy, (i.e. a neuron's response to the same stimulus on different experiments (trials) will vary) (Stein et al., 2005) and

2. noise is predominately positively correlated (i.e. the spiking rates of neurons, generally, increase and decrease together in response to the same stimulus on different trials[1]).

it is thought that noise correlation may increase or decrease the capacity of a population rate code (Averbeck et al., 2006; Quian Quiroga et al., 2009). That is, noise correlation may determine whether neurons contribute independent, synergistic or redundant information to the ensemble.

In this chapter we firstly discuss methods that can be used with MEA sorted datasets to investigate the population rate coding hypothesis. Secondly, we introduce our proposed analysis method. We then discuss studies that have investigated the effects of noise correlation on the capacity of population rate codes, and we briefly discuss our investigation.

[1]Many empirical studies have shown that, in the brain, the noise correlation coefficient is on average ~ 0.2. For examples see (Averbeck et al., 2006; Kohn et al., 2005; Reich et al., 2001.). We show how the noise correlation coefficient is calculated in chapter 7

4.1 Tools To Use With MEA Data To Address The Population Rate Coding Hypothesis

Currently, there are only a few tools that can be used with MEA sorted data to investigate the population rate coding hypothesis. In other words, there are only a few tools that can be used to determine whether there is a relationship between the spiking rates of the multiple single neurons recorded and the stimulus presented (Brown et al., 2004; Buzsaki et al., 2004). These include:

- Mutual information algorithms. These methods can be used to measure the association between the spiking rates of a neural ensemble and stimuli (Cover et al., 1991, Shannon 1948; Rieke et., 1997;Borst et al., 1999; Reich et al.,2001;Nirenberg 2001; Franco et al., 2004; Victor et al., 2006; Panzeri et al., 2007). This association can also be measured using Fisher information (Cover 2006).

- Decoding methods. These methods can be used to study how the spiking rates of a neural ensemble can be used to represent external stimuli and biological signals (Bialek et al., 1991; Rieke et al.,1997;Brown et al., 1998; Oram et al., 1998; Panzeri et al., 1999; Samengo 2002; Anderson et al., 2002; Quian Quiroga et al., 2006; Huxter et al., 2008).

- Statistical Analysis methods. These methods include likelihood methods and hypothesis tests. Likelihood methods can be used to describe the likelihood of a spiking rate of a neuron occurring due to a particular stimulus (Read et al., 2003; Ventura et al., 2005). Hypothesis tests can be used to determine whether a stimulus evoked a significant change in the spiking rates of a neuron. Tests include: paired t-test (Zimmerman et al., 1997; Chang et al., 2002; Joelving et al., 2007), Kolmogorov Smirnov test (Gibbons 1985; Stark et al., 2009; Shimokawa et al., 2009), multi-variate analysis of variance (MANOVA) test (Johnson et al., 1988; Norris et al., 2006; Deshmukh et al., 2003), ANOVA, (Analysis of Variance (Lindman 1974). We note that, apart from MANOVA, these methods are designed to be used with, at most, the responses of two neurons.

- Visualisation methods. These can be used to visualise the spiking rates of the

neurons recorded (Stuart et al., 2002).

4.1.1 A New Statistical Analysis Tool For The Purpose Of Investigating The Population Rate Coding Hypothesis Using MEA data

The methods discussed in the previous section can extract useful information from MEA sorted data. However, we are still lacking methods to test/investigate the population rate coding hypothesis in detail. That is, methods that, from the MEA sorted data recorded before and during stimulus presentation, provide a detailed statistical analysis of the stimulus evoked changes in the spiking rates of the neurons, and show, the responsive, and most responsive, subareas of neurons (i.e. the subareas exhibiting the most significant change in spiking rates) within the MEA recorded area. We refer to these responsive areas as hot spots.

We require such methods so that we can easily address questions related to the hypothesis such as:

1. does the whole recorded area respond to the stimulus, or, does one or many small subareas respond. In other words, is the stimulus represented by small groups of neurons, or, is it represented in a large group of neurons.

2. do neurons respond to a stimulus with the same or different spiking rates.

The standard tests for determining significant differences between the spiking rates of a population of neurons before and during stimulus presentation are:

1. the ANOVA (Analysis Of Variance) test, which would be used on each neuron.

2. the MANOVA (Multi variate analysis of variance) test, which performs one test on the whole group of neurons to determine/show whether the neurons, in general, responded to the stimulus.

However, these methods are not designed to identify hot spots, i.e show the most responsive subareas (groups) of neurons, within an MEA recorded area. Thus, we propose a method to identify these subareas using MANOVA. We concentrate on MANOVA and not ANOVA as we are interested in identifying the groups of

neurons within the recorded area that were, overall, the most responsive. Before we discuss our proposal, we firstly discuss 2 limitations of MANOVA, which our method overcomes. The standard application of MANOVA:

1. only shows whether a change in the spiking rates of neurons is significant and not how significant. In other words, they are not designed to show how responsive neurons are.

2. may incorrectly state that a stimulus had no effect on the recorded area. This is because MANOVA is designed to analyse the changes in the spiking rates of all of the neurons as a whole, and therefore, if only a small proportion of the neurons responded to the stimulus and a large number remained unresponsive (which we would naturally expect), the former might be completely obscured or diluted by the latter in the test, and therefore, the MANOVA test may assert that there were no stimulus-evoked changes[2].

We therefore propose a novel extension of MANOVA in chapter 6, called MEANOVA. This method identifies hot spots, i.e. the responsive, and most responsive, subareas, within the recorded area and avoids the limitations of MANOVA. Our method consists of:

1. using a MANOVA test, where we modified its output. We modified the output of MANOVA so that it produces a significance score to show how significant the changes in the spiking rates of a population of neurons were overall when a stimulus was presented. In other words, the score shows how responsive a population of neurons was, overall, to a stimulus. How responsive neurons are is not shown by MANOVA (as discussed above), and therefore, our modified version is more informative than the standard MANOVA test.

2. applying the modified version of MANOVA to the spiking rates (recorded before and during stimulus presentation) of subareas of neurons, small (local)

[2]We show an example of this in chapter 6. In brief, we simulated the spiking activity of 81 cells and only one cell responded to a stimulus, i.e. the cell showed a slight increase in activity (spiking rate). The proper application of MANOVA did not detect this, i.e. the null hypothesis that the stimulus caused no significant change in the spiking rates of the neurons was accepted ($P>0.05$). However, our method detected this responsive cell amongst the large number of unresponsive ones.

and large (global), within the recorded area. In other words, MEANOVA produces a score for each set of neurons recorded by individual, combinations, and all of the electrodes on an MEA. Each combination constructed contains a connected set of electrodes. In other words, each electrode in a subset is adjacent to, i.e. is an MEA neighbour of, one of the other electrodes in the subset. For example, we would obtain a set of 13 combinations, shown in table 4.1, if an MEA contained 4 electrodes on a 2x2 layout. That is, the first row had two electrodes labelled 01 and 02, and the second row had two electrodes labelled 11 and 12. Thus, the MEANOVA scores produced from small and large combinations represent how significant the changes in the spiking rates of the neurons were, in general, in small (local) and large (global) neuronal subareas respectively. In addition, the scores would show which subareas were most responsive. That is, subareas (i.e. the neuronal areas recorded by individual and/or combinations of electrodes) that exhibit higher scores than others represent subareas containing neurons that were, generally, more responsive (i.e. subareas that conveyed a more significant change, overall, in spiking activity). Our method, therefore, can detect hot spots, i.e. can identify the responsive, and the most responsive areas, within the recorded area, thus, achieving our initial aim discussed at the beginning of the section. In addition, as our method consists of initially analysing the responses of small numbers of neurons it would uncover small numbers of responsive neurons amongst a large number of unresponsive ones. Thus, our method avoids the limitation of the standard MANOVA test (discussed earlier)

We validate MEANOVA using data recorded from a rat's olfactory bulb and data obtained from a simple computational network model. We used the computational model and biological data to show that MEANOVA works, i.e. it can identify and show the responsive, and the most responsive, subareas within a neuronal area. We also used the computational model to show the limitation of MANOVA (discussed above). That is, that MANOVA can incorrectly state that if one neuron responded to a stimulus amongst a large number of unresponsive ones, as MANOVA would take all neurons into account, MANOVA could incorrectly suggest the stimulus had no effect on the spiking activity of the large population of neurons. We also show

Electrode Combination Number	Electrode Combination
1	01
2	02
3	11
4	12
5	01,11
6	01,02
7	02,12
8	11,12
9	01,02,11
10	01,02,12
11	11,12,01
12	02,11,12
13	01,02,11,12

Table 4.1: This table shows an example set of electrode combinations produced by MEANOVA, if an MEA contained 4 electrodes on a 2x2 layout. That is, if the first row on the array had two electrodes labelled 01 and 02, and the second row had two electrodes labelled 11 and 12

that MEANOVA overcomes this limitation. In chapter 6, we discuss our method and the results in more detail.

4.2 The Effects Of Noise Correlation On Population Rate Codes

In this section we discuss studies that have investigated the effects of noise correlation on the capacity of population rate codes. However, before we discuss the results found from these studies, we firstly discuss how the effect of noise correlation is typically quantified. We then, briefly, discuss our investigation of the effects of noise correlation on population rate codes in the rat olfactory bulb, and, the results we obtained.

4.2.1 Method To Quantify The Effect Of Noise Correlation On Population Codes

Empirical studies typically quantify the effects of noise correlation using Shannon's mutual information algorithm. This algorithm is used to calculate the mutual information between the population response (i.e. a population response vector $\overrightarrow{r}$, where each element in the vector refers to the number of spikes fired by a neuron in a relevant time window) and a set of stimuli (Cover et al., 1991, Shannon 1948). Mutual information is defined as the reduction of uncertainty (or gained information) about the stimulus obtained by knowing the neuronal response. The mutual information I between the stimuli and responses is measured in logarithms of base 2 (bits). Thus, every bit of information provided by the neurons reduces the overall uncertainty about the stimulus by a factor of two. The algorithm is defined by:

$$I = \sum_{s \epsilon S} \sum_{\overrightarrow{r}} P(s, \overrightarrow{r}) \log_2 \frac{P(s, \overrightarrow{r})}{P(s)P(\overrightarrow{r})} \tag{4.1}$$

$P(s, \overrightarrow{r})$ denotes the joint probability of observing the response $\overrightarrow{R} = \overrightarrow{r}$ together with the stimulus $S = s$. These probabilities are calculated from the responses found on different trials (experiments) when the same stimulus is presented. The effect of noise correlation is typically quantified by comparing the information about

the stimuli a population conveys when noise is correlated with the information conveyed by the population when we remove the noise correlation. We calculate the latter information by substituting $P(\overrightarrow{r}, s)$ with $P_s(\overrightarrow{r}, s)$ in the shannon mutual information algorithm. $P_s(\overrightarrow{r}, s)$ is obtained by shuffling the trials corresponding to each stimulus independently for each neuron. Thus, the effect of noise correlation ($\triangle I$) corresponds to $I - I_s$, where I and I_s refer to the mutual information calculated using $P(\overrightarrow{r}, s)$ and $P_s(\overrightarrow{r}, s)$ respectively. Therefore, ($\triangle I > 0$) indicates that noise correlation has increased the information a population conveys and ($\triangle I < 0$) indicates a decrease.

4.2.2 The Effects Of Noise Correlation On Population Codes Found From Empirical And Theoretical Studies

Many empirical and theoretical studies have addressed the question: can noise correlation have an effect on the information a population rate code conveys. Many of the empirical studies have addressed the question using the responses of < 30 neurons to several stimuli (Averbeck et al.,2003; Averecd 2006; Golledge et al., 2003; Montani et al., 2007; Nirenberg et al., 2001; Panzeri et al.,2001; Petersen et al., 2001; Quian Quiroga et al., 2007). They showed the effect of noise correlation on the information these population codes could convey was small (i.e. $\frac{\triangle I}{I_s} \cdot 100 < \pm 5\%$). In addition, they showed the effect (i.e. whether it increases or decreases information) was dependent on the tuning similarity (i.e. whether the signal correlation[3] between the neurons was > 0 or < 0), and the noise correlation (i.e. whether it was > 0 or < 0) between the neurons pooled. More specifically, when the noise and signal correlation between neurons in a population was positive, or, when the noise and signal correlation between neurons in a population was negative, noise correlation decreased the information the population could convey (i.e. the effect of noise correlation was harmful). In addition, when the noise and signal correlation between neurons in

[3]Signal correlation refers to the correlation between the mean response (i.e. the average response across trials to a stimulus) of two neurons. Thus, signal correlation is positive when the means of both neurons increase and decrease together in response to stimuli, or negative, when the average of one neuron increases and the other decreases in response to stimuli. We show how the signal correlation coefficient is calculated in chapter 7.

a population was negative and positive respectively, or, when the noise and signal correlation between neurons in a population was positive and negative respectively, noise correlation increased the information the population could convey (i.e. the effect of noise correlation was beneficial).

Theoretical studies have showed that noise correlation may have a much larger effect on the information large populations (> 30) convey (Abbott et al., 1999; Averbeck et al., 2006; Durrant et al., 2006; Quinn Quiroga et al., 2009; Shadlen et al., 1996; Sompolinsky et al., 2001; Vogels 1990; Wilke et al., 2002 Wu et al., 2002; Zohary et al., 1994). For example, as noise correlation is predominately positive in the brain, (Zohary et al. 1994) investigated whether positive noise correlation can have a large effect on a population size > 100. They found that positive noise correlation limited the information conveyed when the population contained neurons with similar tuning (i.e. the signal correlation between all neurons was > 0). In other words, they found that noise correlation caused information to saturate when the size of the population reached 100. Other studies have showed that noise correlation may not cause information to saturate if the population contains a mixture of neurons with similar and dissimilar tuning. That is, as noise correlation can cause a harmful and beneficial effect within a population, both effects may cancel each other out (Vogels, 1990; Abbott et al., 1999; Sompolinsky et al 2001; Wilke et al.,2002 Wu et al.2002, Durrant et al., 2006; Quinn Quiroga et al 2009). Nevertheless, whether noise correlation has a large effect on large population rate codes in the brain still remains unknown and needs to be addressed. This motivated us to address it in this thesis.

In chapter 7 we address the question: can noise correlation significantly affect the amount of information a large population of neurons conveys about a set of external stimuli. To address this, we used the MEA sorted data of 236 neurons recorded from a rat's olfactory bulb when 6 stimuli was presented (we described the recording/experiment procedure in chapter 3). We used this data as, from initial tests, we found that noise was predominately positively correlated, i.e the noise correlation coefficient between all pairs of neurons was, on average, ~ 0.21. We investigated whether correlated noise could affect the amount of information the neurons convey in their spiking rates about the six stimuli presented. To quantify

the effect of noise correlation we used a mutual information measurement method developed by (Franco et al., (2004))[4]. We used it to calculate the mutual information between the population response and the six stimuli presented, when the responses in the ensemble were correlated and uncorrelated. Thus, the difference between the information the correlated and uncorrelated responses conveyed about the stimuli represents the effect of noise correlation (as described above).

We found the effect of noise correlation was dependent on the tuning similarity of the neurons, and the number of neurons, pooled. That is, overall, as noise correlation was predominately positive, noise correlation significantly[5] decreased the information a set of neurons conveyed when they were tuned to similar features (i.e. when the signal correlation between neurons was > 0), and, significantly increased the information conveyed by a set of neurons when they were tuned to dissimilar features (i.e. when the signal correlation between neurons was < 0)[6]. Furthermore, the magnitude of the harmful and beneficial effects increased with the number of neurons pooled. However, when we pooled all of the neurons, the overall effect of noise correlation limited the information gain. That is, information saturated when the population size reached $\sim$45 neurons, as the beneficial effects in the population did not fully cancel out the harmful effects. Therefore, the pooled response of all the neurons conveyed $\sim$70% of the maximum information that could be conveyed about all six stimuli. In other words, the population could have conveyed 100% of the information about the stimuli if the noise was not correlated. We therefore conclude that, even though noise correlation can be weak, it can have a substantial effect on the amount of information a large population of neurons conveys. We discuss all our results in chapter 7.

[4]We used this method rather than the Shannon's mutual information algorithm for reasons discussed in chapter 7.

[5]We explain in chapter 7 how we determine a significant change.

[6]We note that the noise between some neurons was negatively correlated, which increased and decreased the information a population rate code conveyed when the neurons pooled were tuned to dissimilar and similar features respectively. However, the effect found was insignificant.

Chapter 5

Spike Sorting Based Upon Machine Learning Algorithms (SOMA)

5.1 Introduction

Spike sorting is necessary when interrogating multi-electrode (MEA) electrophysiological data. It refers to the process of determining the number of neurons and their spiking activity from the spikes (spike shapes) extracted during the spike detection stage. Spike sorting is a 2 stage process: feature extraction (which refers to the process of extracting distinctive features from the spike shapes) and clustering (which refers to the process of clustering the spikes based on the features extracted in the feature extraction stage). Several spike sorting techniques exist. These include manual and automatic methods. Manual methods can be extremely time-consuming/impractical when used with datasets recorded by a multi-electrode array as they can contain hundreds of thousands of mixed spike waveforms. Automatic methods are therefore necessary, but, while current methods can perform well, they can miss clusters, misclassify or leave many spikes unclassified. This is because some are based on assumptions about the distribution of the clusters, which, for some datasets, may be incorrect. In addition, some methods may need to be tuned to the dataset by the user (i.e. the user is required to provide the method with optimal parameter values such as the number of clusters, cluster boundaries

or number of features to extract from the spike shapes) to achieve a high performance. However, in practice, this is difficult as the user has no information about the contents of the dataset such as the number of neurons recorded and the number of spikes they fired to base the tuning on. In this chapter we present a fully automatic spike sorting method (i.e. a method that automatically extracts/determines a number of features from the spike shapes that form a number of separable clusters in the feature space, and, automatically determines the number of, and groups the spikes into, clusters) that avoids the limitations of other automatic methods. In other words, our method does not need optimal pre-defined criteria to achieve a high performance and our clustering method is not based on assumptions about the distribution of the clusters. Our method consists of using a novel pre-processing stage in combination with a novel self-organising map-based clustering method.

One current method used for clustering is a Kohonen's Self organising map (SOM), a standard unsupervised learning algorithm (Kohonen 2001). However, a simple application of this technique would only identify a known, or estimated, number of clusters. Here we discuss extensions to the SOM learning and classification process, so that the SOM automatically identifies the number of clusters within the dataset, and automatically identifies the edge of each cluster to improve cluster separation. In other words, our method forms a better implicit boundary around each cluster, and thus, as the SOM classification process groups points together that reside within each boundary, more points in each cluster will be correctly grouped together. This process, therefore, reduces the chance of misclassification. However, the performance of a clustering method relies on the components produced through pre-processing. Two common approaches are PCA or wavelets to classify the data. While these work well, the former method can be insufficient to form separable clusters in the feature space and the latter may need to be tuned to the dataset by the user to achieve a high performance (Pavlov et al., 2007). We therefore also discuss a new pre-processing technique, which transforms the spike waveform data into lower dimensions to form separable clusters, and works efficiently with no user involvement (i.e. without the need for complex or near optimal parameter choices). Our new approach sequentially appends features (i.e. continuously extracts more detail from the spike shapes) describing the geometric shapes that constitute a spike

waveform based on curvature to PCA components until the number of spikes in each cluster formed represents a firing rate below a threshold level. We used curvature as it can show crucial differences in the spike shapes that appear on small time scales (Hsuing 1997), which principal components may not show (Pavlov et al., 2007).

To validate our new spike sorting approach we applied it to multi-electrode array datasets acquired from the rat olfactory bulb, and from the sheep infero-temporal cortex, and using simulated data. We also compared our method with other standard automatic spike sorting methods, namely the wavelet pre-processing procedure (described in (Quian Quiroga et al., 2004)), and the clustering techniques, Klustakwik (Harris 2000), which uses the CEM algorithm (Celeux et., 1992) and Waveclus (Quian Quiroga et al., 2004), which uses the super-paramagnetic clustering (SPC) technique (Blatt et al., 1996). To compare these methods we used them on biological datasets containing a known number of neurons and the number of spikes they fired. In addition, we used them automatically, and, how we would use them in practice, i.e. with the default values. We found that our method was at least as good as the others, i.e. wavelets formed separable clusters, and Klustakwik and Waveclus found the correct number of clusters and classified a high percentage of the spikes correctly. However, our pre-processing and clustering method outperformed them when the spike shapes emitted by each neuron were very similar in shape, and, when the spike shapes were highly distorted. Thus, the results show that our new pre-processing technique and clustering method worked efficiently, in terms of the number of clusters found and the number of spikes correctly classified, without assumptions and without the need for user involvement, i.e. without the need for optimal parameter values.

Parts of all the sections in this chapter were published as: Horton P M, Nicol A U, Kendrick K M, Feng J F. Spike Sorting Based Upon Machine Learning Algorithms (SOMA) J Neurosci Meth 2007;160:52-68. I confirm that I performed all simulations and developed the ideas. The other authors are my supervisor, J Feng, who advised me, and K Kendrick and A Nicol who provided the electrophysiological data. We note that this spike sorting algorithm was used in other studies (Tate et al., 2006; Nicol et al., 2005).

5.2 Methods

5.2.1 Extracting Experimental Spike Data

In this chapter we used spikes recorded from the rat olfactory bulb and sheep infero-temporal cortex to explain and test our methodology. The spikes were extracted from the continuous electrophysiological trace sampled at an electrode when the signal exceeded a given threshold (twice the standard deviation of the signal (trace)). We note the sampling rate in all experiments was 29.09khz. Each spike was sampled for a period spanning 0.4ms preceding, to 1.25ms after the threshold was crossed. Most spikes collected were comprised of 48 sample points ($t = 1, \cdots, 48$), where the signal crossed the spike-triggering threshold at $t = 11$. However, some of the example spikes were comprised of 40 sample points, i.e. 0.4ms preceding, to 0.96ms after the threshold was crossed.

5.3 Spike Sorting Pre-Processing Method Through Feature Extraction

In spike sorting algorithms, using the entire spike shape with clustering methods can perform badly and be computationally expensive (Quiroga et al., 2004, and discussed at the end of the chapter). Therefore, the first stage of any methodology is to extract features which describe the spikes and, crucially, discriminate between them, thus revealing separable clusters within the feature space. One commonly used technique is principal component analysis (PCA), which can automatically extract spike shape features (Bishop 1995; Csicsvari et al., 1998; Jolliffe, 2002). However, using PCA is not always reliable and can produce results that are not biologically plausible. This is because PCA selects the directions of maximum variance of the data, which, are not necessarily the directions of best separation. For example, when we used PCA on a spike dataset recorded from a rat's olfactory bulb (shown in fig. 5.1), we found that the PCA components formed one cluster suggesting that one neuron was recorded with a firing rate >200Hz.

To resolve the problems associated with PCA we propose a new feature extraction technique. The process involves appending extra feature components, which

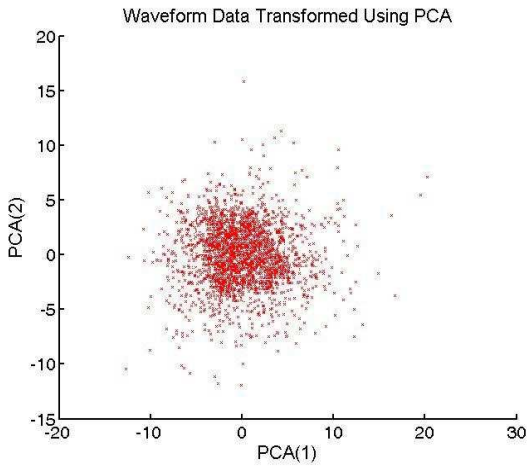


Figure 5.1: This shows that one cluster was formed from a spike set recorded from a rat's olfactory bulb using PCA, where components 1-2 are shown. These results suggest that one neuron was recorded with a firing rate exceeding 200 spikes/sec, which is an unfeasibly high rate for a mitral cell to sustain

describe a waveform's geometrical shape as a set of curvature components, to the corresponding waveform's PCA set. To obtain curvature components, we calculate the curvature at every sampling point on the waveform, divide every waveform into G sections along the time axis t and calculate G components as the average of the curvature values in the section. That is, we calculate a component for each section, where each describes the curve shape. The intuition behind this approach is that different curve shape combinations constitute different waveforms and so adding information about the curve shape to PCA will form distinguishable clusters

Clearly, the success of this process will depend on the number of extra components G used. The optimal number relates to the current clusters and their associated firing rates, but this is not known in advance and we therefore want to determine this automatically. In addition, to limit computational costs, we want to use the smallest number of components possible. Thus, we start with $G = 3$ (reason discussed below) and incrementally increase it by 1, until the number of spikes in each cluster within the feature space represents a firing rate below a threshold level. In other words, we increase the number of curve shapes which describe the waveforms, thus, forming better separated clusters. We start with a minimum of $G = 3$ as we found that, in many cases, using less did not form better separated clusters.

Consequently, the pre-processing method identifies a minimal number of extra components to use, thus, improving the efficiency of the spike sorting process.

Another commonly used method for feature extraction is Wavelets (Mallat, 1989; Chui, 1992; Samar et al., 1995; Quian Quiroga et al., 2001; Quian Quiroga, et al., 2003) which can be a viable alternative to PCA (Hulata, et al., 2000; Letelier et al., 2000; Hulata et al, 2002; Quian Quiroga et al., 2004; Smith et al., 2007). Indeed, we could have appended wavelets to PCA instead of the curvature components in our approach. However, the wavelet transform may not always extract optimal information from the spike shapes, as the transform is very dependent on the parameters chosen (e.g. the mother wavelet and wavelet selection procedure) and, selecting optimal parameter values can be difficult. Therefore, the wavelet coefficients extracted from the spike waveforms may not always form separable clusters within the feature space (Pavlov et al., 2007). As we wanted an automatic system, we therefore used the curvature-based methods, and compare our method with wavelets.

The *Pstage* algorithm

We refer to our pre-processing stage as *Pstage* and the feature components extracted by *Pstage* as *Pcuv* throughout the chapter. As stated above, (*Pstage*) proceeds by firstly extracting PCA components and then appending 3 curvature components to the PCA componenets. The 3 curvature components correspond to the average of the curvature components within 3 equal sections of a waveform. Thus, if the number of spikes in each cluster does not represent a firing rate below a threshold level using these components, we incrementally divide the waveform into more equal sections, calculating the curvature component of each section, and stopping when the number of spikes within each cluster represents a firing rate below the threshold. This is implemented by:

1. **Extracting out the higher frequency components of each waveform, i.e. smooth the waveform, using a moving average filter.** The formula is:

$$y(i) = \frac{1}{M} \sum_{j=0}^{M-1} x(i+j) \tag{5.1}$$

where $x()$ is the input signal, $y()$ is the output signal, M is the number of points used in the moving average and i is the i^{th} sampling point. We use this formula on each sampling point. The optimal amount of smoothing to use is a matter of trial and error. Fig. 5.2 shows the results of smoothing a waveform (Panel (a)) using a filter length of $M = 5$, $M = 10$ and $M = 25$ (Panels (b)-(d) respectively). We can see when $M = 10$, noise is reduced and the underlying signal is still present, however, when $M = 25$ (Panel (d)) the signal is distorted and is very different from the underlying waveform shape (Panel (a)). In this chapter we therefore used $M = 10$, which was found to generally reduce the noise but still maintain the underlying form of the signal, i.e. waveform shape.

2. **Initialise the extra number of components to 3** i.e. $G = 3$. We use 3 for the reasons stated earlier.

3. **We normalise the smoothed data and use it with PCA:** We exclude principal components that contribute to $< 0.5\%$ variance. From our experience, using the excluded PCA components did not improve the clustering performance, i.e. enhance the separation between the clusters.

4. **Calculate a set of curvature scores K for every point within the smoothed spike waveform:** (Explained in more detail below).

5. **Identify the number of clusters formed using the PCA components and the current G components:** We identify the number of clusters present using the clustering stage described in the next section.

6. **Quit process: if every cluster of spikes corresponds to a firing rate $<$ threshold level.** We use a default threshold value of 50Hz. We found this value to be sufficient in all our tests, i.e. on every test we performed, the method found the correct number of clusters (neurons). However, this value can be changed if one knows the maximum firing rate of the neurons in the

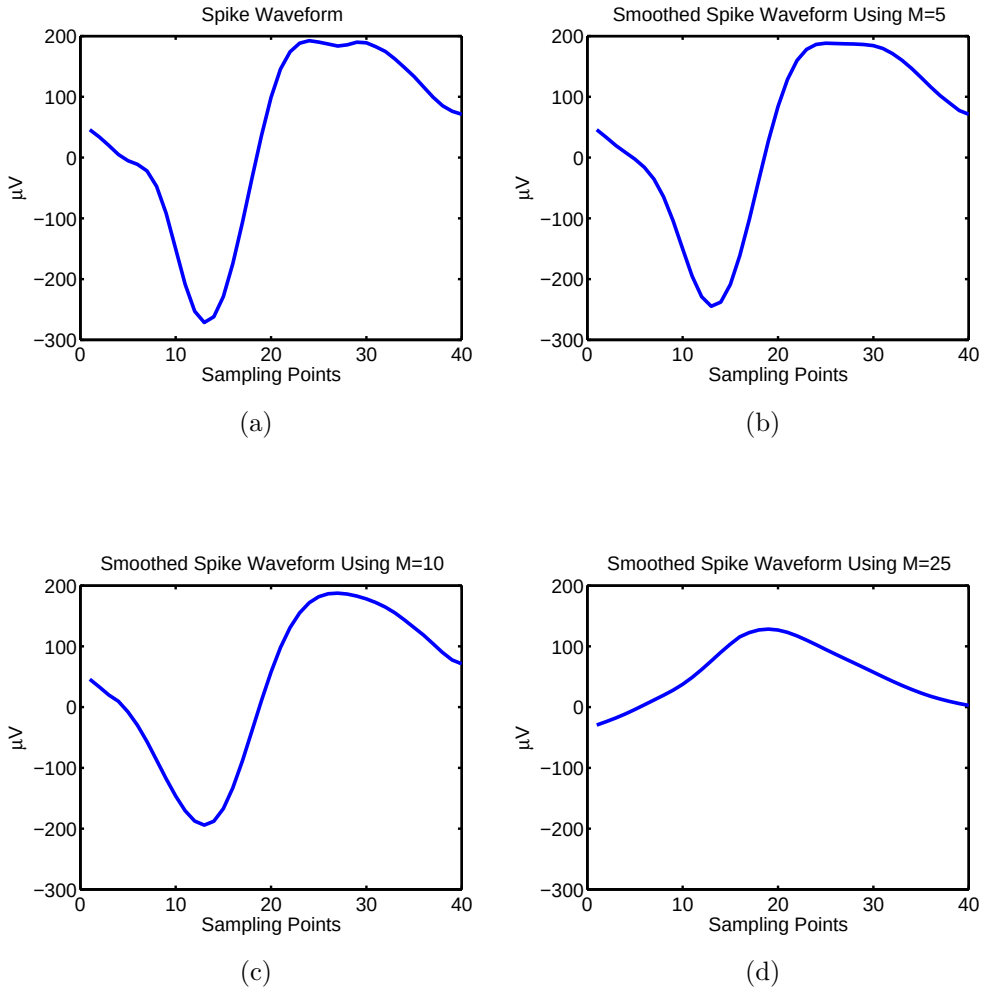


Figure 5.2: This shows the results of using the moving average with different filter sizes. **Panel(a)** shows an example spike waveform shape recorded by an electrode. **Panels (b)-(d)** show the results when the original spike waveform (**Panel(a)**) was smoothed using the moving average filter when the filter lengths were M=5, M=10 and M25 respectively.

area recorded. We also note that if all of the detail about the spike shapes based on curvature is extracted (i.e. G has reached the highest possible value) and the condition is still not satisfied, the output of the method, i.e. the clustering of the spike shapes, will be based on the maximum (feature) detail extracted (i.e. the highest G value).

7. **Increase G by 1.**

8. **Calculate a new set of G components for every waveform:** We divide the set of K curvature values into G consecutive subsets, which partitions the corresponding waveform into G sections and calculate the average curvature value within each section. We discuss this in detail below.

9. **For every waveform, replace the previous extra components by appending the new of G components to the corresponding PCA set.**

10. Repeat steps 5-9.

Calculating A Set Of Curvature Scores K For A Spike Waveform

A set K is defined by $K = \{k(2), ..., k(T-1)\}$, where $k(t)$ refers to the curvature score at sampling point t on the waveform. $k(t)$ describes the curve shape and its curvature, i.e. how curved it is, at point t. T refers to the total number of sampling points. We do not calculate $k(t)$ for the first and last points of the waveform, as they do not have two adjacent points. For example, if there were $T = 48$ sampling points, K would contain 46 $k(t)$ scores.

To calculate $k(t)$, we follow the procedure of Rutter (2000). We acquired the values of the waveform's electrophysiological trace at $t-1$, t and $t+1$. We then approximated the first and second derivatives at t, using the 3 values with the central difference approximation (we note that more than 3 values could be used to approximate the derivatives at t but from our experience we found 3 was sufficient) and estimated the curvature $k(t)$, which is defined by:

$$k(t) \qquad : \mathcal{C}^2 \Rightarrow \mathbb{R} \tag{5.2}$$

$$f \mapsto \frac{f''(t)}{(1+f'(t)^2)^{\frac{3}{2}}} \tag{5.3}$$

For further information see (Rutter, 2000).

The central difference approximation for $f^{'}(t)$ is:

$$f^{'}(t) \approx \frac{f(t+h) - f(t-h)}{2h} \tag{5.4}$$

The central difference approximation for $f^{''}(t)$ is:

$$f^{''}(t) \approx \frac{f(t+h) - 2f(t) + f(t-h)}{h^2} \tag{5.5}$$

where h is the interval size. In this work we used $h = 1$, as stated above.

We acquire the i^{th} component of G, i.e. G_i, for a waveform by averaging the curvature values over the G consecutive subsets of the corresponding K set using:

We define the subset size using $S = A/G$, where A is the number of scores in K. We note if S is not a whole number, we decrease the size of the K set until it is. Thus, A is initially $T - 2$, where T is the number of sampling points

$$G_i = \frac{1}{S} \sum_{j=(i-1)S+1}^{iS} k_j \tag{5.6}$$

where k_j refers to the j^{th} element of K.

5.4 Spike Sorting Clustering Method

Having explained the pre-processing stage, this section describes the spike clustering method which extends Kohonen's clustering method (Kohonen 1984; Ritter et al., 1989). In this method, clusters are firstly identified and then points (spikes) are classified based on these clusters. The method we present extends the basic method in two ways:

1. identifies, automatically, the number of clusters within the feature space. This is achieved by identifying the densest areas in the feature space (section 5.4.2).

2. automatically identifies the edge of each cluster to improve cluster separation. In other words, our method forms an implicit boundary around each cluster, and thus, the points inside the boundary of each cluster are grouped together. This process, therefore, reduces the chance of misclassification. (section 5.4.2).

5.4.1 Simulated Data To Test And Explain Methodology

To explain and test our methodology, we generated several datasets containing 2-D or higher dimensional datapoints. For each dataset, we distributed the data into different cluster sizes and positions. To generate a dataset we firstly chose a set of mean vector datapoints, where each set represented a cluster centre and position. We then created the clusters by adding sets of vectors to each mean datapoint vector. We selected these randomly from a normal distribution specified by the mean datapoint vector and a chosen standard deviation[1]. In this chapter, we refer to the datapoint dimension as C_i. For example, a 2-D dataset would have datapoint dimensions C_1 and C_2. We show a simulated dataset example containing 2 clusters, one much larger than the other, in fig. 5.3 Panel (a). The mean datapoint vectors of the left and right clusters were [-10,-4] and [21.5,-6.3] respectively. We created the left using 1000 sets of 2-D vectors, which we selected using a standard deviation of 8. We created the right cluster in the same way but with 500 sets of 2-D vectors with a standard deviation of 4. We use this simulated dataset to describe our methodology throughout the chapter, unless otherwise stated[2,3].

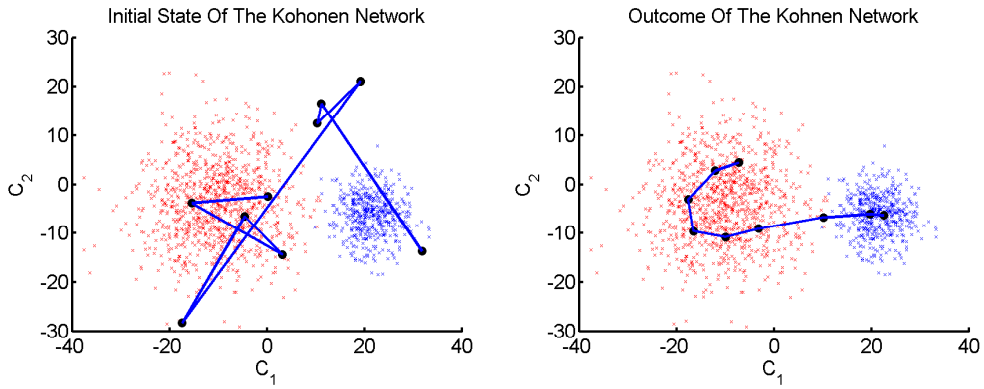
5.4.2 Clustering Method

There are many unsupervised spike sorting clustering methods (Brown et al., 2004), but, many have problems. That is, they sometimes fail to detect the correct number of clusters automatically and may result in many unclassified and/or misclassified spikes. This is because some methods group the spikes into clusters, either, using incorrect assumptions about the distribution and variance of the data (Harris 2003; Pouzat et al., 2002), or, using suboptimal pre-defined criteria, i.e. the default values (Fee et al., 1996b; Quian Quiroga et al., 2004). As discussed in chapter 2, choosing

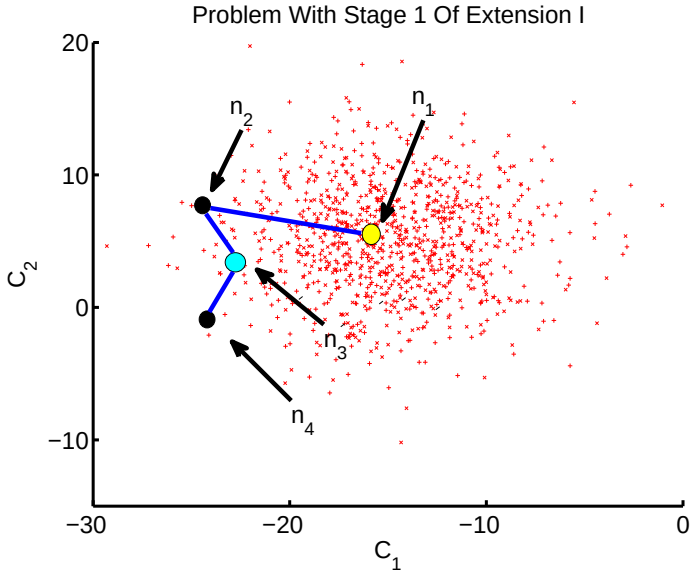
[1]We note that our simulated data was created for the purposes of explaining, and testing, our clustering methodology and not to recreate realistic recordings. We test our method on real electrophysiological data later in the results section.

[2]We use 2 components, i.e. dimensions, throughout this chapter for explanation, and presentation, purposes. The entire method is applicable to many more components as in practice 2 dimensions would not reveal separable clusters within the feature space. At the end of the chapter, we show the results of applying the method to >3 dimensions.

[3]We refer to a feature component set as a datapoint throughout the chapter.



(a) (b)



(c)

Figure 5.3: **Panel (a)** shows the simulated data set and the initial state of the Kohonen network. The circular markers (nodes) represent the outputs. **Panel (b)** shows the outcome of the Kohonen network after using 250 epochs. The positions (weight vectors) of all 9 nodes are shown, where each represents a cluster but two represent the denser areas, i.e. the cluster centres. The lines that connect them represent the nodes' neighbours. **Panel (c)** shows nodes n_1 and n_3 represent the same cluster, and, n_2 and n_4 are n_3's neighbours.

optimal parameter values so that the method achieves a high performance is difficult as, in practice, we would have no information about the dataset, i.e. what the optimal value or result of the method should be.

As we need an automatic method, we propose a clustering method, which uses and extends the Kohonen clustering process. The method automatically detects the correct number of clusters (neurons) (Extension 1) and automatically identifies the cluster sizes (i.e. automatically determines cluster boundaries) thereby reducing the chance of misclassified spikes (Extension 2). In other words, the method groups the spikes into clusters without the above limitations. That is, our method achieves a high performance without assumptions about the distribution or variance of the data and without the need for optimal predefined criteria.

Kohonen Network

The first stage of our clustering process is to locate, using an unsupervised Kohonen network, the denser areas of the feature space that may represent cluster centres. A detailed discussion on the Kohonen network is not included in this chapter, but can be found in many papers and books, e.g. (Kohonen 2001). However, the basic Kohonen algorithm is described below:

Before the training process, we have to select several network features:

1. **Select a number of network outputs:** For a typical unsupervised Kohonen network, we would normally use the same number of outputs as the number of clusters in the feature space, but we would, typically, not know this information. Due to this problem there exists no optimal choice, as an advantage of using one number in one dataset may not be the same in another. However, we found that using the maximum number of neurons that could be in the dataset calculated from a minimum firing rate (we used a default value of 15Hz) that could feasibly be sustained by a neuron as the number of outputs was a sufficient choice. For example, if 1000 spikes was recorded over a 1s period and we use a minimum firing rate of 15Hz, we would, therefore, expect the dataset to contain the activity of at most 1000/15=66 neurons. 66 would, therefore, correspond to the number of outputs we would use with the

example dataset containing 1000 spikes. We note that the default value of 15Hz could be changed if a dataset used with our spike sorting process contains neurons with a known lower firing rate. However, in all our tests the number of outputs acquired using 15Hz was usually more than the number of clusters in the feature space, which is important for the Kohonen extensions 1 and 2 discussed later. In this chapter, we refer to the set of network outputs as N, i.e. $N = \{n_1, ..., n_U\}$, where U is the number of outputs and n_i refers to i^{th} network output. For explanation purposes throughout the chapter, we also refer to n as a node within the feature space, i.e. the node represents the vector the output stores.

2. **Select a Topology.** We used a 1-dimensional topology, where we update the winning node's two adjacent neighbours by half the amount. For example, we update nodes 1 and 3 if the winning node was 2. We used a 1-D topology to adapt the nodes (outputs) so that each had a chance of residing in a cluster. As the later training stage (extension 2) spreads the nodes out more optimally, the choice of topology in the initial training stage, from our experience, was generally not important. However, we do note that using a 1-D topology can be computationally less expensive than a higher dimensional topology. For example, we would need to update more network outputs using a grid like topology, i.e. the winning node and its surrounding neighbours (5 outputs), compared to a 1-D topology (3 outputs). Therefore, training a network that has a grid like topology using large datasets e.g.(> 100000) spikes, where a large input vector (> 10) represents each spike, may take longer. We note that a grid like topology can slightly increase the overall performance of SOMA, i.e. reduce the number of missclassified spikes. We discuss the effects of a grid like output on our method, in terms of speed and performance, in the results section of this chapter.

3. **Select the number of training epochs and the learning rate:** Again, as with selecting the number of outputs, there are no optimal choices. However, from our experience, we found that using 250 epochs[4] and a learning rate of

[4]An epoch refers to every datapoint being used once to train the network.

0.5, which decreased by 0.001 at the start of every epoch, was sufficient. That is, using these values with the Kohonen network, generally, caused the nodes to spread out evenly across the clusters in the feature space. This is because the movement of each node gradually decreased as training got closer to finishing. This is important for Kohonen extensions 1 and 2 discussed later.

4. **Select the initial connection values (weights) to the outputs. In other words, select the initial node positions:** We select the initial position of n, i.e. the weight vector W_n, at random from the range of feature component values.

5. **Select distance measure:** We used the Euclidean distance measure. We refer to this distance measure as $d(W_x, W_y)$, i.e. the distance between the weight vectors x and y, throughout the chapter.

Fig. 5.3 Panels (a) and (b) show the state of a Kohonen network before and after training respectively. We used the network with 9 outputs, the above features, and the simulated dataset discussed earlier. We can see from the figure that the outputs (nodes (black markers)) of the trained network are sufficiently spread across the clusters due to the network features chosen. We next discuss the Kohonen extensions 1 and 2, where extension 1 identifies the central nodes, i.e. the nodes nearer the cluster centres, and extension 2 uses the residual nodes to reduce misclassification.

Extension to Kohonen Network I (identifying the number of clusters)

This section describes how we extend the Kohonen network clustering procedure to identify the number of clusters present within the feature space. We achieve this by comparing the data distributions that a node and its two adjacent neighbours represent.

Subsets of Kohonen network outputs (nodes) will represent different clusters (for an example, see fig. 5.3 Panel(b)) after training the network. We identify the nodes closest to a cluster centre as the ones representing a denser area, i.e. the nodes nearer to a number of datapoints compared to their neighbours[5]. However, the set

[5]The Kohonen network features chosen at the beginning will always result in a subset of nodes representing every cluster, where one will be nearer the centre. Additionally, the outputs will

compiled can contain nodes representing the same cluster that we need to remove by assessing the density of the distribution between every two nodes in the set. Thus, as we would expect the area between clusters to be less dense than the density of the clusters, if we have two nodes a and b and we analyse the density at various positions between a and b, if the density at any position is less than both a and b we would assume that they reside in different clusters, otherwise, we would assume they reside in the same cluster. If they reside in the same cluster we remove the node residing in an area less dense than the other node. We discuss this in the next section. Therefore, we implement two stages to identify the central nodes, which we describe briefly below:

- Stage 1: Identify A Set Of Central Nodes Z
    1. **Calculate a mean distance D score for every node n:** This describes the mean distance between a node n and n's X nearest datapoints.
    2. **Identify a set Z of possible central nodes:** We achieve this by comparing the D score of every node and its two adjacent neighbours.

- Stage 2: Verify that each node in Z represents a different cluster
    1. **Analyse the changes in the datapoint distribution between two nodes, from Z, using the D and B (we introduce B later) scores. B essentially estimates the density of the feature space a node resides in by counting the number of datapoints within a boundary surrounding the node.**

We describe these stages in the subsequent two sections.

Stage 1: Identifying A Set Of Nodes That Represent Potential Cluster Centres

maintain their number order, therefore, comparing each node with its two adjacent neighbours is always possible.

Calculating A Mean Distance Score D For A Node

The score D describes the mean distance between a node n and its X nearest datapoints. A low score represents a set of close datapoints to n, i.e. a potential cluster centre (high density area), and a high score represents a set of distant datapoints , i.e. the outer area of a cluster (low density area). To acquire the score, we calculate the average euclidean distance between n's weight vector W_n and several of its closest datapoints. The equation is shown below:

$$D(n) = \frac{\sum_{i=1}^{X} d(W_n, J_i)}{X} \tag{5.7}$$

where J refers to a set of X datapoints that are close to n, J_i refers to the i^{th} element of set J and $d(W_n, J_i)$ refers to the Euclidean distance between W_n and J_i. X refers to the number of n's closest datapoints used to calculate $D(n)$. Generally, no value for X provides an optimal choice, as an advantage of one value in one dataset may not be the same in another. However, we want $D(n)$ to reflect the density of the cluster area n resides in. Therefore, X cannot be larger than the number of datapoints in the cluster n resides in. For example, if X was larger than the number of datapoints in the cluster, $D(n)$ would be calculated from the datapoints of other clusters, and, if n resides in an area with a very dense distribution, the $D(n)$ score may suggest the area is sparse. Furthermore, X cannot be too small. This is because using a few very close datapoints to calculate $D(n)$ may, also, not accurately represent the density of the cluster area n resides in. For example, if we calculate $D(n)$ from a very few close datapoints, but n resides in an area with a sparse distribution, $D(n)$ may suggest the area is dense[6]. Therefore, we use an X value, which is a fraction of the minimum cluster size that could be in the dataset. We calculate the minimum cluster size using a minimum firing rate that a neuron could feasibly sustain. We, again, used a default value of 15Hz. Thus, $X = round(\frac{m \cdot T}{q})$, where m is the chosen minimum firing rate, T is the total recording time, and $\frac{1}{q}$ is the fraction used. From our experience we found that using $q = 4$ and the default value of 15Hz was sufficient on many tests. That is, these values tended to produce X values that were not too small or too big.

[6]We show the effects of large and small X in the next section.

Identify A Set Of Central Nodes Z

We identify a set of central nodes Z by picking those in the densest areas of datapoints. We achieve this by identifying the nodes which have a lower D score compared to its two adjacent neighbours, so a node n_g, where $g \neq 1$ or $g \neq U$, is in a dense area if:

$$(D(n_g) < D(n_{g+1})) \wedge (D(n_g) < D(n_{g-1}))$$

where n_{g-1} and n_{g+1} represent n's adjacent neighbours. When $g = 1$ and $g = u$, we analyse n_1 and n_U differently as they do not have two adjacent neighbours. So, n_1 is in a dense area if:

$$(D(n_1) < D(n_2))$$

and if $g = U$, n_U is in a dense area if

$$(D(n_U) < D(n_{U-1}))$$

Therefore, Z contains the nodes that satisfy one of the above conditions, i.e. they are near or within cluster centres. From Z we form another set E that contains the nodes within the clusters' outer areas, i.e. E contains the nodes that are not in Z.

We used this method on our simulated data and trained network, shown in fig. 5.3, Panel (b), which identified a set Z comprising two nodes, n_5 and n_9. We show the D scores for all n in table 5.1.

n	1	2	3	4	5	6	7	8	9
D scores	4.9900	4.8939	4.5988	3.6290	2.8974	3.6538	5.5923	2.6230	2.1342

Table 5.1: The table shows that nodes n_5 and n_9 represent potential cluster centres. The nodes surrounding these have higher scores, thus represent the cluster outer areas.

Stage 2: Verifying The Nodes In Set Z Represent Different Clusters

Set Z can contain nodes that represent the same cluster. For example, from the D scores in Table 5.2 we would conclude that nodes (n_1 and n_3) constitute the set Z, i.e. these two nodes represent different clusters. This is incorrect (see fig. 5.3 Panel (c)). The process includes n_3 in Z because its two neighbours are positioned further out causing $((D(n_3) < D(n_4)) \wedge (D(n_3) < D(n_2)))$. So, as n_3 has a higher

D score than n_1, n_3 would need removing.

n	1	2	3	4
D scores	2.6459	5.7898	3.9678	4.6898

Table 5.2: The table shows there are two nodes, n_1 and n_3, that represent potential cluster centres.

Determining Whether Two Nodes Represent Different Clusters:

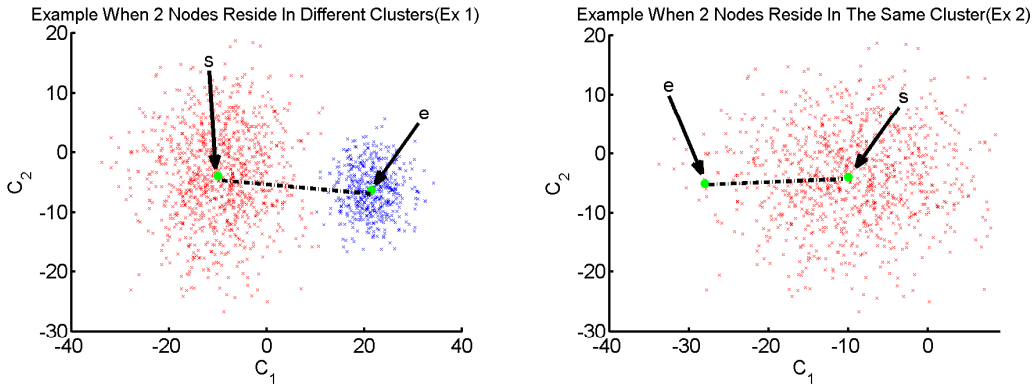
To determine whether two nodes, s and e, represent different clusters, we analyse the datapoint distribution at different positions between them using the D and B scores (the B score is discussed below). Essentially, to determine if s and e are in different clusters, the distribution of the datapoints between s and e should be less dense compared to the areas s and e reside in. Thus, we calculate the density, which is represented by D and B, at various positions between s and e to determine whether a less dense area at one of the position exists between the two, and, if one exists, we would conclude that s and e reside in different clusters. We calculate both D and B for efficiency (discussed below).

The B score for a node n, i.e. $B(n)$, is the number of datapoints that satisfy $d(W_n, x) < r$, where x is a datapoint vector, W_n is the node's weight vector and r is a boundary value. In other words, $B(n)$ is the number of datapoints within the boundary r surrounding n. We use both D and B scores to determine whether two nodes represent different clusters to improve efficiency, i.e. speed up the process. From our experience, the D scores on some tests can show that s and e are in different clusters before the B scores do, and, vice versa.

To calculate B at different positions between s and e we need to calculate a value for r. We found the choice requirements of r was the same as choosing X, i.e. r could not be too large or small. In other words, if r is too big or small, the datapoints used to calculate the B score may not accurately represent the density of the cluster area. Therefore, we found calculating r using the distance between s and e, i.e.:

$$d(W_s, W_e)/q \tag{5.8}$$

with $q = 4$ was sufficient. We discuss the effects of different r values in the next section.



(a) (b)

Figure 5.4: This figure shows the simulated data (EX1) and (EX2). **Panels (a) and (b):** show the simulated dataset containing 2 clusters (EX1) and 1 cluster (EX2) respectively. The markers in the middle of the clusters represent central nodes, labelled s and e, and the striped line represents the path of positions. We use T equally spaced positions between s and e to calculate $D(m_j)$ and $B(m_j)$ to reduce computational costs.

In fig. 5.4 Panel (a), we show an example where two nodes, s and e, represent different clusters. We refer to this dataset as (Ex 1). To determine whether s and e represent different clusters, we analyse the changes in the D and B scores at T positions between s and e. The line connecting s and e in fig. 5.4 Panel (a) indicates the set of potential positions. For our discussion we use $T = 40$ (From our tests, we found no optimal value for T, as an advantage for one value in a dataset may not be the same for another. However, we discuss the effects different T values can have later in this section). Furthermore, we refer to a single position as j and the vector at position j as m_j. Therefore, the vector at position $j = 1$ would equal W_s, i.e. $m_1 = W_s$, and the vector at position 40 would equal W_e, i.e $m_{40} = W_e$. So, if s and e represent different clusters and m moves from s to e, we would expect m to satisfy the condition (which we refer to as *clust* throughtout the chapter) below:

$$((D(m_j) > D(s)) \wedge (D(m_j) > D(e))) \vee ((B(m_j) < B(s)) \wedge (B(m_j) < B(e))$$

at a position j. In other words, we would expect m to move through an area, between s and e, with a lower density of datapoints than the areas s and e are in.

We tested this condition using the data from (EX1)(fig. 5.4, Panel (a)), i.e. when s and e are in different clusters, and show the results in fig. 5.5.

Fig. 5.5 Panel (a) shows the $D(m_j)$ score, and Panel (c) shows the $B(m_j)$ score, at 40 positions between s and e respectively. We calculated each $D(m_j)$ score using $X = 125$, and each $B(m_j)$ score using $r = 5.12$. Both r and X were acquired using the calculations discussed earlier. X was calculated from the smallest cluster, which contained 500 datapoints. Furthermore, r was the same for s, e and m and was calculated using s and e with equation 5.8. The results show as j increased to $j = 23$, i.e. when m moved from s, the $D(m_j)$ score increased from 2.8 ($D(s)$) and peaked at 6.32 (Panel (a)). Furthermore, $B(m_j)$ score decreased from 182 ($B(s)$) to 32 (Panel (c)). When j increased from 23 to $j = T$, i.e. when m moved from $j = 23$ to e, $D(m_j)$ decreased to 2.07 ($D(e)$) (Panel (a)) and $B(m_j)$ increased to 272 ($B(e)$)(Panel (c)). We note that $D(m_j)$ peaked, and $B(m_j)$'s lowest score was, at $j = 23$, as it was the least dense area between the two dense areas (clusters). Fig. 5.5 Panels (b) and (d) show the positions of s, e and m_{23} in the feature space and the datapoints used to calculate s, e and m_{23}'s D (Panel (b)) and B (Panel (d)) scores. The nodes are represented as circular markers. The lines in Panel (b) refer to the distance between the nodes and its X closest datapoints. We can see the distance (lines) between m_{23} and its closest datapoints, overall, are longer (which corresponds to the score $D(m_{23}) = 6.32$) than the ones emerging from s and e (which correspond to the scores $D(s) = 2.8$ and $D(e) = 2.07$ respectively (Panel (a))). In Panel (d) the circles around the nodes refer to the boundary r. We can see less datapoints within the boundary of m_{23} (which corresponds to the score $B(m_j) = 32$) than the boundaries of s and e (which correspond to the scores $B(s) = 182$ and $B(e) = 272$ respectively (Panel (c))). The results in fig. 5.5 showed that m moved between two dense areas, i.e. m moved between clusters, where the *clust* condition was satisifed when $j = 2$. Thus, the D and B scores correctly showed, when used with the condition *clust*, that s and e represent different clusters.

We next tested this condition using the data from (EX2)(fig. 5.4, Panel (b)), i.e. when s and e are in the same cluster, and, we show the results in fig. 5.6. Panel (a) shows the $D(m_j)$ score, and Panel (b) shows the $B(m_j)$ score, at 40 positions between s and e. We, again, calculated all $D(m_j)$ scores using $X = 125$, as the

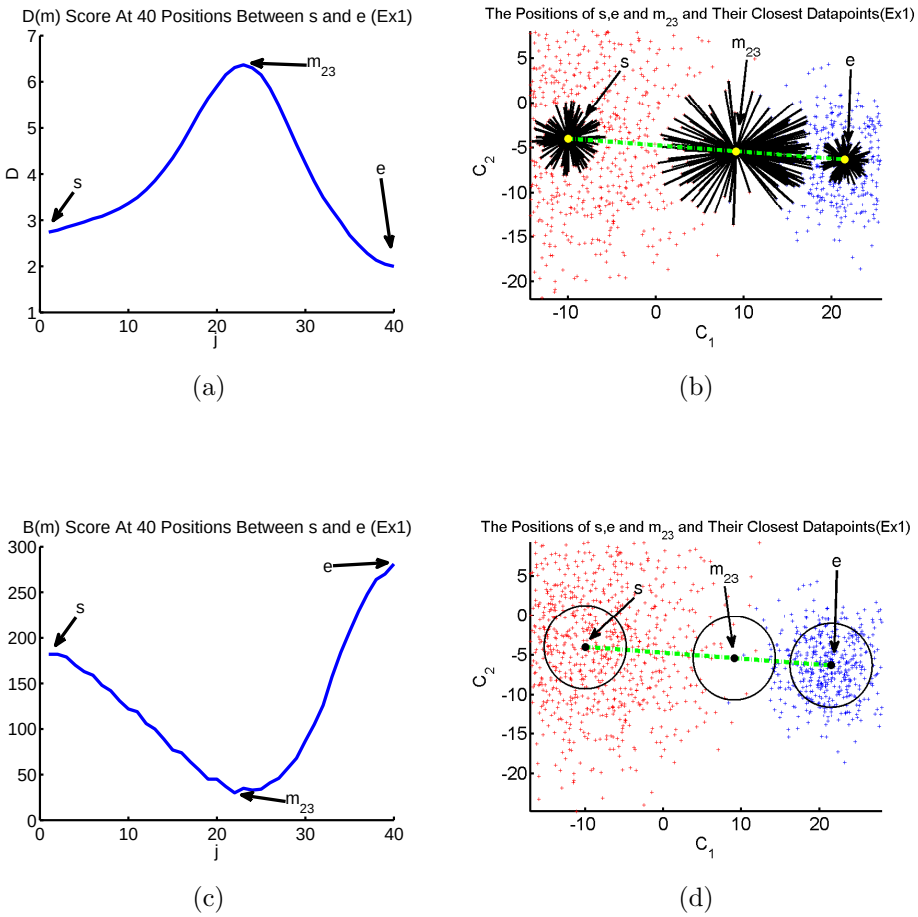
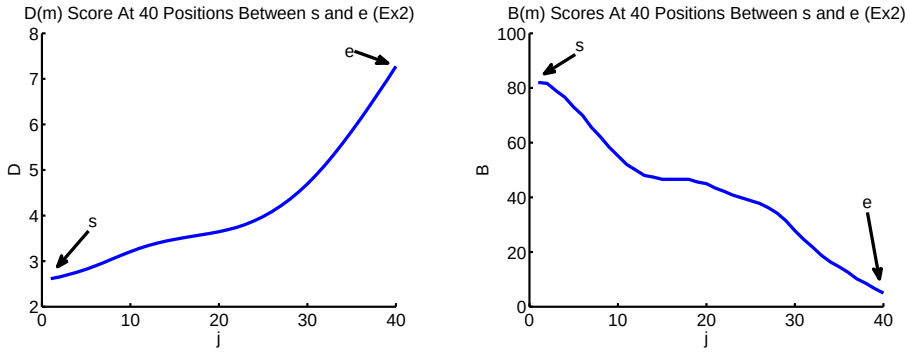


Figure 5.5: These results show that, using the dataset (EX1)(fig. 5.4, Panel (a)), s and e are correctly identified as representing different clusters. **Panel (a)** shows the $D(m_j)$ score, and **Panel (c)** shows the $B(m_j)$ score, at 40 positions between s and e. We calculated each $D(m_j)$ score using $X = 125$, and each $B(m_j)$ score using $r = 5.12$. Both r and X were acquired using the calculations discussed earlier. X was calculated from the smallest cluster, which contained 500 datapoints. Furthermore, r was the same for s, e and m and was calculated using s and e with equation 5.8. The results show as j increased to $j = 23$, i.e. when m moved from s, the $D(m_j)$ score increased from 2.8 ($D(s)$) and peaked at 6.32 (**Panel (a)**). Furthermore, $B(m_j)$ score decreased from 182 ($D(s)$) to 32 (**Panel (c)**). When $j = 23$ increased to $j = T$, i.e. when m moved from $j = 23$ to e, $D(m_j)$ decreased to 2.07 ($D(e)$) (**Panel (a)**) and $B(m_j)$ increased to 272 ($B(e)$)(**Panel (c)**). We note that $D(m_j)$ peaked, and $B(m_j)$'s lowest score was, at $j = 23$, as it was the least dense area between the two dense areas (clusters). The results show that m moved between two dense areas, i.e. m moved between clusters, where the *clust* condition was correctly satisifed when $j = 2$. **Panels (b) and (d)** show the positions of s, e and m_{23} in the feature space and the datapoints used to calculate s,e and m_{23}'s D and B scores. The nodes are represented as circular markers. The lines in **Panel (b)** refer to the distance between the nodes and its X closest datapoints. We can see the distance (lines) between m_{23} and its closest datapoints, overall, are longer than the ones emerging from s,and e. The circles around the nodes in **Panel (d)** refer to the boundary r. We can see there are less datapoints within the boundary of m_{23} than the boundaries of s and e.

minimum cluster size was 500. However, we used $r = 4.89$ for the m_j, s and e B scores, calculated using s and e with equation 5.8. The results show as j increased, i.e. when m moved from s to e, the $D(m_j)$ score increased from 2.76 ($D(s)$) to 7.3 ($D(e)$) (Panel (a)). Further, the $B(m_j)$ score decreased from 83 ($B(s)$) to 7 ($B(e)$) (Panel (b)). The results showed that m moved continuously through a less dense area than the area e was in. That is, the B and D scores at each position between s and e did not satisfy the *clust* condition, and therefore, correctly showed that s and e were in the same cluster.



(a) (b)

Figure 5.6: These results show that, using the dataset (EX2) (fig. 5.4, Panel (b)), s and e were correctly identified as representing the same cluster. **Panel (a)** shows the $D(m_j)$ score, and **Panel (b)** shows the $B(m_j)$ score, at 40 positions between s and e respectively. We, again, calculated all of the $D(m_j)$ scores using $X = 125$ as the minimum cluster size was 500. However, we used $r = 4.89$ for the s, e and m B scores, calculated using s and e with equation 5.8. The results show as j increased i.e. when m moved from s to e, the $D(m_j)$ score increased from 2.76 ($D(s)$) to 7.3 ($D(e)$) (**Panel (a)**). Furthermore, the $B(m_j)$ score decreased from 83 ($B(s)$) to 7 ($B(e)$) (**Panel (b)**). The results showed that m moved through a less dense area compared to the area e was in. That is, the *clust* condition was not satisfied.

We have successfully demonstrated here that our condition *clust* and the D and B scores can be used to determine whether two nodes are in different clusters. We refer to the process of identifying the nodes that represent different clusters as *movingm* throughout the chapter. We summarise *movingm* in the next section.

The Algorithm To Identify Whether Two Nodes Represent Different Clusters (*movingm*):

We identify whether two nodes represent different clusters using the following algorithm, *movingm*:

1. **Add an extra output (node) m to the Kohonen network**
2. **Update node m to move from a start node s to a destination node e in $T = 15$ steps (positions):**i.e. 15 steps[7] from $W_{m_j} = W_s$ to $W_{m_j} = W_e$. More specifically, we:
   (a) **Calculate the $D(e)$ and $D(s)$ scores (using equation 5.7) and $B(e)$ and $B(s)$ scores (using equation 5.8 with s and e to calculate r).**
   (b) **Calculate the $D(m_j)$ (using equation 5.7) and $B(m_j)$ (where we use the same r value as in step (a) to calculate $B(m_j)$) scores for every new position.**
   (c) **For every position j m moves to, we analyse the newly calculated $D(m_j)$ and $B(m_j)$ scores using the *clust* condition.**
   (d) **Quit process: when condition *clust* is satisfied before $W_{m_j} = W_e$.** Otherwise repeat steps b) and c)

We next show the effects of varying X, r and T on the *movingm* procedure.

The Effects Of Varying X, r and T on *movingm*

As we stated earlier, different values for X, r and T can produce $D(m_j)$ and $B(m_j)$ scores that suboptimally describe the density of the datapoint distribution. We investigated the effects of varying X, r and T on the outcome of *movingm*, i.e. whether the condition *clust* is correctly satisfied when these parameters vary. We discuss the results below.

We firstly tested *movingm* on the dataset in fig. 5.4 Panel (b)(EX2), i.e. when s and e were in the same cluster, using small X and r values. We show the results in

[7]We used $T = 15$ steps as we found this was sufficient in many tests, i.e. on every test we performed, s and e were correctly determined as being in different clusters. In addition, had lower computational costs compared to using higher T values. We show the effects of different T values in the next section.

fig. 5.7. The figure shows the $D(m_j)$ score when $X = 10$ (Panel (a)), and the $B(m_j)$ score when $r = 0.5$ (Panel (c)) at 40 positions between s and e. The results in both panels show the $D(m_j)$ and $B(m_j)$ scores fluctuated more erratically as j increased compared to the scores produced when $X = 125$ and $r = 5.19$ (see fig. 5.5 Panels (a) and (c) in the last section). Furthermore, the $B(m_j)$ score at $j = 4$ was $<$ both $B(s)$ and $B(e)$, and, the $D(m_j)$ score at $j = 33$ was $>$ both $D(s)$ and $D(e)$. In other words, the B and D scores at position $j = 4$ and $j = 33$ respectively incorrectly satisfied the *clust* condition, and therefore, showed s and e were in different clusters. The scores fluctuated and the condition was incorrectly satisfied because the datapoints used to calculate the scores did not accurately represent the density of the cluster at each of the j positions m was in between s and e.

We next tested *movingm* on the dataset fig. 5.4 Panel (a)(EX1), i.e. when s and e were in different clusters, using large X and r values. In fig. 5.7 we show the $D(m_j)$ score when $X = 1000$ (Panel (b)), and the $B(m_j)$ score when $r = 20$ (Panel (d)), at 40 positions between s and e. The panels show that at each position j, $D(m_j)$ was never $> D(e)$ (fig. 5.7, Panel (b)), and, $B(m_j)$ was never $< B(e)$ (Panel (d)). These scores incorrectly show that both s and e were in the same cluster. This happened because $X >$ the number of datapoints contained in the smallest cluster, which in this case was 500 (the cluster that e represents). Furthermore, $r >$ the variance of the smallest cluster, which had a standard deviation of 4. In other words, the D and B scores for m and e were calculated from the datapoints of both clusters, where m was always closer to the datapoints in the larger cluster where s resides. Therefore, at each of the j positions the D and B scores produced showed that m was always in a denser area than e.

We finally tested *movingm* on the dataset in fig. 5.4 Panel (a)(EX1), i.e. when s and e are in different clusters, using large and small T values. We calculated D, using $X = 125$ (calculated from the smallest cluster), and B, using $r = 5.19$ (calculated using equation 5.8 with s and e), at 40 positions between s and e. In fig. 5.8 we show the results when $T = 3$ and $T = 100$. Panel (a) shows the $D(m_j)$ scores, and Panel (c) shows the $B(m_j)$ scores, when $T = 3$. Panel (b) shows the $D(m_j)$ scores, and Panel (d) shows the $B(m_j)$ scores, when $T = 100$. All the results show that the nodes s and e were in different clusters, i.e. the *clust* condition was correctly

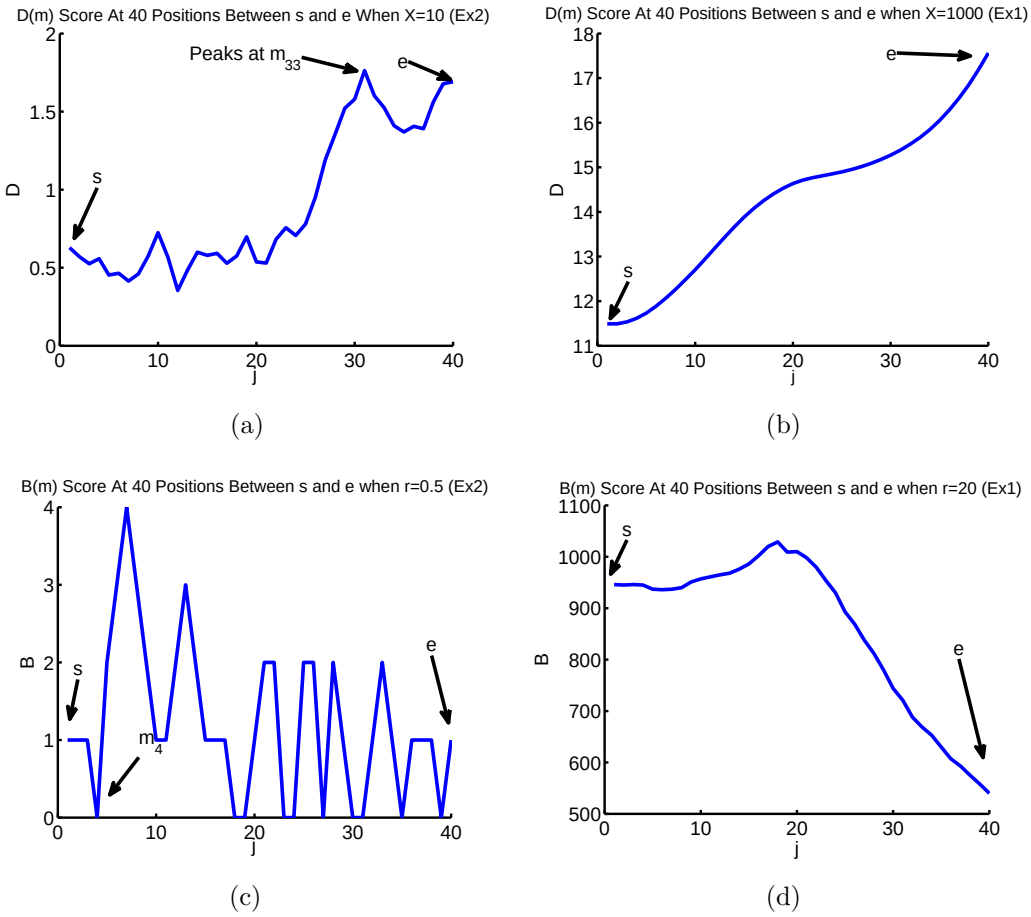


Figure 5.7: This figure shows the effects of varying X and r on the outcome of *movingm* when s and e were in the same cluster (Panels (a) and (c)) (i.e. when we used the (EX2) dataset (fig. 5.4 Panel (b))) and in different clusters (Panels (b) and (d)) (i.e. when we used the (EX1) dataset fig. 5.4 Panel (a)) . We calculated $D(m_j)$ and $B(m_j)$ at $T = 40$ positions. **Panel (a)** shows the $D(m_j)$ scores when $X = 10$ and **Panel (c)** shows the $B(m_j)$ scores when $r = 0.5$. The panels show the scores fluctuated more erratically as j increased compared to the scores when $X = 125$ and $r = 5.19$ (fig. 5.6, Panels (a) and (b) respectively). The results also imply that s and e were in different clusters. That is, the D score at $j = 33$ i.e. $D(m_{33})$, was $>$ both $D(s)$ and $D(e)$ (**Panel (a)**) and the B score at m_4, i.e. $B(m_4)$, was $<$ both $B(e)$ and $B(s)$ (**Panel (c)**). **Panel (b)** shows the $D(m_j)$ scores when $X = 1000$ and **Panel (d)** shows the $B(m_j)$ scores when $r = 20$. The results in **Panels (b) and (d)** incorrectly imply that s and e were in the same cluster. That is, at each position j, $D(m_j)$ was never $> D(e)$ (**Panel (b)**) and $B(m_j)$ was never $< B(e)$ (**Panel (d)**).

satisfied. That is, when $T = 3$ and $j = 2$, $(D(m_2) > D(s)) \wedge (D(m_2) > D(e))$ (Panel (a)) and $(B(m_2) < B(s)) \wedge (B(m_2) < B(e))$ (Panel (c)). Furthermore, when $T = 100$ and $j = 58$ (example position), $(D(m_{58}) > D(s)) \wedge (D(m_{58}) > D(e))$ (Panel (b)) and $(B(m_{59}) < B(s)) \wedge (B(m_{59}) < B(e))$ (Panel (d)). The results also show, when $T = 100$, a smoother transition in the $D(m_j)$ and $B(m_j)$ scores as j increased. That is, $T = 100$ provided a more accurate description of the datapoint distribution between s and e compared to when $T = 3$. However, sometimes a more accurate description can be a pitfull. For example, when m moves from s to e, the D and B scores produced may show very dense abnormal areas in the feature space, which may incorrectly satisfy the *clust* condition. We show an example of this in fig. 5.9, where s and e are in the same cluster. The $D(m_j)$ score at position $j = 98$ shows a slightly lower density than $D(e)$. Therefore, $D(m_{98})$ is $>$ both $D(s)$ and $D(e)$, incorrectly satisfying the *clust* condition, i.e. showing that s and e are in different clusters. In addition, when $T = 3$, D and B may incorrectly not satisty the *clust* condition. For example, as m only moves in this instance from s in 2 steps, and if s and e are in two clusters, m could leap over the area separating the clusters. This could, therefore, imply incorrectly that s and e are in the same cluster, i.e. the condition *clust* is not satisfied.

Generally, there are no optimal values for X, r and T as a value used in one dataset may have problems in another. In this section, we showed the values used for X, r and T should not be too large or small as they can affect the outcome of *movingm*. That is, the D and B scores can incorrectly satisfy the *clust* condition. The r and X generated using our guidelines, and, using $T = 15$, was sufficient in all our tests. That is, the nodes were always correctly identified as residing in the same or different clusters. We note that using a larger T value can be computationally inefficient, especially if many network outputs are used. That is, if many nodes, e.g. > 30, are in set Z we would have to investigate the datapoint distribution between many nodes in the set to identify the ones that represent different clusters. For example, when we used SOMA on datasets containing 40000 5-D datapoints the whole SOMA process took 25mins when $T = 100$ and 10mins when $T = 20$. These times were produced by SOMA when we used 40 outputs and all the above default parameter values were used.

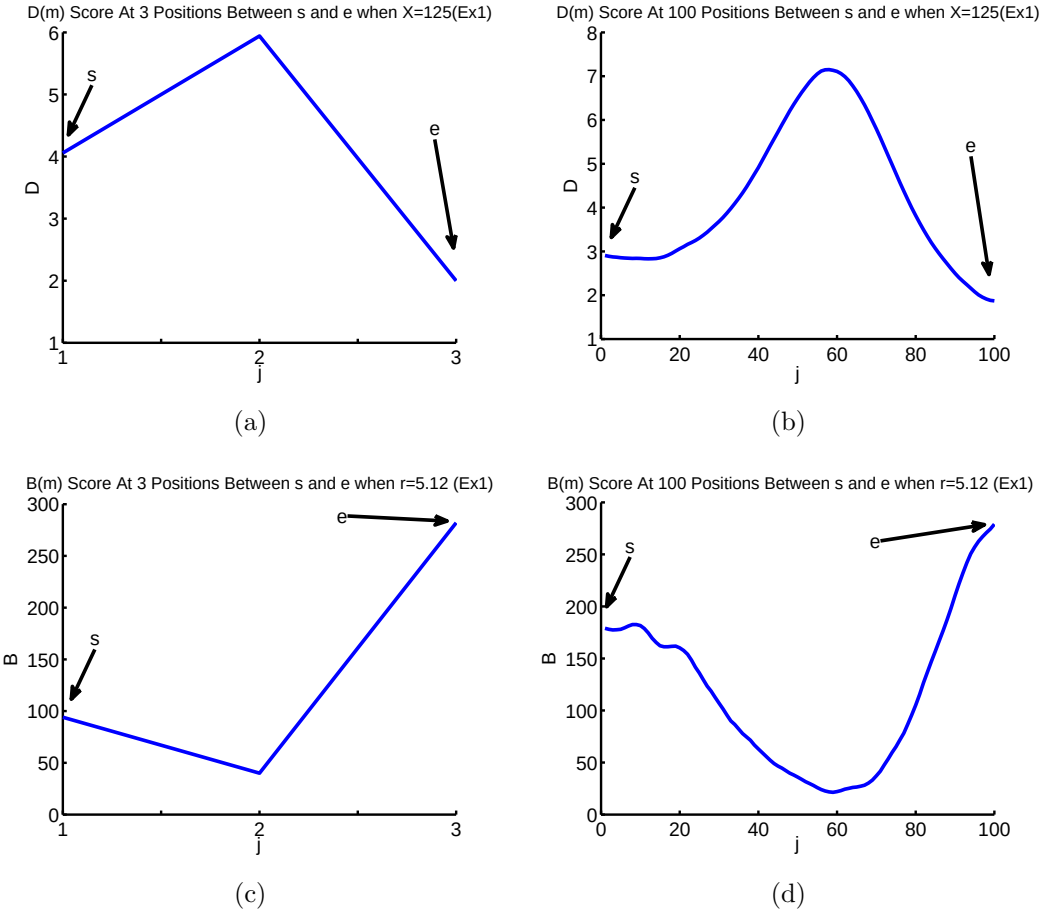
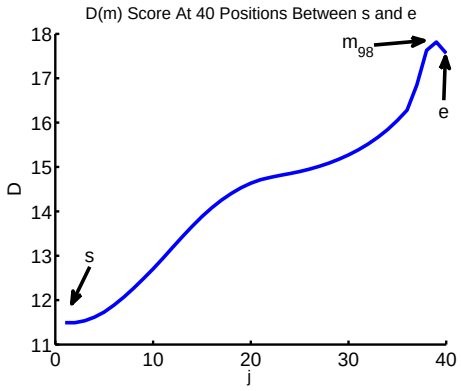


Figure 5.8: This figure shows the effects of varying T on the outcome of *movingm* when s and e were in different clusters. For this investigation we used the (EX1) dataset (5.4 Panel (a)). We calculate $D(m_j)$ when $X = 125$, and $B(m_j)$ when r=5.12. **Panel (a)** shows the $D(m_j)$ scores, and **Panel (c)** shows the $B(m_j)$ scores when $T = 3$. **Panel (b)** shows the $D(m_j)$ scores, and **Panel (d)** shows the $B(m_j)$ scores, when $T = 100$. The sets of $D(m_j)$ and $B(m_j)$ scores show the same regardless of T. That is, when $T = 3$ at $j = 2$ (**Panel (a)**) and $T = 100$, e.g. at $j = 58$, **Panel (b)**, $D(m_j)$ was $>$ both $D(s)$ and $D(e)$, and, $B(m_j)$ is $<$ both $B(s)$ and $B(e)$. These results correctly show that s and e were in different clusters. Furthermore, when $T = 100$, the panels show a smoother transition in the $D(m_j)$ and $B(m_j)$ scores as j increased (Panels (b) and (d)). That is, $T = 100$ provides a more optimal description of the datapoint distribution between s and e.



(a)

Figure 5.9: This figure shows the effect of using $T = 100$ on the outcome of *movingm* when s and e were in the same cluster. For this investigation we used a modified version of the (EX2) dataset (fig. 5.4 Panel (b))). That is, we formed a less dense area of datapoints near e. The $D(m_j)$ score at $j = 98$ incorrectly suggests that s and e are in different clusters.

In this chapter, we use our guidelines to calculate X and r, and, we use $T = 15$ with SOMA, unless otherwise stated.

Identifying the Central Nodes From Z

To determine whether all of the central nodes in Z reside in different clusters, we need to determine that the density of the distribution between all of the nodes in Z in the feature space is less than the density of the areas each of the nodes reside in. We refer to the set of central nodes as R and the rest of the nodes as set E. To acquire set R, we use the above procedures and implement the following:

1. Acquire a set L, by sorting the nodes in Z into D ascending order.
2. Identify the nodes from L that represent the same clusters.
   (a) Select the s node from L that has the lowest D score.
   (b) Append s to R.
   (c) Remove s from L
   (d) Use the procedure *movingm* with s and every node in L, where L corresponds to a set of e nodes. If the condition *clust* is not satisfied, i.e. s and e are in the same cluster:
      i. Remove e from L.

ii. Append e to E.

(e) Repeat steps (a)-(d) until L is empty.

In other words, R contains the correct central nodes and E is the set of nodes residing in the clusters around the central nodes. We refer to E as the set of outer nodes. We use the central nodes R and set E in extension 2 to reduce misclassification.

Extension to the Kohonen Network II (Reducing Misclassification)

This section describes a further extension to the Kohonen network clustering process, which uses the nodes in the R set and the nodes in the E set to reduce misclassification. Briefly, classification is implemented through a Kohonen network by clustering together the datapoints closest to each node. In other words, the classification procedure forms hyperplanes between each of the nodes, which, form implicit regions surrounding the nodes. Thus, any datapoints within the region surrounding the node are clustered together.

Using only the central nodes in the classification process[8] can result in hyperplanes not separating the clusters sufficiently (i.e. do not form regions which fully encapsulate each cluster), thereby causing misclassification. We show an example of this in fig. 5.10 Panel (f), where a subset of the larger cluster would be misclassified as it resides on the wrong side of the hyperplane (boundary). In other words, some of the points are closer, with respect to the hyperplane, to the incorrect node, and therefore, classified to it.

Therefore, using, and moving (updating) the extra nodes towards the clusters' edges, repositions the hyperplanes between the central nodes to enhance cluster separation (fig. 5.10 Panel (e)), which therefore, form better implicit regions.

[8]We note that this is the equivalent of using the standard Kohonen self organising map clustering method. That is, the standard method assumes that the number of outputs provided by the user is the number of clusters in the feature space, and therefore, when the network is trained, the nodes, if the number of them equates to the number of clusters in the feature space, would correspond to our set of central nodes.

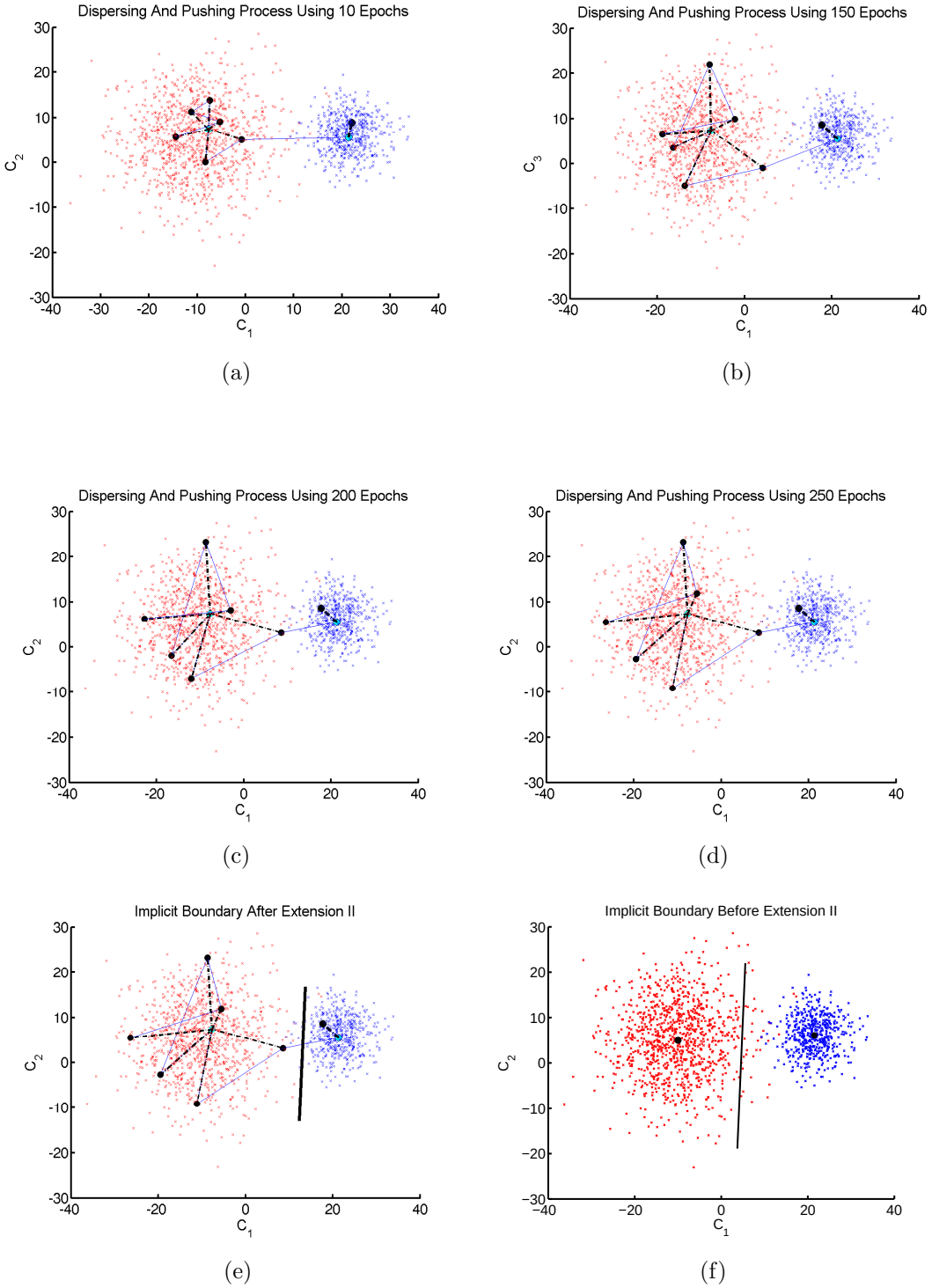


Figure 5.10: **Panels (a)-(f):** These show the results of the pushing and dispersing stages. It shows the process has continuously pushed and spread out the clusters' outer nodes, i.e. sets O_R, over several epochs. **Panel (e):** This shows the result of the SOMA process when used with all 10 nodes, where the hyperplane (black line) formed from the outer nodes (sets O_R) improves cluster separation. **Panel (f):** This shows the result of the SOMA process when only the central nodes are used. The hyperplane (black line) formed between them does not sufficiently separate the clusters.

To achieve a reduction in misclassification we use all of the nodes in E and the central nodes in the classification procedure to form better hyperplanes (regions). In other words, as it is likely that subsets of the nodes in E are spread around each of the central nodes in each cluster, we can identify which nodes in E reside in the same cluster as each of the central nodes, and then, cluster together all of the datapoints closest to each of the nodes (i.e. cluster together all of the datapoints closest to each of the nodes in subset E or the central node,) in the same cluster. Using the extra nodes with the classification process may not reduce misclassification immensely as the first stage (Kohonen training) may not distribute the nodes to represent every cluster's area sufficiently. Therefore, we move the non-central nodes (i.e. the nodes in set E) further into and around their cluster's outer area. This should reduce misclassification as it will move the nodes as close to the clusters edges as possible, and therefore, improves cluster separation, i.e. form better implicit regions surrounding the clusters. To achieve a reduction in misclassification we implement the following:

1. **The Identification Process:** Associate a set of nodes O_R, a subset of E, to every node in the R set. Thus, a node from R is a central node of a cluster and its corresponding O_R set represents the outer nodes of the same cluster.

To achieve more optimal hyperplanes, i.e. better cluster separation, we use a second set of 250 epochs during which we use a novel algorithm that first pushes and then disperses the nodes in each O_R set out, and around, their corresponding cluster. We designed these two techniques to slowly push and space out the nodes in an O_R set, preventing the confinement of node subsets to one area. Therefore, an entire cluster area, regardless of shape and size, is sufficiently represented. To visualise this extension, we show an example of the pushing and dispersing stages in fig. 5.10 Panels (a)-(d).

From our experience, there exists no optimal choice for the number of epochs to run this process, but 250 epochs when used with the learning rate (discussed later) and extension 2, generally spread the nodes out more optimally around the clusters. However, this may need to be lowered if using large datasets, as it can be

computationally expensive. We discuss the outcome of extension 2 later. For every epoch we implement the following:

2. **The Pushing Process:** At the start of every epoch, we push a set of nodes, which contains a node from every O_R set, further towards their cluster's edge.

3. **The Dispersing Process:** For the rest of the epoch, we use the Kohonen training process to disperse the outer nodes (the O_R sets) around their cluster.

Once these stages are complete, we classify each of the datapoints using the new classification process described below:

4. **The Classification Process: Classify each datapoint to a node in set R.** We identify the closest node to a datapoint using the euclidean distance measure. The identified node is either a central node in R or from an O_R set, thus, we classify the datapoint to the corresponing node in R.

We explain the first three processes in more detail below:

The Identification Process

To identify a set of nodes O_R for every node in R, we find the nodes that are within the same cluster as R, via a two stage-process:

1. **Identify the node in R to which each node in E is closest. Therefore, we form a set of nodes Q for every node in R:** There is a high chance the nodes in set R and the nodes in their corresponding Q set represent the same cluster.

2. **Verify that every node in R and its corresponding Q set represent the same cluster:** We achieve this using the *movingm* procedure (Extension I, Stage 2) to verify the nodes s, which is a node from R, and e, which is a node from the corresponding Q set, represent the same cluster, so:

i. We append e to the corresponding O_R set if the condition *clust* in *movingm* is not satisfied, i.e. s and e are in the same cluster.

ii. If e does satisfy the condition *clust* we test it against the other nodes in R. We ignore it if on every test the condition *clust* is satisfied.

The Pushing Process

At the start of every epoch, we choose a node from every set O_R, and push them further towards their cluster's edge.

We achieve this pushing process by initally associating the same r value, calculated using the equation 5.8, to pairs of nodes, i.e. a node from R and a node in its O_R set. Throughout the epochs, the r value for the outer nodes remains the same and we retain the central nodes within the cluster centres, but, increase their boundaries so that they correspond to the distance between the central node and the chosen outer node. We then push out the outer node so that their boundaries, i.e. the boundaries of the central nodes and the outer nodes, do not overlap. This process continues until all the outer nodes (O_R sets) represent an area near their cluster's edge, i.e. a stopping condition is met. We show this process visually in fig. 5.11.

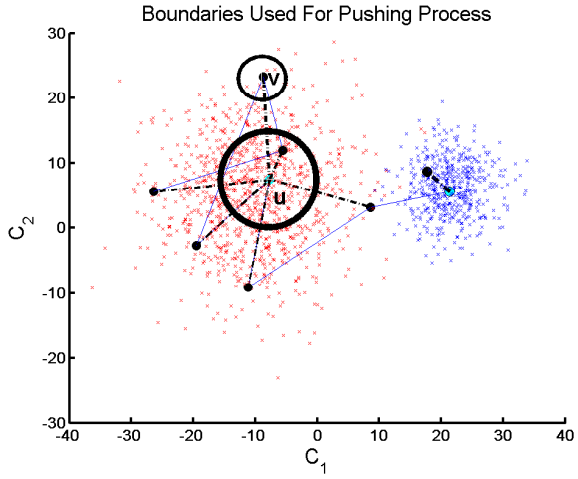


Figure 5.11: This shows the boundaries used in the pushing process for a central node u and a node from u's corresponding O_R set v. The boundary surrounding the central node u increases, and v is pushed out, over time. The pushing process continues until a stopping condition is satisfied, i.e v is near the cluster's edge.

The r value associated to the outer node is the distance we push out the node each

time. We use r as the pushing distance due to it being based on the cluster size, i.e. the initial distance between the central and outer node. Thus, if there is, initially, a small distance between them, we assume that they reside in a small cluster so we want to push the outer nodes out in small steps. If the initial distance between them is large we assume they are in a larger cluster so we push the outer nodes out in larger steps. We implement the pushing process in this way, i.e. increase the central nodes boundaries, so that we know how far from the centre we have the pushed out the nodes. In other words, we exclude the datapoints positioned within the central nodes' boundaries from updating the outer nodes' in the dispersing stage. Thus, a outer node's movement is confined to its cluster's outer area.

As clusters can vary in shape and size, we need to push some nodes out further than others. For example, if a cluster had an elongated spherical shape, we would need to push some nodes further out so that they represent the edges of the elongated sections compared to the nodes that represent the reduced sections. To achieve this, we associate a set of boundaries to every node u in R, where the size of the set corresponds to the number of nodes in u's O_R set.

To achieve this pushing process we:

1. **Associate a set of boundary values I to every central node u in R, where the number of boundaries in I is equal to the number of nodes in u's O_R set. We also associate an r value to each node in O_R.** If we refer to a node in u's O_R set as v, then the boundary value of v associated to u is I_v, which refers to the v^{th} element of I. I_v's and v's r value is initially the same and calculated using equation 5.8 with nodes u and v. During extension 2, I_v will be updated in the pushing process (explained below) and the value r associated to v stays the same throughout training.

2. **Identify and push out a node from every set of O_R:** We implement this at the start of every epoch. This is described in more detail below:

Identifying and pushing a node from an O_R set

To achieve this, we:

1. Identify a node v from u's O_R set, which is closest to u.

2. Quit the process: if v satisfies the condition *pushstop* explained in the next section.

3. Increase the boundary value I_v, i.e. the boundary value of u associated to v using:

$$I_v = d(W_u, W_v) \tag{5.9}$$

4. Update node v's weight vector i.e. push out node v so the boundaries of v and u do not overlap using:

$$W_v(t+1) = W_v(t) - (W_u(t) - W_v(t))(\frac{r}{I_v}) \tag{5.10}$$

where r refers to the boundary around node v and t refers to the t^{th} epoch.

Verifying a node's new position

We need to verify the pushing process will not push the chosen node v out from, or into another, cluster. Thus, we do not increase u's boundary value I_v or push out v if v_n, which we refer to as v's new (push) position, satisfies the condition *pushstop* below.

The Process And The Condition To Prevent Nodes Being Pushed Into Another Cluster

To determine whether v will be pushed into another cluster we calculate two B and two D values for v_n. The first B and D values refer to the datapoint distribution in between v_n and u, and the second set of D and B values refer to the datapoints behind v_n. Therefore, if the density of the datapoint distribution behind v_n is $>$ the density in front of v_n, we do not increase u's I_v value or push out v to v_n. This is because the higher density behind v_n suggests the area behind v is another cluster. Furthermore, we do not push v or increase u's I_v value if the area behind v_n contains no datapoints. To describe this procedure we refer to v_n^b as the datapoints behind v_n and v_n^f as the datapoints in front of v_n.

Calculating the $B(v_n^f)$ score

To derive the $B(v_n^f)$ score, we implicitly divide the spherical boundary i.e. r, surrounding v_n through the middle into two halves, where the division line is perpendicular to the line between v_n and u (central node). Thus, the number of datapoints within the half sphere facing s (We show this graphically in fig. 5.12) constitutes the $B(v_n^f)$ score.

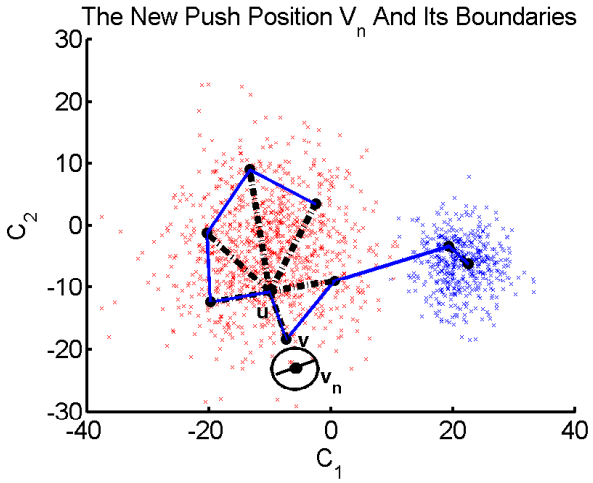


Figure 5.12: This shows the boundary used to calculate the $B(v_n^f)$ and $B(v_n^b)$ scores. The marker in the middle represents the central node u. The node v residing in the outer area of the cluster represent a node from u's O_R set. The v node is the one selected to be pushed and v_n is the new push position of v. The circular sphere surrounding node v_n, and the line through the middle, represent the half boundaries used to calculate $B(v_n^f)$, i.e. we calculate $B(v_n^f)$ using the datapoints within the half boundary facing u, and $B(v_n^b)$, i.e. we calculate $B(v_n^b)$ using the datapoints within the half boundary behind v_n.

To identify if a datapoint's position is within the correct half of the boundary, we implement the following:

1. **Calculate an angle α that identifies if datapoint x is positioned within u and v_n:** We calculate angle α using W_{v_n}, W_u and x (datapoint vector), so if $\alpha < 90$, x is between them. (Explained below).

2. **Identify whether the datapoint x is also within the specified boundary r:** i.e. $(d(W_{v_n}, x) < r)$, where $d(W_{v_n}, x)$ represents the euclidean distance between W_{v_n} and x.

Calculating The Angle α

To do this, we need to rearrange the cosine law (shown below):

$$p^2 = z^2 + l^2 - 2zl \cdot cos\alpha \tag{5.11}$$

where $z = d(W_{v_n}, x), l = d(W_u, W_{v_n}), p = d(W_u, x)$.

Calculating The $B(v_n^f)$ Score

The $B(v_n^f)$ score is the number of datapoints that satisfy

$$(d(W_{v_n}, x) < r) \wedge (\alpha < 90)$$

Calculating the $B(v_n^b)$ score

The datapoints used to calculate $B(v_n^b)$ have to satisfy

$$(d(W_{v_n}, x) < r) \wedge (\alpha > 90)$$

i.e. the datapoints within the boundary r behind v_n. We show these boundaries graphically in fig. 5.12.

Calculating The $D(v_n^f)$ and $D(v_n^b)$ Scores

We calculate the $D(v_n^f)$ using equation 5.7 where the closest datapoints to derive the scores have to be between u and v_n, i.e. $\alpha < 90$. We calculate $D(v_n^b)$ using the closest datapoints behind v_n with respect to u, i.e. $\alpha > 90$.

Condition *pushstop*

We do not increase I_v associated to both v and u, or, push out v to v_n if

$$((B(v_n^b) > B(v_n^f)) \vee (D(v_n^b) < D(v_n^f)) \vee (B(v_n^b) == 0))$$

In other words, we do not increase I_v or move v to v_n if the area behind v_n is more dense than the area in front of v_n.

The Dispersing Process

For the rest of the epoch, we disperse all non-central nodes (sets O_R) around their cluster's outer area.

To achieve this, we use and modify the Kohonen training process, so:

1. We no longer update the winning nodes adjacent neighbours.
2. A winning node is from one of the O_R sets.
3. If the current datapoint is not within the centre or outside the winning node's cluster (explained below) we move (update) the winning node.

Identifying And Moving The Winning Node

To achieve this, we:

1. Identify a winning (closest) node f for every datapoint x.
2. Do not update f using x, i.e. do not move f closer to x, if x is:
   (a) Within the boundary of the corresponding central node u, i.e. within I_f.
   (b) Behind the winning node f with respect to u, i.e. if $\alpha > 90$. We rearrange equation 5.11 to calculate α, where $z = d(W_f, x), l = d(W_u, W_f), p = d(W_u, x)$

   These constraints confine f's movement to its cluster's outer area. Therefore, we update W_f, i.e. move f, if:

$$(d(W_u, x) \geq I_f + r) \wedge (\alpha \leq 90)$$

   using x, where r is the boundary value for f. We add r to prevent u and f's boundaries overlapping, i.e. to prevent f from moving nearer the central node f which would negate the effect of the pushing process.

The above process is shown in fig. 5.13. The figure shows an example of the datapoints the process excludes from updating a node f (left cluster).

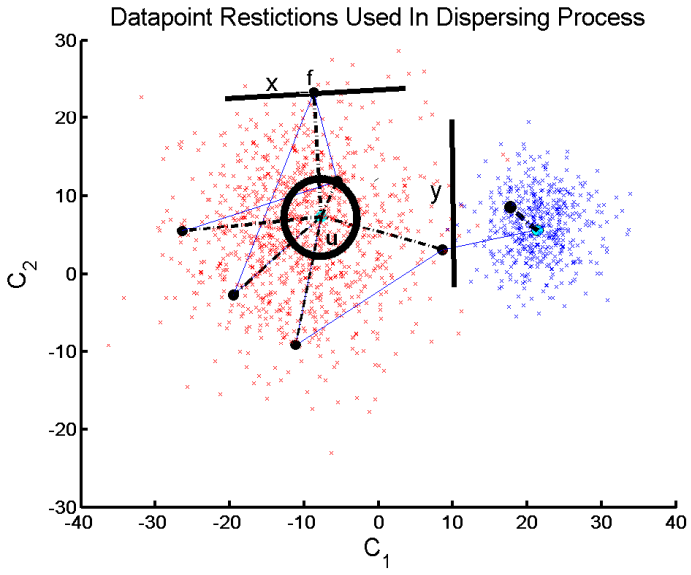


Figure 5.13: This shows the datapoints to which node f (from an O_R set) can move towards. That is, f can move between the boundaries: line (x), which is perpendicular to the line between f and central node u, and (I_f) (where I is the boundary set associated to u) the circle surrounding the central node u. These boundaries change overtime, i.e. (I_f) increases in size, and line (x) changes with respect to u, as f moves around the cluster's outer area. Any changes in the boundaries define new datapoints, which the dispersing process temporarily excludes. If node f moved to the cluster's right side its line boundary would change, e.g. Line (y).

5.5 Results Generated From Using The Spike Sorting Process On Simulated Data

To test the efficiency of our clustering method we created 10 simulated datasets in the same way as discussed in section (5.4.1), each containing 2 or 3 normally distributed clusters, which were made from either 10000 2-D, 40000 5-D or 40000 20-D datapoints. In addition, the clusters varied in size, position and standard deviation. Table 5.3 states the contents of each dataset.

Firstly, we investigated the effectiveness of, i.e. determined the usefulness of, our extensions to the Kohonen's self organising map clustering method[9]. To investigate this, we used SOMA with each dataset and after each stage of the process we classified the data using the network. In other words, after initial training we classified the data using only the central nodes (set R)[10] after extension 1 we classified the

[9]We note that in all our tests SOMA identified the correct number of clusters, and therefore, we only report the percentage of the datasets SOMA correctly classified.

[10]Using only the central nodes, i.e. one node per cluster, is the equivlent of using the standard

Dataset	**ND**	**D**	**NC**	**CI**	**Dist**
1	10000	2-D	2	Cluster 1 (5000)(8) Cluster 2 (5000)(4)	23.2
2	10000	2-D	2	Cluster 1 (5000)(8) Cluster 2 (5000)(4)	20
3	10000	2-D	2	Cluster 1 (5000)(8) Cluster 2 (5000)(4)	18
4	10000	2-D	2	Cluster 1 (5000)(4) Cluster 2 (5000)(4)	20.2
5	10000	2-D	2	Cluster 1 (5000)(4) Cluster 2 (5000)(4)	17
6	10000	2-D	2	Cluster 1 (5000)(4) Cluster 2 (5000)(4)	13.3
7	10000	2-D	3	Cluster 1 (5000)(1.5) Cluster 2 (5000)(1.5) Cluster 3 (5000)(1.5)	Avg 7.2
8	10000	2-D	3	Cluster 1 (5000)(1.5) Cluster 2 (5000)(1.5) Cluster 3 (5000)(1.5)	Avg 6.4
9	40000	5-D	2	Cluster 1 (20000)(8) Cluster 2 (20000)(4)	20.2
10	40000	20-D	2	Cluster 1 (20000)(8) Cluster 2 (20000)(4)	20.2

Table 5.3: This table shows the contents of each simulated dataset we used to test SOMA. G represents the dataset number. ND represents the number of datapoints in the dataset. D represents the number of dimensions, i.e. the datapoint vector size. NC represents the number of clusters. CI represents the cluster information. The first number in the bracket represents the number of datapoints in the cluster and the second number represents the standard deviation of the cluster. $Dist$ represents the euclidean distance between the cluster centres. For more than 2 clusters we show the average distance. The clusters in datasets 4,5,7,8 did not vary in shape and did not overlap. The rest contained clusters that either varied in shape and/or had some overlap.

data using the central and non-central nodes, and, after extension 2 (i.e. after the non-central nodes were spaced out more optimally around the clusters) we classified the data again using the central and non-central nodes. Thus, if the number of datapoints SOMA correctly classified increased with each extension used then, we can conclude the extensions are useful. We tested SOMA on each dataset 20 times, used 9 outputs (nodes), and calculated the percentage of the dataset SOMA correctly classified after each stage and averaged the percentage across runs. Throughout this section, when we discuss the percentage of the results we are referring to the average percentage over runs unless otherwise stated. We show the results in table 5.4. We found that using SOMA with datasets containing clusters with similar distributions, and, did not overlap (table 5.4 dataset 4,5,7,8), SOMA's performance was similar after each stage. That is, as the central nodes classified a high percentage of these datasets correctly, the extensions only slightly improved the classification result. However, when the cluster sizes varied and had some overlap, the results improved with each extension (table 5.4 datasets 1,2,3,6,9,10). For example, the percentage of datapoints SOMA correctly classified after each stage increased from 76% to 92% when we used dataset 3. The results show that cluster separation improved with each extension. The results also show that each stage took a similar time to complete, i.e. the time SOMA took to cluster the data increased with each extension used. The results suggest that using the 2 new extensions may increase performance but, as a consequence, increases computational costs. Finally, the results show that the entire clustering process took longer when we used large datasets and when the datapoints were represented in higher dimensions (table 5.4 datasets 9,10).

Secondly, we investigated whether increasing the number of outputs (nodes) used with SOMA would increase the number of datapoints SOMA correctly classifies. We investigated this by running SOMA with both extensions 1 and 2 and 20 outputs on datasets 1,2,3,6,9,10 (contents of the datasets shown in table 5.3) 20 times. We calculated the percentage of the datapoints SOMA correctly classified after each run and how long it took. We then averaged the results across runs. We show the average results after extension 2 in table 5.5. The results show that using more outputs (nodes) with SOMA reduced the number of datapoints misclassified (see

Kohonen clustering method with the same number of outputs as the number of clusters.

Dataset	**C1**	**T1**	**C2**	**T2**	**C3**	**T3**	**TT**
1	87%	1.3	93%	1.2	97%	1.3	3.8
2	79%	1.2	87%	1.1	95%	1.3	3.6
3	76%	1.3	83%	1.2	92%	1.2	3.7
4	100%	1.2	100%	1.0	100%	1.3	3.5
5	95%	1.3	98%	1.1	99%	1.2	3.6
6	91%	1.2	95%	1.2	97%	1.1	3.5
7	100%	1.1	100%	1.2	100%	1.3	3.6
8	96%	1.1	98%	1.2	100%	1.2	3.5
9	80%	4.0	88%	4.1	95%	4.2	12.3
10	79%	5.0	86%	5.2	94%	5.1	15.3

Table 5.4: This table shows the average percentage of the data in each dataset (contents shown in table 5.3) across 20 runs SOMA (using 9 outputs (nodes)) correctly classified after each stage of the process. In addition, we also show the average time (in minutes) SOMA took, across 20 runs, to complete each stage of the process. *Dataset* represents the dataset set we used. $C1$ represents the average percentage of the data SOMA correctly classified after the initial training stage (i.e. the percentage of the data SOMA correctly classified when we used only the central nodes in the classification process) and $T1$ represents the time (in minutes) the stage took to complete. $C2$ and $C3$ represent the average percentage of the datasets SOMA correctly classified after extension 1, i.e. using all the nodes (central and non-central nodes), and after extension 2 (i.e. using the central and non-central nodes once the non-central nodes were more optimally spaced out around the clusters) respectively. $T2$ and $T3$ represent the average time extension 1 and extension 2 took to complete, and TT represents the average total time, i.e. the time it took for all extensions to complete. The results show that for datasets containing similar sized clusters and were spaced apart (clusters 4,5,7,8) using the central nodes was sufficient for clustering. However, when we used SOMA on datasets containing clusters of varying size and overlapped (the rest of the datasets) the percentage of the datasets SOMA correctly classified increased with each extension. For example see dataset 3. We also note that the time SOMA took to complete clustering the data increased with each extension. We also note that the process took longer when SOMA was used with larger datasets, and, when the datapoints were represented in higher dimensions (datasets 9 and 10).

Dataset	**C**	**TT**
1	97%	4.7
2	97%	4.9
3	96%	4.8
6	97%	4.6
9	98%	15.3
10	96%	17.3

Table 5.5: This table shows the average percentage of the data in each dataset 1,2,3,6,9,10 (contents shown in table 5.3) across 20 runs SOMA (using 20 outputs) correctly classified after both extensions. In addition, we also show the time (in minutes) SOMA took, on average, across 20 runs, to complete the whole process. *Dataset* represents the dataset we used. C represents the average percentage of the data SOMA correctly classified after both extensions. TT represents the average total time. These results show that when we used SOMA with 20 outputs (nodes) it classified a higher percentage of the dataset correctly compared to using 9 outputs (nodes) (see table 5.4). For example, when we used dataset 10 with SOMA, it classified 96% and 94% of the dataset correctly when we used 20 and 9 outputs (nodes) (see table 5.4) respectively. However, SOMA took longer to finish when we used 20 outputs (nodes). For example, when we used SOMA with dataset 10, SOMA took 17.3 minutes to complete the process when used with 20 outputs but 15.3 minutes when we used 9 outputs (see table 5.4).

tables 5.4 and 5.5 for the results when we used SOMA with 9 and 20 outputs (nodes) respectively). For example, when we used dataset 3 containing two clusters that partially overlapped, SOMA classified 92% of the datapoints correctly using 9 outputs (nodes) (table 5.4), which increased to 96% when we used 20 outputs (nodes) (table 5.5). However, extension 1 took longer when we used more outputs in SOMA. In other words, SOMA took longer to identify the non-central nodes belonging to each cluster as SOMA had to analyse the distribution between more nodes. For example, using extension 1, 200 epochs and 40000 5-D datapoints with a network containing 9 and 20 outputs (nodes) took 12.3 minutes (table 5.4) and 15.3 minutes (table 5.5) respectively. The results suggest that increasing the number of outputs used may increase performance but, as a consequence, may increase computational costs.

Finally, we investigated whether using a grid like output (2-D topology) would affect the outcome and speed of SOMA. We used a 3x3 grid, i.e. 9 outputs (nodes), where the winning node's neighbours were updated in the same way as discussed

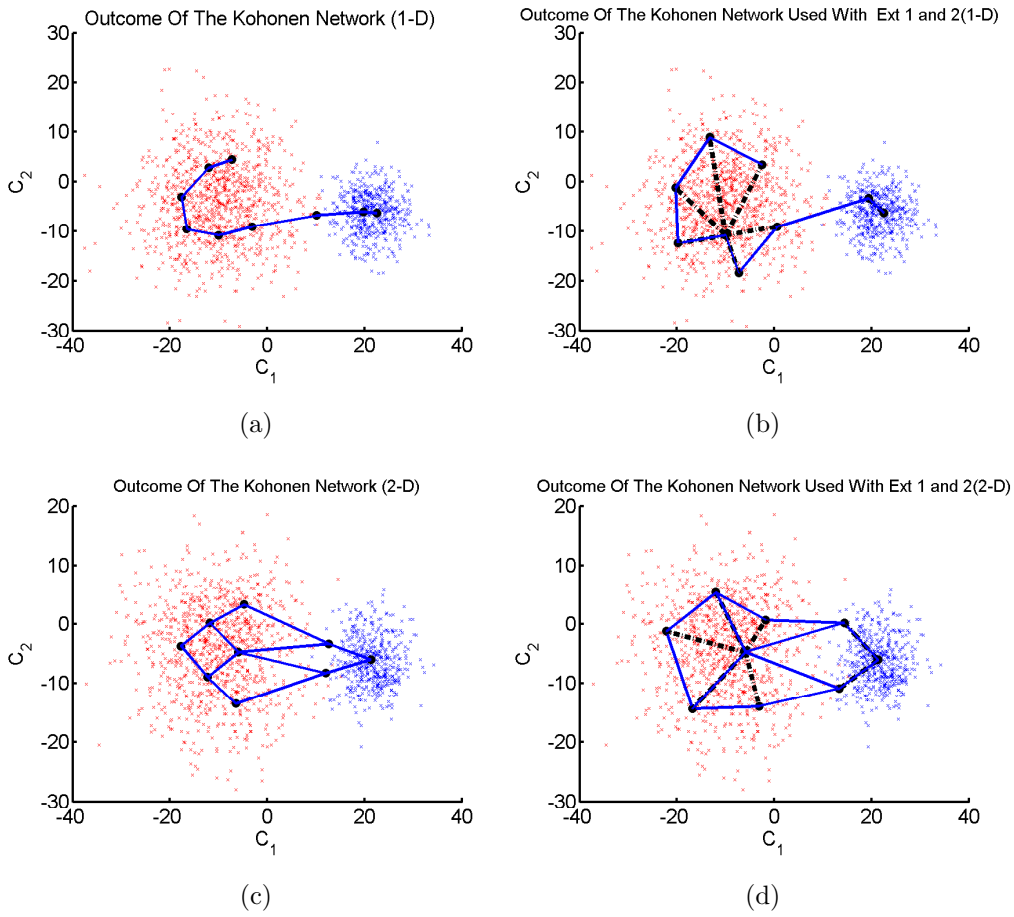


Figure 5.14: The figure shows the trained state of a Kohonen network with a 1-D and 2-D topology when used with dataset 1 after the initial training stage and both extensions. Panels (a) and (c) show the state of the 1-D and 2-D topologies, respectively, after the initial training stage. The circular markers represent the nodes and the thin line connecting them respresents the nodes' neighbours. The results show the nodes in a 2-D topology (Panel (c)) were more evenly spaced out after the initial training stage. **Panels (b) and (d)** show the state of the 1-D and 2-D topology networks, respsectively, after extensions 1 and 2. The markers in the middle of the clusters represent the identified central nodes, i.e. set R. Futhermore, the nodes connected to the central nodes with a striped line represent the central nodes corresponding outer node O_R sets. The results show that after extension 1 and 2 the nodes in both network topologies were spaced out more optimally, which, improved cluster seperation. That is, the network with a 1-D and 2-D topology correctly classified 97% and 98% of the dataset respectively

in the methods section. We used this network on datasets 1,2,3,6,9,10, i.e. the datasets containing clusters that varied in size and were close together (dataset contents shown in table 5.3). To determine whether a 2-D topology increases SOMA's performance, we classified the datapoints using the 2-D network after the initial training stage (i.e. using only the central nodes), after extension1 (i.e. using all the nodes (central and non-central nodes) and extension 2 (i.e. using the central and non-central nodes once the non-central nodes were more optimally spread out around the clusters). We tested this topology on each dataset 20 times and calculated the percentage of the data SOMA correctly classified after each stage and the time it took. We then averaged the results across runs. We show the average results in table 5.6.

Firstly, the results in tables 5.4 and 5.6 show that, overall, the percentage of the datasets SOMA correctly classified was similar after initial training (i.e. when SOMA classified the data using the central nodes) using either a 1-D or 2-D topology. However, the percentage of the data SOMA correctly classified after extension 1 was greater when we used a 2-D topology. For example, when we used dataset 1 with SOMA, it classified 95% and 93% of the dataset correctly when we used a 2-D and 1-D topology respectively. This is because the nodes in a 2-D topology tended to be more evenly distributed across the feature space after the initial training stage. We show a graphical example of this in fig. 5.14 Panels (c) and (a). The panels show a 1-D (Panel (a)) and 2-D (Panel (c)) network topology after they were trained (i.e. after the initial training stage) with dataset 1. We can see the nodes in a 2-D topology are more evenly spaced out (Panel (c)) compared to a 1-D topology (Panel (a)).

Secondly, the results in tables 5.6 and 5.4 show that, overall, the percentage of datapoints correctly classified by both network topologies increased when we used extension 2, but, both networks correctly classified a similar percentage of datapoints. For example, when we used dataset 1 with SOMA, it classified 98% and 97% of the dataset correctly when we used a 2-D and 1-D topology respectively. This is because, after the extension 2 stage, the nodes in both networks, regardless of topology, were more optimally spaced out. We can see this visually in fig. 5.14 Panels (b) and (d), which show the 1-D and 2-D networks after extension 2, when

Dataset	**C1**	**T1**	**C2**	**T2**	**C3**	**T3**	**TT**
1	88%	2.3	95%	1.3	98%	1.2	4
2	79%	1.9	91%	1.2	97%	1.3	4.1
3	70%	2.2	87%	1.5	93%	1.1	4.2
6	90%	2.1	95%	1.4	98%	1.3	4.1
9	82%	5.3	91%	4.2	95%	4.1	13.6
10	78%	7.5	92%	5.2	94%	5.3	18

Table 5.6: This table shows the average percentage of the datasets 1,2,3,6,9,10 (shown in table 5.3) across 20 runs SOMA (using a 3x3 grid of outputs (nodes) (i.e. a 2-D network topology)) correctly classified after each stage of the process. In addition, we also show the average time SOMA took (in minutes), across 20 runs, to complete each stage of the process. *Dataset* represents the dataset set we used. $C1$ represents the average percentage of the data SOMA correctly classified after the initial training stage (i.e. the percentage of the data correctly classified when we used only the central nodes in the classification process). $T1$ represents the average time (in minutes) the initial training stage took to complete. $C2$ and $C3$ represent the average percentage of the data SOMA correctly classified after extension 1, i.e. using all the nodes (central and non-central nodes) and after extension 2 (i.e. using the central and non-central nodes once the non-central nodes were moved further towards the outer edges of the clusters) respectively. $T2$ and $T3$ represents the average time extension 1 and extension 2 took to complete respectively, and, TT represents the total time, on average, the whole process, i.e. all extensions, took to complete. The results show the percentage of the datasets SOMA correctly classified after the initial traning stage was the same regardless of the topology used (see table 5.4 for the 1-D results). In addition, the results show that the initial training stage took longer to complete when we used a 2-D topology compared to a 1-D topology. For example, when we used dataset 10 containing 40000 20-D datapoints, SOMA took 6.5 minutes and 5 minutes to finish the initial training stage when used with a 2-D and 1-D topology respectively (see table 5.4 for 1-D topology results). After extension 1 the 2-D topology classified a higher percentage of each dataset correctly compared to a 1-D topology (see table 5.4). For example, when we used dataset 1 with SOMA, it classified 95% and 93% of the dataset correctly when we used a 2-D and 1-D topology respectively. However, after extension 2, both network topologies classified a similar percentage of the dataset correctly.

we trained them with dataset 1. We can see the nodes in both the 1-D (Panel (b)) and the 2-D topology (Panel (d)) are more evenly spread out compared to when we used the initial training stage (Panels (a) and (c)). We note that when we used SOMA with a 2-D topology on datasets 1,3,6, it correctly classified a slightly higher percentage of datapoints after extension 2.

Finally, the results in tables 5.6 and 5.4 show that the initial training stage took longer to complete when we used a 2-D topology compared to a 1-D topology with datasets 1,2,3,6,9,10, and therefore, SOMA took longer to complete the whole process. For example, when we used dataset 10, i.e. 40000 20-D datapoints, the initial training stage took 7.5 and 5 minutes to finish when we used a 2-D (3x3 grid) and 1-D (9 nodes) topology respectively. That is, the initial training stage took $\sim$ 2.5 minutes longer to complete when we used a 2-D topology. This is because SOMA updates more nodes using a 2-D topology as many of the nodes have more neighbours compared to a 1-D topology. We next investigated whether increasing the number of nodes, i.e. using a 4x4 grid, affected the speed of SOMA. We applied this network to datasets 9 (i.e. 40000 5-D datapoints) and 10 (i.e. 40000 20-D daapoints). When we used dataset 9 with SOMA 20 times, the initial training stage took, on average, 6.4 minutes. In addition, the initial training stage took, on average, 9.9 minutes when we used the network with dataset 10. In other words, when we used SOMA with a 2-D network topology containing 16 nodes the initial training stage took an extra 1.1 and 2.4 minutes compared to using a 2-D topology with 9 nodes and datasets 9 and 10 respectively. In conclusion, the results in tables 5.4 and 5.6 suggest that regardless of datapoint vector size, cluster size and position, the number of datapoints misclassified by a network was similar regardless of topology after extension 2. However, in some cases, using a 2-D topology may slightly increase performance but, as a consequence, may increase computational costs. For the results in the next sections we use a 1-D topology for simplicity unless otherwise stated. However, if time or computing power is not a concern, 2D topologies could be used.

5.6 Application To Experimental Data

We now apply our method to experimental data and compare it to other methods. We use both data where we know, and don't know, the number of neurons and the spikes each fired.

5.6.1 Datasets

In this section we apply SOMA to data from both the rat olfactory bulb and the inferotemporal cortex in sheep. The datasets were provided by the Babraham Institute, Cambridge. In chapter 3 we discussed how the rat olfactory bulb datasets were acquired. For the sheep datasets, spiking activity was recorded from unanaesthetised sheep while they performed an operant discrimination task in which different pairs of sheep faces were presented and a correct panel-press response elicited a food reward. For both preparations, individual electrodes were fabricated from tungsten wires (125μ diam.) sharpened to a $< 1\mu$ tip and insulated with epoxylite. Electrode impedances were $\sim 200k\Omega$. Neuronal activity (spikes) was sampled extracellularly from each electrode.

5.6.2 Application To Recordings From The Rat Olfactory Bulb

The spike data used in this section was recorded from the mitral cell layer of the OB when no odour stimulus was presented. SOMA was used with the default values discussed in the methods section. The datasets discussed here were produced either by a Multi-Electrode Array (MEA) (which we refer to as multi-neuron datasets) or single electrodes (single neuron datasets). A multi-neuron dataset contains the spiking activity of several, but an unknown number of, neurons. A single neuron dataset contains the spiking activity of one neuron.

Using SOMA With A Dataset Containing A Known Number Of Cells

In this section we used, and merged, two single neurons datasets, where each dataset contained approximately 7000 spikes, into one. The spikes were recorded from a sampling period of 100 seconds. Our objective here was to determine whether

the results acquired by SOMA, using the merged dataset, corresponded to the information known. The merged dataset contained 13256 spikes.

SOMA used 9 outputs (nodes), and, extracted and used 6 *Pcuv* components from this dataset. These were calculated as discussed in the method section. We show in fig. 5.15 (Panels (a) and (b)), the sets of waveforms SOMA attributed to each group, and Panel (c) shows the average waveform shape from each, i.e the underlying signal recorded from the neurons.

We compared the classification results SOMA produced with the two single cell datasets, and found that SOMA had classified 100% of the spikes correctly, i.e. the process identified the correct number of clusters (neurons) and grouped together the correct spikes.

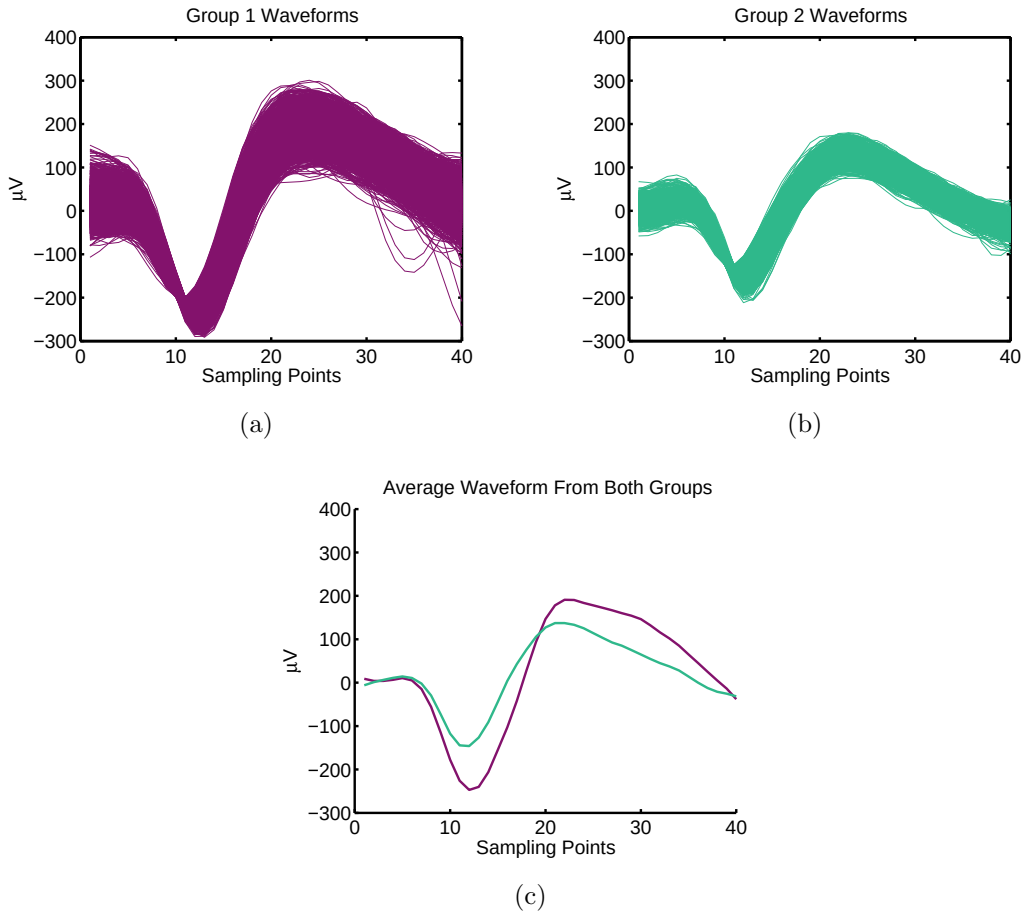


Figure 5.15: **Panels (a) and (b)** show the spike waveforms attributed to the cluster groups 1 and 2 respectively. **Panel (c)** shows the average waveform from both groups

Using SOMA With A Dataset Containing An Unknown Number Of Cells

In this section, we used data recorded from a multi-electrode array. More specif-

ically, we used the data acquired from one electrode, which contained the spiking activity of several neurons. The dataset contained 65256 spikes recorded from a sampling period of 160s.

We know that the firing rate of mitral cells is typically in the range of 15-50Hz. Thus, our objective here was to determine whether SOMA could form and identify several clusters (neurons) from the dataset, where each cluster of spikes represents a firing rate in the range of 15-50Hz.

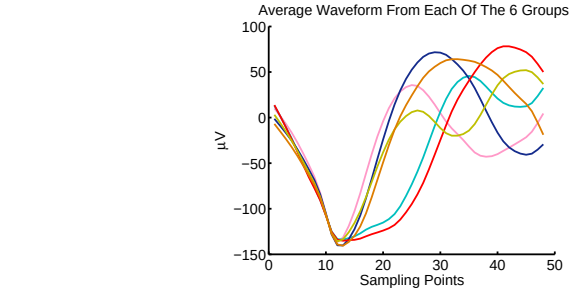
SOMA used 28 outputs (nodes), and, extracted and used 8 *Pcuv* features. These were calculated using the process described in the methods section. We show the groups of spikes found by SOMA in fig. 5.16, panels (b)-(g).

The groups can be associated to different neurons, as each group corresponds to a distinctive shape (fig. 5.16, panel (a)), and, the number of spikes in each group represents a firing rate in the known range. Thus, we can be fairly certain that each group represents the spikes fired by one neuron. However, to increase the validity of the results, we should, when possible, use other evaluatory measures. For example, we could test whether the interspike intervals associated to each cluster of spikes fits a known distribution and/or that the minimum interspike interval associated to each cluster correspond to a known refractory period of the neurons recorded. We note that in our case we do not have any supplementary information about our datasets, and therefore, we cannot increase the validity of our results.

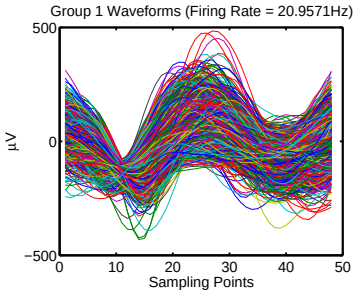
5.6.3 Application to Recordings from Sheep Temporal Cortex

The spike data used in this section was recorded from the sheep temporal cortex when no stimulus was presented. The dataset we used contained 46456 spikes recorded from a sampling period of 100s. We know that the approximate firing rate of these neurons is in the range 15-45 spikes/sec (Nicol et al., 2003). Thus, our objective was to determine whether SOMA could identify a number of clusters, where the number of spikes in each represent a firing rate in the above range.

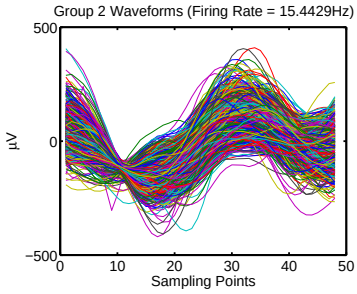
SOMA used 31 outputs, and, extracted and used 10 *Pcuv* features (2 more than the number extracted with the rat recordings). We found that SOMA was similarly successful when applied to this dataset. We show the spike waveforms SOMA



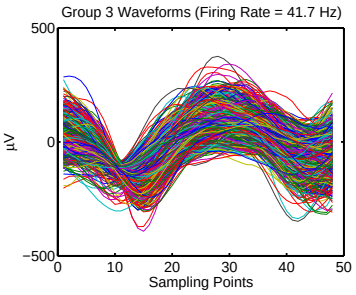
(a)



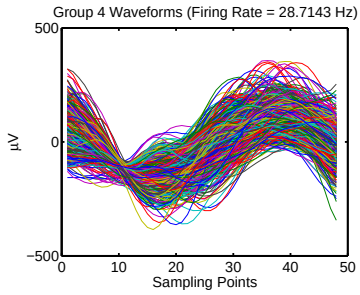
(b)



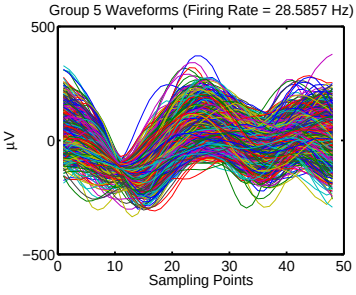
(c)



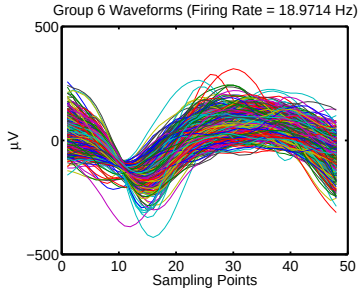
(d)



(e)



(f)



(g)

Figure 5.16: **Panels (b)-(g):** show the spike waveforms SOMA grouped together. **Panel (a):** shows the average spike waveform from each group.

grouped together in fig. 5.17 (Panels (b)-(g)).

SOMA extracted more components from these spike shapes as they were more complex, thus, more difficult than the rat recordings to distinguish. In fig. 5.17 (Panel (a)), we show the average spike waveform from each group.

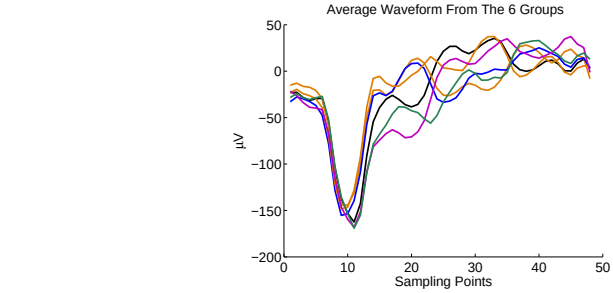
We have shown, from the rat and sheep result sections, that SOMA can identify several groups of spike shapes (neurons) within a dataset regardless of waveform shape complexity by adapting its process according to the data presented, i.e. increasing the number of components to describe more complex waveform shapes. The process, therefore, is not restricted to work with a particular dataset and can be used with recordings from any animal or brain area, not just the ones presented here.

5.7 Comparing SOMA With Other Spike Sorting Methods

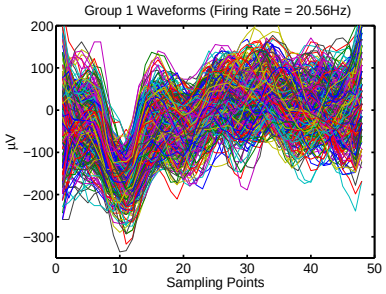
In this section we compare the feature extraction method of SOMA, *Pstage*, with Wavelets (Quiroga et al. (2004) and discussed in Chapter 2). We also compare the SOMA clustering method with Klustakwik ((Harris, 2003), which uses the CEM algorithm (Celeux et al., 1992)) and Waveclus (Quiroga et al. (2004) and discussed in Chapter 2), which uses the super-paramagnetic clustering (SPC) algorithm. To compare these methods we used several single neuron datasets recorded from the rat olfactory bulb. More specifically, we merged combinations of these datasets (how we chose them is discussed below) into different datasets and used them with the spike sorting methods. Thus, as we know the number of neurons recorded and the spikes that belong to each neuron in a dataset, we evaluate a method's performance by:

1. **the number of neurons detected by the clustering algorithms.**
2. **the number of spikes correctly associated to each neuron.**

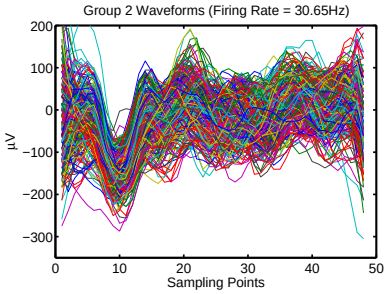
Therefore, using the above as performance measures we also evaluate the:



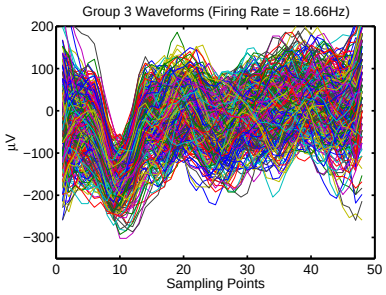
(a)



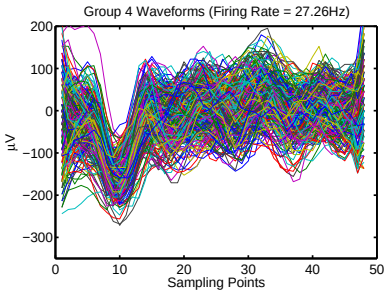
(b)



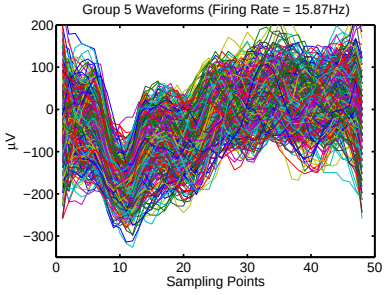
(c)



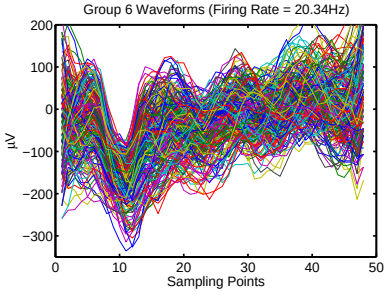
(d)



(e)



(f)



(g)

Figure 5.17: **Panel (b)-(g):** show the spike waveforms SOMA grouped together. **Panel (a):** shows the average spike waveform from each group.

1. **method's performance when we use neuron spike recordings that vary in shape and noise.**

2. **method's performance when we use them in an automated way and how we would use them in practice, i.e. with the default values.**

5.7.1 The Datasets Used

Defintions

To explain the datasets used we firstly define some terms. We define the i^{th} waveform w, from a set W as a vector of T voltage samples, i.e. $w^i = [v_1^i, ..., v_T^i]$, where v_j^i refers to the j^{th} voltage sample. Therefore, we define set W, which contains k waveforms as:

$$W = \begin{bmatrix} v_1^1, ..., v_T^1 \\ . \\ . \\ . \\ v_1^k, ..., v_T^k \end{bmatrix} \tag{5.12}$$

where the mean of the set W, i.e. $\bar{W}$, is defined by $\bar{W} = [\bar{v_1}, ..., \bar{v_T}]$, and the standard deviation of the set W, i.e. σ_W, is defined by $\sigma_W = [\sigma v_1, ..., \sigma v_T]$. $\bar{v_j}$ refers to the mean, and σv_j refers to the standard deviation, of the voltage trace in set W at the j^{th} sampling point. In other words, $\bar{v_j}$ and σv_j is the mean and the standard deviation of column j in set W respectively.

Signal to Noise Ratio

We define the signal to noise ratio as the root-mean square (rms) of the mean spike waveform, i.e. $\bar{W}$, divided by the noise, i.e. σ_w (Rutishauser., et al 2006), so:

$$SNR = \frac{||\bar{W}||}{2\sigma_w} \tag{5.13}$$

where:

$$\sigma_w = \sqrt{\frac{\sum_{i=1}^{k} \sum_{j=1}^{T} (w_j^i - \bar{W}_j)^2}{n}} \tag{5.14}$$

$n = T \cdot k$, w_j^i is the voltage trace at the j^{th} sampling point of the i^{th} spike waveform in set W, and $\bar{W}_j$ is the voltage trace at the j^{th} sampling point of the mean spike waveform $\bar{W}$.

The Datasets

In our comparisons, we used single neuron spike waveform datasets recorded from a rat's olfactory bulb. The spikes were recorded from a 200 sec sampling period. We refer to a single neuron dataset as W in this section. We constructed 9 datasets for our comparisons by implementing the following:

1. We compared the mean spike waveform shapes of 11 pre-recorded W sets, and grouped together the W sets that had dissimlar (Group 1), similar (Group 2) and very similar (Group 3) mean waveform shapes. We show the datasets contained in each group, i.e. the mean spike waveform shapes, in fig. 5.18. Group 1 and 2 contained 3 neuron datasets shown in Panels (a) and (b) respectively, and Group 3 contained 2 neurons, shown in Panel (c) .

2. From each group, which we refer to as G (e.g. $G = 1$, refers to group 1) we extracted three datasets, $G dataset1$, $G dataset2$ and $G dataset3$, where each set contained neuron recordings at different signal to noise (SNR) ratios[11]. $G dataset1$ contained the spikes from all of the group G's W sets. $G dataset2$ and $G dataset3$ contained a subset of waveforms from each of the W sets that fluctuated, at all sampling points, between a low noise range and a higher noise range respectively. That is, a waveform w^i from set (W) and group G would be placed in $G dataset2$ if the electrophysiological trace at every sampling point j was $(w_j^i < \sigma_{W_j}) \wedge (w_j^i > -\sigma_{W_j})$ (a low noise range) and/or $G dataset3$ if $(w_j^i < 2\sigma_{W_j}) \wedge (w_j^i > -2\sigma_{W_j})$(a higher noise range). In fig. 5.19 we show an example W set from group 2 (Panel (a)), the waveform subsets extracted from W within the noise range $\pm\sigma_W$ (Panel (c)) and $\pm 2\sigma_W$ (Panel (d)), and the signal to noise ratio of each set.

   Panel (b) shows the $\bar{W}$, $\pm\sigma_W$ and $\pm 2\sigma_W$ intervals of W. In the results section, we refer to a $G dataset$ from group G as the average SNR, where the average is

[11] The G part of $G dataset$ refers to the group number G.

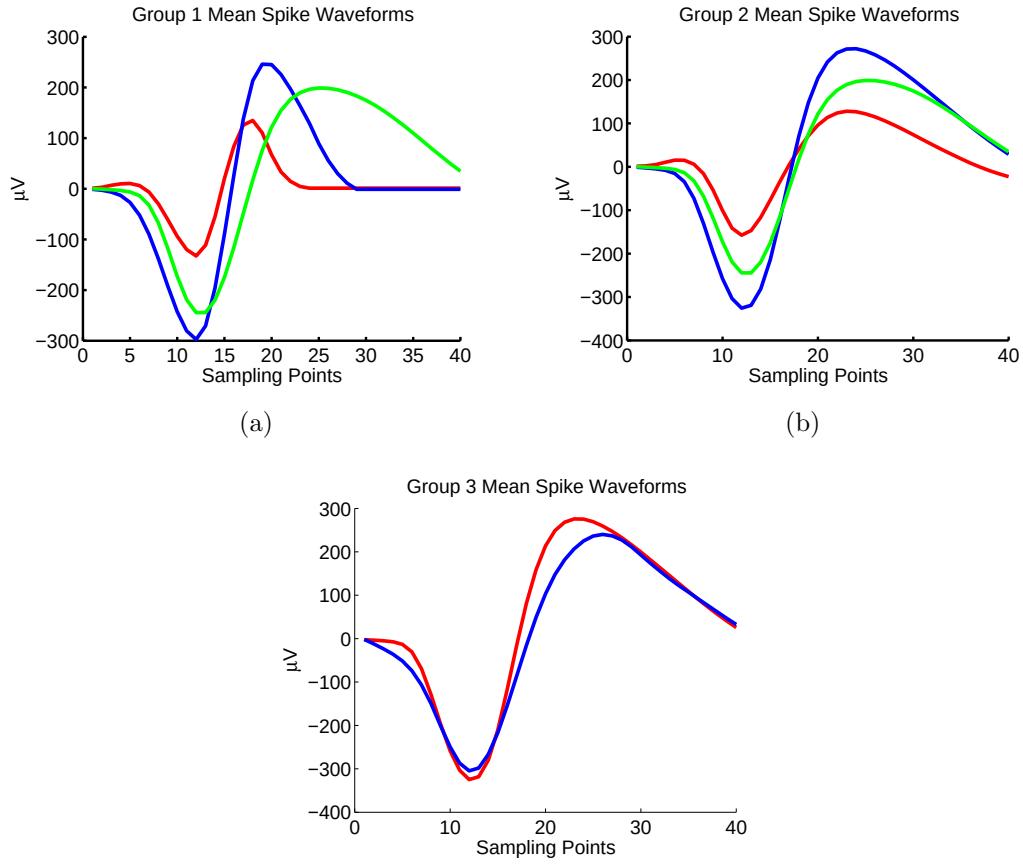


Figure 5.18: These show the mean waveform shapes chosen for each group. **Panels (a) and (b)** show the mean waveform shape of each neuron in Group 1 (dissimilar spike shapes) and Group 2 (similar spike shapes), where both contained 3 neurons. **Panel (c)** shows the mean waveform shape of each neuron in Group 3 (very similar spike shapes), which contained 2 neurons.

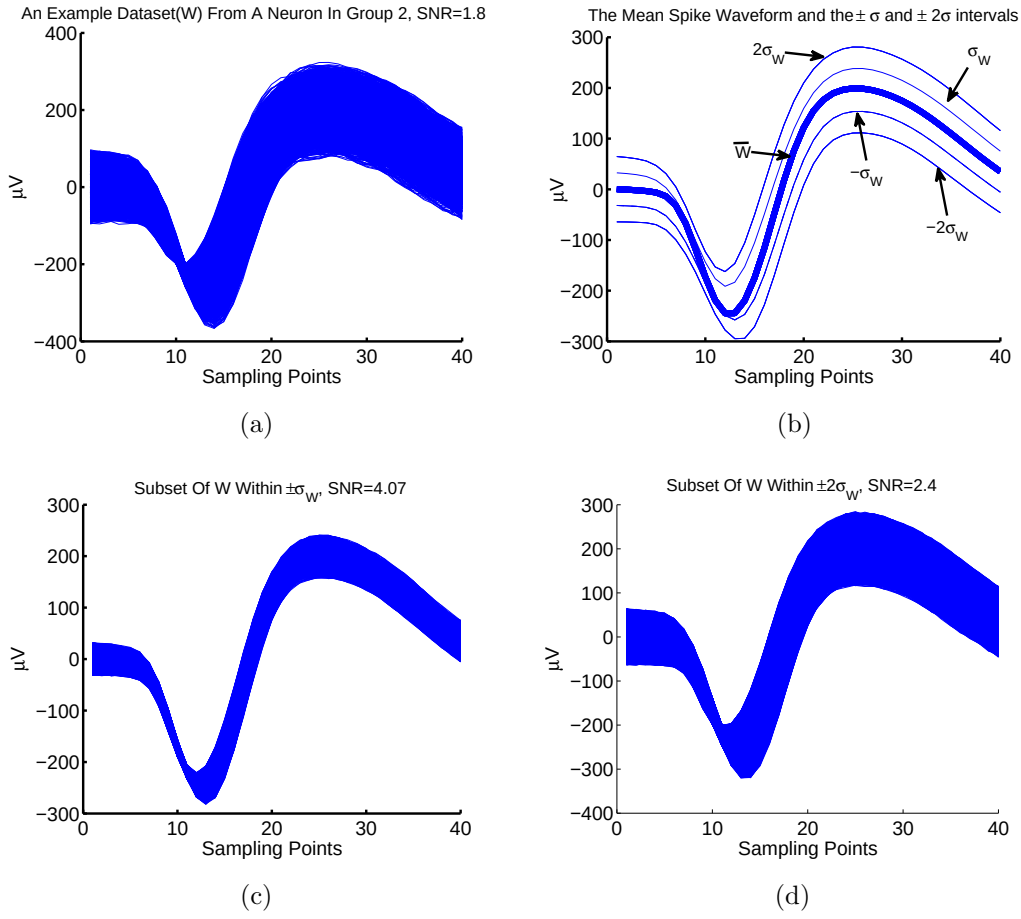


Figure 5.19: This figure shows an example spike waveform set (W) recorded from a neuron (**Panel (a)**) and the waveform subsets extracted from W within the noise range of $\pm\sigma_W$ (**Panel (c)**) and $2 \pm \sigma_W$ (**Panel (d)**). **Panel (b)** shows the $\bar{W}$ (thick middle line), $\pm\sigma_W$ and $\pm 2\sigma_W$ intervals of W

calculated from the SNR's of each neuron set contained in the *Gdataset*. We note, generally, the SNR's in every *Gdataset* were similar. For example, the SNR of each neuron dataset in group 2 *Gdataset*1, was 1.78, 1.93, 1.86.

5.7.2 Evaluating The Performance Of Each Spike Sorting Method

As it can be difficult to compare methods using just the number of misclassified or unclassified spikes they produce, we use a figure of merit for spike sorting derived by (Smith et al., 2004) for our comparisons.

$$M = \frac{1}{n}\sum_{j=1}^{n}(max_i(T_j(i) - \sum_{k=1k\neq j}^{n} T_k(i))/N_j) \tag{5.15}$$

where N_j is the number of spikes from neuron j, n is the number of neurons, and $T_j(i)$ is the number of spikes from neuron j found in cluster i. M measures how well the detected clusters follow the different neurons and reaches the maximum value of 1 when the spikes from each neuron are present in exactly one cluster.

In the results section, we report the figure of merit produced by each method, the number of clusters(neurons) they detect, and the number of unclassified and misclassified spikes they produce. The number of misclassified spikes refers to the spikes from one neuron that are incorrectly associated to another. The number of unclassified spikes refers to the spikes not classified, and/or, the spikes contained in clusters that do not represent a neuron, i.e. the clusters not selected in the figure of merit.

5.7.3 Using The Methods In An Automated Way

For our comparisons we used KlustaWin Version 3.1 (Heitler, 2002), which is Klustakwik with a graphical user interface, and the spike sorting facility of *waveclus* (Quiroga et al. 2004), which uses the super-paramagnetic clustering technique (SPC). In this section we refer to $Waveclus$ and $Klustawin$ as SPC and Klustakwik respectively. We implemented SOMA using Matlab.

We used SOMA with the default values to generate the $Pcuv$ features. To generate wavelets we used waveclus's function, i.e. we used the wavelet pre-processing

method described in (Quian Quiroga et al., 2004). This method performs a four-level decomposition of the spike waveforms using Haar wavelets and then automatically selects 10 wavelet coefficients using the Kolmogorov-Smirnov test for Normality. In other words, the method selects 10 wavelet coefficients that best separate the clusters, i.e. have the least Gaussian distribution. However, in our tests, we used the same number of wavelets, i.e. the best wavelets according to the Kolmogorov-Smirnov test, as the number of *Pcuv* features SOMA extracts in all clustering methods for fair comparisons. We show the number of components used in our tests in the results section. We note that using more wavelet or *Pcuv* components did not increase the clustering methods performances.

For our comparisons, we used each method in a way that reflected our interest in automated spike sorting and how we would use these methods in practice. That is, we compared the results obtained from running each method once with the default values. We ran the methods in this way as, in practice, we would:

1. use the default values with each method as we would not know the optimal values to use, and, we cannot tune (optimise) the parameter values (i.e. run the method multiple times with different values until a particular result is achieved) as we would not know the optimal result (i.e. we would not know the correct number of clusters the method should group the spikes into, and, which spikes should be grouped together). However, we note that we did investigate whether modifying the main parameter values in SPC, Klustakwik and wavelets increased their performance and discuss this in the next section. We also note we did not use the force classification function of waveclus, as we were testing the SPC algorithm and the force classification procedure was not part of it. In addition, we allowed Klustakwik to form any number of clusters, i.e. we did not restrict the number of clusters Klustakwik could form.

2. run the method once, as, running the methods multiple times may produce different results, but, we would not know which run produced the best results. However, we do note that, in general, there was little difference between different runs.

5.7.4 Results

SOMA's pre-processing method *Pstage* extracted 6 *Pcuv* (3 PCA and 3 curvature components), i.e. the minimum number of features, from all 9 datasets. Thus, we used the same number of wavelet coefficients in each clustering method for fair comparisons. We note that on different runs of SOMA, SOMA identified/extracted the same number of components.

Table 5.7 shows the results produced by SPC, Klustakwik, SOMA, wavelets and *Pcuv* when used with all 9 datasets. In general, all pre-processing and clustering methods produced a similar figure of merit when the spike shapes from neurons were different, regardless of the SNR (Group 1 rows 1-6). However, when the spike shapes became more similar (Group Datasets 2 and 3 (rows 7-11 and 12-18) respectively), and the SNR became poorer, the figure of merit produced by SPC, SOMA and Klustakwik decreased. That is, the number of spikes SOMA misclassified, and the number of spikes both SPC and Klustakwik unclassified, increased. However, the figure of merit produced by SOMA, Klustakwik and SPC tended to be higher when they were used with the *Pcuv* features (examples shown in rows 11-12 and 17-18). That is, when *Pcuv* was used, Klustakwik and SPC tended to result in less unclassified and more miclassified spikes. Furthermore, SOMA resulted in less misclassifications. When the spikes shapes were very similar and the SNR was poor (results shown in table 5.7 (rows 17-18)), SOMA produced a higher figure of merit than SPC and Klustakwik, especially when used with *Pcuv* (0.81). Furthermore, SPC produced a lower figure of merit than SOMA and Klustakwik, especially when used with wavelets (0.07).

The results also show that SOMA and SPC always detected the correct number of clusters (neurons) in the datasets regardless of the spike shapes or SNR. Klustakwik generally detected too many clusters, which tended to be a greater number when the SNR was poor (examples shown in rows (11-12 and 17-18)).

Overall, SOMA with *Pcuv* tended to produce the best, and SPC with wavelets tended to produce the worst, figure of merit. This is clearly shown in table 5.8 (rows 2 and 5), which shows the average results from table 5.7.

We note that SOMA classifies all spikes, however, SOMA always performs better than if the spikes were classified correctly by chance. That is, SOMA always classi-

G	SNR	SN	P	Clustering Method											
				SOMA				Klust				SPC			
				FoM	Mis	Un	Cl	FoM	Mis	Un	Cl	FoM	Mis	Un	Cl
1	4.5	13867	C	**1**	0	0	3	**1**	0	0	3	0.98	0	195	3
1	4.5	13867	W	0.99	5	0	3	0.97	0	367	5	0.9	0	1502	3
1	2.6	21312	C	**0.99**	76	0	3	0.98	0	221	3	0.98	0	444	3
1	2.6	21312	W	0.96	523	0	3	0.96	0	809	7	0.89	0	2591	3
1	2.0	30785	C	**0.98**	464	0	3	0.95	89	1289	4	0.91	17	1410	3
1	2.0	30785	W	0.97	735	0	3	0.9	10	2797	5	0.83	21	5357	3
2	4.1	14956	C	**1**	0	0	3	**1**	0	0	3	0.91	18	1303	3
2	4.1	14956	W	0.98	96	0	3	0.97	108	87	3	0.81	6	2625	3
2	2.5	21616	C	**0.96**	467	0	3	0.94	503	0	3	0.68	31	7281	3
2	2.5	21616	W	0.9	978	0	3	0.83	0	3906	5	0.64	37	8409	3
2	1.9	30651	C	**0.84**	2463	0	3	0.79	388	4476	9	0.56	118	13711	3
2	1.9	30651	W	0.8	3153	0	3	0.77	8	7490	9	0.5	110	15687	3
3	5.1	10196	C	**0.89**	530	0	2	0.8	195	1600	4	0.7	34	2933	2
3	5.1	10196	W	0.81	948	0	2	0.77	53	2218	3	0.46	17	5401	2
3	3.5	16648	C	**0.82**	1563	0	2	0.74	550	3180	6	0.46	87	8754	2
3	3.5	16648	W	0.74	2036	0	2	0.67	71	5152	4	0.27	42	11914	2
3	2.7	22762	C	**0.81**	2435	0	2	0.61	800	6530	9	0.32	168	14945	2
3	2.7	22762	W	0.62	4400	0	2	0.6	115	8408	8	0.07	49	20870	2

Table 5.7: This table shows the results from the spike sorting methods. G and SNR represent the group dataset and the average signal to noise ratio respectively. P represents the feature set used, where C and W represent $Pcuv$ and wavelets respectively. $SOMA$ represents our clustering method. $Klust$ and SPC represent the Klustakwik and the super-paramagnetic clustering method respectively. SN refers to the number of spikes, where Mis and Un represent the number of spikes misclassified and unclassified respectively. FoM represents the figure of merit, and Cl represents the number of clusters detected. The best FoM score acquired from each dataset is in bold. In general, all pre-processing and clustering methods produced a similar figure of merit when the spike shapes from neurons were different, regardless of the SNR (rows 1-6). However, when the spikes shapes were very similar and the SNR was poor (results shown in table 5.7 (rows 17-18)), SOMA produced a higher figure of merit than SPC and Klustakwik, especially when used with $Pcuv$ (0.81).

Clustering Method	P	SN	FoM	Mis	Un
SOMA	W	20310.3	0.86	1408.2	0
	C	20310.3	**0.92**	910.8	0
Klust	W	20310.3	0.82	40.5	3470.4
	C	20310.3	0.86	280.5	1921.7
SPC	W	20310.3	0.59	31.3	8261.7
	C	20310.3	0.722	52.5	5664

Table 5.8: This table represents the average values from table 5.7. P represents the feature set used, where C and W represent $Pcuv$ and wavelets respectively. $SOMA$ represents the new clustering method. $Klust$ and SPC represents the Klustakwik and the super-paramagnetic clustering method respectively. SN refers to the number of spikes, where Mis and Un represent the number of spikes misclassified and unclassified respectively. FoM represents the figure of merit. The best average FoM score is in bold. The table shows that, overall, SOMA with $Pcuv$ performed the best (row 2), i.e. they produced the highest figure of merit, and SPC with wavelets performed the worst (row 5), i.e. they produced the lowest figure of merit

fied more than 50% of the datasets correctly when they contained 2 neurons (table 5.7 (group 3)(rows 13-18)) and more than 33% when they contained 3 neurons (table 5.7 (group 1 and 2)(rows 1-12)). Finally, we note that each method took a similar time to complete the spike sorting process regardless of the dataset used, i.e. each method took on average ∼10 minutes.

As discussed in the previous section, we did try different parameter values in SPC (Waveclus), Klustakwik (Klustawin) and the wavelet pre-processing procdure (Waveclus) to determine whether different values increased their performance. For example, firstly, we tried different mother wavelets such as Debauchies and Gaussian in the wavelet pre-processing procedure. Secondly, we changed the parameter value (i.e. the temperatue parameter) that deterimines which spikes are grouped together in SPC (Waveclus). Finally, we changed the parameter value that specifies the maximum number of clusters Klustakwik is allowed to form. We found that changing the above parameters in SPC and wavelets had a negligable effect on their performance, i.e. we could only slightly increase their FoM score by ~ 0.05. However, the maximum cluster value in Klustakwik had a greater effect. We found that specifying the correct number of clusters (neurons) in Klustakwik significantly increased its performance when used with the Group 3 datasets. For example, the

FoM value Klustakwik produced when used with dataset Group 3, SNR=2.1 and the $Pcuv$ feature set increased from 0.61 (table 5.7 row 12) to 0.83. In other words, providing Klustakwik with the correct number of clusters prevented it from forming large numbers of clusters, i.e. prevented it splitting, what should be one cluster, into several. Thus, in this case Klustakwik outperformed our clustering method. However, in practice, we would not know the correct number of clusters to use with Klustakwik, and therefore, in cases when we have no information about the datasets, SOMA is a better method.

5.8 Noise Extraction

Electrical noise is almost invariably incorporated into electrophysiological data. This often originates from electrical equipment, e.g. from electrical actuators for delivering stimuli. Such signals may exceed the spike-triggering threshold and may resemble spikes in their form and amplitude. They are often difficult to exclude from electrophysiological datasets. Here we introduce a process for extracting waveforms that are caused by noise, i.e. artifacts, from the spike waveform datasets. We note this stage is optional and will only work if the noise artifacts are known. Example artifacts found in rat olfactory bulb recordings are shown in fig. 5.20.

To extract these noisy artifacts from the datasets we:

1. Use $Pstage$, i.e. SOMA's pre-processing stage, with the spike dataset and the artifacts. That is, all waveforms, including the artifacts, are transformed into a set of $Pcuv$ feature components. We refer to the transformed noisy artifacts as noise weight vectors (W_o). We use these in the classification stage.

2. Train a Kohonen network with the feature sets using the initial, extension 1 and extension 2 stages discussed in the clustering method section. At this point we do not use the noise weight vectors (W_o).

3. Use the classification stage with the trained weight vectors, i.e. W_n, where n is a network output, and the noisy weight vectors W_o. Thus, a spike waveform feature set x is extracted and classed as noise if it is closer to a noise weight vector (W_o) than a network output (W_n).

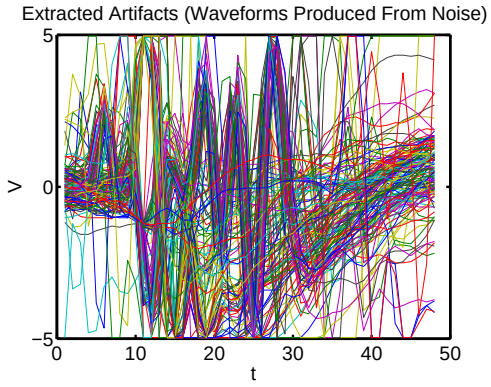


Figure 5.20: An example set of artifacts extracted from the rat olfactory bulb dataset.

5.9 Discussion

In this chapter we have presented a method for spike sorting, i.e. a method that identifies, after the spike detection process, the number of cells, and their spiking activity recorded by multi-electrode arrays. Our method, called SOMA, is a fully automatic process (i.e. it is a method that automatically extracts/determines a number of features from the spike shapes that form a number of separable clusters in the feature space, and, automatically determines the number of, and groups the spikes into, clusters). Producing an automatic method is a vital consideration for spike sorting as:

1. manually spike sorting multi-electrode data can be very time consuming/impractical due to the hundreds of thousands of mixed spike waveforms they can record.

2. manual methods typically perform worse than automatic methods due to the errors users introduce (Wood et al., 2004).

Our method also:

1. avoids the limitations of other automatic spike sorting methods. That is, our method:

   (a) is not based on assumptions about the distribution and variance of the clusters in the feature space, which is important as the contents of datasets can vary, and therefore, makes our method more reliable and consistent.

   (b) does not require the user to specify optimal parameter values (such as providing the method with the number of clusters in the feature space,

criteria associated to the cluster boundaries and/or the number of features to extract from the spikes shapes) to achieve a high performance. This is an important consideration as, typically:

i. we would not know the optimal value(s) to use.

ii. we cannot tune (optimise) the parameter values (i.e. run the method multiple times with different values until a particular result is achieved) as we would not know the optimal result (i.e. we would not know the correct number of clusters the method should group the spikes into, and, which spikes should be grouped together).

2. reduces computational costs where possible. This is an important consideration as processing the large amount of data MEAs can record can be computationally expensive.

To measure SOMA's performance we used it with several simulated and electrophsyiological datasets and compared it to other standard spike sorting methods. To compare them we used datasets containing a known number of neurons and the spikes each fired. In addition, we compared them in a way that reflected our interests in automatic spike sorting, and, how we would use them in practice, i.e. with the default values.

Pstage Feature Extraction And *Pcuv* Feature Components

An advantage of using *Pstage* is that the feature set extracted (*Pcuv*) can discriminate between very similar spike shapes when SNR is low (Dataset:Group 3, SNR=2.1 table 5.7). Furthermore, the *Pcuv* set can discriminate between the spike shapes better than wavelets. For example, SOMA with *Pcuv* and wavelets produced a figure of merit of 0.81 (table 5.7 row 17) and 0.62 (table 5.7 row 18) respectively. In other words, PCA and curvature components (*Pcuv*) formed better separated clusters in the feature space. We note that we tried improving the wavelets performance by using more wavelets and different mother wavelets in (*waveclus*) such as Daubechies and Gaussian wavelets, but, the results acquired were very similar to the ones shown, i.e. we could only improve the wavelets FoM score by ~ 0.04. In addition, it may be possible to improve the wavelets performance by using a different

wavelet selection other than the Kolmogorov-Smirnov test, but, as far as we know, there are no other selection methods, and therefore, this is difficult to determine. Nevertheless, selecting optimal parameter choices, in practice, is difficult, and, is beyond the scope of this chapter, as we were interested in the automatic classifiction of spikes and using them how we would in practice, i.e. with the default values.

Another advantage of using *Pstage* is that a small feature set is needed to form separable clusters within the feature space. The obvious benefit of clustering spike shape features rather than the whole spike itself is the reduction in computational costs. This is very important when we consider the high volume of data e.g. (> 100000) spikes, which MEA's can record. For example, the memory needed to store (> 100000) spikes would be much greater if we used the whole spike shape. Furthermore, using small feature sets in clustering algorithms can reduce computational time. For example, each clustering method took, on average, 25-30 minutes, i.e. 3 times longer, to cluster the dataset (group 3 SNR 2.7) using the whole spike shape compared to using the *Pcuv* or wavelets feature sets (results not shown). We also note that a feature set can form better separated clusters than the whole spike shape. This is because a feature set can contain important information about the spike shape and exclude the parts dominated by noise. For example, when we used the whole spike shape and *Pcuv* features from the dataset (group 3 SNR=2.7) (very similar spike shapes) with SOMA, the figure of merit produced was 0.41 and 0.81 (see table 5.7 row 17) respectively (results not shown). We note that using the whole spike with the other clustering methods also reduced their performance.

SOMA Clustering Algorithm

The advantages of using the SOMA clustering process is that, firstly, SOMA can automatically identify the correct number of clusters (Kohonen Extension 1). Secondly, SOMA reduces misclassification by automatically analysing the cluster distributions to enhance cluster separation (Kohonen Extension 2). In other words, SOMA does not use any assumptions about the variance or the distribution of the data. Furthermore, SOMA does not require optimal predefined parameters from the user (i.e. it does not need to be tuned to the dataset) to achieve a high performance (i.e. to identify the correct number of clusters and/or classify the spike

waveforms). We showed these advantages when we compared SOMA with SPC and Klustakwik. We found that SOMA tended to perform better (produced a higher figure of merit) than SPC and Klustakwik when the spike shapes from neurons were similar and the SNR was low (e.g. dataset Group 3, SNR=2.7, table 5.7). This is because SOMA could detect the differences between the clusters, i.e. identify the cluster boundaries, more optimally than SPC and Klustakwik. In other words, SPC and Klustakwik produced large numbers of unclassified spikes. We also found that when we provided Klustakwik with the correct number of clusters to group the spikes into, Klustakwik can outperform SOMA. However, in practice, it is not possible to provide Klustakwik with the correct number of clusters to form as we would not know the number contained in the dataset. Therefore, when information is not known about the datasets SOMA is a better method to use. Nevertheless, selecting optimal parameter values for spike sorting is beyond the scope of this chapter, as we were interested in the automatic classifiction of spikes and using them how we would in practice, i.e. with the default values.

SOMA's Clustering Parameters

The performance of SOMA appears to be insensitive to small changes in the values of the parameters, however, large changes can affect the performance of the process. In other words, increasing some parameter values such as the number of outputs used with SOMA can enhance the classification result but, as a consequence, decrease the speed of the process. Therefore, the values of the parameters can be modified to achieve what is most important to the user, i.e. acquiring results quickly, acquiring the best results, or a combination of both.

Using The Kohonen Network Extensions With Other Methods

It is possible to use our extensions with different network topologies (as shown in the results section) and other clustering methods such as K-means (Lloyd et al., 1982) and linear vector quantization (LVQ) (Kohonen 2001). For example, we could initially train a network, which has many outputs, using either K-means or LVQ with a dataset containing many clusters. As LVQ is a supervised method, we would assign subsets of outputs to each cluster before training. Thus, once we have trained

the networks we could implement the following. We could use extension 2 on the LVQ trained network to push and disperse the nodes (outputs) out to encompass each cluster better, and then, use our clustering procedure to reduce the number of misclassified spikes. Furthermore, we could use extension 1 with the k-means trained network to identify the number of clusters in the feature space, and then use extension 2 and the classification procedure to achieve the same as above.

5.10 Summary

In this chapter, we have successfully demonstrated the efficiency of SOMA and *Pstage*, by testing it on real electrophysiological data, where the results were known and unknown, simulated data and comparing it with other standard spike sorting methods.

Chapter 6

Applications Of MANOVA To Multielectrode Array Electrophysiology Data

6.1 Introduction

To investigate/test a population coding hypothesis at the level of multiple cell-cell interactions in the brain, neurophysiologists would typically record, using a multi electrode array (MEA), the spiking activity of neurons in the brain of a test subject, when a stimulus is, and is not, presented. The spiking activity of the individual neurons are then extracted (we refer to the extracted spiking activity of individual neurons as MEA sorted data) using a spike detection and spike sorting algorithm (we introduced a technique in chapter 5) from the MEA recordings. Thus, once a number of neurons and their spiking activity is identified, we can use it to address any of the population coding hypotheses.

To address the population rate coding hypothesis (i.e. address questions related to whether external stimuli are conveyed in the spiking rates of a population of neurons, which is the main focus of this thesis), using the sorted MEA data recorded before and during stimulus presentation, we require methods that provide a detailed statistical analysis of the stimulus evoked changes in the spiking rates of the neurons recorded, and, shows the responsive, and most responsive, subareas of neurons (i.e. the subareas exhibiting the most significant change in spiking rates) within the MEA

recorded area. We refer to such subareas as hot spots.

We require such methods to address questions such as:

1. does the whole recorded area respond to the stimulus, or, does one or many small subareas respond. In other words, is the stimulus represented by small groups of neurons, or, is it represented in a large group of neurons.

2. do neurons respond to a stimulus with the same or different spiking rates.

The standard tests for determining significant differences between the spiking rates of a population of neurons before and during stimulus presentation are:

1. the ANOVA (Analysis Of Variance) test, which would be used on each neuron.

2. the MANOVA (Multi variate analysis of variance) test, which performs one test on the whole group of neurons to determine whether a stimulus caused a significant change, overall, in the spiking activity of the population.

However, these methods are not designed to achieve the above analysis, i.e. identify and show the most responsive subareas of neurons (hot spots), within an MEA recorded area. Thus, we propose a method to identify these subareas using MANOVA called MEANOVA (=MEA+MANOVA). We concentrate on MANOVA and not ANOVA as we are interested in identifying the groups of neurons within the recorded area that are, overall, the most responsive. Our proposed method:

1. consists of using a MANOVA test, where we modified its output. We modified the output of MANOVA so that it produces a significance score to show how responsive, overall, a set of neurons was to a stimulus. The higher the score produced, the more responsive the group of neurons was. The standard MANOVA test is not designed to show this, i.e. it would only show whether a change in the spiking rates of a population of neurons was significant and not how significant.

2. consists of applying the modified MANOVA test to small (local) and large (global) areas of neurons recorded by a multi-electrode array. In other words, we apply the modified version of MANOVA to sets of neurons recorded by individual, combinations, and all of the electrodes on an MEA. Thus, any sets

of neurons showing a higher significance score would correspond to the most responsive subareas of the MEA recorded area. Applying MANOVA in this way also avoids the limitation of its standard application. That is, MANOVA is designed to analyse the changes in the spiking rates of all of the neurons as a whole, and therefore, if only a small proportion of the neurons responded to the stimulus and a large number remained unresponsive (which we would naturally expect), the former might be completely obscured or diluted by the latter in the test, and therefore, the MANOVA test may assert that there were no stimulus-evoked changes. Our method avoids this limitation by analysing small and large subsets of neurons, and therefore, is more likely to show a small set of responsive neurons amongst a very large set of unresponsive ones.

‘ We validate MEANOVA using data recorded from a rat's olfactory bulb and data obtained from a simple computational network model. We used the computational model and biological data to show that MEANOVA works, i.e. it can identify and show the responsive, and the most responsive, subareas within a neuronal area. We also used the computational model to show the limitation of MANOVA (discussed above), i.e. that MANOVA can incorrectly state that a stimulus had no effect on a large population of neurons when one responsive neuron is embedded in a large number of unresponsive neurons. We also show that MEANOVA overcomes this limitation.

Parts of all the sections in this chapter were published as: Horton P M, Bonny L, Nicol A U, Kendrick K M, Feng J F. Applications Of Multi-Variate Analysis Of Variance (MANOVA) To Multi-Electrode Array Electrophysiology Data. J Neurosci Meth 2005;145:22-41. I confirm that I performed all simulations and developed the ideas. The other authors are my supervisor, J Feng, who advised me, L Bonny, who produced some parts of the MANOVA software, and, K Kendrick and A Nicol who provided the electrophysiological data. We note that this method was used in other work (Nicol et al., 2005).

6.2 Analytical Methods

In this section we first describe the two-way MANOVA method and then our method (MEANOVA) of applying MANOVA to MEA data to detect hot spots.

6.2.1 Two-Way MANOVA

To determine whether two factors, such as a stimulus and/or the number of trials (i.e. the repeated presentation of the same stimulus), changes the spiking activity of a population of neurons, a neurophysiologist would typically use a multi-electrode array (MEA) to record brain activity under different experimental conditions, i.e. at various levels of the two factors. The neurophysiologist would then compare the datasets acquired under the different experimental conditions to determine if the factors had an effect on the spiking activity of the neurons. If a neurophysiologist wanted to determine whether the spiking activity (spiking rates) of the population were changed by one or both factors they would typically use a two-way MANOVA test. A MANOVA or multivariate analysis of variance tests the hypothesis that one or more independent variables (IVs), or factors, had an effect on a set of two or more dependent variables, e.g. the spiking activity of two or more neurons. MANOVA achieves this by analysing the variance in the spiking activity of the neurons across the various experimental conditions. That is, MANOVA firstly creates a new dependent variable which is a linear combination of the multiple dependent variables (DVs). This particular combination of DVs is chosen to maximise the difference between the independent variable groups. The MANOVA procedure then assesses whether this new DV differs significantly between the IV groups (Francis 2007). To illustrate the two-way MANOVA process, let us suppose we recorded the spiking activity of P neurons using Q electrodes during time T on two sampling periods (pre- and during-stimulus presentation) on three trials. Thus, we have P dependent variables and two independent variables I and J, where I refers to the stimulus factor (factor 1) and J refers to the trial factor (factor 2). Therefore, factor 1 has two levels: $i = 1$ (pre-stimulus) and $i = 2$ (during-stimulus), and, factor 2 has 3 levels: $j = 1$ (trial one), $j = 2$ (trial two) and $j = 3$ (trial three). Let us also suppose we observe the number of spikes fired by each neuron in M windows during T for each

of the 6 $(I.J)$ combinations of the two factor levels. In other words, for each neuron we have 6 sets of $M = T/t$ time bins for a given bin size t, where each time bin, i.e. the m^{th} window $(m = 1, ..., M)$, contains the number of spikes fired by the neuron[1]. Finally, let X_{ijm} denote the m^{th} observation of the activity of the P neurons (i.e. X_{ijm} is a P dimensional vector, where each element contains the number of spikes fired by each neuron in the m^{th} window) at the i^{th} level of factor 1 and the j^{th} level of factor 2. Thus, for a fixed i, j, m we have:

$$X_{ijm} = \mu + F_i^1 + F_j^2 + F_{ij}^{1,2} + \xi_{ijm} \tag{6.1}$$

where μ is the mean vector across all observations, F^1 is the effect of the first factor, a stimulus in this example, F^2 is the effect of the second factor, the trial, and $F^{1,2}$ is the interaction between the two factors. ξ is a normally distributed random error vector with mean zero, $i = 1, \cdots, I, j = 1, \cdots, J$ and $m = 1, \cdots, M$. Note that X_{ijm} is a P-dimensional random vector. For simplicity of notation, let us further

[1]We note that the MANOVA method assumes that for each of the $(I.J)$ combinations, the spike counts for each neuron across the M intervals are normally distributed. If the spikes of each neuron are independent then according to the renewal theorem (Feller, 1971):

$$\frac{N_t/\sqrt{t} - \sqrt{t}/\mu}{\sqrt{\mu^2/\sigma^3}} \sim N(0, 1)$$

as $t \to \infty$, where μ is the mean interspike interval, σ is the standard deviation of the interspike interval, and N_t is the spike counting inside time window $[0, t]$. Thus, if the spikes of each neuron are independent throughout T, the distribution of the spike counts across the M intervals for each neuron when t is large would approximately follow a normal distribution, and therefore, we can apply MANOVA to the MEA data. Of course these conditions rarely hold for neurons in the brain. However, there are a few ways to overcome this problem. For example, when we calculate N_t for each of the M intervals, we could introduce weights for different interspike intervals, a typical de-trend method.

Of course, in the data from our modelling approach, which we discuss later, since the interspike intervals are generated from an integrate-and-fire model, the interspike intervals are perfect renewal processes. Therefore, the results produced from using MANOVA with the simulation data are valid. For multi-dimensional renewal theorem, we refer the reader to (Hunter 1974a; Hunter 1974b).

For experimental data all statistical approaches including the one we developed here are valid only to a certain degree. For example, a routine statistical test such as t-test, ANOVA or Kolmogorov test would produce errors if used on the spike counts across M intervals from a single cell that were not normally distributed.

introduce that

$$\bar{X}_{i\cdot} = \sum_{j=1}^{J}\sum_{m=1}^{M} X_{ijm}/(JM), \tag{6.2}$$

$$\bar{X}_{\cdot j} = \sum_{i=1}^{I}\sum_{m=1}^{M} X_{ijm}/(IM), \tag{6.3}$$

$$\bar{X}_{ij} = \sum_{m=1}^{M} X_{ijm}/M, \tag{6.4}$$

$$\bar{X} = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{M} X_{ijm}/(IJM) \tag{6.5}$$

where $\bar{X}$ is the average of all the observations, $\bar{X}_{i\cdot}$ is the average of the observation vectors at the ith level of factor 1. Furthermore, $\bar{X}_{\cdot j}$ is the average of the observation vectors at the j^{th} level of factor 2 and $\bar{X}_{ij}$ is the average of the observation vectors at the i^{th} level of factor 1 and the j^{th} level of factor 2. From the equations 6.2 to 6.5, we can decompose the equation above in 6.1 to describe X_{ijm} the observational vector, using the newly defined variables.

$$X_{ijm} = \bar{X} + (\bar{X}_{i\cdot} - \bar{X}) + (\bar{X}_{\cdot j} - \bar{X}) + (\bar{X}_{ij} - \bar{X}_{i\cdot} - \bar{X}_{\cdot j} + \bar{X}) + (X_{ijm} - \bar{X}_{ij}) \tag{6.6}$$

This equation shows how the observed result is achieved by examining the factors that have contributed to get the found result. That is, how much factor 1 $(\bar{X}_{i\cdot} - \bar{X})$, factor 2 $(\bar{X}_{\cdot j} - \bar{X})$, interaction $(\bar{X}_{ij} - \bar{X}_{i\cdot} - \bar{X}_{\cdot j} + \bar{X})$, and the remainder of the variance (residual error) $(X_{ijm} - \bar{X}_{ij})$ have contributed to the total variance, thus, resulting in the observation. The interaction describes the relationship between the various levels of factor 1 and factor 2. For example, if we presented a stimulus on three trials and the spiking activity of a neuronal area on trial 1 and 3 increased but decreased on trial 2, the interaction would show that it is very difficult to conclude what the effects of the factors were.

The residual error describes the variance not accountable by either factor 1 or 2. For example, if we presented a stimulus and the result showed the variance of the residual error was large and the variances of factors 1 and 2 were small, we would conclude that the observation recorded was caused by unknown (random) factors and not by either of the factors under examination.

Thus, according to the classical MANOVA (Johnson et al., 1988) we should firstly, to determine the effects of the IVs on the DVs, calculate the following matrices:

$$S_1 = \sum_{i=1}^{I} JM(\bar{X}_{i\cdot} - \bar{X})(\bar{X}_{i\cdot} - \bar{X})^T \tag{6.7}$$

$$S_2 = \sum_{j=1}^{J} IM(\bar{X}_{\cdot j} - \bar{X})(\bar{X}_{\cdot j} - \bar{X})^T \tag{6.8}$$

$$S_3 = \sum_{i=1}^{I}\sum_{j=1}^{J} M(\bar{X}_{ij} - \bar{X}_{i\cdot} - \bar{X}_{\cdot j} + \bar{X})(\bar{X}_{ij} - \bar{X}_{i\cdot} - \bar{X}_{\cdot j} + \bar{X})^T \tag{6.9}$$

$$S_r = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{M}(X_{ijm} - \bar{X}_{ij})(X_{ijm} - \bar{X}_{ij})^T \tag{6.10}$$

where S_1 is the residual error due to the first factor, S_2 is the residual error due to the second factor, S_3 is the residual error due to the interaction between the first factor and the second factor, and S_r is the total residual. We note in the equations, T refers to the transpose. The next stage is to test the likelihood that any of the factors (Interactions, Factor 1 or Factor 2) caused a change in neuron activity. We achieve this by using Wilks' lambda.

$$\Lambda_3 = \frac{|S_r|}{|S_3 + S_r|} \qquad \Lambda_1 = \frac{|S_r|}{|S_1 + S_r|} \qquad \Lambda_2 = \frac{|S_r|}{|S_2 + S_r|}. \tag{6.11}$$

Intuitively when $\Lambda_i, i = 1, 2, 3$ is small, the factor (factor 1, factor 2 or the interaction) is important and so we can reject the hypothesis that the factor is negligible. This is exactly the case in MANOVA.

For large samples, we can use Wilks' Lambda with a chi-squared percentile, to provide us with confidence that a change in the spiking activity was caused by the factors e.g. a stimulus, and not by chance. Using Barletts multiplier (Bartlett, 1954) to improve the chi-squared approximation, we reject the hypothesis that there were no significant changes in interactions if

$$-\left|IJ(M-1) - \frac{P + 1 - (I-1)(J-1)}{2}\right| \ln \Lambda_3 > \chi^2_{(I-1)(J-1)P}(\alpha) \tag{6.12}$$

where $\chi^2_{(I-1)(J-1)P}(\alpha)$ is the upper (100α)percentile of a chi-squared distribution with $(I-1)(J-1)P$ d.f. (degrees of freedom). We used the significance level (α) of 5% for all interaction tests.

Under the circumstances that Eq. (6.12) is not true, i.e, the interaction is not strong enough, we can test whether factor 1 and factor 2 had any effect on the spiking behaviour. We reject the hypothesis that factor 1 caused no significant change if

$$-\left|IJ(M-1)-\frac{P+1-(I-1)}{2}\right|\ln\Lambda_1 > \chi^2_{(I-1)P}(\alpha), \tag{6.13}$$

where $\chi^2_{(I-1)P}(\alpha)$ is the upper (100α)percentile of a chi-squared distribution with $(I-1)P$ d.f. We used the significance level (α) of 5% for all factor 1 tests.

We reject the hypothesis that factor 2 caused no significant changes if

$$-\left|IJ(M-1)-\frac{P+1-(J-1)}{2}\right|\ln\Lambda_2 > \chi^2_{(J-1)P}(\alpha). \tag{6.14}$$

where $\chi^2_{(J-1)P}(\alpha)$ is the upper (100α)percentile of a chi-square distribution with $(J-1)P$ d.f. We used the significance level (α) of 5% for all factor 2 tests.

6.2.2 MEANOVA=MEA+MANOVA

As we intend to detect hot spots, i.e. the responsive and most responsive groups (subareas) of neurons, within the MEA recorded area, we cannot directly apply the text book version of MANOVA to the MEA data. This is because the standard MANOVA technique is designed to take all of the neurons into account and show whether a factor, such as a stimulus, had a significant effect, overall, on the spiking activity of the population, and, would only show whether a change was significant and not how significant. In other words, MANOVA is not designed to detect hot spots in a recorded neuronal area. Furthermore, if we simply put all of the recorded neurons in a MANOVA analysis, where only a few respond to a stimulus and the rest remained unresponsive, the former may dilute the latter in the test. Therefore, the output of MANOVA may incorrectly show there was no stimulus evoked changes in the spiking activity of the neurons. We show an example of this in section (6.3.3).

With the above considerations in mind, we propose a process that consists of applying MANOVA, where we modified its output so that it produces a significance score to show how significant a change in activity is, to subsets of neurons recorded by individual, combinations (we discuss how we form combinations in the next section), and all of the electrodes.

We define a significance score for a subset $\mathcal{S}$ of $\mathcal{Q}$ as

$$\begin{aligned} S_3(\mathcal{S}) &= -\left| IJ(M-1) - \frac{|\mathcal{S}|+1-(I-1)(J-1)}{2} \right| \ln \Lambda_3 / \chi^2_{(I-1)(J-1)|\mathcal{S}|}(\alpha), \\ S_1(\mathcal{S}) &= -\left| IJ(M-1) - \frac{|\mathcal{S}|+1-(I-1)}{2} \right| \ln \Lambda_1 / \chi^2_{(I-1)|\mathcal{S}|}(\alpha), \\ S_2(\mathcal{S}) &= -\left| IJ(M-1) - \frac{|\mathcal{S}|+1-(J-1)}{2} \right| \ln \Lambda_2 / \chi^2_{(J-1)|\mathcal{S}|}(\alpha), \end{aligned}$$

where $|\mathcal{S}|$ is the number of neurons recorded by electrodes in $\mathcal{S}$. It is routinely required that $IJ(M-1) > |\mathcal{S}|$.

The significance score is made up from dividing the improved Wilks' lambda with the test statistic drawn from the χ^2 distribution. A score of 1 and over will be deemed as significant at the 5% significance level, i.e we have 95% confidence that the change in spiking activity, if any, was caused by one of the observed factors such as a stimulus. Furthermore, the bigger the change in neural activity, the higher the significance score.

Therefore, a subset $\mathcal{S}$ which attains the highest significance score is the 'hottest spot' in the recorded area. In other words, the hottest spots are the subarea(s) of neurons that showed the most significant change in spiking activity.

We note that using all neurons in MEANOVA is the equivalent to using a MANOVA test, but, without the significance score.

6.2.3 Calculating The MEANOVA Scores Using Electrode Subsets

To detect hotspots within a recorded area using MEANOVA, we calculate significance scores for, firstly, groups of neurons recorded by single electrodes on the multi electrode array (MEA). We then calculate the scores for groups of neurons recorded by subsets of electrodes on the (MEA), and finally, we calculate a score for all of the neurons recorded. We, therefore, calculate the scores for subsets of electrodes that range in size from 1 to T, where T is the total number of electrodes. Each subset we use contains a connected set of electrodes. In other words, each electrode we add to a subset is adjacent to, i.e. is an MEA neighbour of, one electrode already in the subset. We discuss this in more detail below. Let $\mathcal{Q}$ represent a two dimensional array (matrix) of electrodes, i.e. $Q = [q_{i,j}]_{i=1\ldots Q;\ j=1\ldots Q}$, where i refers

to the i^{th} row and j refers to the j^{th} column, and Q refers to the number of rows and columns. Furthermore, $q_{i,j}$ refers to the electrode q in the i^{th} row and j^{th} column. Each electrode, therefore, has a maximum of four adjacent neighbours. For example, if $Q = 20$, electrodes $q_{4,2}, q_{2,2}, q_{3,1}, q_{3,3}$ would be adjacent (connected) to electrode $q_{3,2}$. An example subset from Q containing 4 electrodes is $q_{1,1}, q_{2,1}, q_{2,2}, q_{3,2}$. We show more example subsets in Appendix A and B. We constructed the subsets in this way so the MEANOVA scores produced by small and large subsets represent the changes in spiking activity of small (local) and large (global) neuronal areas respectively. Therefore, the subsets with the largest scores represent the neuronal subarea or subareas showing the most significant change in activity. We note the number of subsets we have to calculate increases exponentially with the total number of electrodes. So we can show this using a formula, as it is difficult to show the number of all possible subsets containing connected electrodes, we consider the number of subsets if:

1. we assume every electrode in set Q has sampled neuronal activity.

2. we constructed each subset using the same (chosen) set of $(T/2)$ electrodes from set Q, which we refer to as set A, and a combination of the rest of the electrodes from set Q, which we refer to as set B. Therefore, each subset we obtain contains at least $(T/2)$ electrodes. We choose set A so that all of the electrodes in the set are connected, and, so that each electrode in set B is adjacent (connected) to at least one electrode in set A.

Therefore, with these points in mind, we would obtain $2^{(T/2)}$ subsets. For example, when $T = 4$ we would obtain at least 4 combinations. That is, for 4 electrodes 1,2,3,4, i.e. $1 = q_{1,1}, 2 = q_{1,2}, 3 = q_{2,1}, 4 = q_{2,2}$, and, set A contains electrodes 1 and 2 and set B contains 3 and 4, we obtain combinations of (1,2),(1,2,3),(1,2,3,4),(1,2,4).

So, for any T electrodes we would obtain a:

$$\text{Total number of subsets } > 2^{(T/2)}$$

.

6.3 Modelling Approach

We created an olfactory bulb model, adopted from a simple model (Margrie et al., 2003), for the main purpose of proving that the MEANOVA approach works. To prove this, particular mitral cells were chosen to increase their spiking activity in response to stimulus presentation, so the cells responding most actively were already known to us. We produced 6 datasets, where the datasets correspond to the spiking activity before (3 datasets (trials)) and during (3 datasets (trials)) stimulus presentation. The spiking activity of the cells in each trial set were similar. Furthermore, the chosen responsive cells in the stimulus presentation trial set were the same. We produced and used these datasets with MEANOVA to see if it yields the expected results. That is, we would expect the MEANOVA results to show that the stimulus presented evoked a more significant change in the activity of the chosen cells than the surrounding cells and, furthermore, the change was not evoked by some other random factor. In other words, we would expect the interaction and factor 2 (trial) scores for all of the groups of cells to be < 1, and, the factor 1 (stimulus) scores for the majority of the cells to be > 1. Furthermore, we would expect the factor 1 score to be much greater for the chosen and surrounding cells. The following two subsections, discuss the model we used, and the results gained.

6.3.1 Network Model Used

Simulations were created and performed in Matlab 6.5 (the MathWorks, Natick, MA, USA). The network consisted of a connected grid of mitral cells, which we discuss in more detail below. We modelled the mitral cells as leaky integrate-and-fire neurons, which describe a neuron's membrane potential V over time t. The model we used was from (Abbott et al., 2001) and is described below:

$$\tau_m \cdot \frac{dV}{dt} = (E_l - V) - Pc + R_m \cdot I_e \tag{6.15}$$

This shows the rate of change of the membrane potential is proportional to the rate at which charge builds up inside the cell. The rate of charge buildup is equal to the total amount of current entering into the neuron.

We note the model (equation 6.15) was formed by multiplying the integrate and

fire neuron model:

$$c_m \cdot \frac{dV}{dt} = -g_L(V - E_l) + \frac{I_e}{A} \tag{6.16}$$

by r_m, which, in this case $r_m = 1/g_L$, and, adding a postsynaptic current (Pc). The term R_m ($R_m = r_m \cdot A$) refers to the total membrane resistance, where A refers to the total surface area of the neuron. r_m, c_m and g_L refer to the membrane resistance, capacitance and conductance per unit area of the neuron respectively. We also note the total capacitance of the neuron is $C_m = c_m \cdot A$. Furthermore, I_e is an externally applied (inward) current, E_l is the resting potential and τ_m is the membrane time constant of the neuron, i.e. $\tau_m = r_m c_m$.

For our experiments we used the model (equation 6.15) with τ_m=10ms, E_l=-70mV, R_m=10MΩ, a time step of 1ms, and we implemented a refractory period by clamping V to -70mV for 2ms. Furthermore, an action potential was generated by our model when V reached a threshold level of -55mV. Once a neuron fired we reset V to -70mV. We describe I_e and Pc below.

We used an input(I_e) of:

$$I_e = \text{4Hz oscillation current + fixed current + noise}$$

We injected a 4Hz oscillatory current into each mitral cell with a peak amplitude of 0.95nA. This mimicked several sniff cycles. We also injected each mitral cell with a constant current at the outset of the experiment, which we chose randomly from the range of $[0.6, 0.65]$nA. We also chose the *noise* randomly in the range of $[0, 0.1]$nA from a Gaussian distribution. We chose the inputs from a small range so that each mitral cell differed slightly from the rest and the neuron's firing rate would be similar on different trials. If we used a larger range, the inputs to each neuron on each trial could vary greatly, thereby influencing the MEANOVA interaction scores, i.e producing results larger than 1.

The final input we injected into each mitral cell was a postsynaptic current (Pc), which we describe as $Pc = r_m g_s \cdot P_s \cdot (V - E_s)$. The strength of the connection (conductance) corresponds to the value of $r_m g_s$. We set all of the mitral cell connection strengths to 0.05 nS. To produce an inhibitory connection we set E_s (reversal potential) to -100mV. For an excitatory connection we set E_s to 35mV.

We modelled the synaptic open probability (P_s) variable so that when an action potential occurs at time t:

$$P_s(t + \triangle t) = P_s(t) + P_{max} \cdot (1 - P_s(t))$$

where we use P_{max}=1. Immediately after an action potential the P_s variable decays exponentially to zero using the following equation:

$$\tau_s \cdot \frac{dP_s}{dt} = -P_s$$

where τ_s is the time constant for the connection. We used 2ms and 6ms for the excitatory and inhibitory connections respectively.

To test MEANOVA, i.e. test whether it can detect and show hot spots within a recorded area, we constructed grids of cells and induced hot spots within them. The grid represents a neuronal area that could be recorded by an MEA, and therefore, a hot spot occurring in, for example, the top left part of the grid, corresponds to a hot spot that could occur within the top left part of a neuronal area recorded by the electrodes residing in the top left part of the MEA. We induced hot spots by injecting target cells with a stimulus, and, so that the spiking activity of the cells surrounding the target cells also increased, we connected the cells in the grid so that the elevated spiking rate of the target cells elevated the spiking rate of the surrounding cells. A cluster of heightened activity, therefore, corresponds to a hot spot within the grid. We connected the cells in the grid as follows: we connected each mitral cell to all of its adjacent cells in the grid. For example, corner cells are connected to 3 cells, and, the middle cells within the grid are connected to 8 cells. Target cells receive an excitatory connection from all of its neighbours, and, a target cell's neighbours receive an excitatory connection from the target cell. Cells that are not targets receive inhibitory connections from half of their neighbours and excitatory connections from the other half. The stimulus we injected into the target cells represented an increase and decrease of odour concentration over 1 second. In other words, we injected a stimulus into the target cells throughout the whole simulation every second, which we refer to as t_s, for duration d. The stimulus steadily increased with time to $t = t_s + (d/2)$, from 0nA to a peak of 8.5nA (unless otherwise stated). We then steadily decreased the stimulus to 0nA for the rest of the second, i.e. $t = t_s + d$. We used $d = 1$. Once we sampled the activity of the mitral

cells, we applied MEANOVA to the data to determine whether it could detect/show the manually induced hot spots. We discuss our experiments and the results we found in the next section.

To discuss our results and show where the hot spots occurred within the grid of cells, we associated a double digit number to each mitral cell in the grid and refer to these throughout the results section. We labelled each cell in the same way as the Babraham Institute labelled the electrodes on the multi-electrode arrays they used[2]. The first digit, starting from 0, represents the row position of a cell in the grid, and the second digit, starting from 1, represents the column position of the cell.

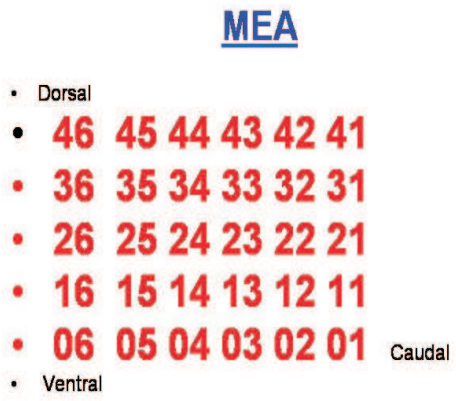


Figure 6.1: This shows how the electrodes on a multi-electrode array (MEA) were numbered in the datasets we acquired from the Babraham Institute

To illustrate the results of MEANOVA and its advantages over MANOVA, we constructed two grids of cells and induced hot spots within them:

1. We first used 16 mitral cells, i.e a 4×4 array, to test MEANOVA. We show the layout of the grid in table 6.1.

2. We next used a 9×9 grid (we show the layout of the grid in section 6.3.4), to show that MEANOVA can detect a significant change in the spiking activity of a few cells amongst a very large group of unresponsive ones.

[2]We used the same numbering system in our simulations as it may reduce confusion when one looks at the MEANOVA results produced, and the electrode (neuron) subsets obtained, from the simulation and biological datasets.

01	02	03	***04***
11	12	***13***	14
21	22	23	24
31	32	***33***	34

Table 6.1: This table shows the layout of the 4×4 grid of cells we used. The target neurons are shown in bold and italic.

We then generated spiking activity from the grids for 10 seconds, 6 times (i.e. we generated 3 sets of spiking activity from the grid when a stimulus was presented and when it was not). As discussed earlier, we produced 6 datasets to use with MEANOVA to determine whether it could correctly show that the stimulus factor had a significant effect on the spiking activity of the neurons and the trial factor did not. This is what the MEANOVA results showed.

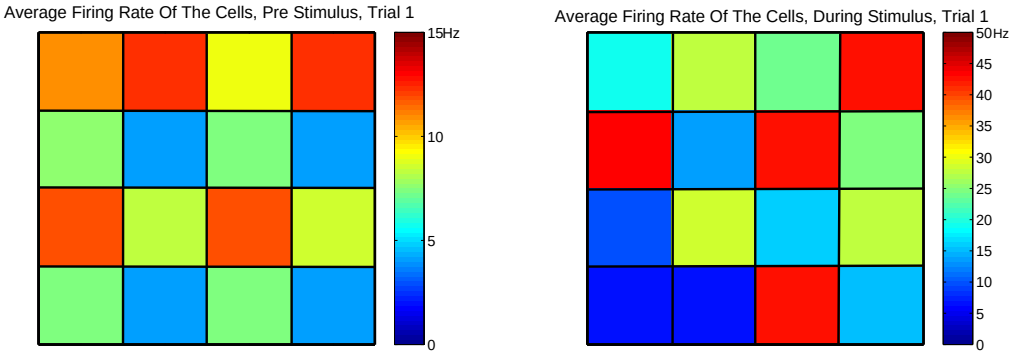
We note that we excluded any cells that showed a firing rate $< 5Hz$ during stimulus presentation from our analyses of the neurons in the 4×4 grid. We excluded these for clarity. That is, we lowered the number of combinations MEANOVA could produce from the grid of cells, so we could easily introduce (discuss), and show, all of the scores MEANOVA can produce from a number of neurons.

We firstly discuss the 4×4 grid simulation and the MEANOVA results we acquired from the data generated.

6.3.2 The Results Of Using MEANOVA with the Data Generated From The 4×4 Grid Of Mitral Cells

In fig. 6.2 we show the average firing rate of the cells in the 4×4 grid across the 10 second simulation period when no stimulus was present (Panel (a)) and when we injected a stimulus into 4 target cells (04,11,13,33) (Panel (b)) (to see the positions of these cells in the grid we refer the reader to table 6.1). We note that the activity represented in fig. 6.2 is from trial 1 and was similar across trials 2 and 3 (results not shown). Each square in the grid represents the activity of one mitral cell. The activity of each cell was in the range of 5-13hz when no stimulus was present (Panel (a)) .

By analysing both grids (Panels (a) and (b)) we can see that the stimulus injected



(a) (b)

Figure 6.2: This show the average firing rate of the cells in the 4×4 grid across the 10 second simulation, before (Panel (a)) and during (Panel (b)) stimulus presentation. These results are from trial 1, and were similar across trials (results not shown). Panel (b) shows that the firing rates of the four chosen mitral cells, i.e. 04,11,13,33 (red cells) and their surrounding cells increased.

into the 4 target cells elevated their firing rates (red cells) (Panel (b)), i.e. the firing rates of these cells increased to be in the range of 30-40hz. The spiking activity of the cells surrounding the target cells also increased (Panel (b)), i.e. the firing rates of these cells increased to be in the range of 15-25Hz.

These results, therefore, show that the spiking activity of the chosen and surrounding cells increased to stimulus presentation. We next used MEANOVA on this data to see if it yields the expected results, i.e. whether it shows the spiking activity[3] of the target, and their surrounding, cells increased, and, that the spiking activity of the chosen cells increased the most. We show the results in the subsequent sections.

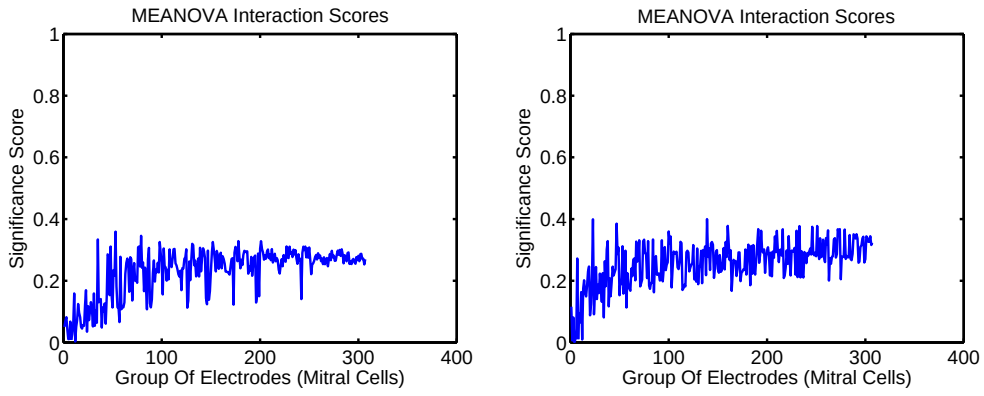
MEANOVA Interaction Significance Scores

We firstly note that the interaction, factor 1 and 2 scores tended to be dependent on the number of time bins and the simulation duration used[4]. That is, when we analysed the significance scores produced from small to large subsets, the scores

[3]We note that we refer to the spiking activity of a neuron as the number of spikes fired during the simulation time.

[4]We note that the number of spikes contained in the time bins for each neuron on all our tests (trials) followed a normal distribution, i.e. (Kolmogorov-Smirnov, $P < 0.05$), which is a requirement of the MEANOVA test.

stabilised around different values. For example, in the 10 and 8 second simulations the interaction significance scores began to stabilise around 0.25 when we used 320 time bins (see fig. 6.3). In a 2 second simulation the interaction significance score began to stabilise around 0.4 and 0.3 when we used 20 and 40 time bins respectively (results not shown). In a 4 second simulation (results not shown), the interaction significance score began to stabilise around 0.4 and 0.25 when we used 20 and 40 time bins respectively.



(a) (b)

Figure 6.3: The Interaction scores produced from the 8s (Panel (a)) and 10s (Panel (b)) simulations when we used 320 time bins. In both panels, group numbers 1 to 307 show the scores produced from small to large subsets of cells respectively. In other words, the first 12 scores represent the change in spiking activity of the individual cells, and the final score represents the change in the activity of all of the cells as a whole. For more detail, i.e. to see the cell combination that each group number represents, we refer the reader to Appendix B. These results show that all the interaction scores are < 1, i.e. there was no interactions between factor 1 (stimulus) and factor 2 (trials).

The results show the scores can stabilise around different values when we use different sampling periods or different numbers of time bins. In other words, the sampling period and number of time bins can affect the reliability of the information produced. This happens because, firstly, the sampling period is not long enough to sufficiently represent the true spiking activity of the mitral cells. Secondly, a low number of time bins provides insufficient detail as we are analysing less points in time. Therefore, we assume that the scores produced using a sampling period of 8s or 10s with 320 time bins are reliable due to the scores within these simulations stabilising at the same value. We therefore use these durations and time bins

throughout the chapter (unless otherwise stated).

We examine the interaction scores to see if factors 1 and/or 2 have a clear effect on the neuronal activity. Any interactions, i.e scores > 1, would show that other factors are present which are affecting the neuronal activity. If this is the case, it would be harder to interpret if factor 1 or factor 2 are having any effect, and therefore, would show no cause to investigate the two factors separately. If there are no interactions, i.e scores < 1, then we can conclude that any changes in neuronal activity was caused by either factor 1 and/or factor 2.

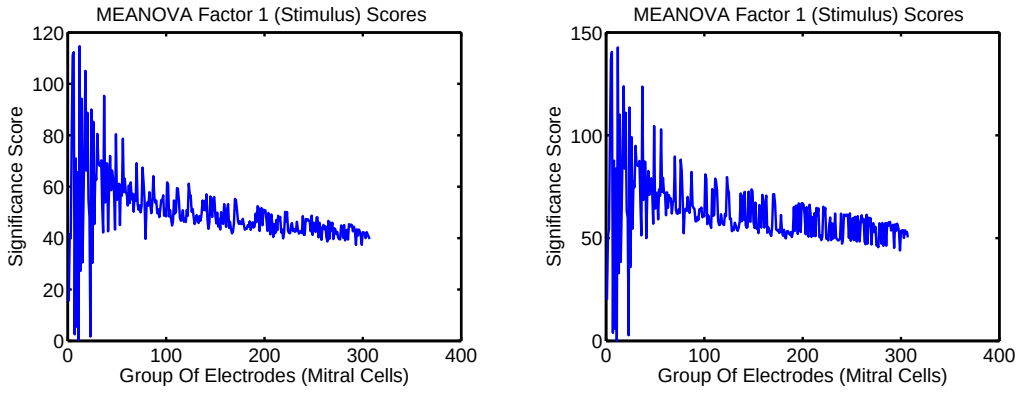
The interaction scores produced from the 8s and 10s simulations (fig. 6.3) show that all the interaction scores produced from all of the cell combinations are under 1. That is, there was no significant interaction between factor 1 (stimulus) and factor 2 (trials). Thus, it is possible to determine from these datasets whether factor 1, factor 2 or both caused any change in the spiking rates of the neurons sampled.

In the subsequent sections, we show the MEANOVA factor 1 and factor 2 scores produced from the 8s and 10s simulation datasets.

MEANOVA Factor 1 (Stimulus) Significance Scores

We show the factor 1 scores for the 8s and 10s simulations in fig. 6.4 and the scores corresponding to the individual cells in table 6.2.

The scores show the effects factor 1 (stimulus) had on the spiking activity of individual and groups of cells. The results show that the scores produced from many of the cell groups were > 1. In other words, the groups exhibiting a score > 1 contained one or more of the responsive neurons, i.e. the chosen neurons and/or its surrounding cells. The highest scores produced correspond to the individual chosen cells. For example, the significance score produced from cell 13 (chosen cell) was 140.5853 in the 10 second simulation. In addition, the highest scores produced from each group size corresponded to the groups containing the single cells with the highest scores, i.e. the most responsive chosen and surrounding cells. For example, in the 10 second simulation, cell combination (01,11) (group number 14) conveyed a score of 110, where 11 was a chosen cell and 01 was next to 11 in the grid. However, cell combination (21,31) (group number 23) conveyed a score of 2.71, where 21 was



(a) (b)

Figure 6.4: The factor 1 (stimulus) results for the 8s (Panel (a)) and 10s (Panel (b)) simulations when we used 320 time bins. In both panels, group numbers 1 to 307 show the scores produced from small to large subsets of cells respectively. In other words, the first 12 scores represent the change in activity of the individual cells, and the final score represents the change in the activity of all of the cells as a whole. For more detail, i.e. to see the cell combination that each group number represents, we refer the reader to Appendix B. The results show that the scores produced from many of the cell groups were > 1. In other words, the groups exhibiting a score > 1 contained one or more of the responsive neurons. The highest scores produced correspond to the individual chosen cells. In addition, the highest scores produced from each group size corresponded to the groups containing the single cells with the highest scores, i.e. the most responsive chosen and surrounding cells. The scores corresponding to the individual cells are shown in table 6.2.

Group Number	Cell Identity	Score (8s)	Score (10s)
1	01	15.34	19.9811
2	02	41.4529	50.6718
3	03	40.5671	54.5224
4	***04***	83.056	100.3722
5	***11***	111.1561	136.8276
6	***13***	112.3447	140.5853
7	21	1.5754	2.9188
8	22	71.274	88.0452
9	23	5.5029	5.8454
10	24	65.7228	83.7931
11	31	0.73358	0.826792
12	***33***	114.6635	142.7194

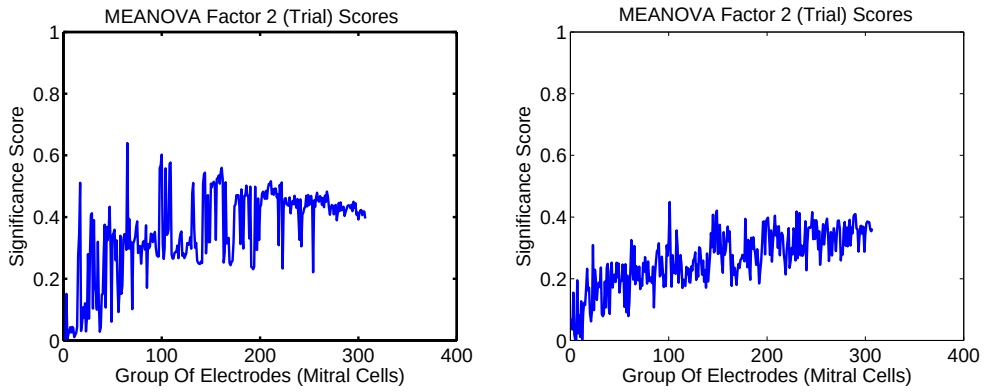
Table 6.2: This table shows the significance factor 1 scores produced from the spiking activity of all the single cells in the 4×4 grid during the 8s and 10s simulation period. The scores show the effect factor 1 (stimulus) had on the spiking activity of the individual cells. The scores produced from the 10s simulation were larger than the 8s simulation. That is, the longer duration revealed the truer spiking activity of the cells, which therefore, increased the likelihood that the change evoked by factor 1 was significant. We can also see that the highest scores correspond to the chosen cells (shown in italic), i.e. the chosen cells fired more spikes in response to a stimulus than the other cells. In addition, the scores produced from the chosen cells neighbours were lower than the chosen cells. This is because the stimulus evoked a smaller change in the chosen cells neighbours compared to the chosen cells. We can also see from the results that the stimulus did not significantly change the firing rate of Cell 31 as it was not a chosen cell and, furthermore, it was not connected to any of the target cells.

adjacent to the chosen cell 11, and, 31 was not a chosen cell or next to any of the chosen cells. The lowest scores, i.e. < 1, correspond to the cells that were not chosen, or, were not adjacent to the target cells. For example, cell 31 conveyed a score of 0.826792 in the 10 second simulation.

The highest scores produced by MEANOVA correspond to the cells selected to be the most responsive to the stimulus and the cells surrounding them, thus, MEANOVA has clearly indicated the hot spots in the area.

MEANOVA Factor 2 (Trials) Significance Scores

Fig. 6.5 shows the factor 2 (trial) significance scores, which indicate whether the spiking activity of the cells changed across trials. The scores are < 1, which show that factor 2 did not significantly change the spiking activity of the neurons. However, the results show there were some slight (non-significant) changes in mitral cell activity across trials, which were due to the random inputs injected into each cell.



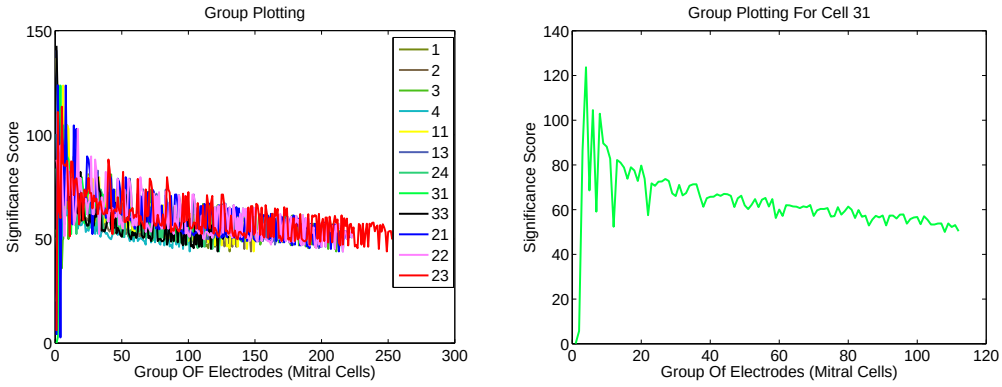
(a) (b)

Figure 6.5: The factor 2 scores (trial factor) for the 8s (Panel (a)) and 10s (Panel (b)) simulation when we used 320 time bins. In both panels, group numbers 1 to 307 show the scores produced from small to large subsets of cells respectively. In other words, the first 12 scores represent the change in activity of the individual cells, and the final score represents the change in the activity of all of the cells as a whole. For more detail, i.e. to see the cell combination that each group number represents, we refer the reader to Appendix B. The results show that all of the scores are < 1. This, therefore, shows there was no significant change in the spiking activity of the neurons due to the trial factor.

From analysing the MEANOVA results, we would conclude that the stimulus evoked a more significant change in the spiking activity of some neurons, i.e. neuron areas. Furthermore, MEANOVA has shown that the cells selected to be the most responsive at the outset of the simulation, and the surrounding cells, were the ones that responded most actively.

Group Plotting

In order to easily assess the impact of stimulus presentation on the spiking activity of each cell and its surroundings, we can plot the MEANOVA scores for each cell and all groups containing that cell. For example, we show the scores produced from all cells from the 10s simulation in fig. 6.6 Panel (a). In fig. 6.6 (Panel (b)),



(a) (b)

Figure 6.6: Group plot of MEANOVA scores. Panel (a), shows the significance score for all 12 cells and their associated groups. Panel (b) shows the significance scores for the groups containing cell 31. The first point corresponds to cell 31, the second point to the cell combination (21,31), the third point to (11,21,31) etc. We show all the groups containing cell 31 in Appendix B.

we show the significance score for all of the groups containing cell 31.

6.3.3 Results Generated From a 9×9 Grid Of Mitral Cells

In this section we used the spiking data produced from a 9×9 grid of mitral cells with MEANOVA. We implemented this to demonstrate that if there is a small stimulus-related activity change in small areas of the grid, MEANOVA can detect these areas and show which were the most responsive. We connected the cells in the same way

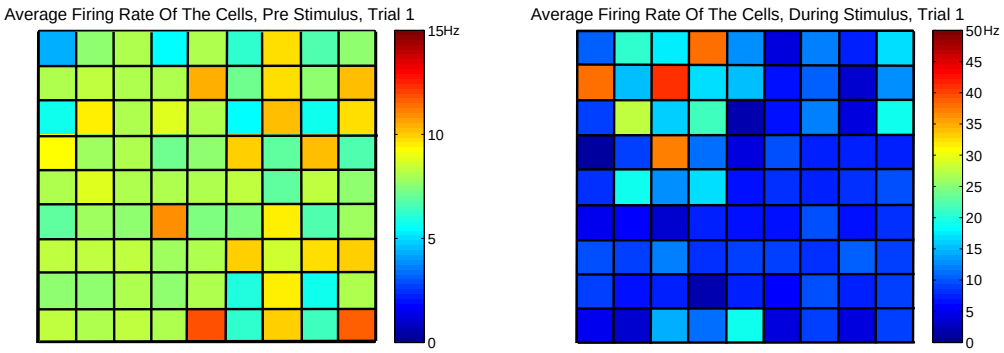
as the 4×4 grid. We conducted three tests:

- For the first test, 4 cells were selected as responsive, and, they strongly influenced the spiking activity of their adjacent cells.
- For the second test, 4 mitral cells were selected as responsive, and, did not influence the spiking activity of their adjacent cells.
- In the third and final test only a single mitral cell was selected as responsive, and, did not influence the spiking activity of its adjacent cells.

In these tests we presented the network with a lower stimulus strength of 7.5nA, to show the effects a smaller change in spiking activity has on the significance scores. We calculated the scores for a maximum number of 50000 groups, which was a reasonable ceiling as by this point the score stabilised. For each test, we conducted six 10s simulations (where the datasets correspond to the spiking activity of the cells before (3 datasets (trials)) and during (3 datasets (trials)) stimulus presentation) and we used 320 time bins. The inputs injected into each cell were fixed across all tests and trials. The cells we selected as responsive were the same across all tests. The target cells were (04, 11, 13, 33). We show the 9×9 grid layout, and where these target cells reside, in table 6.3.

01	02	03	***04***	05	06	07	08	09
11	12	***13***	14	15	16	17	18	19
21	22	23	24	25	26	27	28	29
31	32	***33***	34	35	36	37	38	39
41	42	43	44	45	46	47	48	49
51	52	53	54	55	56	57	58	59
61	62	63	64	65	66	67	68	69
71	72	73	74	75	76	77	78	79
81	82	83	84	85	86	87	88	89

Table 6.3: This shows the 9×9 grid Layout. The target neurons were 04, 11, 13, 33, and are shown in italic.



(a) (b)

Figure 6.7: The average firing rate of the cells in the 9x9 cell grid across the 10s simulation, before (Panel (a)), and during (Panel (b)) stimulus presentation. These results are from trial 1, and were similar across trials (results not shown). We selected cells 04, 11, 13, 33 to be responsive (to see the positions of these cells in the grid we refer the reader to table 6.3) and each chosen cell influenced the activity of their adjacent cells. This is shown in (Panel (b)) where the activity of the chosen and the surrounding cells increased when we presented a stimulus. Note that the increase in firing rates was lower compared to the 4x4 grid in figure 6.2, as we injected a weaker stimulus into this network.

Test 1 Results

The interactions, and the trial factor scores (factor 2), were all under 1 (results not shown). This was the same for tests 2 and 3.

In fig. 6.7 we show the average firing rate of the cells in the 9×9 grid across a 10 second simulation period when no stimulus was present (Panel (a)) and when we injected a stimulus into 4 target cells (04,11,13,33) (Panel (b)) (to see the positions of these cells in the grid we refer the reader to table 6.3). The target cells in this test strongly influenced the activity of their adjacent cells. We note the spiking activity in Panels (a) and (b) were from trial 1. Trials 2 and 3 showed similar activity (results not shown). We can see in Panel (b) that the firing rates of the chosen and surrounding cells were greater (i.e. the firing rates of the neurons were in the range of 20-45Hz) when we presented a stimulus compared to when we did not (i.e. the firing rates of the neurons were in the range of 5-12Hz (Panel (a)). The scores in fig. 6.8 show the effects factor 1 (stimulus) had on the spiking activity of the individual

and groups of cells. Table 6.4 (test 1 column) shows the scores produced from the individual cells (chosen and surrounding). The results show that the scores produced from many of the cell groups were > 1. In other words, the groups exhibiting a score > 1 contained neurons from the top left quadrant of the grid (fig. 6.7 Panel (b)), i.e. one or more of the chosen neurons and/or the chosen neurons neighbours. The highest scores produced correspond to the individual chosen cells. For example, the score produced from the chosen cell 11 was 52.8 and the score produced from cell 81 (not a chosen cell or not adjacent to a chosen cell) was 0.02. In addition, the highest scores produced from each group size corresponded to the groups containing the single cells with the highest scores, i.e. the most responsive chosen and surrounding cells. For example, the score produced from cell combination (11,12) (a chosen cell and its neighbour, from the top left quadrant of the grid (Panel (b))) was 39.7766, and, the score produced from cell combination (67,77) (both cells were not targets or neighbours to the targets, i.e. were from the bottom right quadrant of the grid (Panel (b))) was 0.42871. Table 6.4 (test 1 column), shows the scores for some of the chosen, and their surrounding, cells. The scores acquired from the spiking activity of the chosen and surrounding cells were not as high as the 4x4 grid results, as we injected a lower stimulus strength into the 9×9 network.

Test 2 Results

Fig. 6.9 (Panel (a)) shows the average firing rate of the cells in the 9×9 grid across a 10 second simulation period when we injected a stimulus into 4 target cells (04,11,13,33) (to see the positions of these cells in the grid we refer the reader to table 6.3). The target cells in this test did not influence the activity of their adjacent cells. We note the spiking activity in Panel (a) was from trial 1. Trials 2 and 3 showed similar activity (results not shown)). We can see that the activity of only the chosen cells increased when we presented a stimulus (i.e. the firing rates of the chosen cells increased to be in the range of 30-35Hz), (see fig. 6.7 Panel (a) for the pre stimulus activity). Again, the interaction, and the trial factor (factor 2) scores produced from the simulation data were all under 1 (results not shown). The scores in fig. 6.9 (Panel (b)) show the effects factor 1 (stimulus) had on the spiking activity

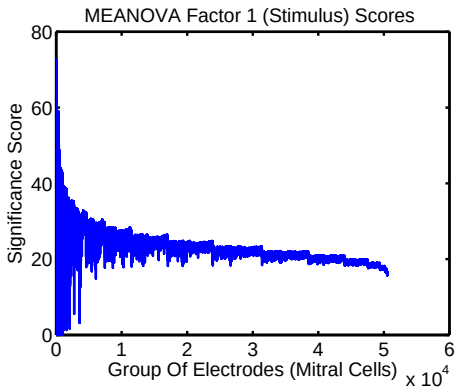


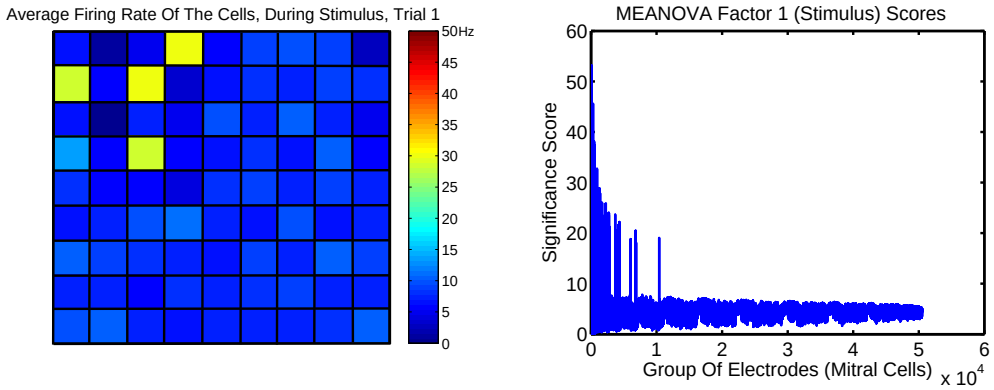
Figure 6.8: The scores show the effect factor 1 (stimulus) had on the spiking activity of the individual and groups of cells. Group numbers 1 to 50000 show the scores produced from small to large groups of cells respectively. In other words, the first few scores (groups 1-81) represent the change in activity of one cell, and the final score represents the change in the activity of all of the cells as a whole. The results show that the scores produced from many of the cell groups were > 1. In other words, the groups exhibiting a score > 1 contained one or more of the chosen neurons and/or the chosen neurons neighbours. The highest scores produced correspond to the individual chosen cells. In addition, the highest scores produced from each group size corresponded to the groups containing the single cells with the highest scores, i.e. the most responsive chosen and surrounding cells. See Table 6.4, test 1 column for the scores associated to the individual cells. We note that all of the scores are lower than the scores acquired from the 4×4 grid (shown in figure 6.4), as we injected a lower stimulus strength into the 9×9 grid.

of the individual and groups of cells. The results show that the scores produced from many of the cell groups were > 1. In other words, the groups exhibiting a score > 1 contained one or more of the responsive (chosen) neurons. The highest scores produced correspond to the individual chosen cells, i.e. the cells from the top left quadrant of the grid (Panel (a)). In addition, the highest scores produced from each group size corresponded to the groups containing the single cells with the highest scores, i.e. the most responsive chosen cells. For example, the group of cells (11, 21, 01), which contains one of the chosen neurons, conveyed a score of 2.6, whereas a group containing 2 responsive neurons (11, 12, 13) conveyed a score of 28.6. Some of the individual cells had a score < 1 as the stimulus did not effect their spiking activity. We show the scores for the chosen and some of the surrounding cells in table 6.4 (test 2 column). We note that the scores produced from all of the groups were less than the scores produced in test 1, as the excitatory connections between all of the cells were disconnected, i.e. the spiking activity of less cells were changed by the presence of the stimulus.

Test 3 Results

For our last simulation, we selected one neuron to be responsive, which was cell 33 (to see the position of this cell in the grid we refer the reader to table 6.3). This cell did not influence the activity of its adjacent cells. Furthermore, we injected the cell with a weak stimulus strength of $0.5nA$. That is, the activity of cell 33 only slightly increased, i.e. it increased from 7Hz to 13Hz, when we presented the stimulus (see fig. 6.10 Panel (a), which shows the average firing rate of the mitral cell grid across the 10s simulation period. We note the spiking activity in Panel (a) was from trial 1. Trials 2 and 3 showed similar activity (results not shown)) (see fig. 6.7 Panel (a) for the pre stimulus activity)). Again, the interaction, and the trial factor (factor 2) scores produced from the simulation data were all under 1 (not shown). The scores in fig. 6.10 Panel (b) show the effects factor 1 (stimulus) had on the spiking activity of the individual and groups of cells.

The results show that the stimulus significantly changed the activity of a few groups of cells as they had a score > 1. That is, any group producing a score > 1



(a) (b)

Figure 6.9: The grid in Panel (a) shows the average firing rate of cells in the 9x9 cell grid across a 10 second simulation during stimulus presentation. These results are from trial 1, and were similar across trials (results not shown). We selected cells 04, 11, 13, 33 to be responsive (to see the positions of these cells in the grid we refer the reader to table 6.3), and, they did not influence the activity of their adjacent cells. We can see this in (Panel (a)) where the activity of only the chosen cells increased when we presented a stimulus (see fig. 6.7 Panel (a) for the pre stimulus activity), but the activity of the surrounding cells were similar (table 6.3 shows where these responsive cells reside in the grid). The scores in Panel (b) show the effects factor 1 (stimulus) had on the spiking activity of the individual and groups of cells. Group numbers 1 to 50000 show the scores produced from small to large groups of cells respectively. In other words, the first few scores (groups 1-81) represent the change in activity of individual cells, and the final score represents the change in the activity of all the cells. The results show that the scores produced from many of the cell groups were > 1. In other words, the groups exhibiting a score > 1 contained one or more of the responsive (chosen) neurons. The highest scores produced correspond to the individual chosen cells. In addition, the highest scores produced from each group size corresponded to the groups containing the single cells with the highest scores, i.e. the most responsive chosen cells. We show the scores for some of the individual cells in Table 6.4 (test 2 column).

Group Number	Mitral Cell Identity	Score Test 1	Score Test 2	Score Test 3
2	02	30.252	0.321	0.09
3	03	23.137	0.576	0.067
4	***04***	60.525	54.7198	0.00226
5	05	2.219	0.019	0.02
10	***11***	52.859	44.4104	0.0039322
11	12	14.592	0.79	0.025
12	***13***	53.582	47.8202	0.070774
13	14	19.829	0.15	0.064
14	15	8.274	0.572	0.005
20	21	6.719	0.247	0.4553
21	22	36.343	0.569	0.4110
22	23	18.56	0.389	0.0054
23	24	26.287	0.4168	0.051
31	32	2.583	0.256	0.094
32	***33***	51.331	46.5248	3.7146
33	34	6.056	0.064	0.026
34	35	0.031	0.495	0.112
40	41	0.022	0.015	0.064
41	42	21.958	0.067	0.009
42	43	10.079	0.385	0.0646

Table 6.4: This table shows the scores produced from the chosen and a few surrounding cells in the 9×9 grid used in all three tests. The table shows the highest scores from tests 1 and 2 correspond to the four cells chosen to be responsive (shown in italic). The chosen cells only influenced the activity of their adjacent cells in test 1. The table also shows the scores from test 3, where we chose cell 33 to be responsive. The test 1 scores show that some of the cells adjacent to the target ones also had a score > 1 due to the excitatory connections between the cells. The scores corresponding to the chosen cells in test 1 are slightly higher than in test 2, as the excitatory connections between the cells in test 2 were disconnected. That is, the stimulus in test 2 had less influence on the spiking activity across the grid of cells. The chosen cells in tests 2 and 3 only had scores > 1, as the stimulus did not change the activity of the other cells. In test 3, the score for cell 33 was lower compared to the other 2 tests as it fired less spikes in test 3, as we injected a weaker stimulus into the cell. We also note the scores produced from all of the cells varied across tests, due to the random inputs we injected into each of the cells.

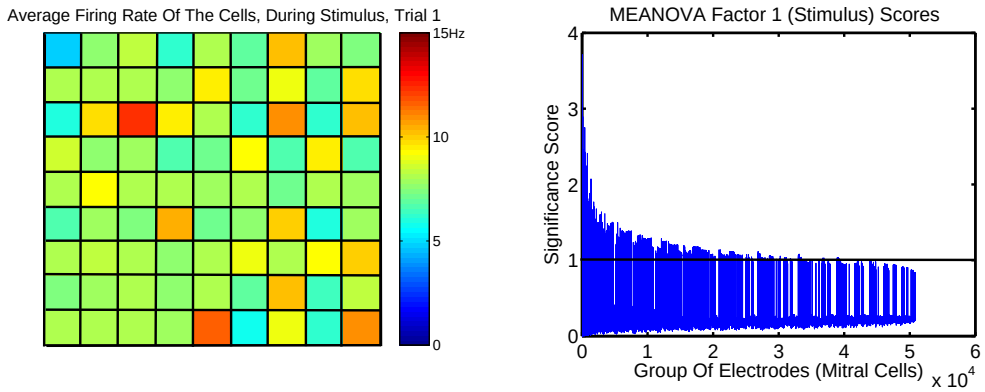
contained cell 33, which was the only cell that responded to the stimulus. Furthermore, the highest score corresponds to cell 33. Thus, with only one significantly responsive neuron, analysis of the entire group of neurons, i.e. group number 50000, reveals no significant change in activity, bearing out the assumption that brute-force application of MANOVA buries sparse information. It also demonstrates that a slight increase in activity can be found by MEANOVA and linked to a presented stimulus.

The simulation results have shown that using MEANOVA can be important in practical applications. This is because MEANOVA can detect one or more neuronal areas, large or small, which have responded to either a high or low level of stimulus. In addition, MEANOVA can also show the most responsive areas, and, the results produced from MEANOVA show that the spiking activity of the cells changed are due to the presented stimulus and not by chance. We next apply MEANOVA to biological data and show the results in the next section.

6.4 Application to Recordings from Rat Olfactory Bulb

By using MEANOVA to analyse the neuronal activity captured by an MEA in the olfactory bulb of the anaesthetised rat, we hope to find areas (hotspot) of neurons that were the most responsive, i.e. showed the most significant change in activity, when we presented a stimulus (odour). We can use MEANOVA to address many questions, for example, we can determine whether a stimulus is represented spatially in small groups, i.e in few neurons, or in a much larger area. We can also use MEANOVA to determine, due to the sequential trials, whether the neurons respond to a repeated presentation of a stimulus. Examining the results created from MEANOVA will give a quick and clear interpretation of the changes in the neuronal activity recorded by an MEA when a stimulus is present. We note the data used in this section was provided by the Babraham Institute. The experiment procedures were discussed in chapter 3.

So we could use MEANOVA on the data from the rat olfactory bulb, we used the spike detection stage discussed in chapter 3 and the spike sorting process discussed



(a) (b)

Figure 6.10: The grid in Panel (a) shows the average firing rate of the cells in the 9x9 cell grid across the 10 second simulation during stimulus presentation. These results are from trial 1, and were similar across trials (results not shown). We selected cell 33 to be responsive (to see the position of this cell in the grid we refer the reader to table 6.3), and, did not influence the activity of its adjacent cells. This is shown in (Panel (a)) where the activity of the chosen cell increased when we presented a stimulus, but the activity of the surrounding cells did not. The scores in Panel (b) show the effects factor 1 (stimulus) had on the spiking activity of the individual and groups of cells. Group numbers 1 to 50000 show the scores produced from small to large groups of cells respectively. In other words, the first few scores (1-81) represent the changes in activity of individual cells, and the final score represents the change in the activity of all of the cells as a whole. The results show that the stimulus significantly changed the activity of some groups of cells as they had a score > 1. That is, the groups of cells with a score > 1 contained cell 33. We also note that when the spiking activity of all of the cells were analysed as a whole (group 50000), the score was < 1 suggesting there was no significant change in the spiking activity of the population (this is the result MANOVA would produce), which is incorrect, as cell 33 responded, i.e. its score was > 1.

in chapter 5 on the spiking data recorded by each electrode. We note that these processes extracted between 3-8 neurons from each electrode recording.

As discussed in the simulation section the recording time and the width of the time bins need to be chosen wisely when analysing neural activity, so we can obtain, and interpret, accurate scores from the MEANOVA process. The parameter and experiment details we used to acquire MEANOVA scores are as follows:

We used the spiking data recorded on every trial from a 5s sampling period, i.e. $T = 5s$. To acquire the MEANOVA scores, we used $M = T/t = 25$ time bins for each cell, i.e. t = 200ms. We note that using a larger t did not change the MEANOVA results, and therefore, we assume the results acquired using $t = 200$ ms are reliable. We also note the spike count distributions, i.e. the number of spikes contained in the M time bins for every cell on every trial were normally distributed (Kolmogorov-Smirnov, $P < 0.05$), which is a requirement of the MEANOVA test. We note that we excluded the electrodes on the MEA that did not record any spiking activity from the analyses. In this case, only 11 electrodes sampled spiking activity.

6.4.1 Results

In accordance with MEANOVA, we can compare neuronal activity with many factors: trials, odours, concentration etc. However, here we consider two factors: pre and during stimulus (factor 1), and different trials (factor 2). 4 odours were presented

In figs.6.11, 6.12 and 6.13 we show the significance scores produced by small to large electrode subsets respectively. In other words, the first 11 scores are produced from the neurons recorded by the individual electrodes, and the final score is produced from all of the neurons recorded by all of the electrodes.

Interactions

In fig. 6.11 we show the interaction results. As we mentioned earlier, in order to assess the changes caused by each factor, we need to look at the effect of interactions first.

We can see that the interaction scores are generally smaller than 1. However, for

odours 1, 2, and 4, some scores are greater than 1, indicating that we have to be careful when we interpret these results.

For odour 3, it is perfectly safe to look at the effect of each factor.

In the next two sections we investigate the effect of individual factors on the neuronal behaviour.

Stimulus

Firstly, we tested whether there were significant changes between pre- and during-stimulus activity (see fig. 6.12). The results show some significant changes, i.e some of the areas recorded by the MEA achieved scores > 1 for the first and the third odours. Some of the scores are higher than others, which means that some parts of the olfactory bulb were responding more actively than others. For example, the scores produced from the group of neurons recorded by electrodes (43 44) (group 20) when odour 1 and odour 3 were presented was 2.35 and 0.24 respectively. In other words the neurons recorded by electrodes 43 and 44 were more responsive to odour 1 than odour 3. From all the results, we can conclude that not all of the recorded area responded to a stimulus, but a proportion did.

The second and fourth odours presented to the olfactory bulb show that although the activity recorded by some electrodes showed significant changes, the fluctuations were large and possibly random.

Trials

We now use MEANOVA to determine whether the neural activity significantly changed across trials. That is, whether repeated stimulus presentation had any effect on the spiking activity. We show the results in fig. 6.13.

From the results, we can see that the trial factor is quite significant, scoring much higher than the score for the comparison between the pre- and during-stimulus activity. This change in activity is showed over the entire area recorded by the MEA, i.e. all of the significance scores are > 1. Remembering that the recordings were made in a sequential way, the results imply that the effect of habituation becomes dominant within a five second period of odour delivery. To check our results, we performed further tests (Kolmogorov-Smirnov test) on single neuron activity. For

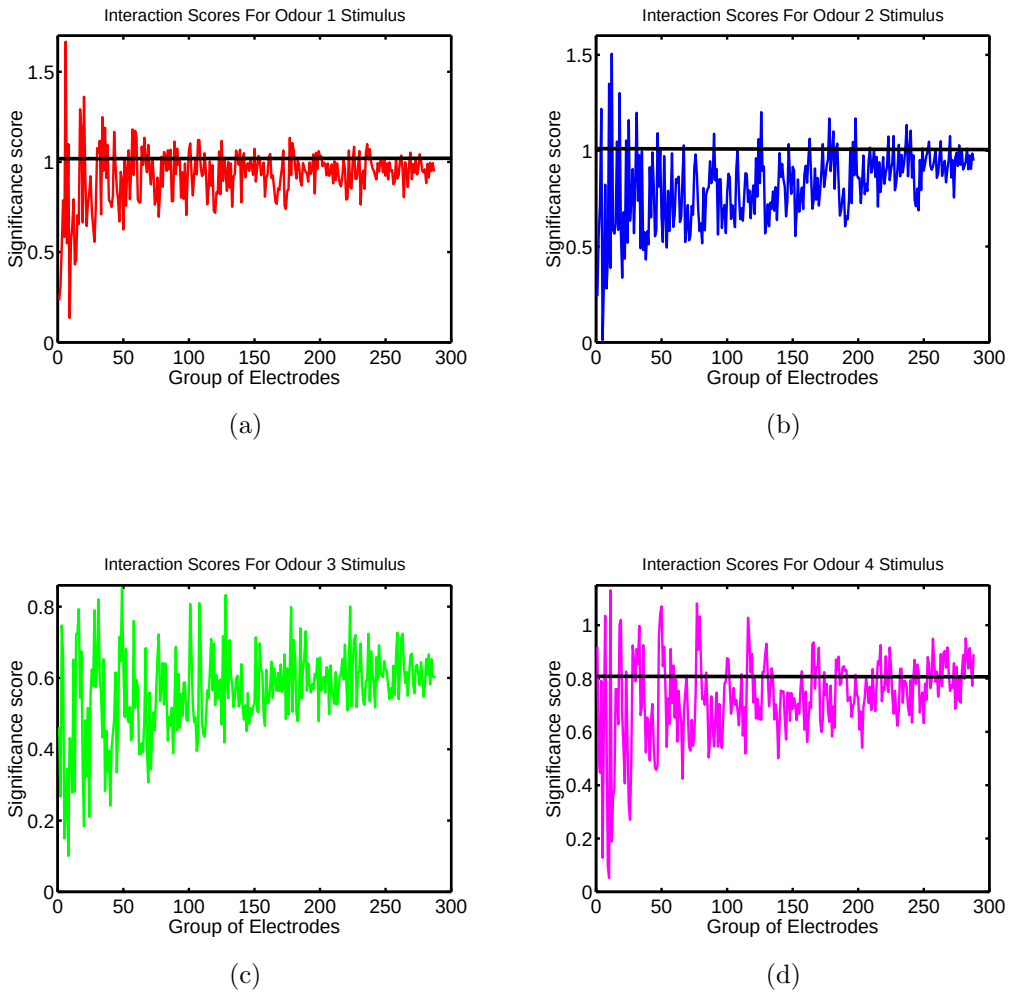


Figure 6.11: MEANOVA interaction results: odour 1 (Panel (a)), 2 (Panel (b)), 3 (Panel (c)) and 4 (Panel (d)). The Group numbers 1 to 285 in all of the graphs show the scores produced by small to large electrode subsets respectively. In other words, the first 11 scores represent the change in spiking activity of the groups of neurons recorded by each of the 11 electrodes (i.e. the electrodes that sampled activity), and the final score represents the change in the activity of all the neurons recorded. For more detail, i.e. to see the electrode combination that each group number represents, we refer the reader to Appendix A. The results show that the scores produced from all of the electrode combinations when odour 3 was presented (Panel (c)) are < 1. When odours 1,2,4 were presented, some electrode combinations were below 1, i.e. they show there were no interactions between factor 1 (stimulus) and factor 2 (trials). Therefore, we can conclude that any changes in neuronal activity recorded by all of the electrode combinations when odour 3 was presented, and some of the electrode combinations when odours 1,2 and 4 was presented, was caused by factor 1 (stimulus), factor 2 (trials (repeating stimulus presentation)) or both.

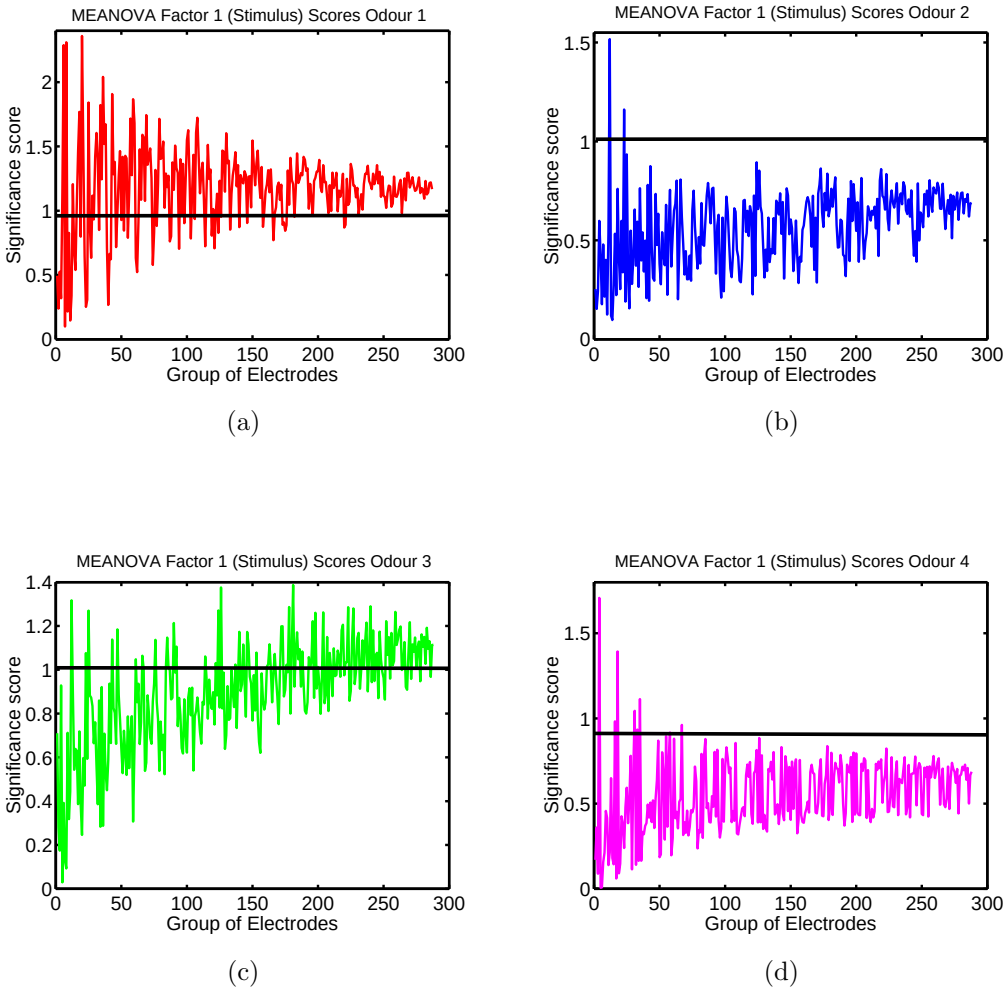


Figure 6.12: The scores show the effect factor 1 (stimulus) had on the spiking activity recorded by individual and subsets of electrodes when we presented odour 1 (Panel (a)), 2 (Panel (b)), 3 (Panel (c)) and 4 (Panel (d)). The Group numbers 1 to 285 in all of the graphs show the scores produced by small to large electrode subsets respectively. In other words, each of the first 11 scores represent the change in the spiking activity of each group of neurons recorded by each of the 11 electrodes (i.e. the electrodes that sampled activity), and the final score represents the change in the spiking activity of all the neurons recorded. For more detail, i.e. to see the electrode combination that each group number represents, we refer the reader to Appendix A. The results show that the scores produced from some of the electrode groups (subsets) were (> 1) when a particular odour was presented, but, < 1 when other odours were presented. In other words, the results show that presenting different odours changed the spiking activity of different neuronal areas in the olfactory bulb.

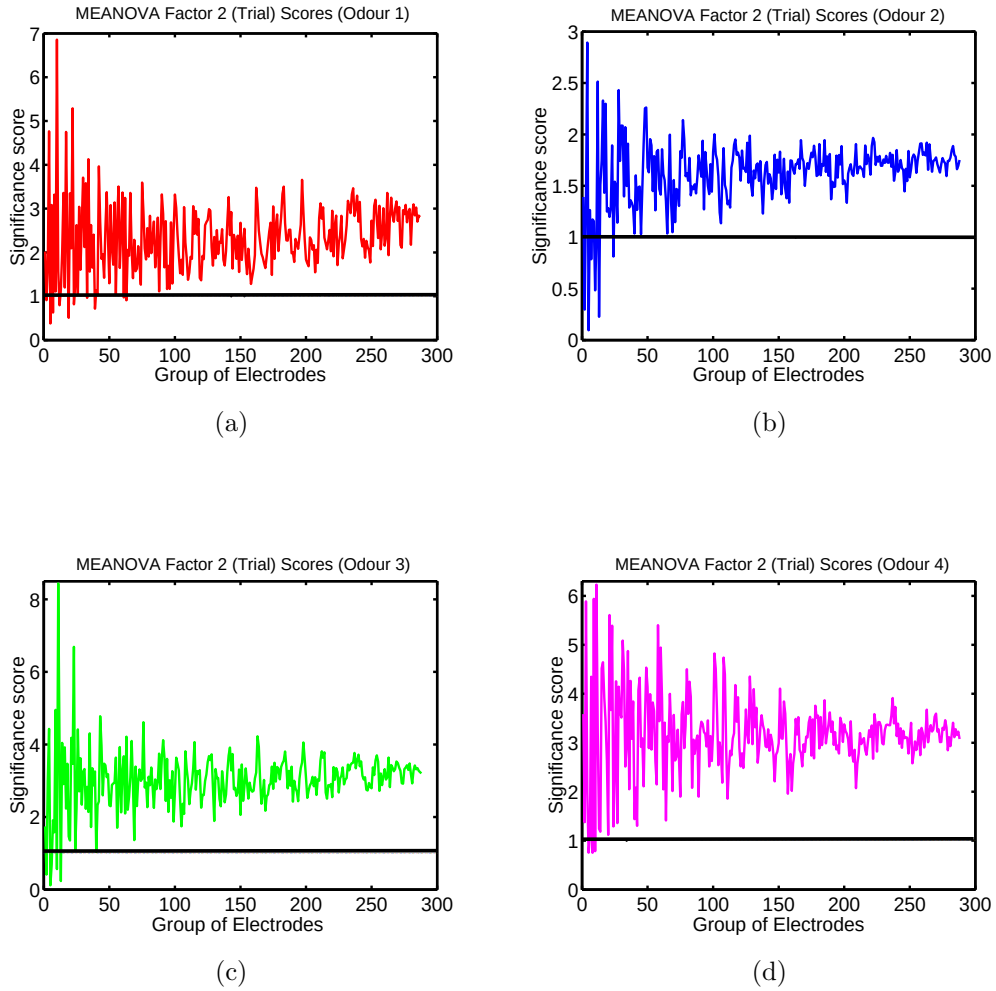


Figure 6.13: The scores show the effect factor 2 (trial factor) had on the spiking activity recorded by individual and subsets of electrodes when we presented odour 1 (Panel (a)), 2 (Panel (b)), 3 (Panel (c)) and 4 (Panel (d)). The Group numbers 1 to 285 in all of the graphs show the scores produced by small to large electrode subsets respectively. In other words, the first 11 scores represent the change in the spiking activity of the groups of neurons recorded by each of the 11 electrodes (i.e. the electrodes that sampled activity), and the final score represents the change in the spiking activity of all the neurons recorded. For more detail, i.e. to see the electrode combination that each group number represents, we refer the reader to Appendix A. The results show that many electrode subsets produced scores > 1, i.e. the results show that presenting a stimulus repeatedly to the olfactory bulb significantly changed the neural activity across the recorded area.

the same data, we found that on average there were 22 neurons showing significant changes in activity across the different trials (trial 1 against 2, trial 1 against 3 and trial 2 against 3) pre-stimulus. However, only the activity of 16 neurons changed significantly between pre- and during-stimulus (over the three trials). Therefore, the single cell results are in agreement with our MEANOVA results.

6.5 Fluctuations

From our results, it is interesting that the fluctuation of the significance score decreases as $|\mathcal{S}|$ increases. In other words, with a large value for $|\mathcal{S}|$, the obtained results are more reliable. Thus, it is natural to ask whether the observed phenomena in the previous subsections are general, or depend on our specific data. In Appendix C, we show theoretically that the score should stabilise as $|\mathcal{S}|$ increases.

6.6 Discussion

In this chapter, we have presented a novel method of applying MANOVA to electrophysiological data recorded by a multi-electrode array (MEA). By applying MEANOVA to both biological and simulated data, we conclude that this approach is useful for detecting hot spots, i.e. detecting the subareas that were the most responsive to external stimulation, within the recorded area.

There are many advantages of applying MEANOVA rather than MANOVA to MEA data. Firstly, the standard MANOVA method cannot detect hot spots, i.e. the most responsive subareas, within an MEA recorded area. That is, MANOVA is designed to perform one test on all of the neurons as a whole, and show, whether, a stimulus caused a significant change overall in the spiking activity of the population. MEANOVA detects hot spots by producing a significance score, which details the difference in the activity of a cluster, i.e. subsets of neurons, and if compared with the other scores, provides an insight into whether there are different levels of change throughout the area recorded. In addition to this, the score also provides a confidence level that the change in neural activity was caused by one of the factors under investigation and not by chance. Secondly, MEANOVA can detect small

groups of responsive neurons even when they are embedded within a large sample of unresponsive cells. MANOVA would hide such sparse yet highly significant change. We demonstrated this in fig. 6.10 (simulation section (6.3.3) (Test 3 results)).

From the perspective of methodology, we can say that MEANOVA is a valuable extension of MANOVA. As the number of electrodes increases (i.e. the more neurons we use in our analysis) the variation in the significance score becomes stable (i.e. we obtain a more reliable result). The MEA employed in acquiring the biological data presented here was quite small; neuronal activity was recorded from 11 electrodes (we refer the reader to Appendix A). However, even with relatively small MEA, we achieve near stability in the results.

Analysis of simulated data also demonstrated that MEANOVA yields useful results irrespective of the size of the multi-electrode array used. Whilst only 11 electrodes sampled biological data, the simulations have demonstrated that MEANOVA is equally effective with a much larger number of electrodes. Furthermore, we established that the number of observations (time bins) and duration of sampling, are both important factors. For example, the application of MEANOVA to a short period of recording can yield inaccurate information about the true character of the data.

Finally we want to emphasise that although we exclusively discuss our approach for dealing with MEA data, this technique is readily applicable to many other types of data including gene microarrays (Riemer et al., 2004) and brain imaging (Todd et al., 2004) where, again, effects may be sparse across a large sample of loci.

We have shown in this chapter that MEANOVA is a viable solution to our initial aim. That is, it can identify hotspots (i.e. identify the responsive, and most responsive, neuronal subareas) within the MEA recorded area. MEANOVA will, therefore, play a vital role in determining and showing how a stimulus affects the spiking activity (spiking rates) of neurons within areas of the brain.

Chapter 7

The Effect Of Noise Correlation In The Olfactory Bulb

7.1 Introduction

To determine the true capacity of a population rate code (i.e. how much information a population of neurons can convey about a stimulus in their firing rates[1]), we need to determine the effects of noise correlation[2]. Many studies have shown that noise correlation in the brain is predominately positive, where the average noise correlation coefficient (which is typically calculated using the equation discussed in section 7.2.3) is ∼0.2 (for examples see: van Kan et al.,1985; Mastronarde, 1983; Zohary et al. 1994; Gawne et al., 1993; Gawne et al. 1996; Lee et al. 1998; Bair et al. 2001). Thus, it has been hypothesised that noise correlation between neurons can affect the encoding accuracy of a population rate code, i.e. noise correlation may increase or decrease the information a set of neurons convey in their firing rates. Previous empirical studies have showed that noise correlation has a negligible effect. However, theoretical studies have showed that noise correlation can have a much larger effect on the accuracy of large population codes (> 50) and limit the information the population conveys. Thus, the question as to whether noise

[1] In this chapter, we refer to the firing rate of a cell as the number of spikes fired within a particular time window.

[2] In this chapter we define noise as the trial to trial variability in a neuron's response, i.e. firing rate, to the repeated presentation of a single stimulus, and therefore, noise correlation is the correlation in the variability of the responses of 2 neurons to a stimulus.

correlation effects encoding accuracy in the brain remains.

In this chapter we address, empirically, the question: can noise correlation significantly affect the amount of information a large population of neurons convey in their spiking rates about a set of external stimuli. To address this, we used the MEA spiking data recorded from the rat olfactory bulb[3], as we found noise was predominately positively correlated across neurons, to investigate whether correlated noise can affect the amount of information the spiking rates of a set of mitral cells convey about the six stimuli presented. We calculated the mutual information (using a decoding algorithm discussed later) between the population response and the stimuli when the responses of the population were correlated and not correlated. Therefore, the difference between the information quantities is the effect noise correlation has.

We found the effect of noise correlation was dependent on the tuning similarity of the neurons, and the number of neurons, pooled. That is, as noise correlation was predominately positive, noise correlation tended to decrease the information a set of neurons convey when they are tuned to similar features (we refer to this as the harmful effect), and, increase the information a set of neurons convey when they are tuned to dissimilar features (we refer to this as the beneficial effect). Furthermore, the magnitude of the beneficial and harmful effect increased with the number of neurons pooled. However, when we pooled all of the neurons with similar and dissimilar tuning, the overall effect of noise correlation limited the information gain. That is, information increased with the number of neurons but eventually saturated when the population size reached $\sim$45 neurons, as the beneficial effects in the population did not fully cancel out the harmful effects. Therefore, the pooled response of all the neurons conveyed $\sim$70% of the maximum information that could be conveyed about all six stimuli. In other words, the population could have conveyed $\sim$100% of the information about the stimuli if the noise was not correlated. We therefore conclude that, even though noise correlation can be weak, i.e. the average noise correlation coefficient across neurons was 0.21, it can have a substantial effect on the amount of information a large population of neurons can convey. We discuss and show all of the results in this chapter.

[3]The experiment was discussed in chapter 3

7.2 Materials And Methods

7.2.1 Data Analysis

For this work, we used the spiking data recorded from a rat's olfactory bulb discussed in chapter 3. The experiment, in brief, involved the experimenters sampling spiking activity from a rat's olfactory bulb using a Multi-electrode array when 6 stimuli at 4 concentrations were presented 3 times for 10secs. In other words, spiking activity was recorded on 72 trials. We extracted 236 neurons from the recordings using the spike detection stage discussed in chapter 3 and the spike sorting algorithm in chapter 5. To calculate the level of noise and signal correlation (section 7.2.3), and the effects of correlated noise in the rat olfactory bulb (section 7.3.2), we used the neurons responses, i.e. the number of spikes fired, within the 10s time window[4] after stimulus onset. In this study we ignore the concentration factor and thus associate to each stimulus the responses evoked by all stimulus concentrations. Therefore, each stimulus is associated to 12 responses (trials). We implement this as we cannot obtain an accurate information estimate about the stimulus concentrations from the responses, as the number of trials per concentration is insufficient. We note the responses from every neuron showed no significant difference when they were presented with different stimulus concentrations (one way ANOVA, $P>0.05$). To determine the effects noise correlation can have, we calculated the information the responses conveyed when noise is correlated, and when it is not, using the mutual information measurement algorithm discussed in the next section. Therefore, any significant difference between the two information quantities would determine whether noise correlation can affect the amount of information a population conveys. We discuss this in more detail in the subsequent sections.

[4]We note the information, the effect of noise correlation and the correlations calculated using the responses extracted from the 10s time window was slightly greater than using the responses extracted from shorter time windows. This is because the responses of the neurons in the 10s time window revealed more information about the stimuli than shorter time windows.

7.2.2 Information Measurement Algorithm

To quantify the effect of noise correlation, we need to evaluate the information populations of neurons convey, when the responses of the neurons are correlated and uncorrelated. The difference between the two quantities is the effect noise correlation has. In this section, we first describe the standard method, which cannot be used here (reason discussed below) and then describe the method we use (Franco et al., 2004).

The direct approach to compute the information the responses, i.e. the firing rates, of a set of neurons conveyed about a set of stimuli, is to apply the Shannon mutual information measure (Cover et al., 1991, Shannon 1948). Mutual information is defined as the reduction of uncertainty (or gained information) about the stimulus obtained by knowing the neuronal response. The mutual information between the stimuli and the responses is measured in logarithms of base 2 (bits). Thus, every bit of information provided by the neurons reduces the overall uncertainty about the stimulus by a factor of two. The algorithm is defined by:

$$I(s, \overrightarrow{r}) = \sum_{s \epsilon S} \sum_{\overrightarrow{r}} P(s, \overrightarrow{r}) \log_2 \frac{P(s, \overrightarrow{r})}{P(s)P(\overrightarrow{r})} \tag{7.1}$$

where $P(s, \overrightarrow{r})$ is a probability table embodying a relationship between the variable s (here, the stimulus) and $\overrightarrow{r}$, which is a vector where each element contained in it is the number of spikes fired by 1 neuron. However, we cannot use this method to obtain an accurate estimate of the information the large number of mitral cells conveyed in our datasets as we have a limited number of responses per stimulus[5]. That is, to produce an accurate probability table ($P(s, \overrightarrow{r})$) containing the relationship between the six stimuli presented and the responses from the 236 neurons recorded, we would need a far greater number of responses (trials) per stimulus. Therefore, for the results in this chapter, we used a mutual information measurement method developed by (Franco et al., (2004)), which produces an information estimate as accurate as the direct approach without the need for large numbers of trials (as shown by Franco et al., (2004)). That is, the method by (Franco et al., (2004)) produces an accurate information estimate when the number of trials for each stimulus, used to

[5]We note the neuronal activity was sampled on a limited number of trials as a recording session cannot, at least with mammals, last for a long period of time.

calculate the information, is at least twice the number of stimuli. The method uses a decoding procedure in which on each trial the probability that each stimulus (called s') was shown is estimated from the vector of neuronal responses. This estimate is made by comparing the vector of neuronal responses on that trial to the average response vectors to each stimulus. Then, knowing the actual stimulus shown on that trial, the mutual information I, measured in bits, between the estimated stimulus s' and the real stimulus s over the set of stimuli S can be calculated as

$$I = \sum_{s \epsilon S} \sum_{s' \epsilon S} P(s, s') \log_2 \frac{P(s, s')}{P(s)P(s')} \tag{7.2}$$

The decoding procedure we used for the results presented in this chapter was Bayesian probability estimate (PE) decoding using a Gaussian fit, as described by Rolls et al.(1997), Rolls and Treves (1998), Rolls and Deco (2002) and Franco et al., (2004). The paper (Franco et al., (2004)) also explains the bias correction procedure we used to correct the information for undersampling.

The mutual information between the responses of a population and a stimulus is also measured by the equation 7.2 in logarithms of base 2 (bits). Therefore, according to equation 7.2 the total information that a set of neuron can convey about S stimuli is $log_2(S)$. With this in mind, the total information our set of neurons could convey about the six stimuli presented was $log_2(6) = 2.56$ bits. If the responses of the neurons conveyed this amount of information, each response showed, with 100% certainty, which stimulus evoked it. We quantify the effect of noise correlation by comparing the information the population conveyed about the stimuli, which we refer to as I, with the information the population conveys when we remove the noise correlation across cells, which we refer to as I_s (i.e. I_s shows the information the population would convey if each neuron contributed independent information). For the former quantity, we calculate I using the population responses with the decoding procedure described above and equation 7.2. To calculate I_s we implement the following:

1. We implement the following $N = 100$ times:

   (a) We remove the noise correlation from the responses by shuffling the trials, i.e. we randomly permute the order of the 12 trials (responses), corresponding to each stimulus independently for each neuron.

(b) We calculate the information using the uncorrelated responses with the decoding procedure and equation 7.2.

2. We then average the information calculated over all N iterations. The average value corresponds to I_s.

We calculate I_s in this manner, because, as we randomly shuffle the trials (responses), discussed in point (a) above, we cannot be certain that the responses are completely uncorrelated, therefore, I_s could be vastly under or overestimated. Thus, averaging the information calculated from many shuffled response sets is more likely to be a better estimate of I_s. For this work we used $N = 100$, however, we note I_s did not significantly increase or decrease when $N > 100$.

The effect of noise correlation, which we refer to as $\triangle I$, corresponds to $I - I_s$. Thus, $\triangle I > 0$ signifies that noise correlation increased the information and $\triangle I < 0$ signifies that noise correlation decreased the information population conveyed.

7.2.3 Measuring Signal And Noise Correlations

Before we investigated the effects of noise correlation in the rat olfactory bulb, we firstly investigated whether noise in this neural system was correlated. We used the equation below to calculate the noise correlation and discuss the results later. In this work, the noise correlation between a pair of neurons was defined as the correlation in the neuronal response variability (trial to trial) from the average to a stimulus. It is derived by (and shown in (Averbeck et al., 2004)):

$$NC_{12}(s) = \frac{\langle r_{1sn} r_{2sn} \rangle_N - m_{1s} m_{2s}}{\sqrt{\langle (r_{1sn} - m_{1s})^2 \rangle_N \langle (r_{2sn} - m_{2s})^2 \rangle_N}} \tag{7.3}$$

where $\langle ..... \rangle_N$ is the average response across trials in which only stimulus s was presented, r_{isn} is the response of neuron i to stimulus s on trial n, and, m_{is} is the mean response of neuron i to stimulus s.

As noise correlation is thought to be stimulus independent we compared the strength of the noise correlation across all stimuli for each pair. The number of pairs recorded was 27730. We found the correlation between the neurons in every pair were not significantly different across stimuli (ANOVA One-Way Test, P >

0.05) . We therefore, in this chapter, report the average noise correlation across the stimuli for each pair.

It has been suggested that signal correlation, i.e. the correlation in the mean responses of the neurons across the set of stimuli, may affect the impact noise correlation can have on the information a population of neurons conveys (Averbeck et al., 2006). We, therefore, calculated the signal correlation for each pair of neurons to determine whether a difference in signal correlation, i.e. if signal correlation is positive or negative, corresponds to a different noise correlation effect. The signal correlation between a pair of neurons is derived by (and shown in (Averbeck et al., 2004)):

$$SC_{12} = \frac{\langle (m_{1s} - \langle m_{1s} \rangle_s)(m_{2s} - \langle m_{2s} \rangle_s) \rangle_s}{\sqrt{\langle (m_{1s} - \langle m_{1s} \rangle_s)^2 \rangle_s \langle (m_{2s} - \langle m_{2s} \rangle_s)^2 \rangle_s}} \tag{7.4}$$

where $\langle ..... \rangle_s$ indicates the average over the s stimuli, and, m_{is} is the mean response rate of cell i (where i=1,2) to stimulus s over all the trials in which that stimulus was present.

We note that if we compared distributions of correlation coefficients for any statistical analysis, we firstly applied Fishers z transformation to obtain normal distributions.

7.3 Results

The results in (Friedrich et al., 2001) showed the amount of information single mitral cells convey about the odour stimuli presented was small. Therefore, as we expected information about an odour stimulus to improve when the number of neurons pooled increased, we investigated whether any information gain was dependent on the noise correlation between the neurons recorded. To address this, we began our analysis by quantifying the noise and signal correlation between the neurons, we then, investigated whether there was a relationship between signal and noise correlation.

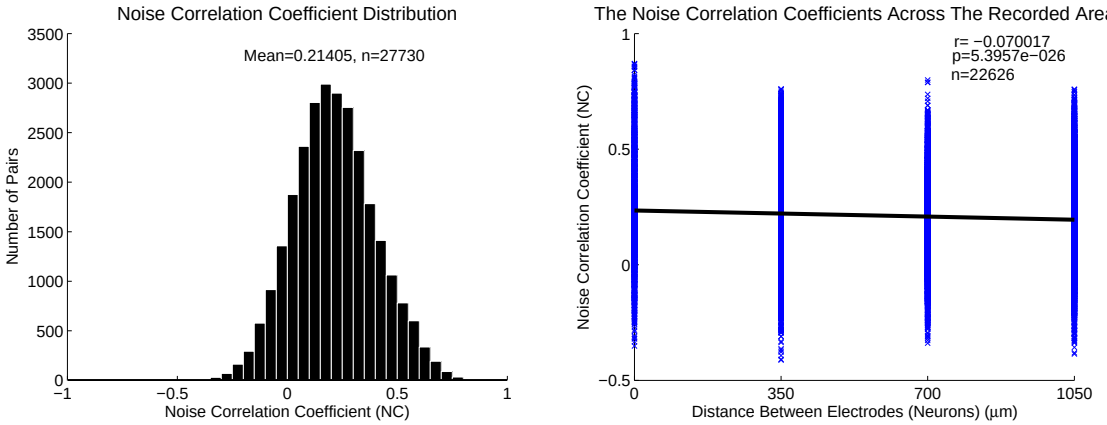
7.3.1 Noise Correlation

We firstly calculated the noise correlation coefficient (NC) for each of the 27730 pairs recorded. We show all of the noise correlation coefficients, i.e. the distribution, in fig. 7.1 Panel (a). From this, we found the average noise correlation coefficient was 0.2140, and, was significantly different from 0 (Wilcoxon Signed Rank Test; $p<0.05$). We next investigated whether the strength of noise correlation was correlated with the distance between neurons. To address this, we firstly divided the pairs into different groups according to the distance between the neurons in each pair. The distance between the cells was measured as the distance between the electrodes that recorded them. We acquired 4 groups. Group 1 contained 2342 pairs where the neurons in each pair were recorded by the same electrode, group 2 contained 7928 pairs where the distance between the neurons in each pair was 350μm, group 3 contained 7494 pairs where the distance between the neurons in each pair was 700μm and group 4 contained 9966 pairs where the distance between the neurons in each pair was 1050μm[6]. We then calculated the Pearson's correlation coefficient (r) between the noise correlation and the groups of pairs (distance). We also calculated a p value for r using a Student's t distribution for a transformation of r to test the hypothesis that $r = 0$ against the alternative that $r \neq 0$. We found a small (r=-0.0700), but statistically significant ($p<0.05$) relationship between the two, i.e. we found a small gradual decrease in noise correlation from 0μm (same electrode) to 1050μm (Fig. 7.1 Panel (b)).

Relationship Between Signal Correlation And Noise Correlation

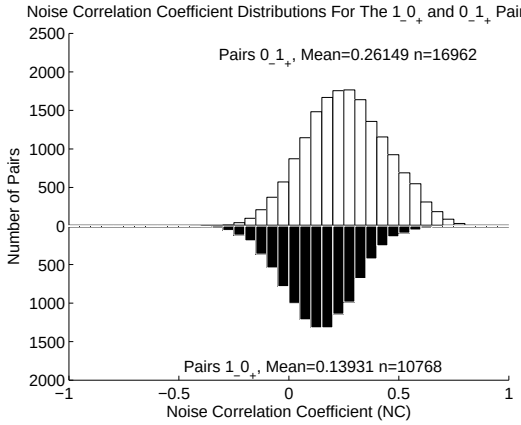
We calculated the signal correlation between every pair of neurons and found that neurons in a pair were either both tuned to similar features, i.e. the signal correlation coefficient (SC) was > 0, or, they were tuned to different features, i.e. the signal correlation coefficient was <0. We therefore investigated whether there was a correlation between noise correlation and signal correlation. Before we discuss the results, we introduce some notation. Throughout the chapter, we refer to a

[6]We note that pairs of neurons recorded from electrodes further apart were not considered due to insufficient numbers of them.



(a)

(b)



(c)

Figure 7.1: **Panel (a):** shows the noise correlation coefficient distribution for all pairs of neurons, where the mean is 0.2140, and, was significantly different from 0 (Wilcoxon Signed Rank Test; $p<0.05$). **Panel (b):** shows the relationship between the noise correlation coefficients and the distance between neurons. The results show a small (r=-0.0700), but statistically significant ($p<0.05$) relationship between the two, i.e. r slightly decreased with distance. **Panel (c):** shows the noise correlation coefficient distributions for the pairs with a signal correlation coefficient > 0 (0_1_+ pairs, white bars) and for the pairs with a signal correlation coefficient <0 (1_0_+ pairs, black bars). The results show the distributions slightly overlap but are significantly different (Wilcoxon Rank Sum Test; $p<0.05$). Furthermore, the noise correlation coefficients for the 1_0_+ pairs tended to be smaller (mean=0.13931) than the noise correlation coefficients for the 0_1_+ pairs (mean=0.26149).

group of neurons using A_B_+. A refers to the number of pairs in the group tuned to dissimilar features, i.e. have a signal correlation coefficient < 0, and B refers to the number of pairs in the group tuned to similar features, i.e. have a signal correlation coefficient > 0. We have also included a $-$ and $+$ in the notation, which refer to negative and positive signal correlation respectively, so that the notation clearly shows that A and B correspond to the number of pairs in a group that have a positive and negative signal correlation. So, for example, a pair of neurons that has a signal correlation coefficient > 0, we would refer to as 0_1_+, and a pair that has a signal correlation coefficient < 0, we would refer to as 1_0_+. Also, for example, we would refer to a group of 4 neurons as 2_4_+, if the group contained 4 pairs that have a signal correlation coefficient > 0 and 2 pairs that have a signal correlation coefficient < 0. We note in this example only 6 pair combinations can be acquired from a 4 group combination. To investigate whether the strength of noise correlation and signal correlation was correlated, we firstly calculated the noise correlation coefficients for the 0_1_+ and 1_0_+ pairs. We show the results of both, i.e. distributions, in fig. 7.1 Panel (c), where the white and black bars represent the noise correlation coefficient distributions for the 0_1_+ and 1_0_+ pairs respectively. The results show the two distributions slightly overlap but are significantly different (Wilcoxon Rank Sum Test $p<0.05$). Furthermore, the noise correlation coefficients for the 1_0_+ pairs tended to be smaller (mean=0.13931) than the noise correlation coefficients for the 0_1_+ pairs (mean=0.26149). In addition there was a weak but statistically significant correlation between the signal and noise correlation coefficients ($r=0.35878$; $p<0.05$) (where we calculated r and p using the same method described in the previous section). We next investigated whether the relationship between the signal and noise correlation coefficients were distance dependent. We split the pairs into four groups as described in the previous section and calculated r and p for each group. We show the signal and noise correlation coefficients for the pairs in each group in fig. 7.2 Panels (a)-(d), and, we show r as a function of the distance between neurons in fig. 7.2 Panel (e). We found the highest correlation between the signal and noise correlation coefficients was for pairs comprised of neurons recorded through the same electrodes (fig. 7.2 Panel (a)), where, the relationship (r) gradually decreased as the distance between electrodes increased

(fig. 7.2 Panel (e)). We note, it is conceivable that the pairs of neurons recorded by the same electrode have a higher correlation between signal and noise correlation because of some uncertainty in the spike sorting. That is, as it is difficult to assign all the spikes unequivocally to their proper sources when two neurons produce temporally overlapping spikes recorded by the same electrode, it is conceivable that neurons with similar tuning gave rise to artificially high noise correlation. However, if there was such a random misassignment of spikes, we would have expected the noise correlation between these neurons to have been lower.

7.3.2 Effects of Noise Correlation On The Amount Of Information Pairs Of Neurons Convey

We firstly investigated whether noise correlation could influence the amount of information pairs of neurons convey about the six stimuli. We note that the maximum amount of information the neuronal responses could convey about the six stimuli was $log_2(6) = 2.56$ bits. In this section, and in subsequent sections, we refer to the average of I, I_s (I and I_s refer to the information conveyed by the neurons when the responses are correlated and uncorrelated respectively), and $\triangle I$ ($\triangle I$ refers to the effect of noise correlation, i.e. $I - I_s$) across different groups of pairs as $\bar{I}$, $\bar{I}_s$ and $\bar{\triangle} I$ respectively. Below, we discuss the results found when we analysed the responses of pairs of neurons.

There were 10768 1_0+ pairs, which conveyed $\bar{I}$=0.3347 bits of information. When noise correlation was absent, these pairs on average conveyed $\bar{I}_s$=0.3246 bits of information. Therefore, $\bar{\triangle} I$ was 0.01, and, was significantly larger than 0 (Wilcoxon signed-rank test, $p<0.05$). We show the distribution of $\triangle I$ for these pairs in fig. 7.3 (Panel (a), dotted black line).

In addition, there were 16962 0_1_+ pairs which conveyed, $\bar{I}$ =0.3090 bits. When noise correlation was absent, these pairs on average conveyed $\bar{I}_s$=0.3196 bits. Therefore, $\bar{\triangle} I$ was -0.0106, and, was significantly smaller than 0 (Wilcoxon signed-rank test, $p<0.05$). We show the distribution of $\triangle I$ for these pairs in fig. 7.3 (Panel (a), solid black line).

The results from the 0_1_+ and 1_0_+ pairs show that as noise correlation be-

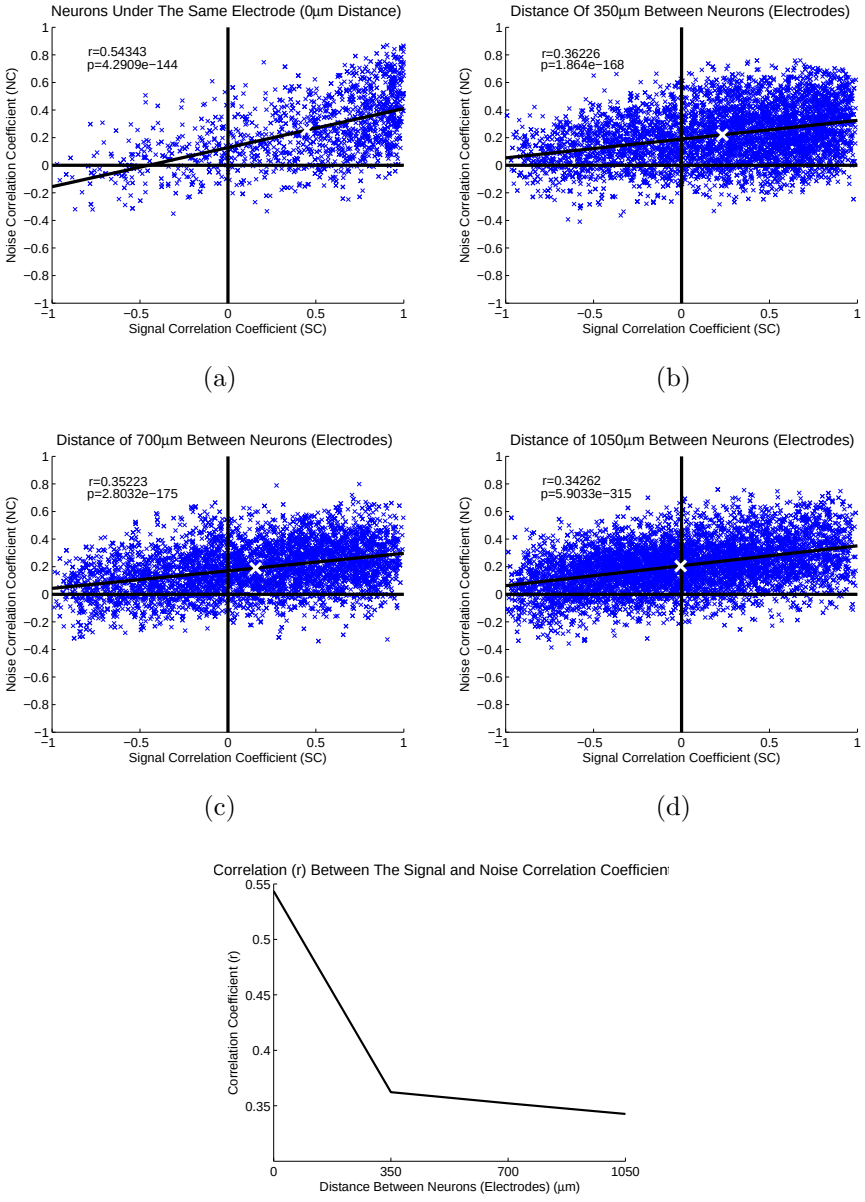


Figure 7.2: This figure shows the relationship between distance and the signal and noise correlation coefficients. **Panels (a)-(d):** show the signal and noise correlation coefficients for 4 groups of pairs. The distance between the neurons in each pair in **Panels (a)-(d)** was 0μm (same electrode), 350μm, 700μm, and 1050μm respectively. The large crosses in each panel represent the mean signal and noise correlation. The solid line in each panel was determined by a linear regression and shows the relationship between the signal and noise correlation coefficients. The lines in the panels show the correlation between the signal and noise correlation coefficients was positive, i.e. ($r > 0$), regardless of the distance between the neurons in the pairs. However, the r value was larger for pairs recorded by the same electrodes and decreased when the distance between the neurons in the pairs increased (**Panel (e)**). We note that the r values calculated for each group were significantly different from 0 ($p<0.05$).

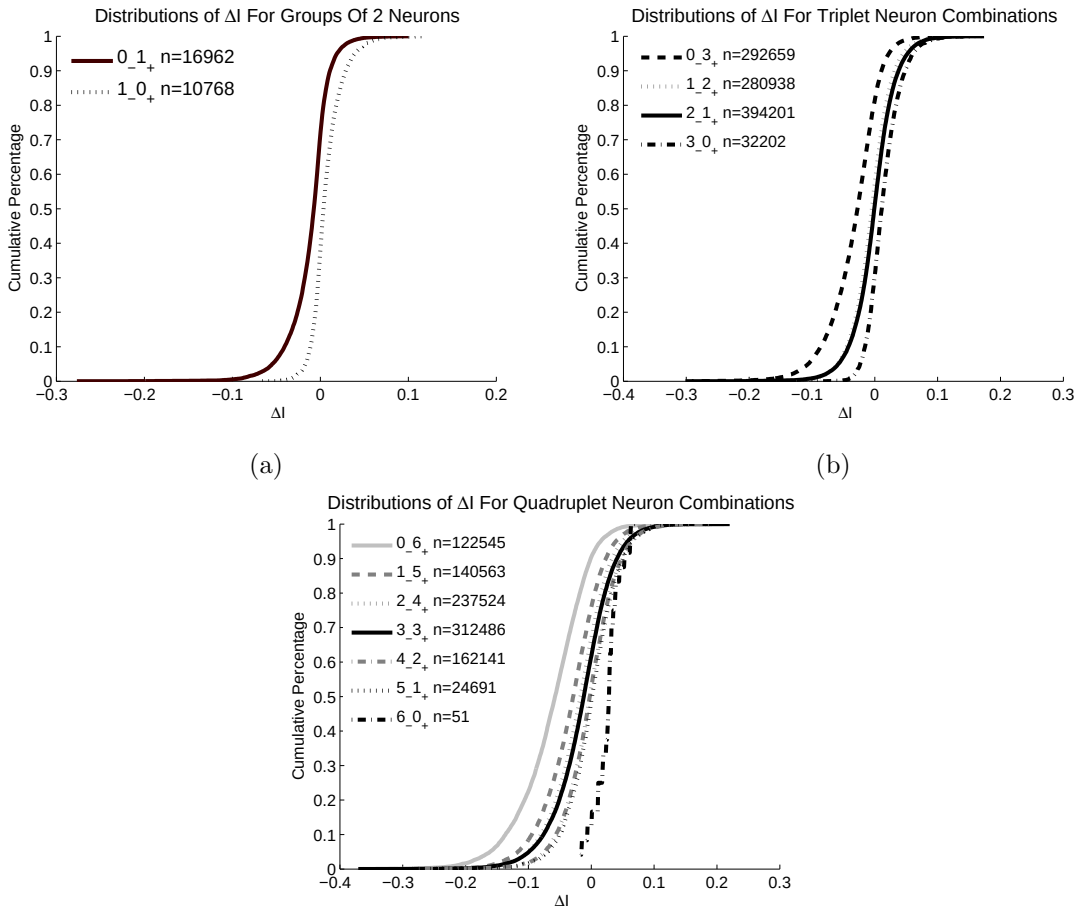


Figure 7.3: Distributions of $\triangle I$; the difference between I and I_s. **Panel (a):** shows the distribution of $\triangle I$ for pairs of neurons tuned to dissimilar features (1_0_+ pairs, black dotted line) and pairs of neurons tuned to similar features (0_1_+ pairs, solid black line). **Panel (b):** shows the distributions of $\triangle I$ for 0_3+ triplets (black dash line), 1_2+ triplets (grey dotted line), 2_1+ triplets (black solid line) and 3_0+ triplets (black dash dot line). **Panel (c):** shows the distribution of $\triangle I$ for 0_6_+ quadruplets (grey solid line), 1_5_+ quadruplets (grey dash line), 2_4_+ quadruplets (grey dot line), 3_3_+ quadruplets (black solid line), 4_2_+ quadruplets (grey dash dot line), 5_1_+ quadruplets (black dot line) and 6_0_+ quadruplets (black dash dot line). The results show that, in general, noise correlation had a greater enhancement and reduction effect on the information conveyed by a set of neurons all tuned to dissimilar (e.g. 6_0_+) and similar (e.g. 0_6_+) features respectively.

tween neurons was predominately positive[7], noise correlation tended to increase the amount of information pairs of neurons convey when they had a negative signal correlation and decrease it when pairs had a positive signal correlation.

In fig. 7.4 we illustrate (using the response distributions, uncorrelated (shuffled) (shuffling procedure is described in section 7.2.2) and correlated (unshuffled), of two example pairs of neurons that responded to two different stimuli) why positive noise correlation increases the amount of information a pair of neurons convey when they are tuned to dissimilar features (signal correlation < 0) but decrease it when they are tuned to similar features (signal correlation > 0).

Thus, as noise correlation was predominately positive between neurons, noise correlation had a similar effect on the information larger populations convey. That is, noise correlation tended to increase the amount of information a population conveyed when all of the neurons were tuned to dissimilar features and decrease the amount of information a population conveyed when all of the neurons were tuned to similar features. We discuss the results found in the subsequent sections.

7.3.3 Effects Of Noise Correlation On The Amount Of Information More Than 2 Neurons Convey

We continued the previous investigation by investigating the effects of noise correlation on the amount of information different neuron combinations, i.e. combinations containing different numbers of neurons tuned to similar and dissimilar tuning, of size 3 to 236 convey. We firstly calculated both I and I_s for 1000000 neuron combinations of each group size. We were limited to calculating 1000000 combinations due to insufficient computer resources, therefore, the combinations used were randomly chosen from all possible combinations. We then divided the I and I_s values into different groups according to the number of pairs in the combinations that had positive and negative signal correlation. We then calculated $\bar{I}$, $\bar{I}_s$ and $\bar{\triangle I}$ for each

[7]We note that some pairs exhibited negative noise correlation, which increased the amount of information a pair of neurons conveyed when they had a positive signal correlation between them and decreased the amount of information a pair of neurons conveyed when they had a negative signal correlation. However, the effect found was very small and not significant (Wilcoxon signed-rank test, $p>0.05$).

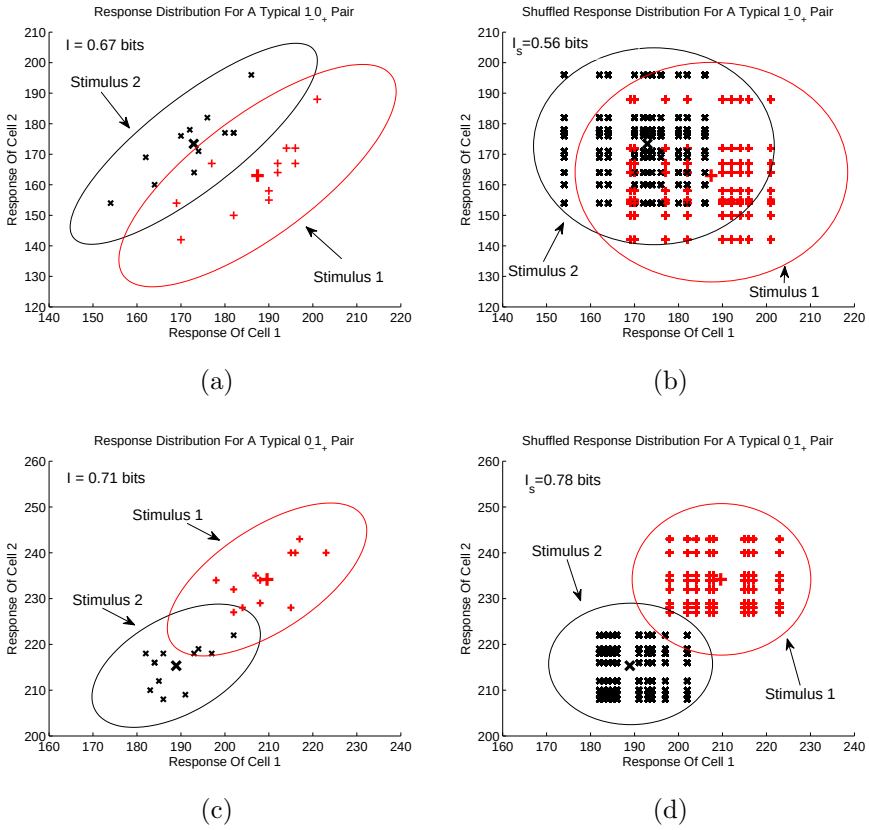


Figure 7.4: This figure shows the response distributions, uncorrelated (shuffled) and correlated (unshuffled), of two example pairs of neurons that responded to two different stimuli. **Panel (a)** shows the correlated responses of an example pair of neurons tuned to dissimilar features (1_0+) and **panel (b)** shows the uncorrelated responses of the pair, i.e. when we shuffled (decorrelated) the responses. **Panel (c)** shows the correlated responses of an example pair of neurons tuned to similar features (0_1+) and **panel (d)** shows the uncorrelated responses of the pair. The 'x' markers represent the responses recorded on trials when stimulus 1 (Amyl acetate) was presented and the '+' markers represent the responses recorded on trials when stimulus 2 (M-Butanol) was presented. We note, in panels (b) and (d), more responses are shown compared to the responses in panels (a) and (c). This is because we show all of the shuffled responses from the 100 shuffled sets we used to calculate I_s. We discussed how we calculated I_s in the information measurement algorithm section. The large markers in all panels represent the mean response of both neurons. **Panels (c)** and **(d)** show the signal correlation is positive, i.e. the average response of both neurons to both stimuli is positively correlated, and, **panels (a)** and **(b)** show the signal correlation is negative. **Panels (a)** and **(c)** show the trial-to-trial variability in the responses of both neurons to both stimuli is positively correlated. That is, the average noise correlation coefficient corresponding to the responses in panels **(a)** and **(c)** is 0.66 and 0.59 respectively. Each ellipse (which appears as a circle in the uncorrelated, i.e. shuffled, plots) represents a 95% confidence interval for the responses. Thus, the overlap between the ellipses represent the level of uncertainty as to which stimulus evoked the responses, i.e. the smaller the overlap the more information the responses convey. The results in **Panels (a) and (b)** show when noise correlation is positive and signal correlation is negative, noise correlation increased the information the pair of neurons conveyed, i.e. $I = 0.67$, $I_s = 0.57$ bits and therefore $\triangle I > 0$. This is because overlap between the response distributions increased when we removed noise correlation, i.e. removing noise correlation increased the uncertainty as to which stimulus evoked the responses. However, **Panels (c) and (d)** show that when signal and noise correlation is positive, noise correlation decreased the information the pair of neurons conveyed, i.e. $I = 0.71$, $I_s = 0.78$ bits and therefore $\triangle I < 0$. This is because overlap between the response distributions decreased when we removed noise correlation, i.e. removing noise correlation decreased the uncertainty as to which stimulus evoked the responses.

subgroup. We refer to each subgroup using the same notation as discussed earlier. For example, if we had a set of triplet combinations containing 1 pair with a signal correlation > 0 and 2 pairs with a signal correlation < 0 (we note that 3 is the total number of pairs we can acquire from a triplet combination) we would associate this set to subgroup 2_1_+. We analysed the information conveyed by different subgroups so we could determine whether the effects of noise correlation found at the pair level were the same in larger groups. That is, whether noise correlation can decrease the amount of information large groups of neurons convey when they are tuned to similar features, and, increase the information large groups of neurons convey when they are tuned to dissimilar features. We firstly discuss the results acquired when we analysed the amount of information triplets of neurons can convey.

Triplets

Out of the 1000000 combinations used, 292659 were 0_3_+ triplets (that is, all pairs in this subgroup had a signal correlation >0), 280938 were 1_2_+ triplets, 394201 were 2_1_+ triplets and 32202 were 3_0_+ triplets (that is, all pairs in this subgroup had a signal correlation <0). For the 0_3_+ triplets, $\bar{I} = 0.4427$ and $\bar{\triangle I} = -0.0328$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 1_2_+ triplets, $\bar{I} = 0.4756$ and $\bar{\triangle I} = -0.0048$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 2_1_+ triplets, $\bar{I} = 0.4859$ and $\bar{\triangle I} = -0.0010$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 3_0_+ triplets, $\bar{I} = 0.4922$ and $\bar{\triangle I} = 0.0154$, where $\bar{\triangle I}$ was significantly >0 (Wilcoxon signed rank test, p<0.05). In fig. 7.3 Panel (b) we show the $\triangle I$ distributions for the 0_3_+, 1_2_+, 2_1_+ and 3_0_+ triplets as a black dash line, grey dotted line, black solid line and a black dash dot line respectively.

Quadruplets

Out of the 1000000 combinations used, 122545 were 0_6_+ quadruplets (that is, all pairs in this subgroup had a signal correlation >0), 140563 were 1_5_+ quadruplets, 237524 were 2_4_+ quadruplets, 312486 were 3_3_+ quadruplets, 162141 were 4_2_+

quadruplets, 24691 were 5_1_+ quadruplets and 51 were 6_0_+ quadruplets (that is, all pairs in this subgroup had a signal correlation <0). For the 0_6_+ quadruplets, $\bar{I} = 0.5608$ and $\bar{\triangle I} = -0.0623$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 1_5_+ quadruplets, $\bar{I} = 0.5975$ and $\bar{\triangle I} = -0.0316$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 2_4_+ quadruplets, $\bar{I} = 0.6200$ and $\bar{\triangle I} = -0.0202$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 3_3_+ quadruplets, $\bar{I} = 0.6207$ and $\bar{\triangle I} = -0.0140$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 4_2_+ quadruplets, $\bar{I} = 0.6333$ and $\bar{\triangle I} = -0.0035$, where $\bar{\triangle I}$ was significantly <0 (Wilcoxon signed rank test, p<0.05). For the 5_1_+ quadruplets, $\bar{I} = 0.6445$ and $\bar{\triangle I} = 0.0010$, where $\bar{\triangle I}$ was significantly >0 (Wilcoxon signed rank test, p<0.05). For the 6_0_+ quadruplets, $\bar{I} = 0.6543$ and $\bar{\triangle I} = 0.0234$, where $\bar{\triangle I}$ was significantly >0 (Wilcoxon signed rank test, p<0.05). In fig. 7.3 Panel (c) we show the $\triangle I$ distributions for the 0_6_+, 1_5_+, 2_4_+, 3_3_+, 4_2_+, 5_1_+ and 6_0_+ quadruplets as a grey solid line, grey dash line, grey dot line, black solid line, grey dash dot line, black dot line and a black dash dot line respectively. We discuss the quadruplet, the triplet and the pair results in the next section.

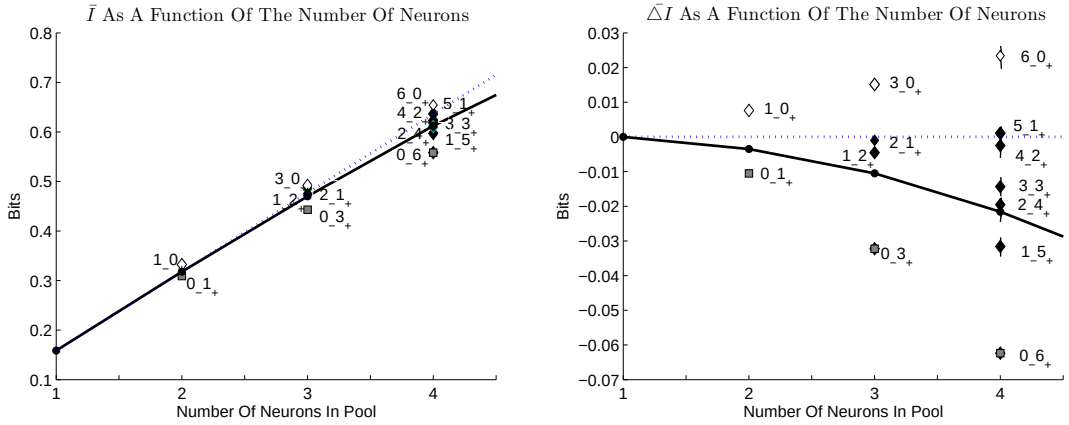
7.3.4 Summary Of Analysis So Far

We summarise the results in fig. 7.5 for the pair, triplet and quadruplet information analyses and discuss the results so far. The figure shows the information quantities $\bar{I}$ (Panel (a)) and $\bar{\triangle I}$ (Panel (b)) calculated from each of the neuron subgroups discussed previously. The dotted line in Panel (a) shows $\bar{I}_s$, i.e. the average information conveyed by the shuffled responses across all subgroup combinations. In other words, $\bar{I}_s$ shows the average information the neurons could have conveyed if they fired independently. The results in panel (a) shows that $\bar{I}$ for the neuron groups containing all pairs with a signal correlation > 0 (white diamonds) were above the dotted line. That is, $\bar{\triangle I}$ for these subgroups was > 0 (Panel (b), white diamonds). These results, therefore, show noise correlation increased the amount of information these subgroups could convey. Furthermore, the results in panel (a) shows $\bar{I}$ for the subgroups containing all pairs with a signal correlation < 0 (grey squares), and, subgroups that had a mixture of pairs with a signal correlation > 0 and < 0

(black diamonds), were below the dotted line. That is, $\bar{\triangle}I$ for these subgroups was < 0 (panel (b), black diamonds and grey squares). These results, therefore, show noise correlation decreased the amount of information these subgroups could convey. When we averaged across all subgroups the average information (the black line in panel (a)) was below the dotted line. That is, the average effect of noise correlation (the black line in panel (b)), was < 0. Therefore, the results show that, overall, noise correlation decreased the amount of information the neuron ensembles could convey. However, the magnitude of the overall effect was less compared to if all the neurons were tuned to similar features. That is, the overall effect of noise correlation would be more harmful in the population if all the neurons were tuned to similar features. For example in the quadruplet results, $\bar{\triangle}I$ was -0.062 for the 0_6_+ combinations, i.e. when all the neurons pooled were tuned to similar features, but, when we averaged the effect across all quadruplet combinations, i.e. when the neurons pooled had similar and dissimilar tuning, $\bar{\triangle}I$ was -0.02 (panel (b)). This shows that pooling neurons tuned to dissimilar and similar features would improve the accuracy of the information a population conveys when noise correlation is present, as the beneficial effects would partially reduce the harmful effects. We also note that panels (a) and (b) show the overall harmful effect increased with the number of neurons. In summary, the results taken together suggest that the impact of noise correlation in the olfactory bulb is dependent on the tuning similarity of the neurons, and the number of neurons, pooled. In the next section we show the effects of noise correlation on larger populations.

7.3.5 Summary Of Results For Larger Populations

We can see from the results in fig. 7.5 panel (b) that the overall effect (black line) of noise correlation was harmful and was dependent on the tuning similarity of the neurons pooled. This was consistent in larger populations. We therefore, for clarity due to the number of subgroups we can acquire increasing with group size, summarise the results acquired from the larger populations in fig. 7.6. Panels (a) and (b) show $\bar{I}$, $\bar{I}_s$ and $\bar{\triangle}I$ as a function of the number of neurons, where panel (a) shows $\bar{I}$ and $\bar{I}_s$ and panel (b) shows $\bar{\triangle}I$. In this section $\bar{I}$, $\bar{I}_s$and $\bar{\triangle}I$ refer to the average I, I_s and I_s, respectively, across all subgroup combinations. We found $\bar{I}$ (fig. 7.6

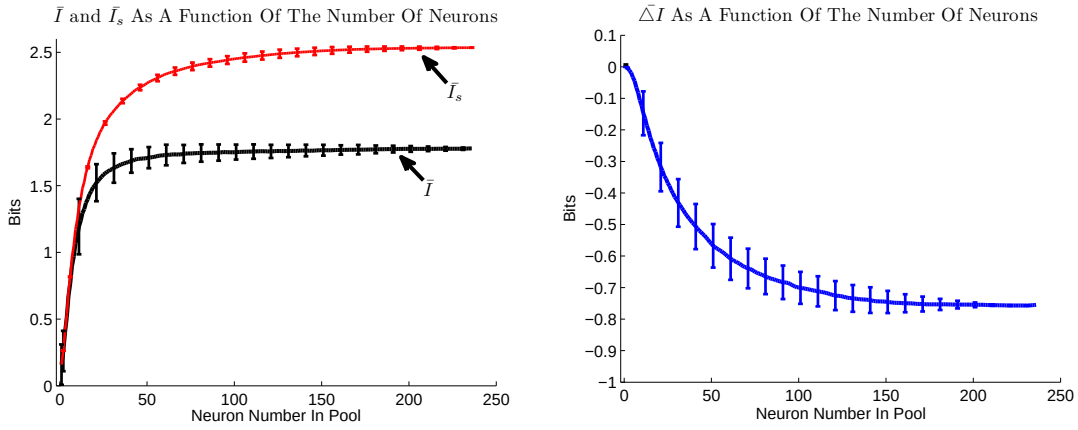


(a) (b)

Figure 7.5: **Panel (a)** shows the $\bar{I}$ values, and **Panel (b)** shows the $\bar{\triangle I}$ values, for the pair, triplet and quadruplet combinations. The panels show the $\bar{I}$ and $\bar{\triangle I}$ values for the subgroups containing all pairs with a signal correlation > 0 (grey squares), all pairs with a signal correlation < 0 (white diamonds), and a mixture of pairs with a signal correlation < 0 and > 0 (black diamonds). The dotted line and black line in panel (a) represents the average $\bar{I}_s$ and $\bar{I}$ quantities across all subgroups. The black line in panel (b) represents the average $\bar{\triangle I}$ across all subgroups. Error bars show the standard error of the mean of the subgroups.

panel (a)) increased with the number of neurons and peaked when the ensemble size was ∼45. For example, $\bar{I}$ was 0.2606±0.156 (±S.D), 1.1213±0.208, 1.7078±0.0795, 1.7511±0.056, 1.7661±0.043 for group sizes 2, 10, 50, 100, 150, 236 respectively. We found $\bar{\triangle I}$ (fig. 7.6 Panel (b)) was relatively small on the information encoded on ensembles (<22), but the effect increased with the ensemble size thereafter. For example, $\bar{\triangle I}$ was -0.0038±0.0326,-0.1227±0.069, -0.5607±0.0691, -0.6995±0.113, -0.7448±0.0725, -0.7549 bits for group sizes 2, 10, 50, 100, 150, 236 respectively. We note that the average effect of noise correlation across all group sizes was significantly different from 0 (Wilcoxon signed rank test, p<0.05). These results indicate that correlated noise caused information to saturate as the number of neurons reached ∼45 neurons. Thus, the population conveyed ∼70% (i.e. $\frac{1.77}{2.56}.100$)[8] of the maximum information that could be conveyed about the six stimuli. However, the population would have conveyed nearly all the information about all six stimuli (2.56 bits) if noise was not correlated, i.e. I_s was 2.52 bits (shown in fig. 7.6 Panel (a)).

[8]We note the value 1.77 was the total amount of information the population conveyed about all six stimuli (shown in fig. 7.6 Panel (a)). Furthermore, the value 2.56 was the maximum amount of information that could be conveyed about all six stimuli, i.e. $log_2(6)$.



(a) (b)

Figure 7.6: **Panels (a)** and **(b)** show $\bar{I}$, $\bar{I}_s$ and $\bar{\triangle}I$ as a function of the number of neurons, where **Panel (a)** shows $\bar{I}$ and $\bar{I}_s$, and **Panel (b)** shows $\bar{\triangle}I$. In these results, $\bar{I}$, $\bar{I}_s$and $\bar{\triangle}I$ refer to the average I, I_s and $\triangle I$, respectively, across all combinations. The error bars in both panels represent the standard deviation. **Panel (a)** shows that information ($\bar{I}$) increased with the number of neurons but saturated when the number of neurons reached ~ 45, conveying, 1.77 bits of information. When we decorrelated the noise, the information (I_s) the population conveyed was 2.52 bits (**Panel (a)**). That is, noise correlation decreased the amount of information the population conveyed by 0.7549 bits (**Panel (b)**).

7.4 Discussion

The aim of our study was to determine whether noise correlation can influence the accuracy of a population rate code. We addressed this by calculating the amount of information the responses of 236 mitral cells conveyed about all six stimuli, when, the response variability (noise) of the neurons was correlated and uncorrelated. To quantify the amount of information the neurons conveyed, we used a decoding method, which measured the information in terms of bits. The results showed that the impact of noise correlation was dependent on the neurons pooled. That is, as noise correlation was predominately positive, noise correlation tended to reduce the amount of information pairs of neurons conveyed when they were tuned to similar features and increased the information pairs of neurons conveyed when they were tuned to dissimilar features. However, when we pooled all neurons, the opposing effects limited the information gain of the population, as the beneficial effects did not fully cancel out the harmful effects. That is, as the overall reduction effect of noise correlation increased with the number of neurons, information saturated when the number of

neurons pooled reached ∼45. Thus, the population conveyed ∼ 70% (1.77 bits, shown in the previous section) of the maximum information that could be conveyed about the six stimuli (i.e. $log_2(6) = 2.56$). However, the population would have conveyed ∼ 100% of the information (as shown in the previous section) if the noise was not correlated. We, therefore, conclude that noise correlation can have an effect on encoding accuracy in the brain, where the effect is dependent on the tuning similarity of the neurons, and the number of neurons, pooled.

7.4.1 Noise Correlation

We found the average noise correlation coefficient between neurons in the olfactory bulb was 0.21. This value is consistent with other studies of noise correlation in other brain areas. For example, the average noise correlation coefficient found in the:

- V1 was 0.20 (Kohn et al., 2005), 0.25 (Reich et al., 2001) and 0.22 (Gawne et al., 1996), in the middle temporal visual area was 0.19 (Zohary et al.,1994).
- inferior temporal cortex was 0.23 (Gawne et al., 1993).
- supplementary motor area was 0.13 (Averbeck et al., 2006) .
- parietal areas of the primate cortex was 0.16 (Lee et al., 1998).
- secondary somatosensory cortex was 0.12 (Romo et al., 2003)

Our other noise correlation investigations are also comparable with the work of (Lee et al., 1998). That is, we, and (Lee et al., 1998), found a weak relationship between noise correlation and distance, where, noise correlation showed a small gradual decrease with distance. We, and (Lee et al.,1998), found the noise correlation coefficient tended to be greater for pairs of neurons that had a signal correlation >0 compared to pairs of neurons with a signal correlation <0. Furthermore, the correlation between signal and noise correlation was stronger for pairs of neurons in close proximity compared to pairs of neurons further apart. Thus, as the levels of noise correlation were similar in both the rat olfactory bulb and other areas of the brain, the effects of noise correlation reported in this work may also occur in other parts of the brain. This could be the basis of future work.

7.4.2 Influence Of Noise Correlation On The Population Code

Many studies have examined the impact positive noise correlation can have on the amount of information a population rate code conveys (Averbeck et al., 2006). The first theoretical study investigated the effects of positive noise correlation on neurons with similar tuning functions (Zohary et al., 1994). They showed that positive noise correlation slightly decreased the information pairs of neurons conveyed, but, greatly decreased, i.e. limited, the information conveyed by large populations of neurons. Other theoretical studies have also shown the effect of noise correlation can increase with the number of neurons (Oram et al., 1998; Panzeri et al., 1999; Abbot et al., 1999; Averbeck et al., 2004; Averbeck et al., 2006). However, they also showed that positive noise correlation does not necessarily decrease, i.e. limit, information, as the effect of noise correlation is dependent on the tuning similarity of the neurons pooled (Oram et al., 1998; Panzeri et al., 1999; Abbot et al., 1999; Averbeck et al., 2004; Averbeck et al., 2006). In other words, positive noise correlation is harmful on populations with similar tuning, but, beneficial on populations with dissimilar tuning. This was also found in some empirical studies (Panzeri et al., 2001; Petersen et al., 2001; Averbeck et al., 2003; Golledge et al., 2003; Romo et al., 2003), where they studied the effects of noise correlation on the information pairs of neurons convey. Overall, the theoretical and empirical work has suggested that the effect of noise correlation is dependent on the tuning similarity of the neurons, and the number of neurons, pooled. Our work also suggests this. We found as noise correlation was predominately positive, noise correlation tended to decrease the amount of information pairs of neurons convey when they were tuned to similar features, i.e. when signal correlation >0, and increase the information pairs of neurons convey when they were tuned to different features, i.e. when signal correlation <0[9]. Furthermore, when we averaged the effect of noise correlation across pairs, the overall effect was harmful, i.e. $\bar{\triangle I} < 0$. These results were also shown in the triplet and quadruplet neuron results. This, therefore, shows that pooling neurons tuned to dissimilar features with neurons tuned to similar features would

[9]We note that some pairs exhibited negative noise correlation, which, had a negligable effect on the information neurons convey.

improve the accuracy of the information a population conveys when noise correlation is present, as the beneficial effects would partially reduce the harmful effects. Finally, we showed that, when we pooled all of the neurons with similar and dissimilar tuning, the overall effect of noise correlation limited the information gain. That is, information increased with the number of neurons but eventually saturated when the size of the population reached ~ 45 neurons. In summary, we have shown that noise correlation can have a substantial effect on the capacity of population rate codes in the rat olfactory bulb. However, as we are uncertain as to whether the correlations in the area we analysed are representative of the whole olfactory bulb, and which neurons are pooled together within this system, i.e. which neurons are connected to the downstream neurons, we are also uncertain as to which effects of noise correlation, and at what magnitude, occurs in the rat olfactory bulb. Thus, if the brain utilises population rate coding and noise correlation is predominately positive, the brain may pool particular neurons with specific tuning to achieve certain aims. For example, if the brain's aim is to extract optimal information from a few neurons in the presence of positive noise correlation then we would expect that downstream neurons would be connected to neurons either:

1. tuned to dissimilar features, or,

2. tuned to similar and dissimilar features so that the harmful effect of noise correlation is canceled out.

Alternatively, the brain may use noise correlation to limit (optimise) the amount of information a large set of neurons convey. That is, if downstream neurons need to be connected to large numbers of neurons for one or both of the following reasons:

1. Information is encoded in a large neuronal area.

2. To resolve the problem of cells dying. That is, a large number of neurons may be connected to the downstream neurons so that they convey some redundant information, thus, if any cells die the information they encode/convey is not lost.

the brain may connect downstream neurons to a combination of neurons tuned to similar and dissimilar features. This combination, for example, may be chosen so

that the above points are addressed, and, so that noise correlation limits (optimises) the amount of information the neurons transmit to the downstream neurons. The brain may use noise correlation to limit the information a large population of neurons conveys for one or both of the following reasons:

1. To limit (optimise) the amount of information a large population conveys to prevent the amount exceeding, i.e. overestimating, the amount extracted by, for example, external stimulus receptors. The amount extracted by the receptors could be exceeded by a large population if the information is allowed to increase linearly with the number of neurons pooled (as discussed in Averbeck et al., 2006). If the brain forces information to increase nonlinearly with the number of neurons using correlations, it raises the question as to how much information is lost and whether this loss has a considerable effect on ones ability to discriminate between particular stimuli. From our results we showed that noise correlation reduced the amount of information a population conveyed by ∼30% (shown in fig. 7.6 panel (a)). Therefore, if the levels of noise and signal correlation discussed are representative of the entire olfactory bulb, the information provided by the firing rates of the mitral cells is limited.

2. To limit (optimise) the amount of information populations transmit to downstream neuronal areas to, for example, speed up information processing. In other words, the brain may achieve faster information processing if many downstream neuronal areas receive and process a limited amount of information in parallel rather than one area receiving and processing all the information.

Thus, if we want to fully understand the effects of noise correlation on population rate codes in the brain we need to study the effects of noise correlation in conjunction with how the neurons under analysis are connected. This could be a future investigation topic.

I_s replacing I in information analyses

The measure of I_s has been commonly used, for example in (Hung et al., 2005; Olshausen et al., 1996; Simoncelli et al., 2001), as a surrogate for the true informa-

tion I. This is because estimating the true information a large neuronal population can convey requires many neurons to be recorded simultaneously, whereas I_s requires only single cell recordings, and, fewer trials. Thus, if one uses I_s, one would also assume there is no noise correlation between the neurons. In this chapter we showed that I_s and I were significantly different, and therefore, proves we cannot simply assume noise is not correlated as the results or conclusions acquired may be incorrect. Any work that has substituted I_s with I should be treated with caution especially when studies involved the analysis of large neuron ensembles.

7.5 Summary

In this chapter we have shown that noise correlation can affect the encoding accuracy of population rate codes in the brain. However, whether noise correlation increases or decreases information in the brain, and at what magnitude, is dependent on the tuning similarity of the neurons pooled. Thus, if we wish to determine whether noise correlation does effects encoding accuracy in the brain we would need to determine which neurons, with respect to their tuning similarity, are pooled together. This could be a future investigation point. In addition, the results have also demonstrated that we cannot simply ignore correlated noise in the brain when investigating population rate codes, as it will bias our conclusions as to the how much information population rate codes can convey in the brain.

Chapter 8

Discussion

How a stimulus, such as an odour, an image or a sound, is represented in the brain has been controversially discussed over the years. It is becoming clearer that a stimulus is represented in the brain by a population code. That is, populations of neurons encode all the information about a stimulus and that each neuron encodes information about more than one stimulus (Averbeck et al., 2006; Barlow et al., 1964; Rolls et al,. 1997). However, how information is transmitted by a population of neurons is uncertain. The two main theories as to how a population transmits information are:

1. the population rate coding hypothesis, which assumes that stimulus information is transmitted in the spiking rates of each neuron (i.e. in the number of spikes fired by each neuron within a relevant time period) in an ensemble. (Averbeck et al., 2006; Barlow et al., 1964; Georgopoulos et al., 1986; Hubel et al., 1959; Maunsell et al., 1983; Mehring et al., 2003; Pouget et al., 2003; Quian Quiroga et al., 2005; Quian Quiroga et al., 2009; Reich et al., 2001; Rolls et al., 1997; Stopfer et al., 2003). In addition, the capacity of a population rate code is thought to be dependent on the noise correlation between the neurons in the ensemble (Averbeck et al., 2006). That is, noise correlation between neurons may contribute to, or reduce, the information a population conveys.

2. the population temporal coding hypothesis, which assumes that information is transmitted in the precise times each neuron in the ensemble fires spikes,

and/or, in the relationships (correlations) between the precise times multiple neurons fire spikes (Abeles 1991; Abeles et al., 1993; Aertsen et al., 1989; Arieli et al., 1995; Bair 1999; Castelo-Branco et al., 2000; Eckhorn et al., 1988; Engel et al., 1991; Fries et al., 1997; Gray et al., 1989; Krahe et al., 2002; Kreiter et al., 1996; Lestienne 2001; von der Malsburg, 1981; Markram et al., 1996; Panzeri et al., 2001b; Prut et al., 1998; Reich et al., 2001b; Riehle et al., 1997; Singer, 1999; Volgushev et al., 1998).

Thus, to acquire a more definitive answer as to how stimulus information is represented by neuronal populations in the brain, we need to test/investigate, i.e. support or falsify, each of the population coding hypotheses. In other words, we need to determine whether neurons respond to stimuli in the hypothesised ways, what, and how much, information such responses convey about the stimuli, and address any questions related to the hypotheses. A promising way of testing/investigating these hypotheses, i.e. acquire an understanding of how a neuronal population conveys stimulus information, at the level of multiple cell-cell interactions is to record and analyse the simultaneous spiking activity of multiple individual neurons from the brain of a test subject when a stimulus is, and is not, presented. With the advent of multi-electrode arrays it is now possible to record such data. However, before we can investigate/test the population coding hypotheses using such recordings, we need to determine the number of neurons recorded by the MEA and their spiking activity, after spike detection, using an automatic spike sorting algorithm (we refer to the spiking activity of the neurons extracted from the MEA recordings as MEA sorted data). While there are many automatic spike sorting methods available, they have limitations. In addition, we are lacking methods to test/investigate the population coding hypotheses in detail using the MEA sorted data. That is, methods that show whether neurons respond in a hypothesised way and, if they do, shows how the stimulus is represented within the recorded area. Thus, in this thesis, we proposed an automatic spike sorting method, which avoids the limitations of other standard spike sorting methods. In addition, we addressed the population rate coding hypothesis. More specifically, we developed a method for testing/investigating the population rate coding hypothesis using MEA data, i.e. detecting hot spots within the recorded area. We then addressed the question: can noise correlation affect the amount of

information a population rate code conveys, using MEA sorted data recorded from a rat's olfactory bulb. We discuss our results below.

8.1 Spike Sorting

To interrogate MEA data we need to determine the number of neurons and their corresponding spiking activity from the extracellular recordings. To achieve this we implement two stages, spike detection and then spike sorting. Spike detection refers to the process of identifying the spike events or action potentials from the voltage trace recorded by an electrode and extracting spike shapes. Spike sorting refers to the process of determining the number of neurons recorded by each electrode and the assignment of each spike to the neuron that produced it. In other words, it refers to the process of grouping spike shapes into clusters based on the similarity of their shapes, where each group of spike shapes are different from the ones belonging to other groups. Therefore, the clusters formed correspond to the spiking activity of different neurons. Spike sorting is a 2 stage process: feature extraction (which refers to the process of extracting distinctive features from the spike shapes) and clustering (which refers to the process of clustering the spikes based on the features extracted in the feature extraction stage).

In chapter 5, we proposed a spike sorting method called SOMA. SOMA is a fully automatic process (i.e. a method that automatically extracts/determines a number of features from the spike shapes that form a number of separable clusters in the feature space, and, automatically determines the number of, and groups the spikes into, clusters). We require automatic methods to avoid the problems/limitations of manual methods. This is important because:

1. manually spike sorting multi-electrode array data can be very time consuming/impractical due to the hundreds of thousands of mixed spike waveforms they can record.

2. manual methods typically perform worse than automatic methods due to the errors users introduce (Wood et al., 2004).

Our method also:

1. avoids the limitations of other automatic spike sorting methods. That is, our method:

   (a) is not based on assumptions about the distribution and variance of the clusters in the feature space, which is important as the contents of datasets can vary, and therefore, makes our method more reliable and consistent.

   (b) does not require the user to specify optimal parameter values (such as providing the method with the number of clusters in the feature space, criteria associated to the cluster boundaries and/or the number of features to extract from the spikes shapes) to achieve a high performance. This is an important consideration as, typically:

      i. we would not know the optimal value(s) to use.

      ii. we cannot tune (optimise) the parameter values (i.e. run the method multiple times with different values until a particular result is achieved) as we would not know the optimal result (i.e. we would not know the correct number of clusters the method should group the spikes into, and, which spikes should be grouped together).

2. reduces computational costs where possible. This is an important consideration as processing the large amount of data MEAs can record can be computationally expensive.

Our method consists of using a novel pre-processing stage, which involves extracting features (based on the principal components and the curvature of the spike shapes) in combination with a novel self-organising map-based clustering method.

Our clustering method is an improved version of the standard self organising map clustering procedure (Kohonen 1984; Ritter et al., 1989) and avoids its limitations (discussed in chapter 2). That is, our clustering method automatically identifies the number of clusters in the feature space and improves cluster separation. In other words, our method forms an implicit boundary around each cluster, and thus, the points inside the boundary of each cluster are grouped together. This process, therefore, reduces the chance of misclassification. The standard SOM clustering method does not accomplish this.

We validated our new spike sorting approach by applying it to multi-electrode array datasets acquired from the rat olfactory bulb, and from the sheep infero-temporal cortex, and using simulated data. We also compared our method with other standard automatic spike sorting methods, namely the wavelet pre-processing procedure (described in (Quian Quiroga et al., 2004)), and the clustering techniques, Klustakwik (Harris 2000), which uses the CEM algorithm (Celeux et., 1992) and Waveclus (Quian Quiroga et al., 2004), which uses the super-paramagnetic clustering (SPC) technique (Blatt et al., 1996) (we discussed these methods in chapter 2 and 5). We compared these methods by:

1. using datasets containing single neuron recordings, i.e. datasets that contain a known number of neurons and the spikes each fired. We used these datasets so we could evaluate the performance of each method. That is, we could determine whether each method identified the correct number of neurons and whether each spike was correctly classified.

2. running them in a way that reflected our interests in automatic spike sorting, and, how we would use them in practice. That is, we compared the results obtained from running each method once with the default values. We ran the methods in this way as, in practice, we would:

   (a) use the default values with each method for the reasons discussed in bulleting points 1. b) i) and 1. b) ii) at the beginning of this section.

   (b) run the method once, as, running the methods multiple times may produce different results, but, we would not know which run produced the best results.

We found that our method was at least as good as the others, i.e. wavelets formed separable clusters, and Klustakwik and Waveclus found the correct number of clusters and classified a high percentage of the spikes correctly. However, our pre-processing and clustering method outperformed them when the spike shapes from each neuron were very similar in shape, and, when the spike shapes were highly distorted. The results, therefore, show that our new pre-processing technique and clustering method worked efficiently, in terms of the number of clusters (neurons)

found and the number of spikes correctly classified, without assumptions and without the need for user involvement (i.e. optimal pre-defined criteria).

We note that, in practice, it is difficult to evaluate a method's performance. This is because we would usually have no information about the number of neurons recorded and which spikes correspond to each neuron. Thus, it is difficult to know whether each group of spikes acquired by SOMA, or by any other spike sorting method, refers to the spiking activity of one neuron. Therefore, it is always worthwhile to confirm the validity of the results from SOMA with other evaluation measures when possible. For example, we could determine whether each group of spikes fits a particular interspike interval distribution or contains a particular refractory period.

In this thesis our main aim was to address the issue of unsupervised spike sorting. However, there are other spike sorting issues we did not address, in particular, the issues of overlapping spikes and bursting cells. The issues are explained in more detail below:

1. the issue of overlapping spikes refers to the problem of two neurons firing at, or near, the same time. That is, if two neurons fire in synchrony or with a small delay, it is possible that they generate overlapping spikes, i.e. a shape generated by the sum of the two spikes. Thus, when two neurons fire with a slight delay the spike shape they generate would have double peaks. In addition, when two neurons fire in synchrony, the spike shape generated may be very different from the two neurons and could be regarded as a spike from a third neuron. Thus, we need to be careful that the overlapping spikes, if they are not detected at the spike detection stage, are not considered by our spike sorting algorithm as the spiking activity of a third neuron.

2. the issue of bursting cells refers to the problem of a neuron firing a fast sequence of spikes. That is, the amplitude of each spike shape in a burst usually differs, therefore, we need to be careful that a burst of spikes from a neuron are not considered as spikes from different neurons.

Some solutions have been proposed for the above two issues but have yet to show show optimal results (Quian Quiroga 2007). Thus, for future work, we could address these issues and incorporate them into our process.

Finally, for future work, it would be worth investigating whether we could adapt/apply SOMA to tetrode recordings or in real time applications. For example, to enable multiple spike trains to be inputs to neural prosthetic devices and brain machine interfaces (Taylor et al., 2002; Serruya et al., 2002; Chapin et al., 2004).

8.2 MEA Analysis Tool For Investigating The Population Rate Code Hypothesis

To test/investigate the population rate coding hypothesis in detail using MEA sorted data recorded before and during stimulus presentation, we need methods that provide a detailed statistical analysis of the stimulus evoked changes in the spiking rates of the neurons recorded, and show, the responsive, and most responsive, subareas of neurons (i.e. the subareas exhibiting the most significant change in spiking rates) within the MEA recorded area. This type of analysis method enables us to address questions such as:

1. does the whole recorded area respond to the stimulus, or, does one or many small subareas respond? In other words, is the stimulus represented spatially in the responses of small groups of neurons, or, is it represented in the responses of a large group of neurons?

2. do neurons respond to a stimulus with the same or different spiking rates?

In chapter 6, we proposed a new method to achieve the above analysis called MEANOVA, which consists of applying the multi variate analysis of variance test (MANOVA) on MEA sorted data. The standard application of MANOVA does not easily achieve the above analysis as it is designed to analyse the changes in the spiking rates of all of the neurons as a whole. In other words, MANOVA is designed to perform one test on the whole group of neurons to determine/show whether the neurons, in general, responded to a stimulus, and not, to highlight the responsive and the most responsive subareas within the recorded area. MANOVA also has limitations. Firstly, MANOVA is not very informative. That is, it only shows whether a change in the spiking rates of a population of neurons is significant and not how significant the change is. Secondly, the standard MANOVA test, when used on a

dataset containing a large set of unresponsive neurons and a very small set of responsive ones, may incorrectly assert there was no stimulus evoked changes in the spiking rates of the neurons. In other words, the responsive neurons may be hidden amongst the vast number of unresponsive neurons in a MANOVA test. We showed an example of this in chapter 6.

Our method consists of:

1. using the MANOVA test, where we modified its output. We modified the output of MANOVA so that it produces a significance score to show how significant the changes in the spiking rates of a population of neurons were overall when a stimulus was presented. In other words, the score shows how responsive a population of neurons was, in general, to a stimulus. Thus, our method avoids the first limitation of MANOVA discussed above, i.e. our modified version is more informative than the standard MANOVA test.

2. applying the modified version of MANOVA to the spiking rates of neurons in subareas, small (local) and large (global), within the recorded area. In other words, our method associates a significance score to different subareas of neurons in the recorded MEA area. Thus, the higher the significance score associated to a subarea, the more responsive the neurons in that subarea were, in general, to a stimulus. Thus, our method avoids the second limitation of MANOVA discussed above, i.e. as it analyses the differences in the spiking rates of small and large numbers of neurons, it would uncover a small set of responsive neurons amongst a large set of unresponsive ones (we showed this in chapter 6).

We validated MEANOVA using data recorded from a rat's olfactory bulb and data obtained from a simple computational network model. We used the computational model and biological data to show that MEANOVA works, i.e., it can identify and show the responsive, and the most responsive, subareas within a neuronal area. We also used the computational model to show the limitation of MANOVA (discussed above), i.e. that MANOVA can incorrectly state that a stimulus had no effect on a large population of neurons when one responsive neuron is embedded in a large number of unresponsive neurons. We also showed that MEANOVA overcomes this

limitation. The results showed that MEANOVA is a valid method of applying MANOVA to MEA data for the purpose of acquiring information about how and whether a stimulus affects the spiking rates of neurons across an MEA recorded area.

8.3 The Effects Of Noise Correlation On The Information Encoded In A Population Rate Code

The capacity of a population rate code is thought to be dependent on the noise correlation between neurons in the ensemble (Averbeck et al., 2006; Quian Quiroga et al., 2009). That is, noise correlation between neurons may determine whether neurons contribute independent, synergistic or redundant information to the ensemble.

In chapter 7 we addressed this hypothesis. More specifically, we addressed the question: can noise correlation significantly affect the amount of information a large population of neurons conveys about a set of external stimuli. To address this, we used the MEA sorted data containing 236 neurons recorded from a rat's olfactory bulb when 6 stimuli was presented (we described the recording/experiment procedure in chapter 3). We used this data as we found from initial tests that noise was predominately positively correlated in this system. We investigated whether correlated noise can affect the amount of information the neurons convey in their spiking rates about the six stimuli presented. To quantify the effect of noise correlation we used a mutual information measurement method developed by (Franco et al., (2004)). We used it to calculate the mutual information between the population response and the six stimuli presented, when the responses were correlated and uncorrelated,. Thus, the difference between the information the correlated and uncorrelated responses conveyed about the stimuli represents the effect of noise correlation.

We found from initial investigations of the rat olfactory bulb, as have other empirical studies of the brain (e.g. Gawne et al., 1993; Lee et al., 1998; Averbeck et al., 2006), that noise correlation is predominately positive and the average noise correlation coefficient is small, i.e. in the range 0.1-0.2. We also found, as have many empirical studies (Panzeri et al., 2001; Petersen et al., 2001; Averbeck et al.,

2003; Golledge et al., 2003; Romo et al., 2003), that the effect noise correlation has on the information small population codes (< 30) convey is small. In addition, we, and other studies have found the effect (i.e. whether noise correlation increases or decreases information) to be dependent on the tuning similarity (signal correlation), and the noise correlation, between the responses of the neurons pooled. That is, when the noise and signal correlation between neurons in a population is positive, or, when the noise and signal correlation between neurons in a population is negative, noise correlation decreases the information the population can convey (i.e. the effect of noise correlation is harmful). In addition, when the noise and signal correlation between neurons in a population is negative and positive respectively, or, when the noise and signal correlation between neurons in a population is positive and negative respectively, noise correlation increases the information the population can convey (i.e. the effect of noise correlation is beneficial)[1]. When we invesigated the effect of noise correlation on larger populations (> 30) we found the effect of noise correlation to be much larger. More specifically, we found that, as noise correlation was predominately positive, noise correlation limited the information conveyed by the entire population. This is because the beneficial effects of noise correlation did not fully cancel out the harmful effects. That is, the overall harmful effect of noise correlation increased with the number of neurons pooled, and as a consequence, information saturated when the size of the population reached $\sim$45. Thus, the population conveyed $\sim 70\%$ of the maximum information that could be conveyed about the six stimuli. In other words, the population would have conveyed 100% of the information if the noise was not correlated. These results, therefore, supports the theoretical results in (e.g. Abbot et al., 1999; Averbeck et al., 2004; Oram et al., 1998; Panzeri et al., 1999; Zohary et al., 1994). That is, the effect of noise correlation can increase with the size of the population and can substantially limit the information gain of a large population rate code.

While our results show that noise correlation can have a substantial effect on, i.e. limit, the capacity of population rate codes, we cannot be certain that this limiting effect, and at what magnitude, occurs in the olfactory bulb, or more specifically, in

[1]However, we note that in all our studies the effect of negative noise correlation was not significant.

the brain. This is because we are uncertain as to whether the correlations in the area we analysed are representative of the whole brain, and, we are uncertain as to which neurons are pooled together within it, i.e. which neurons are connected to the downstream neurons. However, if the brain does utilise population rate codes and noise correlation is predominately positive in the brain, the brain may connect (pool) neurons tuned to similar and/or dissimilar features to achieve certain aims. For example, if the brain's aim is to extract optimal information from population rate codes in the presence of predominately positive noise correlation, the brain may connect neurons tuned to dissimilar features so that noise correlation has no harmful effect (for other examples see chapter 7). Thus, to obtain a more definitive answer as to how noise correlation affects the amount of information population rate codes convey, we need to address the question with respect to how neurons are connected within the brain. This could be a point for future investigation. Nevertheless, our results do show that noise correlation can have a significant effect on encoding accuracy, and therefore, shows the importance of analysing the effects of noise correlation in the brain. In other words, we have shown that we cannot simply study the capacity of population rate codes by ignoring/omitting noise correlation, which some work has done (e.g. Hung et al., 2005; Olshausen et al., 1996).

In summary, we have achieved our main aims of producing tools to analyse MEA recordings, i.e. a spike sorting method and a method to use with MEA sorted data to investigate/test the population rate coding hypothesis. In addition, we have showed that noise correlation can significantly effect the accuracy of population rate codes. This thesis has therefore furthered, and has provided ways to further using MEA data, our understanding of how the brain works.

References

Abbott LF, Dayan P (2001) Theoretical Neuroscience. Cambridge, MA: MIT Press.

Abbott LF, Dayan P (1999) The effect of correlated variability on the accuracy of a population code. Neural Comput. 11:91-101.

Abeles M, Goldstein M (1977) Multispike train analysis. Proc. IEEE, 65:762-773.

Abeles M (1991) Corticonics. Cambridge: Cambridge Univ. Press.

Abeles M, Bergman H, Margalit E, Vaadia E (1993) Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. J. Neurophysiol. 70:1629-1638.

Adrian ED and Zotterman Y (1926) The impulses produced by sensory nerve endings: Part II: The response of a single end organ. J. Neurophysiol 61:151-71.

Adrian ED (1928) The Basis of Sensations. Norton, New York.

Aertsen AM, Gerstein GL, Habib MK, Palm G (1989) Dynamics of neuronal firing correlation: modulation of effective connectivity. J. Neurophysiol. 61:900-917.

Albright TD, Jessell TM, Kandel ER, Posner MI (2000) Neural science: A century of progress and the mysteries that remain. CELL. 100:S1-S55.

Amen D, Wu JC, Bracha H (2000) Kaplan and Sadock. ed. Functional neuroimaging in clinical practice.

Andersen, RA, Burdick, J. W., Musallam, S.,Pesaran, B. Cham, J. G. (2004) Cognitive neural prosthetics. Trends Cogn. Sci. 8:486-493.

Anderson T (1958) An introduction to multivariate statistical Analsis. Wiley, New York.

Arieli A, Shoham D, Hildesheim R, Grinvald A (1995) Coherent spatiotemporal patterns of ongoing activity revealed by real-time optical imaging coupled with single-unit recording in the cat visual cortex. J. Neurophysiol. 73:2072-2093.

Atick JJ, Redlich AN (1990) Towards a theory of early visual processing. Neural Comput 2:308-320.

Atick JJ, Li Z, Redlich AN (1992) Understanding retinal color coding from first principles. Neural Comput 4:559-572.

Averbeck BB, Crowe DA, Chafee MV, Georgopoulos AP (2003) Neural activity in prefrontal cortex during copying geometrical shapes. II. Decoding shape segments from neural ensembles. Exp. Brain Res. 150:142-153.

Averbeck BB, Chafee MV, Crowe DA, Georgopoulos AP (2005) Parietal representation of hand velocity in a copy task. J Neurophysiol 93:508-518.

Averbeck BB, Lee D (2004) Coding And Transmission Of Information By Neural Ensembles. Trends Neurosci. 27:225-230.

Averbeck BB, Latham PE, Pouget A (2006) Neural correlations, population coding and computation. Nat Rev Neurosci. 7:358-366.

Bair W (1999) Spike timing in the mammalian visual system. Curr. Oppin. Neurobiol 9:447-453.

Bair W, Zohary E, Newsome WT (2001) Correlated firing in macaque visual area MT: time scales and relationship to behavior. J Neurosci. 21:1676-1697.

Baker JF, Petersen SE, Newsome WT, Allman JM (1981) Visual response properties of neurons in four extrastriate visual areas of the owl monkey (Aotus trivirgatus): A quantitative comparison of medial, dorsomedial, dorsolateral, and middle temporal areas. J. Neurophysiol. 45:397-416.

Baker S, Lemon R (2000) Precise spatiotemporal repeating patterns in monkey primary and supple- mentary motor areas occur at chance level. J. Physiol. (Lond) 84:1770-1780.

Barlow HB, Hill RM, Levick WR (1964) Retinal ganglion cells responding selectively to direction and speed of image motion in the rabbit. J. Physiol. (Lond.) 173:377-407.

Barlow HB (1972) Single units and sensation: a neuron doctrine for perceptual psychology? Percep- tion 1:371-394.

Bartlett MS (1954) A note on the multiplying factors for various χ^2 approximations. Journal Of The Royal Statistical Society 16:294-298.

Belliveau JW, Kennedy DN, McKinstry RC, Buchbinder BR, Weisskoff RM, Cohen MS, Vevea JM, Brady TJ, and Rosen BR (1991) Functional mapping of the human visual cortex by magnetic resonance imaging. Science 254:716-719.

Bergman H, DeLong M (1992) A personal computer-based spike detector and sorter: implementation and evaluation. J Neurosci Methods 41:187-197.

Bhandawat, V. (2007) Sensory processing in the Drosophila antennal lobe increases reliability and separability of ensemble odor representations. Nat. Neurosci. 10:1474-1482

Bialek W, Rieke F, De Ruyter van Steveninck RR, Warland D (1991) Reading a neural code. Science 252:1854-1857.

Bialek W, Rieke F (1992) Reliability and information transmission in spiking neurons. TINS 15:428-434.

Biffi E, Ghezzi D, Pedrocchi A , Ferrigno G (2010) Development and Validation of a Spike Detection and Classification Algorithm Aimed at Implementation on Hardware Devices. Computational Intelligence and NeuroscienceVolume 2010 (2010), Article ID 659050, 15 pagesdoi:10.1155/2010/659050

Bishop C (1995) Neural Networks For Pattern Recognition, first ed. Clarendon Press: Oxford 310-318.

Blatt M, Wiseman S, Domany E (1996) Super-paramagnetic clustering of data. Phys. Rev. Lett. 76:3251-3254.

Blatt M, Wiseman S, Domany E (1997) Data clustering using a model granular magnet. Neural Computation, 9:1805-1842.

Borst A. and Theunissen FE (1999) Information theory and neural coding. Nat Neurosci, 2(11):947-957

Bracci E, Centonze D, Bernardi G, Calabresi P (2003) Voltage-dependent membrane potential oscillations of rat striatal fast-spiking interneurons. J Physiol 549:121-130.

Bracci E, Centonze D, Bernardi G, Calabresi P (2003b) Voltage-dependent membrane potential oscillations of rat striatal fast-spiking interneurons. J Physiol 549:121-130.

Brody TA, Flores J, French JB, Mello PA, Pandey A, Wong SSM (1981) Randommatirx physics: spectrum and strength fluctuations. Review of Modern Physics 53:385-479.

Brody CD (1999) Correlations without synchrony. Neural Comp. 11:1537-1551.

Braitenberg V, Schuz A (1998) Cortex: Statistics and Geometry of Neuronal Con-

nectivity Springer- Verlag, Berlin, 2nd edition.

Britten, KH, Shadlen, MN, Newsome, WT, Movshon, JA (1992) The analysis of visual motion: a comparison of neuronal and psychophysical performance. J. Neurosci. 12:4745-4765.

Brown EN, Frank LM, Tang D, Quirk MC, Wilson MA (1998) A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. J Neurosci 18:7411-7425

Brown EN, Kass RE, Mitra PP (2004) Multiple Neural Spike Train Data Analysis: State Of The Art And Future Challenges. Nat Neurosci 7(5):456-461.

Buhusi CV, Meck WH (2005) "What makes us tick? Functional and neural mechanisms of interval timing". Nat Rev Neurosci 6 (10): 755-65.

Buracas GT (1998) Efficient discrimination of temporal patterns by motion-sensitive neurons in primate visual cortex. Neuron 20:959-969

Butts DA, Weng C, Jin J, Yeh C, Lesica N, Alonso J, Stanley G (2007) Temporal precision in the neural code and the timescales of natural vision. Nature 449:92-95

Buszaki G (2004) Large Scale Recording Of Neuronal Ensembles. Nat Neurosci 7(5):446-451.

Buszaki G (2006) Rhythms of the brain. Oxford University Press.

Cariani P (2001) Temporal coding of sensory information in the brain. Acoust Sci Tech 22:77-84.

Carr C, Heiligenberg W, Rose G (1986) A time comparison circuit in the electric fish midbrain. I. Behavior and physiology. J. Neurosci. 6:107-10.

Castelo-Branco M, Goebel R, Neuenschwander S, Singer W (2000) Neural synchrony correlates with surface segregation rules. Nature 8:685-689.

Celeux G, Govaert G (1992) A classification EM algorithm for clustering andtwo stochastic versions. Comput Statistics Data Anal 14(3):315-322.

Chandra R, Optican LM (1997) Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network. IEEE Transactions on Biomedical Engineering, vol. 44 (5):403-412.

Chang JY, Chen LF, Luo L, Shi H, Woodward DJ (2002) Neuronal responses in the frontal cortico-basal ganglia system during delayed matching-to-sample

task:ensemble recording in freely moving rats. Exp. Brain. Res. 142:67-80

Chapin J K (2004) Using multi-neuron population recordings for neural prosthetics Nat. Neurosci. 7:452-455.

Christen M, Nicol AU, Kendrick KM, Ott T, Stoop R. Stabilization not synchronization: stable neuron clusters signal odour presentation. Neuroreport (In Press).

Chui, C. (1992) An introduction to wavelets. San Diego, CA: Academic Press.

Clark A (2000) A Theory of Sentience. Oxford University Press. p. 43.

Cohen JE, Newman CM (1984) The stability of large random matrices and their products. The Annals of Probability 12:283-310.

Connor C (2005) Friends and grandmothers. Nature 435 (7045):1036-1037

Constantinidis C, Franowicz MN, Goldman-Rakic PS (2001) Coding specificity in cortical microcircuits: a multiple-electrode analysis of primate prefrontal cortex. J Neurosci 21:3646-3655.

Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13 (1):21-27.

Cover TM, Thomas JA (1991) Elements of information theory. Wiley, New York

Cover TM, Thomas JA (2006) Elements of Information Theory. Wiley sons, Hoboken, New Jersey.

Creutzfeldt OD, Watanabe S, Lux HD (1966) Relations between EEG phenomena and potentials of single cortical cells. I. Evoked responses after thalamic and epicortical stimulation. Electroencephalogr Clin Neurophysiol 20 (1):1-18.

Csicsvari J, Hirase H, Czurko A, Buzsaki G (1998) Reliability and state dependence of pyramidal cell-interneuron synapses in the hippocampus: an ensemble approach in the behaving rat. Neuron 21:179-189.

Davison A, Feng JF, Brown D (2003) Dendrodendritic inhibition and odour-induced synchronization in a detailed olfactory bulb model. J Neurophysiology 90:181-193.

Dayan P, Abbott LF (2001) Theoretical Neuroscience: Computational and Matehmatical Modeling of Neural Systems. MIT Press.

Debarbieux F, Audinat E, Charpak S (2003) Action potential propagation in dendrites of rate mitral cells in vivo. J Neurosci 23: 553-560.

Delescluse M and Pouzat C (2006) Efficient spike-sorting of multi-state neurons using inter-spike intervals information J. Neurosci. Methods 150: 16-2.

Deshmukh SS, Bhalla US (2003) Representation of Odor Habituation and Timing in the Hippocampus. The Journal of Neuroscience. 23(5):1903

Desimone R, Albright T, Gross C, Bruce C (1984) Stimulus-selective properties of inferior temporal neurons in the macaque. J. Neurosci. 4:51-62.

Desmaisons D, Vincent JD, Lledo PM (1999) Control of action potential timing by intrinsic subthreshold oscillations in olfactory bulb ouput neurons. J Neurosci 19:1027-1037.

Dong HW, Heinbockel T, Hamilton KA, Hayar A, Ennis M (2009) Metabotropic glutamate receptors and dendrodendritic synapses in the main olfactory bulb. Ann N Y Acad Sci 1170:224-238.

Durrant S, and Feng JF (2006) Negatively-correlated firing: the functional meaning of lateral inhibition within cortical columns. Bio. Cyber. 95: 431-453

Eckhorn R, Bauer R, Jordan W, Brosch M, Kruse W, Munk M, Reitbock HJ (1988) Coherent oscillations: A mechanism of feature linking in the visual cortex? Biol. Cyber. 60:121-130.

Egger V, Svoboda K, Mainen ZF (2003) Mechanisms of lateral inhibition inthe olfactory bulb: efficiency and modulation of spike-evoked calcium influx into granule cells. J Neurosci 23:7551-7558.

Eggermont JJ (1990) The Correlative Brain: Theory and Experiment in Neural Interaction. Berlin, Springer-Verlag.

Engel AK, Konig P, Singer W (1991) Direct physiological evidence for scene segmentation by temporal coding. Proc. Natl Acad. Sci. USA 88:9136-9140.

Engel A, Singer W (2001) Temporal binding and the neural correlates of sensory awareness. Trends Cogn. Sci. 5:16-25.

Engineer CT (2008) Cortical activity patterns predictspeech discrimination ability. Nature Neurosci. 11:603-608.

Ermentrout GB, Galn RF, Urban NN (2008) Reliability, synchrony and noise. Trends Neurosci. 31(8):428-34.

Faisal AA, Selen L, Wolpert D (2009) Noise in the nervous system. Nature review neuroscience 9:292-303.

Fantana P. (2008) Rat Olfactory Bulb Mitral Cells Receive Sparse Glomerular Inputs. Neuron 59:802-814.

Fee MS, Mitra PP, Kleinfeld D (1996) Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability. J Neurosci Methods. 69(2):175-188.

Feller W (1971) An Introduction to Probability Theory and Its Application. Jhon Wiley and Sons: New York.

Feng, JF (2004) Computational Neuroscience: A Comprehensive Approach Chapman and Hall/CRC Press.

Fischer H, Leigh AE, Tate AJ, Kendrick KM. Neural activity patterns in sheep temporal cortex during face emotion recognition tasks. Behav Brain Funct (submitted).

Feng JF, editor. Computational Neuroscience: A Comprehensive Approach. Chapman and Hall/CRC: Florida, 375-396.

Fitzpatrick DC, Batra R, Stanford TR, Kuwada S (1997) A neuronal population code for sound localization. Nature 388 (6645):871-874.

Franco L, Rolls ET, Aggelopoulos NC, Treves A (2004) The Use Of Decoding To Analyze The Contribution To The Information Of The Correlations Between The Firing Of Simultaneously Recorded Neurons. Exp Brain Res 155:370-384.

Francis, G. (2007) Introduction to SPSS for Windows: v. 15.0 and 14.0 with Notes for Studentware (5th ed.). Sydney: Pearson Education. (Section 5.3).

Friedrich RW, Stopfer M (2001) Recent dynamics in olfactory population coding. Curr. Opin. Neurobiol. 11:468-474.

Fries P, Roelfsema PR, Engel AK, Konig P, Singer W (1997) Synchronization of oscillatory responses in visual cortex correlates with perception in interocular rivalry. Proc. Natl. Acad. Sci. USA 94:699-704.

Fries P (2001) A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. TICS 9: 474-480.

Garcia P, Suarez CP, Rodriguez J, Rodriguez M (1998) Unsupervised classification of neural spikes with a hybrid multi-layer artificial neural network. J Neurosci Methods 82(1):59-73.

Gawne TJ, Richmond BJ (1993) How independent are the messages carried by adjacent inferior temporal cortical neurons? J Neurosci. 13:2758-2771.

Gawne TJ, Kjaer TW, Hertz JA, Richmond BJ (1996) Adjacent visual cortical

complex cells share about 20% of their stimulus-related information. Cereb Cortex. 6:482-489.

Georgopoulos AP, Schwartz AB, Kettner RE (1982) On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. J. Neurosci. 2 (11):1527-1537.

Georgopoulos AP, Schwartz AB, Kettner RE (1986) Neuronal population coding of movement direction. Science 233 (4771):1416-1419.

Gibbons, JD (1985) Nonparametric Statistical Inference. 2nd edition, M. Dekker.

Gray CM, Singer W (1987) Stimulus-specific neuronal oscillations in the cat visual cortex: A cortical functional. Soc. Neurosci. Abstr. 13:404-413.

Gray CM, Konig P, Engel AK, Singer W (1989) Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. Nature 338:334-337.

Gray CM, Maldonado PE, Wilson M, McNaughton B (1995) Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. Journal of Neuroscience Methods 63:43-54.

Gross CG (1994) How inferior temporal cortex became a visual area. Cereb. Cortex. 5:455-69.

Gross CG, (2002) Genealogy of the Grandmother Cell. Neuroscientist. 8(5):512-518.

Gold C, Henze DA, Koch C, Buzsaki G (2006) On the Origin of the Extracellular Action Potential Waveform: A Modeling Study. J Neurophysiol 95:3113-3128.

Golledge HD (2003) Correlations, feature-binding and population coding in primary visual cortex. Neuroreport 14:1045-1050.

Hahnloser RH, Kozhevnikov AA, Fee MS (2002) An ultra-sparse code underlies the generation of neural sequences in a songbird. Nature 419:65-70.

Hahnloser RH, Kozhevnikov AA, Fee MS (2002b) An ultra-sparse code underlies the generation of neural sequences in a songbird. Nature 419:65-70.

Hamalainen M, Hari R, Ilmoniemi RJ, Knuutila J, Lounasmaa OV (1993) Magnetoencphalography - Theory, instrumentation, and applications to noninvasive studies of the working human brain. Reviews of Modern Physics 65: 413-497.

Harada Y, Takahashi T (1983) The calcium component of the action potential in spinal motoneurones of the rat. J. Physiol. 335:89-100.

Harris K, Henze D, Csicsvari J, Hirase H, Buzsaki G (2000) Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J Neurophysiol 84:401-414.

Harris K (2003) Klustakwik spike sorting, at http://klustakwik.sourceforge.net/.

Heitler B (2002) Klustawin spike sorting, at http://www.st-andrews.ac.uk/ wjh/klustawin/.

Hillel, AB, Spiro A, Stark E (2006) Spike sorting: Bayesian clustering of non-stationary data. J Neurosci Methods 157:303-316.

Hochberg LR (2006) Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature 442:164-71.

Hodgkin AL, Huxley AF (1939) Action potentials recorded from inside a nerve fibre. Nature 144:710-711.

Hollander M, and DA Wolfe (1973) Nonparametric Statistical Methods, Wiley.

Hopfield, J J (1996) Pattern recognition computation usingaction potential timing for stimulus representation. Nature 376:33-36.

Hsuing C (1997) A First Course in Differential Geometry (Series in Undergraduate Texts). International Press of Boston Inc.

Hubel DH, Wiesel TN (1959) Receptive fields of single neurons in the cat's striate cortex. Journal of Physiology 148:574-591.

Hubel, D.H., Wiesel, T.N. (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol. 160:106-154.

Hulata E, Segev R, Shapira Y, Benveniste M, Ben-Jacob E (2000) Detection and sorting of neural spikes using a packets. Phys. Rev. Lett. 85:4637-4640.

Hulata E, Segev R, Ben-Jacob E (2002) A method for spike sorting and detection-based on wavelet packets and Shannons mutual information. J Neurosci Methods 117:1-12.

Hung CP, Kreiman G, Poggio T, DiCarlo JJ (2005) Fast readout of object identity from macaque inferior temporal cortex. Science 310:863-866.

Hunter JJ (1974a) Renewal Theory In Two Dimensions: Basic Results. Adv. Appl. Probab 6:376-391.

Hunter JJ (1974b) Renewal Theory In Two Dimensions: Asymptotic Results. Adv. Appl. Probab 6:546-562.

Huxter JR, Burgess N, OKeefe J (2003) Independent rate and temporal coding in

hippocampal pyramidal cells. Nature 425:828-832. Jolliffe IT (2002) Principal Component Analysis, second ed. Springer-Verlag: New York. pg:1-10.

Huxter JR, Senior TJ, Allen K, Csicsvari J (2008) Theta phase-specific codes for two-dimensional position, trajectory and heading in the hippocampus. Nature Neurosci. 11:587-594.

Joelving FC, Compte A, Constantinidis C (2007) Temporal Properties of Posterior Parietal Neuron Discharges During Working Memory and Passive Viewing. J Neurophysiol 97:2254-2266.

Johnson KO (1980) Sensory discrimination: decision process. J. Neurophysiol. 43:1771-1792

Johnson JL, Welsh JP (2003) Independently movable multielectrode array to record multiple fast-spiking neurons in the cerebral cortex during cognition. Methods 30:64-78.

Johnson RA, Wichern DW (1988) Applied Multivariate Statistical Analysis fourth Ed. Prentice Hall.

Juergens E, Guettler A, Eckhorn R (1999) Visual stimulation elicits locked and induced gamma oscillations in monkey intracortical and EEGpotentials, but not in human EEG. Exp. Brain Res. 129:247-259.

Kamondi A, Acsady L, Wang XJ, Buzsaki, G (1998) Theta oscillations in somata and dendrites of hippocampal pyramidal cells in vivo: activitydependent phase-precession of action potentials. Hippocampus 8:244-261.

Kandel ER, Schwarzt JH and Jessel TM (1991) Principles of Neural Science. 3rd ed. New York:McGraw-Hill.

Kendrick KM, GuevaraGuzman R, Zorrilla J (1997) Formation of olfactory memories mediated by nitric oxide. Nature 388:670-674.

Kim KH, Kim SJ (2000) Neural spike sorting under nearly 0 dB signal-to-noise ratio using non-linear energy operator and artificial neural-network classifier. IEEE Trans. Biomed. Eng. 47(10):1406-11.

Klimesch, W. (1999) EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. Brain Res. Rev. 29:169-195.

Klein S, Thorne BM (2007) Biological psychology. New York, N.Y.: Worth.

Kohonen T (2001) Self-Organizing Maps, second ed. Springer-Verlag: Berlin. pg:1-

40.

Kohonen T (1988) Self-Organizing and Associative Memory, New York: Springer-Verlag.

Krahe R, Kreiman G, Gabbiani F, Koch C,Metzner W (2002) Stimulus encoding and feature extraction by multiple pyramidal cells in the hindbrain of weakly electric fish. J. Neurosci. 22:2374-82.

Kreiman G, Koch C, Fried I (2000a) Category-specific visual responses of single neurons in the human medial temporal lobe. Nat. Neurosci. 3:946-53.

Kreiman G, Koch C, Fried I (2000b) Imagery neurons in the human brain. Nature 408:57-61.

Kreiman G, Fried I, Koch C (2002) Single neuron correlates of subjective vision in the human medial temporal lobe. Proc. Natl. Acad. Sci. USA 99:8378-83.

Kreiman G (2004) Neural coding: computational and biophysical perspectives. Physics of Life Reviews 1:71-102.

Kreiter A, Singer W (1996) Stimulus-dependent synchronization of neuronal responses in the visual cortex of awake macaque monkey. J. Neurosci. 16:2381-2396.

Latham PE, Nirenberg S (2005) Synergy, redundancy, and independence in population codes, revisited. J. Neurosci. 25:5195-5206.

Laurent G (2002) Olfactory network dynamics and the coding of multidimensional signals. Nat. Rev. Neurosci. 3:884-95.

Lee, C., Rohrer, W.H., Sparks, D.L. (1988) Population coding of saccadic eye movements by neurons in the superior colliculus. Nature 332 (6162), 357-360.

Lee D, Port L, Kruse W, Georgopoulos A (1998) Variability and Correlated Noise in the Discharge of Neurons in Motor and Parietal Areas of the Primate Cortex. J Neurosci 18(3):1161-1170.

Legatt, A. D., Arezzo, J., and Vaughan, H. G. (1980) Averaged multiple unit activity as an estimate of phasic changes in local neuronal activity: effects of volume-conducted potentials. Journal of Neuroscience Methods, 2(2):203-217.

Lestienne R (2001) Spike timing, synchronization and information processing on the sensory side of the central nervous system. Progr. Neurobiol. 65:545-91.

Lestienne R (2001) Spike timing, synchronization and information processing on the sensory side of the central nervous system. Progr. Neurobiol. 65:545-91.

Letelier JC, Weber PP (2000) Spike sorting based on discrete wavelet transform coefficients. J. Neurosci. Methods 101:93-106.

Lewicki MS (1994) Bayesian modeling and classification of neural signals. Neural Comput 6:1005-30.

Lewickiy M (1998) A review of methods for spike sorting: the detection and classification of neural action potentials.Network-Comp Neural R53-R78

Lindman HR (1974) Analysis of variance in complex experimental designs. San Francisco: W. H. Freeman Co. pg.33.

Lloyd SP (1982) Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (2):129-137.

Logothetis NK, Sheinberg DL (1996) Visual object recognition. Annu Rev Neurosci 19:577-621.

Mainen ZF, Sejnowski TJ (1995) Reliability of spike timing in neocortical neurons. Science 268:1503-6.

Mallat S (1989) A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans.PatternAnalysisandMachine Intell. 2:674-693.

Mamlouk AM, Sharp H, Menne KML, Hofmann UG, Martinetz T (2005) Unsupervised spike sorting with ICA and its evaluation using GENESIS simulations Neurocomputing 65(6):275-82. Mandairon R, Linster P (2009) Odor perception and olfactory bulb plasticity in adult mammals. Journal of Neurophysiology 101:2204-9.

Markram H, Tsodyks M (1996) Redistribution of synaptic ecacy between neocortical pyramidal neurons. Nature 382:807-10.

Margrie T, Schaefer A (2003) Theta Osciallation Coupled Spike Latencies Yield Computational Vigour In A Mammalian Sensory System. J Physiol 546:363-374.

Mastronarde DN (1983) Correlated firing of cat retinal ganglion cells. II. Responses of X- and Y-cells to single quantal events. J. Neurophysiol. 49(2):325-349.

Maunsell JHR, Van Essen DC (1983) Functional properties of neurons in middle temporal visual area of the Macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. Journal of Neurophysiology 49:1127-1147.

Maynard EM, Hatsopoulos NG, Ojakangas CL, Acuna BD, Sanes JN, Normann RA, Donoghue JP (1999) Neuronal interactions improve cortical population coding of

movement direction. Journal of Neuroscience 19:8083-8093.

McAlpine D, Jiang D, Palmer AR (2001) A neural code for low-frequency sound localization in mammals. Nat. Neurosci. 4 (4): 396-401.

McClurkin, JW (1991) Lateral geniculate neurons in behaving primates. II. Encoding of visual information in the temporal shape of the response. J. Neurophys. 66:794-308.

Mehring C, Rickert J, Vaadia E, Cardosa de Oliveira S, Aertsen A, Rotter S(2003) Inference of hand movements from local field potentials in monkey motor cortex. Nat. Neurosci. 6:1244-1253.

Miller J P, Jacobs G A, and Theunissen F E(1991) . Representation of sensory information in the cricket cercal sensory system. i. response properties of the primary interneurons. J Neurophysiol, 66(5):1680-1689.

Mitzdorf, U (1987) Properties of the evoked potential generators: current source-density analysis of visually evoked potentials in the cat cortex. Int. J. Neurosci. 33:33-59.

Montani F, Kohn A, Smith MA, Schultz SR (2007) The role of correlations in direction and contrast coding in the primary visual cortex. J. Neurosci. 27:2338-2348.

Montemurro MA, Rasch MJ, Murayama Y, Logothetis NK, Panzeri S (2008) Phase-of-Firing Coding of Natural Visual Stimuli in Primary Visual Cortex, Current Biology, 18(5):375-380.

Narayanan NS, Kimchi EY, and Laubach M (2005) Redundancy and synergy of neuronal ensembles in motor cortex. J Neurosci 25: 4207-4216.

Nelken I, Chechik G, Mrsic-Flogel TD , King A J Schnupp JWH (2005) Encoding stimulus information by spike numbers and mean response time in primary auditory cortex. J Comput Neurosci, 19(2):199-221.

Nicol A, Man MS, Feng JF, Mason R, Kendrick K (2003) Differential spatial activation patterns evoled by odor stimui in the rate olfactory bulb. Annual Meeting of Neuroscience.

Nicol AU, Magnusson MS, Segonds-Pichon A, Tate A, Feng JF, Kendrick KM (2005) Local and global encoding of odor stimuli by olfactory bulb neural networks. Annual Meeting of the Society for Neuroscience 2005, Program No. 476.6. Abstract

Viewer/Itinerary Planner. Washington DC:Society for Neuroscience,2005. Online

Neville KR, Haberly LB (2005) Beta and gamma oscillations in the olfactory system of the urethane-anesthetized rat. J Neurophysio 90:3921-3930.

Niedermeyer E, Lopes da Silva F (2004) Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Lippincot Williams and Wilkins.

Nirenberg S, Carcieri SM, Jacobs AL, Latham PE (2001) Retinal ganglion cells act largely as independent encoders. Nature 411: 698-701.

Nirenberg, S Latham (2003) P E Decoding neuronal spike trains: how important are correlations. Proc. Natl Acad. Sci. USA 100: 7348-7353.

Norris BJ, Weaver AL, Morris LG ,Wenning A , PA Garcy, Calabrese RL (2006) A Central Pattern Generator Producing Alternative Outputs: Temporal Pattern of Premotor Activity. J Neurophysiol 96: 309-326.

Nunez PL, Srinivasan R (1981) Electric fields of the brain: The neurophysics of EEG. Oxford University Press.

Olshausen BA, Field D J(1996) Emergence of simplecell receptive field properties by learning a sparse code for natural images. Nature 381:607-609.

Optican LM, Richmond BJ (1987)Temporal encoding of two-dimensional patterns by single units in primate inferior temporal cortex. III. Information theoretic analysis. J. Neurophysiol. 57:162-178.

Oram MW, Foldiak P, Perrett DI, Sengpiel F (1998) The Ideal Homunculus: decoding neural population signals. Trends Neurosci. 21:259-265.

Oram MW, Wiener MC, Lestienne R, Richmond BJ (1999) Stochastic nature of precisely timed spike patterns in visual system neuronal responses. J. Neurophysiol. 81:3021-3033.

Paninski L, Fellows MR, Hatsopoulos NG, Donoghue JP (2004) Spatiotemporal tuning of motor cortical neurons for hand position and velocity. J Neurophysiol 91: 515-532.

Panzeri S, Schultz SR, Treves A, Rolls ET (1999) Correlations and the encoding of information in the nervous system. Proc R Soc Lond B Biol Sci 266:1001-1012.

Panzeri S, Golledge HD, Zheng F, Tove MJ, Young MP, (2001) Objective assessment of the functional role of spike train correlations using information measures. Vis. Cogn. 8:531-547.

Panzeri S, Petersen RS, Schultz SR ,Lebedev M, Diamond M E (2001b) The role of spike timing in the coding of stimulus location in rat somatosensory cortex. Neuron 29:769-777.

Panzeri, S, Senatore R, Montemurro MA,

Pasupathy A, Connor CE (2002) Population coding of shape in area V4. Nat. Neurosci. 5 (12):1332-1338.

Patterson WR, Song YK, Bull CW, Ozden I, Deangellis AP, Lay C, McKay JL, Nurmikko AV, Donoghue JD (2004) Connors BW. A microelectrode/microelectronic hybrid device for brain Implantable neuroprosthesis applications. IEEE Transactions on biomedical engineering 51:1845-1853.

Pavlov, A, Makarov VA, Makarova I, and F Panetsos (2007) Sorting of neural spikes: When wavelet based methods outperform principal component analysis. Natural Computing, 6(3):269-281.

Pawitan Y. In all Likelihood:Statistical Modelling and Inference Using Likelihood (2001) (Oxford Univ Press, New York)

Perez-Orive J, Mazor O, Turner GC, Cassenaer S, Wilson RI, Laurent G (2002) Oscillations and sparsening of odor representations in the mushroom body. Science 297:359-65.

Perrett DI, Rolls ET, Caan W (1982) Visual neurons responsive to faces in the monkey temporal cortex. Exp Brain Res 47:329-42.

Petersen, RS, Panzeri S, Diamond M E (2001) Population coding of stimulus location in rat somatosensory cortex. Neuron 32:503-514

Petersen R, Panzeri S (2004) A Case Study of Population Coding: Stimulus Localisation in the Barrel Cortex

Petersen RS (2007) Correcting for the sampling bias problem in spike train information measures. J. Neurophysiol. 98:1064-1072

Petersen RS, Panzeri S, Diamond ME (2002) Population coding in somatosensory cortex. Curr. Opin. Neurobiol. 12 (4):441-447.

Pinato G, Midtgaard J (2003) Regulation of granule cell excitability by a low-threshold calcium spike in turtle olfactory bulb. J Neurophysiol 90:3341-3351.

Pouget A, Zhang K, Deneve S, Latham P (1998) Statistically efficient estimation using population coding. Neural Comput 10:373-401

Pouget A, Dayan P, Zemel R (2000) Information Processing With Population Codes. Nat Rev Neurosci 1:125-132.

Pouget A, Dayan P, Zemel R (2003) Inference and computation with population codes. Annu. Rev. Neurosci. 26:381-410.

Pouzat C, Mazor O, Laurent G (2002) Using noise signature to optimize spike-sorting and to assess neuronal classification quality. Journal of Neuroscience Methods 122:43-57.

Pouzat C, Delescluse M, Viot P, Diebolt J (2004) Improved spike-sorting by modelling firing statistics and burst-dependent spike amplitude attenuation: a Markov chain Monte Carlo approach. J Neurophysiol 91:2910-28.

Prut Y, Vaadia E, Bergman H, Haalman I, Hamutal S, Abeles M (1998) Spatiotemporal structure of cortical activity: Properties and behavioral relevance. J. Neurophysiol. 79:2857-2874.

Quian Quiroga R, Sakowicz O, Basar E, Schurmann M (2001) Wavelet Transform in the analysis of the frequency composition of evoked potentials. Brain Research Protocols, 8: 16-24.

Quian Quiroga R, Kraskov A, Kreuz T, Grassberger P (2002) Performance of different synchronization measures in real data:A case study on electroencephalographic signals. Phys. Rev. E, 65: 896-903.

Quian Quiroga R, Garcia H (2003) Single-trial event-related potentials with wavelet denoising. Clin. Neurophysiol., 114:376-390.

Quian Quiroga R, Nadasdy Z, Ben-Shaul Y (2004) Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput 16:1661-87.

Quian Quiroga R, Reddy L, Kreiman G, Koch C, Fried I (2005) Invariant visual representation by single neurons in the human brain. Nature 435: 1102-1107.

Quian Quiroga R, Snyder LH, Batista AP, Cui H, Andersen RA (2006) Movement intention is better predicted than attention in the posterior parietal cortex. J. Neurosci. 26:3615-3620.

Quian Quiroga (2007) Spike sorting. Scholarpedia 2(12):3583.

Quian Quiroga R, Reddy L, Koch C, Fried I (2007) Decoding visual inputs from multiple neurons in the human temporal lobe. J. Neurophysiol. 98:199-206.

Quian Quiroga R, Panzeri S. (2009) Extracting information from neuronal populations: information theory and decoding approaches. Nat Neurosci 10: 173-185.

Rao SG, Williams GV, Goldman-Rakic PS (1999) Isodirectional tuning of adjacent interneurons and pyramidal cells during working memory: evidence for microcolumnar organization in PFC. J Neurophysiol 81:1903-1916

Razak KA, Fuzessery, ZM, (2002) Functional organization of the pallid bat auditory cortex: emphasis on binaural organization. J. Neurophysiol. 87 (1):72-86.

Read JC, Cumming BG (2003) Measuring V1 Receptive Fields Despite Eye Movements in Awake Monkeys. J Neurophysiol 90:946-960,

Reich DS, Mechler F, Victor JD (2001) Independent and redundant information in nearby cortical neurons. Science 294:2566-2568.

Richmond BJ, TJ Gawne, (1998) The relationship between neuronal codes and cortical organization, in Neuronal Ensembles: Strategies for Recording and Decoding (ed. H.B. Eichenbaum and J.L. Davis, New York, Wiley-Liss), pp. 57-80.

Richmond BJ, Optican LM, Gawne TJ (1989) Neurons use multiple messages encoded in temporally modulated spike trains to represent pictures. in Seeing Contour and Colour J.J. Kulikowski C.M. Dickinson, eds., Pergamon Press, pp: 701-710, Oxford

Riehle A, Grun S, Diesmann M, Aertsen A (1997) Spike synchronization and rate modulation differentially involved in motor cortical function. Science 278:1950-1953.

Rieke F, Warland D, de Ruyter van Steveninck RR, Bialek W (1997) Spikes: Exploring the Neural Code (MIT Press, Cambridge, Massachusetts.

Reich DS, Mechler F, Victor JD (2001) Temporal coding of contrast in primary visual cortex: when,what, and why. J. Neurophysiol. 85:1039-1050.

Reinagel P. Information theory in the brain (2000) Curr. Biol. 10(15):R542-R544. Rieke D.S, Melcher F, Victor J D. (2001) Independent and redundant information in nearby cortical neurons. Science 294:2566-2568

Rieke F, Warland D, Rutyer van Steveniink R, Bailek W (1998) Spikes - Exploring the Neural Code. MIT Press, Cambidge, MA.

Riemer C, Neidhold S, Burwinkel M, Schwarz A, Schultz J,Kratzschmar J, Monning U, Baier M (2004) Gene expression profiling of scrapie-infected brain tissue.

Biochemical and Biophysical Research Communications 323:556-564.

Riesenhuber M, Poggio T (1999) Are cortical models really bound by the binding problem?. Neuron 24:87-93.

Rinberg et al. (2006) Sparse odor coding in awake behaving mice. Journal of Neuroscience 26(88):57-65.

Ritter H, and Kohonen T, (1989) Self-organizing semantic maps. Biological Cybernetics, 61:241-254.

Rolls ET. (1984) Neurons in the cortex of the temporal lobe and in the amygdala of the monkey with responses selective for faces. Hum Neurobiol 3:209-22.

Rolls ET, Treves A, Tovee MJ (1997) The representational capacity of the distributed encoding of information provided by populations of neurons in the primate temporal visual cortex. Exp Brain Res 114:149-162

Romo R, Hernandez A, Zainos A, Salinas E (2003) Correlated neuronal discharges that increase coding efficiency during perceptual discrimination. Neuron 38: 649-657. Rutishauser U, Schumand EM, Mamelakb AN, (2006) Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo. J Neurosci Meth 154:204-224.

Rutishauser U, Schumand E Mamelakb, A (2006) Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo. J. Neurosci. Meth. 154:204-224 Rutishauser U, Ross IB, Mamelak AN, Schuman EM (2010) Human memory strength is predicted by theta-frequency phase-locking of single neurons. Nature 464:903-907

Rutter J (2003) Geometry Of Curves, third ed. Chapman and Hall/CRC: Florida, pg:133-147.

Sahani M, Pezaris JS, Andersen RA (1998) On the separation of signals from neighboring cells in tetrode recordings. In: Jordan MI, Kearns MJ, Solla SA, editors. Advanced Neural Information Processing Systems, 10. Cambridge, MA: MIT Press.

Samar VJ, Swartz KP, Raghveer MR (1995) Multiresolution analysis of event-related potentials by wavelet decomposition. Brain and Cognition. 27: 398-438.

Samengo I (2002) Information loss in an optimal maximum likelihood decoding. Neural Comput. 14:771-779 Scalaidhe SP, Wilson FA, Goldman-Rakic PS (1999)

Face selective neurons during passive viewing and working memory. Nat. Neurosci. 14: 771-779

Schmidt EM (1984) Computer separation of multi-unit neuroelectric data: a review. J Neurosci Methods, vol. 12(2), pp. 95-111. Seris P, Latham PE, Pouget A (2004) Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. Nat Neurosci 7: 1129-1135.

Serruya M D, Hatsopoulos N G, Paninski L, Fellows M R Donoghue J P (2002) Instant neural control of a movement signal Nature 416:141-152.

Shadlen MN, Newsome WT (1994) Noise, neural codes and cortical organization. Curr. Opin. Neurobiol. 4:569-579. Shadlen MN, Movshon J (1994) Synchrony unbound: A critical evaluation of the temporal binding hypothesis. Neuron 24:67-77.

Shadlen MN, Britten KH, Newsom WT, Movshon JA (1996) A computational analysis of the relationship between neuronal and behavioral responses to visual motion. J. Neurosci. 16 (4): 1486-1510.

Shadlen MN, Newsome WT (1998) The variable discharge of cortical neurons: Implications for connec- tivity, computation, and information coding. J. Neurosci. 18:3870-3896.

Shahid S, Walker J. Smith LS (2010) A new spike detection algorithm for extracellular neural recordings IEEE Transactions on Biomedical Engineering, 57(4): 853-866.

Shimokawa T, Shinomoto S (2009) Estimating Instantaneous Irregularity of Neuronal Firing. Neural Computation 21: 1931-1951.

Shoham S, Fellows MR, Normann RA(2003) Robust, automatic spike sorting using mixtures of multivariate t-distributions. J Neurosci Methods 127(2):111-22.

Siapas AG, Wilson MA, (1998) Coordinated interactions between hippocampal ripples and cortical spindles during slow-wave sleep. Neuron, 21: 1123-1128.

Simoncelli E P, Olshausen BA (2001) Natural image statistics and neural representation. Annu. Rev. Neurosci. 24: 1193-1216.

Singer W, Gray C, Engel A, Konig P (1988) Spatio-temporal distribution of stimulus-specific oscillations in the cat viausl cortex ii: Global interactions. Soc. Neurosci. Abstr. 14:28-39. Singer W (1993) Synchronization of cortical activity and its puta-

tive role in information processing and learning. Annu. Rev. Physiol. 55:349-374.

Singer W, Gray CM (1995) Visual feature integration and the temporal correlation hypthesis. Ann Rev Neurosci 18: 555-586.

Singer W (1999) Neuronal synchrony: a versatile code for the definition of relations? Neuron 24:29-45.

Sirota A, Csicsvari J, Buhl D, Buzsaki G (2003) Communication between neocortex and hippocampus during sleep in rodents. Proc. Natl. Acad. Sci. USA, 100: 2065-2069.

Smith LS, Mtetwa N (2007) A tool for synthesizing spike trains with realistic interference. J. Neurosci. Meth. 159:170-180

Sompolinsky H, Yoon H, Kang K, Shamir M (2001) Population coding in neuronal systems with correlated noise. Phys. Rev. E 64 (5 Pt 1):051904.

Sparks D L, Holland R, Guthrie B (1976) Size and distributionof movement fields in the monkey superior colliculus. Brain Res. 113:21-34

Starka E, Abeles M(2009) Unbiased estimation of precise temporal correlations between spike trains. J. Neurosci. Meth. 179 : 90-100

Stein R, Gossen E, Jones K (2005) Neuronal variability: noise or part of the signal? Nat Rev Neuros 6:389-397

Stopfer M, Jayaraman V, Laurent G (2003) Intensity versus identity coding in an olfactory system. Neuron 39: 991-1004

Stuart L, Walter M, Borisyuk R.(2002) Visualisation of synchronous firing in multidimensional spike trains. Biosystems 67:265-279

Soucy P. (2009) Precision and diversity in an odor map on the olfactory bulb. Nat. Neurosci. 12:210-20.

Takahashi S, Anzai Y , Sakurai Y (2003) Automatic sorting for multineuronal activity recorded with tetrodes in the presence of overlapping spikes, J Neurophys. 89:2245-2258. Takahashi S, Sakurai Y (2005) Real-time and automatic sorting of multi-neuronal activity for sub-millisecond interactions in vivo. Neuroscience 134 :301-15.

Tate AJ, Nicol AU, Fischer H, Segonds-Pichon A, Feng JF, Magnusson MS, Kendrick KM(2005) Lateralised local and global encoding of face stimuli by neural networks in the temporal cortex. Annual Meeting of the Society for Neuroscience 2005, Pro-

gram No 362.1.005 Abstract Viewer/Itinerary Planner. Washington DC: Society for Neuroscience, 2005. Online.

Taylor DM, Tillery SI, Schwartz AB (2002) Direct Cortical Control Of 3D neuroprosthetic Devices. Science 296:1892-1832

Thakur P H, Lu H, Hsiao S S and Johnson K O (2007) Automated optimal detection and classification of neural action potentials in extra-cellular recordings J. Neurosci. Methods 162: 364-376.

Todd JJ, Marois R (2004) Capacity limit of visual short-term memory in human posterior parietal cortex. Nature 428:751-754.

Treves A, Panzeri S, Rolls ET, Booth M, Wakeman EA (1999) Firing rate distributions and efficiency of information transmission of inferior temporal cortex neurons to natural visual stimuli. Neural Comput. 11:611-641.

Uchida N, Mainen ZF (2003) Speed and accuracy of olfactory discrimination in the rat. Nat Neuro. 2003:1224-1229. van Kan PL, Scobey RP, Gabor AJ (1985) Response covariance in cat visual cortex. Exp. Brain. Res. 60 (3): 559-563.

Vargas-Irwin C , Donoghue J P (2007) Automated spike sorting using density grid contour clustering and subtractive waveform decomposition J. Neurosci. Methods 164: 1-18.

Ventura V, Cai C, Kass RE (2005) Statistical Assessment of Time-Varying Dependency Between Two Neurons. J Neurophysiol. 94: 2940-2947. Victor JD (1994) Population encoding of spatial frequency, orientation, and color in Macaque VI, J. Neurophysiol. 72: 2151-2166 .

Victor JD, Purpura KP (1996) Nature and precision of temporal coding in visual cortex: a metric-space analysis. J. Neurophysiol. 76:1310-1326.

Victor JD (2006) Approaches to information-theoretic analysis of neural activity. Biol. Theory 1: 302-316.

Vogels R, Spileers W, Orban GA (1989) The response variability of striate cortical neurons in the behaving monkey. Exp. Brain Res. 77 (2): 432-436.

Vogels R (1990) Population coding of stimulus orientation by striate cortical cells. Biol. Cybern. 64 (1): 25-31.

Volgushev M, Christakova MWS (1998) Modification of discharge patterns of neocortical neurons by induced oscillations of the membrane potential. Neuroscience

83:15-25.

von der Malsburg C (1981) The correlation theory of brain function Internal report 81-2, Max-Planck-Institute for Biophysical Chemistry, Gottingen, FRG.

Warwick K, Gasson M, Hutt B, Goodhew I, Kyberd P, Andrews B, Teddy P, Shad A (2004) The application of implant technology for cybernetic systems. Archives of Neurology 60:1369-1373.

Wehr M, Pezaris J, Sahani M (1999) Simultaneous paired intracellular and tetrode recordings for evaluating the performance of spike sorting algorithms. Neurocomputing. 26:1061-1068

Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, Nicolelis MA. (2000) Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. Nature 408: 361-365.

Wilke S D, Eurich C W (2002) Represenational accuracy of stochastic neural populations. Neural Comput. 14:155-189.

Williams P, Li S, Feng JF, Wu S (2005) Scaling the kernel function to improve performance of the support vector machine. LECT NOTES COMPUT SC 3496:831-836.

Wilson MA, McNaughton BL (1993) Dynamics of the hippocampal ensemble code for space. Science. 261 (5124): 1055-1058.

Wood F, Black M J, Vargas-Irwin C, Fellows M and Donoghue J P (2004) On the variability of manual spike sorting IEEE Trans. Biomed. Eng. 51 :912-8

Wu S, Amari S, Nakahara H (2002) Population coding and decoding in a neural field: a computational study. Neural Comput. 14 (5): 999-1026.

Yamane S, Kaji S, Kawano K (1988) What facial features activate face neurons in the inferotemporal cortex of the monkey? Exp Brain Res 73:209-14.

Young MP, Yamane S (1992) Sparse population coding of faces in the inferotemporal cortex. Science 256 (5061): 1327-1331.

Young H, Baum R, Cremerius U, et al. (1999) Measurement of clinical and subclinical tumour response using [18F]-fluorodeoxyglucose and positron emission tomography: review and 1999 EORTC recommendations. European Journal of Cancer 35 (13):1773-1782.

Zimmerman P, Donald W (1997) A Note on Interpretation of the Paired-Samples t Test. Journal of Educational and Behavioral Statistics. 22 (3): 349-360.

Zhang P, Wu J, Zhou Y, Liang P, Yuan J (2004) Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. J. Neurosci. Meth. 135:55-65.

Zohary E, Shadlen M N, Newsome WT (1994) Correlated neuronal discharge rate and its implications for psychophysical performance. Nature. 370: 140-143.

Zouridakis G, Tam DC (2000) Identification of reliable spike templates in multiunit extracellular recordings using fuzzy clustering. Comput Methods Prog Biomed. 61(2):91-8.

Appendix A

Appendix To MEANOVA Chapter: Groups Attained From Biological Data

Numbers inside (·) represent the electrode number (see fig. 6.1 in chapter 6 for the electrode layout). The numbers on the left hand side of the equality represent the group number, i.e. the x-axis (for an example see fig. 6.11). Each group of electrodes contains a connected set of electrodes. In other words, each electrode we added to a subset was adjacent to, i.e. was an MEA neighbour of, one electrode already in the subset. We constructed the subsets in this way so the MEANOVA scores produced by small and large subsets represented the changes in spiking activity of small (local) and large (global) neuronal areas respectively. This is discussed in more detail in chapter 6.

Groups Attained From Real Data:

1=(05) 2=(15) 3=(16) 4=(25) 5=(26) 6=(35) 7=(36) 8=(43) 9=(44) 10=(45) 11=(46) 12=(05 15) 13=(15 16) 14=(15 25) 15=(16 26) 16=(25 26) 17=(25 35) 18=(26 36) 19=(35 36) 20=(43 44) 21=(35 45) 22=(44 45) 23=(36 46) 24=(45 46) 25=(05 15 16) 26=(05 15 25) 27=(15 16 25) 28=(15 16 26) 29=(15 25 26) 30=(16 25 26) 31=(15 25 35) 32=(16 26 36) 33=(25 26 35) 34=(25 26 36) 35=(25 35 36) 36=(26 35 36) 37=(25 35 45) 38=(35 44 45) 39=(35 45 46) 40=(43 44 45) 41=(26 36 46) 42=(35 36 45) 43=(35 36 46) 44=(44 45 46) 45=(05 15 16 25) 46=(05 15 16 26) 47=(05 15 25 26) 48=(05 15 25 35) 49=(15 16 25

26) 50=(15 16 25 35) 51=(15 25 35 36) 52=(16 26 35 36) 53=(15 25 35 45) 54=(16 25 26 35) 55=(15 16 26 36) 56=(15 25 26 35) 57=(15 25 26 36) 58=(16 25 26 36) 59=(25 35 44 45) 60=(35 43 44 45) 61=(25 35 45 46) 62=(26 35 36 45) 63=(35 36 44 45) 64=(16 26 36 46) 65=(25 26 35 36) 66=(25 26 35 45) 67=(25 26 36 46) 68=(25 35 36 45) 69=(25 35 36 46) 70=(26 35 36 46) 71=(35 36 45 46) 72=(35 44 45 46) 73=(43 44 45 46) 74=(05 15 16 25 26) 75=(05 15 16 25 35) 76=(05 15 25 26 35) 77=(05 15 25 26 36) 78=(05 15 25 35 36) 79=(15 16 26 35 36) 80=(15 25 26 35 36) 81=(16 25 26 35 45) 82=(16 26 35 36 45) 83=(25 35 36 44 45) 84=(26 35 36 44 45) 85=(25 35 43 44 45) 86=(25 35 44 45 46) 87=(26 36 44 45 46) 88=(35 36 43 44 45) 89=(16 25 26 36 46) 90=(15 25 35 36 45) 91=(16 26 35 36 46) 92=(15 25 35 44 45) 93=(15 25 35 36 46) 94=(15 25 35 45 46) 95=(16 25 26 35 36) 96=(05 15 25 35 45) 97=(15 16 25 26 35) 98=(05 15 16 26 36) 99=(15 16 25 26 36) 100=(15 16 25 35 36) 101=(15 16 25 35 45) 102=(15 25 26 35 45) 103=(25 26 35 36 45) 104=(25 26 35 44 45) 15=(15 16 26 36 46) 106=(15 25 26 36 46) 107=(25 26 35 36 46) 108=(25 26 35 45 46) 109=(25 35 36 45 46) 110=(26 35 36 45 46) 111=(35 36 44 45 46) 112=(35 43 44 45 46) 113=(05 15 16 25 26 35) 114=(05 15 16 25 26 36) 115=(05 15 16 25 35 36) 116=(05 15 16 26 35 36) 117=(05 15 25 35 44 45) 118=(05 15 25 35 45 46) 119=(15 16 25 35 44 45) 120=(15 16 25 35 45 46) 121=(16 26 35 36 44 45) 122=(15 25 35 43 44 45) 123=(16 26 36 44 45 46) 124=(25 26 35 43 44 45) 125=(05 15 16 26 36 46) 126=(05 15 25 26 35 36) 127=(05 15 25 26 35 45) 128=(05 15 25 26 36 46) 129=(05 15 25 35 36 46) 130=(15 16 25 26 35 36) 131=(05 15 16 25 35 45) 132=(15 16 25 26 35 45) 133=(15 16 25 35 36 46) 134=(15 16 26 35 36 45) 135=(15 25 26 35 36 45) 136=(15 25 35 36 44 45) 137=(25 26 36 44 45 46) 138=(25 35 43 44 45 46) 139=(26 35 36 43 44 45) 140=(16 26 35 36 45 46) 141=(25 35 36 43 44 45) 142=(15 25 35 36 45 46) 143=(16 25 26 35 36 45) 144=(16 25 26 35 44 45) 145=(25 26 35 36 44 45) 146=(16 25 26 35 36 46) 147=(15 25 26 35 44 45) 148=(15 16 25 26 36 46) 149=(05 15 25 35 36 45) 150=(15 16 25 35 36 45) 151=(15 16 26 35 36 46) 152=(15 25 26 35 36 46) 153=(15 25 26 35 45 46) 154=(16 25 26 35 45 46) 155=(25 26 35 36 45 46) 156=(15 25 35 44 45 46) 157=(25 26 35 44 45 46) 158=(25 35 36 44 45 46) 159=(26 35 36 44 45 46) 160=(26 36 43 44 45 46) 161=(35 36 43 44 45 46) 162=(05 15 16 25 26 35 36) 163=(05 15 16 25 26 35 45) 164=(05 15 16 25 35 44 45) 165=(05 15 16 25 35 45 46) 166=(05 15 25 26 35 44 45) 167=(05 15 25 26 35 45 46) 168=(05 15 25 35 36 44 45) 169=(15 16 25 35 36 44 45) 170=(15 16 26 35 36 44 45) 171=(05 15 25 35 43 44 45) 172=(05 15 25 35 44 45 46) 173=(15 16 25 35 43 44 45) 174=(15 16 25 35 44 45 46) 175=(15 16 26 36 44 45 46) 176=(15 25 26 35 43 44 45) 177=(15 25 26 36 44 45 46) 178=(16

25 26 36 44 45 46) 179=(15 25 35 36 43 44 45) 180=(16 25 26 35 43 44 45) 181=(16 26 35 36 43 44 45) 182=(05 15 16 25 26 36 46) 183=(05 15 16 25 35 36 45) 184=(05 15 16 25 35 36 46) 185=(05 15 16 26 35 36 45) 186=(05 15 16 26 35 36 46) 187=(05 15 25 26 35 36 45) 188=(05 15 25 26 35 36 46) 189=(05 15 25 35 36 45 46) 190=(15 16 25 26 35 36 45) 191=(15 16 25 26 35 44 45) 192=(15 16 25 35 36 45 46) 193=(16 26 35 36 44 45 46) 194=(15 25 35 43 44 45 46) 195=(16 26 36 43 44 45 46) 196=(25 26 35 36 43 44 45) 197=(15 16 26 35 36 45 46) 198=(15 25 26 35 36 44 45) 199=(15 25 26 35 36 45 46) 200=(15 25 35 36 44 45 46) 201=(16 25 26 35 36 44 45) 202=(15 16 25 26 35 36 46) 203=(15 16 25 26 35 45 46) 204=(16 25 26 35 36 45 46) 25=(15 25 26 35 44 45 46) 206=(16 25 26 35 44 45 46) 207=(25 26 35 36 44 45 46) 208=(25 26 35 43 44 45 46) 209=(25 26 36 43 44 45 46) 210=(25 35 36 43 44 45 46) 211=(26 35 36 43 44 45 46) 212=(05 15 16 25 26 35 36 45) 213=(05 15 16 25 26 35 44 45) 214=(05 15 16 25 26 35 45 46) 215=(05 15 16 25 35 36 44 45) 216=(05 15 16 26 35 36 44 45) 217=(05 15 25 26 35 36 44 45) 218=(05 15 16 25 35 43 44 45) 219=(05 15 16 25 35 44 45 46) 220=(05 15 16 26 36 44 45 46) 221=(05 15 25 26 35 43 44 45) 222=(05 15 25 26 35 44 45 46) 223=(05 15 25 26 36 44 45 46) 224=(15 16 25 26 36 44 45 46) 225=(05 15 25 35 36 43 44 45) 226=(05 15 25 35 36 44 45 46) 227=(15 16 25 35 36 44 45 46) 228=(15 16 26 35 36 44 45 46) 229=(05 15 25 35 43 44 45 46) 230=(15 16 25 26 35 43 44 45) 231=(15 16 25 35 36 43 44 45) 232=(15 16 25 35 43 44 45 46) 233=(15 16 26 35 36 43 44 45) 234=(15 16 26 36 43 44 45 46) 235=(15 25 26 35 36 43 44 45) 236=(15 25 26 36 43 44 45 46) 237=(16 25 26 36 43 44 45 46) 238=(15 25 35 36 43 44 45 46) 239=(16 25 26 35 36 43 44 45) 240=(05 15 16 25 26 35 36 46) 241=(05 15 16 25 35 36 45 46) 242=(05 15 16 26 35 36 45 46) 243=(05 15 25 26 35 36 45 46) 244=(15 16 25 26 35 36 44 45) 245=(15 16 25 26 35 36 45 46) 246=(15 16 25 26 35 44 45 46) 247=(15 25 26 35 36 44 45 46) 248=(15 25 26 35 43 44 45 46) 249=(16 25 26 35 36 44 45 46) 250=(16 25 26 35 43 44 45 46) 251=(16 26 35 36 43 44 45 46) 252=(25 26 35 36 43 44 45 46) 253=(05 15 16 25 26 35 36 44 45) 254=(05 15 16 25 26 35 43 44 45) 255=(05 15 16 25 26 35 44 45 46) 256=(05 15 16 25 26 36 44 45 46) 257=(05 15 16 25 35 36 43 44 45) 258=(05 15 16 25 35 36 44 45 46) 259=(05 15 16 26 35 36 44 45 46) 260=(05 15 16 26 36 43 44 45 46) 261=(05 15 25 26 35 36 44 45 46) 262=(05 15 16 25 35 43 44 45 46) 263=(05 15 16 26 35 36 43 44 45) 264=(05 15 25 26 35 36 43 44 45) 265=(05 15 25 26 35 43 44 45 46) 266=(05 15 25 26 36 43 44 45 46) 267=(15 16 25 26 36 43 44 45 46) 268=(05 15 25 35 36 43 44 45 46) 269=(15 16 25 26 35 36 43 44 45) 270=(05 15 16 25 26 35 36 45 46) 271=(15 16 25 26 35 36 44 45 46) 272=(15 16 25 26 35 43 44 45 46) 273=(15 16 25 35 36 43 44 45 46) 274=(15 16 26 35 36 43 44 45 46) 275=(15 25 26 35 36 43 44

45 46) 276=(16 25 26 35 36 43 44 45 46) 277=(05 15 16 25 26 35 36 43 44 45) 278=(05 15 16 25 26 35 36 44 45 46) 279=(05 15 16 25 26 35 43 44 45 46) 280=(05 15 16 25 26 36 43 44 45 46) 281=(05 15 16 25 35 36 43 44 45 46) 282=(05 15 16 26 35 36 43 44 45 46) 283=(05 15 25 26 35 36 43 44 45 46) 284=(15 16 25 26 35 36 43 44 45 46) 285=(05 15 16 25 26 35 36 43 44 45 46)

Appendix B

Appendix To MEANOVA Chapter: Groups Attained From Simulated Data

Numbers inside $(\cdot)$ represents the cell number in the 4×4 simulation grid (see fig. 6.1 for grid layout). The numbers on the left hand side of the equality represent the group number, i.e. the x-axis, for example see fig. 6.3. Each group of cells contains a connected set of cells. In other words, each cell we added to a subset was adjacent to, i.e. was a cell neighbour of, one of the cells already in the subset. We constructed the subsets in this way so the MEANOVA scores produced by small and large subsets represented the changes in spiking activity of small (local) and large (global) neuronal areas respectively. This is discussed in more detail in chapter 6.

Groups Attained From The Simulated Data:

1=(01) 2=(02) 3=(03) 4=(04) 5=(11) 6=(13) 7=(21) 8=(22) 9=(23) 10=(24) 11=(31) 12=(33) 13=(01 02) 14=(01 11) 15=(02 03) 16=(03 04) 17=(03 13) 18=(11 21) 19=(21 22) 20=(13 23) 21=(22 23) 22=(23 24) 23=(21 31) 24=(23 33) 25=(01 02 03) 26=(01 11 21) 27=(02 03 04) 28=(02 03 13) 29=(03 04 13) 30=(11 21 22) 31=(03 13 23) 32=(13 22 23) 33=(13 23 24) 34=(13 23 33) 35=(21 22 23) 36=(22 23 24) 37=(11 21 31) 38=(21 22 31) 39=(22 23 33) 40=(23 24 33) 41=(01 02 03 04) 42=(01 02 03 13) 43=(01 11 21 22) 44=(03 13 22 23) 45=(11 21 22 23) 46=(13 21 22 23) 47=(03 13 23 24) 48=(21 22 23 24) 49=(01 11 21 31) 50=(02

03 04 13) 51=(02 03 13 23) 52=(03 04 13 23) 53=(21 22 23 31) 54=(03 13 23 33) 55=(13 22 23 24) 56=(11 21 22 31) 57=(13 22 23 33) 58=(21 22 23 33) 59=(13 23 24 33) 60=(22 23 24 33) 61=(01 02 03 04 13) 62=(01 02 03 11 21) 63=(01 02 03 13 23) 64=(01 11 21 22 23) 65=(03 13 21 22 23) 66=(11 13 21 22 23) 67=(03 13 22 23 24) 68=(11 21 22 23 24) 69=(13 21 22 23 24) 70=(01 11 21 22 31) 71=(02 03 04 13 23) 72=(02 03 13 22 23) 73=(03 04 13 22 23) 74=(02 03 13 23 24) 75=(03 04 13 23 24) 76=(11 21 22 23 31) 77=(13 21 22 23 31) 78=(13 21 22 23 33) 79=(21 22 23 24 31) 80=(02 03 13 23 33) 81=(03 04 13 23 33) 82=(03 13 22 23 33) 83=(11 21 22 23 33) 84=(03 13 23 24 33) 85=(13 22 23 24 33) 86=(21 22 23 24 33) 87=(21 22 23 31 33) 88=(01 02 03 04 11 21) 89=(01 02 03 11 13 21) 90=(01 02 03 11 21 22) 91=(01 02 03 04 13 23) 92=(01 02 03 13 22 23) 93=(01 02 03 13 23 24) 94=(01 02 03 13 23 33) 95=(01 11 13 21 22 23) 96=(01 11 21 22 23 24) 97=(01 11 21 22 23 33) 98=(02 03 13 21 22 23) 99=(03 04 13 21 22 23) 100=(03 13 21 22 23 24) 101=(01 02 03 11 21 31) 102=(01 11 21 22 23 31) 103=(02 03 04 13 22 23) 104=(03 11 13 21 22 23) 105=(02 03 04 13 23 24) 106=(02 03 13 22 23 24) 107=(03 04 13 22 23 24) 108=(03 13 21 22 23 31) 109=(03 13 21 22 23 33) 110=(11 13 21 22 23 24) 111=(11 13 21 22 23 31) 112=(11 21 22 23 24 31) 113=(13 21 22 23 24 31) 114=(02 03 04 13 23 33) 115=(02 03 13 22 23 33) 116=(03 04 13 22 23 33) 117=(11 13 21 22 23 33) 118=(02 03 13 23 24 33) 119=(03 04 13 23 24 33) 120=(03 13 22 23 24 33) 121=(11 21 22 23 24 33) 122=(13 21 22 23 24 33) 123=(11 21 22 23 31 33) 124=(13 21 22 23 31 33) 125=(21 22 23 24 31 33) 126=(01 02 03 04 11 13 21) 127=(01 02 03 04 11 21 22) 128=(01 02 03 04 13 22 23) 129=(01 02 03 04 13 23 24) 130=(01 02 03 04 13 23 33) 131=(01 02 03 11 21 22 23) 132=(01 02 03 13 21 22 23) 133=(01 02 03 13 22 23 24) 134=(01 02 03 13 22 23 33) 135=(01 02 03 13 23 24 33) 136=(01 11 13 21 22 23 24) 137=(01 11 13 21 22 23 31) 138=(01 11 13 21 22 23 33) 139=(01 11 21 22 23 24 31) 140=(01 11 21 22 23 24 33) 141=(01 11 21 22 23 31 33) 142=(02 03 04 13 21 22 23) 143=(02 03 13 21 22 23 24) 144=(03 04 13 21 22 23 24) 145=(01 02 03 04 11 21 31) 146=(01 02 03 11 13 21 22) 147=(01 02 03 11 13 21 23) 148=(01 02 03 11 13 21 31) 149=(01 02 03 11 21 22 31) 150=(01 03 11 13 21 22 23) 151=(02 03 11 13 21 22 23) 152=(02 03 13 21 22 23 31) 153=(02 03 13 21 22 23 33) 154=(03 04 11 13 21 22 23) 155=(02 03 04 13 22 23 24) 156=(03 04 13 21 22 23 31) 157=(03 04 13 21 22 23 33) 158=(03 11 13 21 22 23 24) 159=(03 11 13 21 22 23 31) 160=(03 13 21 22 23 24 31) 161=(03 13 21 22 23 24 33) 162=(03 13 21 22 23 31 33) 163=(11 13 21 22 23 24 31) 164=(02 03 04 13 22 23 33) 165=(03 11 13 21 22 23 33) 166=(02 03 04 13 23 24 33) 167=(02 03 13 22 23 24 33) 168=(03 04 13 22 23 24 33) 169=(11 13 21 22 23 24 33) 170=(11 13 21 22 23 31 33) 171=(11 21 22 23 24 31 33) 172=(13 21 22 23 24 31 33) 173=(01 02

03 04 11 13 21 22) 174=(01 02 03 04 11 13 21 23) 175=(01 02 03 04 11 21 22 23) 176=(01 02 03 04 13 21 22 23) 177=(01 03 04 11 13 21 22 23) 178=(01 02 03 13 21 22 23 31) 179=(01 02 03 11 13 21 23 33) 180=(01 02 03 04 13 23 24 33) 181=(01 02 03 11 13 21 23 24) 182=(01 02 03 04 13 22 23 24) 183=(01 02 03 04 13 22 23 33) 184=(01 02 03 11 21 22 23 24) 185=(01 02 03 11 21 22 23 33) 186=(01 02 03 13 21 22 23 24) 187=(01 02 03 13 21 22 23 33) 188=(01 03 11 13 21 22 23 33) 189=(01 02 03 13 22 23 24 33) 190=(01 03 11 13 21 22 23 24) 191=(01 11 13 21 22 23 24 31) 192=(01 11 13 21 22 23 24 33) 193=(01 11 13 21 22 23 31 33) 194=(01 11 21 22 23 24 31 33) 195=(02 03 04 13 21 22 23 24) 196=(01 02 03 04 11 13 21 31) 197=(01 02 03 04 11 21 22 31) 198=(01 02 03 11 13 21 22 23) 199=(01 02 03 11 13 21 22 31) 200=(01 02 03 11 13 21 23 31) 201=(01 02 03 11 21 22 23 31) 202=(02 03 04 13 21 22 23 31) 203=(02 03 04 13 21 22 23 33) 204=(01 03 11 13 21 22 23 31) 205=(02 03 04 11 13 21 22 23) 206=(02 03 11 13 21 22 23 24) 207=(02 03 11 13 21 22 23 31) 208=(02 03 13 21 22 23 24 31) 209=(02 03 13 21 22 23 24 33) 210=(03 04 13 21 22 23 24 31) 211=(03 04 13 21 22 23 24 33) 212=(02 03 13 21 22 23 31 33) 213=(03 04 11 13 21 22 23 24) 214=(03 04 11 13 21 22 23 31) 215=(03 04 13 21 22 23 31 33) 216=(03 11 13 21 22 23 24 31) 217=(02 03 11 13 21 22 23 33) 218=(03 04 11 13 21 22 23 33) 219=(02 03 04 13 22 23 24 33) 220=(03 11 13 21 22 23 24 33) 221=(03 11 13 21 22 23 31 33) 222=(03 13 21 22 23 24 31 33) 223=(11 13 21 22 23 24 31 33) 224=(01 02 03 04 11 13 21 22 23) 225=(01 02 03 04 11 13 21 23 24) 226=(01 03 04 11 13 21 22 23 24) 227=(01 02 03 04 13 21 22 23 31) 228=(01 03 04 11 13 21 22 23 31) 229=(01 03 04 11 13 21 22 23 33) 230=(01 02 03 13 21 22 23 24 31) 231=(01 02 03 04 11 13 21 23 33) 232=(01 02 03 04 11 21 22 23 24) 233=(01 02 03 11 13 21 23 24 31) 234=(01 02 03 04 11 21 22 23 33) 235=(01 02 03 04 13 21 22 23 24) 236=(01 02 03 04 13 21 22 23 33) 237=(01 02 03 11 13 21 22 23 33) 238=(01 02 03 11 13 21 23 24 33) 239=(01 02 03 11 13 21 23 31 33) 240=(01 02 03 04 13 22 23 24 33) 241=(01 02 03 11 13 21 22 23 24) 242=(01 02 03 04 11 13 21 22 31) 243=(01 02 03 04 11 13 21 23 31) 244=(01 02 03 04 11 21 22 23 31) 245=(01 02 03 11 13 21 22 23 31) 246=(01 02 03 11 21 22 23 24 31) 247=(01 02 03 11 21 22 23 24 33) 248=(01 02 03 11 21 22 23 31 33) 249=(01 02 03 13 21 22 23 24 33) 250=(01 02 03 13 21 22 23 31 33) 251=(01 03 11 13 21 22 23 24 31) 252=(01 03 11 13 21 22 23 24 33) 253=(01 03 11 13 21 22 23 31 33) 254=(01 11 13 21 22 23 24 31 33) 255=(02 03 04 11 13 21 22 23 24) 256=(02 03 04 11 13 21 22 23 31) 257=(02 03 04 13 21 22 23 24 31) 258=(02 03 04 13 21 22 23 24 33) 259=(02 03 04 13 21 22 23 31 33) 260=(02 03 11 13 21 22 23 24 31) 261=(02 03 13 21 22 23 24 31 33) 262=(03 04 11 13 21 22 23 24 31) 263=(02 03 04 11 13 21 22 23 33) 264=(02 03 11 13 21 22 23 24 33) 265=(02 03 11 13 21 22 23 31 33) 266=(03 04 11 13 21 22 23 24 33)

267=(03 04 11 13 21 22 23 31 33) 268=(03 04 13 21 22 23 24 31 33) 269=(03 11 13 21 22 23 24 31 33) 270=(1 2 3 4 11 13 21 22 23 24) 271=(01 02 03 04 11 13 21 22 23 31) 272=(01 02 03 04 11 13 21 23 24 31) 273=(01 02 03 04 13 21 22 23 24 31) 274=(01 03 04 11 13 21 22 23 24 31) 275=(01 02 03 04 11 13 21 22 23 33) 276=(01 02 03 04 11 13 21 23 24 33) 277=(01 03 04 11 13 21 22 23 24 33) 278=(01 02 03 04 11 13 21 23 31 33) 279=(01 02 03 04 11 21 22 23 24 31) 280=(01 02 03 04 11 21 22 23 24 33) 281=(01 02 03 04 11 21 22 23 31 33) 282=(01 02 03 04 13 21 22 23 24 33) 283=(01 02 03 04 13 21 22 23 31 33) 284=(01 02 03 11 13 21 22 23 24 31) 285=(01 02 03 11 13 21 22 23 24 33) 286=(01 02 03 11 13 21 22 23 31 33) 287=(01 03 04 11 13 21 22 23 31 33) 288=(01 02 03 11 13 21 23 24 31 33) 289=(01 02 03 11 21 22 23 24 31 33) 290=(01 02 03 13 21 22 23 24 31 33) 291=(01 03 11 13 21 22 23 24 31 33) 292=(02 03 04 11 13 21 22 23 24 31) 293=(02 03 04 11 13 21 22 23 24 33) 294=(02 03 04 11 13 21 22 23 31 33) 295=(02 03 04 13 21 22 23 24 31 33) 296=(02 03 11 13 21 22 23 24 31 33) 297=(03 04 11 13 21 22 23 24 31 33) 298=(01 02 03 04 11 13 21 22 23 24 31) 299=(01 02 03 04 11 13 21 22 23 24 33) 300=(01 02 03 04 11 13 21 22 23 31 33) 301=(01 02 03 04 11 13 21 23 24 31 33) 302=(01 02 03 04 11 21 22 23 24 31 33) 303=(01 02 03 04 13 21 22 23 24 31 33) 304=(01 02 03 11 13 21 22 23 24 31 33) 305=(01 03 04 11 13 21 22 23 24 31 33) 306=(02 03 04 11 13 21 22 23 24 31 33) 307=(01 02 03 04 11 13 21 22 23 24 31 33)

Appendix C

Appendix To MEANOVA Chapter: MEANOVA Fluctuations

Here we want to theoretically show that the fluctuations will become smaller as observed in MEANOVA results.

Let us look at $S_2(\mathcal{S})$,

$$S_2(\mathcal{S}) = -\left| IJ(M-1) - \frac{|\mathcal{S}| + 1 - (J-1)}{2} \right| \ln \Lambda_2 / \chi^2_{(J-1)|\mathcal{S}|}(\alpha)$$

We could assume that $IJ(M-1)/\chi^2_{(J-1)|\mathcal{S}|}(\alpha)$ is a constant when $|\mathcal{S}|$ is large. It is easily seen that

$$\lim_{|\mathcal{S}|\to\infty} \frac{|\mathcal{S}|}{\chi^2_{(J-1)|\mathcal{S}|}(\alpha)}$$

is a constant. For example, when $J = 2$, $\alpha = 0.05$, the constant is one. To assess the existence of limit, we need to consider the limit of Λ_2, which is more mathematically involved.

Let us assume that S_r is a random, symmetric matrix. From a general results in random matrix (the Quarter circle law)(Cohen et al., 1984; Brody et al., 2003), we know that the asymptotic eigenvalue density function of S_r converges in probability to

$$P_{S_r}(x) = \frac{1}{2\pi}\sqrt{\frac{4-x}{x}} \qquad 0 < x < 4$$

We know that

$$|S_r| = \lambda_1 \cdots \lambda_{|\mathcal{S}|}$$

Hence

$$\frac{\log|S_r|}{|\mathcal{S}|}\frac{\left[\log\lambda_1+\cdots+\log\lambda_{|\mathcal{S}|}\right]}{|\mathcal{S}|}\to E\log\lambda_1$$

where λ_1 is distributed according to P_{S_r}. We then have

$$\log|S_r|\sim|\mathcal{S}|E\log\lambda_1$$

Similarly, for the matrix S_2+S_r, we conclude that

$$\log|S_r+S_2|\sim|\mathcal{S}|\log E\bar{\lambda}_1$$

where $\bar{\lambda}_1$ is the eigenvalue of the matrix S_r+S_2.

Combining all arguments above, we assert that the limit exists and so the fluctuation will become smaller and smaller.