

Optical Studies of Neural Networks and Behaviour

A High-Throughput Pipeline for the Simultaneous Recording and Analysis of Neural Activity and Behaviour in *Hydra Vulgaris*



Noah Telerman

2198991

1st Academic Supervisor: Professor Thomas Franke

2nd Academic Supervisor: Dr. Alasdair Clark

MEng Biomedical Engineering

University of Glasgow

January 2020

Abstract

Although long thought to be the functional unit of the nervous system, the study of individual neurons has not been able to address some of our key questions about the way our brains encode information. As more research is being directed towards the study of large populations of neurons, it is becoming clear that networks (or ensembles) of neurons may hold some of the answers.

By utilising calcium imaging techniques and a custom analysis tool-kit, large datasets of both single-cell neural activity and behavioural recordings were generated for the model organism *Hydra Vulgaris*. These recordings allowed for the detection and analysis of neural ensembles, and novel deep learning methods were used to investigate the link between neural activity and behaviour.

This report documents the design and implementation of an automated, high-throughput analysis pipeline, capable of recording the activity of most of the neurons in a *Hydra* while simultaneously tracking the animals behaviour. This involved the creation of custom processing software to allow the tracking of individual neurons, extraction of fluorescence signals, processing, and correlation of neurons into ensembles. Along with this, various imaging methods, transgenic strains of *Hydra*, and processing techniques were evaluated to select the optimum methodology. The developed pipeline is also tested on recordings of behaving *Hydra* and the results used to investigate the role of neural ensembles in behaviour.

Acknowledgements

I would first like to thank Rafael Yuste supervising this project and providing direction, advice, and guidance throughout its course.

I am also especially grateful to Alison Hanson, who was instrumental in motivating this project and provided a constant source of advice and discussion on all of its aspects.

I would like to thank Eduardo Soler, Wataru Yamamoto, Jonathan Lovas, and Robert Steele for their support and assistance on all things Hydra.

I am also grateful to my University of Glasgow supervisors, Thomas Franke and Alasdair Clark, for their input and suggestions.

List of Figures

2.1	Anatomy of mammalian neurons. From Molecular Cell Biology, 5th edition. by H. Lodish et al. [4]	4
2.2	Optical path of a typical fluorescence microscope. Figure from M. J. Sanderson et al [10]	6
2.3	The anatomy of a Hydra. Image from U. Technau and R. Steele [16]	8
2.4	The nervous system of Hydra. Image from Encyclopaedia Britannica [18]	8
3.1	Diagram depicting the preparation of Hydra for imaging neural activity. Figure from C. Dupre and R. Yuste [3]	13
3.2	Diagram of a Yokogawa spinning disk confocal scanner unit. Image from M. Nelson et al. [34]	17
4.1	Information flow through the developed Python script. Clockwise: raw video with tracked neurons, raw neural activity plot, motion corrected neural activity plot, denoised neural activity plot, heatmap of activity of all neurons, dendrogram showing correlation of all neurons	20
4.2	Effects of applying a Gaussian blur filter to an ROI containing a neuron	22
4.3	Flowchart depicting the process of identifying the most likely target neuron	22
5.1	Images obtained using two-photon microscopy. (a) GFP animals imaged at 10x (left) and 25x (right) ; (b) GCaMP7-tdTomato animals imaged at 10x in the red channel (left) and 25x in the green channel (right). Scale bars are 250 μ m.	28
5.2	Images obtained using SPIM. Showing both the top plane (left) and the bottom plane (right) of the animal. Scale bars are 250 μ m.	30
5.3	Images obtained using confocal microscopy. (a) Recording from the tdTomato channel (left) and corresponding recording from the GCaMP channel (right); (b) close-up view of td-Tomato labelled neurons (left) and GCaMP7 labelled neurons (right). Scale bars are 250 μ m for (a), and 25 μ m for (b).	31

5.4	DeepLabCut tracking of the head, centre, and foot of a Hydra	33
5.5	Continuous behaviour recordings and analysis using DeepLabCut data. (a) depict hydra length over time; (b) angle of bending at centre of Hydra over time; (c) contraction and elongation behaviours over time.	33
5.6	(a) tracked points in the tdTomato channel (left) and corresponding points in the GCaMP channel (right); (b) close-up view of a tracked neuron in the tdTomato channel; (c) corresponding view of the neuron in the GCaMP channel, showing imperfect alignment.	35
5.7	(a) Comparison of tdTomato and GCaMP tracking with Icy over differing video lengths, 5 lengths of 3 different videos were used for comparison and average performance found; (b) boxplots of the improvement found using tdTomato at different numbers of frames.	36
5.8	Effects of the Sub-ROI tracking algorithm. Tracked neuron is circled in yellow, and the activity of non-target neuron is seen in the bottom left corner.	37
5.9	Effects of SubROI tracking algorithm on recorded neural signals. (a), (c), (e) show signals recorded using the algorithm and (b), (d), (e) show signals recorded without. (a), (b) show raw signals; (c), (d) denoised signals; (e), (f) rasterplots of deconvolved spike times.	38
5.10	Effects of Processing on population neural data. Unprocessed heatmap of neural data (left) and fully processed heatmap of neural data (right).	39
5.11	Extraction of calcium activity signal from raw fluorescence recording using ICA. (a) Raw fluorescence recording; (b) extracted calcium activity signal.	40
5.12	Comparison of ratiometric and ICA methods for motion correction. (a) Raw signal from GCaMP channel; (b) raw signal from tdTomato channel; (c) Effects of ratiometric correction, showing spurious spike circled in red; (d) ICA corrected signal.	40
5.13	Detrending of a motion corrected neural signal. (a) motion corrected neural signal plotted with its fitted polynomial; (b) detrended signal achieved by subtracting polynomial from the motion corrected signal.	41
5.14	Example signals removed by the Gaussian distribution filter	42
5.15	Denoising and deconvolution of both a CB and an RP1 neural signal. (a) RP1 raw signal; (b) CB raw signal; (c), (d) denoised calcium signal using CaImAn; (e), (f) inferred spike times from this signal.	44
5.16	Cluster dendrogram showing relationships between neurons and clusters of neurons. Generated using the 'ward' method of clustering.	45

5.17	Investigation into the RP1 neuron ensemble. (a) Representative signal from an RP1 neuron in the analysed ensemble; (b) Spatial locations of neurons in the RP1 cluster circled in red	46
5.18	Correlation Matrix showing the correlations between all neurons in the dataset. Dendograms are plotted along the top and left sides, and neuron indices along the bottom and right sides.	47
5.19	Motion of a Hydra in hydrogel. (a) Hydra with superimposed locations of neurons at frame 0; (b) Hydra at frame 2000, 200 seconds later.	48
5.20	Correlation matrix showing the correlation between neurons in an immobilised Hydra.	49
5.21	Ensembles detected in immobilised Hydra. (a) Locations of tentacle ensemble; (b) locations of RP2 ensemble; (c) representative activity of tentacle ensemble neuron; (d) representative activity of RP2 ensemble neuron.	50
5.22	Raster plot of neural activity for all recorded neurons with contraction behaviour plotted in green and elongation plotted in blue.	51
5.23	Plot of actual Hydra behaviour with the behaviour predicted by an LSTM model using only neural signals. The grey section indicates the portion of data used for training, and the white section represents data used for testing.	52
5.24	Neural activity signals of the two most important neurons in the dataset for prediction of contraction behaviours, identified via permutation importance.	54

Contents

1	Introduction	1
2	Literature Review	3
2.1	The Neuron Doctrine and Neural Ensembles	3
2.2	Calcium Imaging	5
2.3	Model Organisms and Hydra	7
2.4	Behaviour and Neural Activity	9
2.5	Machine Learning	10
3	Materials and Methods	11
3.1	Rational for Methodology	11
3.2	Hydra Culturing and Care	11
3.3	Transgenic Hydra	12
3.4	Imaging Preparation	13
3.5	Imaging Methods	13
3.6	Behaviour Recordings with DeepLabCut	17
3.7	Spot Tracking with Icy	18
3.8	Immobilisation	18
3.9	Machine Learning	19
4	Neural Activity Analysis Script	20
4.1	Neuron Tracking	21
4.2	Processing Signals	23
4.3	Correlation of Neural Activity	25

5 Results and Discussion	27
5.1 Selecting an Imaging Methodology for Single-Cell Resolution Recordings	27
5.2 Recording and Quantifying Behaviour	32
5.3 Neuron Tracking	34
5.4 Processing Neural Signals	39
5.5 Correlation and Ensembles	45
5.6 Novel Neural Activity During Immobilisation	48
5.7 Connecting Neural Activity and Behaviour	50
6 Conclusions	55
6.1 Limitations	55
6.2 Future Work	56
References	57
Appendices	61

1 Introduction

Exploring how neurons process information is a common research theme in neuroscience and has been investigated from numerous angles in the past. Recently, research has begun to show a diversion from the previously held notion that individual neurons are the functional unit of the nervous system in favour of a model in which groups of neurons – ensembles – provide this unit. Advances in optical methods have allowed the widespread *in vivo* recordings of large populations of neurons simultaneously allowing these neural ensembles and their function to be explored in ways that would not be possible with other techniques. One of the most popular optical methods for studying neural activity is Calcium imaging, a technique allowing the visualisation of neural activity by causing neurons to fluoresce as they fire [1].

While the use of calcium imaging is widespread in groups studying the neural activity of mice, the brains of mammals are vastly complicated - with over 75 million neurons in a mouse brain [2] - and gaining insight into the fundamental workings of neural ensembles can be difficult. For this reason, simpler model organisms can often provide a starting point to be developed upon using more complex models once the theory has been established. The Cnidarian '*Hydra Vulgaris*' makes a promising model organism for studying neural networks at a simpler scale due to the small number of neurons each animal has – a few hundred to a thousand – and the fact that their nervous system is diffuse means that there are no dense areas where neurons are so close together they become challenging to resolve [3].

Current work has identified three distinct neural networks in *Hydra* using optical techniques, each related to a form of behaviour. These networks are the Contraction Burst (CB) network, which fires when the animal contracts, the Rhythmic Potential 1 (RP1) and Rhythmic Potential 2 (RP2) networks which fire during elongation and radial contraction respectively [3]. The associated function of the RP networks is not as clearly defined as the CB network.

This work suggests that ensembles of neurons play a role in the behaviour of *Hydra*, however the amount of data generated in this study was limited as neurons were painstakingly annotated by hand. This limited

dataset, combined with a lack of quantitative behavioural recordings limited the ability for any thorough investigations into the relationship between the function of these ensembles and the behaviours of the animal. However, the methodology described in this study uses new techniques - both genetic and computational - to allow for the large scale automated collection and analysis of a far larger pool of data to examine the role of neural ensembles further in Hydra. This report develops the previous work performed in Hydra neuroscience to allow for automated collection of behavioural and neural data to allow for larger, more complete datasets to be generated in order to study neural ensembles in this simple model organism.

A high throughput analysis pipeline, built from both existing methods and custom-build software, was designed and implemented for the acquisition of single-cell neural recordings and behaviour tracking of *Hydra Vulgaris*. This pipeline was then used to rapidly generate large data sets containing recordings from up to 100 individual neurons while simultaneously recording the behaviour of the animal, and powerful new analysis techniques were then implemented from the field of deep learning to analyse these datasets.

2 Literature Review

In this section the general research leading to this report will be summarised along with some pieces of contemporary work that are particularly relevant. A more technical background on the particular methods used in this report is given in chapters 3 and 4, where these methods are discussed in detail. The following works provide a strong starting point for further study of ensembles of neurons and how they contribute to the processing and encoding of information.

2.1 The Neuron Doctrine and Neural Ensembles

Neurons are electrically excitable cells capable of transmitting and even processing information in the form of electrical signals. Neurons form the brains and nervous systems of animals and allow for the detection and processing of stimuli, the control of movement, and higher processes such as memory and possibly consciousness. Neurons are primarily composed of a cell body (soma) containing the nucleus and functioning to produce the various proteins needed by the cell to function; an axon, which functions to conduct signals along the neuron; and dendrites, which receive signals from other cells. These signals, 'Action Potentials', are a particular type of impulse generated by neurons and consists of a rapid change in the membrane potential of the neural cell. Action potentials are generated when the membrane potential of the neuron exceeds a predefined threshold, this is commonly caused by the stimulation of a neuron's dendrites. Dendrites are the small receptors which originate from the cell body and are responsible for receiving the intra-cellular chemical signals delivered by the axons of other neurons across a synapse. Dendrites act as receptors for these chemical signals and convert them into electrical impulses. If the resulting signal meets the criteria to activate the cell, then an action potential is generated. When a neuron generates an action potential it is commonly said to have 'fired' or 'spiked'. The anatomy of common neurons can be seen in Figure 2.1 [4].

Since the time of Ramon y Cajal the neuron has been considered the fundamental functional component of the nervous system [5]. Cajal famously used Golgi's staining technique to observe the structure of neurons,

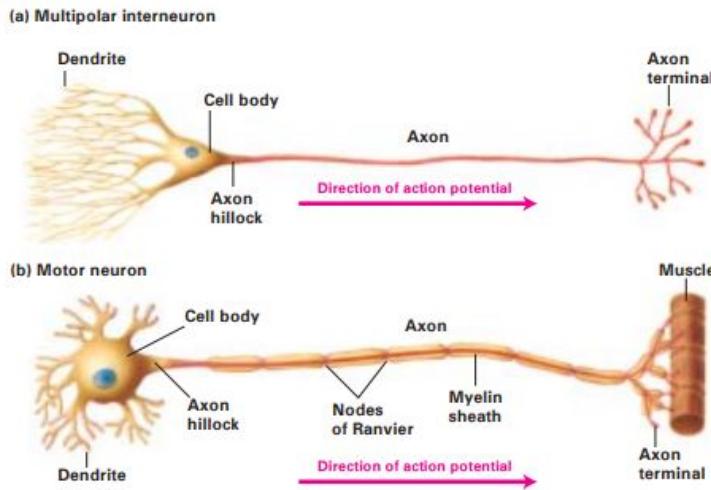


Figure 2.1: Anatomy of mammalian neurons. From Molecular Cell Biology, 5th edition. by H. Lodish et al. [4]

and upon studying the dendrites, axons, and soma of nerve cells, he was able to infer that neurons were self-contained – separate – individual elements. This gave rise to the prevailing notion that it is individual neurons that are the fundamental functional units comprising a nervous system – this notion forms what most would now call the neuron doctrine.

However, more recent work has begun to challenge this doctrine, suggesting that it is in fact groups of neurons – often referred to as ensembles or networks – which may be the basic functional unit of nervous systems. These ensembles are defined as groups of neurons whose constituent cells display the same synchronised activity as each other, often in response to a stimuli or corresponding to a behaviour. While the neuron doctrine provides tremendous insight into many of the various mechanics of phenomena which occur in animal nervous systems, there are many facets of their dynamics that it has so far failed to address – such as behaviour [6]. It is possible that a model of the nervous system based on ensembles, and the emergent properties they may give rise to, could provide some insight into these phenomena and allow us to gain a deeper understanding of the computations that our own brains make.

This notion has been supported by numerous observations, particularly those made with the use of optical techniques such as calcium imaging – which allow for the recording of large populations of neurons simultaneously and less-invasively than using electrodes [7]. One key illustration of the use of these techniques to investigate neural circuits is the recent paper by Corder et al [8]. This paper presented evidence that a neuronal ensemble within the amygdala is in fact responsible for encoding pain. Here the authors used

calcium imaging in freely behaving mice to identify a circuit of neurons that reliably fired together when the mice were exposed to painful stimuli, then the authors disrupted the ensemble and observed a reduced presentation of pain affective-motivational behaviours.

This work suggests that ensembles indeed play a role in the complex and intricate nature of the brain, and that Calcium imaging is a particularly effective method for recording these ensembles – particularly *in vivo*.

2.2 Calcium Imaging

Calcium imaging is a technique which takes advantage of the influx of Calcium ions into a neuron as it during an action potential to allow fluorescence microscopy techniques to be used to image not only the structure of neuron, but its activity. During an action potential, the ubiquitous calcium channels in the membranes of neurons open and allow calcium ions into the cell. This increase in calcium concentration within a neuron is then converted into a fluorescent signal via a calcium sensitive fluorescent molecule. This method allows for action potential of a neuron to be detected as a flash of fluorescence located in the cell. Calcium imaging allows for the activity of far more neurons to be recorded simultaneously than is possible with other methods, albeit with slower temporal resolution. Although calcium dyes were often used for this in the past, modern genetically encoded calcium indicators are now able to rival dyes while having the added benefit of cell specific targeting. [7] [9]

2.2.1 Fluorescence Microscopy

To capitalise on the benefits of calcium imaging, both the calcium indicators and imaging methods must be developed to work together to produce ideal results. Fluorescence microscopy is a common technique which, at its most fundamental, allows for the visualisation of features and processes that would otherwise be observable.

The characteristic aspect of fluorescence microscopy is the presence of a fluorophore. These are molecules capable of absorbing incoming light of a particular wavelength to emit light of a longer wavelength. These molecules are excited by some illumination source with a narrow emission wavelength - achieved either by using a laser, or a filter to remove all but a narrow band of wavelengths - and return light of a different wavelength which can then be separated from the excitation light by a filter such as a dichoric mirror [10]. A diagram of a simple fluorescence microscope set-up can be seen in Figure 2.2.

Many biological techniques allow for these molecules to be attached to specific structures in a sample, either by immunostaining, or by using genetic methods to encode fluorophores into the genome of a biological sample. These properties allows for fluorophores - and therefore the structures they are attached to - to be detected clearly in a sample where the underlying structures would be impossible to observe with traditional microscopy techniques by illuminating the sample with light of the fluorophores excitation wavelength, and recording light of the fluorophores emission wavelength. Perhaps the most famous fluorophore is Green Fluorescent Protein (GFP) [11]. This molecule - derived from the *Aequorea victoria* Jellyfish - revolutionised the field of biology and allowed for the genetically encoded expression of fluorophores.

To capitalise on these fluorophores, many light microscopy techniques have been developed. These range from simple methods such as widefield microscopy, where a sample is bathed in excitation light and the resulting emission collected through a filter, to more complex methods such as two-photon and confocal microscopy. These techniques revolve around exciting fluorophores in specific locations maximal contrast and resolution [10]. Four fluorescence microscopy techniques are discussed in detail in Chapter 3 and their various advantages and disadvantages compared in Chapter 5.

Modern fluorophore design techniques even allow for biological processes to be visualised using these methods. These modern fluorophores are designed to fluoresce only when certain conditions are met, allowing for the detection of underlying biological process that could not previously be detected using light microscopy techniques. One such example of this type of fluorophore is GCaMP, a modified version of GFP designed to facilitate the imaging of calcium flux.

2.2.2 GCaMP

The most popular optical method for studying neural activity is calcium imaging. This technique is made practical by the use of the genetically encoded calcium indicator 'GCaMP'. Genetically encoded calcium indicators are modified fluorophores which are edited into the genome of the target organism. They can be

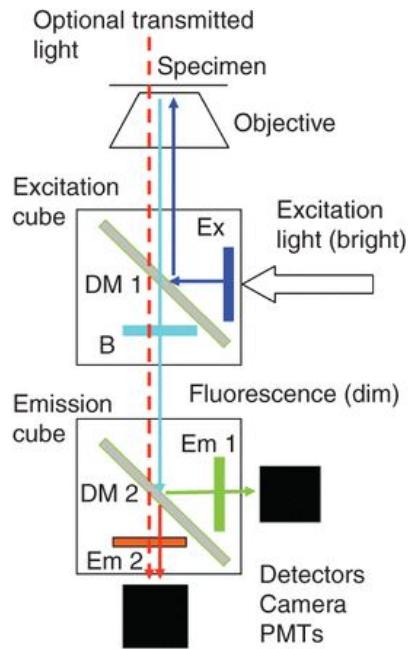


Figure 2.2: Optical path of a typical fluorescence microscope. Figure from M. J. Sanderson et al [10]

designed to be expressed only in neural tissue, or even subsets of neural cells. These cells will then begin to produce the encoded protein themselves, allowing for fluorescent indicators to be localised to the target cells. In turn this allows the accurate labelling and activity monitoring of neural cells using the GCaMP protein [1].

The GCaMP protein is a fusion of Green Fluorescence Protein (GFP) with calmodulin and the M13 domain of the myosin light chain kinase [12]. The resulting molecule will only fluoresce when calcium binds to it and this feature is taken advantage of to allow the optical recording of neural activity. As neurons fire there is a large calcium influx stimulated by the change in membrane potential across the cell. This feature allows calcium imaging – which acts as a direct indicator of calcium activity of neurons – to act as an indicator of neural activity [1]. The most advanced calcium indicators [13] are able to provide varying levels of imaging speeds and intensities to suit the needs of the user. Both the GCaMP6 and GCaMP7 variants were used throughout this project.

One issue with using GCaMP - and calcium imaging in general - is the lower temporal resolution. The fluorescence detected from the resulting calcium flux is inherently slower than the membrane potential changes seen during an action potential [9], meaning that post processing techniques must be employed to recover the underlying neural activity. These methods are discussed in chapter 5.

2.2.3 tdTomato

In addition to GCaMP, other genetically encoded fluorescent indicators may be utilised to aid the identification of cells. Since GCaMP is only visible during an action potential, a calcium insensitive molecule of another excitation colour is often desirable to aid neuron tracking [14]. These molecules emit a constant signal allowing for a neuron's location to be recorded even when there is no neural activity. The indicator used for this purpose throughout this study was tdTomato. This molecule is a high brightness indicator with an emission in the red end of the spectrum (581nm), ideal for use with the green emission spectrum of GCaMP without minimal cross talk between the fluorophores. [15]

2.3 Model Organisms and Hydra

Even with the optical and genetic tools to record from large populations of neurons in place, understanding the complexity of mammalian brains is a daunting mountain to climb. A common first step to understanding hugely complex systems is to build understanding from the study of a simpler model. Commonly, the model organism *Caenorhabditis elegans* has been used to study neuroscience. Recent *C. Elegans* work has been

able to utilise calcium imaging to record the activity of some of the animal's neurons during behaviour [14]. However, due to the presence of centralised ganglia - where neurons become too densely packed for all of them to be resolved individually - it becomes impossible to image the totality of the neurons in the animal simultaneously.

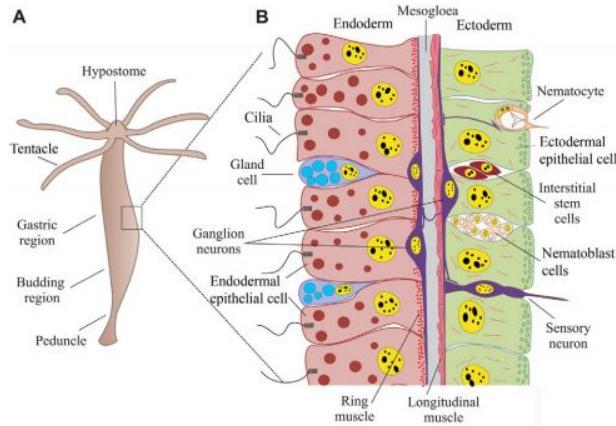


Figure 2.3: The anatomy of a Hydra. Image from U. Technau and R. Steele [16]

ring around their mouth and peduncle. This nerve net is depicted in Figure 2.4. While Hydra have few of the complex sensory or motor systems of their bilaterian cousins they are able to respond to stimuli and to perform spontaneous behaviours. Their simple nervous system is believed to be a precursor to ours, allowing us to investigate the workings of simpler model of our own nervous system while possibly gaining insight on some of the fundamental principles that still underpin its function. [17]

There are many species of Hydra, however *Hydra Vulgaris* are the model organism of choice due to the relative ease in which transgenic animals can be produced [19], and it was this species used throughout this investigation. *Hydra Vulgaris* have been proposed as a viable model organism for neuroscience due to a number of key reasons. The limited number of neurons allows for the imaging of their entire nervous system, conveniently a nerve net, devoid of ganglia that would make neurons too densely packed to resolve; and the limited

Yet novel model organisms such as *Hydra* may provide an interesting solution to this problem. *Hydra* are a biologically immortal species of freshwater Cnidarian, a phylum including species such as jellyfish. *Hydra* are small deformable animals found all over the world with a basic anatomy consisting of a head, a peduncle (foot), tentacles and a mouth. *Hydra* are composed of two layers - an endoderm and an ectoderm - and the neurons of these animals are arranged at the boundary between these layers. The anatomy of a *Hydra* can be seen in Figure 2.3. These simple organisms have no centralised nervous system, instead they have a nerve net with dense nerve

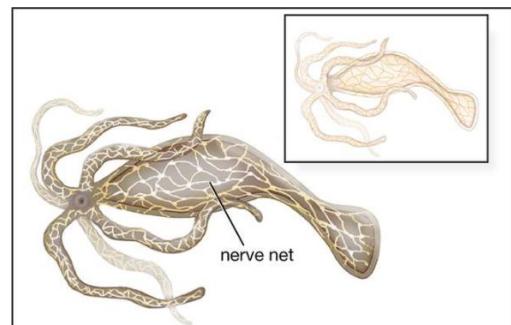


Figure 2.4: The nervous system of *Hydra*. Image from Encyclopaedia Britannica [18]

behavioural repertoire makes classifying their behaviour simple. Recent work by Dupre and Yuste demonstrated the ability to image from almost all of the neurons in a Hydra simultaneously [3]. This paper also presented evidence of three distinct neural ensembles within Hydra's nervous system, each correlating to distinct behaviours. CB neurons are high frequency neurons associated with the longitudinal contractions, RP1 neurons are slower frequency neurons associated with elongation, and RP2 neurons are very slow frequency neurons possibly associated with radial contraction. Although this paper made important insights into the workings of the nervous system of Hydra, the study was limited due to a lack of automation, forcing the investigators to painstakingly label neurons by hand, thus reducing the size of the dataset generated. Now – with a wealth of new computational and genetic tools at our disposal – performing analysis of the nervous system of Hydra on a far larger scale is possible, possibly allowing for a deeper and more complete understanding of their nervous system, and nervous systems as a whole, to be developed.

2.4 Behaviour and Neural Activity

One key goal in neuroscience is to understand the connection between neural activity and behaviour, and to predict the effects that certain patterns of neural activity will have. In a recent study by M. Scholz et al. the behaviour of a *C. Elegans* worm was predicted from its neural activity using a simple regressive model [20]. The prediction of behaviour from neural activity can open doors allowing for insight to be gained into how neurons encode and process information by allowing us to see what patterns of activity relate to what displayed behaviour. This study relied heavily on Principle Component Analysis (PCA) to reduce the number of dimensions data had, essentially simplifying the recorded data to allow for easier computation. This was useful to allow the recording of a complex dataset such as behaviour, and to allow a linear regressive model the best chance of connecting the recorded variables, however more modern deep learning methods would not require such simplification.

This suggests that with adequate recording methods for behaviour and neural activity, powerful machine learning tools can be implemented to infer the connection between these variables. Although the predictions made using the method described were not perfectly accurate, with the implementation of cutting edge machine learning tools this accurate could likely be improved.

Although neural recordings of Hydra [3] and behavioural classifications have been made [21], these methods are not designed to work together to acquire data of behaviour and neural activity simultaneously as each method relied on different preparations and imaging methods. Current neural recordings require the user to manually annotate and track every neuron throughout every frame of a recording and as a result, is not

capable of collecting large scale datasets due to the large amount of time this would take. Behavioural classification was performed using a bag-of-words machine learning algorithm. While this provided stable classification of behaviour, it was only able to organise behaviours into broad classes and as such would not be able to capture the small nuance and details of the animal's movements.

2.5 Machine Learning

Machine Learning refers to algorithms which operate to produce results without receiving specific instructions. These algorithms are designed in a way in which the computer can learn from the data presented to it in order to generate its own solutions to the problem given to it.

Deep learning is a sub-field of machine learning built around the use of artificial neural networks. These algorithms are able - in theory - to learn any mathematical relationship given adequate training data. Deep learning algorithms are designed to receive a set of input data along with corresponding, user defined, outputs. The model is then 'trained' on this dataset to iteratively generate a model which can accurately predict the outputs from the inputs, and - once trained - can predict results from new input data. However, these deep learning models operate as 'black-boxes' where the internal workings and logic used to determine the results they produce are unknown. This unfortunately means that determining which features (input data, in this case each feature represents the signal generated from a single neuron) are important to the prediction of a result [22].

Despite this there are some methods which allow for information to be recovered from deep neural networks. One of these is the method of sensitivity analysis [23]. This method involved the removal of features from use in the prediction process and evaluating the effect their removal has on the accuracy of the model. This provides an insight into which features are most important to the accuracy of a model, implying that these features contain the information used by the model to predict the output.

Recent developments of time-series deep learning approaches have revolutionised our ability to interpret sequential signals, where information from the past bears significant relevance to the meaning of recordings in the present. A key example of this type of signal is speech, or even the signal generated by a neuron. In order to interpret these signals, development of new deep learning architectures was required. Primarily used is the Long Short Term Memory (LSTM) architecture. This is a form of recurrent neural network (RNN) where signals are stored as they are received and used to generate some internal state. This allows past inputs to affect the way the current input is interpreted [22].

3 Materials and Methods

3.1 Rational for Methodology

In order to analyse the dynamics of neural ensembles and their relationship to behaviour, it is necessary to obtain data from the nervous system with single cell resolution. Once the activity of individual neurons can be extracted it then becomes possible to group them into ensembles and analyse their role in behaviour and perception.

In order to record from large populations of neurons simultaneously, three main components were needed. First, a suitable transgenic Hydra line was needed. This line would need to allow for the use calcium imaging to record the activity of its neurons during behaviour. Second, an imaging methodology capable of recording this neural activity was necessary. This imaging method would need to provide adequate resolution, contrast, and signal to noise ratio (SNR) for clean signal acquisition and cell tracking. Finally, software tools were developed to track neurons throughout video recordings, extract the signals from neurons, and process these signals to extract the underlying neural activity. Each of these components needed to synergise to create optimal results.

This methodology describes a complete processing pipeline, from animal care to analysis of data, designed to allow for the rapid collection and analysis of simultaneous neural activity and behaviour data and to facilitate further research into the function of neural ensembles.

3.2 Hydra Culturing and Care

Hydra were the most fundamental component of this investigation and their culture and care were of the upmost importance. Hydra were incubated at 18°C and cultured in petri-dishes filled half way with Hydra media. Hydra media was created in-house following the recipe provided by the Steele lab at UC Irvine

[19] [25]. Animals were fed between one and three times weekly depending on whether the animals were being imaged or the culture was being grown. A feeding schedule of three times weekly would result in rapid colony growth, and once weekly would keep the animals at a small size which was advantageous for imaging since more of the animal could fit in the field of view of the microscope. Hydra were fed Artemia Salinas nauplii (brine shrimp) by pipetting shrimp over the Hydra. The medium was then changed and the Hydra washed approximately three hours after feeding. Animals were imaged two days after feeding to improve image clarity and returned to their medium within an hour of being removed. [24]

An automatic Hydra culturing system was also implemented allowing for feeding and cleaning of Hydra with minimum effort from the user. This system was built following a design developed previously by members of the Steele and Yuste Labs and can be seen in Appendix A.

3.3 Transgenic Hydra

Many transgenic lines of *Hydra Vulgaris* were utilised throughout the course of this project, however most common were the neuronal GCaMP6 and the neuronal GCaMP7-tdTomato lines developed by the Yuste lab following the protocol outlined in Juliano et al. [3] [19]. All lines used expressed genetically encoded fluorescence indicators in their neurons in order to allow the visualisation of the the location or activity of the cells in the Hydra's nervous system.

The GCaMP6 strain expressed only GCaMP6 in its neurons and was used initially for widefield recordings due to its strong signal, however the advantages of the GCaMP7-tdTomato line in allowing single cell tracking far outweighed the seemingly dimmer fluorescence from GCaMP7. As a result, the GCaMP7-tdTomato line was used as the primary strain of Hydra for investigating neural activity with single cell resolution.

The GCaMP7-tdTomato line expressed two types of genetically encoded fluorescent indicators in all cells derived from its interstitial cell lineage. The GCamp7 calcium indicator was expressed in the cytoplasm of the animals' neurons, and tdTomato expressed in the neurons' nucleus. This strain allows for the spatial tracking of the continuous signal from tdTomato, facilitating the tracking of individual neurons and the extraction of their GCaMP7 fluorescence signal produced when neurons fired. Unfortunately, the interstitial cell lineage also contains cnidocytes and gland cells. This meant that not only neurons were labelled and signal would be collected from non-neuronal cells [3]. This limitation was mostly mitigated through filtering of the resulting signals based on the differences in signals produced by neural and non-neuronal cells.

A neuronal GFP strain was also used to evaluate procedures and imaging techniques prior to using the more

delicate and less abundant GCaMP7-tdTomato line. These animals expressed GFP in their neuronal cells giving a constant, strong, signal allowing for simple evaluation of a systems ability to detect fluorescence before attempting to detect the weaker and discontinuous signal from one of the strains expressing a GCaMP protein.

3.4 Imaging Preparation

Hydra were imaged following the method described by C. Dupre and R. Yuste [3] and is depicted in Figure 3.1. However, a 120 μ m spacer was used to allow the hydra maximum freedom to perform behaviours without being impeded my friction. In this preparation Hydra are squished between two coverslips creating an essentially flat object to image whilst minimally restricting the behavioural repertoire of the animal. Hydra - along with a small volume of Hydra medium - were pipetted onto the bottom coverslip and the second coverslip placed on top of them. The coverslips were then gently pressed together to form a seal and to flatten the animal without harming it.

This preparation allows for the neurons in the animal to be imaged in a single plane, eliminating the need to take multiple images at different depths and allowing faster imaging times. The preparation's ease of use and robustness, combined with its facilitation of imaging large numbers of neurons simultaneously in a freely behaving animal, made it an ideal method to investigate the relationship between neural activity and behaviour in Hydra.

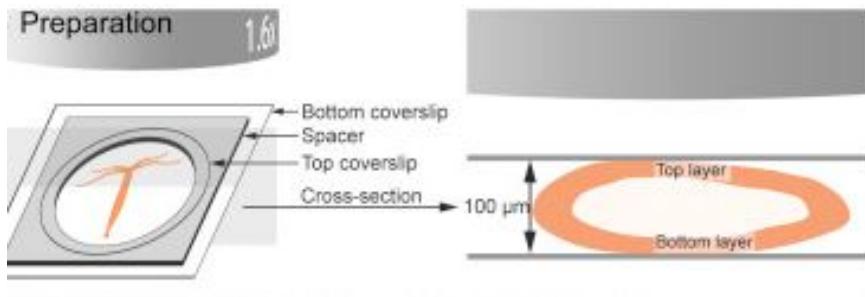


Figure 3.1: Diagram depicting the preparation of Hydra for imaging neural activity. Figure from C. Dupre and R. Yuste [3]

3.5 Imaging Methods

Multiple fluorescence microscopy methods were used over the course of this investigation. This was done in order to select the most appropriate method to capture neural activity and behaviour in Hydra. These meth-

ods were tested for their ability to perform simultaneous two-colour recordings with a resolution, contrast, and SNR adequate for automated single cell tracking when recording at a frame of 10Hz - high enough to minimise motion blur in a contracting Hydra.

The four imaging set-ups used are detailed here. While widefield imaging was still used for some recordings due to its simplicity, confocal microscopy proved to be the optimum imaging method and was used to collect all of the single-cell resolution data recorded in this report. The other methods tested both showed potential, however both SPIM and two-photon microscopy would require significant modifications to allow for adequate image acquisition.

Once acquired, recordings were then processed and adjusted for brightness and contrast using ImageJ Image analysis software (NIH)

3.5.1 Widefield Microscopy

Widefield microscopy is the simplest method of imaging used. A simple LED illumination source is passed through a filter set to provide narrow wavelength illumination of the sample and the emitted light is then separated using another filter. This emitted light is then recorded by a sensitive camera. [26]

This technique provided rapid and simple imaging of Hydra and was commonly used to screen animals for GCaMP expression and to assess the general well being of Hydra cultures. It was also used to perform preliminary studies into Hydra's neural activity to assess the feasibility of recording single cell activity of neurons. While this was a convenient imaging method, the poor spatial resolution compared to the other methods studied and poorer contrast due to a lack of optical sectioning capabilities (the ability to reject light from outwith the focal plane) made it unsuitable for automated single cell resolution recordings.

Widefield imagine was performed using a Leica M165 fluorescent stereo microscope with a filter set for GFP (ET GFP M205FA/M165FC) and a Hamamatsu ORCA-Flash 4.0 sCMOS camera. HCImage Live software (Hamamatsu) was used for image acquisition and a SOLIS-445C Blue High-Power LED (ThorLabs) was used to provided illumination. This LED was controlled by a DC2200 LED driver (ThorLabs). This setup allowed for the delivery of high intensity light at the optimum wavelength to stimulate CGaMP.

3.5.2 Two-Photon Microscopy

Two-photon microscropy - and multi-photon microscopy in general - is a powerful technique which takes advantage of the multiphoton excitation effect. Through the simultaneous absorption of multiple lower

energy photons, a fluorophore which normally absorbs photons of a higher energy may be excited. This effect allows for excitation of a fluorophore only the focal plane of the excitation light, as only here will the density of photons be high enough to generate excitations [27].

This effect allows molecules such as GCaMP to be excited using long wavelength light in the infrared range which has many advantages. First, it allows for greater penetration into tissue due to the lower absorbance of long wavelength light. This alone has made the technique highly advantageous for applications such as in-vivo imaging of the cortex. Secondly the lack of absorption outwith the focal plane greatly reduces photobleaching and allows for accurate optical sectioning - improving contrast and spatial resolution in all three dimensions. There are disadvantages to this technique however. Due to the reduced efficiency of multi-photon absorption compared to single photon absorption, the supplied photon density must be far greater to cause enough absorption events to produce a significant emitted signal from the target fluorophore [28]. This requires a significant increase in laser power compared to other methods and still produces a weaker signal than comparable single-photon techniques. Two-photon microscopy is widely used for calcium imaging in neuroscience, particularly for studying the neural activity in awake behaving animals, and has been responsible for a number of important discoveries [29].

During this investigation, two-photon microscopy was performed using a custom set-up similar to the design described in Han et al. [31]. Laser illumination was provided by a Chameleon Ultra II Ti:sapphire laser (Coherent) tuned to a wavelength of 920nm. The illumination was focused onto the sample using an electrically tunable lens (EL-10-30-C-NIR-LD-MV, Optotune) and scanned using both a galvanometric scanner (6215HM40B, Cambridge Technology) and a resonant scanner(CRS 8K resonant scanning system, Cambridge Technology). The resulting signal was then recorded using two H7422P-40 photomultiplier tubes (Hamamatsu), one for each colour channel.

3.5.3 Selective Plane Illumination Microscopy

Selective Plane Illumination Microscopy (SPIM) is a fluorescence microscopy technique designed to allow for the illumination of a narrow section of a sample with a thin sheet of laser light referred to as a lightsheet. SPIM differs from other methods of imaging as illumination is provided perpendicular to the imaging objective - and from the side of the sample - requiring separate light paths for illumination and collection of emitted light [32].

This imaging method provides a lower phototoxicity compared to methods such as confocal microscopy, as well as allowing narrow optical sectioning unlike widefield microscopy. However, the spatial resolution is

comparable to widefield microscopy, and worse than the resolution achieved when using comparable two-photon or confocal methods [32]. SPIM has advantages over widefield microscopy in its improved contrast and optical sectioning abilities, and advantages over both two-photon and confocal microscopy in image acquisition speed as it does not require scanning to illuminate a sample. Unfortunately, the preparation of samples for use with a lightsheet is more complicated than that used with other imaging methods and the preparation described in section 3.4 was found to be unusable without major modifications.

The SPIM set-up used was built from a modified dissection scope housing a MizarTILT lightsheet microscopy system (Cairn Research) [33], as well as two 10x imaging objectives - one above the sample and one below. This allowed for the imaging of Hydra from top and bottom, improving the signal obtained from neurons on the bottom of the animal. This set-up was used courtesy of the Maddox lab at UNC Chapel Hill.

3.5.4 Confocal Microscopy

Confocal microscopy is a reliable fluorescence microscopy technique characterised by the use of a pinhole in front of the detector to occlude any out-of-focus light. This has the effect of rejecting any light not originating from the focal plane of the microscope's objective, resulting in improved contrast over widefield microscopy and the ability to perform optical sectioning. Typically, this technique uses a laser scanning method of sample illumination, however this imposes limitations on the speed of image acquisition based on how fast the system is capable of scanning over the entire imaging area. When imaging live specimens which move during the imaging process this limitation often makes laser scanning confocal microscopy unsuitable as its imaging speed is too slow. The use of a spinning disk scanning method, as shown in Figure 3.2, allows for multiple points to be rapidly, simultaneously, scanned across the imaging area allowing for far faster image acquisition. Additionally, this allows for the use of sCMOS or CCD cameras for image acquisition. These have greater quantum efficiencies (probability of converting a photon into an electron) than the photomultiplier tubes used in laser scanning confocal microscopy [34].

Confocal imaging proved to be the most appropriate methodology for this study and, for the presented results, all data was collected using a two-colour spinning disk confocal microscope (Olympus BX50W1 microscope with a yokogawa CSU-X1 spinning disk confocal scanner unit (Solamere Technology Group)). To capture both the red signal from tdTomato and the green signal from GCaMP separately and simultaneously a dichoric mirror was used to direct green light towards one detector and red light to the other. The detectors used were two CCD Cameras (XR-MEGA10, Stanford Photonics) and excitation illumination was provided

by two lasers (Coherent OBIS), one with a 488nm wavelength and the other a 561nm wavelength to excite both GCaMP and tdTomato. A 10x magnification objective (UMPlanFl 10x/0.30 W, Olympus) was used to capture as much of the animal within the FOV while allowing for high enough resolution to capture neural activity at a single cell resolution. Piper Camera Control software (Stanford Photonics) was used to control the set-up and to acquire images from the detectors.

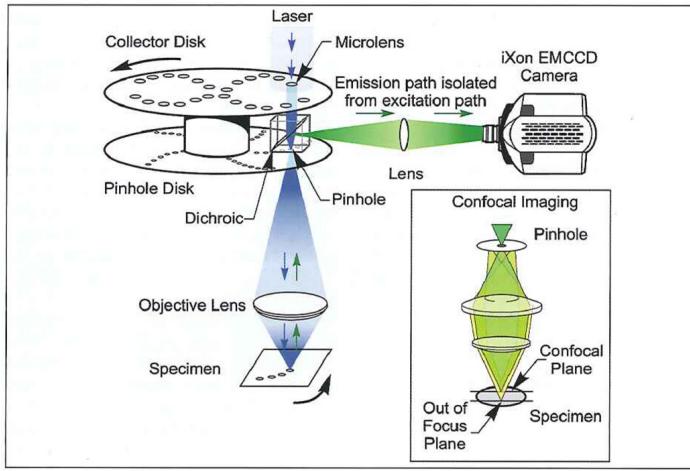


Figure 3.2: Diagram of a Yokogawa spinning disk confocal scanner unit. Image from M. Nelson et al. [34]

3.6 Behaviour Recordings with DeepLabCut

To record animal behaviour, the software package DeepLabCut was used. This Python package allows for markerless pose estimation using deep neural networks and allows for user labelled features to be tracked throughout a video of any arbitrary length. The user labels as few as 20 frames by hand and DeepLabCut is able to accurately detect these features in the remaining unlabelled frames. This software allows for accurate pose estimation for behavioural recordings with only a small amount of work required from the user to generate a training dataset [36].

DeepLabCut was used to develop a model capable of identifying and tracking the head, centre, and foot of a Hydra. Multiple videos of different animals were manually labelled and the resulting datasets used to train a model to identify these features autonomously [35]. Due to the highly deformable bodies of Hydra, features such as the head appear very different when the animal is contracted versus when it is elongated. To combat this, frames were selected manually, as opposed to being automatically selected by DeepLabCut, to include a good representation of frames from both behaviours. Even with this training method it proved difficult to generate a general model able of labelling novel videos accurately, so the approach was taken to train

and test these models on the same dataset by manually annotating 20 frames from a larger video and letting DeepLabCut label the rest.

DeepLabCut utilises deep learning and as a result a GPU is required to train the models in a reasonable time frame. This was initially done using Google Colab, a cloud based GPU computing service, however during the course of the project this process was moved to an in house analysis computer equipped with a powerful graphics card (NVIDIA TITAN RTX) capable of quickly training models.

3.7 Spot Tracking with Icy

Once recordings were made, the video produced from the red (tdTomato) channel was analysed using the 'Icy' software package (Institut Pasteur, France Bioimaging) using the spot tracking plugin. This software package was used to track the positions of neurons by automatically detecting the bright spots generated by the tdTomato signals and tracking their positions throughout the video. These positions were saved for further analysis using the custom Python script detailed in Chapter 4.

The spot tracking plugin relies on the Multiple Hypothesis Tracking algorithm and allows for the automatic detection and tracking of bright spots in an image throughout the duration of the video [37]. Although this was an adequate tracking method, the algorithm does lose points as videos progress. This meant recordings over a few thousand frames would successfully track far fewer points than shorter videos, limiting the duration of recordings used.

Icy is also capable of tracking disappearing particles - such as neurons during calcium imaging. This functionality was initially used to attempt to track neurons, however the performance was poor and few neurons were able to be tracked. The performance of this tracking method compared to the tracking of tdTomato signals is evaluated in Chapter 5.

3.8 Immobilisation

Hydra immobilisation was performed following the protocol detailed by K. Burnett, E. Edsinger and D. Albrecht [38]. Following this preparation and replacing deionised water with Hydra medium - following advice from E. Edsinger - a photo-crosslinked polyethylene glycol hydrogel was produced capable of rapidly immobilising a Hydra. This solution was liquid until photo-crosslinked by exposure to UV light when it would solidify into a stiff gel and immobilise the sample within it.

5-6 μ L of this solution was pipetted over a Hydra placed on a cover slip with a 120 μ m spacer. A second coverslip was then placed on top of the solution and the preparation was placed on a transilluminator and exposed to UV light (approximately 365nm) for a period of 100s. This long exposure time was to maximally solidify the hydrogel ensuring Hydra were thoroughly immobilised despite their relatively strong muscle activity. This protocol was used to reversibly immobilise Hydra without using chemical agents which may interfere with the neural activity of the animals.

The hydrogel was created by combining PEGDA hydrogel (PEGDA, 1mL, ESI BIO) with a UV photoinitiator (2-Hydroxy-4'-(2-hydroxyethoxy)-2-methylpropiophenone, Sigma-Aldrich) [38].

3.9 Machine Learning

Machine learning techniques - specifically deep learning methods - were employed to investigate the relationship between neural activity and behaviour. Since neural activity is a complex sequential signal to interpret, the LSTM deep neural network architecture was selected due to its capabilities in analysing time-series data [22].

An LSTM model was built using the Keras and Tensorflow Python libraries. This model was then trained using behaviour data generated from the behavioural analysis step using DeepLabCut outlined previously, and neural activity data extracted using a custom neural analysis script developed for this project (detailed in chapter 4). The model was trained to predict behaviours 50 frames before they occurred. This was done in an attempt to force the network to look for patterns in neural activity that would precede a particular behaviour and allow the model to produce information about the neural activity involved in the planning and execution of behaviour.

Additionally, a method of detecting feature importance was employed based on the principle of sensitivity analysis [23]. In order to achieve this, a function implementing a method of permutation importance was employed [39]. This function sequentially obscures the signal of each neuron by combining them with high-amplitude artificial noise, effectively removing any information contained in these signals. Using this method, the total number of features remains the same - a key constraint of deep learning models such as these. Once a signal has been obfuscated, the accuracy of the model is evaluated and recorded. This process is repeated for each neuron allowing for the importance of every neural signal to be recorded based on its effect on the overall model accuracy.

4 Neural Activity Analysis Script

Despite the powerful methods available for analysing calcium imaging data from other model organisms, there were no tools suitable for analysing neural activity from the mobile and deformable nerve nets found in Hydra. This lead to the need for a new tool to be created to extract neural signals from raw microscopy data and allow for high-throughput analysis of calcium imaging in Hydra with single-cell resolution.

In order to achieve this, a novel Python package was developed. This was designed to allow for Hydra neural data to be analysed using the methods currently implemented for more established model organisms such as mice. The developed code takes data from raw microscopy videos, extracts fluorescent signals from all tracked neurons, processes these signals, extracts the underlying neural activity, and correlates these neural signals into ensembles. An overview of the workflow of this package can be seen in Figure 4.1.

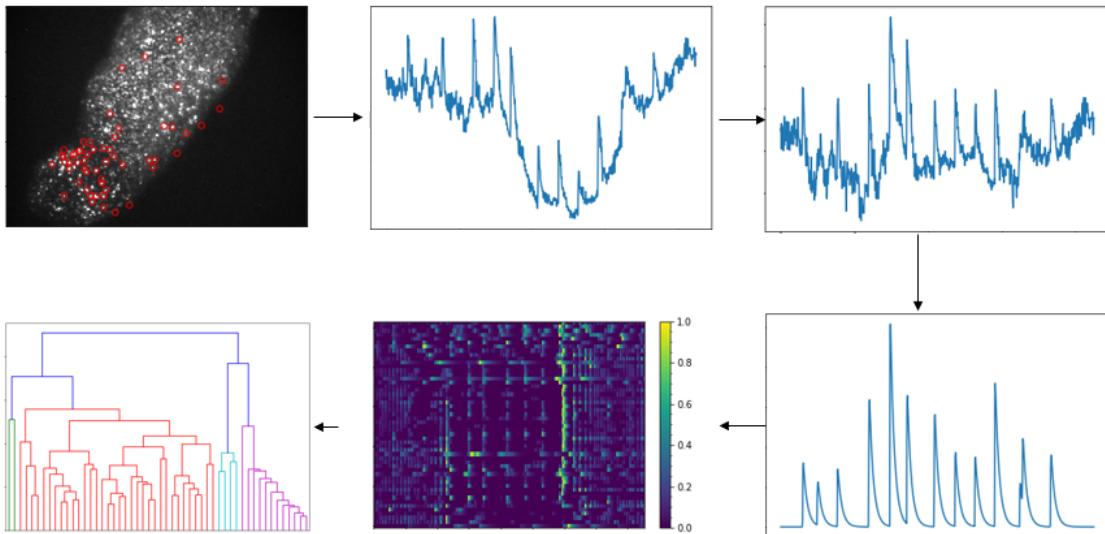


Figure 4.1: Information flow through the developed Python script. Clockwise: raw video with tracked neurons, raw neural activity plot, motion corrected neural activity plot, denoised neural activity plot, heatmap of activity of all neurons, dendrogram showing correlation of all neurons

The main focus of this project was to develop a user-friendly, adaptable, and robust Python package which was capable of automating the entire process of analysing the activity of large numbers of neurons simultaneously. This package - along with a novice-usuable Jupyter notebook (an interactive environment to use and develop Python code, see Appendix B) containing instructions and comments - is designed to be human-readable and usable to those with no programming experience to maximise the accessibility of the final product for those studying neural ensembles in Hydra or any similar organism.

Along with analysing neural activity, either user defined behavioural patterns or automatically predicted behaviours from DeepLabCut data can be plotted with neural signals and inferences made between them.

4.1 Neuron Tracking

While Icy provided excellent tracking of a neurons general location, it was not able to accurately track individual neurons alone. This is due to the signal seen from the GCaMP channel and the tdTomato channel never perfectly overlapping. This imperfect alignment was likely caused by a combination of the two detectors not being perfectly aligned, the imperfect chromatic aberration correction provided by the optical system used, and the fact that the different indicators were expressed in different areas of the cells. Due to this, the actual neuron - when viewing the GCaMP channel - would not be in the exact position detected by Icy, making extracting the neural activity of only that cell and excluding others nearby difficult. In order to rectify this, and allow for accurate single-cell resolution recordings of neural activity, a second - 'fine-tuning' - tracking step was developed and implemented. This function utilised the OpenCV2 Python package, a computer vision package allowing for the analysis of images.

First, a region of interest (ROI) slightly larger than the diameter of a neuron was taken from the video, where the centre of the ROI was the point recorded by Icy as the location of the neuron. This was done to create an ROI as small as possible, excluding the maximum number of cells while still capturing the 'real' neuron. The algorithm then utilised a Gaussian blur filter to reduce the brightness of pixels far from the centre of the image and increase the brightness of those closer to the centre. The effect of this can be seen in Figure 4.2. This was done as neurons were often near the centre of the extracted ROI, making cells closer to the centre more likely to be the neuron tracked in red by Icy - the target neuron. This step was designed to bias the algorithm towards selecting these cells.

Once this was complete, a novel algorithm was developed to allow for the selection and tracking of the cell which was deemed most likely to be the target neuron. To achieve this, the brightest point in a frame was

selected and the likeliness of it being from the target neuron was calculated based on its distance from the centre of the image, brightness, and - for all frames bar the first in a video - distance from the previously identified neuron location. Then this neuron was removed from the image and the new brightest point was selected and a new score generated. This was repeated for a user defined number of iterations, however more than four iterations seemed to have little improvement on results and four iterations were used for the data collected in this report. Figure 4.3 shows a flow chart depicting this process. This function was also able to prevent the selected point drifting when neurons were close enough to overlap. This feature kept track of the previous user-defined number of positions, and would correct the predicted position if it was seen to steadily change location.

Once this point was selected by the algorithm, the average intensity of the pixels within a circle around the neuron was recorded, giving the intensity of the tracked neuron for the frame investigated. This process was repeated for every frame recorded for every neuron tracked to generate a large dataset containing single-cell neural activity recordings for all of the neurons tracked by Icy. The code for this function can be seen in Appendix C.1.

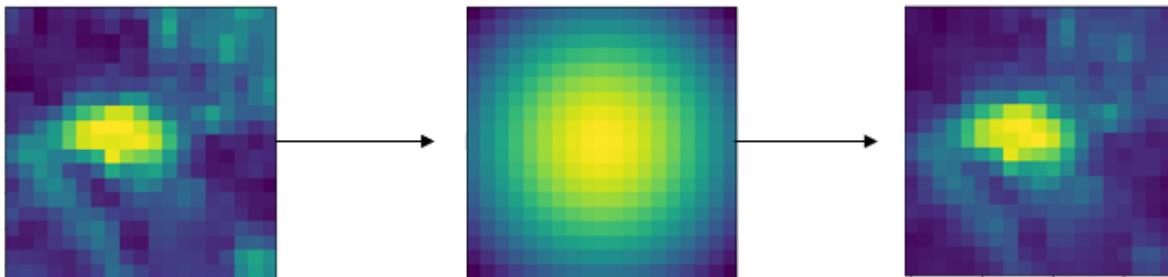


Figure 4.2: Effects of applying a Gaussian blur filter to an ROI containing a neuron

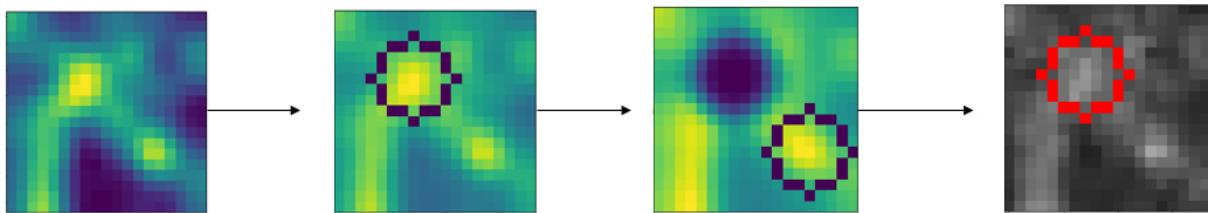


Figure 4.3: Flowchart depicting the process of identifying the most likely target neuron

4.2 Processing Signals

Once the raw signals had been extracted, the challenge then became extracting the underlying neural activity from these fluorescence recordings. This was a challenge of two parts. First, the Calcium activity signal - the component of a signal resulting only from GCaMP7 fluorescence due to the calcium flux which accompanies the firing of neurons - needed to be extracted and the component resulting from the motion of the animal discarded. Second, the underlying neural activity - i.e. the spike times of action potentials - needed to be deconvolved from the recorded calcium signal.

4.2.1 Correction of Motion Artefacts

To tackle this first hurdle, two methods were implemented to correct for motion artefacts and their results compared. A ratiometric approach, and the Independent Component Analysis (ICA) algorithm. When testing these methods on real data it became clear that ICA was the superior method for extracting calcium signals, confirming the results found by M. Scholz et al. [20]. Both of these methods utilised intensity recordings taken from the tdTomato channel in same area used for the GCaMP channel. This allowed for two signals to be used, the GCaMP channel recording - containing both signal from motion artefacts and from the fluorescence of GCaMP7 - and the calcium-insensitive tdTomato channel recording - containing signal only from motion artefacts. There was however some slight cross talk between the two channels and as a result there was occasionally some minimal GCaMP7 signal present in the tdTomato channel recordings.

The ratiometric method of motion correction utilised the ratio of the the GCaMP channel signal and the tdTomato channel signal as a way to reduce the effects of motion artefacts. The algorithm then corrected these values using the baseline ratio of the channels. The equation used for this calculation can be seen in eqn. 4.1.

$$I = \frac{\Delta R}{R_0} \tag{4.1}$$

where:

I = Motion corrected intensity of GCaMP7 fluorescence

R = Ratio of intensities from GCaMP channel and intensities from tdTomato channel

R_0 = Baseline intensity ratio, taken as the value of lower 20th percentile of R values for a neuron

ΔR = $R - R_0$

ICA is an algorithm designed to separate a signal into maximally independent components, where the number of components is equal to the number of different recordings made [40]. In this case two separate recordings were made - one from the GCaMP channel and another from the tdTomato channel. The two components separated by this algorithm were the signal generated by motion artefacts, and the signal generated from CGaMP7 fluorescence. The signal generated by GCaMP7 fluorescence was then recorded and used for further analysis.

ICA was implemented using the FastICA function from the scikit-learn Python package and customising a similar function from M. Scholz et al. to work with the data formats generated from the Hydra videos and to add additional functionality. This additional functionality eliminated the effects of the inherent randomness of the FastICA algorithm through iterating the function multiple times and selecting the result which least correlated to the signal recorded from the tdTomato channel.

Following the correction of motion artefacts using either method, a further detrending step was performed to remove any residual non-neuronal artefacts in the resulting signals. To perform this a polynomial of a user-defined degree was created to fit the general form of a signal, then subtracted from said signal. This had the effect of removing any aspect of the signal which was not a spike or a motion artefact, such as photobleaching. This was implemented using the polyfit function from the numpy Python library. The code for this section can be seen in Appendix C.2.

4.2.2 Filtering Non-Neuronal Cells

Due to the nature of the genetic labelling process in Hydra, the fluorescent indicators expressed in neuronal cells are also expressed in all cells derived from the interstitial cell line and as a result, some signals are recorded from non-neuronal cells [3].

This is rectified by a filtering step included in the developed code. This filtering step worked by evaluating the closeness of the fit of a signal to a Gaussian distribution and removing those above a certain fit threshold. Neuronal cells would fit this distribution far worse than non-neuronal cells due to the presence of spikes in their signals. This method was implemented using the normaltest function from stats module of the scipy Python library. This function was used on all of the motion corrected and detrended signals. The code for this section can be seen in Appendix C.3.

4.2.3 Denoising and Spike Deconvolution

Once accurate signals had been extracted from the raw videos, the underlying neural dynamics were recovered from the calcium signals. To achieve this the Python library 'CaImAn' was used, this toolbox was designed for the analysis of calcium imaging data [41]. However, CaImAn is designed for use with data recorded from organisms such as mice - where neurons are mostly stationary - and as a result is unusable for Hydra. To circumvent this challenge, only the deconvolution algorithms from CaImAn were used and all other processing was completed by the custom script described in this chapter. This allowed data to be collected and processed with tools built to work with Hydra, to produce a signal that would be suitable for deconvolution with CaImAn.

CaImAn utilises sparse non-negative deconvolution as described by Volgenstein et al. [42] and implements this approach using the constrained deconvolution algorithm described by E. Pnevmatikakis et al [44]. The OASIS algorithm described by Friedrich, Zhou, and Paninski [43] is then used to deconvolve underlying neural activity from calcium traces. This method fits an autoregressive model based on the calcium dynamics of neurons, then uses the OASIS algorithm to infer spikes from the generated model. Although parameters can be auto generated based on the input data, performance was optimised by manually tweaking the noise parameter for each new dataset. This was implemented by calling the Constrained_FOOPSI function from CaImAn's deconvolution module and a small function was build to automatically apply it to multiple signals simultaneously. This method produced both fully denoised calcium traces and predicted spike times of neural activity which were then used for further analysis. The code described in this section can be seen in Appendix C.4.

4.3 Correlation of Neural Activity

After this data had been extracted, neurons were clustered into ensembles using hierarchical cluster analysis - specifically, agglomerative clustering using ward's linkage method [45]. Once these ensembles were identified further investigation could be made into how they function, encode behaviour, and how different networks interact with each other.

Hierarchical clustering is a method used to iteratively join data points together into tiers of similarity. It can either be performed via divisive clustering, where data is first considered as one large cluster then divided into smaller clusters by finding the least similar data points, or by agglomerative clustering - the more common method - where data points are all considered as individuals and then iteratively grouped together

based on similarity until all datapoints are grouped into one large cluster. Similarity was calculated as the euclidean distance between clusters as defined by eqn. 4.2, and clustering was performed using the Ward variance minimisation algorithm as described in eqn. 4.3 as the linkage method. This linkage method chooses pairs of clusters to merge based on which pair will produce the minimum within-cluster variance once merged [46].

$$d(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (4.2)$$

where:

$$\begin{aligned} d(q, p) &= \text{Euclidean distance between datapoints } q \text{ and } p \\ n &= \text{Dimensions of datapoint - i.e. number of timesteps recorded in neural signal} \end{aligned}$$

$$d(u, v) = \sqrt{\frac{|v| + |s|}{T} d(v, s)^2 + \frac{|v| + |t|}{T} d(v, t)^2 - \frac{|v|}{T} d(s, t)^2} \quad (4.3)$$

where:

$$\begin{aligned} u &= \text{The newly created cluster consisting of the linkage of the } s \text{ and } t \text{ clusters} \\ T &= |v| + |s| + |t| \end{aligned}$$

Agglomerative clustering was selected over the common K-means clustering method due to the lack of prior knowledge required for hierarchical clustering. When K-means clustering is performed the user must input the number of expected clusters prior to use [45]. Hierarchical clustering allows the user to see the structure of their dataset prior to making assertions on the number of ensembles. The main disadvantage of hierarchical clustering is its computational complexity relative to K-means clustering, however with the relatively small datasets found in this project this was never an issue [45].

There are many hierarchical clustering linkage methods, however the 'Ward' method was chosen due to its better performance in data with small numbers of observations, theoretically allowing it to detect rare ensembles containing few neurons. It also performs better in general than other linkage methods, however it can be slightly biased towards generating smaller clusters when they would be more appropriately combined into a single group [47].

Hierarchical clustering was implemented in Python using the linkage function from the cluster package of the scipy library and the results plotted using the dendrogram function from the same library.

5 Results and Discussion

The various methods used and the overall effectiveness of the developed pipeline is evaluated here. Additionally, the data generated is used for a study of novel ensembles in immobilised Hydra and machine learning tools are used to confirm the role of CB neurons in contraction. These results provide justification for the proposed techniques and present the biological findings made with them.

5.1 Selecting an Imaging Methodology for Single-Cell Resolution Recordings

Using Multiple Strains of Hydra, three imaging modalities - two-photon microscopy, Single Plane Illumination Microscopy, and confocal microscopy - were compared and contrasted in order to select a method best suited to the particular challenges of imaging Hydra. Each method had to be capable of two colour imaging, since this was the basis of the single-cell tracking method. The main considerations here were the method's field of view, signal to noise ratio, and contrast when imaging at 10Hz. Each had distinct advantages and disadvantages which will be discussed in this section.

Widefield imaging was not considered for single-cell resolution recordings due to a lack of optical sectioning ability. This reduced image contrast and caused point tracking algorithms to fail, rendering automated neuron tracking to be impossible.

5.1.1 Two Photon Microscopy

Two photon microscopy is an incredibly powerful method of imaging structures deep within tissue. The two-photon scanning methods available allow for a large FOV to be selected. However, since these microscopes operate on a laser scanning principle, and the use of multi-photon excitation inherently is less efficient than single photon excitation, even with very high powered lasers it was nearly impossible to obtain any signal

from GCaMP animals while using a 10x objective. Any objective with a higher magnification would not allow the full animal to be in the field of view and the signal was still less than that found when using confocal microscopy.

Initially, the two photon microscope was tested on an animal expressing GFP in its neurons. This has a far stronger signal than GCaMP7 and allows for an easy test of the microscope's capabilities. As seen in Figure 5.1a, the resolution achieved at 10x was excellent and the SNR was high enough to clearly distinguish neurons throughout the body - which is challenging on confocal particularly around the centre of the animal - and at 25x not only neurons, but the axons connecting them can be distinguished clearly.

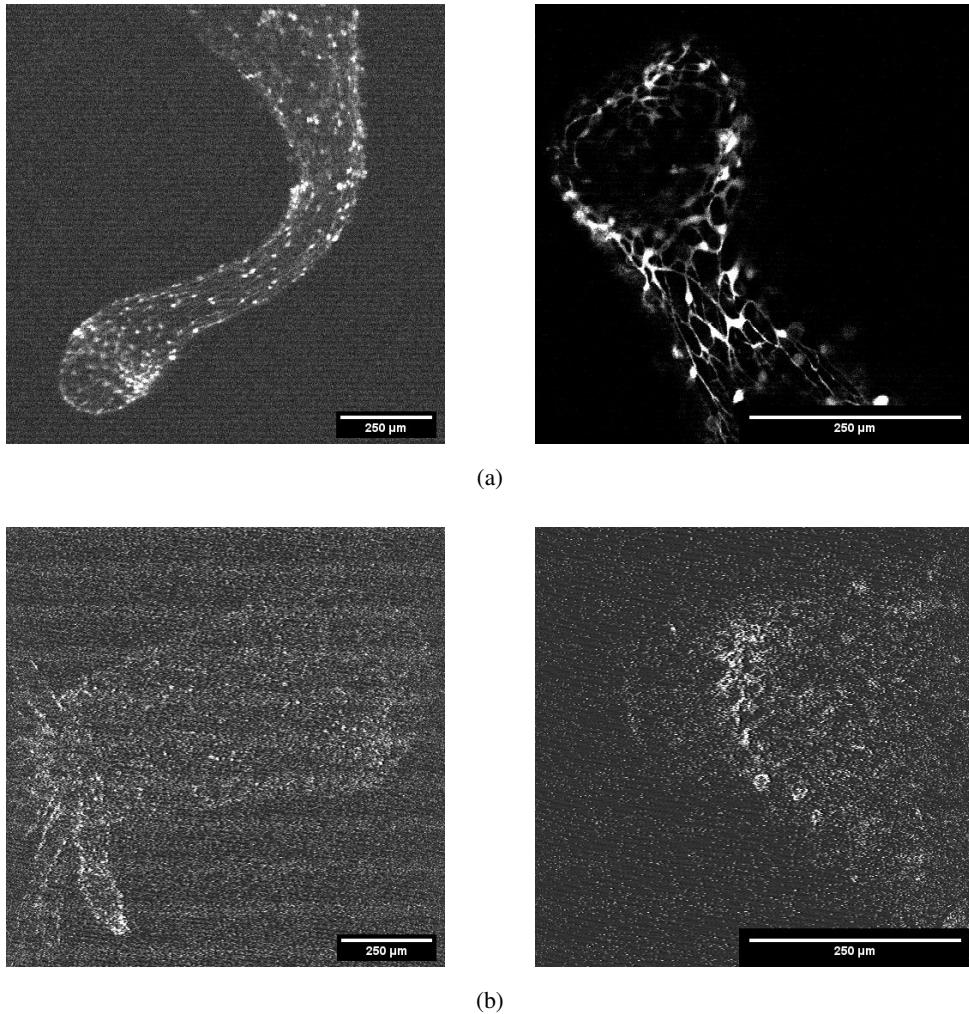


Figure 5.1: Images obtained using two-photon microscopy. (a) GFP animals imaged at 10x (left) and 25x (right) ; (b) GCaMP7-tdTomato animals imaged at 10x in the red channel (left) and 25x in the green channel (right). Scale bars are 250 μ m.

The performance of two photon microscopy on GFP animals appeared incredibly promising, however when tested using the GCaMP7-tdTomato animals it became clear that the reduced efficiency of multi-photon stimulation was a large hurdle to overcome (Figure 5.1b). While the tdTomato signal could be slightly discerned at 10x, the corresponding GCaMP signal was entirely invisible. Even recording through the 25x objective the signal from the GCaMP neurons was too poorly resolved and weak to be useful for analysis and the reduced field of view from the larger magnification meant that simultaneous behaviour recordings would be impossible.

5.1.2 Selective Plane Illumination Microscopy

SPIM has many of the same advantages as two photon microscopy. It has excellent optical sectioning capabilities due to the selectivity of its illumination source (opposed to confocal microscopy, which achieves optical sectioning from discarding out of focus light at its detector), and allows for a large field of view. However it also has some additional advantages. Particularly - due to the fact that SPIM does not rely on multi-photon excitation - it can use a far cheaper laser source for stimulation while having greater excitation efficiency. This allows for a SPIM system to generate images with a large FOV and excellent SNR.

A second advantage, particular to the microscope used, is its ability to image two planes simultaneously without the need to penetrate the depth of the sample. This is particularly advantageous in Hydra as it allows the user to image the top and bottom of the animal - a feat that is normally obstructed by the opaque gastric cavity in its centre. Figure 5.2 shows the top and bottom of a hydra recorded in this manner, however it should be noted that the objective used for the top recording was of a lower quality than that of the bottom, resulting in a lower resolution image.

As seen in Figure 5.2, this imaging methodology has the widest FOV while maintaining a trackable resolution. The single plane selectivity allows for clearer neurons to be observed in the body of the animal. Yet despite these positive factors, the main issue to overcome was the need for immobilisation of the sample. Since the light source comes from the side of the sample, the use of the previously described preparation [3] would no longer be applicable as the coverslip edge would refract the illumination light. This means that most SPIM set-ups instead immobilise a sample in agarose and suspend it below the objective in water. Since agarose has the same refractive index as water, the preparation will not distort the lightsheet [32]. This makes imaging of behaving samples difficult, and although efforts were made to develop a preparation made from FEP - a polymer with the same refractive index as water - this was ultimately abandoned due to time constraints.

The final challenge encountered when using SPIM was the large distance that the illuminations source needed to penetrate to illuminate the entire animal. Since illumination is delivered from the side, it must now travel through the width of the animal (over 250 μ m) opposed its far smaller depth (120 μ m). The effects of this can be seen in Figure 5.2, where the lower half of the animal is brighter than the upper half.

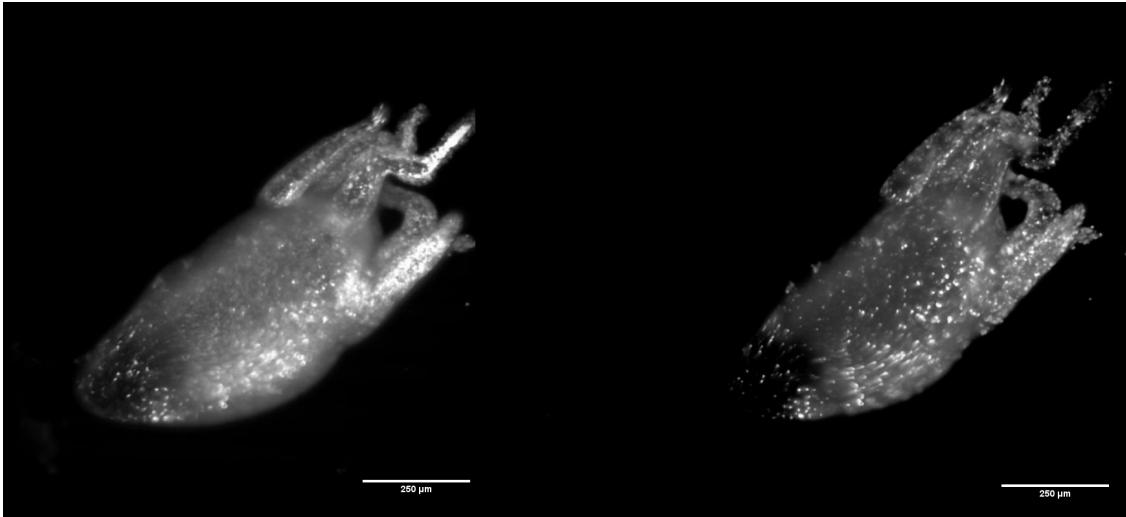


Figure 5.2: Images obtained using SPIM. Showing both the top plane (left) and the bottom plane (right) of the animal. Scale bars are 250 μ m.

5.1.3 Confocal Microscopy

Confocal Microscopy was the simplest and most robust of the methods tested. The spinning disk set-up described previously allowed for fast image acquisition and strong signal to be obtained with a 10x objective. This is achieved at the cost of the microscope's FOV which is limited by the attached spinning disk. As a result confocal microscopy had the smallest FOV of the methdos tested. This was the major limiting factor of confocal microscopy as the entire animal could not fit within the FOV, however this is a solvable problem. The use of lower magnification, high NA objectives combined with a larger FOV spinning disk attachment should increase the FOV enough to easily image the entire animal. Despite this limitation, the spinning disk confocal set up suffered from none of the major hurdles experienced by SPIM and two photon microscopy making it the best solution available. The preparation was simple and the signal received at low magnification was the best of all tested methods.

From Figure 5.3 it can be seen that, when imaged under the spinning disk confocal microscope, it was possible to resolve clearly both tdTomato and GCaMP7 labelled neurons with excellent SNR in the foot of the animal, and slightly reduced SNR in the centre of the body. This is due to the higher auto-fluorescence in

this region and somewhat less effective optical sectioning than other imaging methods due to the lower NA of the objective being used.

Individual neurons are easily resolved in the tdTomato channel (Figure 5.3b), allowing for automated tracking using conventional algorithms, and the neurons display clear fluorescence in the GCaMP channel when active - allowing for the recording of neural activity.

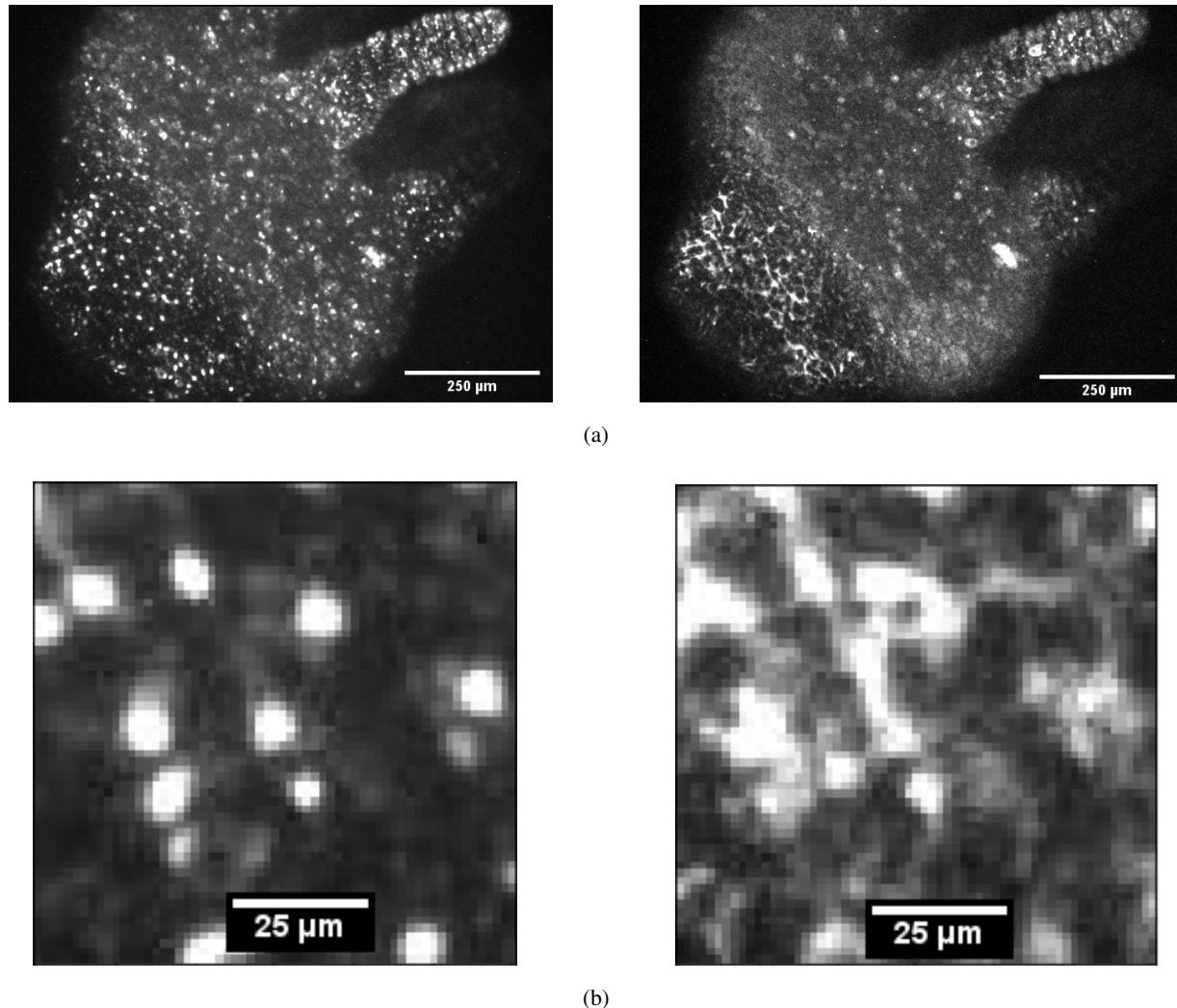


Figure 5.3: Images obtained using confocal microscopy. (a) Recording from the tdTomato channel (left) and corresponding recording from the GCaMP channel (right); (b) close-up view of tdTomato labelled neurons (left) and GCaMP7 labelled neurons (right). Scale bars are 250 μm for (a), and 25 μm for (b).

5.1.4 Comparison of Methods

Although SPIM has many advantages over confocal microscopy, the difficulty of the animal preparation, its dependence on immobilised animals, and the higher resolution images obtained using confocal imaging made it an unsuitable choice in the short term. With the development of simple and robust preparation methods, this could be an ideal choice for imaging hydra. Instead, confocal microscopy was chosen as the method to use moving forwards with imaging neural activity and behaviour.

5.2 Recording and Quantifying Behaviour

In order to investigate the relationship between neural activity and behaviour, it was critically important to obtain a quantitative representation of Hydra movement. DeepLabCut proved to be a powerful tool for tracking selected features throughout a video.

5.2.1 Tracking Motion with DeepLabCut

Figure 5.4 shows the effectiveness of the program's tracking of the head, centre, and foot of a Hydra - shown in blue, green, and red respectively - throughout a widefield microscopy recording. The positions of these points allowed for further analysis of both the length of the animal and the angle of bending at its centre, which in turn allowed for automatic detection of behavioural phenotypes (Figure 5.5).

Tracking a Hydra using deeplabcut proved to be a considerable challenge. This is likely due to the deformability of the species, meaning that features such as the head become almost unrecognisable from contraction to elongation. This meant that a generally applicable model was considerably less accurate than custom models for each video. When this custom approach was taken the challenge became trivial.

5.2.2 Analysing Animal Behaviour

This ability to generate continuous and uninterpreted representations of movement provided a significant change from previous techniques used to analyse Hydra behaviour which allowed only for "off/on" representations of different behaviours [21]. Figure 5.5 depicts a continuous recording made from a behaving animal, showing how the length and angle parameters change over time. Also shown is the predicted behavioural phenotype over time, generated using a simple threshold on the length parameter. Although not a perfect representation, the tracking of the animal's length and bending allow for a robust preliminary investigation of behaviour. However, future work could expand the number of parameters tracked to uncover

more behavioural details. This information was used alongside the recordings of neural data to give a simple representation of the relationship between behaviour phenotype and neural activity.

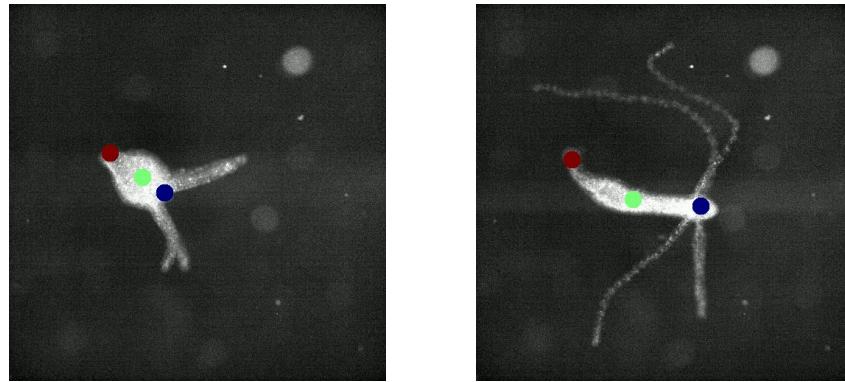


Figure 5.4: DeepLabCut tracking of the head, centre, and foot of a Hydra

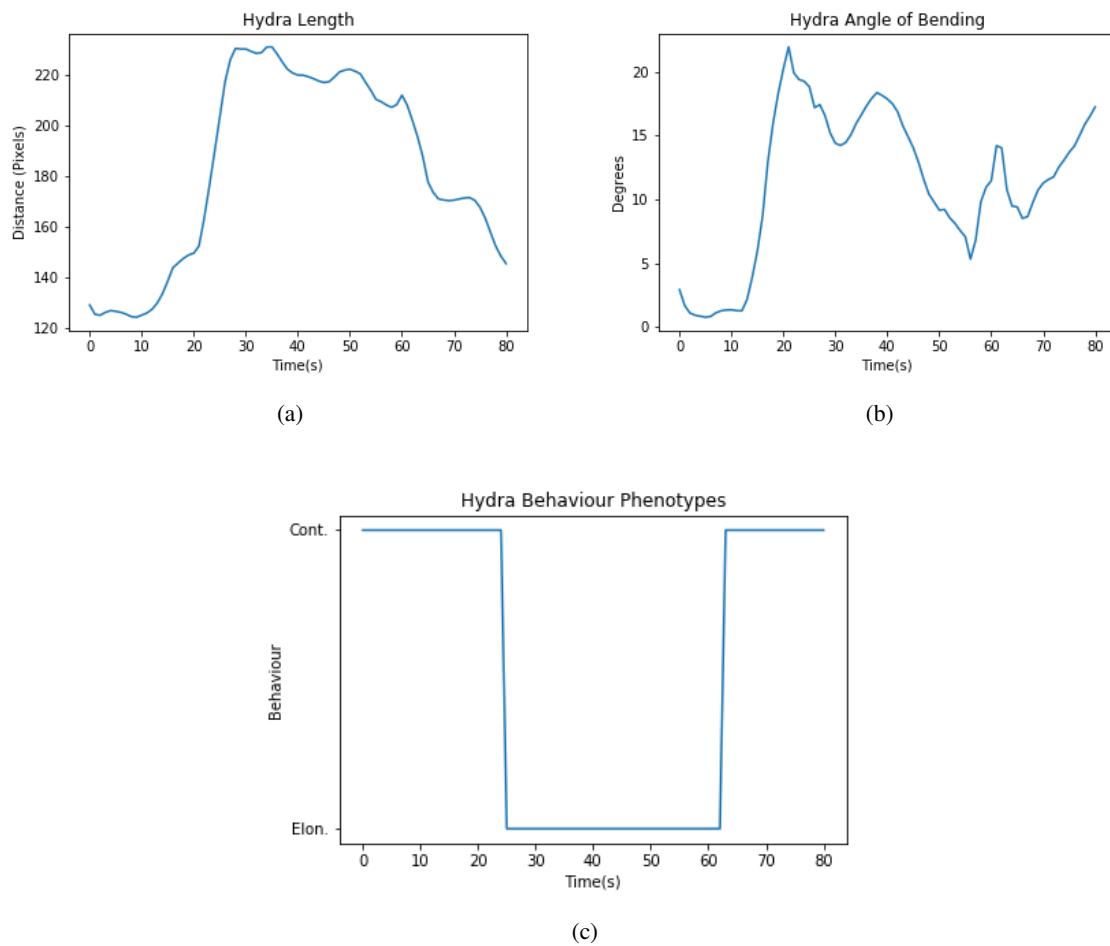


Figure 5.5: Continuous behaviour recordings and analysis using DeepLabCut data. (a) depict hydra length over time; (b) angle of bending at centre of Hydra over time; (c) contraction and elongation behaviours over time.

5.3 Neuron Tracking

The primary obstacle throughout this project has been identifying and tracking individual neurons throughout a recording as the animal moves and deforms. The first challenge was to track a point which is only visible some of the time. For most of a recording, neurons are all but invisible. Previous attempts to solve this involved computationally heavy tracking and performance was poor, especially for longer recordings. However, using genetic strains of Hydra which have a permanent fluorescent marker in their neurons - tdTomto in this case - it is possible to track neurons far more reliably (Figure 5.6).

However, this method leads to a second hurdle, the red and green channels are not perfectly aligned. To rectify this, a further tracking step was used to search the area surrounding the tracked point in the red channel and identify the neuron in the green channel, then follow it throughout the recording.

5.3.1 ICY Evaluation

ICY software allowed for the tracking of neurons from their GCaMP signal alone. While the software is able to make an attempt at following these points, it is far more effective to use the software to track the tdTomato signal and use the information from this to follow the neurons in the GCaMP channel.

A comparison between the effectiveness of the tracking of neurons using both methods can be seen in Figure 5.7. From this figure it can be seen that tracking using tdTomato allows for over double the number of points to be tracked than using GCaMP alone. The number of points tracked for the full duration of the video drops as number of frames increases which is expected since once a point is lost there are no new points being found to replace it. What is slightly unexpected is the increase in performance of the GCaMP tracking relative to tdTomato tracking as frames increase, until frame 500, where it begins to worsen again (Figure 5.7b). Due to the method used to track these neurons, a longer time period of recording allows a higher chance of a neuron activating - allowing the algorithm to link these points together. If the neuron does not activate again during the recording then the software may not detect it again and consider it lost. It appears that there is an optimal video length of between 100 and 250 frames where the GCaMP tracking works best, meaning there are enough frames to see neurons activate multiple times; yet not so many as to lose more neurons than are gained. However, even at optimum GCaMP tracking performance, tdTomato tracking it is still able to track over double the number of points as this method.

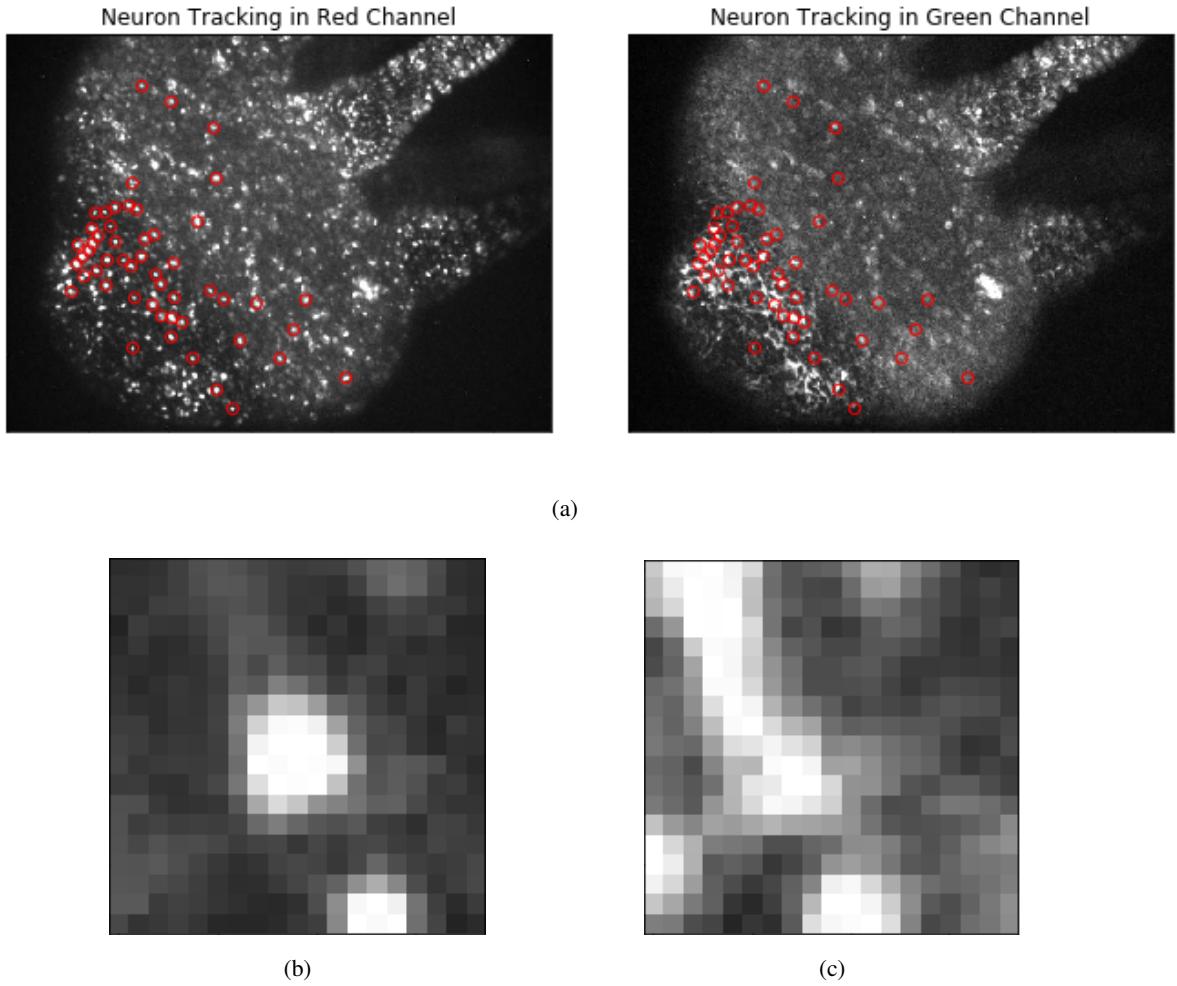


Figure 5.6: (a) tracked points in the tdTomato channel (left) and corresponding points in the GCaMP channel (right); (b) close-up view of a tracked neuron in the tdTomato channel; (c) corresponding view of the neuron in the GCaMP channel, showing imperfect alignment.

5.3.2 Sub-ROI Tracking

Tracking neurons using the tdTomato signal is not a perfect solution however. Although the labelling in red and green is similar it is rarely completely overlapped (Figure 5.6b, c).

To rectify this a second tracking algorithm was implemented to work with Icy and improve results. Figure 5.8 shows the effects of this algorithm when tracking a neuron throughout a video. In this figure the necessity of this algorithm becomes clear: if a large ROI was taken around the tracked point then it will include signals from multiple neurons, making single cell resolution recordings inaccurate and possibly misleading. On the other hand making the ROI too small would exclude signal from neurons - and since this distance between

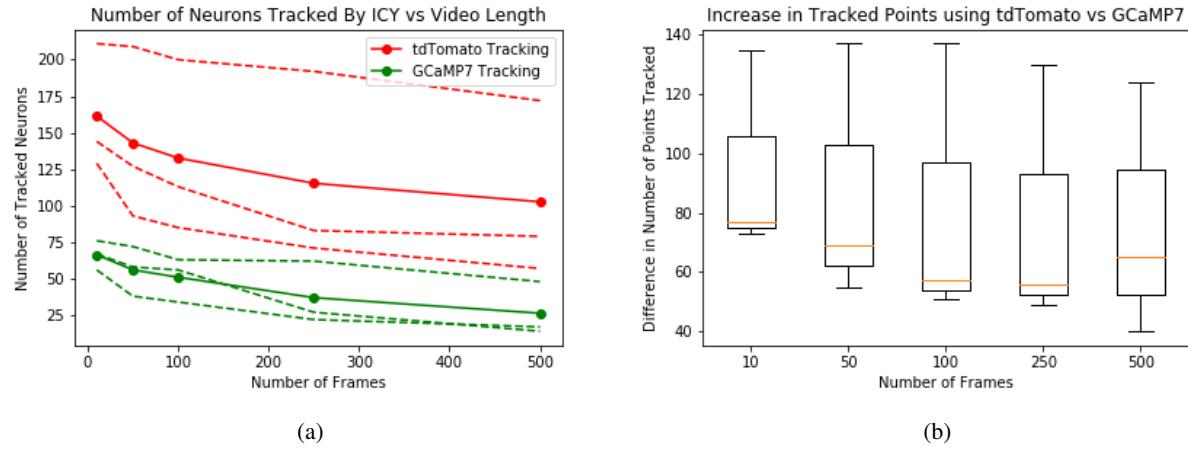


Figure 5.7: (a) Comparison of tdTomato and GCaMP tracking with Icy over differing video lengths, 5 lengths of 3 different videos were used for comparison and average performance found; (b) boxplots of the improvement found using tdTomato at different numbers of frames.

the red point and the green point varies from cell to cell - it is impossible to select a single ROI size that will be appropriate for every cell. Instead the algorithm selects an area the size of a neuron from anywhere inside a larger ROI and uses this area for extracting the signal from the neuron. If this subROI tracking algorithm was not used, the frame shown in Figure 5.8 would be recorded as a spike due to the activity of a nearby neuron.

There are two distinct advantages to this approach. First, as discussed previously, it will exclude signal from other neurons in its immediate proximity allowing for true single-cell recordings; second, the signal of the tracked neuron will have a greater SNR due to the fact that all of the pixels in the ROI will be brighter when a neuron is active as opposed to only a small portion of them, as would be the case when taking recordings from the full ROI. This effect is depicted in Figure 5.9 which shows the resulting signal extracted from the same neuron using both the full ROI and sub-ROI tracking methods. The effect of this method on the resulting rasterplot is particularly significant in this example, without the subROI tracking method this neuron may mistakenly be classified as an RP1 neuron when it is in fact a CB neuron. When using the subROI tracking algorithm there is clearly no functional overlap between the CB and RP1 neurons which is consistent with previous results from investigations into the neural networks of *Hydra Vulgaris* [3]. Without using this subROI tracking method it would appear that this neuron displayed activity from both ensembles, however this is simply an artefact from imperfect tracking.

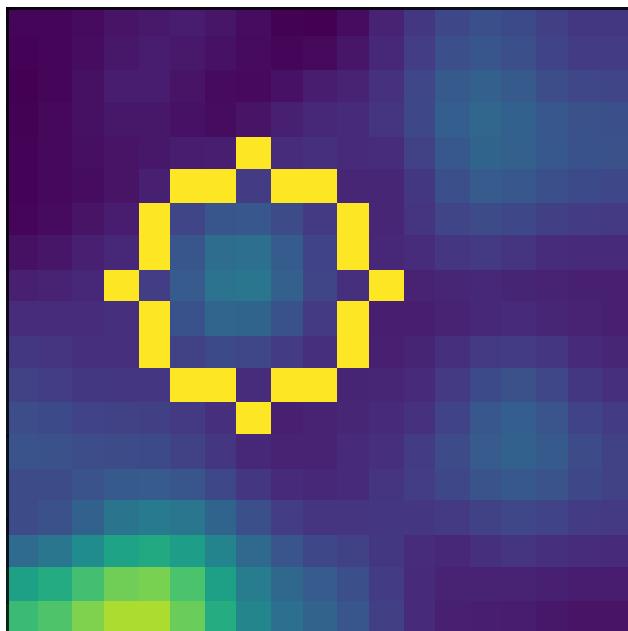


Figure 5.8: Effects of the Sub-ROI tracking algorithm. Tracked neuron is circled in yellow, and the activity of non-target neuron is seen in the bottom left corner.

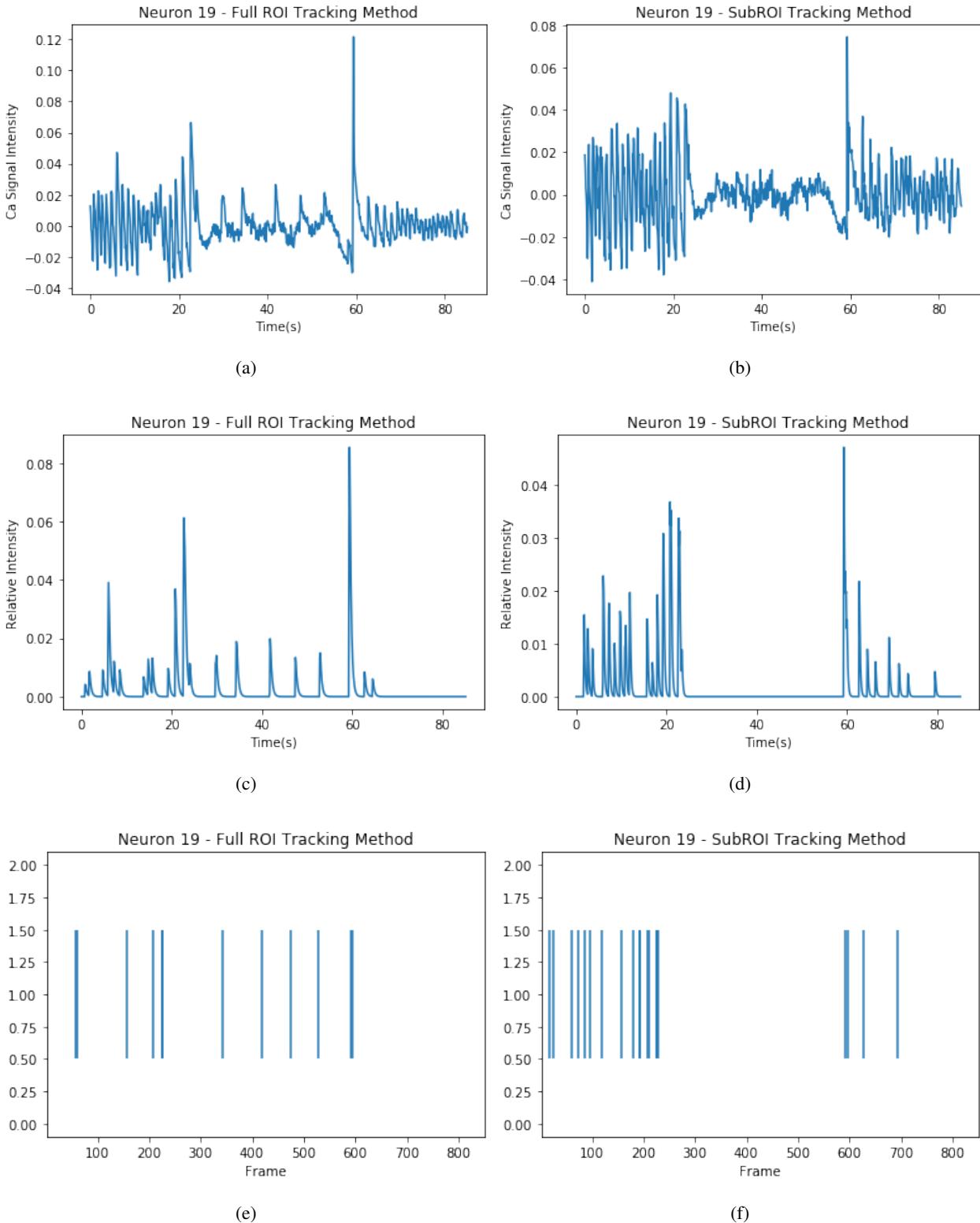


Figure 5.9: Effects of SubROI tracking algorithm on recorded neural signals. (a), (c), (e) show signals recorded using the algorithm and (b), (d), (f) show signals recorded without. (a), (b) show raw signals; (c), (d) denoised signals; (e), (f) rasterplots of deconvolved spike times.

5.4 Processing Neural Signals

Once the neural signals were extracted from the resulting ROIs, it became important to process the signals in a way that provided the most information about the underlying neural activity and was able to reject the maximum amount of signal form other sources such as inherent detector noise and motion artefacts. Figure 5.10 shows the necessity of processing the raw signals to improve the detection of neural activity.

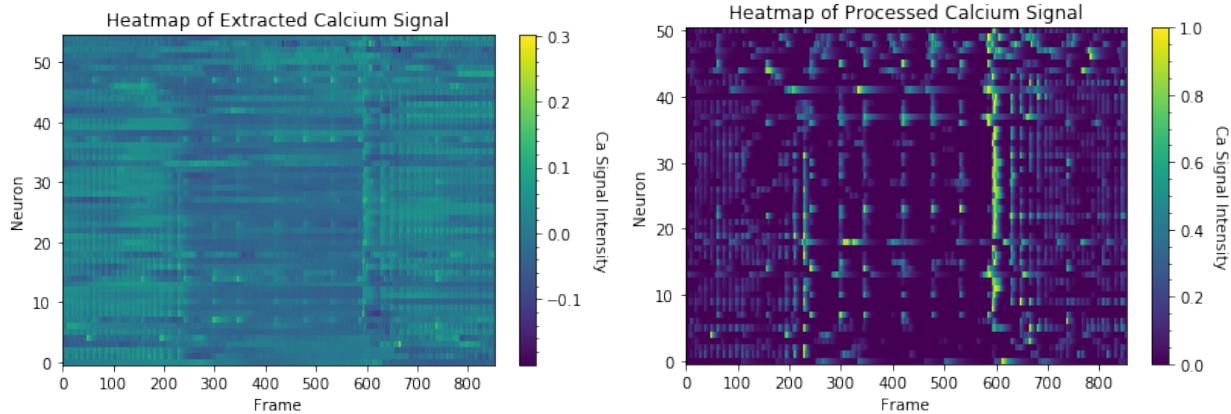


Figure 5.10: Effects of Processing on population neural data. Unprocessed heatmap of neural data (left) and fully processed heatmap of neural data (right).

5.4.1 Correcting Motion Artefacts

The first processing step consisted of removing the effect of motion artefacts from the signal. This was achieved by using independent component analysis to take advantage of the two recording (in red and green) in order to separate signals. This method has the added bonus of being immune to cross-talk between the green and red channels since it looks for separate signals common to both inputs. The effects of using the ICA algorithm to extract only signals resulting neural activity are shown in Figure 5.11.

Two main methods for signal extraction were tested and their effects compared in Figure 5.12: Ratiometric motion correction, and the previously described ICA algorithm. This figure demonstrates, on real data, a particular weakness in the ratiometric approach for the correction of motion artefacts. When used on data with a sharp drop in intensity, the algorithm will fail since points of very low values may still have a large ratio between them even if the absolute difference is relatively small. This can create an artificial spike in the processed data, interfering with analysis and creating inaccurate results.

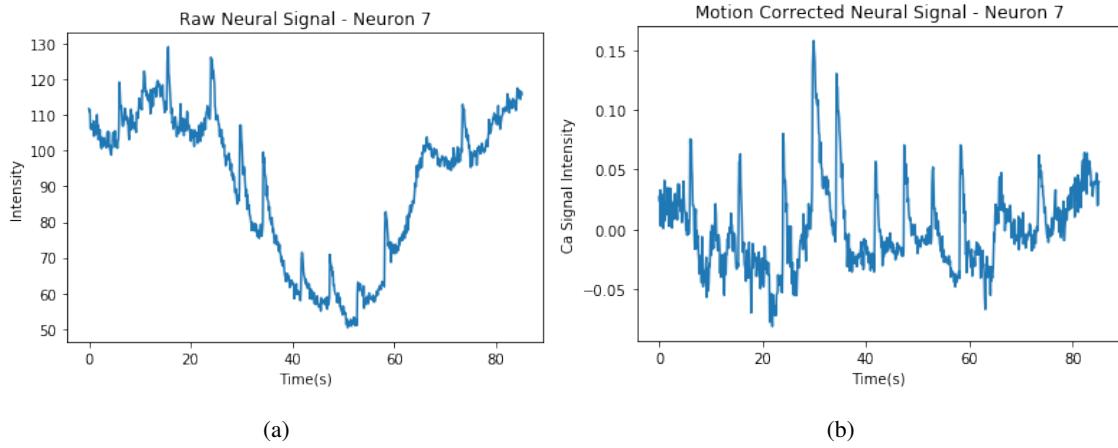


Figure 5.11: Extraction of calcium activity signal from raw fluorescence recording using ICA. (a) Raw fluorescence recording; (b) extracted calcium activity signal.

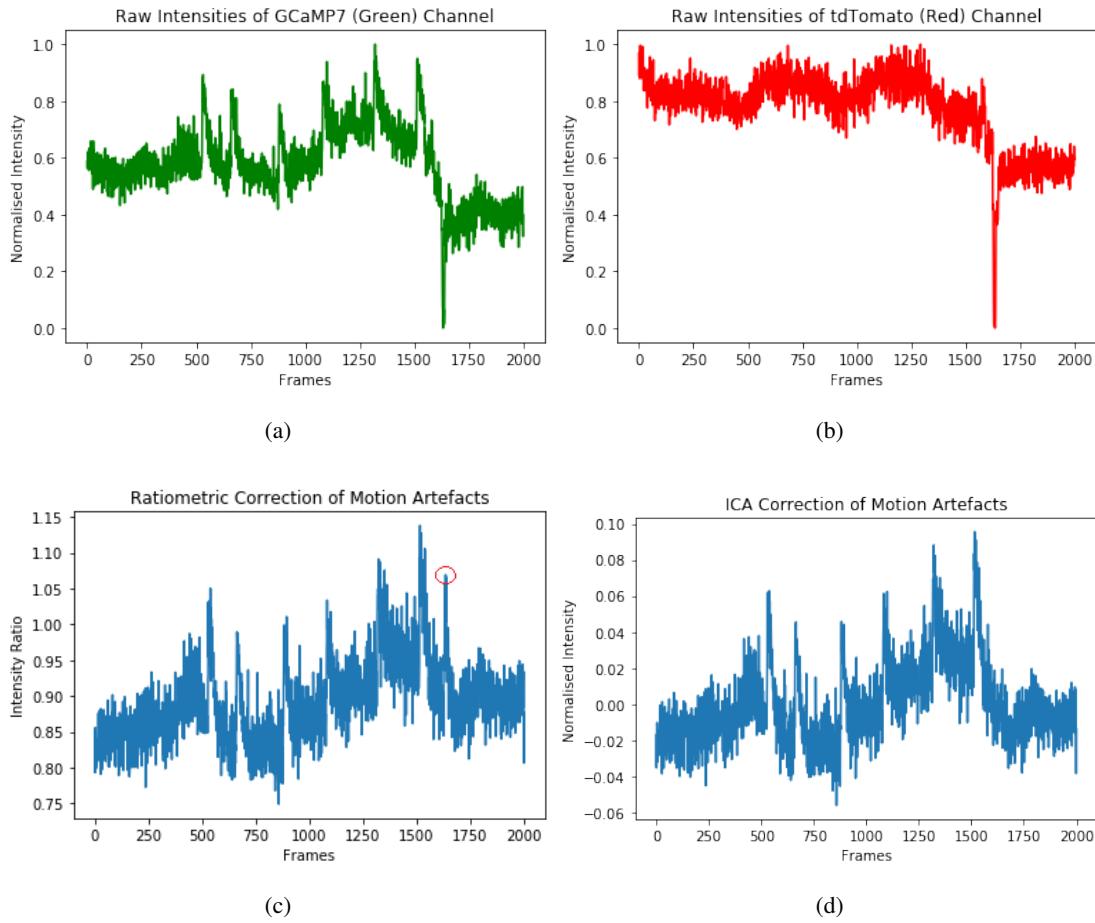


Figure 5.12: Comparison of ratiometric and ICA methods for motion correction. (a) Raw signal from GCaMP channel; (b) raw signal from tdTomato channel; (c) Effects of ratiometric correction, showing spurious spike circled in red; (d) ICA corrected signal.

From these results it is clear that ICA was the superior algorithm for separating artefacts from the signal. It should be noted that it is possible to avoid the shown issue with the ratiometric approach by adding a constant factor to the entire signal. However, this introduces issues of its own, decreasing the SNR. A second flaw with using ratiometric correction is that it is susceptible to any cross-talk between the red and green channels. If there is any amount of neural signal present in the red channel this will reduce the SNR of the resultant neural signal extracted with this method.

Finally, to completely remove any residual artefacts resulting from photobleaching or imperfect motion correction, a detrending step was performed. This step was crucial for the following stages of processing, allowing for the filtering of non-neuronal cells and denoising discussed later in this chapter. Detrending removed almost all remaining signal resulting from sources other than neural activity, allowing for more accurate analysis of activity. The effects of the detrending process are shown in Figure 5.13.

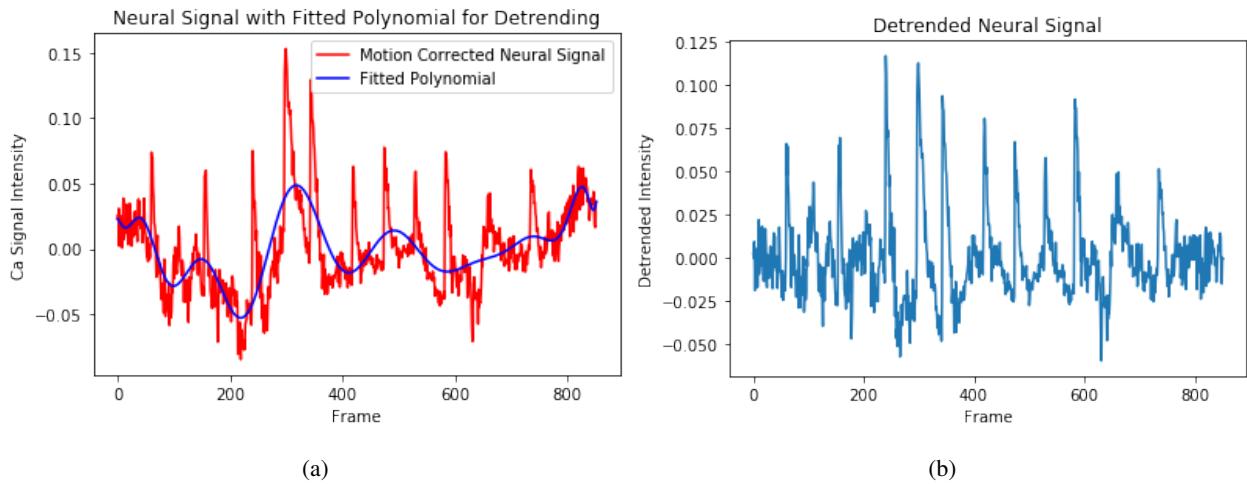


Figure 5.13: Detrending of a motion corrected neural signal. (a) motion corrected neural signal plotted with its fitted polynomial; (b) detrended signal achieved by subtracting polynomial from the motion corrected signal.

5.4.2 Filtering Non-Neuronal Cells

The following step removed any signals coming from non-neuronal cells from the analysis process. Since Cnidocytes and glands stem from the same cell lineage as the neuronal cells, they are labelled in the same way as the neurons. However, it is possible to detect and remove these cells during the analysis process. These signals were significantly different from those of neural cells due to the lack of any spikes. The effects of the Gaussian distribution filter described in chapter 4 are shown in Figure 5.14. These signals show that

this method of filtering was capable of identifying and removing non-neuronal cells with high specificity, the filter removed most non-neuronal signals with a very low probability of mistakenly removing a neural signal.

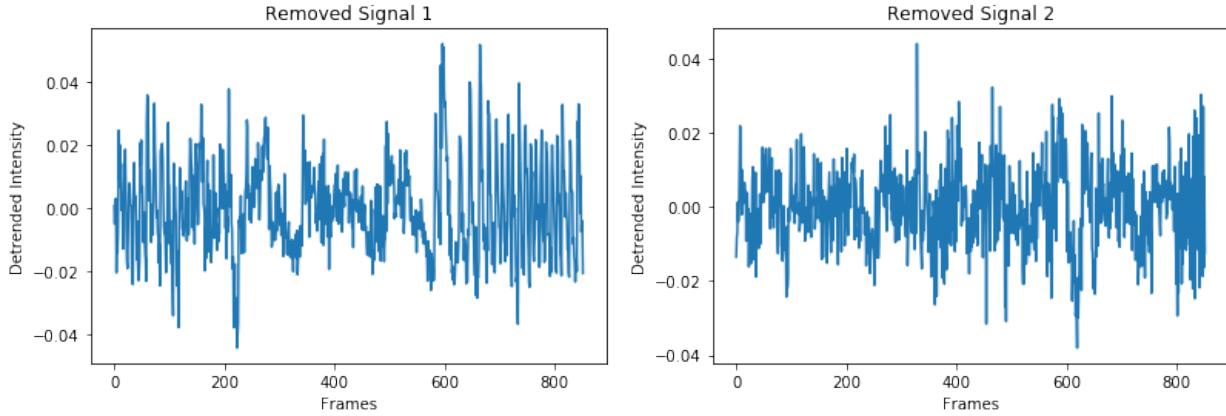


Figure 5.14: Example signals removed by the Gaussian distribution filter

From Figure 5.14 it can be seen that some removed signals do contain some level of neural activity, for example 'Removed Signal 1' appears to contain some signal consistent with the CB neurons' pattern of activity, however the SNR is so poor that this would be useless for analysis and filtering of this signal was deemed acceptable.

5.4.3 Denoising and Spike Deconvolution

Once neural activity signal had been appropriately extracted from the recorded videos, the final processing step was to extract the neural spikes from this data. Since calcium activity is far slower than electrical activity in neurons, it must be processed by complicated algorithms which have been developed to deconvolve this information.

Usually these algorithms can be verified using some form of ground truth data, unfortunately ground truth data for Hydra neural activity is incredibly difficult to acquire. So far there have been no successful attempts to collect single-cell electrophysiology data (i.e. via patch clamping) from a Hydra. This makes inferring the exact underlying neural activity of a cell impossible to perfectly verify. To mitigate this problem, analysis tools that have been robustly tested in more common organisms were employed.

As seen in Figure 5.15 the Foopsi method from CaImAn is able to completely denoise a signal (after the previously described motion correction and detrending) using an autoregressive model based on the dynamics of the GCaMP, and reject noise from other sources. Once this step is complete the algorithm can then infer

spikes from this data providing information on the time at which the neuron fired. Using a simple threshold these spike time probabilities can be used to generate a rasterplot for each neuron. This denoising and spike extraction method is performed on all recorded neurons. This method provided near-perfectly cleaned signals and accurate spike-times from every cell recorded from. These processed signals were relied on by the future correlation steps, and for attempting to link behaviour and neural activity.

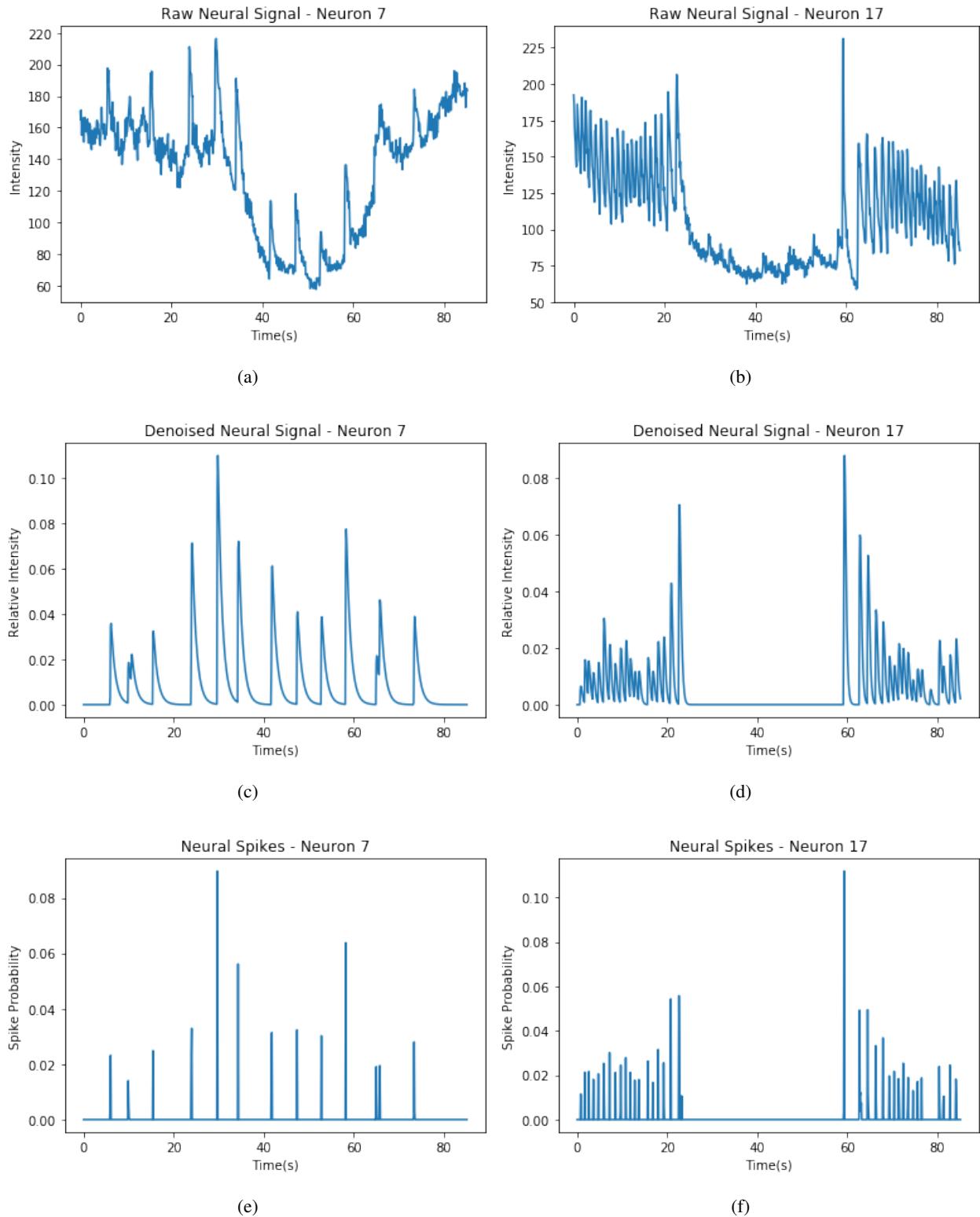


Figure 5.15: Denoising and deconvolution of both a CB and an RP1 neural signal. (a) RP1 raw signal; (b) CB raw signal; (c), (d) denoised calcium signal using CaImAn; (e), (f) inferred spike times from this signal.

5.5 Correlation and Ensembles

Once signals had been processed it became important for analysis to cluster them into groups to attempt to find both previously known ensembles [3], and to investigate the occurrence of novel ensembles. Hierarchical clustering was performed and the results shown in Figure 5.16. The height of the dendrogram represents the difference between clusters. Ensembles can be taken by 'cutting' the dendrogram at any height decided by the user.

The data set shown in Figure 5.16 contained three appropriate clusters, with the green and red coloured groups being merged into one as they both consisted of RP1 neurons. The large purple group contained CB neurons, and the small blue group contained signals with no discernible neural activity - possibly from non-neuronal cells. This method of clustering is also an effective method of isolating signals from non-neuronal cells that may have been missed by previous filtering steps.

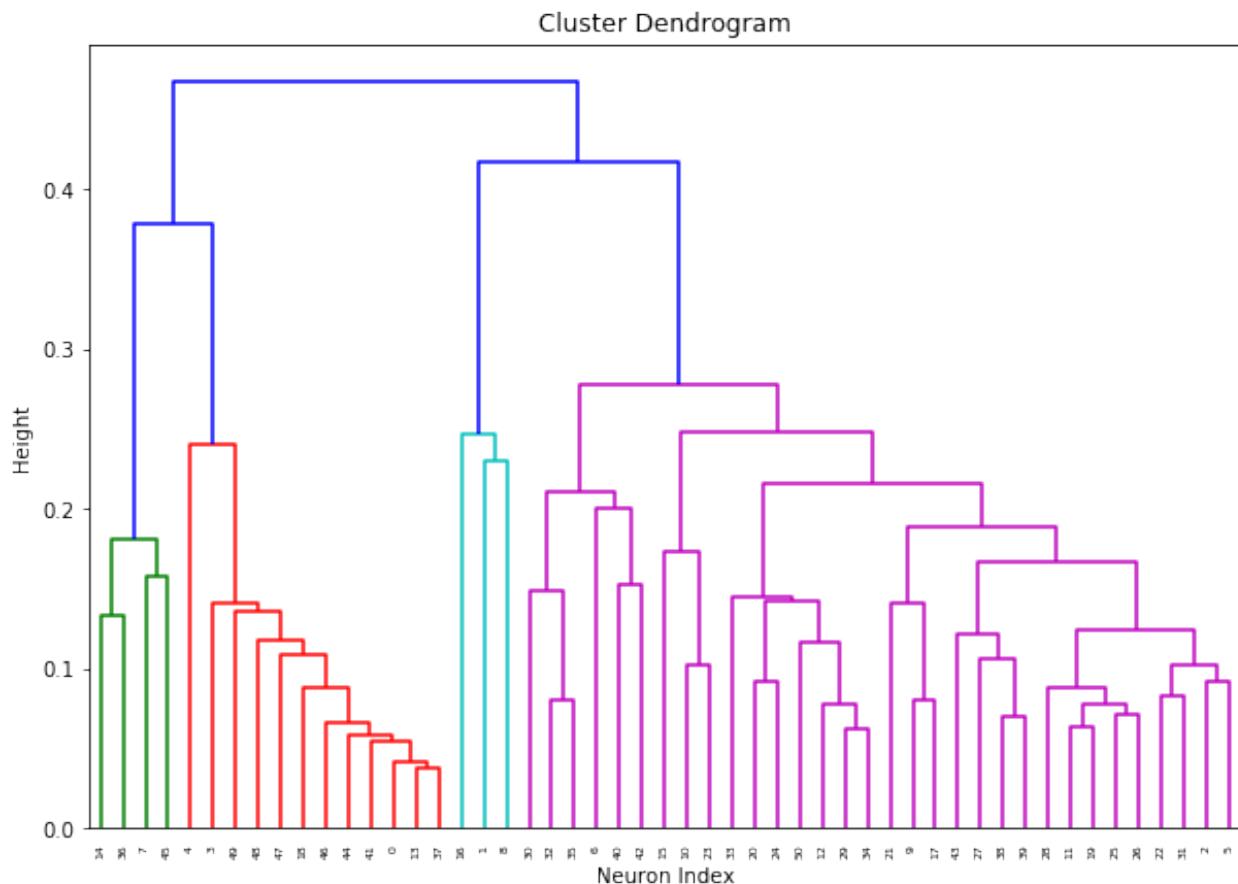


Figure 5.16: Cluster dendrogram showing relationships between neurons and clusters of neurons. Generated using the 'ward' method of clustering.

Once clustering was complete, individual ensembles were investigated and the positions of the neurons in said ensembles were displayed. Figure 5.17 shows the the ability of the script to investigate a cluster. Here, the RP1 ensemble has been identified from the dendrogram and used to plot the locations of all neurons in this cluster, as well as the activity of a representative neuron from the ensemble. This capability allowed for the rapid verification of ensembles as well as providing novel, detailed, spatial information about them.

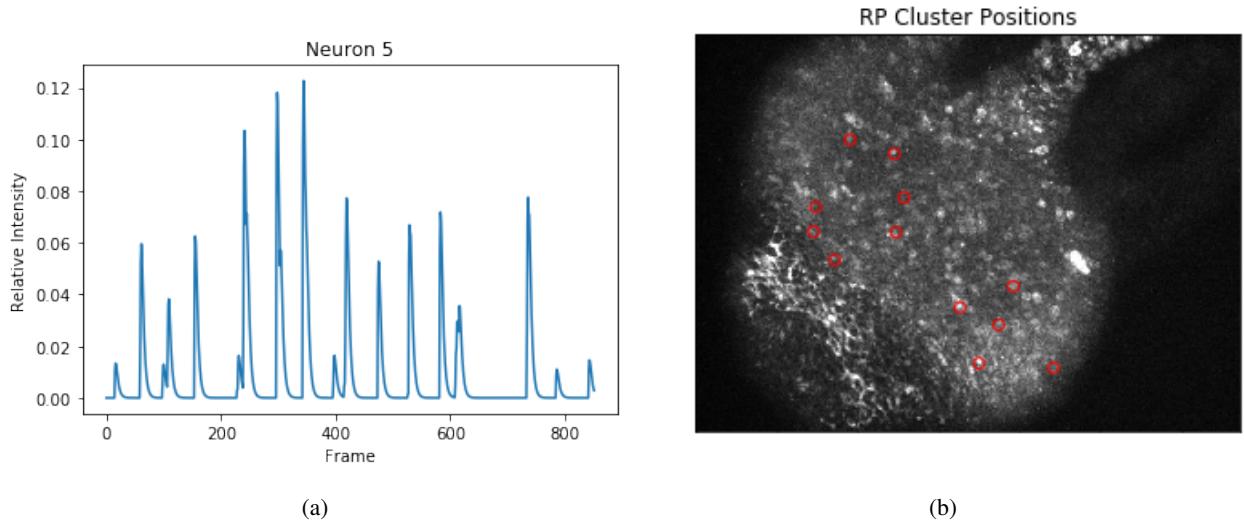


Figure 5.17: Investigation into the RP1 neuron ensemble. (a) Representative signal from an RP1 neuron in the analysed ensemble; (b) Spatial locations of neurons in the RP1 cluster circled in red

Correlation data was also displayed in the form of a correlation matrix, allowing for fast and clear interpretations of the underlying structure within the dataset investigated. In these matrices, ensembles of neurons appear as large red squares where the colour represents the strength of the correlation between the neurons within the network. These matrices allowed for simple visualisation of the correlations between every pair of neurons recorded from. This is shown in Figure 5.18 where the large square in the top left of the image represents the CB neurons and the smaller square to its lower right represents the RP1 neurons. This interpretation is particularly useful for identifying smaller ensembles that may be missed when using a dendrogram alone.

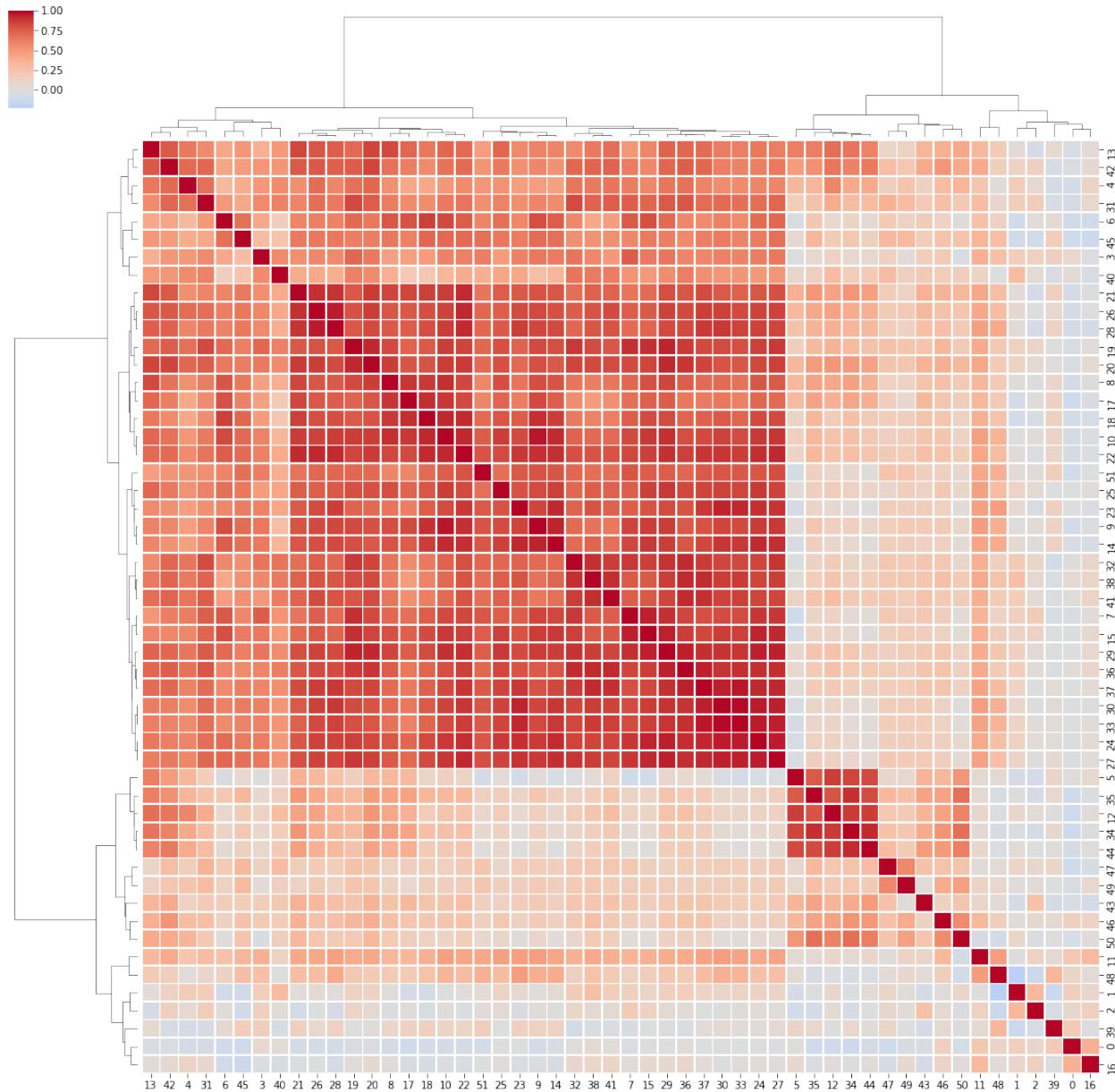


Figure 5.18: Correlation Matrix showing the correlations between all neurons in the dataset. Dendograms are plotted along the top and left sides, and neuron indices along the bottom and right sides.

5.6 Novel Neural Activity During Immobilisation

In order to evaluate neural networks across the entire animal, an experiment involving the immobilisation of a Hydra was performed. This immobilisation protocol was particularly effective at constraining the motion of the animal while allowing them to be safely released after imaging was complete with little to no damage to the Hydra. Figure 5.19 shows the lack of movement of an animal over 2000 frames (200 seconds) using this protocol. When compared to Figure 5.4, a drastic reduction of motion is seen, with neurons staying in almost the exact location they began in even over an extended time period.

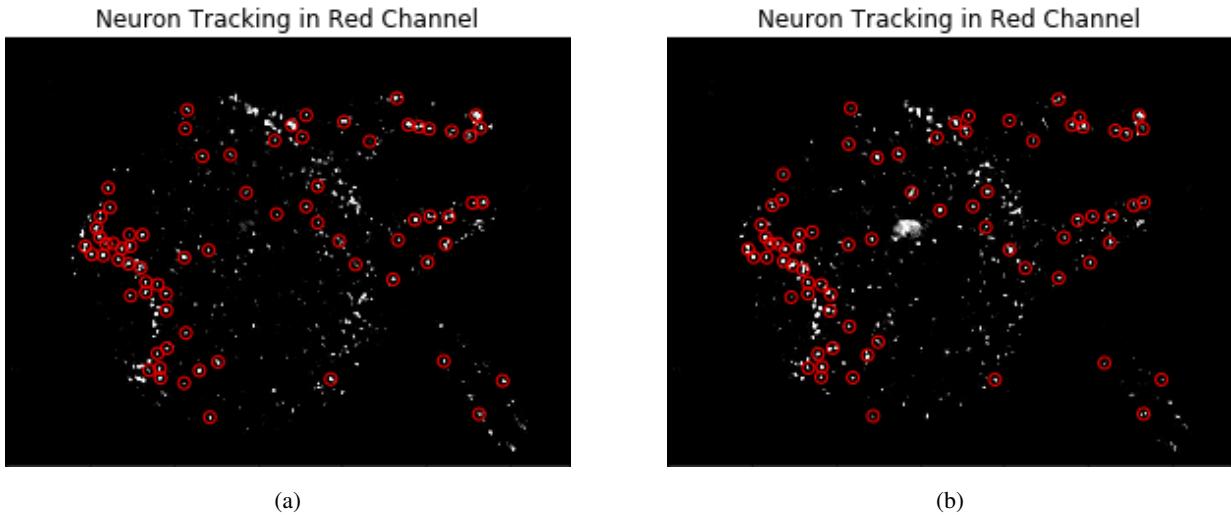


Figure 5.19: Motion of a Hydra in hydrogel. (a) Hydra with superimposed locations of neurons at frame 0; (b) Hydra at frame 2000, 200 seconds later.

Despite this behavioural restriction, neural activity continued and seemed to increase when compared to freely behaving Hydra. This is likely due to the increased mechanical stimulus provided by encapsulation in a stiff material exciting the sensory neurons of the animal. This effect was further compounded by the fact that an increased number of neurons are able to be tracked in an immobilised animal as the lack of motion increases the effectiveness of tracking algorithms. Figure 5.20 shows the structure of the data recorded from the immobilised Hydra. In this figure three clear neural networks can be seen as red squares. The largest square in the upper left represents the CB network and the smaller square in the middle of the figure represents the RP1 network. The structure then begins to diverge from what was seen in freely behaving animals as multiple smaller squares appear toward the lower right corner. These represent the many smaller networks - like those described in Figure 5.21 - which were recorded in immobilised animals. By comparing these results to Figure 5.18 it becomes clear that during immobilisation, Hydra appear to display more varied

neural activity, with multiple small groups of neurons acting as ensembles as opposed to the three large networks described previously [3].

The entire animal was able to be imaged during this preparation as the Hydra was immobilised during its contractile behaviour. This meant neurons across the entire animal were imaged, allowing for investigation into the neurons in the head and tentacles of the animal. This had proven challenging before as movement of the head often caused it to leave the FOV. An additional benefit to using this preparation was the ability to track neurons over far longer timeframes. This allowed for the detection of the RP2 network which has a far slower firing rate than the other networks [3].

These networks are shown in Figure 5.21, where the location and activity of the neurons comprising them are displayed. Most interestingly is the new tentacle network which can be seen in Figure 5.21a and c. This network seems to fire consistently throughout the recordings yet is not part of the RP1 network. Although there is little literature describing the function of these tentacle networks, it is possible that they form a sensory ensemble which was more consistently stimulated while the Hydra was encapsulated than it would be during free behaviour.

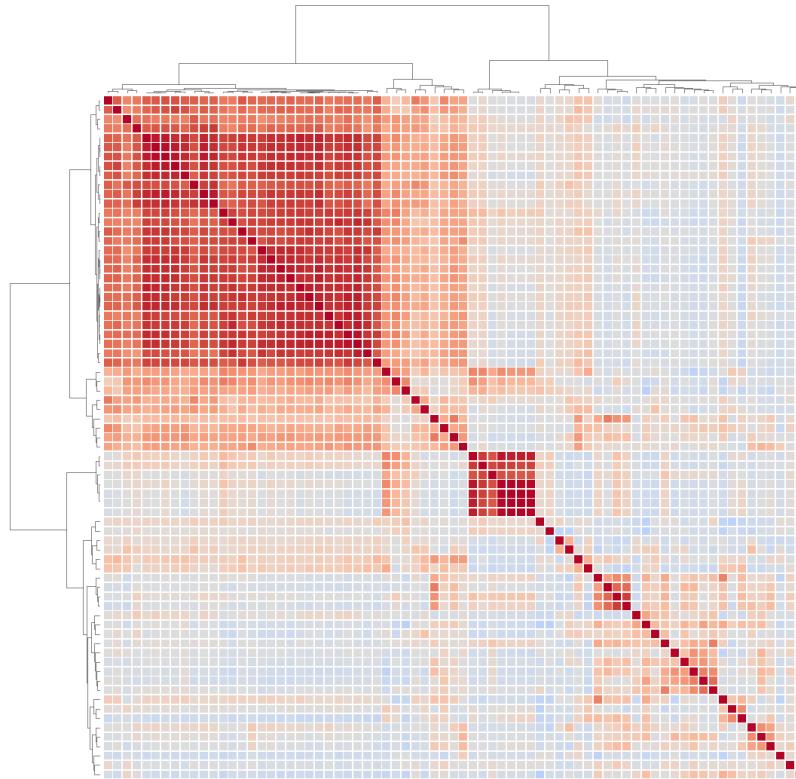


Figure 5.20: Correlation matrix showing the correlation between neurons in an immobilised Hydra.

Although this approach could not record the behaviour of an animal alongside the neural activity, it provided interesting preliminary insight into the neural networks in Hydra, which may later be studied in a behaving animal.

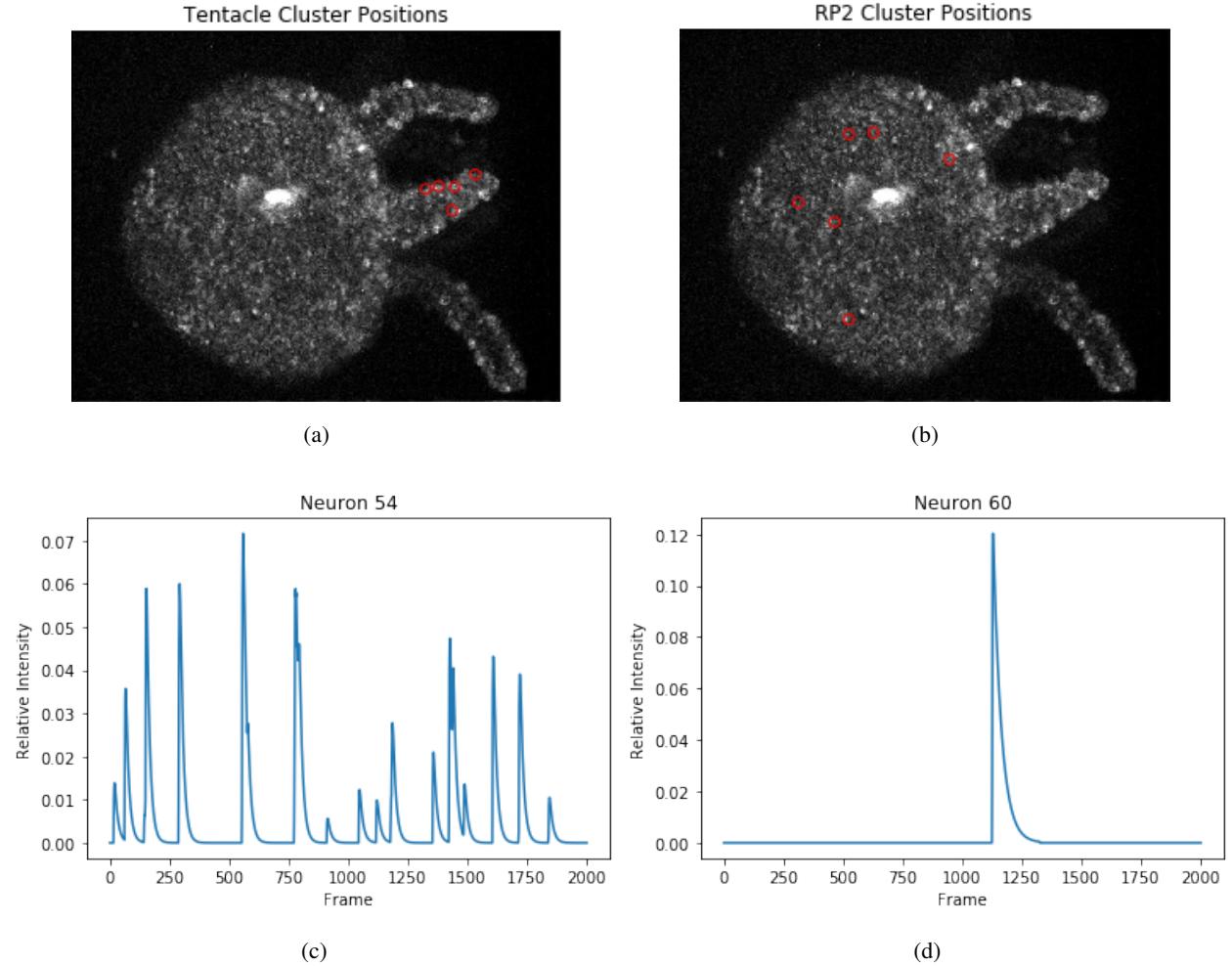


Figure 5.21: Ensembles detected in immobilised Hydra. (a) Locations of tentacle ensemble; (b) locations of RP2 ensemble; (c) representative activity of tentacle ensemble neuron; (d) representative activity of RP2 ensemble neuron.

5.7 Connecting Neural Activity and Behaviour

Figure 5.22 shows the ability of the described methodology to extract and quantify neural activity in the form of a rasterplot, while simultaneously recording and analysing the behaviour of a Hydra. This allows a completely novel dataset to be generated containing information which completely describes the motion of an animal along with the activity of most of its neurons.

However, once both neural activity and behaviour data had been extracted from the raw videos, the challenge became interpreting the link between them. Although there is a clear connection between the high frequency CB neurons and the contractile behaviour seen when they fire, there is little information on their exact function or that of the RP1 or RP2 neural networks that seem to relate to more complex behaviours. In order to attempt to gain insight into this relationship, powerful deep learning tools were employed.

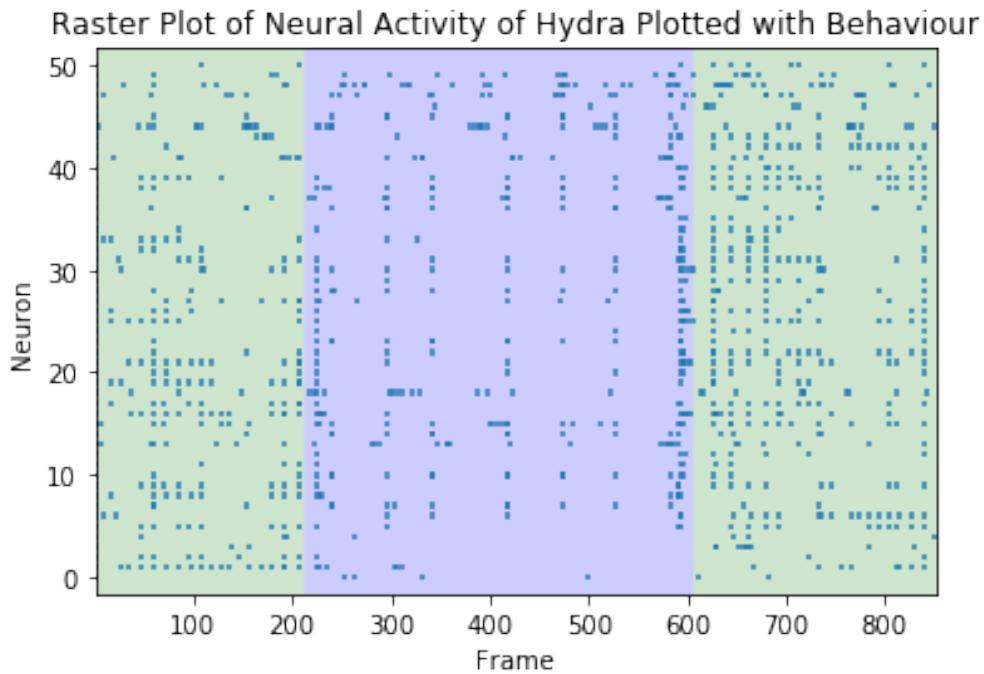


Figure 5.22: Raster plot of neural activity for all recorded neurons with contraction behaviour plotted in green and elongation plotted in blue.

5.7.1 Machine Learning Methods to Predict Behaviour from Neural Activity

The approach taken to uncover this relationship was to build a predictive model connecting neural activity to behaviour. This proved difficult using conventional techniques, so a novel deep learning approach was taken. Initially a simpler machine learning method was employed - a random forest model - however this proved insufficient for the complicated time-series data presented to it.

An LSTM model was built and trained using the data recorded from the neural activity analysis script and the binary behavioural data collected using Deeplabcut. The model was tested using both the denoised calcium signals, the raw fluorescence data, and the binary rasterplot data. It was found that the rasterplot data gave the best performance. The results of this model can be seen in Figure 5.23 showing that the model was

capable of predicting behaviour up to 50 frames before it occurred. This implies that there is information within these neural signals that encodes the planning of behaviour.

This method had a crucial weakness however. The limitations of the LSTM architecture means that the feature array - in this case the signals from the neurons - needed to be the same size and in the same order for every dataset. This would mean that the same number of neurons would need to be recorded and arranged in some similar manner for every Hydra that would be analysed with the resulting model. Instead, this model was both trained and tested on the same dataset from the same Hydra. The data was split in two, with one half being used to train the model and the other to test it.

While this model was effective at predicting the behaviour of the Hydra it likely suffered from over-fitting as only a single data set was used for training and testing. This, combined with the fact that it could not be used across different recordings were significant issues that require further development to solve. A further problem with deep learning itself is the lack of tractability of the algorithm used. Even if it was a perfect predictor of the behaviour of the Hydra it would be challenging to gain insight into what underlying biological factors it was using to make its predictions, so little understanding of the system could be gained.

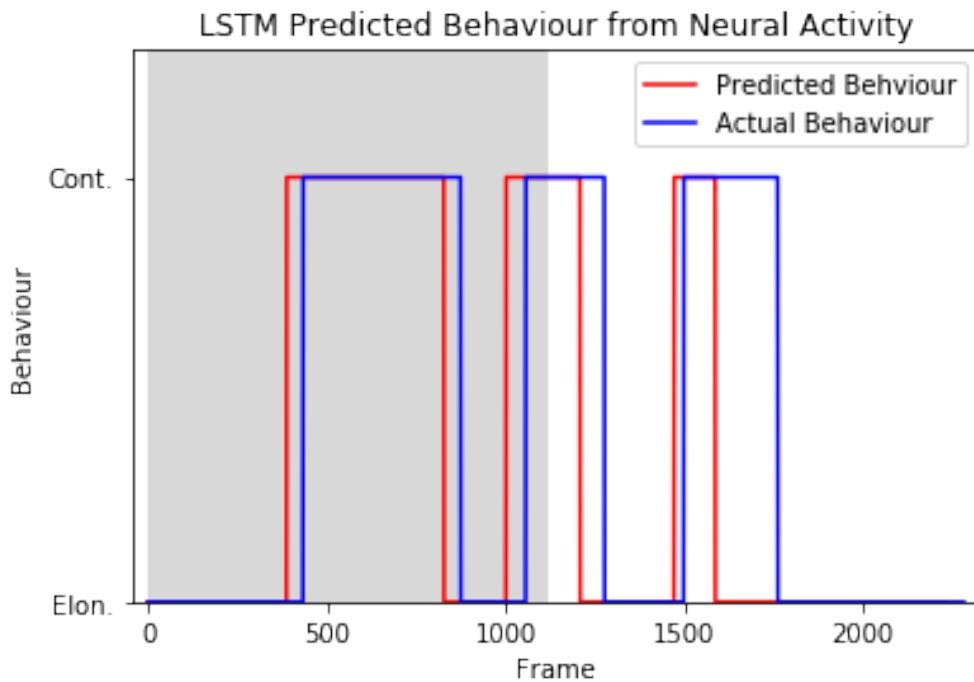


Figure 5.23: Plot of actual Hydra behaviour with the behaviour predicted by an LSTM model using only neural signals. The grey section indicates the portion of data used for training, and the white section represents data used for testing.

5.7.2 Assessing Feature Importance Through Sensitivity Analysis to Investigate Underlying Neural Mechanisms

A partial solution to these problems involved the use of sensitivity analysis [23] to extract the feature importance value for each neural signal presented to the model. This allowed for two major improvements. First, it is possible to use these importances to order the input data in the same way across multiple datasets by organising them based on importance. Once this importance was found for each input the model rank them from most to least important, and use them to predict behaviour. Second, this allowed rudimentary insight into the logic of the LSTM's decision making process. The predicted importances of select neurons found using sensitivity analysis can be seen in Table 5.1.

Insight into the workings of the system is gained by allowing the user to learn which signals were most important for predicting a certain behavioural phenotype. This can be seen in Figure 5.24 where the top two most important features predicted by this method are displayed. This suggests that the model primarily uses CB neurons to predict the contraction behaviour. This conforms with results found in previous studies which imply that CB neurons are responsible for contractile behaviour [3] and strongly suggests that this method is capable of providing insight into the underlying neural dynamics behind behaviour. In the future this method could be performed on many different behavioural phenotypes and allow insight into the neural activity responsible for them.

Table 5.1: Feature importances for the five most important Neurons in a dataset

Neuron Index	Feature Importance (%)
17	1.06
9	1.05
8	0.91
19	0.90
26	0.86

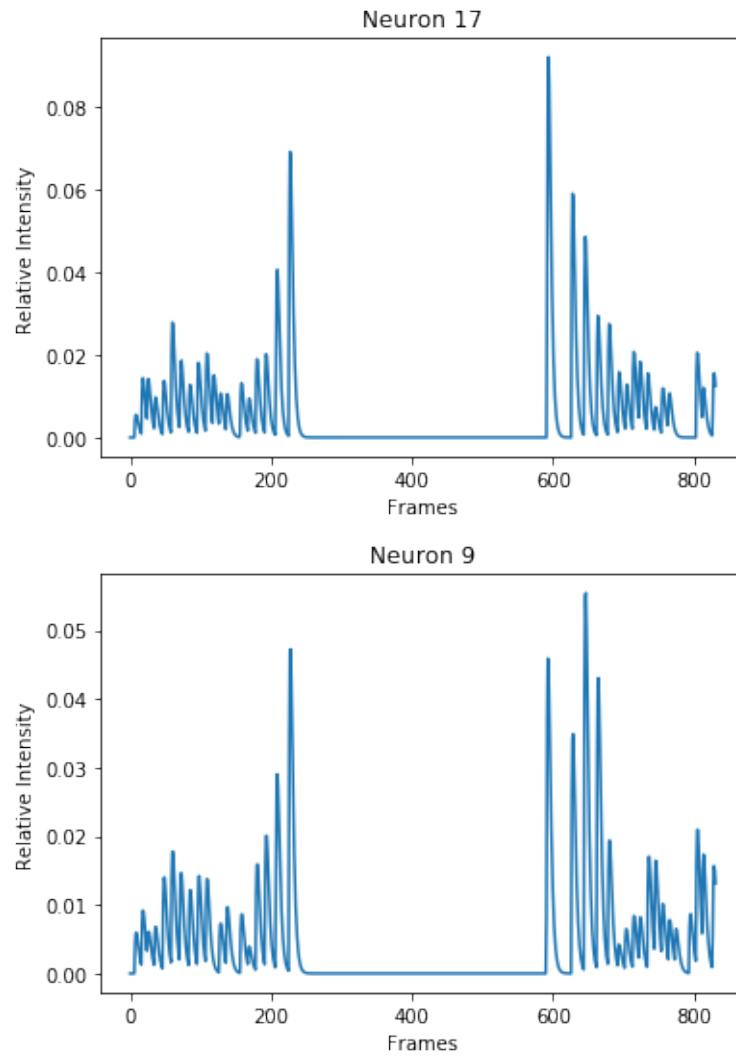


Figure 5.24: Neural activity signals of the two most important neurons in the dataset for prediction of contraction behaviours, identified via permutation importance.

6 Conclusions

As more research is being directed towards the study of large populations of neurons, it is becoming clear that ensembles of neurons may hold some of the answers to outstanding questions in neuroscience. The pipeline developed in this project allows for recordings from large populations of neurons to be made autonomously with single-cell resolution. This allows for datasets to be generated that are far larger than possible with current methods for investigating Hydra and enables the investigation and discovery of ensembles at a far faster rate than was previously possible. Neural activity recordings are complemented by the pipeline's ability to record behavioural information simultaneously. This system allows for studies into neural activity and behaviour to be performed with a near-ideal dataset to work from.

The developed methodology accurately images Hydra neural activity, tracks individual neurons, and processes the resulting signals to extract neural activity, while continuously tracking the Hydra's morphology. These methods have been demonstrated to work robustly on real data. The main benefit of the generation of these datasets of neural activity and behaviour is the ability to apply data hungry machine learning algorithms to find connections between them. The implementation of LSTM models to predict behaviour from neural data proved particularly promising, capable of detecting the onset of a contraction 50 frames (5s) before the animal physically contracted.

This user-friendly pipeline allows for the future investigations of neural ensembles in Hydra, and allows for more rapid experimental workflows than any method used before.

6.1 Limitations

Although the use of the two-colour strain of Hydra allows for robust tracking, it does come at the cost of reduced neural signal from GCaMP. It is not entirely clear whether this decrease in signal is due to GCaMP7 providing worse signal than its predecessor GCaMP6 - which is the version of the protein expressed in the single colour Hydra – or whether the expression of two fluorescence indicators has an adverse effect on

their expression due to increase demand on the Hydra's cells to produce more proteins. Resolution of this issue may improve the SNR of the pipeline and allow for more neurons to be recorded from. This could be addressed by the generation of improved transgenic strains of Hydra.

Another limitation is the lack of an optogenetic Hydra line. These animals would be able to have their neurons stimulated or silenced using light. This would allow for further investigation into the neural ensembles discovered through the ability to manipulate neural ensembles. However, these animals have proven hard to generate.

The most hindering limitation of this study was the inability to fit an entire Hydra in the FOV of the microscope used. This reduced the number of neurons that could be tracked as well as complicating the behavioural recording procedure. This could be rectified in the future with a more advanced optical system however.

6.2 Future Work

Future work should focus on three main areas: improving the tracking capabilities of the developed script, developing further deep learning methods to connect neural activity and behaviour, and using the pipeline to perform experiments.

Work on improving the performance of the tracking and processing algorithms would be useful to increase the number of neurons recorded from. This could potentially allow for more ensembles to be detected. This could be achieved by writing a custom built tracking application to replace Icy, possibly providing better performance. Transgenic and optical improvements would also provide large benefits in this area, however these solutions are more costly and time consuming than improving analysis tools.

Development of the deep learning methods used to link neural activity and behaviour would also be interesting. These methods should work with any organism, and as such their potential benefit is wider-reaching than that of the rest of the pipeline. Deep learning is a rapidly developing field, and it is likely that even better algorithms than the LSTM model described will be created in the future.

This pipeline could be used in the future to study Hydra neural activity in response to stimuli. Currently, no stimulation experiments have been analysed with this method, and novel neural activity is likely to present itself when Hydra react to different conditions. An example of this could be studying the neural dynamics of Hydra at different temperatures.

References

- [1] C. Grienberger and A. Konnerth, "Imaging Calcium in Neurons," *Neuron*, vol. 73, no. 5, pp. 862 - 885, 2008.
- [2] R. W. Williams, *Mouse Brain Development*, New York: Springer, 2000, pp. 21- 49.
- [3] C. Dupre and R. Yuste, "Non-overlapping neural networks in *Hydra vulgaris*," *Current Biology*, vol. 27, no. 8, pp. 1085 - 1097, 2017.
- [4] H. Lodish et al., "Molecular Cell Biology. 5th edition.", New York: W. H. Freeman and Company; 2003. pp. 276-277.
- [5] S. R. Cajal, *Histología del sistema nervioso*, Madrid, 1899.
- [6] R. Yuste, "From the neuron doctrine to neural networks," *Nature Reviews Neuroscience*, vol. 16, p. 487–497, 2015.
- [7] N. Ji, L. Freeman and S. Smith, "Technologies for imaging neural activity in large volumes," *Nature Neuroscience*, vol. 19, pp. 1154-1164, 2016.
- [8] Corder et al, "An amygdalar neural ensemble that encodes the unpleasantness of pain," *Science*, vol. 363, no. 6424, pp. 276-281, 2019.
- [9] W. Yang, R. Yuste, "In vivo imaging of neural activity", *Nature Methods*, vol. 14, pp. 349-359, 2017.
- [10] M. J. Sanderson et al, "Fluorescence microscopy", *Cold Spring Harbor protocols* vol. 2014,10 pdb.top071795. 1 Oct. 2014, doi:10.1101/pdb.top071795.
- [11] O. Shimomura, F. H. Johnson and Y. Saiga, "Extraction, Purification and Properties of Aequorin, a Bioluminescent Protein from the Luminous Hydromedusan, *Aequorea*", *J. Cell. Comp. Physiol.*, vol. 59. pp. 223-239, 1962.

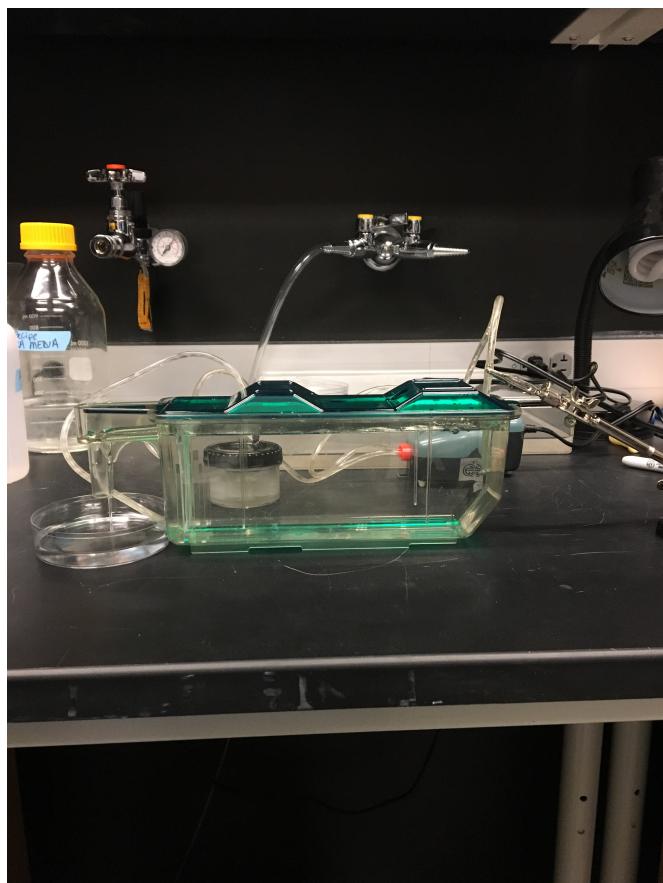
- [12] J. Nakai, M. Ohkura and K. Imoto, "A high signal-to-noise Ca²⁺ probe composed of a single green fluorescent protein." *Nature Biotechnology*, vol. 19, pp. 137 - 141, 2001.
- [13] M. Ohkura et al. "Genetically Encoded Green Fluorescent Ca²⁺ Indicators with Improved Detectability for Neuronal Ca²⁺ Signals", *PLOS ONE*, vol. 7, no. 12, e51286, 2012.
- [14] Nguyen J et al, "Whole-brain calcium imaging with cellular resolution in freely behaving *Caenorhabditis elegans*," *pnas*, vol. 133, no. 8, pp. 1074 - 1081, 2016.
- [15] N. Shaner et al., "Improved monomeric red, orange and yellow fluorescent proteins derived from *Drosophila* sp. red fluorescent protein", *Nature Biotechnology*, vol. 22, pp. 1567–1572, 2004.
- [16] U. Technau, R. Steele, "Evolutionary crossroads in developmental biology: Cnidaria", *Development*, vol. 138, pp. 1447-1458, 2011.
- [17] C. Galizia, P-M. Lledo, "Neurosciences - From Molecule to Behavior: a university textbook." Springer Spektrum, Berlin, Heidelberg, pp. 21-22, 2013.
- [18] S. Erulkar and T. Lentz, "Nervous system", *Encyclopædia Britannica, inc.*, 2019, <https://www.britannica.com/science/nervous-system>, Accessed Jan 10, 2020.
- [19] C. Juliano, H. Lin and R. Steele, "Generation of Transgenic Hydra by Embryo Microinjection", *Journal of Visualized Experiments: JoVE*, vol. 91 51888, 2014.
- [20] M. Scholz et al., "Predicting natural behavior from whole-brain neural dynamics", *BioRxiv*, doi: <http://dx.doi.org/10.1101/445643>, 2017, accessed Nov. 2019.
- [21] S. Han et al, "Comprehensive machine learning analysis of Hydra behavior reveals a stable basal behavioral repertoire", *eLife* 2018;7:e32605.
- [22] C. Aggarwal, "Neural Networks and Deep Learning", Cham: Springer International Publishing, pp. 1-40, 2018.
- [23] J. Olden, M. Joy and R. Death, "An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data", *Ecological Modelling*, vol. 178, pp. 389-397, 2004.
- [24] R. Steele, "Trembley's polyps go transgenic", *pnas*, vol. 103, no. 17, pp. 6415-6416, 2006.
- [25] K. Smith et al., "CnOtx, a Member of the Otx Gene Family, Has a Role in Cell Movement in Hydra", *Developmental Biology*, vol. 212, pp. 392-404, 1999.

- [26] P. Tran and F Chang, "Transmitted Light Fluorescence Microscopy Revisited", *Biol. Bull.*, vol. 201, pp. 235-236, 2001.
- [27] W. Zipfel, R. Williams and W. Webb, "Nonlinear magic: multiphoton microscopy in the biosciences", *Nature Biotechnology*, Vol. 21, pp. 1369–1377, 2003.
- [28] W. Denk, J. Strickler and W. Webb, "Two-Photon Laser Scanning Fluorescence Microscopy", *Science*, vol. 248, no. 4951, pp. 73-76, 1990.
- [29] R. Yuste and W. Denk, "Dendritic spines as basic functional units of neuronal integration", *Nature*, vol. 375, pp. 682-684, 1995.
- [30] W. Yang et al., "Simultaneous two-photon imaging and two-photon optogenetics of cortical circuits in three dimensions", *eLife* 2018;7:e32671.
- [31] S. Han, W. Yang and R. Yuste, "Two-Color Volumetric Imaging of Neuronal Activity of Cortical Columns", *Cell Reports*, vol. 27, pp. 2229–2240, 2019.
- [32] E. Reynaud et al, "Light sheet-based fluorescence microscopy: More dimensions, more photons, and less photodamage", *HFSP journal*, vol. 2, pp. 266-275, 2008.
- [33] T. Fadero et al., "LITE microscopy: a technique for high numerical aperture, low photobleaching fluorescence imaging", *Journal of Cell Biology*, vol. 217, no. 5, pp. 1869–1882, 2018.
- [34] M. Nelson et al., "Spinning Disk Confocal Microscopy of Calcium Signalling in Blood Vessel Walls", *Microscopy and analysis (Americas ed.)*, vol. 24, no. 2, pp. 5-8, 2010.
- [35] T. Nath et al. "Using DeepLabCut for 3D markerless pose estimation across species and behaviors", *Nature Protocols*, vol. 14, 2152–2176, 2019.
- [36] A. Mathis et al., "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning", *Nature Neuroscience*, vol. 21, pp. 1281-1289, 2018.
- [37] N. Chenouard, I. Bloch and JC. Olivo-Marin, "Multiple hypothesis tracking for cluttered biological image sequences", *IEEE Trans Pattern Anal Mach Intell*, vol. 35, no. 11, pp. 2736-3750, 2013.
- [38] K. Burnett, E. Edsinger and D. Albrecht, "Rapid and gentle hydrogel encapsulation of living organisms enables long-term microscopy over multiple hours", *Communications Biology*, vol. 1, no. 73, 2018.
- [39] L. Breiman, "Random Forests", *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

- [40] A. Hyvarinen and E. Oja, "Independent Component Analysis: Algorithms and Applications", *Neural Networks*, vol. 13(4-5), pp. 411-430, 2000.
- [41] A. Giovannucci et al., "CaImAn: An open source tool for scalable Calcium Imaging data Analysis", *eLife*, vol. 8, e38173, 2019.
- [42] J. Volgenstein et al., "Fast Nonnegative Deconvolution for Spike Train Inference From Population Calcium Imaging", *J.Neurophysiol*, vol. 104, pp. 3691–3704, 2010.
- [43] J. Friedrich, P. Zhou, L. Paninski, "Fast online deconvolution of calcium imaging data", *PLoS Comput Biol*, vol. 13, no. 3, e1005423, 2017.
- [44] E. Pnevmatikakis et al., "Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data", *Neuron*, vol 89, pp. 285-299, 2016.
- [45] A. Jain, "Data clustering: 50 years beyond K-means", *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651-666, 2010.
- [46] D. Mullner, "Modern hierarchical, agglomerative clustering algorithms", *arXiv*, arXiv:1109.2378 [stat.ML], 2011.
- [47] F. Kuiper and L. Fisher, "A Monte Carlo comparison of six clustering procedures", *Biometrics*, vol. 31, pp. 777-783, 1975.

Appendices

Appendix A



Hydra culture tank. Consists of a pump, a tank, and a filter system allowing for the automated feeding and cleaning of Hydra.

Appendix B

Single Cell Resolution Analysis of Ca Imaging Data for Hydra v4

(Use with data from *tdTomato_GCamp Animals*)

Requires:

- Green and Red Channel Videos From 2-Colour Confocal (GCaMP Channel and tdTomato)
- Tracking Position Information From ICY Spot Tracking Protocol (Exported as CSV)
- Conda Environment: Caiman_NOAH3

INPUT PATHS

Enter paths to the appropriate files

Videos must be .avi (convert in ImageJ if not)

Can also use a Tif sequence folder - change the `read_data` function to `Read_Data_TIFseq` for this

If using windows: Paths must start with an 'r' character: e.g. `vid_path = r'C:\Users\rylab\path to your file\clip.avi'`

```
In [1]: #input path to your .csv from ICY
csv_path = r"C:\Users\rylab\Documents\Noah2019_desktop\Columbia_Project\Project_Behaviour_and_Neural_Activity\Single_Cell_Analysis\ICY\output\2019-07-10_14-45-00\2019-07-10_14-45-00_icy.csv"

#input path to .avi of GCaMP Video
vid_path = r"C:\Users\rylab\Documents\Noah2019_desktop\Columbia_Project\Project_Behaviour_and_Neural_Activity\Single_Cell_Analysis\Videos\2019-07-10_14-45-00\clip.avi"

#input path to .avi of tdTomato Video
red_vid_path = r"C:\Users\rylab\Documents\Noah2019_desktop\Columbia_Project\Project_Behaviour_and_Neural_Activity\Single_Cell_Analysis\Videos\2019-07-10_14-45-00\clip_red.avi"
```

SET UP DATA

Run the following cells to set up the data to be analysed

Set the FFmpeg path before running this cell

```
In [2]: #importing packages
import numpy as np
import scipy as sp
import skimage.io as iio
```

Sample of the developed Jupyter Notebook

Appendix C

C.1

SubROI neuron tracking code

```

def SingleCellIntensity(neuron, video, positions, dimentionROI, Circle_radius,
                        distance_threshold, display_on = False):
    display = []
    mask_circle_radius = Circle_radius - 0
    neuron_points = []
    intensities = []
    positions_corrected = []
    backframes = 10
    past_thresh = 3
    for frame in range(len(video)):
        # print('Frame: ', frame)
        #Load raw image and copy to avoid affecting the original video
        raw_image = ROIextractor(video[frame],positions[neuron][frame], dimentionROI)
        image = raw_image.copy()

        #correction for ROI leaving field
        dim_image = np.min(image.shape[0:1])
        if dim_image < 5:
            dim_image = 5
        image = image[0:dim_image, 0:dim_image]

        #Convert to grayscale
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        #Copy Image pre processing to use for finding final intensity value from
        image_out = image.copy()
        image_display = image.copy()
        #Create Gaussian Kernel to make points close to the centre appear brighter than
        points on the edges
        kernel_gauss = gkern(image.shape[0], 0.7)
        image = kernel_gauss*image
        #find Centre Point of the image to use as starting point for finding neuron
        centrept = [int(len(image[0])/2), int(len(image[1])/2)]

        if frame == 0:

```

```
#find first 3 highest pixels highest pixel
highpt_prev = centrept
highpts = []
distances = []

for k in range(4):
    image = cv2.GaussianBlur(image, (5,5), 0)
    low,high,lowpt,highpt = cv2.minMaxLoc(image)
    highpts.append(highpt)

#add brightness coefficient to counteract possibility of darker point
#near centre
bright_coeff = (1/high)*0.5

cv2.circle(image, highpt, Circle_radius, 0, -1)

distances.append(bright_coeff*euclidean(highpts[k], highpt_prev))

#take point closest to centre
mindist = np.argmin(distances)
neuronpt = highpts[mindist]

elif frame > 0:
    image = cv2.GaussianBlur(image, (5,5), 0)
    low,high,lowpt,highpt = cv2.minMaxLoc(image)
    distance = euclidean(highpt, highpt_prev)

    if frame < backframes:
        if distance <= distance_threshold:
            neuronpt = highpt

    elif distance > distance_threshold:
        for k in range(3):
            cv2.circle(image, highpt, Circle_radius, 0, -1)
            image = cv2.GaussianBlur(image, (5,5), 0)
            low,high,lowpt,highpt = cv2.minMaxLoc(image)
            distance = euclidean(highpt, highpt_prev)
            if distance <= distance_threshold:
                neuronpt = highpt
                break
```

```

else:

    distance_past = euclidean(highpt, neuron_points[frame-backframes])

    if distance <= distance_threshold and distance_past <= past_thresh:
        neuronpt = highpt

    if distance > distance_threshold:
        # print('thresh1')

        for k in range(3):
            cv2.circle(image, highpt, Circle_radius-1, 0, -1)
            image = cv2.GaussianBlur(image, (5,5), 0)
            low,high,lowpt,highpt = cv2.minMaxLoc(image)
            distance = euclidean(highpt, highpt_prev)
            if distance <= distance_threshold:
                neuronpt = highpt
                distance_past = euclidean(highpt, neuron_points[frame-
                                                               backframes])
                break

        if distance_past > past_thresh: # and distance <= distance_threshold:
            # print('past')
            neuronpt = neuron_points[frame-2]

#save_values for next iteration
highpt_prev = neuronpt
neuron_points.append(neuronpt)
# print('npt ',neuronpt)

#extract fluorescence within selected circle
circle_mask = np.zeros((image_out.shape[0],image_out.shape[1]),dtype = np.uint8
                      )
cv2.circle(circle_mask, neuronpt, mask_circle_radius, 255, -1)

if display_on == True:
    image_display = cv2.GaussianBlur(image_display, (5,5), 0)
    cv2.circle(image_display, neuronpt, Circle_radius, 255)
    plt.imshow(image_display)
    plt.show()

```

```
    display.append(image_display)

    image_out = cv2.GaussianBlur(image_out, (5,5), 0)
    mean = np.max(cv2.mean(image_out, mask=circle_mask))

#        print('mean: ', mean)
#        print('max: ', np.max(image_out))

    intensities.append(mean)
    positions_corrected.append(posit)

if display_on == True:
    return intensities, positions_corrected, neuron_points, display
else:
    return intensities, positions_corrected, neuron_points
```

C.2

Ratiometric Correction Code

```
#deltaR/R

def find_dR_R(intensities_green,intensities_red):
    dR_R = []
    for i in range(len(intensities_red)):
        dR_R_neuron = []

        #create R array
        R = np.zeros(len(intensities_red[i]))
        for k in range(len(intensities_red[i])):
            R[k] = intensities_green[i][k]/intensities_red[i][k]

        #lowest 20th percentile used as base r (from C. Elegans Paper)
        range_R = np.max(R) - np.min(R)
        percentile = range_R/100
        percentile_boundary = 20
        R_bkg = np.min(R)+(percentile*percentile_boundary)

        for j in range(len(intensities_red[i])):
            dR_R_frame = ((R[j] - R_bkg)/R_bkg)
            dR_R_neuron.append(dR_R_frame)
        dR_R.append(dR_R_neuron)

    return dR_R
```

Independent Component Analysis Code

```
def ICAdecorr(G, R, tolerance, repeats):
    #edited function from Scholz et al, 2019 to account for randomness of ICA by
    #repeating multiple times and selecting
    #best outcome

    R = np.asanyarray(R)
    G = np.asanyarray(G)
    Ynew = []
    for li in range(len(R)):
        possible_outcomes = []
        for k in range(repeats):
            ica = FastICA(n_components = 2, tol = tolerance)
            Y = np.vstack([G[li], R[li]]).T
            Ynew.append(ica.fit_transform(Y))
```

```
sclar2= StandardScaler(copy=True, with_mean=True, with_std=True)
Y = sclar2.fit_transform(Y)
S = ica.fit_transform(Y)

# order components by max correlation with red signal
v = [np.corrcoef(s,R[li])[0,1] for s in S.T]
idn = np.argmin(np.abs(v))

# check if signal needs to be inverted
sign = np.sign(np.corrcoef(S[:,idn],G[li])[0,1])
signal = sign*(S[:,idn])
possible_outcomes.append(signal)

#best_outcome = possible outcome least correlated with the red signal (
#including anticorrelation)

correlations = []
for j in range(len(possible_outcomes)):
    correlations.append(np.corrcoef(possible_outcomes[j], R[li])[0,1])
min_corr_index = np.argmin(np.abs(correlations))
best_outcome = possible_outcomes[min_corr_index]
Ynew.append(best_outcome)

return np.array(Ynew)
```

C.3

Non-Neuronal Cell Filtering Code

```
#Filters non-neuronal cells based on how well signals follow a gaussian distribution
def Gaussian_noise_filter(intensities, alpha, positions):
    new_data = []
    positions_new = []
    filtered_tracks = []
    for i in range(len(intensities)):
        a,b = sp.stats.normaltest(intensities[i])
        if b >= alpha:
            #follows normal dist
            filtered_tracks.append(intensities[i])
        elif b < alpha:
            #does not follow normal dist
            new_data.append(intensities[i])
            positions_new.append(positions[i])
    return new_data, positions_new, filtered_tracks
```

C.4

Denoising and deconvolution of neural signals

```
#Uses FOOPSI algorithm from CAIMAN package to decompose calcium signal and estimate
#neural spikes

def FOOPSI_all(detrended_data):
    detrended = detrended_data
    spikes_signal_dR = np.zeros((len(detrended), len(detrended[1])))
    ca_foopsi_traces = np.zeros((len(detrended), len(detrended[1])))
    for i in range(len(detrended)):
        ca_foopsi, cb, bl, g, cl, spikes_foopsi, lam = deconv.constrained_foopsi(np.
            asanyarray(detrended[i]), p=2)
        spikes_signal_dR[i] = spikes_foopsi
        ca_foopsi_traces[i] = ca_foopsi
    return ca_foopsi_traces, spikes_signal_dR
```

```
#Uses FOOPSI spike information to generate an array to be plotted as a raster plot

def Find_Raster(foopsi_spikes, threshold):
    spikes_signal_dR = foopsi_spikes
    spike_thresh_dR = threshold
    raster_array_dR = np.zeros((len(spikes_signal_dR), len(spikes_signal_dR[1])))
    for i in range(len(spikes_signal_dR)):
        for j in range(len(spikes_signal_dR[i])):
            if max(spikes_signal_dR[i]) > 0:
                if spikes_signal_dR[i][j] >= spike_thresh_dR: #*np.mean(
                    spikes_signal_dR[i]):
                    raster_array_dR[i][j] = j
    return raster_array_dR
```