# Toward a Processing Pipeline for Two-photon Calcium Imaging of Neural Populations

*A dissertation submitted to the graduate school in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Statistics and Neural Computation*

by

Bronwyn Lewisia Woods

Department of Statistics
Program in Neural Computation
Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213

August, 2013

**Advisor**

William F. Eddy

**Committee**

Robert E. Kass

Seong-Gi Kim

Alberto Vazquez

# Abstract

Two-photon calcium imaging (TPCI) is a functional neuroimaging technique that simultaneously reveals the function of small populations of cells as well as the structure of surrounding brain tissue. These unique properties cause TPCI to be increasingly popular for experimental basic neuroscience. Unfortunately, methodological development for data processing has not kept pace with experimental needs. I address this lack by developing and testing new methodology for several key tasks.

Specifically, I address two primary analysis steps which are nearly universally required in early data processing: region of interest segmentation and motion correction. For each task I organize the sparse existing literature, clearly define the requirements of the problem, propose a solution, and evaluate it on experimental data. I develop MaSCS, an automated adaptable multi-class segmentation system that improves with use. I carefully define and describe the impact of motion artifacts on imaging data, and quantify the effects of standard and innovative motion correction approaches. Finally, I apply my work on segmentation and motion correction to explore one scientific target, namely discovering correlation-based cell clustering. I show that estimating such correlation-based clustering remains an open question, as it is highly sensitive to motion artifacts, even after motion correction techniques are applied.

The contributions of this work include the organization of existing resources, methodological advances in segmentation, motion correction and clustering, and the development of prototype analysis software.

# Acknowledgements

This dissertation would not have been possible without support and assistance from a large number of people.

I thank my advisor Bill Eddy for helping me navigate the world of academic research and for never questioning my ability to succeed. I also thank the rest of my committee members for their confidence and support throughout the dissertation process. Special thanks go to Alberto Vazquez for collecting all the data used in this work and for helping me learn about the experimental field.

My thanks also go to my fellow grad students, both in the statistics department and the Center for the Neural Basis of Cognition. Their companionship, empathy, support, and occasional distraction were central to celebrating the ups and surviving the downs of the last five years.

Thanks to my parents for their unconditional support. They never doubted I could find something I loved and do it well. They were a great help in keeping grad school in perspective. Thanks to Mark for being there and for patiently listening to and distracting me from the frequent bouts of grad school angst.

Finally, I would like to thank all of the people in my life who helped me successfully get through grad school by keeping me grounded in the world outside the office. These people are far too numerous to list, but the last five years would not have been possible without the potlucks, board games, conversations, dance weekends, bike trips, and other gatherings.

# Contents

# List of Figures

# Part I

# Introduction

# 1

# Overview

Two-photon calcium imaging (TPCI) is a quickly growing experimental field, but unfortunately the development of analysis methodology and tools has not kept pace with experimental interest. In this dissertation I present improvements to basic processing tasks faced in the analysis of TPCI data. The analysis that I discuss is often considered pre-processing; its development is not the focus of experimental labs. Nevertheless, appropriate and clearly described early data processing is crucial to producing reliable, comparable, and reproducible scientific results.

Because of the dearth of resources currently available to experimenters entering the field, many experimental labs develop their own algorithms and code for universal processing tasks. The details of the resulting processing pipelines are often only sparsely documented in the methods sections of papers, and the code is not always made available. This makes it very difficult to compare new methodology against existing techniques. Experimentalists without a computational focus have few resources available to aid in data processing. Computational scientists who wish to make contributions to the field may not be able to determine what advances would be most useful. With this in mind, the goal of this dissertation is to assist in the development of a standard and easily accessible analysis toolkit for TPCI research.

## 1.1 Context of this work

With increasing experimental interest in TPCI, the development of dedicated data analysis tools is becoming a necessity. There are innumerable analysis challenges that an experimenter using TPCI might face. Some tasks are arguably inherent to the imaging modality whereas others are specific to the scientific questions being asked. Improvements to the first may have a broad impact on the field, whereas improvements to the second may allow for specific innovation and discovery. I focus on two analysis tasks which are faced by the majority of experimenters and one which is more specific to particular scientific questions.

The first task I address is automation of region of interest segmentation. In the current literature, this is frequently accomplished through tedious manual annotation. I propose the MaSCS framework, an automated system based on supervised multi-class classification. MaSCS is more efficient than manual systems, and more flexible than existing automated systems. It allows for customization, improves its performance with use, and encourages consistent reporting and evaluation metrics.

The second task I address is motion correction. Motion is unavoidable in *in vivo* imaging, and while most experimenters perform at least some form of correction, there is very little work examining the necessity or impact of these techniques. My work advances this area by examining the impact of both standard and original motion correction approaches on several aspects of data.

Finally, I combine the tools I develop for segmentation and motion correction to consider the task of clustering cells based on the correlation of their calcium fluorescence traces. This is an unexplored area in the literature, but is important in experimental paradigms that are not conducive to spike train analysis. I show that with currently available methodology, these clusters are heavily corrupted by motion artifacts. Correlation-based clustering remains an open area of research.

The tasks of region of interest segmentation and motion correction are already acknowledged by the experimental community to be part of any typical analysis. However, the exact structure and components of an analysis pipeline are not established. I believe that segmentation and motion correction must be early steps in any such pipeline. Nevertheless, it is worth noting some of the related tasks and context that I do not develop in this work.

Many experimenters wish to focus on neural activity in the form of spike trains. TPCI measures the calcium transients associated with spikes, from which spike trains can be inferred. The process of performing this inference is one of the best studied analysis task for TPCI (Hill et al., 2010; Vogelstein et al., 2010, 2009; Yaksi and Friedrich, 2006; Smetters et al., 1999). I chose not to focus on this task partly because the existing literature provides some information and tools already. In addition, this problem can be best addressed with joint TPCI and electrophysiology experiments, which were not available for this work.

There are a number of more specialized analysis tasks that I do not address here. For instance, TPCI provides a unique opportunity to study properties of the neural vasculature. This requires quantification of properties of blood vessels and blood flow (Drew et al., 2011). The estimation of these quantities is relatively unstudied. As another example, TPCI can record properties of cells that are not electrically active, such as glial cells. There is increasing interest in the role that glial cells may play in neural processing. Answering this question with TPCI requires quantifying and identifying the signatures of glial calcium activity. This is less studied than the spike detection problem for neurons. Though these are important analysis tasks without a clear solution, I leave their development to future work.

On the data collection side of TPCI, there are many researchers working to improve the speed, quality and flexibility of data collection. This work can take the form developing microscopes and scanning techniques (Katona et al., 2012; Ranganathan

and Koester, 2010; Mittmann et al., 2011), fluorescent dye innovation (Lütcke et al., 2010), or improvement of surgical protocols. Improvements to the data collection process are critical, and will certainly impact the types and extent of post-collection analysis required. However, for this work I restrict my attention to a particular common data collection framework, described in section 2.5.

Finally, though developments to individual data collection and analysis tasks is important and a logical place to start, it will become increasingly important to formally consider the design of analysis pipelines. For instance, though most experimental TPCI papers report some form of motion correction and region of interest segmentation, the ordering of these tasks is inconsistent. It is unclear which preprocessing steps are necessary for particular types of subsequent analysis. The number of experimenters using TPCI is growing, and we are rapidly discovering the large collection of scientific questions made accessible by the technology. To take full advantage of this experimental effort, it is critical that we also develop a statistically solid set of analysis tools, and research how to choose and combine them effectively and intelligently.

## 1.2 Contributions of this work

The main contributions of this work are three-fold, listed below.

**Goal:** Develop statistically motivated methodology for important tasks in the TPCI analysis pipeline.

**Contribution:** I have researched analysis methodology for three tasks: region of interest segmentation (chapter 3), motion correction (chapter 4), and correlation-based cell clustering (chapter 5). The description of this work constitutes the bulk of this document.

**Goal:** Create easily used tools for experimenters who wish to apply the methodology I have developed.

**Contribution:** All of the processing that I discuss can be accomplished using the R package RCI available at *https://github.com/dancingwoods/RCI*. The package is fully documented, but is a prototype rather than production ready software.

**Goal:** Organize and make easily accessible the tools that already exist for TPCI data analysis.

**Contribution:** This document itself provides a more comprehensive overview of the existing work on TPCI analysis than was previously available. I have also founded an online wiki, *CaliCode.org*, to serve as an introduction to the existing knowledge and tools relating to the analysis of TPCI data. The site provides an overview of analysis tasks encountered by experimenters, basic tutorials and links to more details in papers and books, an index of currently available relevant software toolkits, an annotated bibliography of papers that address analysis issues, and more.

## 1.3 Organization of this document

The remainder of this document is organized as follows. Chapter 2 provides an introduction to the basics of two-photon calcium imaging. Chapters 3 through 5 discuss my work on segmentation, motion correction, and clustering. Each chapter contains a literature review, presents and evaluates my methodology, and discusses future work. Finally, chapter 6 discusses the resources that I have created for the community and summarizes my work.

# 2

# Two-photon calcium imaging

*In vivo* neuroimaging methodology is the focus of a great deal of research for the simple reason that the functioning brain is inherently difficult to measure. A diverse variety of imaging technologies allow measurement of various indicators of neural activity at different temporal and spatial scales. Two-photon calcium imaging (TPCI), an increasingly popular technique, uniquely fills the need to image the activity of small populations of neurons along with their spatial layout and physiological context.

In general, TPCI uses a two-photon laser scanning microscope to image tissue containing a calcium responsive functional indicator as well as (typically) a static structural dye. In combination, these dyes reveal the location, size, shape, and activity of neurons, astrocytes and blood vessels. From this experimenters can deduce neural spike trains (Yaksi and Friedrich, 2006; Vogelstein et al., 2009), calcium transients in astrocytes (Nimmerjahn et al., 2004; Lohr and Deitmer, 2010; Reeves et al., 2011), properties of local blood flow (Drew et al., 2011), and connectivity of the local neural network (Mishchencko et al., 2011). Though invasive, TPCI gives more comprehensive measurements of cortical function than other recording techniques with similar spatial scale (such as array electrophysiology) and is therefore ideal for studying integrative questions in basic neuroscience.

This chapter introduces the engineering and biology behind TPCI. This treatment

is intended to be an overview to aid in understanding the work presented in this dissertation. The interested reader is referred to the cited references for further detail and a more comprehensive view of the experimental field.

## 2.1 Two-photon Excitation Microscopy

Fluorescence microscopy, very generally, has two essential steps: illuminate the sample to excite a fluorophore then collect the emitted fluorescence. There are several general techniques for fluorescence imaging including wide-field, confocal and two-photon microscopy (TPM). TPM is the most complex technologically (and therefore the newest) but has significant advantages in that it can image deep tissue *in vivo* with limited photo-damage. This section introduces TPM as compared to these earlier fluorescence microscopy techniques.

Wide-field fluorescence microscopy involves illuminating the entire sample with a light source such as a mercury vapor lamp. The microscope objective collects reflected light as well as emitted fluorescence. A filter can isolate emitted fluorescence based on wavelength. However, since the sample is illuminated uniformly, emitted fluorescence from outside the focal plane is mixed with the desired signal. In combination with light scattering in tissue, this results in significant background fluorescence and spatial blurring.

Confocal microscopy solves several of these issues. In confocal microscopy, the excitation light (typically from a laser) is focused at a point in the tissue to be imaged. Emitted fluorescence photons must pass through a pinhole detector, restricting detected fluorescence to that originating at the focal point. The focal point is scanned through the tissue to create an image. A significant limitation of confocal fluorescence microscopy derives from the tendency of biological tissue to scatter light. Only a fraction of the excitation photons reach the focal point, and many of the emitted photons

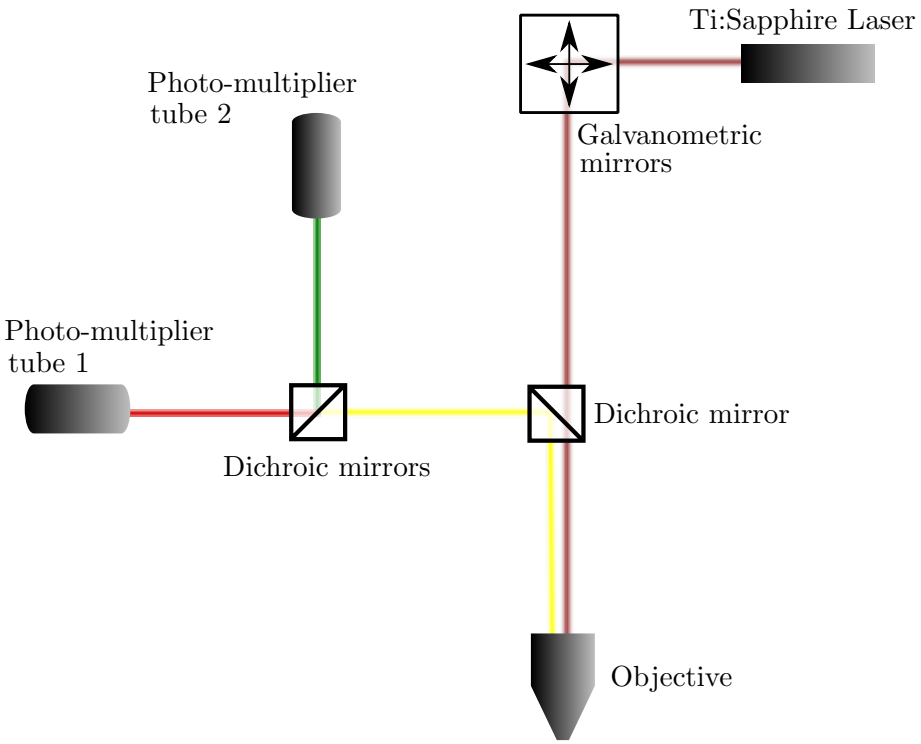



**Figure 2.1:** *Schematic of the Prairie Ultima two-photon laser scanning microscope. A Ti:Sapphire pulsed laser delivers the excitation light. Galvanometric scanning mirrors control the raster scanning pattern. An objective focuses the laser at the desired depth in the sample. Emitted fluorescence is collected through the objective. A dichroic mirror directs the fluorescence wavelengths to an emission cube composed of one or more dichroic mirrors. The cube further separates emission by wavelength, allowing for the collection of fluorescence from two fluorophores. Photo-multiplier tubes measure the emitted fluorescence.*

are scattered and therefore rejected by the pinhole detector. This loss of signal must be countered by increasing the power of the excitation laser, which increases the likelihood of photobleaching and damage to the tissue. In addition, photobleaching is accelerated by the fact that excitation occurs throughout the light cone even though only fluorescence from the focal point is accepted by the detector.

Both wide-field and confocal microscopy generate fluorescence through the interaction of a single photon with a fluorescent molecule. In contrast, two-photon microscopy generates fluorescence by the near simultaneous absorption of two longer-wavelength photons. Such simultaneous absorption events can only occur in an environment with extremely highly concentrated photons. In TPM this required intensity is generated by a focused, femtosecond pulsed laser. The intensity of the laser drops off quadratically with distance from the focal point meaning that simultaneous absorption events occur exclusively in a small volume around this point. This restricts photodamage to only this area. Additionally, since only fluorophores in the focal volume are excited, the source of all emitted photons is known thus removing the need for a pinhole detector. Any emitted photon reaching the detector is signal, meaning TPM suffers much less signal loss due to scattering than confocal microscopy. In addition, the longer wavelength excitation used in TPM is less susceptible to scattering in tissue than that used in confocal microscopy. This allows greater penetration of tissue and increases the feasible imaging depth.

Imaging more than a single point with TPM requires scanning the laser through the sample. Typically this is done with galvonometric mirrors. Two mirrors control scanning in the $X$ and $Y$ dimensions within a fixed $Z$ plane. An image is scanned in a simple raster pattern. Sampling rate depends on the size of the area being scanned, the number of pixels, and the speed with which the mirrors travel, accelerate and decelerate. The speed of the mirrors and the desired pixel size determine the amount of time the laser is focused within a pixel (dwell time), impacting the noise

level. For images large enough to show networks of neurons (one to two hundred microns on a side, with pixels several microns on a side), the scanning time in this set up is dominated by mirror acceleration and deceleration as well as flyback, putting severe limits on temporal resolution. With the equipment available for the work done in this dissertation, frame rates for such imaging are around 8 Hz. Increasing temporal resolution requires reducing the image size, reducing the number of pixels, changing the physical method by which the laser is scanned, and/or changing the raster scanning pattern.

## 2.2 Calcium Sensitive Fluorescent Dyes

There are many fluorescent dyes used in the biological sciences. Each is different in its specifics, but they all work on the same principle. The fluorescent protein or molecule (fluorophore) is exposed to energy in the form of photons, causing electrons in the fluorophore to be excited. These electrons subsequently fall back to a lower energy state, releasing photons of a somewhat longer wavelength than those which caused the excitation. These emitted photons can be collected by a detector, providing a measurement of fluorescence. This measurement corresponds to different properties of the imaged tissue depending on the properties of the fluorophore used.

Fluorophores can be targeted to fluoresce in specific environments. An example of this is calcium-sensitive fluorescent dye. The fluorophore is combined with a calcium indicator which reconfigures the fluorophore in the presence of calcium. This results in fluorescence properties that change according to calcium levels (Johnson, 1998; Verkhratsky and Petersen, 2010). A common indicator used for this purpose is the calcium buffer BAPTA, developed in the 1980s by Roger Tsien and colleagues (Tsien, 1980). Oregon Green BAPTA (OGB), as used in the data for this research, is a green fluorescent dye based on the BAPTA calcium indicator.

As discussed in the next section, a primary quantity of interest in neuronal calcium imaging is calcium levels inside cells. Detecting this requires directing the calcium sensitive dye into cells. This can be accomplished in a number of ways. Historically, the first approach was loading individual neurons with dye using microelectrodes. This restricted imaging experiments to focusing on small numbers of cells (Stosiek et al., 2003). Subsequently, bolus-loading techniques were developed by which dyes could be injected locally to mark larger populations of neurons simultaneously. Most fluorescent dyes do not permeate cell membranes. In a standard bolus-loading technique the dye is bonded to an acetoxymethyl (AM) ester. This deactivates the fluorescence and causes the dye to become cell permeant. The AM form of the dye diffuses across cellular membranes whereupon esterases cleave the AM group. After this, the dye is again fluorescent and unable to cross the cell membrane (Takahashi et al., 1999). Stosiek et al. (2003) first demonstrated the bulk-loading of calcium dyes to image cell populations *in vivo*. Since then it has become a very common technique (Eichhoff et al., 2010).

Recently, some research has focused on developing genetically encoded fluorescent calcium indicators. These can be expressed in transgenic organisms or introduced neonatally or by viral transduction. However, genetically encoded calcium dyes are still under active development. Compared to synthetic dyes injected into the brain, they stain cells less densely with lower fluorescent response (Garaschuk and Griesbeck, 2010).

The data used in this research were collected using OGB injected using the bolus-loading technique. This method stains most or all of the cells in the vicinity of the injection. However, because the dye is distributed by diffusion from the injection site the staining is not spatially uniform. In addition, the OGB dye fluoresces once it crosses any cell membrane. Neurons, astrocytes and neuropil (the background tissue composed of dendrites, axons and other cellular processes) are all stained by the

dye (though some research suggests that astrocytes are preferentially marked at the extreme edges of the dye extent (Eichhoff et al., 2010)). To attain some differentiation between types of cells, a red fluorescent dye (SR-101) was injected along with the OGB. This dye has been shown to preferentially mark astrocytes (Nimmerjahn et al., 2004), allowing them to be differentiated from neurons.

## 2.3 Calcium in the Brain

Calcium ($Ca^{2+}$) is ubiquitous in signaling pathways in a wide variety of cell types throughout normal and pathological development. I will focus here on a particular domain of $Ca^{2+}$ signaling: that occurring in neurons and astrocytes in the brain. Specifically I will describe calcium currents that are thought to drive the signals recorded by calcium imaging on the local network scale. For a more comprehensive presentation of the many roles of $Ca^{2+}$ see, for instance, Berridge et al. (2000) and Dolphin (2006).

Neurons, specifically those of the mammalian cortex, express four or five different types of calcium channels, distributed unevenly around the soma, dendrites and axon terminals (Bean, 2007; Catterall, 2011). These channels include several varieties that are voltage-gated. During an action potential these channels open, allowing a strong electrochemical gradient to drive $Ca^{2+}$ into the cell. Voltage gated calcium channels open near the peak of an action potential, resulting in a strong calcium current during the falling phase of the spike. The coupling of increased calcium levels to the opening of calcium-activated potassium channels helps sharpen the temporal profile of the action potential by inducing a strong influx of potassium which drives the cell's membrane potential back toward resting state. This calcium current during the falling phase of action potentials likely occurs in nearly every type of neuron (Bean, 2007). The universality of calcium currents during spiking activity allows for the use

of calcium sensitive dyes to infer neural activity or even neural spike trains.

Though the shape of calcium currents can be variable, especially on small scales at axon terminals, joint fluorescence imaging and electrophysiology experiments have established a stereotypical shape for the spike-induced calcium fluorescence time course averaged over the soma of a cell (for a review see Kerr and Denk (2008)). It is worth noting that at finer resolutions not all locations within the cell soma respond uniformly to a spike (Ranganathan and Koester, 2010). Complex calcium dynamics within cellular compartments unavoidably impact fluorescence signals.

At the spatial resolution typical of population imaging with two-photon microscopy, the somatic calcium response is easiest to resolve. However, complex calcium dynamics in cellular processes also contribute to the signal as fluorescence from the neuropil. The axons and dendrites that compose the neuropil are typically too small to be individually resolved in calcium imaging aimed at measuring network function. They appear as minimally differentiated background. The neuropil background shows a fluorescence response to stimulation. This is unsurprising given the importance of calcium signaling in synapses. In axon terminals calcium plays a crucial role in the regulation and release of neurotransmitters. As such, processes regulated by calcium are important in synaptic functioning, plasticity and learning (Berridge et al., 2000). Experiments reported by Kerr et al. (2005) suggest that the calcium signal recorded from the neuropil derives primarily from these axonal calcium transients, suggesting that neuropil fluorescence is a measure of local neuronal output.

Calcium signaling in the brain is not restricted to neurons. Glial cells, which are not electrically excitable, have historically been thought to be primarily structural support cells not involved in processing. Recently increased focus has been placed on potential functional roles for glial cells, particularly as mediated by calcium (Ding, 2012; Reeves et al., 2011; Lohr and Deitmer, 2010; Kerr et al., 2005; Nimmerjahn et al., 2004). The temporal profile of calcium currents in glial cells differs substantially from

that in neurons. Nimmerjahn et al. (2004) found oscillatory calcium responses in astrocytes *in vivo* that have slow (10 second) onsets and plateaus lasting for tens of seconds. They also observed transmission of calcium activity between astrocytes, which appeared as spatial waves of calcium activation. Both these oscillatory and wave-like behaviors have been observed *in vitro* by other experimenters (Lohr and Deitmer, 2010). As with neurons, the calcium signal in astrocytes is not uniform throughout the cell. The calcium response profile differs between the soma and the cellular processes (Reeves et al., 2011). Again, the spatial scale of population calcium imaging does not allow cellular processes to be resolved. However, calcium transients from the processes of astrocytes may well contribute to the neuropil fluorescence.

## 2.4 Advantages of two-photon calcium imaging

Two-photon calcium imaging is one of many neuroimaging techniques, but it has some unique advantages. For instance, though TPCI typically measures from a similar field of view as a multi-electrode Utah array, it has much more spatial detail such as the precise location of active neurons and the inactive cells and structures surrounding them. The advantages of TPCI can most clearly be seen by recognizing the range of the things it measures:

1. (in)activity in tens to hundreds of neurons simultaneously.
2. spatial layout of neurons, astrocytes, blood vessels, etc.
3. properties of blood vessels such as volume or velocity of blood.
4. calcium activity in astrocytes.

This variety of measurements allows in turn for experimental leverage on a variety of neuroscientific areas of interest. By combining measurements enumerated above, TPCI allows for the investigation of:

**1+2** changing clustering of neurons over time.

**1+3** neurovascular coupling, the basis of the fMRI BOLD response.

**3+4** role of astrocytes in vascular regulation.

**2+4** spatial calcium waves in astrocytes, and the role of astrocytes in neural function.

These are questions that may not be easily addressed by other imaging methodologies, making TPCI an exciting addition to the neuroimaging arsenal for basic neuroscience.

## 2.5 Data specifics

Throughout this dissertation I use "two-photon calcium imaging" without further modification. Though the problems and solutions I address are, I hope, applicable to a broad range of experimental set-ups and paradigms, the data that I use in my analyses are from one particular experimental lab. The details of this data, described here, may influence whether the specifics of my processing techniques are directly applicable to data from other sources.

The data are recordings from the somatosensory cortex of rats. The rats were imaged under anesthesia while attached to a mechanical respirator. Each experiment consists of continuous recording for several minutes. Some experiments are resting state recordings in which the animal received no experimental stimulation. In other experiments the animal received periodic electric stimulation to the forepaw strong enough to evoke significant neural activity in the region of somatosensory cortex being imaged.

The animals were prepared for imaging with a craniotomy exposing the somatosensory cortex. Sulforhodamine 101 (SR101) and Oregon Green Bapta 1 (OGB1) dyes were injected into the cortical tissue, after which a coverslip was placed over the

craniotomy.

Two-photon recordings were collected using the Prarie Ultima two-photon laser scanning microscope. This microscope used a $900nm$ pulsed laser to excite the sample. Galvonometric mirrors directed the excitation beam in a raster pattern through the objective, typically covering an area approximately 240 microns on a side with 1.88 micron diameter pixels ($128 \times 128$ pixels per frame). The frame scanning rate was approximately $8Hz$.

The emitted fluorescence was separated by dichroic mirrors and two wavelength bands corresponding to the two fluorescent dyes were measured using photo-multiplier tubes. The first channel (hereafter referred to as the 'structural channel') measured the fluorescence from the SR101 dye with wavelengths of $607 \pm 22.5nm$. The second channel (hereafter the 'functional channel') measured fluorescence from the OGB1 dye with wavelengths between $525 \pm 35nm$.

The structural channel measures from the SR101 dye, which stains astrocytes but not neurons. SR101 is not a functional dye, which means that its fluorescence does not change in response to neural activity. Rather than providing information about neuron function, the structural channel provides a way of differentiating neurons from astrocytes. Due to properties of the dye, the structural channel in this data is significantly less noisy than the functional channel.

The functional channel, measuring from OGB1, provides a measure of calcium levels in the brain over time. Since OGB1 is a calcium sensitive dye, these measurements indicate the level of calcium within cells. As discussed before, this is related to firing in neurons and to less well understood activity in astrocytes. Though the functional channel is noisier than the structural channel, it provides the key information about the activity of the brain over time.

# Part II

# Methodological Development and Application

# 3

# Automated segmentation of regions of interest

Two-photon calcium imaging reports measurements of the brain in a regular grid over the field of view, which gives a very informative but complex dataset. To inform neuroscientific findings, these data must be transformed from the pixel coordinate system into one that centers on features of neuroscientific interest. For instance, an experimenter is unlikely to be interested directly in the intensity time series of a particular image pixel. Instead, he may be interested in the mean fluorescence time courses of the neurons in the field of view. The particular regions of interest (neurons, astrocytes, blood vessels, etc.) will vary based on the scientific question and experimental parameters, but the task of inferring properties of neuroscientific features from pixel intensities is extremely common and important.

Formally, the problem of region of interest (ROI) segmentation for TPCI can be posed as follows. Given four-dimensional data of the form $D_{channel,time,x,y}$, identify the number of regions of interest. For each of these $N$ regions, identify a mask $M_n$, a class label $L_n$, and a time series $S_{nt}$. The mask $M$ specifies a set of (typically contiguous) pixels which sample from the ROI. The label $L$ specifies the type or class of ROI (neuron, astrocyte, etc.). The time series $S_t$ gives the temporal activity associated with the ROI.

Typically, ROIs are defined by binary masks, and the activity trace $S_t$ is the

simple spatial mean of the calcium channel over the identified region. The problem of mapping from pixels to neuroscientific features can then be reduced to an image segmentation problem. Image segmentation is ubiquitous in many areas of science, but there are several characteristics that make this instance unusual. Four important properties of TPCI ROI segmentation are the following:

1. Dimensionality. TPC imaging experiments generally measure from two channels, and for each channel we have images taken over time. Ideally, we want to integrate information from both channels and time into our segmentation procedure.

2. Messy background. The background of TPC images (that is, the part of the field of view that is not within a region of interest) is spatially and temporally quite complex. The background, depending on which features are being segmented, can contain sub-resolution cellular processes, blood vessels, un- or barely stained regions, and poorly resolved cells. This background obviously has a great deal of spatial structure. Due to calcium dynamics in cellular processes, it also shows temporal fluorescence changes in the functional channel. The spatial and temporal complexity of the background makes simple pixel-wise testing against a theoretical background distribution impractical.

3. Uneven intensity. Due to uneven dye distribution and surface vessels obstructing fluorescence, the overall magnitude of fluorescence can vary greatly between regions of an image. This means that any segmentation procedure depending on intensity values must be locally adaptive.

4. Multi-class. TPC images record neurons, astrocytes, and blood vessels. Any or all of these might be regions of interest for a neuroscientific study. A segmentation procedure should be able to separate regions of one type (e.g. neurons) from regions of another (e.g. astrocytes).

## 3.1 Existing work on segmentation

There is a small but growing literature directly addressing the problem of automated segmentation in TPC imaging. Existing approaches can be grouped into three main categories: those using matrix factorization, those using one or (typically) more image processing techniques, and those using data features to build statistical models such as regressions or classifiers. In this section I will review existing work in these three classes of approaches. I will highlight the strengths and weaknesses of each approach, and then discuss the general problem of evaluating performance. Finally, I will conclude the section by proposing a set of characteristics that a desirable segmentation system should possess. In section 3.2, I will propose a segmentation system of the classifier type that has these characteristics.

### 3.1.1 Matrix factorization approaches

As highlighted previously, calcium imaging measures the temporal activity of cells as well as their location. Neurons exhibit a distinctive calcium transient when active. Some glial cells may display calcium transients as well. Several groups have exploited the temporal sparsity of these transients along with the spatial sparsity of cells to segment data using matrix factorization approaches.

To my knowledge, this general approach was first proposed for TPCI by Mukamel et al. (2009). The authors use principle components analyses (PCA) for initial dimensionality reduction and noise removal. They follow this with spatio-temporal independent components analysis (ICA), selecting components to optimize a weighted linear combination of spatial and temporal skewness. They select the components with the highest skewness as the candidate cells. The resulting components have a spatial filter (a non-binary mask) as well as an activity trace.

There are several details of the Mukamel at al. approach worth noting. Firstly,

when the activity of several cells is highly correlated, these cells may be included in the same ICA component. For this reason, the authors use a simple image segmentation procedure (thresholding, followed by separation of spatially separated components) on the component's spatial filter to divide the cells. Secondly, the output of the segmentation procedure requires manual filtering to select which ICA components should be retained. The authors state that the components corresponding to real cells have the highest skewness and so the selection is easy. Nevertheless, this plus other parameters of the procedure, such as the number of PCA components to retain in the initial step, must be chosen manually for each experiment.

Recently, Diego et al. (2013) have proposed a related matrix factorization method using dictionary learning (sparse structured PCA) for cell segmentation in confocal calcium imaging. In this approach, the $N_{pixels} \times N_{time}$ data matrix is approximated by $\mathbf{DU}^T$ where $\mathbf{D}$ is a $N_{pixels} \times K$ matrix giving the spatial components and $\mathbf{U}$ is a $N_{time} \times K$ matrix giving the temporal components. $\mathbf{D}$ and $\mathbf{U}$ are found by minimizing the Frobenius norm of the difference between the original data and $\mathbf{DU}^T$ with sparsity constraints on the spatial components (dictionary elements) and temporal coefficients.

Like Mukamel et al., Diego et al. run the output of their procedure through image processing techniques (such as a watershed algorithm) to split multiple cells assigned to the same component or combine cells split into multiple components.

The Mukamel and Diego approaches to segmentation share many of the same characteristics and thus also share many of the same strengths and weaknesses. One strength emphasized by Mukamel et al. is that these approaches can perform some signal separation of the actual activity trace of a cell from noise from nearby cells or neuropil. This is in contrast to ROI segmentation using simple averaging over binary masks. Another advantage of these approaches is that they use the temporal information and variance structure of the dataset to produce their segmentation. This information is not easily visible to a human annotator, who will typically use just the

temporally averaged fluorescence values. In some sense, these approaches may be accessing the 'true' structure of the data in a more reliable fashion than the human.

However, these approaches also suffer from several drawbacks. Though they exploit the temporal structure of the data, they do not incorporate information from multiple channels. Though we do not expect the structural channel to have temporal information about the activity of cells, it does provide cleaner spatial information as well as a means of separating neurons from astrocytes. It is possible that information from the structural channel could be incorporated into these approaches, but a method for doing so is not immediately obvious. Neither paper referenced here mentions a multi-channel analysis. Mukamel et al. find regions of interest corresponding to active neurons as well as glial cells with calcium transients, but these types of cells were differentiated manually based on their activity profiles.

A second weakness of the matrix factorization approaches is the flip side of one of their advantages. They use the temporal dimension of the data, but since they exploit the temporal sparseness of active cells to create components they typically will not find any regions of interest that are not temporally active. Such ROIs could include inactive neurons, astrocytes without calcium transients, and blood vessels. Active cells are likely to be more interesting to experimenters, but losing access to these other features diminishes one of the main strengths of TPCI: the ability to observe and quantify the structural network surrounding active cells.

Finally, both matrix factorization approaches described above use significant post-processing or manual cleaning in order to create components that correspond to single cells.

### 3.1.2 Image processing approaches

Anecdotally, most experimental labs currently use semi-manual segmentation procedures in which a variety of image processing techniques are interactively applied to the temporally averaged fluorescence images to identify cells. For instance, the experimenter might highlight a region of the image containing a cell to initiate a local thresholding procedure which selects the boundary for the cell. The set of image processing techniques that could be used for this purpose is vast: thresholding, peak finding, and spatial filtering are just three.

There has been some recent work attempting to use image processing techniques within completely automated segmentation systems. Tomek et al. (2013) describe the first software toolkit publicly released to perform this task. Their algorithm, named SeNeCA (Search for Neural Cells Accelerated), uses several types of smoothing, locally adaptive thresholding, a watershed algorithm, and constraints on cell size to produce a segmentation. This system is fully automatic, though with 6 tuning parameters (such as amount of smoothing and size limits for cells) which must be set manually.

The SeNeCA algorithm represents a much-needed step of the community toward explicitly defined, automated cell segmentation. However, it has several weaknesses. Like the matrix factorization approaches, SeNeCA does not explicitly incorporate information from multiple channels or segment multiple classes of objects. Presumably the algorithm could be easily extended to do so by incorporating explicit expert knowledge (find cells in both channels, check whether a cell is found in both and if so it is an astrocyte). However, this would make an already somewhat complex procedure more convoluted and would require the explicit specification of rules to differentiate cell types.

Another weakness of SeNeCA is the need to manually set tuning parameters in order to achieve good performance. When evaluating the algorithm, Tomek et al.

chose tuning parameters by optimizing performance on annotated data before testing on unannotated (but very similar) data. The reported performance is therefore more accurately interpreted as an upper threshold on performance, which might decrease substantially on new data.

Finally, the SeNeCA system does not incorporate information from the temporal dimension of the experiment into its segmentation. This is, in fact, quite intentional since SeNeCA is meant to function on individual images (single time points) rather than the entire video at once. Tomek et al. argue that single-frame segmentation is important to allow experiments that require real-time segmentation (such as optogenetic stimulation), but it is unclear why previously recorded information should not be incorporated into segmentation of later frames recorded from the same location. In fact, Tomek et al. themselves mention a version of such integration in the form of removing cells which are deemed to be unreliable due to only appearing in the segmentation of a few frames. Nevertheless, the integration of information over time could be developed farther.

### 3.1.3 Feature-based approaches

The image processing approaches mentioned above nominally have access to a wide variety of features of the data through chaining several processing techniques together and combining their results using rules derived from expert knowledge. Such systems are intuitively appealing since they mimic the logic of a human annotator. However, they are also restricted in that they cannot learn from data, access features or patterns not noticed by human annotators, or easily adapt to new data with different characteristics. What I refer to as feature-based approaches also use features derived from the data, but incorporate them into formal statistical prediction frameworks such as linear regression or supervised classifiers.

Miri et al. (2011) propose a segmentation system based on linear regression of pixel time courses against behavioral experimental correlates. In their experiment, they aimed to detect cells that were responsive to eye movement. They therefore used the expected calcium responses of cells responsive to eye position ($p$) and eye velocity ($v$) as predictors. By linearly regressing each pixel against the predictors and examining normalized $Z$ scores, they identified pixels which were significantly associated with $p$ or $v$. They declared these pixels to belong to cells, and then used manual or semi-manual techniques based on simple image processing to group them into individual cells.

Though Miri et al. successfully used this regression approach for their experiments in zebra fish, it is not generally applicable since we will not usually have access to experimental covariates that we know to predict activity in the cells we are looking for. In addition, like the matrix factorization approaches, this approach can only identify active cells (in fact, only active cells with a particular tuning).

Valmianski et al. (2010) propose a more general feature-based system based on a pair of statistical classifiers used in sequence. The first classifier works on the pixel level, predicting the probability that a pixel measures from a cell based on features such as mean intensity, temporal variance, local correlation, and local covariance. The output of this classifier is thresholded at several levels, and divided into connected components which are the input to the second classifier. This second classifier predicts whether each connected component is an actual cell or a false positive based on features describing its shape, size, and the threshold level at which it was created. The thresholded output of this second classifier gives the segmented cells. The RobustBoost algorithm is used to create both classifiers, using training data provided by a human annotator.

This classification approach has several desirable characteristics, and is in fact the method I extend in this dissertation. The first classifier uses temporal characteristics

of the data such as variance and correlation structure, while the second classifier considers features describing morphological constraints. This is a richer feature set than most of the image processing approaches mentioned above. In addition, the relationship between the features and the classification is learned from data.

Nevertheless, the method has some drawbacks. Firstly, the pixel-level classifier requires pixel-level annotated data which can be difficult and tedious to obtain. Valmianski et al. use very rough training data, for which the annotator only has to identify a few regions of pixels that are definitely part of cells and some that are definitely not. This makes acquiring training data easier, but necessarily provides very little information about border cases. Valmianski et al. partially get around this problem by considering a range of thresholds on the first classifier to create input for the second classifier. However, I claim that the pixel-level classifier is actually unnecessary and that it can be combined with the mask-level classifier to create a more unified system (see section 3.2).

A potential advantage of classifier-based systems is that is is easy to extend standard classifiers to identify more than two classes. Valmianski et al. only discuss one type of region of interest (cells) making theirs a two-class classifier (cells, not cells). However, the classifier-based system that I propose can segment an arbitrary number of classes.

### 3.1.4 Evaluation of segmentations

A pressing concern in the development of segmentation algorithms is creating a good method for evaluating them. Each paper published on automated TPCI segmentation evaluates its method differently on different data, making performance comparisons between methods extremely difficult.

The fundamental problem with evaluation is that the ground truth is inaccessible.

This leaves us with two clear options: we can create simulated data where the truth is known, or we can compare our systems to human annotators rather than ground truth.

Mukamel et al. (2009), Diego et al. (2013) and Tomek et al. (2013) all create artificial datasets to evaluate their segmentation systems. These artificial datasets typically model soma as spherical bodies placed into an imaging field. Often the cells are given an activity time course, either simulated using a spiking process and known calcium transient shape or taken from a 'known' cell in real data. The background region is often populated with static distractor shapes representing blood vessels and cellular processes. Some simulations include blurring to represent the point spread function of the microscope, and all add gaussian or poisson noise to simulate shot noise.

All of the existing data simulations are limited to one channel, ignoring the structural channel. Since none of the existing segmentation methods use information from the structural channel, this is unsurprising.

Simulated data is very appealing for evaluation since results can be compared to known truth. However, the validity of these simulated datasets in predicting performance on real data is unclear. By necessity, many assumptions are made about the structure of the data and noise. Due to the complexity of TPCI data, it is unclear how to test these assumptions or compare the artificial data to real data in a meaningful way.

Even if we assume that we know ground truth, there are many ways of quantifying the quality of a segmentation. Mukamel et al. (2009) evaluate their algorithm by looking at the fidelity of extracted time courses, defined as the correlation coefficient between the extracted time course and the true time course. They use this metric to demonstrate that their method reduces the contamination of cellular signals by neuropil, but they do not discuss the performance of their algorithm in terms of

spatial filters (whether cells are found at all, if their spatial filters are accurate).

Diego et al. (2013) evaluate their algorithm according to whether it separates cells with correlated activity. This metric allows a direct comparison against the Mukamel et al. approach (which performs relatively poorly in this regard), but does not provide any assessment of the discovery rate of cells or the quality of the spatial or temporal mask characteristics.

In contrast, Tomek et al. (2013) use only spatial characteristics to evaluate their segmentation since they only consider single frames. They break down errors into four categories: split, merged, spurious, and missing. Split cells are those which are erroneously assigned multiple masks. Merged cells are groups of cells that are jointly assigned a single mask. Spurious cells are false positives, and missing cells are false negatives. Though this is in some ways a richer evaluation of performance than those described above, it still does not asses the quality of the shape of a mask (does a mask accurately identify the boundary of a cell?).

In addition to simulation results, most papers on TPCI segmentation report results on real data as well. Here, the only way to evaluate performance in more than an anecdotal way is to compare against a segmentation created by a human annotator. Valmianski et al. (2010) and Tomek et al. (2013) report results in this way, though in both cases there is no attempt to evaluate the reliability of the reference segmentation. To my knowledge, no study of inter-rater reliability has been performed for TPCI segmentation.

Valmianski et al. (2010) evaluate their classification-based segmentation system by using 5-fold cross validation on their labeled data. They report errors simply as false positives or false negatives, separately reporting performance from their two classifiers. As such, cells that were not generated as candidates by the first classifier may not be reported as errors at all (remember that the training data for the first classifier was very incomplete). In addition, there is no evaluation of the quality of

the mask shapes or boundaries.

Because of the variety of ways in which performance of these existing segmentation algorithms is quantified and reported, it is difficult to compare them in a meaningful way. Nevertheless, in an attempt to define the sort of performance that is considered state of the art in this field, table 3.1 summarizes the reported performance of existing TPCI ROI segmentation methods. As far as can be determined, the performance of my proposed system (see section 3.3) has comparable performance to these.

### 3.1.5 Characteristics of a desirable segmentation system

This section has summarized the existing work in ROI segmentation for TPCI. This current work has strengths and weaknesses which should be kept in mind when developing new approaches. Here I list what I believe to be important characteristics of future segmentation systems.

1. Uses as much of the available information as possible. Some existing approaches use temporal information as well as spatial structure. As of yet, no segmentation systems use information from both imaging channels.

2. Segments a variety of ROIs. Many existing techniques are limited to segmenting active cells. None explicitly or automatically differentiate between types of cells.

3. Uses data to learn. Matrix factorization approaches find structure in the current data, but can't learn from past data. Expert-designed systems of image processing techniques use human knowledge but can't improve or learn without manual modification of the system. Classifier approaches are able to learn as the amount of training data grows.

In addition, though not a feature of the segmentation systems themselves, future systems should be subjected to rigorous and standardized evaluation procedures. To

| **Authors** | **Technique** | **Reported Performance** |
|---|---|---|
| Mukamel et al. (2009) | spatio-temporal ICA | Median fidelity (correlation of extracted signal with true signal) of 95% on simulated data with SNR above 0.3 |
| Valmianski et al. (2010) | pair of supervised classifiers | Area under the ROC curve of 0.97 for the second classifier. $\sim 0.97$ true positive rate, $\sim 0.2$ false positive rate at chosen threshold. Five-fold cross-validation against human labels on real data. |
| Diego et al. (2013) | sparse structured PCA (confocal imaging) | Sensitivity of 94.3%, even when separating highly correlated cells |
| Tomek et al. (2013) | image processing algorithm | Percentages of split, merged, spurious and missing cells: 0.6, 1.9, 4.8, 4.8 (artificial data); 0.08, 0.3, 48.7, 5.9 (real data compared to human) |

**Table 3.1:** *A summary of the reported performance of existing ROI segmentation procedures. These summaries are necessarily brief, for a more complete report please see the appropriate paper.*

make progress in this area, it is imperative that the community clearly define the goals of segmentation and systematically evaluate how well both manual and automated procedures meet those targets.

## 3.2 MaSCS: Mask-space supervised classification for segmentation

In this section I propose a framework for segmenting TPC images that meets the three criteria defined above, while being flexible and customizable enough to be useful to experimenters. This framework is called MaSCS: Mask-space Supervised Classification for Segmentation.

Since TPC images are pixelated, the obvious unit of inference when doing segmentation is the pixel. However, individual pixels carry very limited information without considering their spatial context. Inferring whether an isolated pixel is part of an ROI is only possibly with unusually detailed knowledge about its expected time course as in the regression approach of Miri et al. (2011). For this reason, even the pixel-level classifier in Valmianski et al. (2010) in fact uses primarily functions of a pixel and it's 5 to 21 surrounding neighbors. Though the pixel is the smallest spatial unit of data easily accessible, we simply lack enough information to do inference on this scale.

In contrast, we have a great deal of information and knowledge relevant to evaluating whether a group of pixels is a likely region of interest. The ROIs are the scientific target of the experiment, and as such we can describe them. For instance, we know the plausible range for the size of a neuron's soma, and know its approximate shape. We can reasonably expect the calcium dynamics within different parts of a cell's soma to be correlated with each other.

Rather than using the pixel as a unit of inference simply because it is the experimental unit of measurement, the MaSCS method focuses on candidate ROI masks

as the fundamental unit. Unconstrained by computational constraints, this method would consider the space of all contiguous reasonably-sized groups of pixels and then select a minuscule fraction of these as an appropriate segmentation. Clearly this is impractical as the set of such candidate masks is enormous even for reasonably small images.

Fortunately, as known from previous work on this problem (and by the many experimental labs who have cobbled together their own semi-automated segmentation procedure) there are innumerable ways of generating a smaller set of candidate masks that is likely to be a superset of the correct ones. These methods include peak-finding, thresholding, spatial filtering, and all of the matrix factorization and feature-based approaches described in the previous section. It is entirely unclear which of these methods is the best, or even if one is universally superior to the others. Perhaps each is uniquely appropriate for a particular set of imaging conditions or characteristics of the target ROI.

MaSCS takes the output of one or more of these mask-generating procedures as input. Crucially, the user need not manually tune parameters of the mask-generating procedure. The goal is simply to generate a superset of the correct masks without more regard for spurious masks (false positives) than is demanded by computational concerns. This can generally be accomplished by using a range of reasonable parameters.

Once the set of candidate masks is generated, MaSCS uses supervised classification, subject to constraints on ROI overlap, to create a segmentation. Though training data is required, the annotator need only make rough judgements, limiting tedium as much as possible. In addition, as the system is used, additional training data is created allowing the system to continue to learn and improve.

Figure 3.1 presents the algorithmic flow of MaSCS in graphical form. The following sections describe each stage in detail. It is worth emphasizing that MaSCS is a general

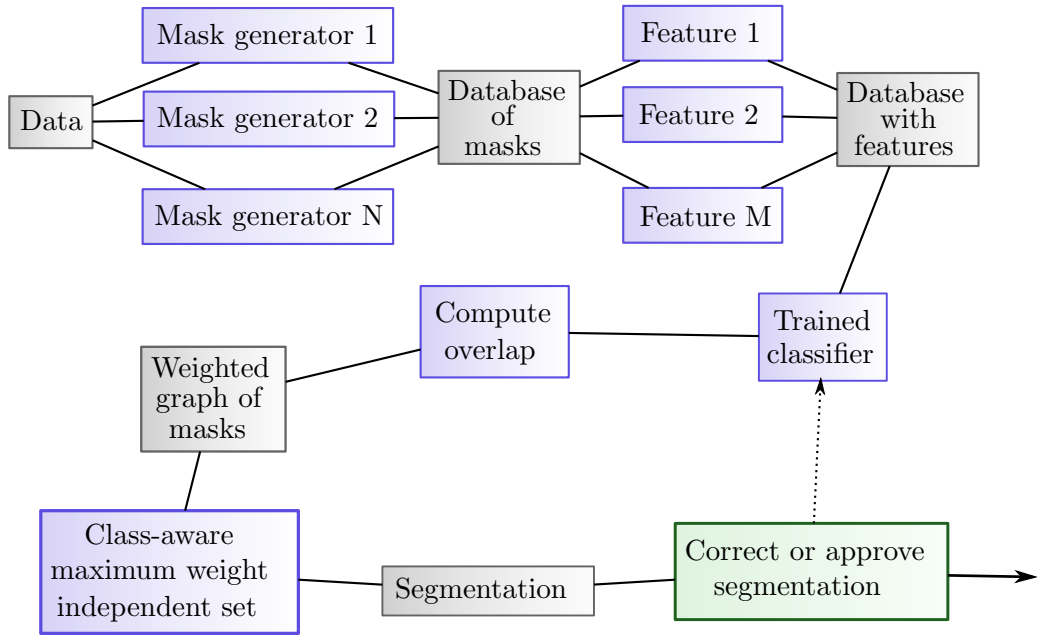



**Figure 3.1:** *Schematic of the segmentation process. Grey boxes describe the states of stored data and processing output. Blue boxes describe the processing steps which transform the data from one state to the next. The green box indicates user input. Sections 3.2.1 through 3.2.4 describes each of the elements of this schematic in greater detail.*

framework for performing segmentation. The specifics of the mask generators, feature space, and classifier are not integral to the approach. However, for the purpose of demonstration, I have chosen particular techniques for these steps to create a prototype MaSCS system. Section 3.3 discusses the evaluation and performance of this system on real data.

In theory, the MaSCS procedure could be used to find any type of ROI. This implementation finds two classes of cell somata – neurons and astrocytes. These are likely to be the most commonly relevant ROIs in TPC imaging. Finding two classes of ROIs demonstrates the multi-class ability of the MaSCS procedure while maintaining a simple prototype system.

### 3.2.1 Mask generation

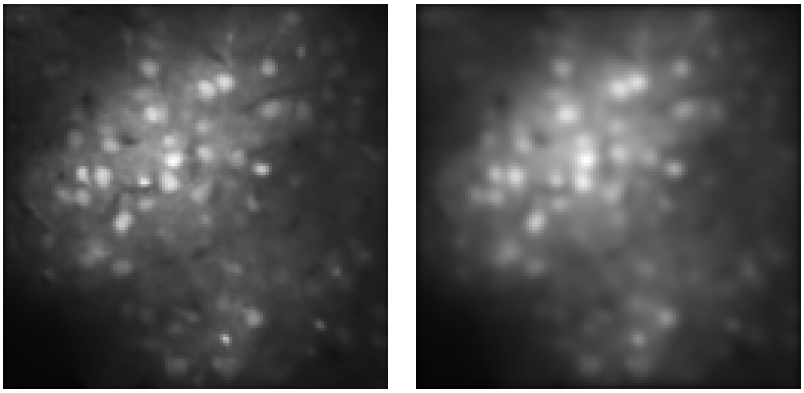
The MaSCS system takes as input a set of candidate ROI masks assumed to be over-complete. One of the strengths of this method is that it can work with and improve upon whatever mask generating technique a research lab currently uses. Nevertheless, I have developed two straightforward techniques adapted from image processing that work well for generating masks. These two techniques, Laplacian-of-Gaussian blob finding and thresholding of locally equalized images, are what I use in the implementation of MaSCS presented here.

Both of the techniques I present here for finding candidate masks work on the time-averaged data. The time dimension of the data is used for selecting masks later in the procedure, but it could still be reasonable to include mask generators that aren't blind to temporal dynamics (such as the matrix factorization procedures).

#### Laplacian-of-Gaussian blob detector

Laplacian-of-Gaussian (LoG) blob detection is a well-known peak-finding technique in the image processing literature (Lindeberg, 1998). This technique finds regions where the estimated second derivative of a smoothed image is negative - bright blobs in a greyscale image. Generally, a range of degrees of smoothing is used to detect blobs at multiple scales. Much of the image processing research on blob detection focuses on automatically selecting the appropriate scale for a particular image feature. Since I want to integrate additional information into making that choice, I do not try to make these scale-space decisions in the mask generation step, instead finding blobs at a range of scales.

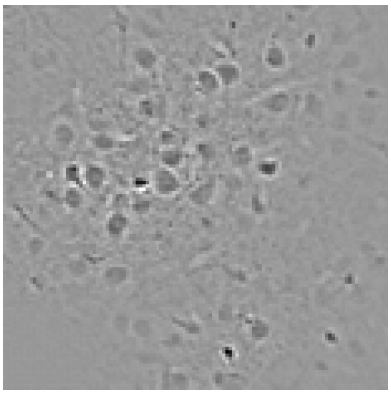
Consider an image. In this case, the image will be the time-averaged data from a single TPCI channel $c^\star$. Represent our (channel-x-y-time) fluorescence imaging data



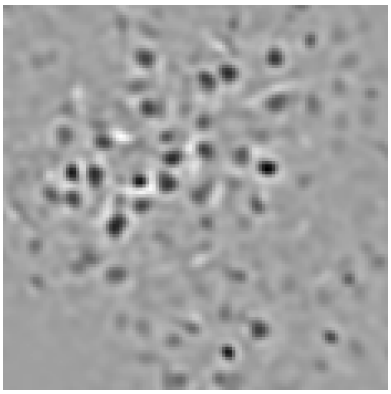
(a) Smoothed image (low)



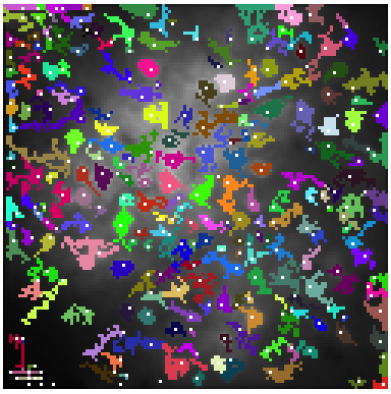
(b) Smoothed image (high)



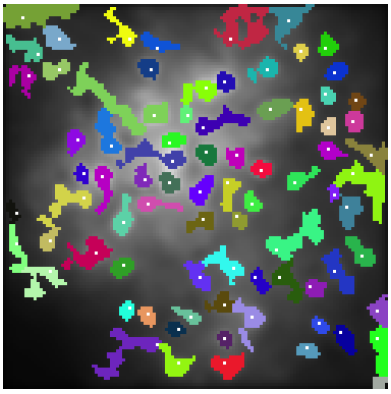
(c) Laplacian image (low)



(d) Laplacian image (high)



(e) Candidate ROIs (low)



(f) Candidate ROIs (high)

**Figure 3.2:** *Demonstration of the LoG mask generating procedure. Examples are chosen for two points in scale space (relatively low smoothing, and relatively high smoothing). (a) and (b) show the original image smoothed at the two levels. (c) and (d) show the result of convolving with the Laplacian kernel. (e) and (f) show the set of generated masks for each level. Each mask is plotted in a different color, with white pixels indicating the local maxima used to separate masks.*

as $D_{c,x,y,t}$, and let

$$\bar{D}_{c^\star,x,y} = \frac{1}{T}\sum_{t=1}^{T} D_{c^\star,x,y,t} \tag{3.1}$$

be this mean image. Chose a range of smoothing scales $s_1, ..., s_N$. For each scale, create a Gaussian kernel with that scale

$$g(x, y, s) = \frac{1}{2\pi s} e^{-(x^2+y^2)/(2s)}. \tag{3.2}$$

Let

$$D'_{c^\star,x,y,s} = \bar{D}_{c^\star,x,y} * g(x, y, s) \tag{3.3}$$

be the smoothed image created by convolving $\bar{D}_{c^\star,x,y}$ with this kernel.

Next convolve each smoothed image with a 3-by-3 kernel $K_L$ which approximates the Laplacian operator

$$K_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \approx L(x, y) = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2}. \tag{3.4}$$

Select the pixels of the resulting image with negative values as described by the binary image

$$D^{01}_{c^\star,x,y,s} = Indicator\left[(D'_{c^\star,x,y,s} * K_L) < 0\right]. \tag{3.5}$$

The last task is to divide the identified pixels in $D^{01}$ into individual ROI masks. Call the region of $D^{01}$ with value 1 the *identified region*. Assume that in the smoothed images, each candidate ROI can be associated with a local maximum in intensity. Restricting attention to the identified region, find all the local maxima and give each of these a unique identifier. Finally, assign each identified pixel in $D^{01}$ to one of the local maxima by hill climbing on the identified region of the smoothed image. That is, for each identified pixel, move to its highest identified neighbor and repeat until

there exists no higher neighbor. If the end-point of this process is one of the labeled local maxima, assign the original pixel the identifier of that maxima. Some identified pixels will not be associated with a local maxima, and these are discarded. Note that restricting the hill-climbing process to the identified region forces candidate ROIs to be contiguous.

This process will result in a set of candidate ROI masks, one for each unique identifier created in the last step. For each of these identifiers, generate a candidate mask $M_i$ which is a binary matrix giving the locations of the pixels assigned that ID.

$$M_{i,x,y} = Indicator\left[D^{01}_{x,y} = i\right] \tag{3.6}$$

Figure 3.2 shows this process in images for example data.

**Thresholding of locally equalized images**

Rather than looking for peaks in the intensity landscape as the LoG blob detector does, a thresholding approach generates candidate cells by finding contiguous regions with intensity value above some cut-off. As emphasized earlier, the range of values of TPC images tends to vary greatly spatially. This means that simple thresholding is ineffective since the appropriate thresholds (and the sensitivity of the output to small changes in threshold value) are different in different parts of the images. The result is that it is difficult to capture cells in darker, lower-contrast areas of the images. Figure 3.4 demonstrates this problem.

To account for the contrast differences across the images, I propose thresholding a locally equalized version. Specifically, I use sliding window histogram equalization.

Histogram equalization transforms an image to increase contrast by linearizing the empirical cumulative distribution function (CDF). Though the image pixel values define a discrete probability mass function, the procedure is motivated by the

continuous probability integral transform.

Let $X$ be a continuous random variable with a CDF $F_X$. The probability integral transform states that the random variable

$$Y = F_x(X) \tag{3.7}$$

has a uniform distribution. For a discrete-valued image $D_{x,y}$, we can approximate this transformation. Let the empirical CDF function at a value $v$ be

$$cdf(v) = \sum_{p \in pixels} Indicator(p <= v). \tag{3.8}$$

The histogram equalization function $H$ for a pixel $p$ is

$$H(p) = round\left[\frac{cdf(p) - cdf_{min}}{N_x \cdot N_y - cdf_{min}}(V_{max} - 1)\right] \tag{3.9}$$

where $cdf_{min}$ is the minimum value of the empirical CDF (can be greater than 1 if there are several pixels with equal values), $cdf(p)$ is the value of the CDF at the value of pixel $p$, and $V_{max}$ is the maximum desired value for the equalized image.

Figure 3.3 demonstrates the application of histogram equalization on an example temporally averaged TPC image. When the equalization procedure is applied to the entire image, the empirical CDF is linearized and the histogram is uniform, as expected.

Though histogram equalization does increase the contrast of the image, when applied to the whole image it does not solve the problem of different regions of the image having very uneven contrast. Thresholding procedures are still not effective.

To locally equalize contrast, I use a sliding window version of the histogram equalization procedure. For each pixel in the image $D_{x,y}$, consider a square window $d_{(x-r):(x+r),(y-r):(y+r)}$ with radius $r$ around the pixel. Perform histogram equalization

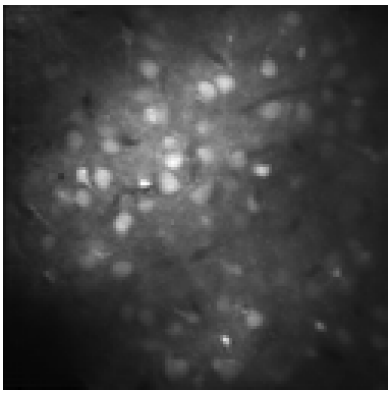
on this smaller image. Assign the equalized value of the center (original) pixel to that pixel in the final result.

The result of the sliding window histogram equalization is shown in figure 3.3. The histogram of pixel values in the image is, of course, not uniform since the equalization procedure was not applied to the whole image simultaneously. However, the contrast is increased adaptively across the image such that all regions have similar contrast levels.

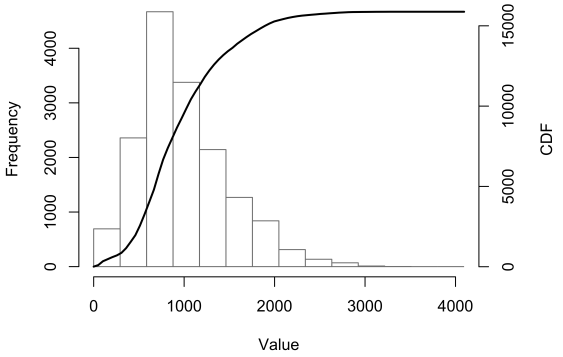
Figure 3.4 contrasts the results of thresholding the original image with that of thresholding the sliding window histogram equalized image. The equalized image allows thresholding to find relatively bright features (candidate ROIs) in all regions of the image simultaneously. One downside of the procedure is that in regions with very little signal, the equalization procedure amplifies the noise excessively. Given that the goal of mask generators for the MaSCS procedure is to generate masks without too much concern about spurious masks, this amplification of the noise is not of concern. The candidate masks that result from thresholding these noise regions do not have other characteristics expected of cells and will be discarded by the MaSCS classifier.

### Creating a database of candidate masks

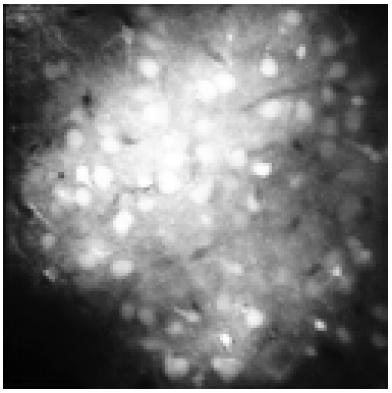
As input to the MaSCS procedure we want an overcomplete set of candidate ROI masks. With a range of smoothing or thresholding values, the two mask generating procedures mentioned above can generate large numbers of candidate masks. However, it is unlikely that one technique will generate the best mask for all cells. For instance, the LoG blob detector, as a peak-finding method, may be better at separating closely spaced cells. Thresholding of the equalized image may be better at finding cells in dim regions of the image. For some cells or regions, a higher smoothing



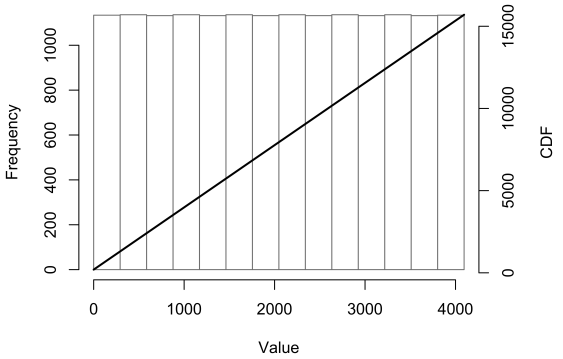
(a) Original image



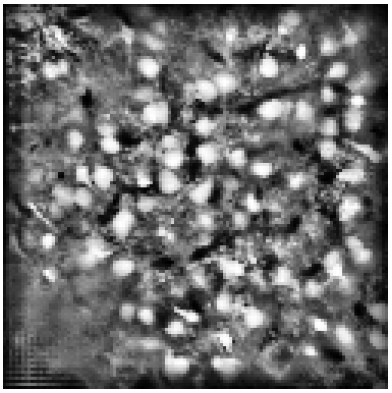

(b) Histogram and CDF (original)



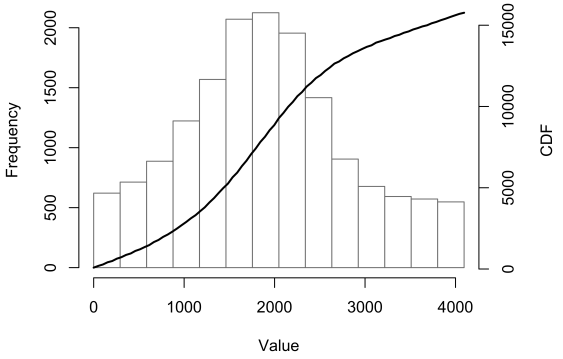
(c) Whole-image equalization




(d) Histogram and CDF (whole image)

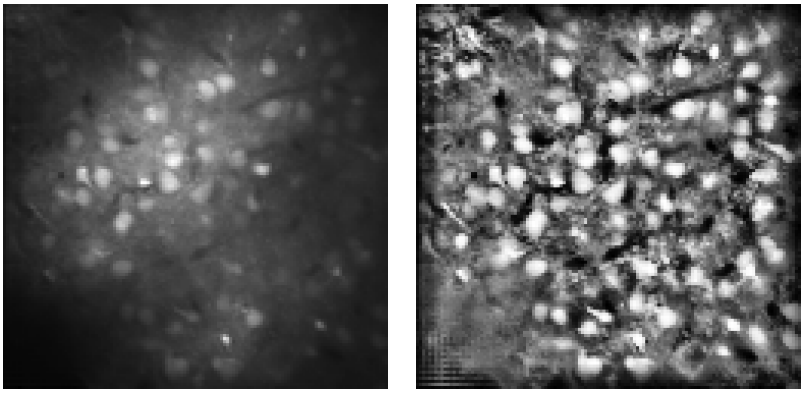


(e) Sliding window equalization




(f) Histogram and CDF (sliding window)

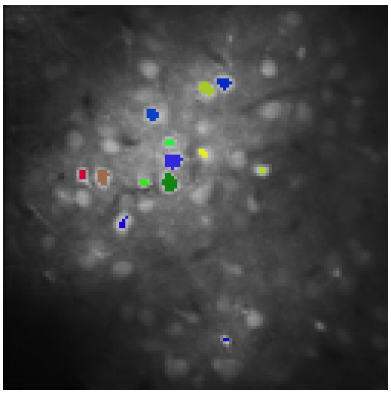
**Figure 3.3:** *Examples of histogram equalization. The first column shows the images while the second column shows the histogram of pixel values and empirical CDF (unnormalized). (a) and (b) show the original image. (c) and (d) show the histogram equalized image. (e) and (f) show the sliding window equalized image.*



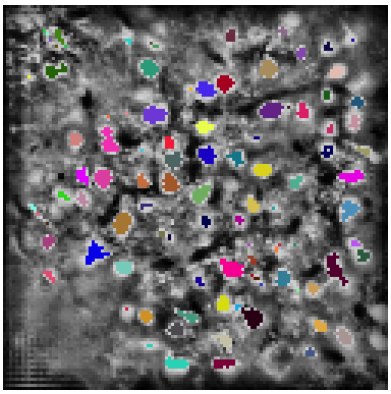
(a) Original image



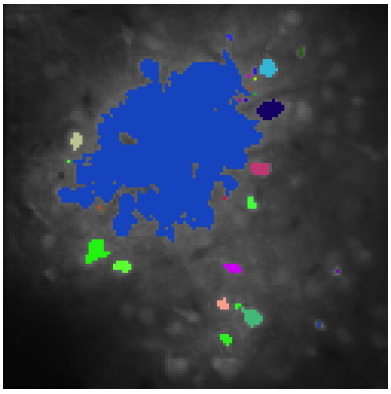
(b) Equalized image



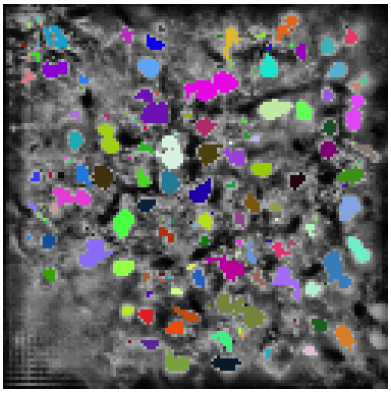
(c) High threshold (original)



(d) High threshold (equalized)

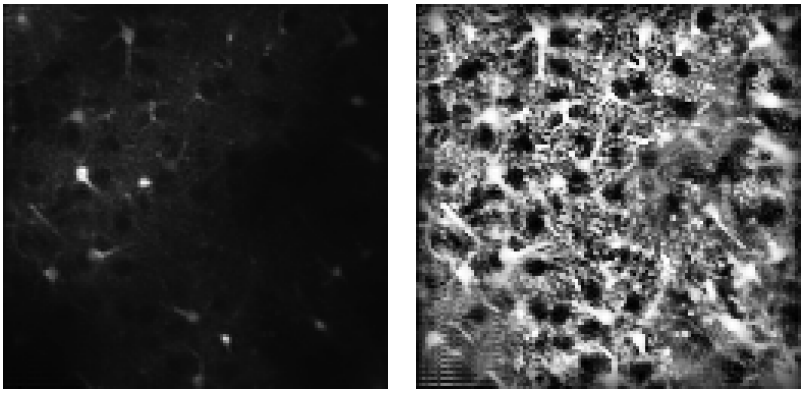


(e) Low threshold (original)

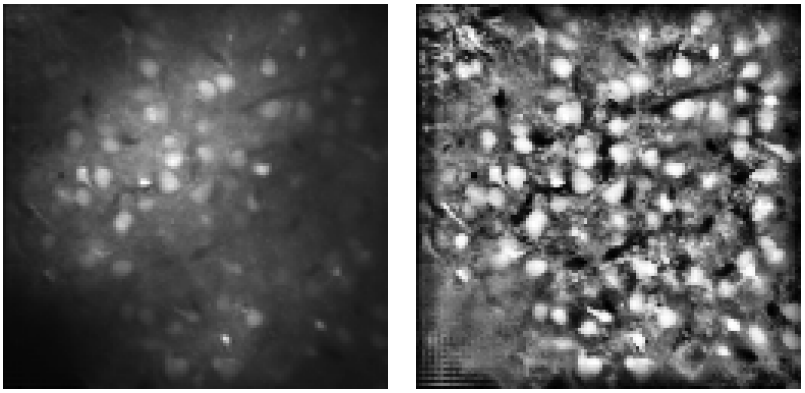


(f) Low threshold (equalized)

**Figure 3.4:** *Example of thresholding as a mask generating procedure. (b) is the original image. (b) is the sliding window histogram equalized image (with a window size of 17 by 17 pixels). (c) and (d) show the masks generated using a relatively high threshold for the original and equalized images. (e) and (f) show the masks generated using a relatively low threshold for the original and equalized images. This demonstrates how spatially unequal contrast in different regions of the original image results in poor results from thresholding, a problem fixed by sliding window histogram equalization.*



(a) Structural image (original) (b) Structural image (equalized)



(c) Functional image (original) (d) Functional image (equalized)

**Figure 3.5:** *The structural and functional mean images (as well as their equalized versions) can both be used for mask generation. Dark regions in the structural image can be informative about the location of neurons and can be found by thresholding or peak finding on the inverted image.*

parameter may be best, whereas for others a lower smoothing level may be necessary.

In addition to having multiple mask generation techniques, we also have multiple channels of data. We can apply both mask generators to both channels at a variety of smoothing levels and thresholds. Because the structural dye does not enter neurons, neural soma are dark regions in the structural image (see figure 3.5). These dark regions can be as informative as bright regions, and can be exploited by applying the previously described mask generating procedures to the inverse of the structural image. If the MaSCS procedure were being used by a different group, the mask

generating procedures might be extended or altered. The two that I use here are simply meant to demonstrate that automatic, unsupervised procedures are capable of generating a sufficiently complete set of masks. The actual performance of these procedures is hard to quantify, but is explored later in the results section (3.3).

The process of automatic mask generation can generate a very large number of masks. To efficiently store and process these masks, I use a SQLite relational database for each experiment. The masks are stored in the databases as vectors giving the indices of the mask pixels. This sparse representation reduces storage costs significantly since most masks are quite small relative to the size of the images. It also allows for faster searching and comparison of masks.

Associated with each mask are its features. These features are discussed in more detail in the next section, but include information about the source(s) of each unique mask. A mask may be generated by multiple mask generators and/or at multiple smoothing levels or thresholds. Each unique mask is only stored in the database once, but each time it is generated a feature associated with the mask indicates which generator(s) produced it.

Even with efficient storage of the masks, the task of generating and processing the masks for storage can be somewhat computationally intensive if many masks are created. For this demonstration of the MaSCS procedure, the mask generators created up to 100,000 masks for each experiment, of which up to approximately 20,000 were unique. Creating and storing this many masks took on the order of 15 minutes with current code, but this could be significantly optimized. Additionally, as the MaSCS procedure gives us some information about the quality of mask generators, we could almost certainly use this information to reduce the number of masks generated in the future without compromising performance.

### 3.2.2 Feature space

The features that can be used in the MaSCS system are limited only by creativity and computation. For this implementation of the system I have used features in three categories: data-based features, shape-based features, and source features.

Data-based features draw information from the data underlying a mask in both channels. Including features based on the two channels and the time dimension allows comprehensive use of the available data. These features can include such things as

- mean (over space and time) fluorescence for both channels
- mean fluorescence of the equalized data
- mean pairwise temporal correlation between pixels in the mask

The size and shape of cell somata are well understood and should be able to be leveraged by the classifier. Shape-based features incorporate this information into the segmentation algorithm. These features include such things as

- size of the mask
- ratio of the mask size to that of its convex hull
- ratio of the mask size to that of its bounding box
- number of holes (non-mask pixels surrounded in 3 or more cardinal direction by mask pixels)

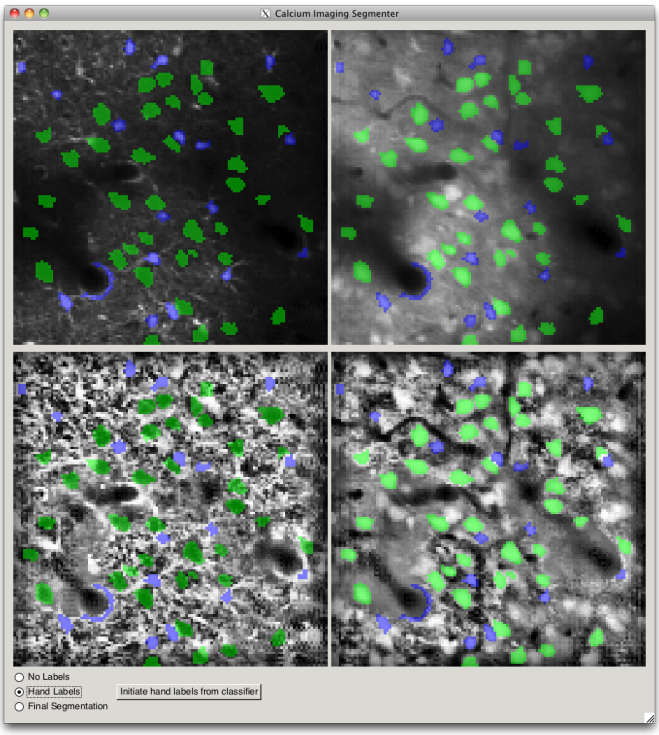
Finally, source features indicate which mask generator(s) created the mask and how often the mask was created by each generator. These features may be helpful in classification, but more importantly, including them can help us learn about the quality of the various mask generators.

### 3.2.3 Annotation

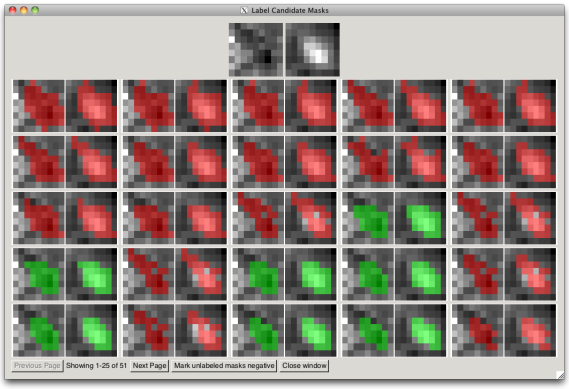
MaSCS uses a supervised classifier to create the final segmentation from the large set of candidate masks originally generated. This supervised classifier requires training data. Since my goal is to automate the human annotator, I want to obtain a large set of masks that are labeled with their ROI classes (or as spurious masks) by a human. However, obtaining this training data is nontrivial for two reasons. First, a human should not be required to examine and annotate tens of thousands of masks for each experiment. Second, though humans may be able to identify the presence or absence of a cell, it is very difficult for a human to annotate on the pixel level. A human will almost certainly not be able to distinguish the exact boundary of cells with confidence, and so will likely find several masks to be equally acceptable for an identified cell.

To help with the annotation process, I built a prototype graphical user interface (GUI) for annotation. Figure 3.6 shows a screen shot of this tool. The user can see a collection of time-averaged images from the experiment being labeled (for instance, the original mean images and their equalized version). He then selects a region containing an ROI by highlighting that region with his mouse. A popup window appears presenting all masks from the mask database that are entirely contained within the selected region. These masks are sorted by size, allowing the user to quickly find the range of masks he deems appropriate for the ROI in question. He indicates masks he wishes to select by clicking on them to cycle through the available list of ROI class labels. Finally, he may specify that all non-labeled masks within the selected region should be labeled as spurious. Using this tool, the annotator is unlikely to label all masks in the database, but is likely to provide labeled masks for each ROI and for spurious masks that are similar.

For each ROI, the annotator will almost always select multiple masks as appro-




(a) Main program




(b) Mask labeling window

**Figure 3.6:** *Screenshots from the prototype GUI for providing training data to the classifier (or correcting an automating segmentation). The main window,shown in (a), shows the original and equalized images from both channels. Masks that have already been selected as appropriate for ROIs are shown in green (neurons) and blue (astrocytes). When the user selects a region of any of the images, the mask labeling window appears. This shows the candidate masks that are contained within the selected region. The user clicks on a particular mask to label it as appropriate for the ROI. In this example, green masks are those selected as appropriate for this neuron and red masks have been deemed incorrect.*

priate. This creates an unusual structure in the training data provided to the MaSCS classifier. Though the data may be seen as standard multi-class labeled data, it more accurately can be seen as clustered labeled data. For each ROI, the annotator labeled a cluster of masks of which he believes *at least one* describes the unknown true shape of the ROI.

### 3.2.4 Mask selection by classification

The task of the MaSCS classifier is to select one mask from the mask database for each ROI in an experiment. In general, this will be an unannotated experiment, but the classifier may also be used to refine the annotation provided by the human. The annotation frequently specifies multiple masks for each ROI and the classifier can be used to select the best one.

The nature of the segmentation problem defines several characteristics that the MaSCS classifier must have. It must be a multi-class classifier, it must handle very imbalanced training sets for different classes, and it may not depend strongly on having known class-conditional distributions for the predictive features. A classification framework that fulfills these criteria and is known to have generally good performance is Random Forests. I therefore use a standard Random Forest (RF) classifier in this work. However, due the unique structure of the problem the RF classifier is only used to estimate class probabilities for each mask and the final classification (or segmentation) is done with a custom procedure that considers the clustering of masks and the need to select just one from each cluster.

The Random Forest, introduced in Breiman (2001) is a randomized ensemble algorithm built on top of the basic decision tree. A decision tree is a classifier which partitions the feature space based on a sequence of binary judgements about features. The tree is built from the root by repeatedly choosing splits which optimize some

criteria.

Consider a set of labeled data where $\mathbf{X_{d,f}}$ is a matrix with $N_d$ data elements on the rows and $N_f$ feature values in the columns. $\mathbf{Y}$ is a vector of length $N_d$ giving the class labels for the data. Before growing the decision tree, all data points are assigned to the root of the tree. As the tree is grown, data points are moved to the appropriate leaf of the tree. At each step of growing a decision tree we split a leaf of the existing tree based on a feature. We choose this feature $f$ to give the best split of the data. 'Best' can be defined in a number of ways, but a common criteria is the Gini impurity measure.

Gini impurity indicates the chance that a randomly chosen datum in a leaf of the tree would be incorrectly labeled if assigned a label according to the empirical distribution of class labels in the leaf. Consider a group of data points $D$, each assigned one of $C$ class labels. The Gini impurity of this group is

$$G(D) = \sum_{c=1}^{C} N_c(1 - N_c) \tag{3.10}$$

where $N_c$ is the number of data points in the group with label $c$.

At each step of building the tree, we choose the split resulting in the lowest average Gini impurity in the resulting leaf nodes. We repeat this process until the data points in each leaf of the tree all have the same label.

Random Forests use an ensemble of decision trees grown on bootstrap samples of the data using random subsets of features. The basic outline of the Random Forest algorithm is as follows. Given $N_d$ data points, each with $N_f$ features, choose a number of trees $N_t$ and a number of features to consider at each decision point $M_f << N_f$. Then, for each tree

1. Create a random training set by drawing a bootstrap sample from the original data (draw $N_d$ data points randomly with replacement).

2. At each branching point of the tree, randomly select $M_f$ features and choose the best split based on those features.

3. Repeat 2 until the tree is fully grown.

Once the ensemble of trees is created, new data is classified by each tree in the collection. The probability of a new data point belonging to a particular class can be estimated as the fraction of the trees in the forest which predict that class.

Random Forests have a number of characteristics which make them a good choice of classifier for the MaSCS system. They are known to avoid overfitting, they make no distributional assumptions on the features included in the classifier, they are relatively fast for both training and testing, and they provide a measure of variable importance. As a model selection technique, these measures of variable importance can be used to work backwards and learn about the quality and informativeness of mask generators and data features. Those generators and features that are uninformative can be reconsidered or removed from the MaSCS procedure for efficiency in the future.

MaSCS as implemented here uses a standard Random Forest classifier (as implemented in the *ramdomForest* R package) to estimate the class probabilities for each candidate mask in an experiment. However, it is not sufficient to simply assign the class label with the highest probability to each mask since this will result in many masks being chosen for each ROI. To solve this problem, the MaSCS procedure frames segmentation as a graph problem – a variant on the maximum weight independent set (MWIS) problem.

Assume that we want only one mask for each ROI, and that a single pixel may only belong to a single ROI. We want to select the best candidate mask for each ROI such that no two masks overlap. Consider all candidate masks and the estimated class probabilities generated by the RF classifier. For each mask, assign a class accordingly. Discard all masks whose assigned class is *noise*. With the remaining masks, create

a graph. Let $G$ be this graph where each vertex $V_i$ corresponds to mask $i$ and $E_{ij}$ defines the edge matrix where $E_{ij} = 1$ if and only if mask $i$ and mask $j$ share at least one pixel.[1] Assign each vertex a weight $w_i$ equal to the classifier's estimated probability that the mask is in its assigned class.

With a small modification discussed below, the problem of choosing the best non-overlapping masks can be framed as the well-known maximum weight independent set (MWIS) problem. Formally, the MWIS problem for the graph $G$ is to find the binary vector $S^*$ which indicates the group of non-connected vertices with maximum total weight

$$\begin{aligned} S^* &= \underset{S}{\operatorname{argmax}}\ \mathbf{w}^T S \\ s.t. \quad & s_i \in 0, 1 \quad \forall i \\ & S^T E S = 0 \end{aligned} \tag{3.11}$$

The MWIS problem is an NP-hard integer program, but in practice for the MaSCS procedure the graph $G$ will consist mostly of a collection of disjoint cliques corresponding to the several masks selected by the classifier for each ROI. The MWIS problem then reduces to selecting the element of each clique with the highest weight. This is not computationally intensive.

If $G$ is indeed a collection of disjoint cliques, the problem is solved. However, occasionally closely-spaced ROIs will result in cliques which partially overlap. A slight modification of the MWIS problem which improves segmentation performance

[1] As a side note, the most computationally intensive part of the modified MWIS mask selection procedure is computing the overlap matrix (edge matrix) for the masks. The naive solution, checking each pixel of two masks against each other, is linear in $p$, the number of pixels in the image. If we use the sparse representations of the masks, we need to compare two vectors of indices to see if they share any elements. The naive solution to this is $m * n$ where $m$ and $n$ are the numbers of pixels in the two masks. If we assume the sparse masks are sorted lists of indices, we can reduce this to $m * log(n)$ by performing a binary search on the second mask for each index present in the first mask. With this runtime, it is possible to compute the complete overlap matrix for tens of thousands of candidate masks in just a few seconds. This same efficient algorithm is also useful in the mask annotation GUI for efficiently finding the masks contained within a target region.

is to acknowledge the multi-class nature of the segmentation by considering both the clique structure and class labels of the vertices in $G$. At a high level, the goal is to assign the masks (vertices) into groups which correspond to ROIs and then require that the segmenter select one mask from each group. The challenge is to define the groups without a priori knowledge of the ROIs. MaSCS uses the following heuristic.

We assume that all masks that correspond to an ROI will have the same assigned class and be mutually overlapping (be a clique). We can therefore start by finding all the maximal cliques in the graph which contain masks of only one class. However, an ROI may have masks which overlap with some of those in a nearby ROI. When this happens, there will be at least three maximal cliques in the graph instead of the two we want (one for each ROI and at least one containing the overlapping masks between the ROIs). To remove these extraneous cliques, first consider the set of all maximal cliques in the graph. Assign a group ID to any clique which contains a mask not in any other clique. We assume that these ID'ed cliques correspond to the ROIs. Each mask will be in at least one of the ID'ed cliques. If it is in more than one, remove it from consideration. Such masks, if selected, would preclude the selection of an additional mask from either of its cliques, violating our assumed correspondence between cliques and ROIs. After removing these masks, we will be left with a collection of disjoint cliques, and can easily solve the MWIS problem.

The clique-forming heuristic described here encourages splitting closely spaces ROIs rather than merging them into one larger ROI. Anecdotally, this is usually the correct decision. The frequency with which the heuristic is necessary is low enough that its impact on overall performance is small.

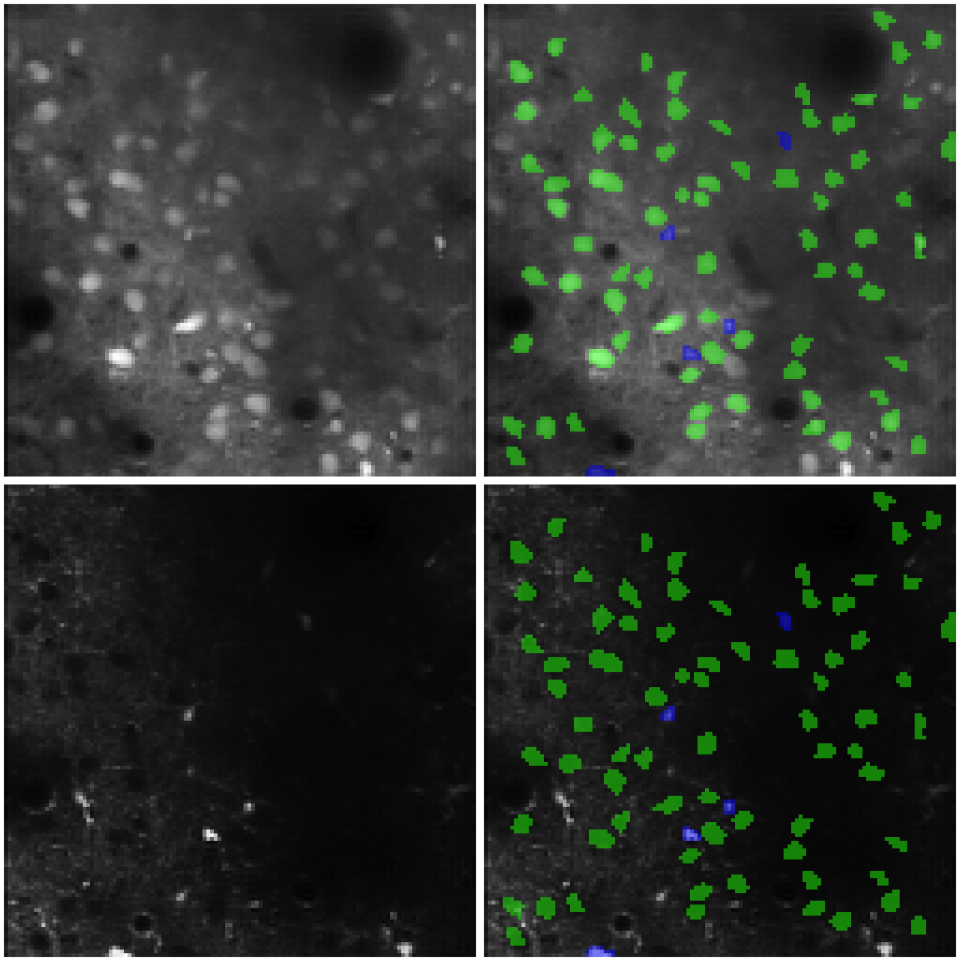


**Figure 3.7:** *An example segmentation for one experiment creating using the MaSCS classifier trained on the remaining 19 experiments. The left images are the functional (top) and structural (bottom) mean images for reference, the right images have the segmentation superimposed. Blue regions are classified as astrocytes, green regions as neurons. Several cells are missed, especially around the edges of the image where the mask generators do not perform as well. Of the 67 neurons segmented, 27 were not identified by the annotator. On post-hoc inspection, many of these are plausible cells missed by the annotator, but some are in fact spurious.*

## 3.3 Results

Once the MaSCS classifier is trained, it can produce a segmentation for new data. Figure 3.7 shows an example of such a segmentation. This example is a completely automated segmentation. However, the MaSCS classifier can also be a part of a semi-manual segmentation system that improves with use. Starting with the segmentation produced by MaSCS, an experimenter may manually correct it using the same interface used to create training data. The corrections provided by the annotator can be used to constrain the MaSCS segmentation (forcing a mask to be selected for each ROI identified by the experimenter and removing masks marked spurious). In addition, the corrections provide additional training data that can be used to retrain the classifier.

In either use case, we need a way to evaluate the performance of the segmenter. As discussed in section 3.1.4, evaluation is a difficult and non-standardized task. For this work, I evaluate the performance of the MaSCS system based on 20 representative TPCI experiments from the data described in 2.5. These 20 experiments are each from a different region of the somatosensory cortex in 4 rats. The type of stimulation, number and distribution of cell types, and density of cells varies between the experiments though they all have the same spatial and temporal resolution. I evaluate performance with reference to a human annotator, since ground truth is unavailable.

For the MaSCS system, there are two places where error could be introduced. The first is in the mask generation step. If no mask generator creates an appropriate mask for an ROI, that ROI cannot be segmented. The second is in the mask selection step. The classifier may not select the same masks as chosen by a human annotator.

Anecdotally, the first type of error is common around the edges of images due to edge effects of the convolution and local equalization procedures. Away from the edges, these errors were rare but did occur occasionally. I attempted to assess

mask generator errors by comparing the segmentation produced by an annotator independently with that produced using the MaSCS mask annotation system. Since pixel-wise independent annotation is difficult and tedious, I compared the MaSCS segmentation to a pseudo-segmentation in which the annotator simply marked each cell with a dot rather than defining the complete ROI. The annotator performed this *free labeling* twice, several weeks apart. Figure 3.8 compares the number of ROIs labeled in the two free labeling sessions and the mask annotation session, all done by the same annotator, with each session separated from the others by several weeks (the mask annotation was done first, followed by the two free labeling sessions).

From this small amount of data, it does not appear that the annotation is severely limited by missing masks since the annotation does not routinely identify fewer ROIs. However, there may be some effect of using the annotation GUI instead of free labeling since the annotation generally finds either more or fewer ROIs than both free labeling sessions for each experiment. Why this would be the case is unclear.

The second, and more common, form of error in the MaSCS system is error in mask selection. This can be assessed by comparing the MaSCS output to held-out annotated training data. Because the annotator generally selects multiple masks per ROI and the MaSCS segmenter selects just one, the comparison is not trivial. I evaluate performance using the following categories

- *Correct* ROIs are those where the segmenter selected one of the masks chosen by the annotator.

- *Marginal* ROIs are those where the segmenter selected a mask that was not selected by the annotator but that overlapped with a chosen mask. That is, the segmenter identified a correct ROI but may have chosen a poor mask shape.

- *Missed* ROIs were annotated by the human but had no mask selected by the segmenter.

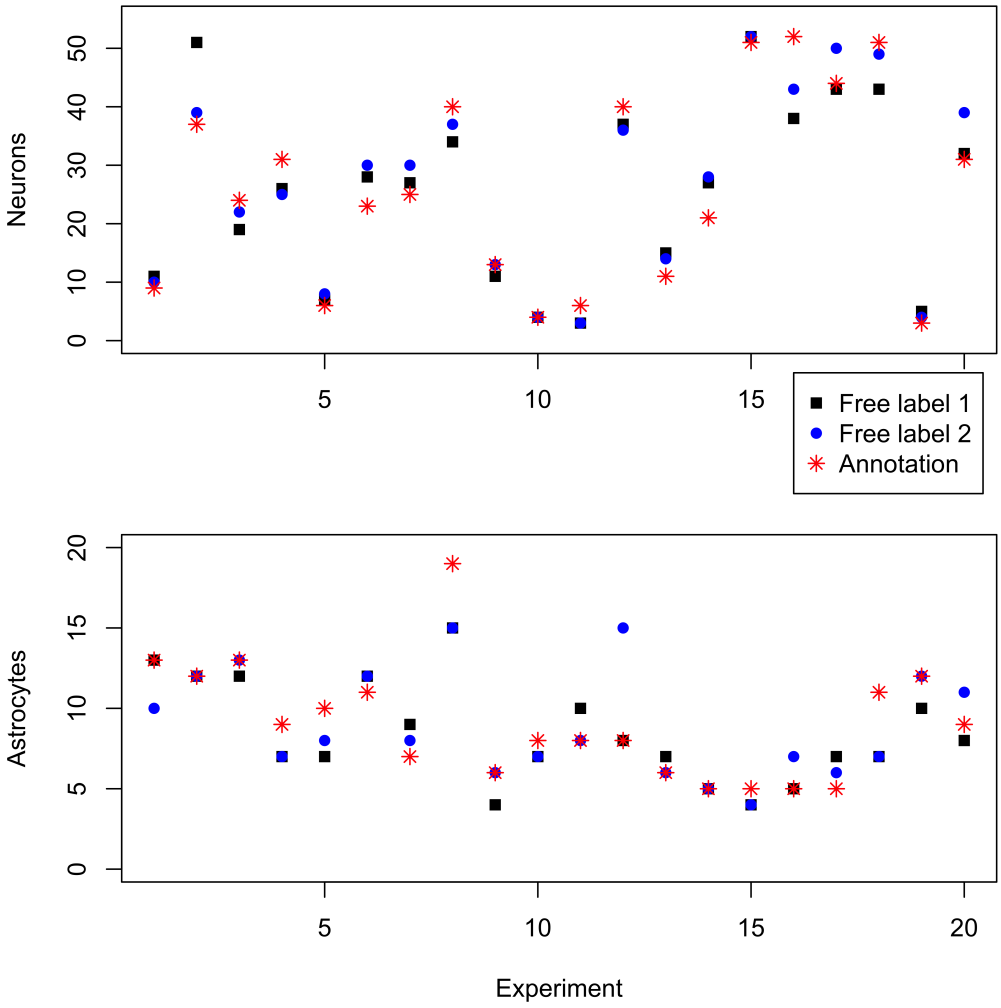


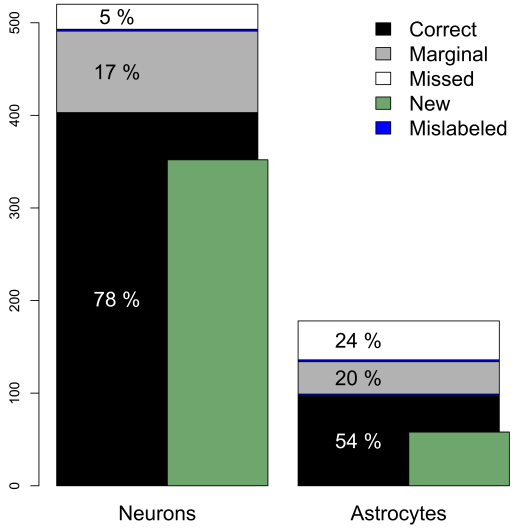

**Figure 3.8:** *Comparison of the number of ROIs found by the same annotator in different annotation sessions. The two free labeling sessions required the annotator to mark each ROI with a dot. The annotation session used the GUI shown in figure 3.6 to select masks from the automatically generated MaSCS database. From this small amount of data, it does not appear that the annotation is severely limited by missing masks since the annotation does not routinely identify fewer ROIs. However, there may be some effect of using the annotation GUI instead of free labeling since the annotation generally finds either more or fewer ROIs than both free labeling sessions for each experiment.*

- *New* ROIs are selected by the segmenter but not marked by the annotator.
- *Mislabeled* ROIs are correct or marginal masks selected by the segmenter that are given the wrong label.

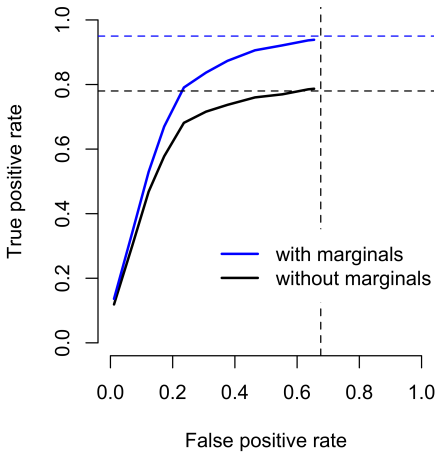
Figure 3.9 shows the performance of the prototype MaSCS system evaluated using 20-fold cross validation on the 20 annotated representative experiments. For each experiment, the MaSCS classifier was trained on the remaining 19 experiments and tested on the held-out experiment. The error is reported separately for the two segmented classes (neurons and astrocytes). The performance is much better for neurons, which may be due to the fact that there are significantly fewer astrocytes and therefore less data available for training. Notably, there are very few mislabeled masks.

By default, the classifier assigns to each mask the class with the highest estimated probability. To get a more complete idea of how performance scales, we can impose thresholds on the minimum required estimated probability for a mask to be classified as an ROI. This allows us to estimate receiver operator characteristic (ROC) curves that show how the number of false positive (*new*) masks scales with the number of true positives (*correct* and *marginal*).

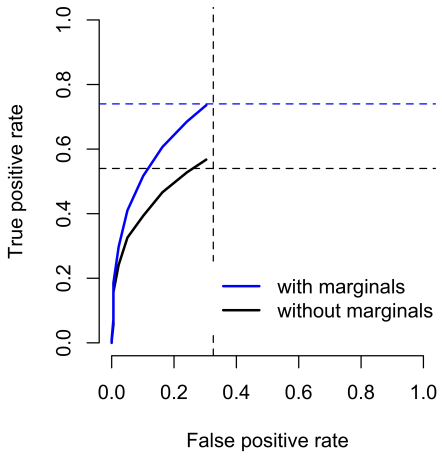
Figure 3.9 gives the results summed over all experiments. It is worth noting, however, that the actual performance varies substantially between experiments, and is, unsurprisingly, related to the number of ROIs present in the experiment. Figure 3.10 shows the performance for individual experiments. Each experiment was segmented using a classifier trained on the remaining 19 experiments, and the performance was evaluated in the same way as described above. This performance is plotted against the number of ROIs (of both types) annotated in the experiment. This figure shows that, especially for astrocytes, performance is highly variable between experiments. Experiments with very few ROIs tend to have the worst performance. This may re-




(a) Performance with most likely class




(b) ROC curve for neurons




(c) ROC curve for astrocytes

**Figure 3.9:** *Performance evaluated against annotated training data using 20-fold cross validation on 20 different experiments. Across all 20 experiments, there were 521 annotated neurons and 178 annotated astrocytes. (a) shows the performance of the default classifier. (b) and (c) show ROC curves for the classifier estimated by imposing a minimum threshold for a mask to be classified as a cell. As this threshold increases, fewer masks are selected by the classifier. The ROC curves show how the number of* new *cells scales with the number of* correct *and* marginal *cells. The dotted lines indicate the performance of the default classifier.*

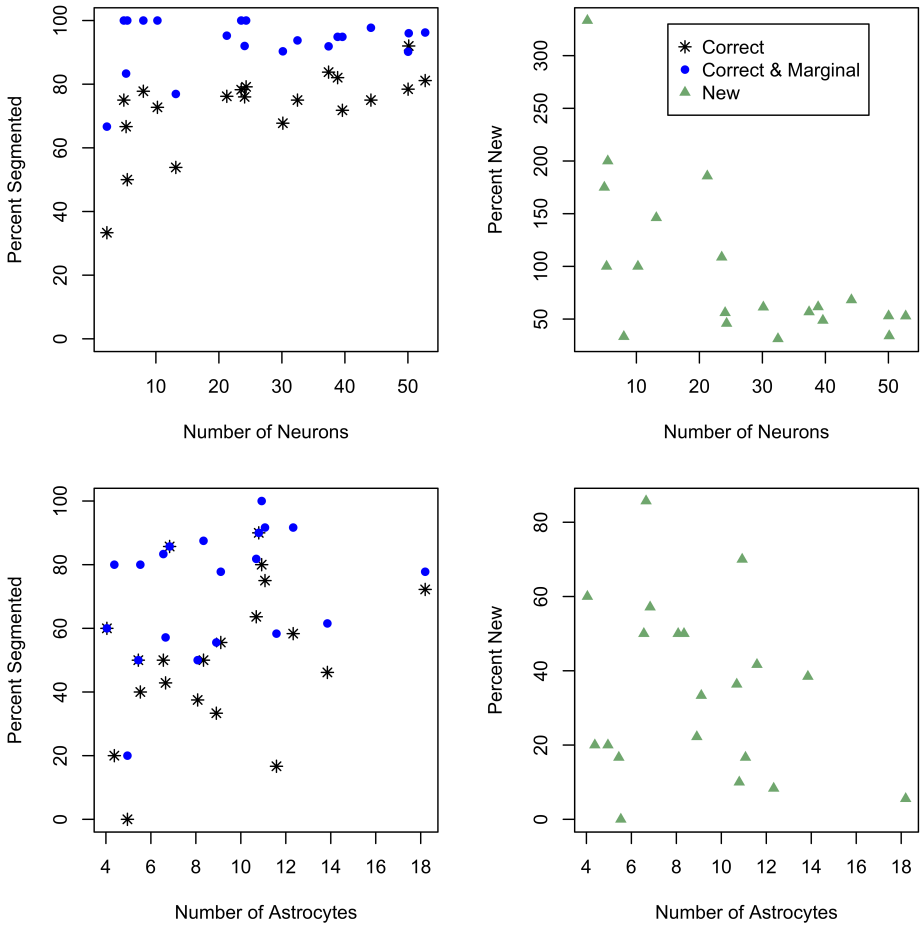



**Figure 3.10:** *Performance for each experiment, evaluated using 20-fold cross-validation. The plots show performance plotted according to the number of annotated ROIs in the experiment. Two patterns are noticeable: the percentage of correctly segmented ROIs increases with the number of ROIs in the image, and the percentage of new/spurious ROIs (calculated as (number annotated)/(number new)) decreases.*

sult from these images having large regions of noise which gets amplified by the mask finding procedures and not adequately differentiated by the features in the MaSCS classifier. Fortunately, these are also the easiest and least tedious experiments to segment or correct by hand.

Finally, one of the advantages of the MaSCS framework is that it can be iteratively improved as it is used in a lab. When analyzing large quantities of data, experimenters

may want to use the output from MaSCS without further editing. But for smaller amounts of data for which better replication of the human annotator is necessary, experimenters may want to treat the MaSCS segmentation as a time saving starting point, correcting the resulting segmentation by hand. These corrections can be used as additional training data. Figure 3.11 demonstrates how performance of the MaSCS algorithm improves as additional training data is used.

To test whether correcting the segmentation improves performance, I initiated a classifier using two annotated experiments. I used this classifier to segment another two experiments, and then corrected these segmentations. I then used this new training data to create a new classifier, repeating this process to generate 8 classifiers. Because the impact of individual experiments on a classifier is variable, I repeated this entire process three times with different random orderings of the experiments. I evaluated each classifier against the previous training annotations of the experiments not used to train it. Figure 3.11 shows the results of this process. The figure shows the combined performance of the three iterations, as well as the performance on individual iterations. This is a preliminary analysis, but does suggest that incorporating additional training data created while correcting segmentations improves the performance of the MaSCS classifier.

Currently, the MaSCS system does not include in its output any notion of confidence in the ROIs that it returns in the segmentation. However, there are several ways that this confidence could be estimated. One is to look at the estimated probability of the selected mask as returned by the classifier. Another is to look at how many candidate masks are identified for each ROI (before the maximum weight independent set procedure to select a single mask). False positive masks tend to have lower confidence according to these measures. Figure 3.12 shows a summary of the confidence analysis. This analysis was done using 20-fold cross-validation (training on 19 experiments, evaluating the 20th). For each correct, marginal, or false positive

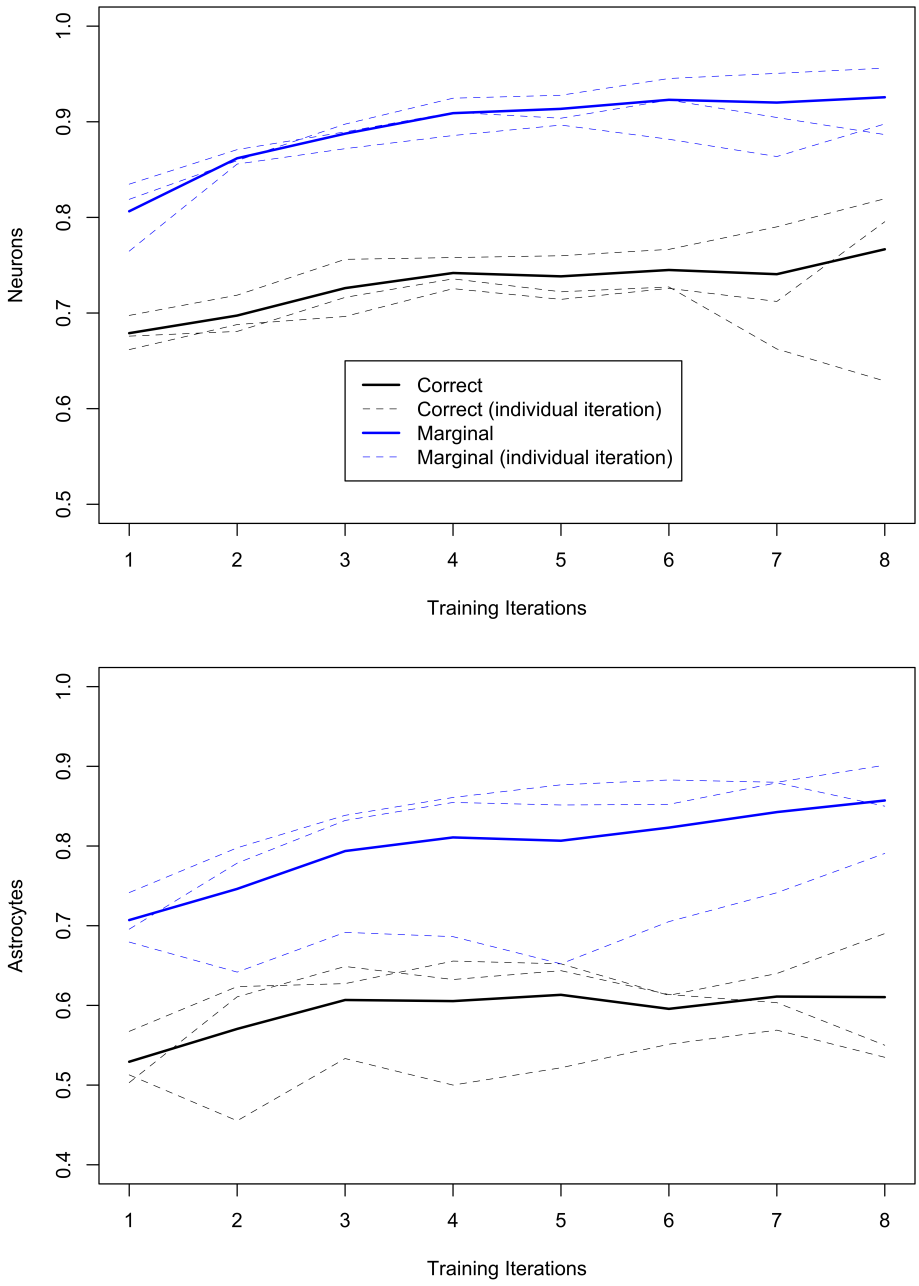



**Figure 3.11:** *Performance changes as corrections to segmentation are used as additional training data. The classifier was used to segment two experiments, those two experiments were corrected, and a new classifier was generated. This was repeated to create 8 classifiers, each with new data. This entire process was repeated 3 times using a different ordering of the experiments. Performance was evaluated by evaluating each classifier on the experiments that didn't contribute training data, with the previous training segmentation of these experiments being used as ground truth. The solid lines above show the combined performance over the three iterations. The dotted lines show performance on each iteration. Different experiments have varying effects on the quality of the classifier, but overall, correcting the segmentation improves performance.*

ROI, the number of identified masks and the probability of the selected mask was recorded. As the boxplots show, the correct ROIs tend to have highest confidence, and the false positive ROIs tend to have the lowest. The distributions of these confidence metrics overlap significantly, making it impossible to accurately remove false positives based on confidence. However, incorporating confidence into the output of the MaSCS procedure could be useful during the manual correction of segmentations or during later analysis.

### 3.3.1 Discussion of results

In the previous section, I presented the results of my prototype MaSCS segmenter in several ways. As discussed in section 3.1.4 there is no established evaluation criteria that can serve as a common metric to compare this performance to that of existing segmenters. Nevertheless, it is worth comparing, as far as is possible, the performance of the MaSCS system with that of the most similar previous systems: the double classifier system of Valmianski et al. (2010) and the image processing algorithm of Tomek et al. (2013).

Valmianski et al. (2010) give the ROC curve for their mask-level classifier. For this second-stage classifier, they report a false positive rate of 20% with 3% of annotated cells missed. This is better than the MaSCS performance (67% false positives with 5% missed neurons). Valmianski do not discuss the idea of marginally correct masks, or discuss whether either their classifier or annotator ever selects multiple overlapping masks for an ROI. Neither do they provide an evaluation of how many cells had no masks generated for their second stage classifier.

Tomek et al. (2013) report a false positive rate of 48.7% with 5.9% of cells missed. This is comparable to the the MaSCS performance in terms of false negatives, though with fewer false positives. However, the Tomek results were reported after tuning

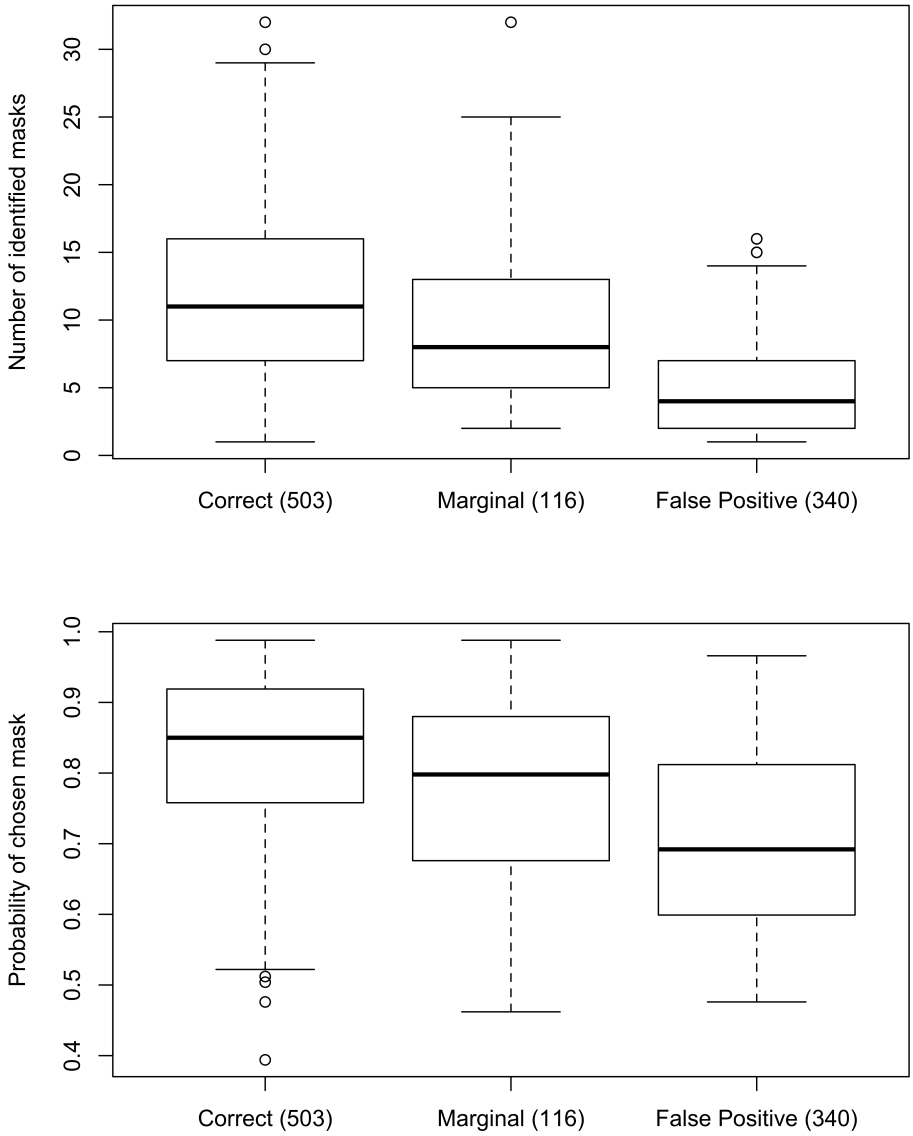



**Figure 3.12:** *Confidence measures by segmentation quality for ROIs. The boxplots show the distribution of two confidence measures for correct, marginal and false positive ROIs, evaluated using 20-fold cross-validation. The top plot shows the number of identified masks for each ROI before the selection procedure to select just one. The bottom plot shows the probability of the chosen mask for each ROI as estimated by the classifier. The numbers in parenthesis are the number of ROIs in each category.*

the algorithm's performance on images from the same experiment as those used for testing.

Note that there are numerous reasons why these direct comparisons may not be valid. The evaluations are performed on different data with different annotators and different evaluation criteria. There is no data or code publicly available for the Valmianski et al. approach. Though Tomek et al. will be publishing their code, it is not yet available as of this writing. This makes true direct comparison on the same data difficult.

When comparing the performance of MaSCS on a single class (neurons) to existing single-class segmenters it is comparable but does not show an improvement, mostly because of a higher false positive rate. Nevertheless, the MaSCS system and the work presented here advances the field of calcium imaging segmentation in several ways. The MaSCS system is the only existing explicitly multi-class segmenter. The MaSCS system is also the first segmenter to explicitly incorporate feedback from use in the lab into improving the system.

Considered on it's own, the most concerning part of the performance of the MaSCS system is the very high false positive rate. Of course, some of these false positives may in fact be true ROIs that were missed by the human annotator, but many of them are likely spurious. Tomek et al. (2013) argue that false positives are less concerning than missed ROIs, especially if segmentation algorithms are used for planning scanning paths which record only from segmented ROIs. However, for path planning as well as analysis, spurious ROIs introduce noise and are not desirable.

One possible reason for the high false positive rate is the selective labeling created by the annotation procedure. The annotator labels ROIs as positive examples, and labels spurious masks within the same region as negative examples. This leaves many masks in other regions of the image unlabeled. It would be possible to treat all unlabeled masks as negative examples for the purpose of training the classifier.

This does indeed decrease the number of false positives generated by the resulting MaSCS classifier, but also increases the number of missed ROIs. Such an approach is also conceptually undesirable as it does not allow the annotator to avoid making a judgement about unclear ROIs. In fact, an evaluation metric that took into consideration ROIs that the annotator considered possible but not certain would be ideal for further assessment of the MaSCS procedure.

As shown in figure 3.12, false positives tend to have lower confidence according to at least two measures. Though these are not sufficient to separate and remove false positives, they may be useful to assist experimenters in identifying and removing false positives from segmentations.

## 3.4 Extensions

The prototype MaSCS system developed here has not been optimized in terms of selecting mask generators or features. This work should be viewed as a solid base for expanding research into classification-based segmenters for TPCI rather than as a finished product. There is a wide open opportunity for developers to create customizable software utilizing the MaSCS framework, and there are a lot of extensions and improvements that can be made.

One important extension would be to use a customized classifier that took into consideration the structure of the annotated data during training. The human annotator identifies sets of acceptable masks for each ROI. Ideally, we would like the classifier to be able to exploit features and structure not visible to the annotator to decide which out of these masks is best. The out-of-the-box classifier used in the prototype system here ignores the structure of the training data and optimizes a loss function which penalizes a solution which selects some but not all of the masks chosen by the human annotator. This may impede the classifier's performance on

segmentation.

Creating a customized group-aware classifier requires modifying the training procedure for the classifier. Instead of trying to minimize the mask-wise misclassification rate, we want to minimize the group-wise misclassification rate. For decision tree learning, it is not immediately obvious how to do this. Perhaps a splitting criteria could be developed based on group-wise characteristics, but this is complicated by the fact that groups of annotated masks will be split among different leaves of the tree and so computing the best split at a branch will no longer be a local procedure and will also no longer be independent of decisions on other branches of the tree. It is possible that decision trees will not be feasible for such a group-aware classifier and a different classification algorithm should be incorporated into MaSCS instead.

A second extension the the MaSCS system would be to incorporate additional training data in a more sophisticated manner. Currently, when the system gets additional training data, the classifier is simply retrained with the larger amount of data. This is possibly inefficient, and also does not incorporate the additional information about error patterns that corrected segmentations provide. The corrections provided by an experimenter are likely to indicate the masks on which the segmenter is making the most obvious or egregious errors, and perhaps should be weighted more highly in the retraining. This is similar to the idea behind boosting algorithms, and might allow for more rapid improvement with additional data.

Finally, the work presented here on evaluation of segmentation algorithms is only a beginning. To really understand the performance of algorithms meant to replicate human annotators, we need a better understanding of the behavior and reliability of these annotators. Currently there is no work assessing inter-rater reliability, which will be a crucial step toward determining whether automated algorithms are a good replacement for human annotators in experimental settings.

# 4

# Accounting for motion during *in vivo* imaging

Consider a sequence of two-photon calcium images $f_t(x, y)$ taken at some depth $z$. We would like for the $(x, y, z)$ coordinates of an image pixel to correspond to some fixed spatial coordinates in the brain. That is, we would like to be able to assume that a particular pixel in our image sequence measures from the same location in the brain at each time. In *in-vivo* imaging, this assumption is unlikely to hold due to motion of the animal's brain during imaging. If we consider $(x, y, z)$ to be coordinates in microscope or image space, the brain location they measure from at time $t$ will be a function of the motion. To give an accurate description of the image data in the presence of motion, we would need to estimate a function $g_t(x, y, z)$ which maps a point $(x, y, z)$ in image space to the location $(x', y', z')$ that it measures from in the brain. Unfortunately, estimating this three dimensional motion trajectory given only a sequence of two-dimensional images is very difficult.

The most common simplifying assumption used to make the motion correction problem tractable is that the out-of-plane $z$ component of motion is negligible and can be ignored. With the additional assumption that the motion is small enough or the scanning is fast enough that no motion occurs during the scanning time for a single image frame, the problem of motion correction becomes one of image alignment. There is a large literature on image alignment generally, but very little published

about its application to TPC imaging despite its pervasive use in the field. After reviewing the existing literature on TPCI motion correction (section 4.1) and giving a concrete description of motion artifacts in real data (section 4.2), I evaluate the effects of applying rigid body image alignment techniques to TPCI data (section 4.3).

As mentioned above, using rigid body alignment for motion correction requires making the assumption that there is no motion outside of the imaging plane. In *in-vivo* imaging, this assumption does not hold. Though out-of-plane motion generally seems to be quite small, I show in section 4.4 that its impacts on various data summaries can be quite dramatic. Most importantly, out-of-plane motion can dramatically corrupt the correlation structure between cells, making it impossibly to separate functional clusters of cells from artifactual groupings (explored farther in chapter 5). I introduce a few techniques for filtering data to remove out-of-plane motion artifacts, but find in chapter 5 that none of these allow for cellular clustering based on activity rather than motion artifacts. In section 4.5 I propose some next steps toward creating a unified approach to modeling motion that might allow for such clustering.

## 4.1 Existing work on motion correction

Brain motion during *in-vivo* imaging can be greatly reduced by sophisticated engineering and surgical techniques, but it cannot be completely eliminated. Even motion of sub-pixel magnitude can create motion artifacts that are visible to the naked eye. Most experimental TPCI papers mention some form of image alignment to correct for this. As stated above, image alignment approaches ignore out-of-plane motion, and most assume that motion does not occur during the collection of a single frame. Despite the violation of both of these assumptions, rigid body image alignment can often remove much of the visible motion distortion.

To my knowledge, no literature exists that specifically evaluates commonly used

rigid body image alignment techniques in the context of TPCI. Many experimental TPCI papers use the existing ImageJ alignment macro TurboReg (Hira et al., 2013; Feldt Muldoon et al., 2013; Reeves et al., 2011). Several papers (Drew et al., 2011; Bonin et al., 2011) use computationally efficient cross-correlation methods developed in the engineering and optics communities (Takita et al., 2003; Guizar-Sicairos et al., 2008). The closest to a custom-designed alignment package is probably a toolbox developed for generic intravital microscopy, the Intravital Microscopy Toolbox (Soulet et al., 2013). This ImageJ plugin is a compilation of existing image alignment packages with an additional mechanism for intelligent selection of reference images and removal of individual images that are highly corrupted by motion. Though Soulet et al. (2013) have evaluated their tool on some data, it has not been applied to TPCI data.

In section 4.3 I discuss various rigid body image alignment techniques specifically in the context of TPC imaging and evaluate them on realistically simulated data. I then extend this analysis real TPCI data.

Two-photon laser scanning microscopes collect the pixels of an image sequentially, which means that the assumption of no motion during a single image is flawed. When the magnitude of motion is large enough relative to experimental parameters such as field of view and frame rate, the distortion within individual frames introduced by motion becomes significant. This problem has been partially addressed in the literature on TPCI in awake behaving animals. This literature does not entirely abandon the idea of image alignment, but performs it on a line-by-line rather than an image-by-image basis. This relaxes the assumption of no motion during an image to one of no motion during a line.

Dombeck et al. (2007) and later Chen et al. (2010) propose hidden Markov model frameworks for estimating the in-plane shifts of individual lines. These models consider both the quality of alignment to a template and the likelihood of transition from one offset to another in sequential lines (a smoothness constraint). Greenberg and

Kerr (2009) propose a separate framework that involves aligning lines or sets of lines to a template through an iterative Newton-Raphson procedure (Lucas and Kanade, 1981). These line-by-line approaches are helpful in allowing image registration when there is significant motion during the collection of a single image. Since that is not the case for the data used in this work, I will not discuss line-by-line approaches further.

Despite the relaxed assumptions of the line-by-line approaches, they still do not address the issue of artifacts due to out-of-plane motion. Even in awake behaving animals, the magnitude of out-of-plane brain motion is typically quite small (Greenberg and Kerr, 2009; Dombeck et al., 2007). Nevertheless, even sub-pixel out-of-plane motion can have a dramatic impact on the fluorescence traces over time (see section 4.3.5). Whether these effects are detrimental to downstream inference is largely unknown. The only paper I am aware of that attempts to quantify the impact of out-of-plane motion artifacts on inference is Dombeck et al. (2007). The authors reason that out-of-plane motion will cause equal numbers of positive-going and negative-going transients in the fluorescence trace, and then use the observed number of negative-going transients to estimate the number of positive-going transients that are plausibly due to motion. They estimate that the percentage of positive-going transients due to motion decreases as the amplitude threshold for transient detection increases, and choose a threshold to limit this percentage to $< 5\%$.

The above work addresses out-of-plane motion by arguing that it does not severely impact downstream inference. This sort of argument is highly dependent on the goals and implementation of said downstream processing. In recent papers (reviewed below), some authors have attempted to remove out-of-plane motion artifacts from the raw data rather than simply assessing their impact on inference. To my knowledge, all of these approaches involve filtering the data in some way to remove distortions due to motion (rather than, for instance, attempting to model the motion trajectory directly). Such filtering is in fact also highly influenced by the choice of downstream

analysis, since this choice will impact the metric by which the filtering technique is chosen and evaluated. I discuss this point in greater depth in section 4.4. Here I briefly summarize the existing out-of-plane motion correction (filtering) approaches.

Following in-plane image alignment using phase-only correlation methods (4.3), Bonin et al. (2011) attempt to capture the remaining motion artifacts using principle components analysis (PCA). They manually identify PCA components that appear (based on unspecified spatial and temporal features) to be dominated by motion artifacts. They remove these components from the data before further processing.

Malik et al. (2011) take a more unified modeling approach by representing fluorescence traces from regions of interest as signal (a harmonic regression model, chosen because of the particular stimulus used in the experiment) plus correlated noise (an auto-regressive model). The AR model can capture much of the periodic impact of brain motion on fluorescence traces, but the signal plus correlated noise modeling approach relies on being able to formulate a model for the signal. This will not be possible in some cases, such as during resting state recordings or when the response properties of neurons are unknown. In section 4.4 I explore a variation of this approach using AR filtering outside of the context of a signal model.

## 4.2 Motion artifacts in a representative experiment

Whenever imaging is done in living animals, some amount of motion is unavoidable. In awake behaving animals, volitional movement can cause significant brain motion even despite the use of head stabilization devices. In anesthetized animals, volitional movement is prevented, but the physiological processes that keep the animal alive, such as respiration and the beating of the heart, still cause small but detectible motion of the brain tissue. This motion is exacerbated by the craniotomy required for TPCI, which reduces the pressure on the underlying cortical tissue. Surgical

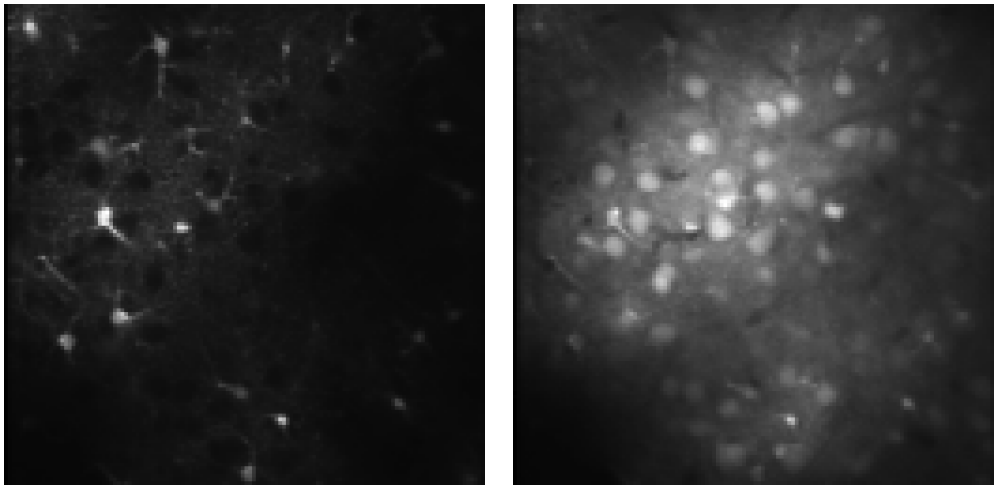
techniques (such as leaving a very thin layer of the skull intact or replacing the skull with a glass coverslip) and good experimental design (such as timing imaging based on physiological triggers) can reduce but not eliminate motion artifacts (Paukert and Bergles, 2012).

In this section I describe the manifestation of these motion artifacts in the data used in this work. I emphasize that artifacts from motion appear in nearly all standard data summaries. Specifically, I describe the impact of motion on pixel-wise variances, spectral characteristics of fluorescence traces, and the correlation structure of pixels and cells.

When examining motion artifacts, we want to separate features of the data that are due to motion from those that are due to neural activity. The structural dye used in these (and many other) TPC imaging experiments provides a very convenient tool for performing this separation. The structural dye, SR101, does not change its fluorescence properties based on neural activity. Ignoring long-term effects such as photobleaching, any fluorescence changes in the structural channel are the result of physical changes to the position or layout of the brain tissue. These changes are primarily physiologically-driven motion. To a first approximation, we can declare that any temporal changes in the structural channel are due to motion, and judge the success of motion correction techniques by the temporal stability of the corrected version of this channel.

It is worth emphasizing that this is an approximation. There are some structural changes to the brain that we do not wish to treat as artifacts. For instance, neural activity results in increased blood flow to a region which in turn causes changes to the diameters of local blood vessels. These changes are important to the study of neurovascular coupling. Nevertheless, the structural channel still provides a means to largely separate motion artifacts from neural activity.

Figures 4.1 through 4.5 explore the effects of motion in an example TPCI ex-



(a) Structural channel (b) Functional channel

**Figure 4.1:** *Time-averaged images showing the spatial layout of the cortical region recorded in this experiment.*

periment according to the three criteria I mentioned above. I show data from both channels, though focus on the structural channel for the reason just mentioned. In this example experiment, there was no stimulus given to the animal. This removes one source of variation in the data, helping to make the motion artifacts clear. However, similar artifacts appear in all experiments.

Figure 4.1 shows the time-averaged mean images for both channels for the example experiment. Since these are time averaged mean intensities, motion artifacts are not visible in these images, but they are useful as a reference for interpreting other presentations of the data. We can see that there are several very clear astrocytes (bright regions in the structural channel, also bright in the functional channel) as well as a number of neurons (bright regions in the functional channel that are dark in the structural channel).

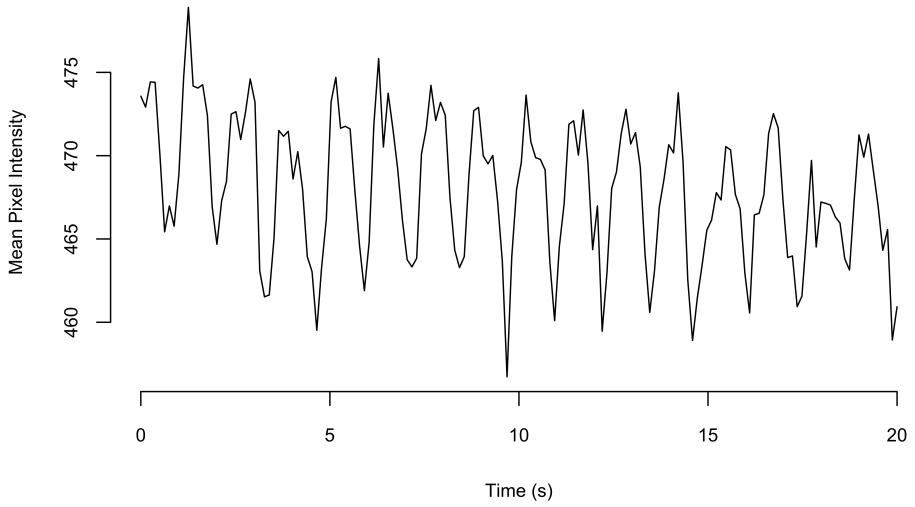
The easiest way to observe motion artifacts is to simply watch a video of the data; motion (especially in-plane motion) is easily detectable by the human eye. However, motion artifacts also manifest themselves in all of the common static summaries of

the data.

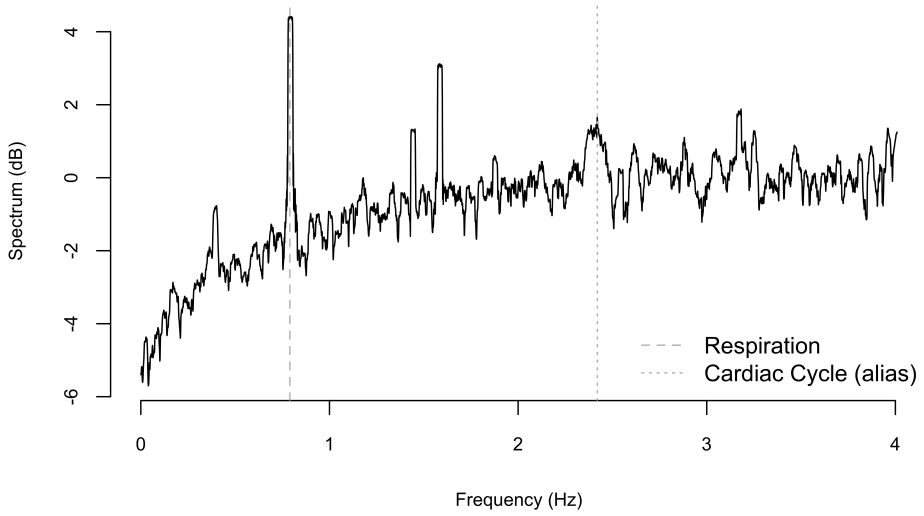
One of the most obvious manifestations of motion in data summaries is its impact on the fluorescence time series. Figure 4.2 demonstrates this in the spatially averaged fluorescence time series from the structural channel. Without motion, the structural channel should be invariant over time. However, we can see that the fluorescence time series is in fact strongly periodic rather than constant. Figure 4.2(b) shows the multi-taper (Thomson, 1982) estimate of the spectrum of the fluorescence time series. There is a clear peak at around 0.8Hz, the frequency of the rat's mechanically controlled respiration. The rat's (uncontrolled) heart rate was approximately 5.6Hz, though it varied over the course of the experiment. This is also apparent in the spectrum as a wide but short peak at 2.4Hz (aliased due to sampling, and wide because of the varying rate). If motion were strictly in-plane, we would not expect to see large changes in mean intensity since in-plane motion would primarily impact the mean through changes to the image boundary. In contrast, out of plane motion could strongly affect the mean as many highly fluorescent cells moved in and out of the imaging plane. In section 4.3.5 I will confirm that this motion-driven variation in mean fluorescence remains after image alignment that corrects for in-plane motion.

To confirm that the peaks in the fluorescence spectrum are due to heart rate and respiration I examined the physiological data that was collected during the imaging experiment. Figure 4.3 shows a portion of the blood pressure recording as well as its spectrum. The peaks at 0.8Hz (respiration) and 5.6Hz (heart rate) are present in this data as well. I verified heart rate by examining the ECG data as well. The settings of the respirator were not recorded.

A second data summary that shows motion artifacts is the pixel-wise variance of fluorescence over time. We expect the variance of a pixel's value over time to be proportional to its mean. We therefore expect pixels recording from cells to have high variance. We see this in the data, but we also see that pixels on the edges of cells

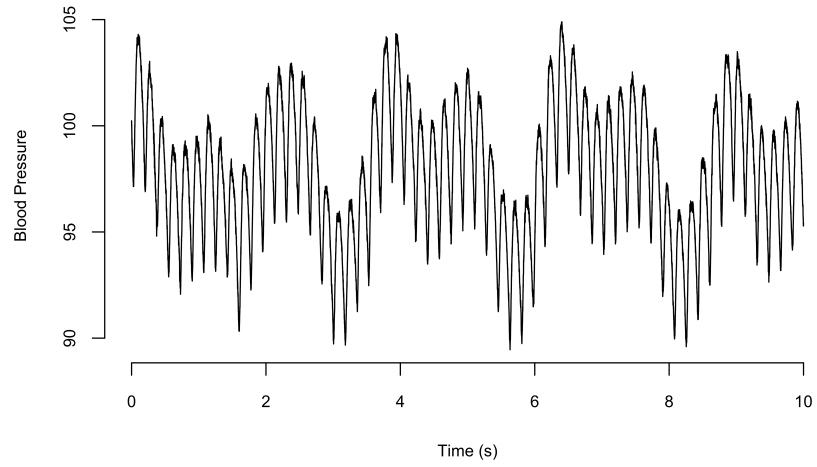



(a) Spatially averaged intensity

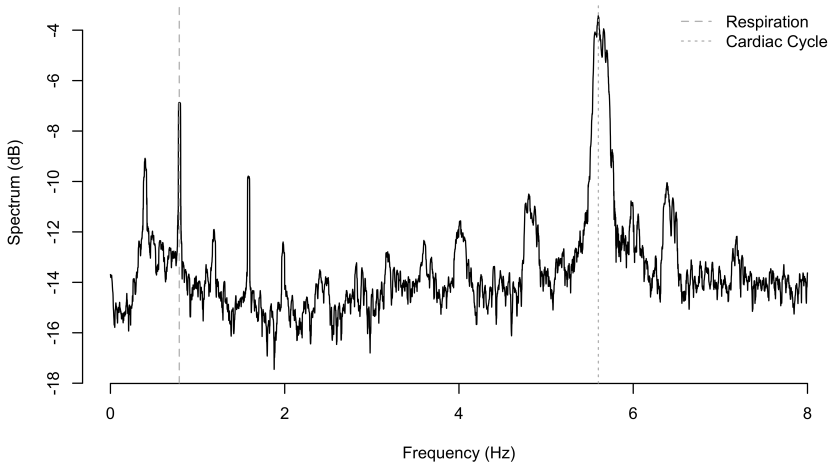



(b) Spectrum of mean intensity time series

**Figure 4.2:** *Mean fluorescence time series and spectral view of motion artifacts. (a) shows a 20 second segment of the spatially averaged intensity in the structural channel. Without motion artifacts, we would expect this to be constant. (b) shows the multi-taper spectrum estimated from the entire mean intensity time series. The mechanically controlled respiration rate appears at 0.8Hz (and harmonics). Less obvious in this data is the aliased heart rate (5.6 Hz aliased to 2.4Hz due to sampling).*

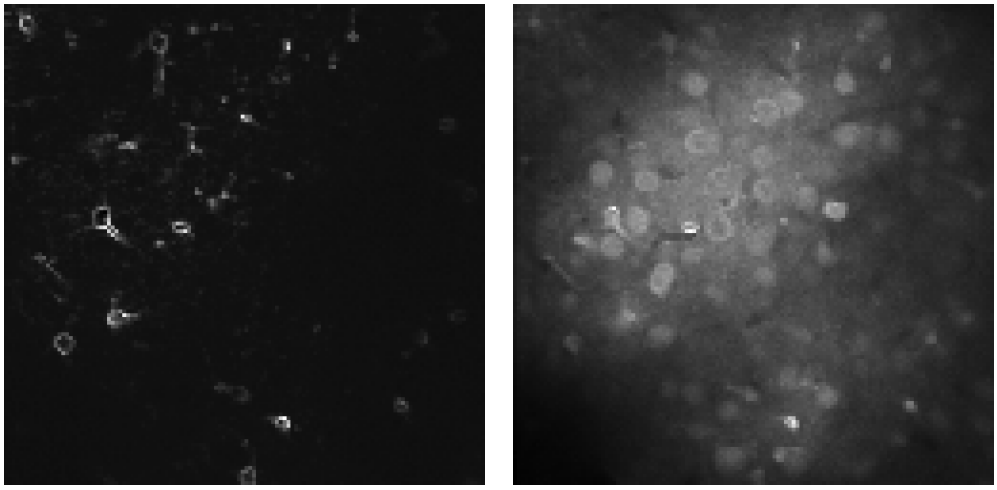



(a) Blood pressure




(b) Spectrum of blood pressure

**Figure 4.3:** *Blood pressure measurements taken during the experiment. (a) A 10 second portion of the blood pressure record. (b) The spectrum of the blood pressure measurements, shown up to 8Hz. Heart rate and respiration appear in this data as they do in the imaging data in figure 4.2.*



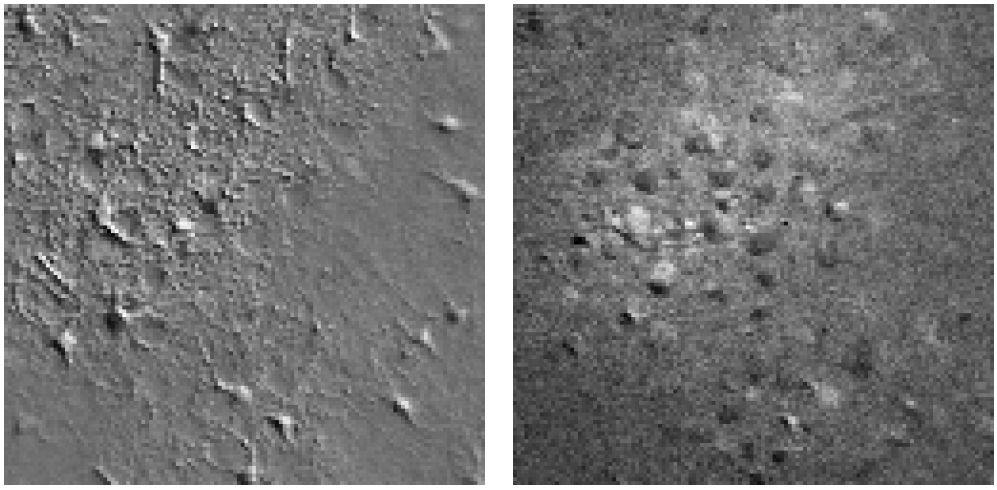
(a) Structural channel: pixel-wise variance divided by mean

(b) Functional channel: pixel-wise variance

**Figure 4.4:** *Motion artifacts in variance structure. These images demonstrate increased variance of intensity over time in pixels on the boundaries of cells. (a) Data from the structural channel. The variance scales strongly with pixel mean, so for clarity the image here shows the variance divided by the mean for each pixel. (b) Data from the functional channel. These variances scale less strongly with mean, so the greyscale is most clear when they are shown directly. Both images demonstrate relatively high variance in pixels on the boundaries of cells, as would be expected if motion caused the cell to move in and out of the volume measured by that pixel.*

have higher variance than those in the brighter center regions. This excess variance is a result of motion causing the cell boundaries to move relative to the image pixels. Figure 4.4 shows this variance effect for both the structural and functional channels.

Finally, motion is also apparent in the correlation structure of the data. Artifacts in correlation structure can be especially important if later analysis uses correlation to define functional clusters of cells where the clustering is intended to reflect similarities in neural activity. Figure 4.5 shows a summary of the pairwise correlations between image pixels in both channels. We can see that there are two clusters of pixels that are highly positively correlated within each cluster and highly negatively correlated between clusters. Both in- and out-of-plane motion contribute to this correlations



(a) Mean pixel-pixel correlation (structural) (b) Mean pixel-pixel correlation (functional)

**Figure 4.5:** *Motion artifacts in correlation structure. These images show one summary of the pairwise correlations between pixels in the data. The intensity of a pixel in the above images is proportional to the mean of the pairwise correlations of that pixel's time series with those of all other pixels. These images show that there are two groups of pixels (very dark pixels and very light pixels) that are highly positively correlated within each group but highly negatively correlated between groups. The in-plane component of motion is clearly visible here as the spatial relationship between these two groups of pixels: dark regions occur to the lower left of bright regions. Motion of cells on that axis causes one set of pixels to dim (the cell moves out of that volume) while another set of pixels brightens (the cell moves in). Similar images result from looking at the pairwise correlation of a single pixel in one of the groups with all other pixels (without averaging).*

structure.

To summarize, even small amounts of motion (in this experiment, I estimated the in-plane motion to be sub-pixel in magnitude, and assume a similar magnitude for the $z$ component) can cause dramatic artifacts in the data. This must be acknowledged, and perhaps corrected or counteracted, in later analysis. In the following sections I propose and evaluate techniques for such correction.

## 4.3 Rigid body image alignment

The most ubiquitous procedure used for addressing motion artifacts in TPCI data is rigid body image alignment. Rigid body alignment attempts to alter each image in an image sequence to align as closely as possible with a reference image using only rigid body shifts (vertical and horizontal translation, plus rotation). Using this procedure to correct TPCI data for motion requires 2 simplifying assumptions.

1. Brain motion is rigid body and only occurs within the imaging plane

2. No motion occurs during the collection of a single image frame

Neither of these assumptions will hold completely for real data. The degree of violation will depend strongly on experimental set-up. Here I explore the application of rigid body alignment techniques to data from anesthetized animals where the magnitude of brain motion is small. This approximately satisfies the second assumption above. I will consider violations of the first assumption in section 4.4.

Though the techniques of rigid body image alignment are not original to this dissertation, there is currently no published work examining their performance in this context. In this section I first review some theory relating to rigid body image alignment. I then describe a simulation study using realistic data, and finally discuss the results of applying these techniques to real data.

The goal of rigid body image alignment is to transform an input image using only rigid-body translation and rotation such that the transformed image matches a target image as closely as possible according to some metric. The process of aligning a pair of images therefore requires two distinct steps:

1. Estimate the alignment parameters

2. Shift one image according to the parameter estimates

There is a great deal of literature on both of these steps. I will present here the details of several approaches, but not all. For more in-depth reviews of image alignment in general, see (for instance) Brown (1992) or Zitova (2003). For applications in other biomedical imaging domains, see part IV of Bankman (2009).

### 4.3.1 Estimating shift parameters

The first task in aligning two images is to estimate the motion parameters that describe the transformation that will cause one image to best match the other. Alignment techniques are differentiated by the method they use to search the parameter space and the metric they use to define alignment quality. This metric can be based on landmarks or features defined either by hand or automatically. In situations where appropriate landmarks are not available or obvious, full-frame metrics can be used instead. These full-frame metrics are what I discuss here.

Consider images $I(x, y)$ and $J(x, y)$, and rigid body shifts represented by the operator $T_{(\mathbf{m},\theta)}$ where $\mathbf{m}$ is a length-two vector giving translations in $x$ and $y$ and $\theta$ gives a rotation angle in radians. Assume for the moment that we can perform this shift. If $I(x, y)$ is the reference image, we wish to compare it to the shifted version of the image to be aligned, $J^{T_{(\mathbf{m},\theta)}}(x, y)$. One very common metric used for this comparison is the cross-correlation (CC) between the two images:

$$CC(\mathbf{m}, \theta) = \sum_x \sum_y I(x, y) J^{T_{(\mathbf{m},\theta)}}(x, y). \tag{4.1}$$

The primary appeal of this metric is that if $\theta$ (the rotation parameter) is set to zero, the CC values can be efficiently computed for all whole-pixel $x$ and $y$ translations simultaneously using the Fast Fourier Transform (FFT). The Fourier transform of the cross-correlation matrix $CC(m_x, m_y)$ is equal to the product of the Fourier transform

of $I(x, y)$ with the complex conjugate of the Fourier transform of $J(x, y)$.

$$\mathcal{F}(CC) = \mathcal{F}(I)\mathcal{F}^*(J) \tag{4.2}$$

where $\mathcal{F}$ is the Fourier transform and $^*$ is the complex conjugate.

Assuming that the two images being compared are in fact circularly shifted versions of each other, the cross-correlation obtained by this procedure will have its maximum at the location of the appropriate shift. However, this maximum may be rather flat, which can make it difficult to locate precisely especially in the presence of noise. The phase-only correlation (POC) is a response to this problem.

By the Fourier shift theorem, circularly shifted images will differ in the Fourier domain by a linear phase shift

$$I(u, v) = J(u, v)e^{-2\pi i\left(\frac{u\mathbf{m_x}}{X} + \frac{v\mathbf{m_y}}{Y}\right)}. \tag{4.3}$$

We can isolate this phase difference using the normalized cross-power spectrum.

$$R(u, v) = \frac{IJ^*}{|IJ^*|} \tag{4.4}$$

$$= \frac{II^*e^{2\pi i\left(\frac{u\mathbf{m_x}}{X} + \frac{v\mathbf{m_y}}{Y}\right)}}{|II^*e^{2\pi i\left(\frac{u\mathbf{m_x}}{X} + \frac{v\mathbf{m_y}}{Y}\right)}|} \tag{4.5}$$

$$= \frac{II^*e^{2\pi i\left(\frac{u\mathbf{m_x}}{X} + \frac{v\mathbf{m_y}}{Y}\right)}}{|II^*|} \tag{4.6}$$

$$= e^{2\pi i\left(\frac{u\mathbf{m_x}}{X} + \frac{v\mathbf{m_y}}{Y}\right)} \tag{4.7}$$

Taking the inverse transform of this isolated phase gives a Kronicker delta function with its peak at $(m_x, m_y)$. This peaky function is the phase-only correlation (POC).

If rotations are being considered in addition to shifts, the computational efficiency of the Fourier computation of the CC or POC functions is lost. Instead, some iterative (or exhaustive) search must be performed over the space of possible shift parameters.

Without the computational advantage of the FFT, other metrics may become more appealing for comparison of images. Examples of such metrics are correlation

$$r(\mathbf{m},\theta) = \frac{\sum_x \sum_y \left(I(x,y) - \bar{I}\right)\left(J^{T_{(\mathbf{m},\theta)}} - \overline{J^{T_{(\mathbf{m},\theta)}}}\right)}{\sigma_I \sigma_{J^{T_{(\mathbf{m},\theta)}}}}, \tag{4.8}$$

mean squared error

$$MSE = \frac{1}{XY}\sum_x \sum_y \left(I(x,y) - J^{T_{(\mathbf{m},\theta)}}(x,y)\right)^2, \tag{4.9}$$

or mean absolute error

$$MAE = \frac{1}{XY}\sum_x \sum_y \left|I(x,y) - J^{T_{(\mathbf{m},\theta)}}(x,y)\right|. \tag{4.10}$$

The performance of these metrics will depend on properties of the images being aligned. Section 4.3.4 describes a simulation study to evaluate these metrics on simulated data with the characteristics of TPC images.

### 4.3.2 Fourier interpolation for implementing shifts

The final alignment of images, as well as any iterative search procedure for estimating shift parameters, requires a method to shift an image by a particular translation and rotation. Unless images are simply translated by whole-pixel amounts and/or rotated by increments of 90 degrees, shifting images requires interpolation to maintain the expected pixel grid.

The optimal method for performing this interpolation depends strongly on properties of the continuous process or spatial structure that underlies the discretized image. Strong spatial structure might suggest a particular interpolation method, but in the absence of such structure it is not immediately clear how to make the decision.

Eddy and Young (2009) suggest that one possible criteria is that the interpolation procedure is information-conserving.

An image transformation $T_{(\mathbf{m},\theta)}$ is *information-conserving* if for any sequence of transformations that result in the image being returned to its original location and orientation, the translated image is identical to the original image (Eddy and Young, 2009). Such a sequence could be a set of rotations that sum to multiples of $2\pi$ radians, or a $(\mathbf{m}, \theta)$ translation followed by its inverse. Eddy and Young (2009) demonstrate that Fourier, or trigonometric, interpolation can be information-conserving whereas a number of other commonly used interpolation techniques are not.

The Fourier shift theorem described in equation 4.3 directly allows for arbitrary translations of an image in $x$ and $y$ through phase shifts in the Fourier domain. Eddy et al. (1996) show that arbitrary rotations (as described by a two-by-two rotation matrix) can be factored into the product of three shearing matrices

$$\begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ sin\theta & 1 \end{pmatrix} \begin{pmatrix} 1 & -tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}. \qquad (4.11)$$

Since each shear involves only one-dimensional translations, they too can be implemented by a sequence of phase shifts in Fourier space. In this way, Fourier interpolation methods can fully implement arbitrary rigid body transformations. This is the interpolation method used in this work.

### 4.3.3 Achieving sub-pixel precision

In practice, biological motion is extremely unlikely to result in whole-pixel shifts between images in an experiment. Therefore, any practical image alignment procedure must be able to work at sub-pixel precision.

When performing an iterative optimization procedure over the space of motion

parameters, sub-pixel precision is limited only by the convergence parameters of the optimization algorithm. However, when using the FFT-based cross-correlation or phase-only correlation calculations, sub-pixel precision requires additional work.

The traditional technique to improve the precision of the POC is to up-sample the Fourier domain representation by embedding the matrix of Fourier coefficients in a larger matrix of 0's. Up-sampling by a factor of $k$ before performing the inverse transform allows identification of the location of the maximum of the POC with a precision of $\frac{1}{k}$ pixels. Unfortunately, this approach does not scale well when the up-sampling factor gets large as the Fourier matrices become prohibitively large.

Efficient approaches to sub-pixel registration using the POC is an active area of research (Yu and Wang, 2012; Guizar-Sicairos et al., 2008; Nagashima et al., 2006; Takita et al., 2003; Foroosh et al., 2002). One approach is to use a small up-sampling factor, find the approximate location of the maximum, and then fit some peaky function to that area of the POC matrix to refine the estimate. Takita et al. (2003) derive the functional form of the POC for minute displacements

$$POC(x,y) = \frac{1}{XY} \frac{\sin[\pi(x+m_x)]}{\sin\left[\frac{\pi}{X}(x+m_x)\right]} \frac{\sin[\pi(y+m_y)]}{\sin\left[\frac{\pi}{Y}(y+m_y)\right]} \tag{4.12}$$

which can be fit to the computed POC matrix. Alternatively, any smooth peaky function (such as a parabola or Gaussian) can be used as an approximation.

As shown in the following simulation study, POC techniques do not perform well in the presence of even small amounts of rotation. Iterative optimization algorithms perform better, but can be sensitive to initialization values when the motion parameters are in the vicinity of 0. Using POC techniques to initialize the translation parameters before beginning the optimization can significantly improve performance.

### 4.3.4 Simulation study

To evaluate the performance of the rigid body image alignment procedures described above, I performed a simulation study using artificial data. To improve the relevance of the simulation study, I generated artificial data using real data from the structural channel (the SR101 non-calcium-sensitive dye) as a starting point. This allowed for artificial images to have realistic spatial structures and features.

I first constructed a the time-averaged image

$$G_{(x,y)} = \frac{1}{T}\sum_{t=1}^{T} I_{(x,y,t)}. \tag{4.13}$$

The average image $G_{(x,y)}$ was, of course, much less noisy than any single image on which alignment would be performed. I simulated realistic noisy images $N_{(x,y)}$ from $G_{(x,y)}$ by adding noise meant to approximate the observed difference between the single and averaged images in real data.

I added independent Gaussian noise to each pixel

$$N_{(x,y)} = G_{(x,y)} + \mathcal{N}(0, \sigma) \tag{4.14}$$

where the noise variance $\sigma$ was chosen to be the sample variance of the difference between pixels in the mean image $G$ and those in the individual images from which it was computed. This process typically resulted in some pixels with values outside of the allowable pixel intensity range $[0, 4096]$. Simply thresholding the values resulted in a large number of pixels that were exactly 0, an unrealistic situation. Instead, I replaced any pixel with a value less than 0 with a uniformly generated value between 0 and the mean of $G_{(x,y)}$.

The procedure described above is an approximation of the noise in real data. Noise introduced by the fluorescence microscopy system is likely Poisson. Additional

sources of biological noise (such motion, metabolic and chemical processes) are likely quite complex. However, the procedure described above does result in realistic pixel intensity distributions in the simulated images as shown in figure 4.6.

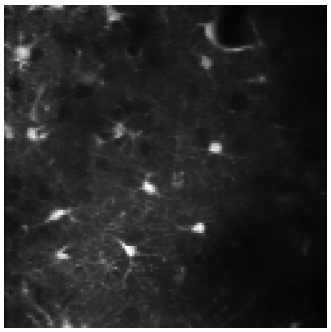
Using the process described above, I created two sets of simulated data. Both sets of data consisted of 100 images. In the *clean condition*, I generated a single noisy image. This single image was then shifted according to each of 100 randomly generated shifts, described below. In the *noisy condition*, I generated 100 different noisy images and then shifted each of these images by one of the randomly generated shifts. Therefore, in the clean condition the images differed only by translation and rotation. In the noisy condition, the images differed by translation, rotation and noise.

The motion parameters used for the simulation were 100 sets of translations and rotations $(\mathbf{m}, \theta)_i$ where the the $x$ and $y$ translations were uniformly randomly generated between $-1.5$ and 1.5 pixels, and the rotations ranged uniformly between $-\frac{\pi}{40}$ and $\frac{\pi}{40}$ radians. I selected these ranges based on exploratory analysis of the real data: motion in these experiments was small, and this simulation study was intended to explore alignment performance at the sub-pixel level.

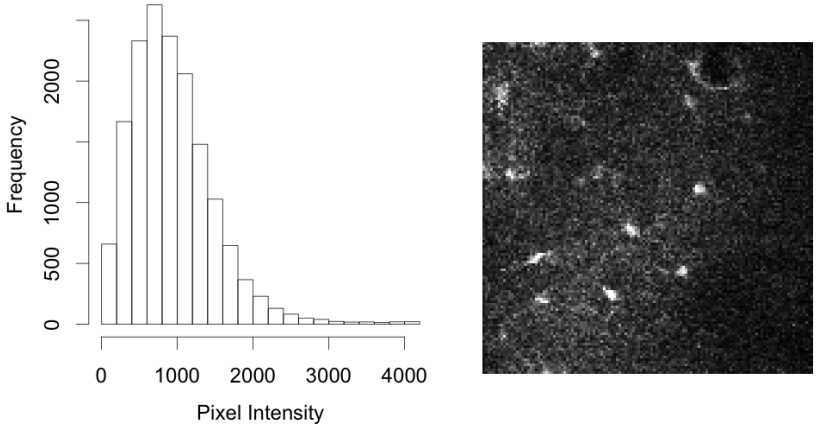
I implemented the shifts in the simulated data using the Fourier interpolation methods described in section 4.3.2. To reduce the edge effects, I first padded each image with zeros and tapered the image edges using a Hanning window

$$w(x) = 0.5\left(1 - cos\left(\frac{2\pi x}{X-1}\right)\right). \tag{4.15}$$

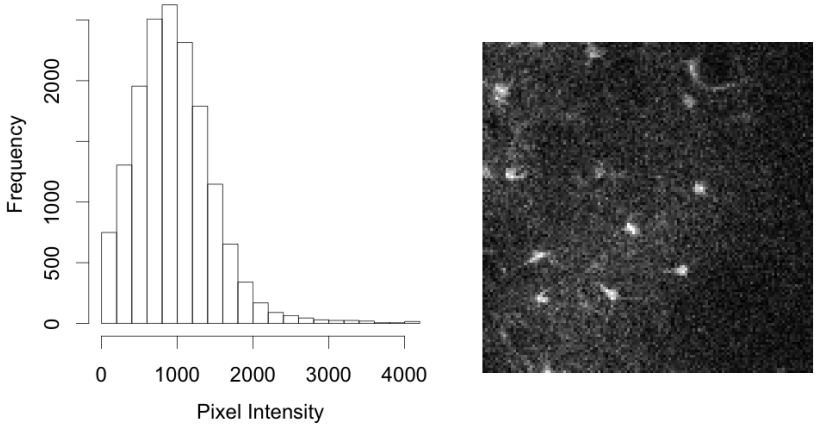
I shifted the padded and tapered image appropriately, and then trimmed the result back to the original image size. This process violates the assumption of circular shift implied by the Fourier methods, but is more realistic and avoids unexpected behavior on edges and corners.



(a) Time averaged image




(b) Real image




(c) Simulated image

**Figure 4.6:** *Comparison of a real and simulated image. (a) shows the time-averaged real data. (b) shows a histogram of pixel values and the corresponding image for a single time point in the real data. (c) shows the same for a simulated noisy image.*

The goal of the simulation study was to estimate the motion parameters as accurately as possible from the simulated data using a variety of rigid body techniques. The techniques I explored were

**Phase-only correlation with function fitting (POC-F)** Up-sample the POC matrix by a factor of 2, then fit the area around the maximum using equation 4.12.

**Phase-only correlation with Gaussian fit (POC-G)** Up-sample the POC matrix by a factor of 2, then fit a Gaussian to the area around the maximum.

**Optimization of MSE (Opt-MSE)** Iterative optimization (Nelder-Mead) of the MSE (equation 4.9), with fixed initialization and with POC-G initialization of the translation parameters.

**Optimization of MAE (Opt-MAE)** Iterative optimization (Nelder-Mead) of the MAE (equation 4.10), with fixed initialization and with POC-G initialization of the translation parameters.

| | Clean Data | | | | Noisy Data | | | |
|---|---|---|---|---|---|---|---|---|
| | Trans | | Rotation | | Trans | | Rotation | |
| | Mean | Med. | Mean | Med. | Mean | Med. | Mean | Med. |
| POC-G + Opt-MSE | 0.38 | 0.22 | 4.38 | 1.71 | 12.55 | 6.40 | 109.57 | 49.34 |
| Opt-MSE | 0.58 | 0.24 | 6.08 | 1.84 | 12.51 | 6.50 | 113.04 | 48.05 |
| POC-G | 1.70 | 1.07 | | | 18.03 | 9.63 | | |
| Opt-MAE | 17.03 | 0.28 | 10.74 | 2.45 | 16.58 | 10.17 | 135.73 | 45.91 |
| POC-G + Opt-MAE | 0.34 | 0.22 | 4.52 | 1.64 | 16.50 | 10.77 | 134.05 | 41.13 |
| POC-F | 9.20 | 6.50 | | | 27.90 | 19.53 | | |

**Table 4.1:** *Results from the simulation experiment comparing image alignment techniques. All numbers are percent error in shift parameters (equation 4.16). Each row gives the results from a particular alignment technique, with results for translation and rotation parameters for both the clean data and the noisy data. Both the mean and the median percentage errors are given, since some methods are susceptible to occasional large errors which can skew the mean. Rows are sorted according to median error on the noisy data (the more realistic case).*

The results of the study are shown in table 4.1. All images were padded with zeros and tapered (as described above for data simulation) before being passed to the alignment algorithms. Each algorithm returned estimated translation (for all methods) and rotation (for the optimization methods) parameters which I then compared against the known ground truth shift parameters used to generate the data. I used percentage error to evaluate performance

$$\% \text{ Error} = \begin{cases} 100 \cdot \frac{\sqrt{(m_x - \hat{m}_x)^2 + (m_y - \hat{m}_y)^2}}{\sqrt{{m_x}^2 + {m_y}^2}} & \text{for translation} \\ 100 \cdot \frac{|\theta - \hat{\theta}|}{\theta} & \text{for rotation} \end{cases} \tag{4.16}$$

All the methods performed reasonably well on the clean data, though there are a couple points to note. The POC methods performed slightly worse than the optimization methods at estimating translation parameters. This is to be expected given that the POC methods are unable to account for rotation. Though the rotations in this simulated data were very small (less than $\frac{\pi}{40}$ radians, which is barely visible to the human eye) they still impaired the performance of the POC methods. I tested the POC methods on corresponding simulated data with the same translations but no rotations, and their performance was comparable to the optimization techniques (POC-G had a mean error in translation of 0.58%). The POC-F results reported here are substantially worse than the POC-G. Anecdotally, the POC-F is very sensitive to the choice of upsampling amount and window size.

Another thing to note from the results on the clean data is that the Opt-MAE method has a very poor performance evaluated by mean translation error. This is due to occasional images where the optimization failed and returned the starting parameters or 0. The mean performance of the Opt-MAE improved greatly when it was initialized with the POC-G translation estimates, since these outlying cases were eliminated. Though the Opt-MSE method did not suffer from this problem as

frequently, initialization with the POC-G estimates improved its performance as well. Finally, the optimization methods were able to estimate the rotation parameter fairly well. Again, the mean performance was sometimes distorted by extreme outliers, but the median performance was between 1.5 and 10% error - which corresponds to errors of between 0.0008 and 0.008 radians.

The clean data case described above demonstrates that these alignment techniques are able to perform well at sub-pixel resolutions for this type of data. However, the noisy data is much more realistic to the application. The performance of the alignment algorithms differs substantially in the noisy data case.

Table 4.1 is sorted according to mean error in the estimation of translation parameters in the noisy data case. The best performing algorithms were Opt-MSE, both with and without initialization with POC-G. Again the POC methods on their own suffered from not being able to account for even small amounts of rotation. The Opt-MAE algorithms were more sensitive to the noise in this simulation than the MSE-based algorithms, and also suffered from more extreme errors. Initialization with POC estimates aided in the estimation of rotation parameters for both optimization algorithms.

Not shown in table 4.1 are the run times for the various alignment algorithms. The POC methods were very fast (100 images were aligned in 50-100 milliseconds), since the only iterative procedure involved is in fitting a function to a small portion of up-sampled POC matrix. The optimization procedures were significantly slower, taking up to 10 seconds per image. Profiling of the code revealed that this poor performance was due primarily to the infrastructure of the general Nelder-Mead optimization package used (R package 'neldermead' available at http://cran.r-project.org/web/packages/neldermead/index.html) rather than to inefficient evaluation of the objective function. A custom-built and streamlined optimization package would almost certainly reduce the time required significantly, though probably not to

the level of the POC methods. Writing that code is beyond the scope of the current work, though the current inefficient implementation is included in the RCI package.

This simulation study suggests that the POC-G+Opt-MSE method is the best choice for aligning TPC imaging data like that considered here. This is the technique used for alignment of real data in this work.

### 4.3.5 Application to data

I used the POC-G+Opt-MSE method described above to register real data. I estimated alignment parameters based on images from the structural channel, and then used these estimates to shift images from both channels. As a reference image, I selected an arbitrary time-point from near the middle of each experiment. This section describes and evaluates the results of this procedure. The data shown here are from the same example experiment used in section 4.2.

Figure 4.7 shows a 10 second segment of the estimated $X$, $Y$, and rotation alignment parameters for the example experiment. Figure 4.8 shows the spectra of the motion parameter estimates. As expected, we see respiration and heart rate in these spectra. As I will demonstrate in this section, some of the artifacts due to motion are reduced or removed by image alignment, but many remain. I will discuss the three views of motion that I introduced in section 4.2: mean fluorescence, variance properties, and correlation structure.

As predicted in section 4.2, the mean fluorescence time series is nearly unaltered by image alignment. We expect slight alterations due to edge effects and interpolation, and indeed the mean fluorescence time series are not identical (see figure 4.9). However, they are very similar, and their spectra are indistinguishable. This is unsurprising.

The effects of image alignment are more apparent on the variance view of motion

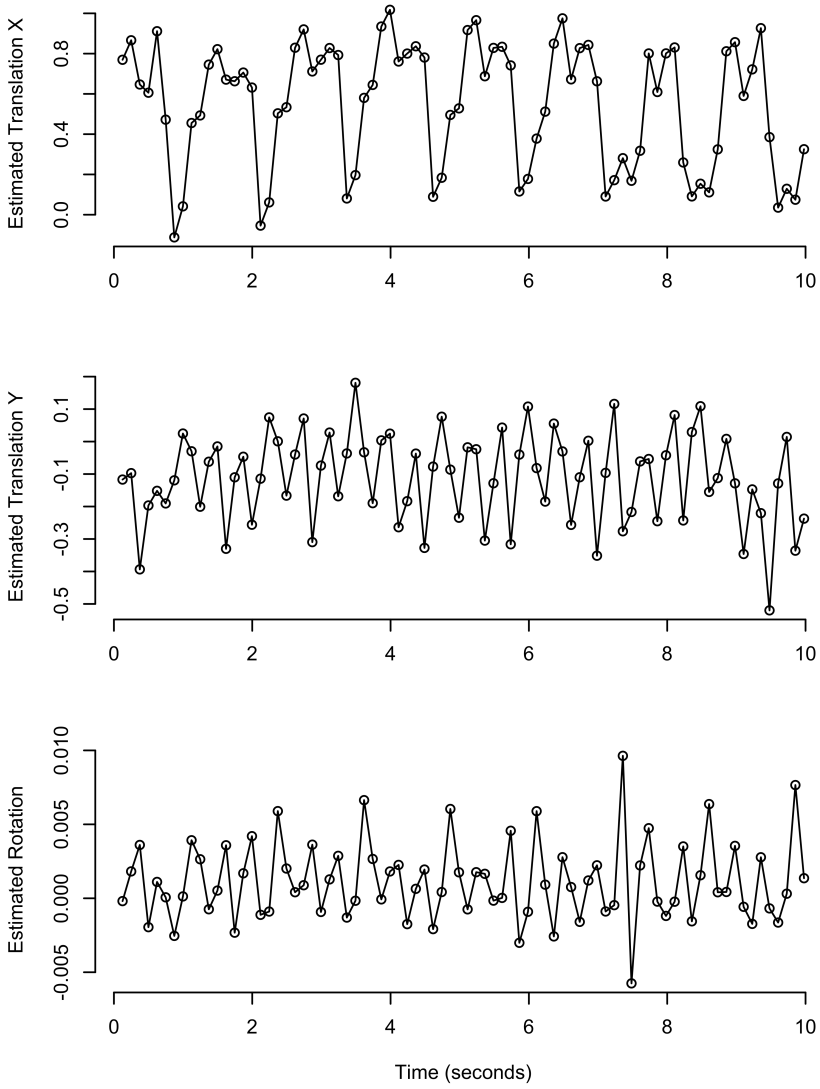



**Figure 4.7:** *Estimated $X$, $Y$, and rotation alignment parameters for a 10 second segment of data. Periodic effects are clearly visible, especially in the $X$ component (which has the highest magnitude).*

artifacts. Recall that in the raw data pixels around the edges of cells had higher variance due to in-plane motion. Figure 4.10 shows the pixel variances after image alignment. The cell edge effects that I attributed to in-plane motion in the raw data are reduced. In addition, as shown in 4.10(c), the overall variance is reduced. Image alignment accounts for at least some of the variation in the data.

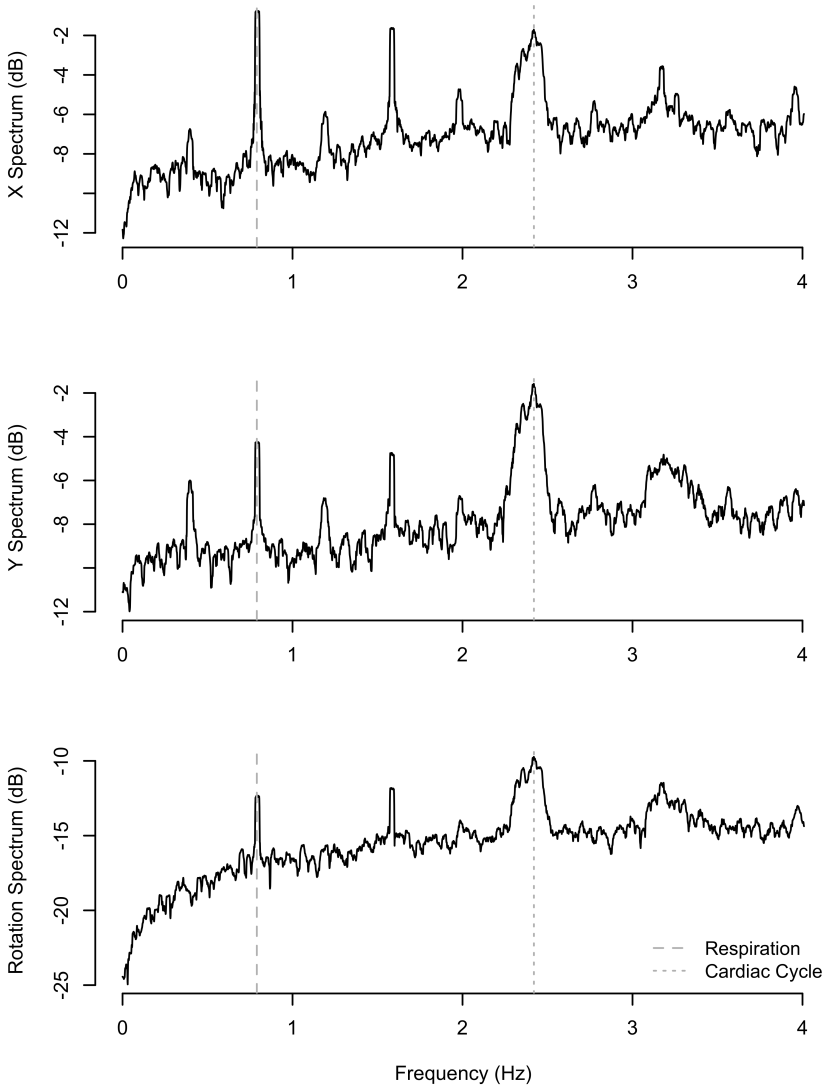



**Figure 4.8:** *Spectrum of estimated alignment parameters. Respiration and heart rate (plus harmonics) are visible.*

The most interesting effect of image alignment appears in the correlation structure. In the raw data, the correlation structure revealed motion through groups of correlated pixels adjacent to each other along the primary axis of in-plane motion. In figure 4.5 this was visible dark regions to the lower left of bright regions (resulting in a three-dimensional effect). After image alignment, motion still induces correlation, but the

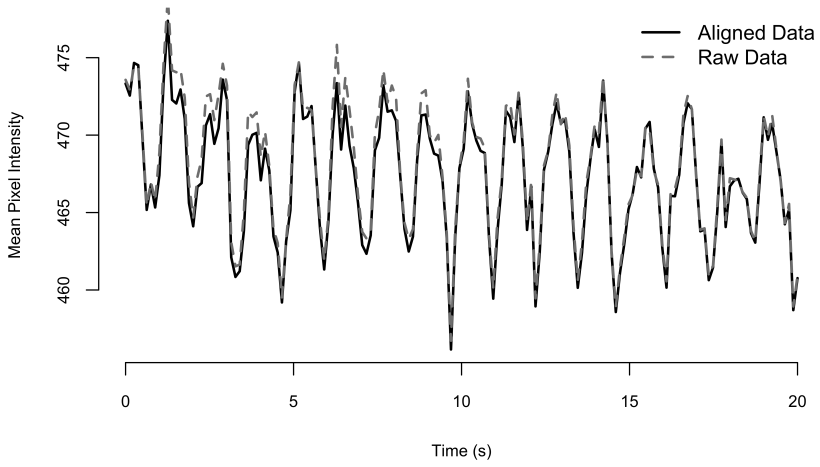
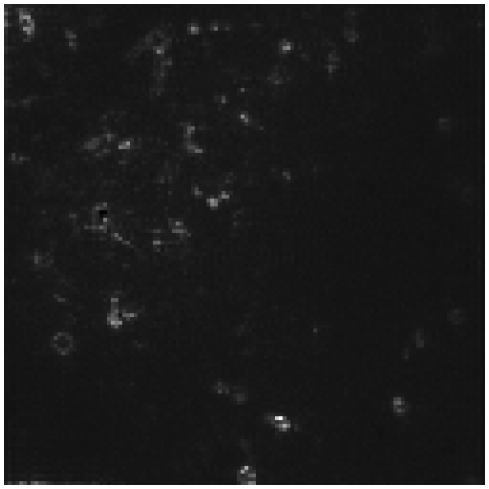



**Figure 4.9:** *Comparison of a 10 second segment of the spatially averaged fluorescence before and after image alignment. The time series are very similar, though not identical. Their spectra are indistinguishable (shown for the raw data in 4.2(b)).*

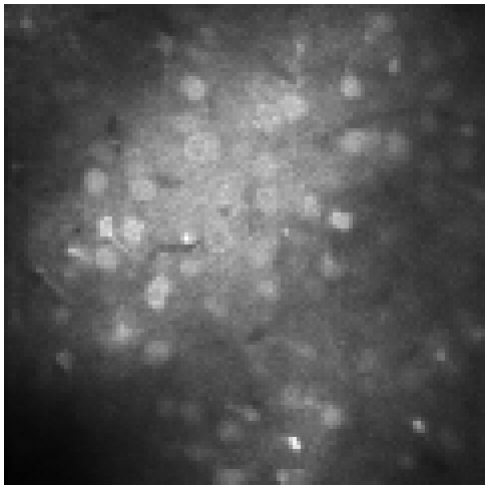
structure of this correlation is more subtle and mimics what we might expect to see if cells were correlated due to activity rather than motion. Figure 4.11 shows the mean pair-wise correlations after image alignment. There are still two highly correlated groups of pixels (bright regions and dark regions) but now these groups indicate entire cells. It would be easy to conclude that the bright cells and dark cells in the correlation images correspond to functional groups. However, this conclusion would be incorrect, as the correlation is actually induced by out-of-plane motion. Section 4.4 demonstrates this, and discusses possible correction techniques.

## 4.4 Out-of-plane intensity correction

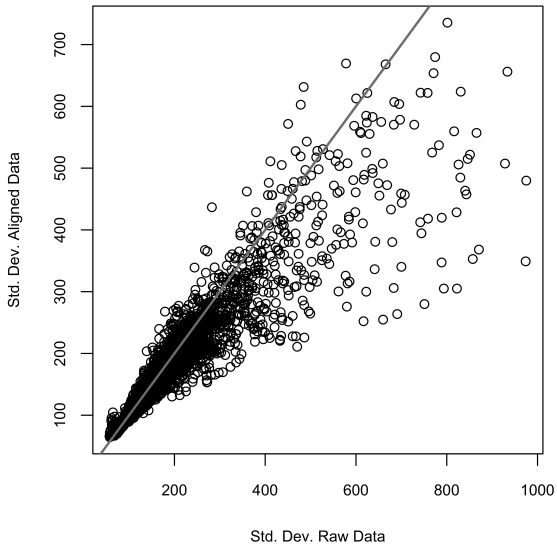
Image alignment is almost always performed in experimental work using TPCI, but out-of-plane motion is rarely considered. Nevertheless, as demonstrated in the previous section, motion artifacts in the spectrum, variance and correlation structure of the data remain after image alignment. The most complete way of addressing this



(a) Structural channel: pixel-wise variance divided by mean

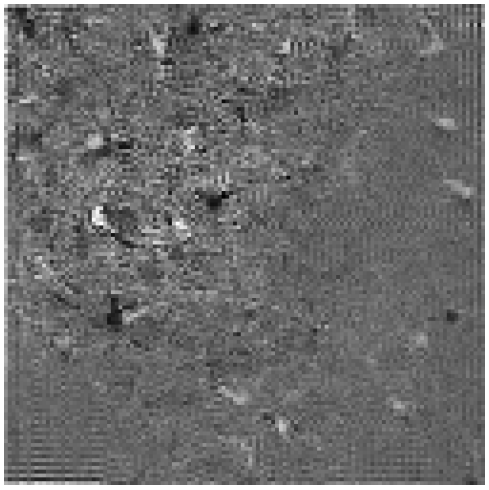


(b) Functional channel: pixel-wise variance

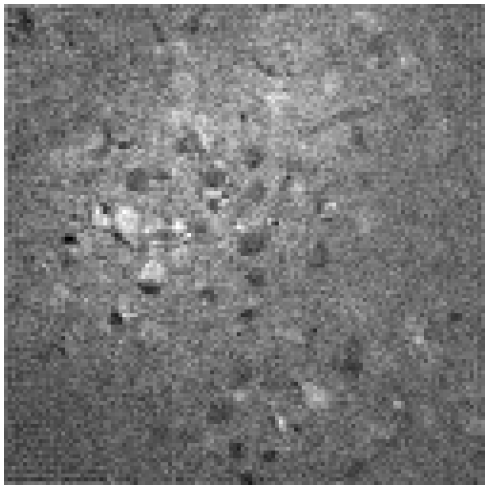



(c) Structural channel standard deviation before and after alignment

**Figure 4.10:** *Variance structure after image alignment. (a) Ratio of variance to mean for the structural channel after image alignment (compare to figure 4.4(a)). (b) Pixel variances of the functional channel after image alignment (compare to figure 4.4(b)). (c) Comparison of pixel standard deviations before and after image alignment for the structural channel. Variance is reduced overall by image alignment, and the boundary effect is reduced but not eliminated.*



(a) Structural channel: mean pairwise correlation



(b) Functional channel: mean pairwise correlation

**Figure 4.11:** *Correlation structure after image alignment. Two groups of correlated pixels are visible (dark regions and bright regions). These groups now correspond to two groups of cells, unlike before image alignment when correlation structured showed the primary axis of in-plane motion (causing a 3D effect in the images). The correlation structure after alignment, as shown here, could easily be mistakenly be interpreted as functional cell groupings. However, the fact that these groups are driven primarily by out-of-plane motion, not cellular function.*

out-of-plane motion would be to try to estimate and model it, incorporating the estimated motion trajectory into any further modeling. This is beyond the scope of this work, though I discuss some ideas for how to tackle this problem in section 4.5. A simpler approach to addressing out-of-plane motion is to attempt to remove the artifacts introduced by the motion from the data by filtering. To differentiate it from more comprehensive motion modeling, I refer to this filtering approach as *intensity correction*. I explore intensity correction in this section.

The definition of success of the intensity correction approach depends very strongly on the down-stream processing and inference, since this will determine which functions of the data need to be free from artifacts. We have seen that out-of-plane motion has a strong impact on the temporal spectrum of the data. It is possible to use frequency-

based notch filtering to remove some of this, but the physiological drivers of motion have a complex impact on the spectrum (heart rate and respiration, plus aliasing and harmonics). An experimenter interested in the spectrum of cellular activity would probably be best served by simply ignoring the spectral peaks accounted for by motion and focusing on the remaining patterns in the spectrum. Any signal at the same frequency as a physiological process would be incredibly difficult to separate from artifacts regardless of filtering.

A slightly more general variant on notch filtering is to perform some whitening of the fluorescence trace which doesn't require the experimenter to specify frequencies to remove. The motivation of this approach is to attempt to remove all the periodic components of the trace, assuming that the scientifically interesting cellular activity will not be periodic over the whole course of the experiment. One way of whitening a fluorescence trace (an approach motivated by Malik et al. (2011)) is to fit a high-order auto-regressive (AR) model to the trace, retaining the residuals as the filtered data.

The $AR(p)$ model represents a trace $X_t$ as

$$X_t = c + \sum_{i=1}^{p} \rho_i X_{t-i} + \epsilon \tag{4.17}$$

where the $\rho_i$ are the parameters of the model specifying the dependence of an element of the time series $X$ on each of the previous $p$ elements. If the order of the model $p$ is sufficiently high (in this work I used $p = 25$), the AR model will fit most persistent periodic components of the data.

I applied $AR(25)$ filtering to the aligned TPCI data pixel-wise. For each pixel, I took the first difference to remove long-term trends, and then fit an $AR(25)$ model to that pixel's time series. I retained the residuals from the $AR$ model. Applied at the pixel level, this AR filtering effectively flattens the spectrum of individual pixels, as expected. It is worth noting that when the traces from collections of pixels are

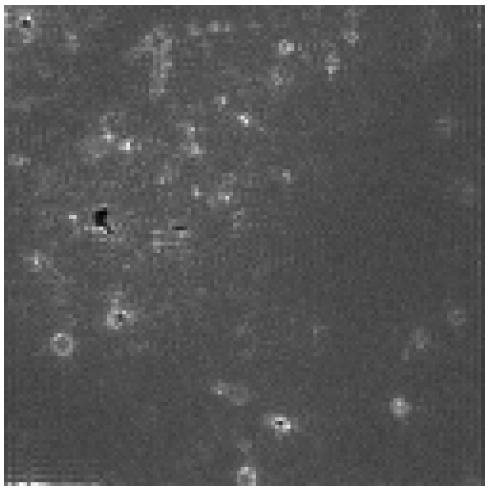
averaged after filtering, the physiologically driven frequency components reappear - though the AR filtering is effective, it is not perfect and the remaining power at these frequency components is clarified by the averaging. It is also possible to apply AR filtering to the mean traces from cell masks instead of on a pixel basis. This provides a flatter spectrum for each cell, but I did not find that the choice of pixel or ROI level filtering impacted later analysis (cell clustering).

Figures 4.12 and 4.13 show the familiar summaries of variance and correlation structure after pixel-wise AR filtering. The filtering reduces the magnitude of the variance and the correlation in the structural channel, as we would like. However, the spatial patterns in variance and correlation are simply damped, not eliminated. As shown in chapter 5, these spatial patterns, despite the reduced magnitude, still corrupt correlation-based clustering.

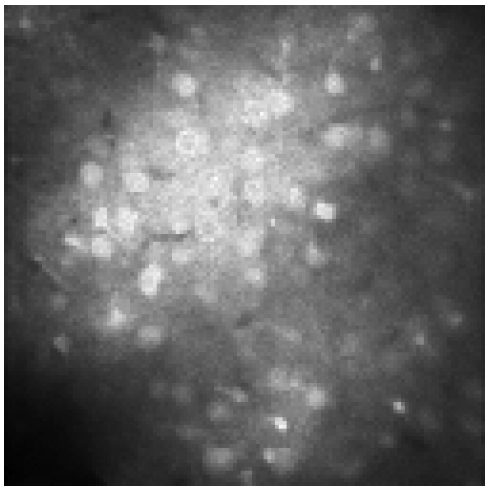
A second filtering approach which removes periodic components but is more flexible than notch filtering is regression on the estimated in-plane motion components. If the path of brain motion is approximately elliptical, it is reasonable to assume that the in-plane motion parameters will predict the magnitude of out of plane motion. If we also assume that out-of-plane motion has a linear effect on intensity of a pixel, we can use basic linear regression on the estimated in-plane motion parameters to filter the pixel time-courses. As in the AR filtering, we retain the residuals after fitting the model. Though it is unlikely that the assumption of linearity holds entirely, the regression approach does reduce the variance and correlation similarly to the AR approach.

Figures 4.14 and 4.15 show the variance and correlation images for the example experiment after regression filtering.

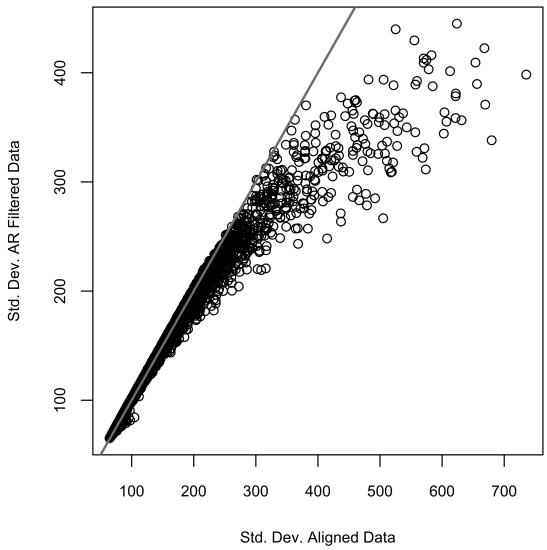
The figures presented here show the effect of these intensity correction approaches on one example experiment. The next chapter argues that the remaining motion artifacts corrupt correlation-based cell clustering. A more comprehensive methodology



(a) Structural channel: pixel-wise variance divided by mean

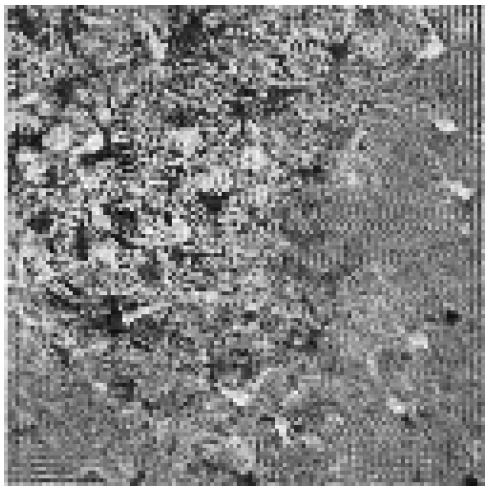


(b) Functional channel: pixel-wise variance

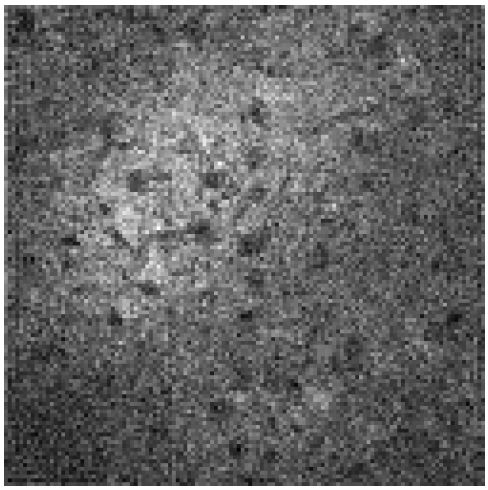



(c) Structural channel standard deviation before and after AR filtering

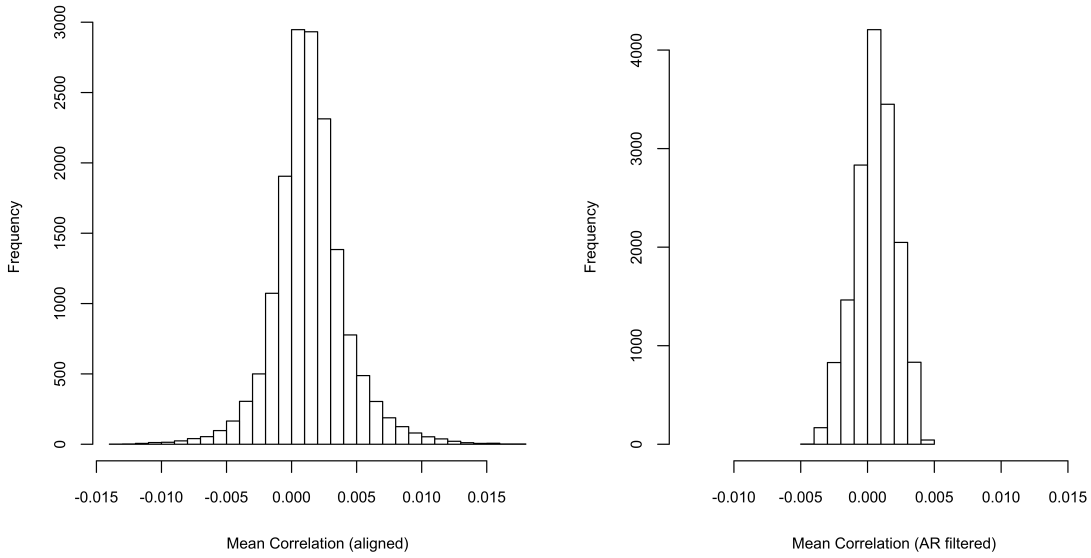
**Figure 4.12:** *Motion artifacts in variance structure after AR filtering. Though the overall variance is reduced in the structural channel (as we want), the spatial patterns remain.*



(a) Mean pixel-pixel correlation (structural)

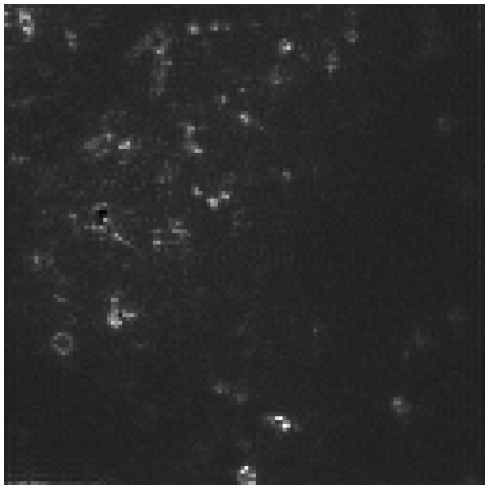


(b) Mean pixel-pixel correlation (functional)

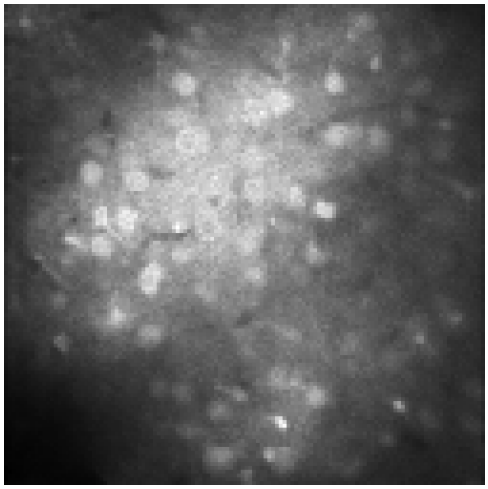



(c) Mean pixel-pixel correlation (functional)

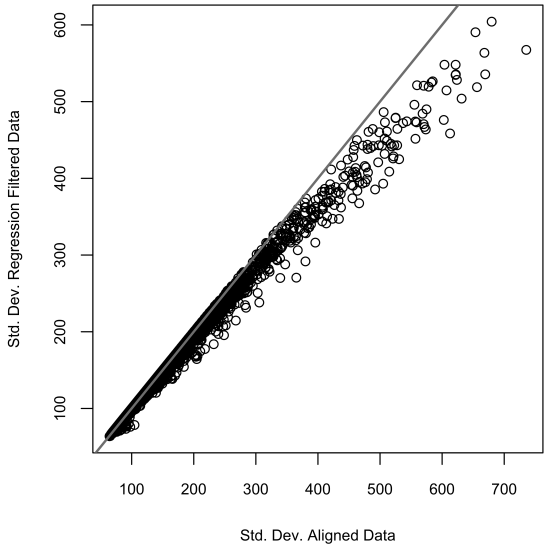
**Figure 4.13:** *Motion artifacts in correlation structure after AR filtering. The spatial patterns in correlation structure are less pronounced, and the average pixel-pixel correlation magnitude is reduced. However, some spatial structure remains, and correlation-based cell clustering is still corrupted by these artifacts.*



(a) Structural channel: pixel-wise variance divided by mean

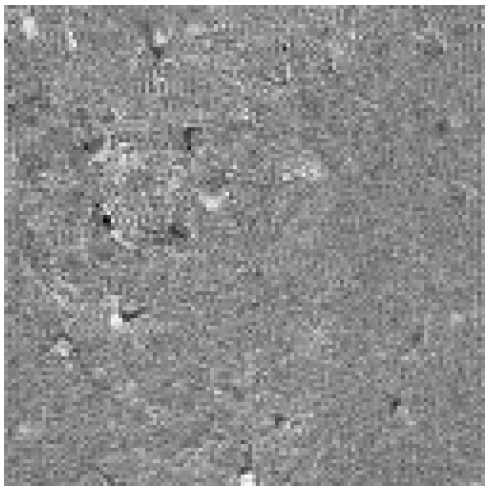


(b) Functional channel: pixel-wise variance

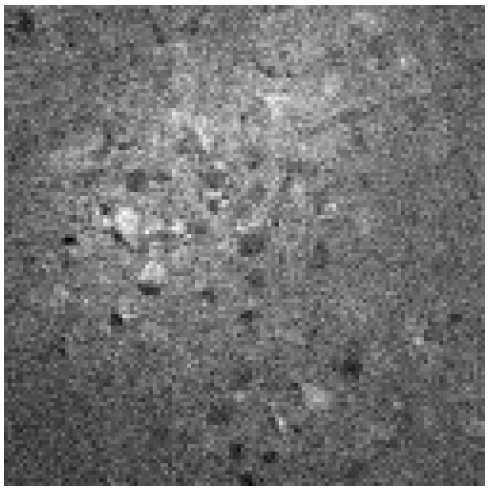



(c) Structural channel standard deviation before and after AR filtering

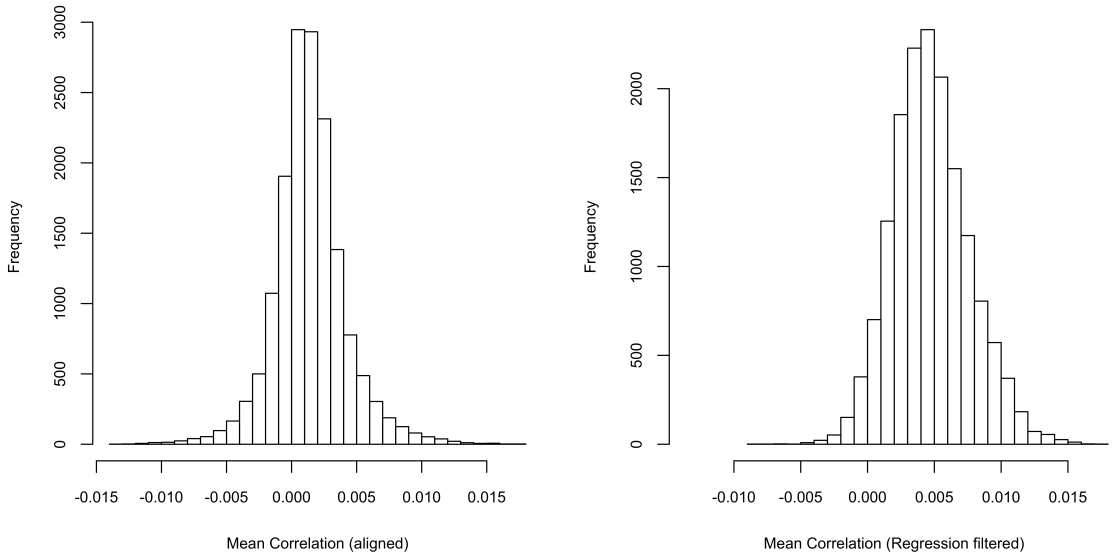
**Figure 4.14:** *Motion artifacts in variance structure after regression filtering. This technique also reduces the magnitude of the variance, but still does not remove the spatial patterns.*



(a) Mean pixel-pixel correlation (structural)



(b) Mean pixel-pixel correlation (functional)




(c) Mean pixel-pixel correlation (functional)

**Figure 4.15:** *Motion artifacts in correlation structure after regression filtering. Regression filtering does not reduce the magnitude of the correlations as much as AR filtering, and the spatial structure remains.*

for evaluating alignment, intensity correction and other motion correction techniques is still lacking. Such an evaluation methodology must clearly define the metric for success. This could be removing detectible artifacts from the structural channel according to one of the three metrics (time course, variance, correlation) discussed here. Alternatively, it could be demonstrating that motion artifacts have no impact on whatever subsequent processing is relevant to a particular lab. What this work demonstrates is that motion must be addressed rather than ignored.

## 4.5 Future work on motion correction

The techniques for dealing with motion described in this chapter are simply the beginning of what is needed for a complete TPCI analysis pipeline. Rigid body image alignment removes much of the visually obvious motion, but many artifacts remain in the time course, variance and correlation structure of the data. Intensity correction techniques which filter the data to account for out-of-plane motion reduce the magnitude of the artifactual variance and correlation but they do not remove the spatial patterns which corrupt further analysis such as cell clustering.

Even before moving to a more complete motion modeling framework, several questions need to be addressed about the current approaches. As mentioned above, a more thorough and defined metric for evaluation is necessary. In addition, the analysis I presented focused on removing artifacts, especially from the structural channel. It will also be important to consider whether these same techniques are removing signal. Determining whether this occurs may require joint TPCI and electrophysiological recordings. Having an electrical recording as the ground truth would allow an analysis of whether motion correction techniques reduced the sensitivity of the TPCI data to detecting neural spikes.

In the long run, it may be important to model the complete motion trajectory

rather than try to filter out its effects. Such an approach would treat the data as a sparse sampling of a three dimensional volume rather than as a dense sampling of a two-dimensional plane. The remaining question would be how to estimate the $z$ component of motion. One general approach to performing this estimation is proposed below.

Assume that motion is periodic and that it doesn't change in period or magnitude over moderate periods of time. Collect data at a particular nominal depth for a short period of time. Change the nominal depth by a very small amount and record for another short period of time. Repeat this process for a number of depths. In theory, due to motion, each recording will actually approximate a sequence of images from a collection of depths surrounding the nominal depth. By aligning the sequences collected at each of the nominal depths, we should be able to estimate the $z$ component of motion.

Even if we can estimate the complete motion trajectory, using the resulting model of motion will require rethinking much of the subsequent processing. Regions of interest will need to be three dimensional, and estimating the fluorescence trace for each ROI will require something more complicated than a simple spatial average.

It is currently unclear to what extent we need to incorporate motion modeling into analysis, but it is clear that the impact of motion on further processing must be acknowledged and either addressed or shown to have no impact.

# 5

# Correlation-based cell clustering

One question that is frequently of interest to experimenters is functional connectivity. Do particular groups of cells have correlated activity? Given sufficient temporal resolution to reliably detect spikes, computing correlations based on spike trains might be the most common approach to this question. There is already significant work looking at correlation patterns in sets of spike trains recorded using elecrophysiological techniques such as multi-electrode arrays. With data recorded at approximately 8Hz, it is unclear whether spike trains can be reliably estimated (see section 5.4). Nevertheless, even without differentiating individual spikes, we might expect the correlation of calcium fluorescence traces to provide information about functional connectivity directly. For non-spiking cells with temporal calcium dynamics, such as astrocytes, there are few alternatives to such a direct approach.

In this chapter I present a preliminary study of correlation-based cell clustering, focusing on the impact of motion artifacts on the results. Though it is, of course, possible to cluster the cells' fluorescence time courses, I show that these clusters are very similar to clusters derived from known artifacts. This suggests that such clustering should not be used to draw scientific conclusions about functional relationships between cells. Though I cannot entirely remove the artifacts influencing the clustering, I show that one particular motion correction approach (regression filtering) shows

promise in minimizing the similarity of estimated functional clusters to known artifactual clusters. This may suggest a direction for future work in finding scientifically meaningful clusterings in *in-vivo* TPCI data.

Throughout this chapter I use an example experiment to demonstrate the challenges and successes of correlation-based clustering. A thorough analysis of many experiments remains to be done. However, I examined several experiments to verify that they followed the patterns described in this chapter. From a preliminary analysis, the chosen experiment appears representative.

## 5.1 Clustering technique

There are numerous approaches to clustering. Most require that there be some similarity metric defined on the space of items to be clustered. The clustering algorithm then attempts to find groups of items that are similar within each group and different between groups. This is a vague goal, with the specifics determined by the particular clustering method. For the exploratory work shown here I used a distance metric of $(1 - correlation)$ and hierarchical clustering with Ward's minimum variance agglomeration method. These are reasonably common methodological choices, with numerous accessible implementations, and it was for this reason that I chose them. Exploratory testing leads me to believe that the results presented in this chapter are not sensitive to changes in the distance metric or clustering technique, but a complete analysis remains to be done.

Bottom-up hierarchical clustering starts by considering each data element as its own cluster of size one. The algorithm then groups the two most similar existing cluster into a single larger cluster. This process repeats until all data elements are joined into a single large cluster. The result of such a hierarchical clustering algorithm is a dendrogram, or tree, which can be cut at any level to produce a particular number

of clusters (see figure 5.3 for an example). The determination of which two clusters to join at each step of the algorithm depends on the measure of cluster similarity used. These can include *single linkage* (minimum distance between points in the two clusters), *complete linkage* (maximum distance between points in the two clusters), *mean linkage* (average distance between pairs of points in the two clusters), and others. Ward's minimum variance method Ward (1963) chooses clusters to join to minimize the total within-cluster variance. I use Ward's method here. Substituting single/complete/mean linkage for Ward's method does change the resulting clustering somewhat, especially in the fine structure of the dendrogram. However, the main argument presented in this chapter hinges on clusterings for small $k$ which do not appear sensitive to the choice of linkage function. In fact, using a partitioning method such as *k-means* in place of hierarchical clustering also appears to change little in the analysis. For future work where changes to the fine structure of a dendrogram or partition are of interest, a more thorough study of appropriate clustering methodology will be called for.

The cluster similarity measures discussed above all depend on their being a measure of distance between data elements. A common distance metric for time series data is the Euclidean distance

$$d(x, y) = \sqrt{\sum_{t=0}^{T} (x_t - y_t)^2}. \tag{5.1}$$

However, it is often useful to normalize the time series first so that differences in centering and scale do not impact the distance metric. If we normalize the time series as

$$\tilde{x_t} = \frac{x_t - \bar{x}}{\sqrt{\sum_t (x_t - \bar{x})^2)/T}} \tag{5.2}$$

then

$$d(\tilde{x}, \tilde{y})^2 = 2T(1 - \rho_{xy}). \tag{5.3}$$

That is, $(1 - correlation)$ is proportional to the squared Euclidean distance between normalized series. Since Ward's minimum variance method for hierarchical clustering requires a distance matrix proportional to squared Euclidean distance, the $(1 - correlation)$ distance metric is a logical choice here.

There is a huge literature on clustering in general, and on each of the many clustering methods. It is beyond the scope of this work to provide a thorough review, but one source that discusses each of the techniques mentioned here (in the context of genetic microarray data) is Chipman et al. (2003).

## 5.2 Artifactual clusters

We know from chapter 4 that motion creates a variety of artifacts, including in the correlation structure. As a result, there is good reason to suspect that clusters computed from this correlation structure may be corrupted by motion. Of course, cellular activity may also, we hope, impact correlation based clustering. We would like to have a way of separating artifactual and functional clustering. A method to prove that a clustering is driven by activity rather than motion would most likely require joint elecrophysiological recording for confirmation. However, it is possible to demonstrate convincingly that some clusterings *are* driven by motion.

As described in chapter 4, a primary driver of motion is respiration, which is controlled by a respirator and therefore has a consistent frequency. As seen in the data spectra (e.g. figure 4.2), respiration causes a periodic impact on the intensity of the calcium trace. As seen in the correlation images (e.g. figure 4.5), motion induces two groups of cells. If we can identify these two motion-driven clusters of cells, we can compare them to clusters intended to represent functional relationships. If the supposed functional clusters are similar to those caused by motion, this is strong evidence that the functional clusters are corrupted.

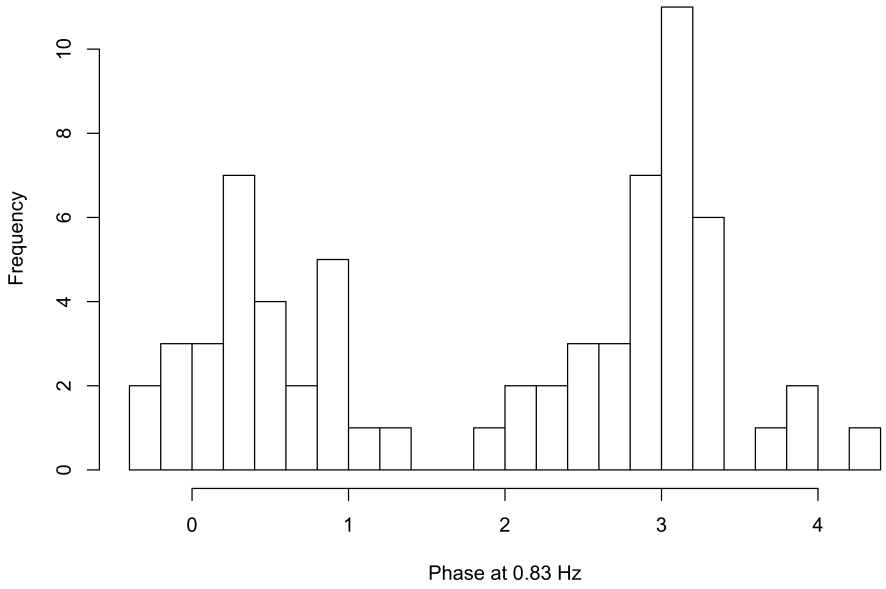



**Figure 5.1:** *The phases of the calcium traces at 0.82 Hz (the frequency of respiration) show a clearly bimodal distribution. This induces an artifactual clustering.*

To identify motion-driven clusters, I chose to isolate the respiration component of motion. In the example experiment used here, respiration had a frequency of $0.82Hz$. For each of the 67 ROIs in the example experiment (as returned by the MaSCS algorithm), I computed the phase of $0.82Hz$ component of the calcium trace using the Fourier transform. The distribution of these phases, shown in figure 5.1, is clearly bimodal. Using circular phase differences as a distance metric, I clustered the ROIs into two clusters. Arguably, these two clusters are primarily a reflection of motion artifacts rather than neural activity.

We wish to know whether these phase-based artifactual clusters drive clusters meant to be functional, and whether any of the motion correction techniques have an impact. To answer this question, we need a measure of cluster similarity. If this similarity measure with phase-based clusters is high, we have strong evidence

that motion is corrupting clustering. If the similarity is low, the clustering may be indicating functional relationships instead of motion artifacts. One common metric of cluster similarity is the Hubert and Arabie adjusted Rand index (Hubert and Arabie, 1985), a modification of the Rand index that accounts for chance similarity.

The Rand index measures the similarity of clusterings based on the correspondence between assignments of pairs of objects. Consider two clusterings $C_1$ and $C_2$, and all pairs of objects involved in the clusterings. Place each pair into one of four categories:

- **(a)** objects are in the same class in $C_1$; objects are in the same class in $C_2$
- **(b)** objects are in different classes in $C_1$; objects are in the same class in $C_2$
- **(c)** objects are in different classes in $C_1$; objects are in different classes in $C_2$
- **(d)** objects are in the same class in $C_1$; objects are in different classes in $C_2$

The Rand index is defined as

$$RI = \frac{a+d}{a+b+c+d} \tag{5.4}$$

where $a$, $b$, $c$, and $d$ are the number of pairs of that type. If $C_1$ and $C_2$ are identical, the Rand index will be 1. Though it is possible for the Rand index to be 0, this will happen rarely in practice. Due to chance agreement between clusterings, the expected value of the Rand index for a reasonable null situation (for instance, for two binary clusterings created independently using Bernoulli coin flips) will be some positive number less than 1.

Several researchers have proposed adjustments to the Rand index such that it has an expected value of 0 for cases where the agreement between clusterings is due to chance. The Hubert and Arabie adjusted Rand index takes the form

$$ARI = \frac{a+d-n_c}{a+b+c+d-n_c} \tag{5.5}$$

where

$$n_c = \frac{n(n^2+1) - (n+1)\sum n_{i\cdot}^2 - (n+1)\sum n_{\cdot j}^2 + 2\sum\sum n_{\cdot j}^2 n_{i\cdot}^2}{2(n-1)},$$

$n_{i\cdot}$ is the number of items in cluster $i$ in $C_1$, $n_{\cdot j}$ is the number of items in cluster $j$ in $C_2$, and $n$ is the total number of objects. See Hubert and Arabie (1985) for details.

Milligan and Cooper (1986) showed through simulation studies that, compared to other common cluster similarity metrics, the Hubert and Arabie adjusted Rand index had the best performance when comparing clusterings with different numbers of clusters (for instance, from different levels of a hierarchical clustering). Their primary metric of improved performance was that the index had an expected value that was consistently very close to 0 across values of $k$ (number of clusters).

I verified the results in Milligan and Cooper (1986) with a short simulation study comparing the two-cluster phase-based clustering with simulated clusterings. I created each simulated clustering with $k$ clusters with $n$ draws from the discrete uniform distribution on the integers $1, ..., k$. Figure 5.2 shows the empirical distribution of the adjusted Rand index (ARI) for values of $k$ between 2 and 10. These values are generally very close to 0, though they show a noticeable positive bias. It is relevant to the following sections that nearly all of the ARI values from this simulated null distribution are less than 0.1, and the majority are less than 0.05.

## 5.3 Calcium trace clustering

An obvious way of clustering the 67 ROIs in the example experiment is to use the correlation matrix between the 67 time series to compute the distance matrix. Figure 5.3 shows the results of hierarchical clustering based on this distance matrix. In the figure, each ROI is labeled with a color corresponding to the cluster membership from the phase-based clustering described in the previous section. Even from visual inspection, it is clear that the main split of the hierarchical clustering strongly

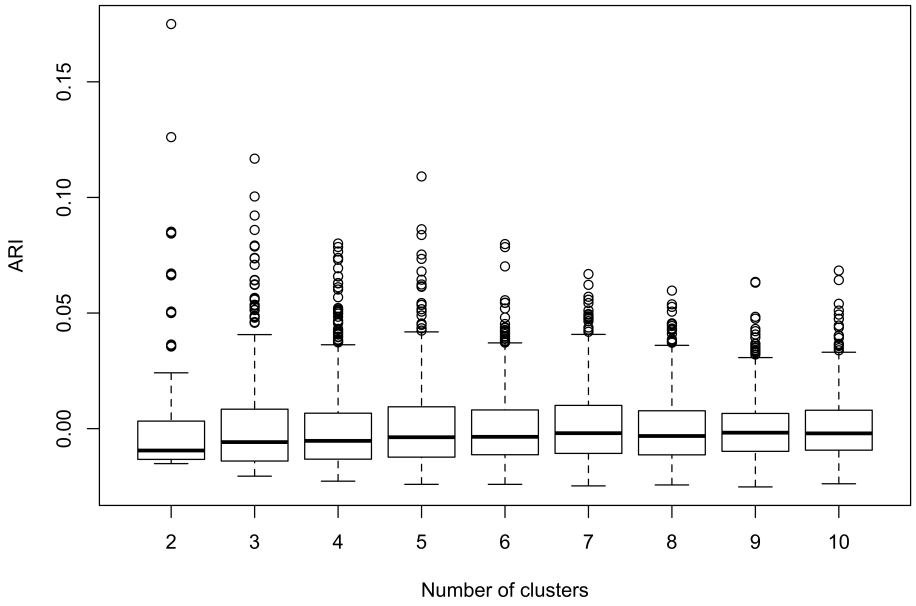



**Figure 5.2:** *Empirical null distribution of ARI values for various $k$. Each boxplot represents the Hubert and Arabie adjusted Rand index of the two-cluster phase-based clustering compared to 1000 clusterings chosen independently from the discrete uniform distribution on the integers $1, ..., k$. The adjusted Rand index has a positive bias, but is very close to 0 for all values of $k$.*

reflects the phase-based grouping. This indicates that the correlation-based clustering, intended to indicate functional relationships between cells, is instead primarily describing motion artifacts.

The same conclusion can be drawn from looking at the adjusted Rand index comparing the correlation-based clustering to the phase-based clustering. I will call this statistic *agreement with phase clustering*, or APC. Figure 5.4 shows the APC at various cuts in the hierarchical clustering (various $k$). For the raw data (shown in figure 5.3), the APC starts at 0.6 for 2 clusters and decreases to 0.2 for 10 clusters. At all points this is significantly above the values we would expect were the hierarchical clustering to be unrelated to the phase-based clustering. Figure 5.4 also shows the

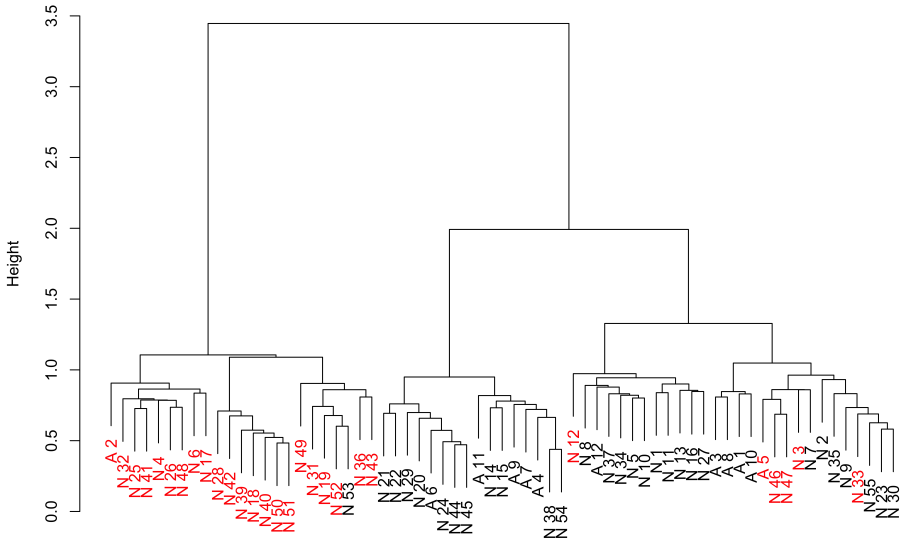



**Figure 5.3:** *Hierarchical clustering of the 67 ROIs in the example experiment based on correlation, computed from the raw data without any motion correction. The label colors indicate the clusters derived from the motion-related phase difference. Each ROI is named by a string starting with N (Neuron) or A (astrocyte) followed by an arbitrary index. It is clear from visual inspection that the hierarchical correlation-based clustering is consistent with the clustering based on phase. This suggests that it is corrupted by motion artifacts.*

same analysis for data on which various motion correction techniques were applied (image alignment alone, image alignment followed by AR filtering, and image alignment followed by regression filtering). Despite the fact that regression filtering did not appear to reduce spurious correlations in the structural channel as much as AR filtering, this analysis shows it to be the only method that reduces the APC. Even then, the reduction is only for small $k$ and the APC remains above what we would expect from chance.

In some ways, the results in figure 5.4 are unsurprising. The correlation due to respiration-driven motion will be consistently present throughout the recording, whereas functional correlation may come and go. Therefore, the signal we are searching for (functional connectivity) will likely be at its weakest when looking at cor-

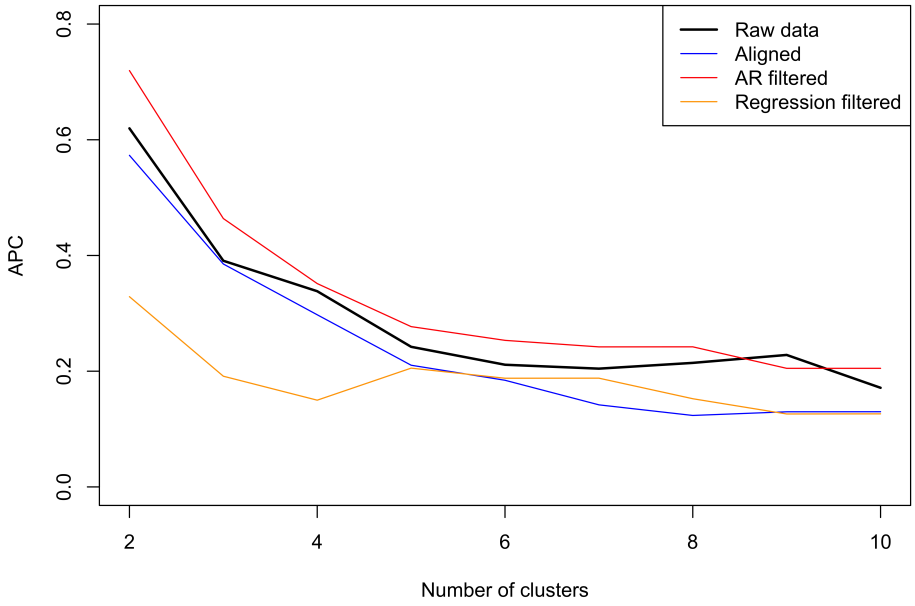



**Figure 5.4:** *Plot of APC against k, the number of clusters, for full-series clustering.*

relations of long series, whereas the noise (artifactual correlation) will likely be at its strongest. Instead of considering the entire experiment, we can look at short sequences of data (perhaps corresponding to stimulation trials, or some other factor of experimental interest). Figure 5.5 shows the same APC analysis for a short (1.2 second, 10 frame) section of data. This amount of time only allows for approximately one full respiration cycle, so we would expect the APC to be lower. In fact it is substantially lower. For most $k$, the raw data still has an APC outside of the range of 95% of the null simulation values. However, all of the motion correction techniques reduce the APC to values indistinguishable from chance.

Figure 5.5 suggests that motion has a smaller influence on correlation-based clustering when considering small segments of data. The obvious follow-up question is *How short is short?* Figure 5.6 shows the APC for a fixed $k$ (3 clusters) for various

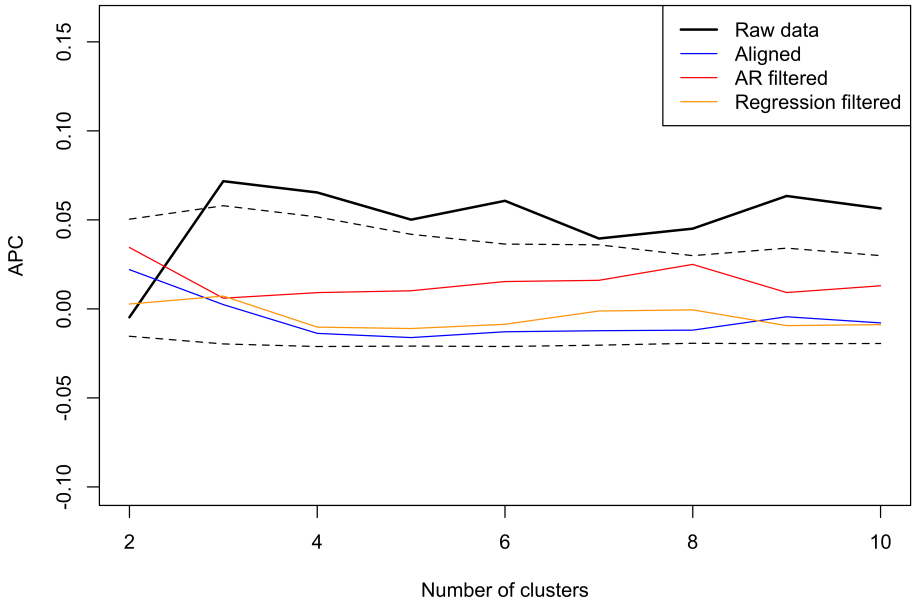



**Figure 5.5:** *Plot of APC against k, the number of clusters, for clustering a short segment of data (about 1.2 seconds). The dotted lines show the empirical 0.025% and 0.975% percentiles of the null distribution of the APC*

data lengths. This analysis shows that the APC tends to rise quickly with the number of timepoints used to compute the correlation distance matrix. Again, we see that the regression filtering approach is the only motion correction technique that has a noticeable impact on the APC. In fact, it appears that regression filtering keeps the APC very low for sequences up to 1000 frames (about 2 minutes). It seems that regression filtering is most effective at removing the artifacts captured by this analysis. Not addressed by this analysis is whether the regression filtering also removes signal. Reducing the calcium traces to noise would, after all, be effective at reducing the APC.

Figures 5.5 and 5.6 suggest that correlation-based clustering based on short segments of data may be free from motion artifacts and that regression filtering reduces

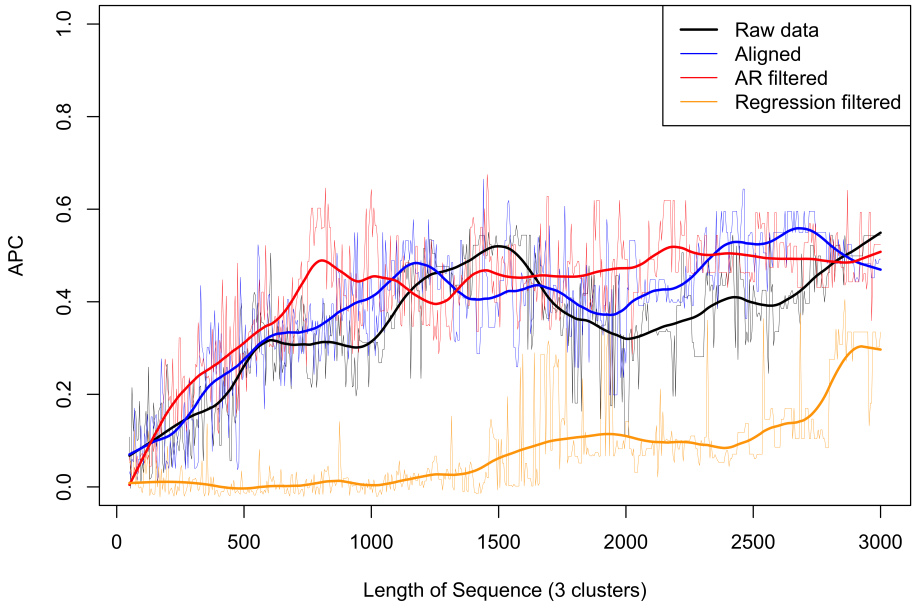



**Figure 5.6:** *Plot of APC for three clusters against t, the length of the sequence (in frames) used for clustering. The light lines are the actual APC values, computed in increments of 5 frames. The dark lines are smoothed versions of the APC series. Regression filtering is the only motion correction approach to consistently reduce APC values, even for moderately long segments of data.*

motion artifacts in clustering based on longer segments. Of course, it is possible that there are artifacts present that are simply not apparent from this analysis. This analysis also gives no information about whether the clusters found with short data segments are meaningful or are simply a reflection of noise. The development of methods to choose an appropriate $k$ and to quantify the significance of clustering for this data is an open area of research.

## 5.4 A note on spike train estimation

Estimating spike trains before computing a distance matrix for clustering may reduce the impact of motion and reduce artifactual clusters. However, the data for this work does not support spike train estimation by the easily available methods (the deconvolution techniques of Vogelstein et al. (2009, 2010) for which code is publicly available). These methods estimate the probability of a spike in each image frame, but to extract any spikes from the data used for the preceding exploration of clustering, the threshold had to be set below 0.5. The resulting spike trains were highly sensitive to any alteration of the data (image alignment, for instance), suggesting that they were largely driven by noise.

Despite the failure of out-of-the-box spike train extraction algorithms, it is clear that there is signal in the data. In experiments with a stimulus, there is a clear (visible even to the naked eye in video) increase in fluorescence after stimulation. However, the frame rate for this data is around $8Hz$. From previous work, we know that the calcium transient from a spike has a rise time of 5-50ms and a decay time of 1-4 seconds (Smetters et al., 1999). Without noise, this would be detectable with an $8Hz$ sampling rate, but with noise it may be vital to sample near the peak of the calcium transient if we want to recover precise spike trains. Most likely, joint *in-vivo* electrophysiological and calcium imaging data will be necessary to explore the feasibility of spike train extraction from this data.

Nevertheless, even with reliable spike train estimation methods, the analysis presented in this chapter is still relevant. The main conclusion to draw from this work is that we must be conscious of the possible corrupting and misleading artifacts from motion, regardless of the processing performed.

## 5.5 Conclusions and future work

The main take-away from the work presented here is that even small amounts of motion can cause dramatic artifacts in correlation-based clustering of fluorescence traces. It is imperative that experimenters wishing to draw scientific conclusions from this or similar analyses first demonstrate convincingly that these artifacts have been mitigated. This may involve estimating spike trains before clustering, clustering based on short segments of data, improving upon the promising regression filtering approach, or something different. One of the strengths of TPCI is that it can reveal the function of populations of cells, so studying clustering and other types of group structure is interesting and important. This is a prime area for further methodological development.

One particular area for future work is in identifying and developing appropriate distance metrics for clustering fluorescence traces. In the work here, I used the common $(1 - \mathit{correlation})$ distance, but without substantial justification. There may be distance metrics that are more suited to this task. Estimating spike trains before calculating correlation distances is one example of a customized metric. By reducing the data to spike trains, we hope that much of the noise and many of the correlation-inducing artifacts may be removed. However, estimating spike trains is not always possible or appropriate. Perhaps there exists a distance metric on the fluorescence traces which emphasizes the functionally meaningful characteristics without reducing them to spike trains. Finding such a metric will require both methodological and experimental study, and is likely to be different for different cell types with different calcium activity profiles (neurons and astrocytes).

Another pressing area of additional study is the analysis of the clusters that result from any clustering analysis. This includes statistical concerns such as measures of cluster separation and compactness, but also neuroscientific questions about the

scientific significance of groups of cells with correlated calcium dynamics. The study of functional connectivity is currently widely studied at the brain-wide scale in fMRI, EEG, and MEG studies. Perhaps there is transferable knowledge that can be applied toward quantifying and understanding cell groupings at the scale of TPCI.

# Part III

# Conclusions

# 6

# Resources and conclusions

## 6.1 Resources for the research community

Encouragingly, TPCI analysis methodology is an area receiving increasing focus from the research community. As a result, the amount of information and number of tools available are growing. Nevertheless, there is still a lot of open area to develop, and an increasing amount of coordination required between developers, computational researchers and experimentalists. The body of this dissertation described my research and development of analysis techniques that I hope will be useful to TPCI experimenters. However, taking analysis methodology from the development to the production stage requires significant additional work. Though I cannot provide production-ready software, I have created several tangible resources for the community beyond this document. These include RCI, the R package with the code used for all the analysis presented here, as well as *CaliCode.org*, a website synthesizing the currently available papers, resources, and tools.

### 6.1.1 Calicode.org

For an experimental lab, the burden of acquiring the experimental tools, skills and paradigms for a new imaging modality is severe. For recently developed modalities

such at TPCI, there is the additional burden of finding or developing data processing tools. I know from experience that is is very hard to search for analysis examples in the literature, as the details of what others have done are often not the focus of the papers (and consequently not in their titles or keywords). The scientific literature is vitally important for directing research in the TPCI field, but it is not an efficient index of available resources for data analysis. As a small step toward providing such an index, I started the wiki *CaliCode.org*. The main contributions of this site are to provide an annotated bibliography of papers discussing TPCI data processing, and to collect links to code and toolkits that are publicly available. My hope is that when researchers enter the field of calcium imaging, this site will increase the chances of them finding the resources they need. It is impossible to know for sure how useful the site has been for this intended audience, but since its beginnings in early 2013, there have been nearly 200 unique returning visitors to the site.

### 6.1.2 RCI software package

I believe strongly that data analysis methodological development must be presented along with the necessary tools to replicate the work. No document on data processing, including this one, can provide the level of algorithmic detail needed to faithfully reproduce any but the simplest code. For future researchers attempting to compare their work to that already done, it becomes impossible to differentiate between differences due to data, implementation, and algorithm. Unfortunately, as is the case here, the development of production-ready software is frequently beyond the scope of initial methodological research. Nevertheless, I have made all the code used in this work publicly available as the RCI package for the R programming language (available on Github at *https://github.com/dancingwoods/RCI*). The code is fully documented, providing the implementation details that are lacking from this dissertation document.

RCI is a prototype version of software that performs the analysis described in this document. It is not meant for widespread experimental deployment, but rather as a model for the future software development. Though I chose to use the R language (in combination with some C code for speed), I would not recommend this choice for tools meant for deployment to the neuroscientific community. Currently, the most common environments for TPCI analysis as reported in the literature are the open source image processing toolkit ImageJ, and the scientific computing language Matlab. Both of these environments support the development of extensions, which could easily include modules for TPCI processing.

## 6.2 Summary

In this dissertation I have presented research on three particular tasks in the analysis of TPCI data: region of interest segmentation, motion correction, and correlation-based cell clustering.

In chapter 3, I presented the MaSCS procedure for ROI segmentation. MaSCS represents a step forward in the field as it provides flexible multi-class, automated segmentation with performance that is similar to existing techniques but improves with use. If adopted into common use, the MaSCS framework has sufficient structure to improve standardization and communication between experimenters, while also being flexible and adaptable enough to accommodate the eccentricities and specific needs of particular labs.

Chapter 4 explored motion artifacts and their impact on various data summaries that may be used for scientific inference. Though image alignment is a widely applied tool for motion correction in TPCI, prior to this work there were no studies of the relative effectiveness of the wide variety of alignment techniques available. I expanded my analysis beyond image alignment to discuss intensity correction approaches de-

signed to account for out-of-plane motion. The two techniques I developed (AR and regression filtering) reduced but did not eliminate the effects of motion on the variance and correlation structure of the data. This work on motion correction is simply the start of a conversation. Better motion correction techniques are a necessity, but before they can be developed we must clearly define what it means to *remove* or *correct for* motion. The data summaries I used to quantify motion artifacts are the beginnings of such a definition.

Finally, in chapter 5 I presented preliminary work on clustering cells based on the correlation of their fluorescence traces. This is a task that is of interest to experimenters when spike train analysis is not practical (either because of experimental parameters or because the target of the experiment is non-spiking cells such as astrocytes). I find that with the current data and motion correction techniques, it is likely that the results of clustering algorithms based on correlation distances are highly corrupted by motion artifacts. I show preliminary evidence that regression filtering may reduce the impact of these artifacts. This work invites future research to describe the parameters that govern when and if correlation based clustering is a justifiable TPCI analysis technique.

There remain large numbers of opportunities to improve the analysis of TPCI data. Each of the three chapters summarized above described relevant future work. Beyond improvements to these three particular tasks, an interesting area of research is the creation of a standardized processing pipeline. A standard toolkit for TPCI data analysis would provide an indexed set of tools for experimenters to use. A processing pipeline would involve the additional step of justifying particular combinations and sequences of analysis methodology under common experimental paradigms. Standard toolkits and pipelines facilitate direct comparison and synthesis of results from different labs. They reduce duplication of effort, and allow researchers to more easily identify targets for scientific or methodological innovation. Fortunately, many scien-

tists are starting to work toward these goals. It is my hope that this document may be useful to those continuing to work on methodological development for TPCI data analysis.

# Bibliography

Bankman, I. N., ed (2009), *Handbook of Medical Image Processing and Analysis*, 2 edn Elsevier.

Bean, B. P. (2007), "The action potential in mammalian central neurons.," *Nature reviews. Neuroscience*, 8(6), 451–65.

Berridge, M. J., Lipp, P., and Bootman, M. D. (2000), "The Versitality and Universality of Calcium Signalling," *Nature Reviews Molecular Cell Biology*, 1(October).

Bonin, V., Histed, M. H., Yurgenson, S., and Reid, R. C. (2011), "Local diversity and fine-scale organization of receptive fields in mouse visual cortex.," *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 31(50), 18506–21.

Breiman, L. (2001), "Random forests," *Machine learning*, pp. 5–32.

Brown, L. G. (1992), "A Survey of Image Registration," *ACM Computing Surveys*, 24(4).

Catterall, W. a. (2011), "Voltage-Gated Calcium Channels.," *Cold Spring Harbor perspectives in biology*, .

Chen, T., Xue, Z., Wang, C., Qu, Z., Wong, K. K., and Wong, S. T. C. (2010), "Motion artifact correction of multi-photon imaging of awake mice models using speed embedded HMM.," *Medical image computing and computer-assisted inter-*

*vention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 13(Pt 3), 473–80.

Chipman, H., Hastie, T. J., and Tibshirani, R. (2003), "Clustering Microarray Data," in *Statistical Analysis of Gene Expression Microarray Data*, ed. T. Speed, pp. 161–204.

Diego, F., Reichinnek, S., Both, M., and Hamprecht, F. A. (2013), Automated Identification of Neuronal Activity from Calcium Imaging by Sparse Dictionary Learning,, in *ISBI*.

Ding, S. (2012), "In Vivo Imaging of Ca 2 + Signaling in Astrocytes Using Two-Photon Laser Scanning Fluorescent Microscopy," in *Astrocytes: Methods and Protocols, Methods in Molecular Biology*, ed. R. Milner, Vol. 814 Springer Science+Business Media, LLC, chapter 36, pp. 545–554.

Dolphin, A. C. (2006), "A short history of voltage-gated calcium channels.," *British journal of pharmacology*, 147 Suppl, S56–62.

Dombeck, D. a., Khabbaz, A. N., Collman, F., Adelman, T. L., and Tank, D. W. (2007), "Imaging large-scale neural activity with cellular resolution in awake, mobile mice.," *Neuron*, 56(1), 43–57.

Drew, P. J., Shih, A. Y., and Kleinfeld, D. (2011), "Fluctuating and sensory-induced vasodynamics in rodent cortex extend arteriole capacity.," *Proceedings of the National Academy of Sciences of the United States of America*, 108(20), 8473–8.

Eddy, W. F., Fitzgerald, M., and Noll, D. C. (1996), "Improved Image Registration by Using Fourier Interpolation," *Magn Reson Med.*, (7), 923–931.

Eddy, W. F., and Young, T. K. (2009), "Optimizing MR Image Resampling," in *Handbook of Medical Image Processing and Analysis*, pp. 675–684.

Eichhoff, G., Kovalchuk, Y., Varga, Z., and Garaschuk, O. (2010), "In Vivo Ca2+ Imaging of the Living Brain Using Multi-cell Bolus Loading Technique," in *Calcium Measurement Methods*, eds. A. Verkhratsky, and O. H. Petersen, Vol. 43 Humana Press.

Feldt Muldoon, S., Soltesz, I., and Cossart, R. (2013), "Spatially clustered neuronal assemblies comprise the microstructure of synchrony in chronically epileptic networks.," *Proceedings of the National Academy of Sciences of the United States of America*, .

Foroosh, H., Zerubia, J. B., and Berthod, M. (2002), "Extension of phase correlation to subpixel registration.," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 11(3), 188–200.

Garaschuk, O., and Griesbeck, O. (2010), "Monitoring Calcium Levels With Genetically Encoded Indicators," in *Calcium Measurement Methods*, eds. A. Verkhratsky, and O. H. Petersen, Vol. 43 of *Neuromethods*, Totowa, NJ: Humana Press, pp. 101–117.

Greenberg, D. S., and Kerr, J. N. D. (2009), "Automated correction of fast motion artifacts for two-photon imaging of awake animals.," *Journal of neuroscience methods*, 176(1), 1–15.

Guizar-Sicairos, M., Thurman, S. T., and Fienup, J. R. (2008), "Efficient subpixel image registration algorithms.," *Optics letters*, 33(2), 156–8.

Hill, E. S., Moore-kochlacs, C., Vasireddi, S. K., Sejnowski, T. J., and Frost, W. N. (2010), "Validation of Independent Component Analysis for Rapid Spike Sorting of Optical Recording Data," *Journal of Neurophysiology*, pp. 3721–3731.

Hira, R., Ohkubo, F., Ozawa, K., Isomura, Y., Kitamura, K., Kano, M., Kasai, H., and Matsuzaki, M. (2013), "Spatiotemporal Dynamics of Functional Clusters of

Neurons in the Mouse Motor Cortex during a Voluntary Movement.," *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 33(4), 1377–1390.

Hubert, L., and Arabie, P. (1985), "Classification ˜1985," *Journal of Classification*, 218, 193–218.

Johnson, I. (1998), "Fluorescent probes for living cells.," *The Histochemical journal*, 30(3), 123–40.

Katona, G., Szalay, G., Maák, P., Kaszás, A., Veress, M., Hillier, D., Chiovini, B., Vizi, E. S., Roska, B., and Rózsa, B. (2012), "Fast two-photon in vivo imaging with three-dimensional random-access scanning in large tissue volumes," *Nature Methods*, 9(2).

Kerr, J. N. D., and Denk, W. (2008), "Imaging in vivo: watching the brain in action.," *Nature reviews. Neuroscience*, 9(3), 195–205.

Kerr, J. N. D., Greenberg, D., and Helmchen, F. (2005), "Imaging input and output of neocortical networks in vivo.," *Proceedings of the National Academy of Sciences of the United States of America*, 102(39), 14063–8.

Lindeberg, T. (1998), "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision*, 30(2), 79–116.

Lohr, C., and Deitmer, J. W. (2010), "Calcium Imaging of Glia," in *Calcium Measurement Methods*, eds. A. Verkhratsky, and O. H. Petersen, Vol. 43 of *Neuromethods*, Totowa, NJ: Humana Press.

Lucas, B., and Kanade, T. (1981), "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proceedings of the 7th international joint ...*, (x), 674–679.

Lütcke, H., Murayama, M., Hahn, T., Margolis, D. J., Astori, S., Zum Alten Borgloh, S. M., Göbel, W., Yang, Y., Tang, W., Kügler, S., Sprengel, R., Nagai, T., Miyawaki, A., Larkum, M. E., Helmchen, F., and Hasan, M. T. (2010), "Optical recording of neuronal activity with a genetically-encoded calcium indicator in anesthetized and freely moving mice.," *Frontiers in neural circuits*, 4(April), 9.

Malik, W. Q., Schummers, J., Sur, M., and Brown, E. N. (2011), "Denoising two-photon calcium imaging data.," *PloS one*, 6(6), e20490.

Milligan, G. W., and Cooper, M. C. (1986), "Multivariate Behavioral A Study of the Comparability of External Criteria for Hierarchical Cluster Analysis," *Multivariate Behavior*, (July 2013), 37–41.

Miri, A., Daie, K., Burdine, R. D., Aksay, E., and Tank, D. W. (2011), "Regression-based identification of behavior-encoding neurons during large-scale optical imaging of neural activity at cellular resolution.," *Journal of neurophysiology*, 105(2), 964–80.

Mishchencko, Y., Vogelstein, J. T., and Paninski, L. (2011), "A Bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data," *The Annals of Applied Statistics*, 5(2B), 1229–1261.

Mittmann, W., Wallace, D. J., Czubayko, U., Herb, J. T., Schaefer, A. T., Looger, L. L., Denk, W., and Kerr, J. N. D. (2011), "Two-photon calcium imaging of evoked activity from L5 somatosensory neurons in vivo," *Nature Neuroscience*, 14(8), 1089–1093.

Mukamel, E. a., Nimmerjahn, A., and Schnitzer, M. J. (2009), "Automated analysis of cellular signals from large-scale calcium imaging data.," *Neuron*, 63(6), 747–60.

Nagashima, S., Aoki, T., and Higuchit, T. (2006), "A Subpixel Image Matching Technique Using Phase-Only Correlation," , pp. 701–704.

Nimmerjahn, A., Kirchhoff, F., Kerr, J. N. D., and Helmchen, F. (2004), "Sulforhodamine 101 as a specific marker of astroglia in the neocortex in vivo," *Nature Methods*, 1(1), 1–7.

Paukert, M., and Bergles, D. E. (2012), "Reduction of motion artifacts during in vivo two-photon imaging of brain through heartbeat triggered scanning.," *The Journal of physiology*, 590(Pt 13), 2955–63.

Ranganathan, G. N., and Koester, H. J. (2010), "Optical recording of neuronal spiking activity from unbiased populations of neurons with high spike detection efficiency and high temporal precision.," *Journal of neurophysiology*, 104(3), 1812–24.

Reeves, A. M. B., Shigetomi, E., and Khakh, B. S. (2011), "Bulk loading of calcium indicator dyes to study astrocyte physiology: key limitations and improvements using morphological maps.," *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 31(25), 9353–8.

Smetters, D., Majewska, A., and Yuste, R. (1999), "Detecting action potentials in neuronal populations with calcium imaging.," *Methods: A Companion to Methods in Enzymology*, 18(2), 215–21.

Soulet, D., Paré, A., Coste, J., and Lacroix, S. (2013), "Automated Filtering of Intrinsic Movement Artifacts during Two-Photon Intravital Microscopy," *PLoS ONE*, 8(1), e53942.

Stosiek, C., Garaschuk, O., Holthoff, K., and Konnerth, A. (2003), "In vivo two-photon calcium imaging of neuronal networks.," *Proceedings of the National Academy of Sciences of the United States of America*, 100(12), 7319–24.

Takahashi, A., Camacho, P., Lechleiter, J. D., and Herman, B. (1999), "Measurement of intracellular calcium.," *Physiological reviews*, 79(4), 1089–125.

Takita, K., Member, S., Aoki, T., Sasaki, Y., and Members, R. (2003), "High-Accuracy Subpixel Image Registration Based on Phase-Only Correlation," *IEICE Trans. Fundamentals*, (8), 1925–1934.

Thomson, D. J. (1982), "Spectrum estimation and harmonic analysis," *Proceedings of the IEEE*, 70.

Tomek, J., Novak, O., and Syka, J. (2013), "Two-Photon Processor and SeNeCA - A freely available software package to process data from two-photon calcium imaging at speeds down to several ms per frame.," *Journal of neurophysiology*, .

Tsien, R. Y. (1980), "New Calcium Indicators and Buffers with High Selectivity against Magnesium and Protons: Design, Synthesis, and Properties of Prototype Structures," *Biochemistry*, 19(11), 2396–2404.

Valmianski, I., Shih, A. Y., Driscoll, J. D., Matthews, D. W., Freund, Y., and Kleinfeld, D. (2010), "Automatic identification of fluorescently labeled brain cells for rapid functional imaging.," *Journal of neurophysiology*, 104(3), 1803–11.

Verkhratsky, A., and Petersen, O. H., eds (2010), *Calcium Measurement Methods*, New York: Humana Press.

Vogelstein, J. T., Packer, A. M., Machado, T. a., Sippy, T., Babadi, B., Yuste, R., and Paninski, L. (2010), "Fast nonnegative deconvolution for spike train inference from population calcium imaging.," *Journal of neurophysiology*, 104(6), 3691–704.

Vogelstein, J. T., Watson, B. O., Packer, A. M., Yuste, R., Jedynak, B., and Paninski, L. (2009), "Spike inference from calcium imaging using sequential Monte Carlo methods.," *Biophysical journal*, 97(2), 636–55.

Ward, J. H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58(301), 236–244.

Yaksi, E., and Friedrich, R. W. (2006), "Reconstruction of firing rate changes across neuronal populations by temporally deconvolved Ca 2 + imaging," *Nature Methods*, 3(5), 377–383.

Yu, Y., and Wang, J. (2012), Highly Accurate Estimation of Sub-pixel Motion Using Phase Correlation,, in *CCPR*, pp. 186–193.

Zitova, B. (2003), "Image registration methods: a survey," *Image and Vision Computing*, 21(11), 977–1000.