UNIVERSITY OF CALIFORNIA, SAN DIEGO

Mapping Functional Connectivity in Cellular Networks

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Bioengineering

by

Marius Buibas

Committee in charge:

Professor Gabriel Silva, Chair
Professor Henry Abarbanel
Professor Jeff Hasty
Professor Terry Sejnowski
Professor Shankar Subramaniam

2011

The dissertation of Marius Buibas is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2011

EPIGRAPH

... as far as the laws of mathematics refer to reality, they are not certain;
and as far as they are certain, they do not refer to reality.
—Albert Einstein

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Prof. Gabriel Silva for his guidance and support. I would also like to acknowledge my fellow lab members for their collaborations and many interesting conversations. I'd like to thank my family for their support in my many long nights of work as a grad student.

Chapter 2, in full, is a reprint of the material as it appears in Buibas M, Yu D, Nizar K, and Silva GA. "Mapping the Spatiotemporal Dynamics of Calcium Signaling in Neural Networks Using Optical Flow" *Annals of Biomedical Engineering.*, 10.1007, 2010. The dissertation author was the primary investigator and author of this paper. The authors retain copyright of this article, and contributing authors have granted permission for inclusion in this thesis.

Chapter 3, in full, is a reprint of the material as it appears in Buibas M, and Silva GA. "A Framework for Simulating and Estimating the State and Functional Topology of Complex Dynamic Geometric Networks" *Neural Computation.*, pp. 183-214 vol. 23 (1), 2011. The dissertation author was the primary investigator and author on this paper. The authors retain copyright of this article.

Chapter 4, in part is currently being prepared for submission for publication of the material. Buibas M and Silva GA. The dissertation author was the primary investigator and author of this paper.

VITA AND PUBLICATIONS

1996	B. S. in Mechanical Engineering, University of California, San Diego
1994-1997	Software Engineer, Biosym (Accelrys), San Diego, Paris
1997-2005	R&D Engineer, Hewlett-Packard, San Diego
2005-2007	HP Resident Fellowship, Master of Science, Bioengineering, University of California, San Diego
2007	M. S. in Bioengineering, University of California, San Diego
2007-2011	Graduate Student Instructor and Researcher, University of California, San Diego
2011	Ph. D. in Bioengineering, University of California, San Diego

M Buibas and GA Silva. "A framework for simulating and estimating the state and functional topology of complex dynamic geometric networks." *Neural Computation*, 23(1):183214, Jan 2011.

M Buibas, D Yu, K Chiao, GA Silva, "Mapping calcium signaling in neurons and glia using optical flow", *Annals of Biomedical Engineering*, 2010.

D Yu, M Buibas, SK Chow, IY Lee, Z Singer, GA Silva, "Characterization of Calcium-Mediated Intracellular and Intercellular Signaling in the rMC-1 Glial Cell Line", *Cellular and Molecular Bioengineering*, vol. 2, pp. 144-155, 2009.

M Hashemi, M Buibas, GA Silva, "Automated detection of intercellular signaling in astrocyte networks using the converging squares algorithm", *Journal of neuroscience methods*, vol. 170, pp. 294-299, 2008.

CL MacDonald, D Yu, M Buibas, GA Silva, "Diffusion modeling of ATP signaling suggests a partially regenerative mechanism underlies astrocyte intercellular calcium waves" *Frontiers in Neuroengineering*, vol. 1, 2008.

ABSTRACT OF THE DISSERTATION

Mapping Functional Connectivity in Cellular Networks

by

Marius Buibas

Doctor of Philosophy in Bioengineering

University of California, San Diego, 2011

Professor Gabriel Silva, Chair

My thesis is a collection of theoretical and practical techniques for mapping functional or effective connectivity in cellular neuronal networks, at the cell scale. This is a challenging scale to work with, primarily because of the difficulty in labeling and measuring the activities of networks of cells. It is also important as it underlies behavior, function, and complex diseases. I present methods to measure and quantify the dynamic activities of cells using the optical flow technique, which can identify activity and directions of information processing using calcium fluorescence measurements.

I present a unified framework for simulation and estimation of neuronal activity, tailored towards interpretation of experimental data, and implemented in a fully parallel fashion on graphics processor unit (GPU) cards. This framework permits

experimenters to estimate hidden quantities in collected data, using any neuronal or astrocyte model. I introduce a technique for mapping functional connectivity in neuronal networks, using experimental data and an arbitrary state space model. The technique makes some simplifications that reduces the dimensionality of the estimation problem, and shows excellent performance for networks of up to 30 possible independent incoming connections.

While the framework and mapping algorithms use a state space, parametric representation of individual cell dynamics, I've also developed a time-embedded, nonparametric technique for estimating input-output relationships, and applied it to estimating current from voltage measurements and spikes from fluorescent calcium. Without any knowledge of the underlying neuronal dynamics, this technique can reconstruct a current signal from measured voltage in mouse pyramidal neurons with an R-value of 0.9.

Finally, I present my findings and theoretical perspectives acquired while developing the framework and methods. Optimization as a means of estimating functional weights is especially challenging due to the topology of the parameter space, with small perturbations in weights resulting in drastically different simulated dynamics. High-dimensional spaces are prone to the curse of dimensionality, and network states represented in such spaces are not likely to be stable or typical. Finally, the effects of the concentration of measure, as I believe I've observed when mapping large networks, makes it unlikely that real-world networks have more than about 7 independent functional inputs at any given time.

Chapter 1

Introduction

1.0.1 Motivation

The scale of cellular networks make it particularly challenging to work with, as a large set of properties is required just to describe this scale. Cellular networks start at the scale of cell-cell communication, typically on the order of $0.1\mu m$, where the intracellular space between neurons and glia serves as the communication medium in neuronal networks. Cells are connected through a variety of configurations, from gap junctions between astrocytes to the classical synapses between neurons, forming networks that receive, process, and transmit information. At the upper bound of a few millimeters, large networks of millions of cells form distinct brain regions of specific shapes and functions. Within this 5-order of magnitude scale range, assumptions of constant diffusion, perfect mixing of the molecular level are no longer valid, as the effects of geometric shapes of the multitude of cells making up the neuropil effectively warp the space into a very complex shape. At the other end of the scale, the average effects of large populations of neurons, as measured through either functional MRI or EEG methods, no longer have any meaning since the spatiotemporal pattern of neuronal firing is much more important than the number of neurons firing. A diagram of the cell network scale is shown in figure 1.1.

The cellular scale is important because all information flows through this scale. Fine scale sensory information from individual neurons like vision or touch are integrated to object representations, and large scale motor commands are processed

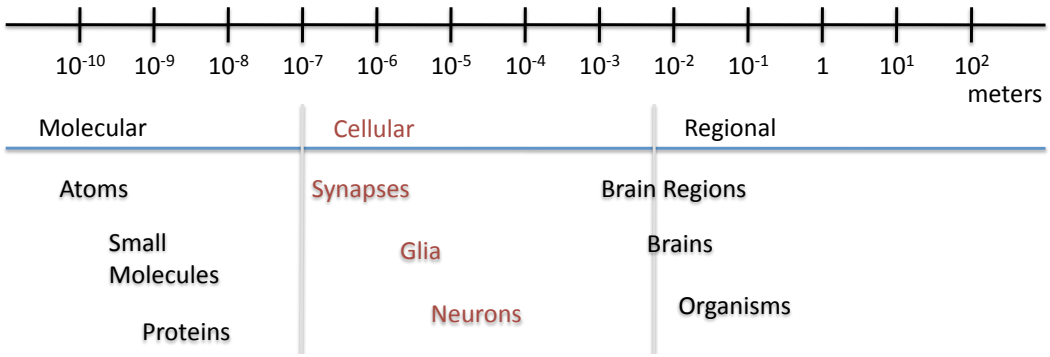

Figure 1.1: Spatial scales of living things. The cellular network scale spans five orders of magnitude, ranging from the resolution of the synapses to the regions of the brain and entire brains. At this scale, there is no correspondence between organisms, as individual neurons, connectivities, and functions are neither numbered nor defined.

into a series of commands sent down to individual muscles with precise strength and timing. In many ways, information can be thought of as flowing up and down spatial and temporal cascades in this scale range, giving rise to the many functions performed by the brain. At this scale there are numerous theoretical challenges ranging from simple description of shape and connectivity to the definitions of function, and characterization of disorders and disease.

To describe this scale in numerical terms poses a significant challenge in itself. If scanned with $0.1\mu m$ voxel resolution, a brain volume of 1 liter would require 10^{18} voxels, which digitized with 1 byte per voxel represents 1 exabyte (10^{18} bytes) of storage. Put in context, 1 exabyte, or 1 million terabytes, is the amount of data that flows through the entire world-wide internet in one day. While other tissues in the human body like lungs or kidneys have similar fine detail, brain tissue data is far more complex and would not be readily compressible or interchangeable. While one can simply move and even remove entire regions of lung or kidney tissue and still maintain function, reshuffling of brain tissue would almost always result in significant loss of function.

Beyond the huge challenges of simply collecting and storing shape information, there is the fundamental problem of labeling and correspondence of individual neurons across different regions or brains for statistical comparisons. For both the

molecular and regional scale, individual molecules and regions are conserved across individuals and even species. Serotonin is the same neurotransmitter across different species and the retina is in the same relative location and performs the same function in virtually all mammals. When something is labeled it can be compared. Labeling individual neurons in a specific network poses some challenges. First there is the problem of where to start numbering and in which direction to increase the count. Second, the number of neurons in a particular region performing the same function will vary from individual to individual, and so will the shapes and configurations of those neurons. Comparing connectivity matrices of two regions in two different individuals is impossible unless there is a correspondence between individual neurons in the two individuals. Even if there was a one-to-one correspondence between individuals, there are likely to be many different connectivities that perform an equivalent information processing function in both networks. This is the problem of degeneracy, also apparent in molecular networks where it greatly complicates the drug discovery process.

Another challenge is how to describe quantitatively the functions performed in these cell networks. At the single cell level, there are many models that describe the dynamics of individual cells: voltage models for neurons [Izh05] and calcium and ATP concentration models for astrocytes [MYBS08]. These models have been built on and validated using experimental data and, while reasonably faithful to the recorded cell dynamics, considerable variability exists between cell types, requiring substantial effort to parameterize new datasets into compact model form using parameter estimation methods [CJA08]. An open problem is how much parameters of these models will vary from cell to cell and individual to individual when operating in a network environment. Scaling up, neurons and astrocytes are connected into network models with different functional weights describing the influences one neuron has on another in the network. The functional connectivity matrix, describing all the connections in a network is unknown for even the smallest creature with neurons, the well-studied, 302-neuron C. Elegans worm. For larger brains, very little is known beyond the scale of single neurons or small circuits consisting of a few neurons. The standing assumption that the dynamical behavior of neurons in circuits is the same as when measured individually remains difficult to prove, yet all simulation

and estimation environments take it for granted.

A common framework and language to describe, interpret, compare, and quantify the shape, connectivity, and dynamics of cellular networks is required for any further progress in research at this scale. The separate disciplines that study morphology, connectivity, and dynamics must be brought together under a common mathematical language so that the constraints imposed by one on the others are taken into account and can be applied to interpret experimental data and further our understanding of network function. Connectivity constraints may be used in establishing bounds on possible geometric arrangements of cellular networks, and particular cell morphology may only produce networks of certain connectivity. The largest unexplored relationship is that between structure and function in networks. Most simulation efforts typically do not use structure information to place constraints on transmission delays, yet we've seen that delays play a major part in the network dynamics, and are present in any biological network [BS11]. While there are many in-silico network simulation environments like NEURON and Genesis, all performing the forward problem of simulating with known parameters, none can take network dynamical data and structure information and estimate parameters from experimental data, given any arbitrary state-space model.

1.0.2 Thesis Summary

The thesis body proper is divided into four main chapters, all aiming at fundamental understanding of cell-network structure and function. All work from chapters has either been published or will soon be published. Chapter 2 presents a novel use of the optical flow technique for measuring, tracing, and labeling calcium activity in cultured astrocytes and hippocampal neurons. It has been published in the Annals of Biomedical Engineering, January 2010 [BYNS10]. It builds on a previous paper I collaborated with M. Hashemi that uses the technique of converging squares to locate cell centers of activity [HBS08] in fluorescence movies. Along with centers, optical flow can be used to trace the patterns of calcium activation in fluorescence movies, providing quantitative and geometric measures of the flow of activity between cells. While applied to cultures, the method has been used for two-photon cultures

and has been shared with other labs seeking similar techniques.

Chapter 3 is a reprint of a major article published in Neural Computation, January 2011 [BS11]. It presents a unified framework for simulation, estimation, and filtering of cellular networks. The paper brings together all neuron and astrocyte individual cell models under a unified mathematical definition, connects them in a geometrically valid, time-delayed network model, and builds a fully parallel architecture for simulation and estimation using graphics processor unit (GPU) cards, for a fraction of the cost and power of conventional CPU clusters. This paper has been well received, and numerous other labs have downloaded the source available on the lab website (http://www.silva.ucsd.edu/Silva_Lab/Downloads.html). Probably the major contribution of this paper is a simplified framework and code base for simulation, which has been well-received and is actively undergoing improvements and enhancements.

Making use of the theoretical framework, I'm completing a short-form paper on mapping functional or effective connectivity in networks. This paper, reprinted in chapter 4, describes the mathematical requirements and a control systems inspired approach to estimating functional connectivity from data using virtually any individual cell model. It highlights the theoretical performance and limitations of functional connectivity mapping using this approach, which to date represents the most general and experimentally relevant method for mapping network function.

Chapter 5 takes a nonparametric approach to the problem of filtering and estimation. Using a completely non-parametric, non-model-based approach, we can estimate applied current from voltage measurements using past combined current-voltage measurements. Similarly, spikes can be estimated from calcium fluorescence, again completely non-parametrically. This approach is superior to deconvolution or particle filter approaches, as it has no dependence on prior models or assumptions. The results, validated on both simulated and actual experimental data, are encouraging, and point to the possibility of estimating connectivity without the need for parametric models of individual cells. This is very much a work in progress, as data is being collected at the time of submission of this thesis.

Finally, chapter 6 is more of an educated speculation on my part on what the architecture of neuronal networks might be. I first describe some of the challenges in

optimization for finding functional weights, which put into question the stability of network models simply made up of individual neuronal models. Additionally, operating in high-dimensional spaces is particularly challenging both for optimization, but for any probabilistic modeling or classification problems. Given these mathematical difficulties, I propose the functional connectivity is relatively low-dimensional for the network to be stable.

Chapter 2

Optical Flow

2.1 Abstract

An optical flow gradient algorithm was applied to spontaneously forming networks of neurons and glia in culture imaged by fluorescence optical microscopy in order to map functional calcium signaling with single pixel resolution. Optical flow estimates the direction and speed of motion of objects in an image between subsequent frames in a recorded digital sequence of images (i.e. a movie). Computed vector field outputs by the algorithm were able to track the spatiotemporal dynamics of calcium signaling pat- terns. We begin by briefly reviewing the mathematics of the optical flow algorithm, and then describe how to solve for the displacement vectors and how to measure their reliability. We then compare computed flow vectors with manually estimated vectors for the progression of a calcium signal recorded from representative astrocyte cultures. Finally, we applied the algorithm to preparations of primary astrocytes and hippocampal neurons and to the rMC-1 Muller glial cell line in order to illustrate the capability of the algorithm for capturing different types of spatiotemporal calcium activity. We discuss the imaging requirements, parameter selection and threshold selection for reliable measurements, and offer perspectives on uses of the vector data.

2.2 Introduction

Calcium signaling is an intermediate step in many of the signaling pathways in neurons and glial cells and is informative of functional neural activity. In neurons calcium signaling precedes sub threshold and threshold (i.e. action potential) changes in membrane voltage, and can be used to infer electrophysiology from optical imaging [SMY99, CDZ06, YF06, VWP$^+$09]. In astrocyte glial cells it underlies the mechanisms by which these cells communicate in astrocyte networks and in bi-directional communication with neurons [APK$^+$07, Ben05, SG06]. Relative changes in cytosolic calcium concentration can be measured using different fluorescence indicator dyes that can be imaged by optical microscopy in the visual light range, such as bulk loaded AM esters and genetically encoded calcium indicators [PEW$^+$08, TL08]. The emitted fluorescence of indicator dyes change as a function of the relative amount of free calcium ions individual indicator molecules are able to interact with. Although the relationship between measured fluorescence signals and the calcium levels that produce them is complex and non-linear, it is assumed that there exists a correlation between measured changes in emitted fluorescence by indicator molecules and differing cytoslic calcium concentrations. In this context, the measured fluorescence signal provides a valuable qualitative metric of changing calcium levels that allow inferences of cell signaling and function. Throughout the rest of this paper, we will use the terms "calcium signal" or "calcium fluorescence" to mean a measured calcium indicator fluorescence signal that reflects a relative cytosolic calcium concentration, as is routinely implied in the literature, even though in practicality we never know the real, i.e. absolute, free ion concentration that gives rise to the measured fluorescence signal.

The data collected by a typical experiment records qualitative movies of imaged changes in calcium fluorescence intensity. One can visualize calcium transients and their relative positions and durations, but there is no inherent quantitative analysis of the data by the experiment itself that allows one to derive the dynamics that characterize such signaling events. For example, things such as propagation speeds and directions (i.e. velocity), the kinetics of measured waveforms, or analysis that depend on such properties, such as identifying and mapping the signaling geometry

of intercellular calcium waves in networks of neurons or astrocytes. Measuring and tracing calcium (or other second messenger) fluorescence signals quantitatively from recorded movies manually is a tedious and labor intensive process for even small data sets, and involves comparing intensities at different frames and locations in order to calculate speeds and directions. It is generally not possible to do so for large data sets that encompass high spatial and temporal resolution detail or large numbers of cells interacting in a circuit or network.

This can be addressed by analyzing experimental data with a filter algorithm called optical flow, which can be used to derive quantitative measurements of observed spatiotemporal calcium signals imaged from fluorescence movies. The resultant vector data has a variety of uses, ranging from deriving basic measurements of signal velocity and direction, to characterizing and classifying spatiotemporal calcium dynamics between different experimental conditions. Optical flow is an imaging technique (i.e. a filter) that calculates a two-dimensional displacement field between two subsequent frames in a movie, based on the local spatial and temporal gradients of the two images. The optical flow filter originated in the computer vision field, where it was designed to approximate object motion in time-ordered image sequences for applications like stereo disparity measurements, motion estimation, movie encoding and compression, and object segmentation [HS81]. The algorithm uses a computed local spatial and temporal gradient to approximate a displacement or flow vector at each pixel in the image. In both neurons and glial cells cytosolic calcium concentration changes manifest themselves as transient responses with a rapid increase, i.e. rising phase, followed by a kinetically slower decaying phase. This is because free calcium is cytotoxic and therefore kept at nanomolar concentrations in the cytoplasm under normal conditions. It is only transiently elevated followed quickly by its re-uptake or extrusion. Temporal changes are typically coupled to spatial changes as a signal propagates through a cell. Measured fluorescence changes then trace specific paths during periods of observation (*c.f.* Fig. 2.1). Calcium transients start at a particular location, travel in some direction at a specific speed and terminate at a different location. The typical kinetics of calcium transients in neural cells are particularly well suited to the computational requirements of the optical flow algorithm.

We have successfully applied optical flow to calcium fluorescence movies and obtained displacement vectors that track the spatiotemporal progression of calcium signals. The filter works for calcium fluorescence data because calcium signals exhibit both spatial and temporal gradients. The computed vectors provide point estimates of the speed and direction of signals. Optical flow is ultimately an imaging filter that works on whole movies, much like edge filters and image segmentation filters are used in static microscopy [GGL+04, MNS09, HBS08], and provides a novel and automated way of analyzing the spatiotemporal dynamics of calcium intracellular signaling in neurons and astrocytes.

We begin by briefly reviewing the mathematics of the optical flow algorithm, describe how to solve for the displacement vectors, and how to measure their reliability. We then compare computed flow vectors with manually estimated vectors for the progression of a calcium signal recorded from representative astrocyte cultures. Finally, we applied the algorithm to preparations of primary astrocytes and hippocampal neurons and to the rMC-1 Muller glial cell line in order to illustrate the capability of the algorithm for capturing different types of spatiotemporal calcium activity. We discuss the imaging requirements, parameter selection and threshold selection for reliable measurements, and offer perspectives on uses of the vector data.

2.3 Optical Flow Algorithm and Computation

In this section we briefly introduce the concepts and mathematics of optical flow, focusing in particular on our own implementation of the algorithm to the experimental data that follows in the Results section. The theory behind the algorithm is well established and the interested reader is referred to a number of excellent texts on the subject (see for example [HS81, Jah03]). Optical flow is an algorithm that operates at the pixel level and calculates local displacement or velocity between time ordered image pairs. Optical flow (or equivalently image flow) is the perceived motion of an object in a field of view (e.g. by the human eye or a camera), defined as the "flow" or change in space and time of gray values at the image plane. It is an estimation of the motion field, which is the actual motion of the object in three dimensional space projected onto the image plane (i.e. what we would like to know).

As long as the frequency of successive frames in an image sequence is shorter than the motion or displacement of the object of interest (in order to avoid confounding ambiguities in detecting the components of the motion caused by aperture and correspondence problems- see [HS81, Jah03]), the optical flow algorithm is able to track the motion of objects in the field of view as a function of changing gray scale levels, subject to appropriate constraints and minimizations. In other words, the algorithm assumes that any changes in gray values are due to the object moving, and that the irradiance of the object is constant from frame to frame. (This is actually a weak assumption that is difficult to satisfy since motion usually causes changes in irradiance, which is why the algorithm is an estimation of the motion field. In cases where irradiance does not change, the optical flow exactly equals the motion field.) The algorithm assumes the conservation of gray levels in the field of view and assumes that any changes in the distribution of gray levels are due to motion. In fact, the optical flow constraint equation (introduced below) can be derived by analogy from the continuity equation in fluid dynamics that conserves mass [BFB94]. By computing the optical flow for all pixels in a field of view, displacement vectors can be calculated for each pixel that map where an object moved to from the pixel in the first frame to that in the second. Intuitively, one can see why the algorithm performs best with objects that have strong contrasts at boundaries or large signal to background noise ratios. The kinetics of calcium transient signals display clearly distinguishable rising and decaying phases that trace specific paths during periods of observation in the form of intracellular calcium waves (Fig. 2.1) that are readily detectable by the algorithm.

The underlying assumption for computation is to constrain local temporal gradients to the product of spatial gradients and displacement vectors. The basic principle of the algorithm takes as inputs two images and computes a vector for each corresponding pixel in the images which approximates the displacement of a small window surrounding that pixel between the two images (Fig. 2.2). Only intensity values inside the window are used for computing the pixel displacement value, so the measurement is localized. Adjacent pixels will have overlapping windows, so their computed vectors will be similar, much like pixels in a blurred image are similar. Following a mathematical description of the algorithm we describe the method for its

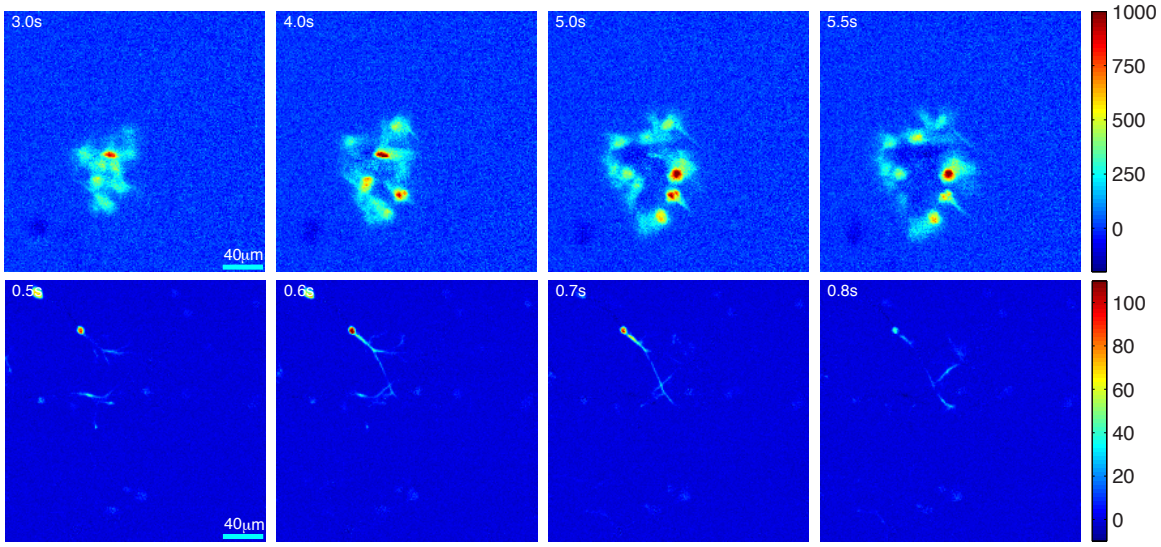

Figure 2.1: Selected frames from recorded movies of imaged calcium fluorescence activity in sparse networks of primary dissociated cortical astrocytes (top) and hippocampal neurons (bottom- where a single neuron at high magnification is shown). The color coded scale bars on the right represent fluorescence intensity I in units of $\Delta I/sec$, as a first derivative of the calcium signal. For neurons the signaling was spontaneous, while for the astrocytes waves were pharmacologically induced. Fluorescence increases followed a relatively smooth spatial progression across the frames at the times indicated by the time stamp in the upper left hand corner of each image. Areas of increasing calcium concentration appear as positive $\Delta I/sec$ values, while areas of decreasing calcium concentration appear as negative values, but at a much smaller magnitudes.

solution and implementation that we used to derive the optical flow for calcium signals. We also discuss parameters and constraints of relevance to calcium fluorescence movies.

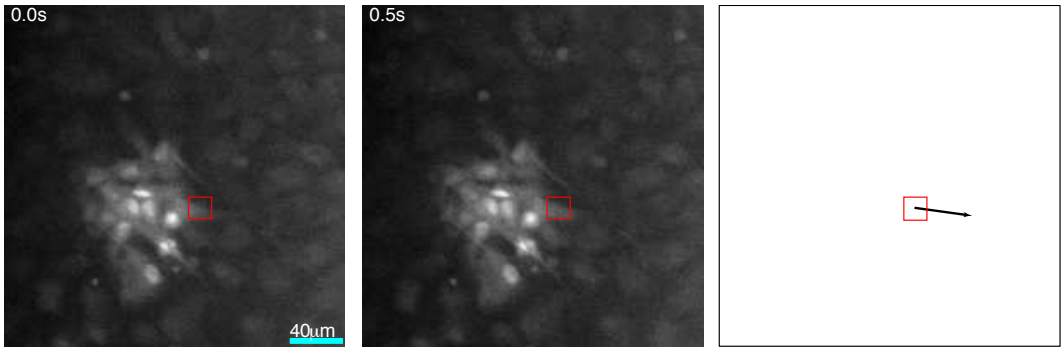

Figure 2.2: The optical flow algorithm. A window Ω in the same location in two subsequent image frames is used to compute a displacement or flow vector (arrow) for the pixel at the center of the window. Only image intensity values in Ω are used for the calculation. Vectors are computed for each pixel in an image frame except in border regions where Ω falls outside of the image. Given the position of the pixel as (x, y) at t seconds, (x, y, t), the displacement vector defines the motion of the pixel at the subsequent frame at δt seconds as $(x + \delta x, y + \delta y, t + \delta t)$.

Consider an arbitrary pixel with gray level intensity $I(x, y, t)$, displaced in the xy plane by δx and δy at time δt in an n x n window Ω (Fig. 2.2b). This implies that

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{2.1}$$

A first order Taylor series approximation of $I(x, y, t)$ by expansion of the right side of 2.1 results in

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x, y, t)+\frac{\partial I}{\partial x}\delta x+\frac{\partial I}{\partial y}\delta y+\frac{\partial I}{\partial t}\delta t+\text{higher order terms} \tag{2.2}$$

Ignoring higher order terms, which provide negligible contributions, and taking into consideration equation 2.1

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0 \tag{2.3}$$

Dividing by δt

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} = 0 \tag{2.4a}$$

$$\frac{\partial I}{\partial x}u_x + \frac{\partial I}{\partial y}u_x + \frac{\partial I}{\partial t} = 0 \tag{2.4b}$$

The two spatial and one temporal gradients are defined by $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$, respectively. $u_x = \frac{\delta x}{\delta t}$ and $v_y = \frac{\delta y}{\delta t}$ represent the x and y spatial components of the optical flow displacement vector $\mathbf{u}(x, y) = (u_x, v_y)$. The basic optical flow formulation is to constrain temporal intensity changes (gradients) to the product of spatial gradients and $\mathbf{u}(x, y)$ to give equation 2.4. In more compact notation this can be written as

$$\nabla I(x, y, t) \cdot \mathbf{u}(x, y) + \frac{\partial I(x, y, t)}{\partial t} = 0 \tag{2.5}$$

Computing optical flow means finding the values of $\mathbf{u}(x, y)$ at each location for every time point that satisfy the above constraint, given the known local image intensity spatial and temporal gradients.

Two factors establish computability of meaningful non-zero flow vector values. First, local spatial gradients must be non-zero at the point of interest (x, y, t). There has to be some image information around the pixel of interest, meaning that neighboring points have to have different values so that gradients are non-zero. If all pixels in a window around (x, y, t) have the same intensity values, then spatial gradients are zero and motion is undetectable by any means. Second, for displacement between subsequent frames to be computed, there has to be a temporal gradient at (x, y, t), or some change in intensity between time points. If there is no temporal change in intensity between subsequent time points, then a value of $\mathbf{u}(x, y) = 0$ satisfies the constraint equation in 2.5. Both of these requirements are limitations on the original application of the optical flow when estimating displacement in natural scenes: objects may have constant intensity in a small window and still be moving, meaning that motion may occur and the recorded intensity spatial and temporal gradients equal zero. These limitations are less important when optical flow is applied to calcium fluorescence movies.

There are many methods for calculating optical flow for recorded movies (see [BFB94] for a review), and all of them work on digitized movies with discrete pixel values of position and time, i.e. $(x, y, t) \in (columns, rows, frames)$. We chose the Lucas and Kanade method because it is conceptually simple and efficient, and flexible in terms of the image processing steps required for computation [BM04, BSL$^+$96, LAG05]. First, computation of the flow vector $\mathbf{u}(x, y)$ is performed on a window or spatial neighborhood Ω of arbitrary size, centered around (x, y), which is more reliable than a single point estimate at (x, y). Second, a window function $W(x, y)$ is defined to favor values in the center over those near the edges. The choice of window size will depend on a variety of factors. It must be large enough to capture the apparent displacement across frames and small enough to resolve features of interest. The capture frame rate must be fast enough for displacements to be observable within the width of the spatial observation window across successive frames. When measuring the spatiotemporal motion of calcium signals the size of the window Ω, the frame rate, and the resolution are all deeply tied to the size of the cells or cellular compartments in which the signal travels. Together, these parameters must be chosen so that the signal is observable and smooth enough to measure reliably as flow vectors across frames. For example, the choice of parameter values used to image calcium signals in part of a dendrite or a fine astrocyte process will necessarily be different than parameter values for broad calcium signals that fill the soma.The constraint equation is redefined as a weighted least-squares fit of local first-order constraints to a constant model of a local $\mathbf{u}(x, y)$ in each small spatial neighborhood Ω around the pixel of interest. The goal is to find the value of $\mathbf{u}(x, y)$ that minimizes

$$\sum_{(x,y)\in\Omega} W^2(x, y)\Big(\nabla I(x, y, t) \cdot \mathbf{u}(x, y) + \frac{\partial I(x, y, t)}{\partial t}\Big)^2 \tag{2.6}$$

The above equation can be rewritten and solved as the linear system:

$$A^T W^2 A \cdot \mathbf{u}(x, y) = A^T W^2 \mathbf{b} \tag{2.7}$$

Where, for neighborhood Ω, consisting of n points centered around the pixel and

time of interest (x, y, t), $\Omega = \{(x_1, y_2, t), (x_2, y_2, t), \ldots, (x_n, y_n, t)\}$:

$$A = \begin{bmatrix} \frac{\partial I}{\partial x}(x_1, y_1, t) & \frac{\partial I}{\partial x}(x_2, y_2, t) & \ldots & \frac{\partial I}{\partial x}(x_n, y_n, t) \\ \\ \frac{\partial I}{\partial y}(x_1, y_1, t) & \frac{\partial I}{\partial y}(x_2, y_2, t) & \ldots & \frac{\partial I}{\partial y}(x_n, y_n, t) \end{bmatrix}^T \tag{2.8}$$

$$W = diag\big[W(x_1, y_1), \ldots, W(x_n, y_n)\big] \tag{2.9}$$

$$\mathbf{b} = -\Big[\frac{\partial I}{\partial t}(x_1, y_1, t), \ldots, \frac{\partial I}{\partial t}(x_n, y_n, t)\Big]^T \tag{2.10}$$

Ω is usually a square window with sizes typically ranging from 3 x 3 to 15 x 15 or $n = 9$ to $n = 225$ points. We have set the weight matrix W to a two dimensional Gaussian with σ^2 equal to 1/6 of the window width. As an example, for a 5 x 5 or $n = 25$ point window:

$$W = \frac{1}{1000} \begin{bmatrix} 1 & 6 & 13 & 6 & 1 \\ 6 & 54 & 112 & 54 & 6 \\ 13 & 112 & 230 & 112 & 13 \\ 6 & 54 & 112 & 54 & 6 \\ 1 & 6 & 13 & 6 & 1 \end{bmatrix}$$

Here, the center values in W have a greater contribution to the calculation than the edge values, favoring gradient values at the pixel of interest.

Solving for the flow vector $\mathbf{u}(x, y)$ in equation 2.7, yields:

$$\mathbf{u}(x, y) = \big[A^T W^2 A\big]^{-1} A^T W^2 \mathbf{b} \tag{2.11}$$

Equation 2.11 describes a linear system in matrix form, where the flow vector $\mathbf{u}$ at spatial and time location (x, y, t) is solved from the quantities of A, W, and $\mathbf{b}$, defined from the spatial and temporal derivates of n points around (x, y, t). The 2×2 matrix $\big[A^T W^2 A\big]$ matrix contains all the image spatial derivatives, and if those values are close to zero, the matrix is poorly conditioned, and flow estimates become unreliable. Ensuring that both eigenvalues of the $\big[A^T W^2 A\big]$ matrix are sufficiently large is a good way to ensure that the matrix is well conditioned, since a measure

of the conditioning number is the ratio of the largest to the smallest eigenvalue [BFB94]. While this is not the only way to compute conditioning, this is the test we used for visualization and measurement reliability of computed vectors for calcium fluorescence data (see Appendix 2.6.3 for more information).

Spatial and temporal derivates were computed using 2 x 2 convolution kernel filters, where the $**$ operator denotes 2-dimensional discrete convolution:

$$\frac{\partial I(x,y,t)}{\partial x} = I(x,y,t) * *\frac{1}{4}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \tag{2.12}$$

$$\frac{\partial I(x,y,t)}{\partial y} = I(x,y,t) * *\frac{1}{4}\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \tag{2.13}$$

$$\frac{\partial I(x,y,t)}{\partial t} = \frac{1}{\Delta t}\big(I(x,y,t+\Delta t) - I(x,y,t)\big) * *\frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{2.14}$$

Here Δt represents the time between frames or the frame rate $1/\Delta t$. Since the temporal derivative calculated in 2.14 forms the basis for the $\mathbf{b}$ vector in 2.11, the frame rate has a linear effect on the magnitude of the flow vector $\mathbf{u}$.

Optical flow outputs a displacement vector in units of pixels, normalized to the time difference between the two frames used for computation. When normalized, the vector takes on velocity units of pixels per frame (for this reason it is called a flow vector). The conversion to physical units will depend on the spatial resolution of the camera and microscope, typically expressed in microns per pixel, and the sampling rate for the movie capture, expressed in frames per second. Spatial resolution is a function of the objectives used as well as the resolution of the imager and any pixel binning used. The frame rate is limited at the high end by the camera sampling rate, and at the low end by the minimum exposure time required to capture a detectable intensity signal. The exposure time may be reduced by increased gain or pixel binning, but those come at a cost of reduced resolution or increased noise. The conversion between units of pixels/frame and units of microns/second is straightforward:

$$\frac{microns}{second} = \frac{pixels}{frame} \cdot \frac{frames}{second} \cdot \frac{microns}{pixel} \tag{2.15}$$

While the optical flow algorithm produces vectors in units of pixels/frame, the analysis of the data in the Results section below have been converted into physical units of microns/second, using the resolution and capture rate of the recordings given the specifics of our imaging system.

2.4 Results

2.4.1 Comparison between computed and manually estimated flow vectors

We manually estimated flow vectors for 12 images equivalent to 6 seconds of calcium signaling in primary dissociated spinal cord astrocyte cultures (orange arrows in Fig. 2.3), and qualitatively compared them to computed optical flow vectors for the same data (green arrows in Fig. 2.3; note that only reliable vectors are shown as determined by the eigenvalue test- *c.f.* equation 2.11, see above). Manual estimation required stepping through frames and approximating roughly how a calcium signal progressed in time, which in this experimental preparation included intercellular calcium waves that propagated through a subset of the cell network. The manually traced signals were not the only ones observable in the small movie sequence used, but were chosen to illustrate four representative signaling paths. Manual estimation was performed in two second intervals, estimating the incremental spatial progression of a given calcium signal across four frames.

Estimation of the flow or displacement of a cell signal such as calcium between frames manually like we did for the data in Fig. 2.3 is a very tedious and labor intensive process, and can only realistically be done under very sparse conditions where the observer can clearly delineate the flow of the signal visually. It is nearly impossible to do at the pixel or small window level. In contrast, optical flow calculates a displacement vector for every pixel in every frame, operating at a much finer scale and capturing much more detail than is possible with manual estimates. Nonetheless, in Fig. 2.3 for the purpose of qualitatively validating derived optical flow vectors to manually estimated ones, in both cases there was a clear overlap in vector direction between manual and flow vectors. By contrast, there were greater differences in

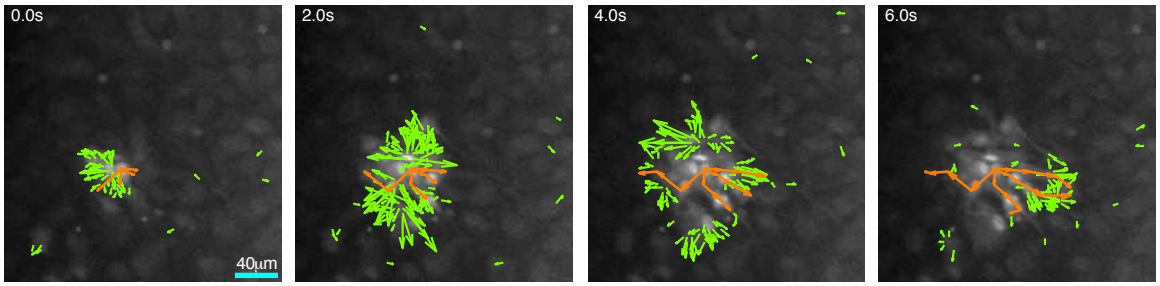

Figure 2.3: Comparison between computed optical flow vectors (green) and manually estimated flow vectors (orange). While computed vectors were calculated for every pixel and every frame, manual vectors were estimated every four frames and only trace a few selected signals. Only reliable optical flow vectors are shown, and only one in four vectors in both horizontal and vertical directions are shown for clarity. Unlike the manual vectors, flow vectors are only shown for the current frame. The top sequence of panels show vectors overlaid on extracted frames from the actual movie at the indicated times for the entire field of view. The bottom panels show the vectors in detail for the 2, 4, and 6 second frames in order to more clearly assess the qualitative overlap between optical flow computed and manually estimated results. For optical flow vectors (in green) only vectors that putatively correspond to manual vectors (in orange) are shown, in contrast to the upper panels which show all computed vectors (see text). See appendix for details regarding experimental preparations, imaging, and parameters for calculation.

vector magnitudes between manually and flow vectors, which is consistent with the fact that the manual estimates spanned four frames while optical flow vectors were calculated across adjacent frames. There is also temporal overlap between the two cases, in the sense that similar displacements were estimated for the same frames using both approaches. The eigenvalue threshold masked out unreliable vectors, and this correlated well with calcium activity; only areas of spatial and temporal changes in the movie produced reliable vectors as assessed visually, which is ultimately the most accurate estimator of complex motions, but only if given the right conditions (e.g. conditions that allow the human eye to separate motion). The optical flow algorithm however, is able to provide reliable quantitative measurements of signaling dynamics at spatial and temporal scales simply not measurable by qualitative visual inspection or manual estimations of the data.

2.4.2 Optical flow characterization of intercellular signaling

We applied the optical flow algorithm to typical calcium fluorescence movies of spontaneously forming sparse networks of neural glial cells and neurons in culture, and looked at the dynamics of intercellular signaling following pharmacological or mechanical stimulation. We purposely chose sparse networks because it facilitates the visual interpretation of the entire resultant vector field, but the algorithm itself can operate on any data that displays an appropriate signal. We recorded movies from the rMC-1 Muller glial like cell line, which mechanistically displays calcium signaling similar to Muller retinal glial cells *in vivo* [YBC$^+$09], primary dissociated spinal cord astrocytes, and primary dissociated hippocampal neurons. Intercellular calcium waves in rMC-1 cells and astrocytes were mechanically induced by gently poking an initial cell without penetrating the cell membrane, while calcium waves in neuronal networks were induced by the localized pharmacological application of glutamate to one or a small group of cells (see the appendix below for details about experimental preparations and imaging parameters). In particular, intercellular calcium waves in astrocytes and related anatomically specialized macroglial cells such as Muller cells in the neural retina or Bergman glia in the cerebellum have been known to occur under experimental conditions for several years now, and have recently been shown *in vivo*

under both physiological and pathophysiological conditions in different parts of the brain, mediated by intracellular calcium transients that induce paracrine signaling, primarily through adenosine triphosphate (ATP) [KLHB09, KNMN09, HKG+09]. Astrocyte and related macroglial cells engage in bi-directional chemical signaling with neurons and have the ability to modulate and directly participate in information processing in the brain, which necessitates more than just interactions between neurons and almost certainly involves astrocytes somehow. The functional roles of glial intercellular calcium waves and their contributions to modulating neuronal information are not yet known, and in fact the dynamics of these signaling events and the conditions under which they occur are just beginning to be explored.

The key parameter for computing optical flow using the Lucas-Kanade method is the window size Ω, specified as a square of a given width (see above). It defines the local neighborhood of pixels along a point of interest that is used to compute the spatial and temporal gradients required for the calculation. Though not required for computation, a minimum value for the eigenvalues for the matrix $A^T W^2 A$ should be specified to mask out unreliable measurements. This ensures that only reliable displacement vectors are displayed and used for analysis. Since the intensity values are a function of the experimental setup, microscope, and camera, the $A^T W^2 A$ matrix and its eigenvalues will scale accordingly. The selection of the eigenvalue threshold is thus arbitrary, much like the selections of the camera gain, exposure time, and other imaging parameters are made to generate easily visible intensity values (see Appendix 2.6.3 for more information on selecting suitable eigenvalue thresholds). Table 2.1 shows the window sizes, eigenvalue thresholds, and capture frame rates used to calculate the vector fields shown in Fig 2.4. The displacement vectors can be converted into velocity by equation 2.15. The original calcium fluorescence movies and Matlab code written to implement the optical flow algorithm are available online as supplemental materials.

Neuronal cultures displayed derived optical flow vectors along processes as the calcium signal propagated throughout the network. As expected, computed vectors and the resultant vector field followed the geometry of connected processes (i.e. axons and dendrites) in the sparse network (Fig. 2.4a). The pattern of activation in this example proceeded diagonally from the site of stimulation in the upper left

Table 2.1: Image capture and optical flow parameters for shown figures. The values were chosen manually through manual iteration and visual comparison.

Parameter	rMC-1 Cells	Astrocytes	Hippocampal Neurons
Frame Capture Rate (Hz)	16.4	8	4
Window Size (pixels at 1.3μm/pixel)	11	9	11
Minimum Eigenvalue - (λ_1, λ_2) greater than	11	1.4	.3

hand corner of the field of view. Some neurons activated at considerably longer times following the stimulus (i.e. out to 7 or 8 seconds) most likely due to recurrent feedback signaling in the network which can last several seconds. Note how since only reliable vectors are plotted, as determined by the eigenvalue test, there are spatial discontinuities in the temporal progression of mapped signals, which reflect areas where the algorithm could not compute reliable vectors given the measured data. This may be especially true at lower magnifications as in the example shown here for comparatively large fields of view that capture many cells. This represents a challenging task for the algorithm. Nonetheless, both the spatial and temporal progression of calcium signals are easily visible. The computed data, being in vector form, can complement existing methods like cross-correlation that use only cell body data to establish relationships between cells for example.

Signal flow patterns were also computed for astrocyte and rMC-1 glial networks (Fig. 2.4b and c, respectively). Astrocyte signaling showed rapid burst like radial patterns that was mostly complete by 2 seconds, with some smaller regions of cells activating later as far out as 6-7 seconds. This is consistent with descriptions of intercellular calcium waves reported previously [CBFCS90, NZ97, AB05, SG06]. rMC-1 cells showed qualitatively similar radial patterns of activation, with signaling occurring within about 3 seconds following stimulation. However, unlike the astrocyte response, where there was uniform signaling across the network near the site of stimulation, rMC-1 cells showed more heterogeneity in spatial activation patterns,

with distinct clusters of cells activating and spreading calcium waves. The distances traveled by the waves in the rMC-1 example roughly agree with previous quantitative characterizations of calcium waves in similar preparations, on average displaying wave distances of about 60 μm over the first 2 seconds or so and distances between 50-100 μm over about 4 seconds [YBC^{+}09]. It is interesting however that the spatial progression of the calcium signals in this example was not linear as a function of time, in the sense that cells roughly equidistant from the site of stimulation activated at different times, within about 1 second for some versus about 3 seconds for others. The relationship and dynamics between the spatial versus temporal properties of such waves are difficult at best and usually not possible to determine by visual inspection of recorded movies alone, and are not captured by calculations such as the one dimensional signaling speed of a progressing wave front. Furthermore, speed and distance calculations of neuronal and glial signaling across networks of cells are usually corse approximations computed using low magnification movies that provide a sufficiently large field of view. In contrast, optical flow provides reliable single pixel vectors for any sized region of interest that represent very fine grain detailed descriptions of calcium signal propagation difficult to achieve otherwise. For the astrocyte data from Fig. 2.4b, Fig. 2.5 illustrates the distribution of signaling speeds in μm/second for 65 optical flow vectors for a small 10 x 10 pixel region equivalent to a 13 x 13 μm region in the field of view (orange box in the figure). Any region size of interest anywhere in the imaged field that might be of functional interest to the investigator can be similarly characterized. By way of rough comparison, optical flow calculated speeds for calcium signals in computed window were distributed from 1-10 μm/sec, and are roughly similar to those reported previously using more approximate and global methods, in the range of 5 to 10 μm/second [CBFCS90, NZ97, AB05, SG06]. The bimodal distribution in the figure reflects what is visually apparent in the source movie: some of the areas in the orange pixel region exhibit spatiotemporal displacement while others do not, indicating that calcium concentration changes propagate in specific regions with specific patterns. Manual estimates from the literature typically look at maximum propagation speeds, as seen in the second peak at about 9 μm/second.

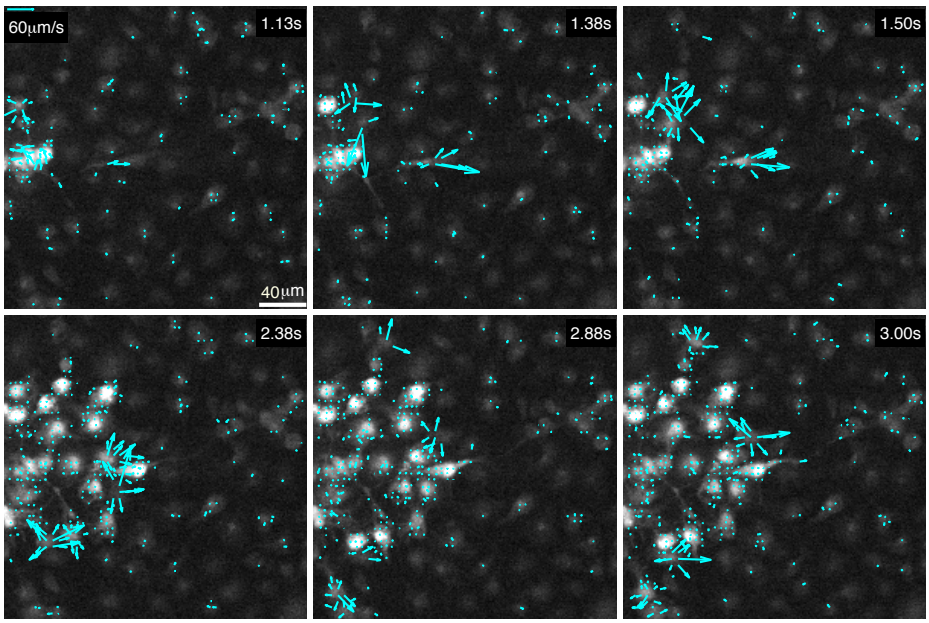

Figure 2.4: Computed optical flow vectors for induced calcium signals in spontaneously forming *in vitro* networks of (a) primary hippocampal neurons, (b) primary spinal cord astrocytes, and (c) the rMC-1 Muller glial-like cell line. Six frames from each representative recorded movie are shown with the computed vector field superimposed at times indicated by the time stamps in each frame (left set of six panels). Right panels: Composite temporal projections of the entire movies. The vector fields show the full spatial progression for the evolving calcium signals, with time (i.e. temporal progression) color coded by the color map (in seconds). Plotting the vector fields in this way allows the full spatiotemporal propagation of derived signals from entire movies to be summarized in a single image. This facilitates the qualitative visualization and identification of complex dynamic signaling patterns that would be difficult to detect otherwise, such as for example by simply "playing back" the movie.

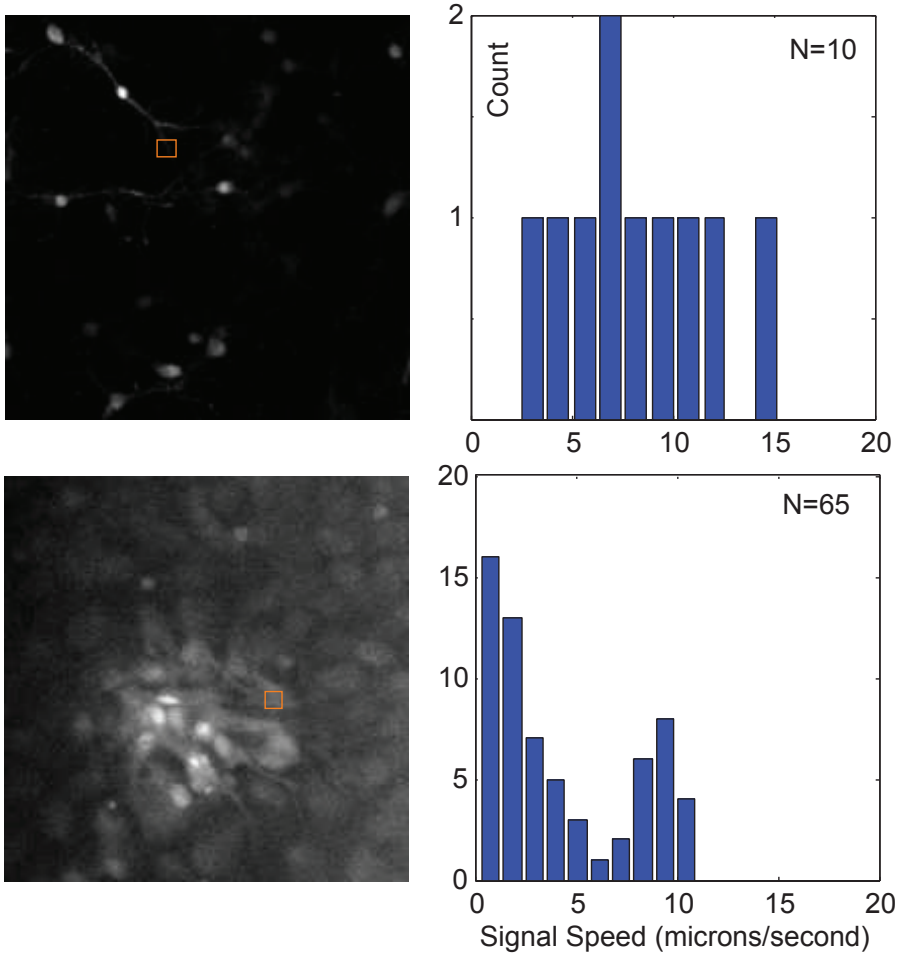

Figure 2.5: Optical flow velocity magnitude distributions for the astrocyte data from Fig. 2.4b. The flow vector magnitudes for reliable measurements in a 10 x 10 pixels (13 x 13 μm) region (orange square) shown as histograms.

2.5 Discussion

We describe and show the application of optical flow gradient methods for identifying and spatiotemporally mapping functional calcium signaling in networks of neurons and glia. Although we focused on networks of cells here, the method can be equally applied to the analysis of spatially detailed sub-cellular compartmentalized regions of interest, such as dendrites or astrocyte processes. The method makes use of the spatial first derivative of moving objects in a field of view, in this case changes in fluorescence levels of calcium indicator dyes associated with the free concentration of intracellular calcium, to track their motion between subsequent frames in an image sequence (i.e. a recorded movie). The mathematical foundations of optical flow are well established and optical flow algorithms have been used in a wide variety of fields including applications to cell and molecular biology to track the movement of proteins, vesicles, and even whole cells [Miu05]. In neuroscience and neural engineering it has been used in electromyography [KPW$^+$02, OGK07] and sensory perception [LA07, PB98], while clinically, it has been used to detect seizures in neonatal infants [KTF$^+$06], among other applications. However, the method has not been previously applied to tracking and visualizing calcium signaling and deriving quantitative measurements of calcium spatiotemporal changes that underlie intracellular and intercellular functional signaling in neural cells.

Although in this paper we applied the optical flow algorithm to two dimensional fluorescence movies, the algorithm itself can be readily applied to a recorded movie made up of three dimensional stacks acquired using two-photon microscopy. Work by others is pushing two photon imaging towards recording real time functional signaling from three dimensional volumes of active cellular neural networks [GKH07, GH07]. If the sampling rate is sufficiently high, optical flow can be computed in three dimensions using a volume instead of a square window around a pixel to generate a three dimensional displacement vector. The same constraints on volume size, sampling, and vector reliability metrics in two dimensions apply to the three dimensional case.

Optical flow methods produce a lot of data, generating a vector for every pixel in every image pair computed, so further processing, rendering and visual-

ization methods are key to making quantitative comparisons between experimental setups. Statistical comparisons can be made from vector values by comparing differences between selected regions in different preparations; velocity averages for each region can be compared using statistical methods such as means, standard deviations, and p-values. While vector values from adjacent pixels are not statistically independent, averaged vector values for a given region of interest may be used for statistical comparison with another, non-overlapping region.

Another potential use of the vectors is to classify spatiotemporal patterns. Similar to using a scalar kernel filter to match an image pattern such as an edge or corner, vector fields themselves can be filtered with a known vector kernel to match a pattern of interest. This method is called Clifford convolution [ES03] and has been used to label physical flow regimes in fluid dynamics applications. By designing a vector field filter and convolving it with computed optical flow vectors, a scalar map identifying specific patterns of flow associated with the saptiotemporal dynamics of the measured signal can be constructed in order to classify regions exhibiting such patterns.

One of the most exciting potential uses of computed flow vectors is in functional network reconstruction. Borrowing again from the field of fluid mechanics, a dynamic vector field can be used to reconstruct the path of a hypothetical particle from a given starting point, tracing out the path that a signal might take between cells, much like a particle in a dynamic flow field [WE05, WSEE05]. Geometrically mapped paths of measured signals that originate in an activating cell and propagate through a network may be very useful for reconstructing the dynamics of the network. This would complement existing network reconstruction algorithms which typically rely on temporal data around fixed regions of interest.

2.6 Appendix

2.6.1 Cell Preparations

rMC-1 glial cells and primary spinal cord astrocyte cultures (the latter dissected and grown similar to previously described [SFMT98, MYBS08] were grown to

approximately 80% confluency and washed twice with Kreb-HEPES buffer (KHB) solution (10 mM HEPES, 4.2 mM $NaHCO_3$, 10 mM glucose, 1.18 mM $MgSO_4$, 7H2O, 1.18 mM KH_2PO_4, 4.69 mM KCL, 118 mM NaCl, 1.29 mM $CaCl_2$, pH 7.4) and incubated with 5μM Fluo-4AM in KHB for 1 hr at room temperature. Excess dye was removed by washing twice with KHB and an additional incubation of 30 min at room temperature was done to equilibrate intracellular dye concentration and ensure complete intracellular hydrolysis. Synchronized calcium transients were initiated by mechanical stimulation of a single cell using a (0.5μm i.d.) micropipette tip (WPI Inc., Sarasota FL) mounted on a M325 Micrometer Slide Micromanipulator (WPI Inc., Sarasota FL). Comparable data were obtained using adenosine triphosphate (ATP) pharmacological stimulation.

For hippocampal cultures, dissociated hippocampal neurons from timed pregnant embryonic day 18 (E18) Sprague-Dawley rats were cultured on glass bottomed tissue culture dishes coated with poly-D-lysine and laminin (BD Biosciences, San Jose, CA). Cultures were plated at a cell density of 10^6 cells/3.8cm^2. Cultures were maintained at 37C in 5% ambient CO_2. Plating media was composed of basal medial Eagle (Invitrogen, Carlsbad, CA) with 1X Glutamax, 1000 U/mL penicillin and streptomycin sulfate, 5% FBS, and 1X N2 supplement. Culture media consisted of Neurobasal (Invitrogen, Carlsbad, CA) with 1X Glutamax, 1000 U/mL penicillin and streptomycin sulfate, 20mM glucose, and 1X B27 supplement. Culture media was supplemented with 10uM Ara-C for 24 hrs at 1DIV to inhibit overgrowth of glia. All imaging was performed on 3-5DIV.

Bulk loading of hippocampal cell cultures was accomplished via incubation in the dark, at room temperature, for 30 min in 1μM of the fluorescent Calcium indicator Fluo-4-AM in Krebs-HEPES buffer (10mM HEPES, 4.2 mM $NaHCO_3$, 10mM glucose, 1.18mM $MgSO_4{\cdot}7H_2O$, 1.18mM KH_2PO_4, 4.69 mM KCl, 118mM NaCl, 1.29 mM $CaCl_2$, pH 7.4), followed by 2x 5 min washes in Krebs-HEPES with 100μM sulfinpyrazone. Hydrolysis was allowed to proceed for an additional 30 min. Stimulation of neurons with glucose was performed by microinjection of 100uL of 10mM glutamate in PBS from a specified-side of the culture dish, well outside of the microscope field of view. The fluorescence signal generated across the monolayer of cells was recorded for 10 sec prior to glutamate injection, and for 120 sec following

injection. Cultured neurons were incubated for 30 min prior to imaging in Mg_{2+}-free PBS to induce the synchronization of calcium transients.

2.6.2 Imaging Setup

Visualization of calcium indicator dye fluorescence was achieved using a 488 nm (FITC) filter on an Olympus IX81 inverted fluorescence confocal microscope (Olympus Optical, Tokyo, Japan) that included epifluoresence, confocal, phase, brightfield, and Hoffman differential interference contrast (DIC) modalities. Real-time movie recordings of calcium transient propagation were acquired with a Hamamatsu ORCA-ER digital camera (Hamamatsu Photonics K.K., Hamamatsu City, Japan) and Image-Pro Plus data acquisition and morphometric software (version 5.1.0.20, Media Cybernetics, Inc., Silver Spring, MD) or LabView custom written data acquisition software (ScopeController). All images were captured with a 10X objective, using a 2x2 binning on the camera, for a resolution on 1.3μm/pixel and a total image size of 612x572 (camera's maximum resolution is 1224x1144). Images sampled at frequencies ranging from 2 to 16.4Hz, or 0.5sec to 0.06sec exposure time.

2.6.3 Reliable Vectors via the Eigenvalue Test

Recall that flow vectors are computed from the linear system in 2.11. This is a typical linear system of the form:

$$M \cdot \mathbf{u} = \mathbf{z}$$

where $\mathbf{u}$ is the unknown and $\mathbf{z}$ and M are known quantities. The condition number of a matrix simply describes how a small deviation in the known $\mathbf{z}$ translates to an error in $\mathbf{u}$. A high condition number means the matrix is ill-conditioned, meaning that a small deviation in $\mathbf{z}$ leads to a large deviation in $\mathbf{u}$, making that computation unreliable. One way to compute the condition number of a matrix is to take the ratio of the largest to smallest eigenvalue of that matrix:

$$\kappa(M) = \left| \frac{\lambda_{max}(M)}{\lambda_{min}(M)} \right|$$

Since M is a 2×2 matrix, it has 2 eigenvalues so ensuring that both are above a certain value makes the condition κ value relatively low. The minimum threshold value depends on the incoming intensity values.

Intensity readings from the CCD camera can take on any number of values, based on the digitization (8-bit, 12-bit, 16-bit, for example), the exposure time, gain setting on the camera, and above all the dye loading in the cell preparation. Typically, during observation, the experimenter manually adjusts gain and exposure time to obtain reasonable intensity values, typically in the middle of the digitization range.

Eigenvalues for the $A^T W A$ used in flow vector calculation typically scale with the range of recorded intensity values and are calculated for every pixel, producing an eigenvalue image map. The values chosen in table 2.1 were manually chosen during examination of the minimum eigenvalue image for a few representative frames, ensuring that they fell between ares where we visually detected spatiotemporal changes in intensity and areas were we did not detect such changes. This is the same process one would undertake when thresholding a regular monochrome image for the counting of cells: the intensity threshold is set to a value between the intensity of an area where there is a cell and an area where there is no cell.

Chapter 2, in full, is a reprint of the material as it appears in Buibas M, Yu D, Nizar K, and Silva GA. "Mapping the Spatiotemporal Dynamics of Calcium Signaling in Neural Networks Using Optical Flow" *Annals of Biomedical Engineering.*, 10.1007, 2010. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Network Mapping Framework

3.1 Abstract

We present a framework for simulating signal propagation in geometric networks (i.e. networks that can be mapped to geometric graphs in some space) and for developing algorithms that estimate (i.e. map) the state and functional topology of complex dynamic geometric networks. Within the framework we define the key features typically present in such networks and of particular relevance to biological cellular neural networks: Dynamics, signaling, observation, and control. The framework is particularly well-suited for estimating functional connectivity in cellular neural networks from experimentally observable data, and has been implemented using graphics processing unit (GPU) high performance computing. Computationally, the framework can simulate cellular network signaling close to or faster than real time. We further propose a standard test set of networks to measure performance and compare different mapping algorithms.

3.2 Introduction

Complex dynamic networks permeate many real world engineering and biological applications. The development of mathematical and computational tools for understanding and predicting network dynamics will be key to manipulating and interacting with such real world networks. Network theory is a subset of graph the-

ory where the connections between vertices have a number value describing some attribute of that connection, such as for example bandwidth, flow rates, or a cost function. Complex networks are defined to have a non-standard topology, i.e. the functional links between nodes in the network, implying some structure in the connectivity pattern of the network beyond a simple lattice or complete random connectivity. Biological cellular neural networks are both complex and dynamic, meaning that the connection attribute between any two vertices may change with respect to time and, more importantly, individual vertices exhibit their own nonlinear signaling dynamics. Complex functional interactions of networks made up of large numbers of neurons and glia produce emergent systems-level phenomena such as consciousness and self-awareness, and are responsible for how neural information is represented and processed. Changes in the structure of such networks presumably underlie the development of multidimensional central nervous system disorders. For example, hypersynchronous neuronal and glial activity in networks of neurons are associated with the paroxysmal depolarization shifts that underlie epilepsy [TAT$^+$05, SA07, WSD08, FOM$^+$07]. Ultimately, the physiologic behavior of a neural cell network is dependent on both its functional topology and the dynamics of individual cells.

Within a complex dynamic network there are two topologies. A static, structural topology that describes all the possible connections within the network, and a dynamic, functional topology that establishes how a signal propagates through the static topology. Functional topologies are subsets of the structural topology and vary depending on the functional connectivity, internal dynamics of individual vertices, and the specific stimulus to the network. While this is the case for biological neural networks, where cells that are physically connected need not necessarily signal each other, in cellular neural circuits and networks structure and function influence each other and the states of cells and the connections between them may change with time as a function of plasticity mechanisms. However, structural changes in the physical connectivity of a cellular neural network leading to changes in the connectivity topology occur on a very different time scale then functional changes that can be influenced relatively quickly by plasticity mechanisms that produce changes in signaling efficacy between cells (i.e. changes in connectivity weights). While the

observation of the structural network topology of cellular neural networks may be experimentally very challenging (and indeed is the focus of much intense research), it is a relatively straightforward task. The observation of functional topologies in biological neural networks however poses additional experimental and theoretical challenges that need to be considered. Signaling events and resultant networks may be unique and be observable only once as a signal propagates through a network. The functional topology is dynamic and may change during observation. Noise and unknown external factors limit observability and reduce repeatability. These factors make the estimation of functional connectivity from observed activity a difficult task, though a critical one for systems neuroscience if we are to understand how dynamic functional signaling in the brain at the level of networks and circuits produces responses and behaviors in the organism.

Current approaches for studying cellular neural networks can be roughly classified into three categories. The first and most popular amongst experimentalists are statistical methods that correlate the activities of two or more neurons in a network. This provides purely descriptive statistics about the behavior of cells. For the most part, statistical approaches make no underlying assumptions about the cellular and systems dynamics that give rise to observed signals in a network of cells. Another way to study networks is through simulation of networks with known connectivities and dynamic parameters in order to simulate real-world observed system level phenomena such as vision and audition. Using well established environments like NEURON or Genesis, many real-world phenomena have been described through simulation. However, dynamic parameters and functional connections are manually specified in simulation environments such as these in order to achieve results that mimic biological function, requiring the estimation of experimentally unobservable variables. The third category is in some ways the reverse process to simulation, where temporal data is used with appropriate models in order to estimate parameters. Within this third category, we introduce a modeling framework for using real-world data to map the functional topology of complex dynamic networks. While not a mapping algorithm or simulation environment, the framework formally defines key features of cellular neural network signaling and experimental constraints associated with observation and stimulus control, and can accommodate any appropriate model of intracellular

dynamics. Alongside the definition of the framework, a test set of synthetic networks with known connectivities is provided to help the development of mapping algorithms by providing a common benchmark any such algorithm should be able to map. In a subsequent paper to this one we will introduce an approach that will estimate and map the functional topology of complex networks with unknown connectivities given limited and often noisy observations that takes advantage of the results introduced here.

The proposed framework has a number of unique properties that makes it particularly applicable to the constraints and experimental limitations imposed by real biological cellular neural networks. First, dynamic activity and signaling is modeled at the individual node (i.e. cell) scale. The dynamics of individual cells are modeled as state sets, with transition functions describing their evolution across discrete time steps. Cellular resolution was chosen because it represents the best compromise between observability, dynamics, and complexity. Large numbers of individual cells can now be observed in parallel in functional neural networks using optical microscopy [HBJ$^+$09, BHP08, GMG$^+$06, NPY07]. Single cell neuronal dynamics are well understood and many models exist (see for example [DA09, Tra10]), while similar models of single cell astrocyte dynamics are beginning to emerge [SM06, NJL08, PVLBJ09, Ben05, LH08]. Attempting to go to a finer, sub-cellular compartmental resolution dramatically increases the complexity of the model, computational demand, and is generally not experimentally observable at a network level. Secondly, cells are located in physical space and their positions are easily determinable during experimental observation. When connected cellular networks form geometric networks. Thirdly, the effect of a signal on a target cell is defined as a state change in the target cell in response to the influence of a source cell that connects to it. That influence is not instantaneous, and is delayed by the physical distance between cells and the speed of transmission. Signals are modulated in strength by functional weights, which establish the magnitude of the influence. Fourthly, to more realistically simulate experimental conditions and measurements, noise can be added to multiple levels within the framework, from parameters to state and observation variables. Finally, experimental user-defined controls at the individual cell level are defined within the framework. Controls should be designed to make observations

more informative of the network dynamics, but should not change the underlying parameters and connectivities. The framework is described in detail in section 3.3. The results section (section 3.4) shows how single cell dynamic models are integrated within the framework (3.4.1), and how network connectivity is established from individual cells (3.4.2). We also describe how the framework accommodates plasticity mechanisms (3.4.2) and experimental observability associated with optical calcium imaging (3.4.3). Section 3.4.4 discusses the practical implementation of the framework using high performance graphical processing unit (GPU) computing.

In section 3.4.5 we use the framework to propose a standard set of benchmark test networks of varying sizes and topologies to evaluate and compare different network mapping algorithms. Mapping algorithms would have access to simulated observable data (i.e. simulated experimental data) generated by the framework as a function of chosen test networks and be required to derive the unobservable parameters and functional network connectivity. The concept of a standardized test to gauge the effectiveness of an algorithm is not new, especially for optimization algorithms. For example, in the field of nonlinear programming and optimization a standard benchmark set was established in a landmark collection of test problems [HS80] that are used for testing any nonlinear optimization algorithm. Test collections have grown and developed into problem environments, providing the underlying problem code to be used directly by the optimizers [BBM99, GOU95]). By providing a set of problems with known solutions, algorithm developers have a standard by which to measure solution accuracies, convergence rates, computation times, and suitability to different problem types. We propose that a similar test set for algorithms designed to identify and map functional cellular neural networks and circuits will be just as useful. To address this, we have developed computer code that generates observable data from a known network and connectivity. The code encompasses all the elements of the framework, runs in real time for all the test networks, and is designed for parallel computation, and can therefore be used as a starting point for mapping algorithms.

3.3 A Framework for Dynamics, Signaling, Control and Observation in Geometric Networks

We develop the proposed framework using standard graph theoretic and set theoretic concepts and terminology. In the most general sense, a network is a type of graph. A graph is defined as an ordered pair of finite disjoint sets (v, E) such that $\mathbf{v}$ is the set of J vertices of G and $\mathbf{E}$ is the set of edges of G, i.e. $\mathbf{v} = v(G)$ is the vertex set of G while $\mathbf{E} = E(G)$ is the edge set of G. An edge e_{ij} is defined if there is a directed connection from vertex i to vertex j. Geometric graphs are graphs where the relative positions of vertices are assigned coordinates in some geometric space. While this is the most generic description of a graph, dynamic geometric networks as we use the term here are more specialized cases of generalized geometric graphs defined as follows. Vertices in a network have two attributes, a known and static position in physical Cartesian space denoted by $\mathbf{x}_j$ for a given vertex j and a time-variant state set $\mathbf{y}_j(t)$ of K_j state variables:

$$\mathbf{y}_j(t) = \{y_{1,j}(t), y_{2,j}(t), \ldots, y_{K_j,j}(t)\} \tag{3.1a}$$

such that formally

$$\mathbf{y}_j(t) = \{y_{k,j}(t) : k \in \mathbb{N}, k \leq K_j\} \text{ for any given vertex } j \tag{3.1b}$$

Next, for all vertices i other than j, let the set $\mathbf{Y}_j(t)$ be the union of all i, i.e. the collection of states of all vertices in the network excluding vertex j, weighted and delayed relative to vertex j, in the sense that every vertex i has the potential to pass information (e.g. a signal) to vertex j with varying amounts of 'influence' as determined by a collection of weights that modulate any directed edges from i to j. Furthermore, such information will be delayed by some finite time as a function of the geometric position of vertex i in the network relative to j and the finite speed of information propagation. We define

$$\mathbf{Y}_j(t) = \cup_{i \in \mathbb{N}; i \leq J; i \neq j} \Omega_{ij}(t) \cdot \mathbf{y}_i(t - \tau_{ij})\} \tag{3.2}$$

where with out loss of generality we define

$$\Omega_{ij}(t) = [\omega_{1,ij}(t), \omega_{2,ij}(t), \ldots, \omega_{K,ij}(t)] \tag{3.3}$$

and restrict (3.1b) for vertex i with temporal delays as vector sets, i.e.

$$\mathbf{y}_i(t) = [y_{k,i}(t) : k \in \mathbb{N}, k \leq K_i] \text{ for any given vertex } i \neq j \tag{3.4}$$

The delays τ_{ij} are non-negative values representing the delay of information passing from i to j. In all cases, here and below we adopt the convention that indexing subscripts given by 'ij' enumerate the variable that uses the subscript as linking vertex pairs i and j.

We then define a transition function $\mathbf{H}_j(\cdot)$ with parameter set Θ_j that describes the temporal progression or evolution of $\mathbf{y}_j(t)$ in discrete time increments Δt:

$$\mathbf{y}_j(t + \Delta t) = \mathbf{H}_j\big(\mathbf{y}_j(t), \mathbf{Y}_j(t), \mathbf{u}_j(t), \Theta_j\big) \tag{3.5}$$

where $\mathbf{H}_j(\cdot)$ is given by

$$\mathbf{H}_j = \cup_{k \in \mathbb{N}; k \leq K_j} H_{k,j}(\mathbf{y}_j(t), \mathbf{Y}_j(t), u_{k,j}(t), \Theta_{k,j}) \tag{3.6}$$

$\mathbf{u}_j(t)$ is a user control or experimental input.

$$\mathbf{u}_j(t) = \cup_{k \in \mathbb{N}; k \leq K_j} u_{k,j}(t) \tag{3.7}$$

and Θ_j is parameter set

$$\Theta_{k,j} = \{\theta_{l,k} : l, k \in \mathbb{N}; l \leq L_k; k \leq K_j\} \tag{3.8a}$$

$$\Theta_j = \cup_{k \in \mathbb{N}; k \leq K_j} \Theta_{k,j} = \cup_{k=1}^{K_j} \Theta_{k,j} \text{ for any given vertex } j \tag{3.8b}$$

$$\text{and } \boldsymbol{\Theta}_J = \cup_{j \in \mathbb{N}; j \leq J} \Theta_j = \cup_{j=1}^{J} \Theta_j \tag{3.8c}$$

L_k is the number of parameters for a given state variable, K_j is the number of state variables for a given vertex j, and J represents the size of the network (i.e. the total

number of vertices). Note that the functions comprising the set $\mathbf{H}_j(\cdot)$, each advance their respective variables in time:

$$\begin{aligned} y_{1,j}(t+\Delta t) &= H_{1,j}(y_{1,j}(t), \mathbf{Y}_j(t), u_{1,j}, \Theta_{1,j}) \\ y_{2,j}(t+\Delta t) &= H_{2,j}(y_{2,j}(t), \mathbf{Y}_j(t), u_{2,j}, \Theta_{2,j}) \\ &\cdots \\ y_{K,j}(t+\Delta t) &= H_{K_j,j}(y_{K_j,j}(t), \mathbf{Y}_j(t), u_{K_j,j}, \Theta_{K_j,j}) \end{aligned}$$

Similarly, we define a function $\mathbf{G}_{ij}(\cdot)$ that describes the time course of the weighing sets $\Omega_{ij}(t)$ with parameter sets Λ_{ij} as follows:

$$\Omega_{ij}(t+\Delta t) = \mathbf{G}_{ij}\big(\Omega_{ij}(t), \mathbf{y}_j(t), \mathbf{y}_i(t-\tau_{ij}), \Lambda_{ij}\big) \tag{3.9}$$

where analogous with equation (3.6), $\mathbf{G}_{ij}(\cdot)$ is given by

$$\mathbf{G}_{ij} = \cup_{k\in\mathbb{N};k\leq K} G_{k,j}\big(\Omega_{ij}(t), \mathbf{y}_j(t), \mathbf{y}_i(t-\tau_{ij}), \Lambda_{k,ij}\big)$$

with parameters

$$\Lambda_{k,ij} = \{\lambda_{l,k} : l, k \in \mathbb{N}; l \leq L_k; k \leq K\} \tag{3.10a}$$

$$\mathbf{\Lambda_{ij}} = \cup_{k\in\mathbb{N};k\leq K}\Lambda_{k,ij} \text{ for any given vertex pair } ij \tag{3.10b}$$

The delays between vertex pairs τ_{ij} are defined as functions of the positions of the two vertices:

$$\tau_{ij} = D\big(\mathbf{x}_i, \mathbf{x}_j, \Gamma_{ij}\big); \quad D(\cdot) \geq 0 \tag{3.11}$$

where the set Γ_{ij} is the set of parameters of the non-negative function D, specific to the pair ij.

Formally, the temporal evolutions of $\mathbf{y}_j(t)$ and $\Omega_{ij}(t)$ are continuous and expressed as a discrete delay differential equations with delays τ_{ij} for all vertices connecting to vertex j, so that the continuous forms of equations (3.5) and (3.9) are

$$\frac{\partial \mathbf{y}_j(t)}{\partial t} = \mathbf{H}_j\big(\mathbf{y}_j(t), \mathbf{Y}_j(t), \mathbf{u}_j(t), \Theta_j\big) \tag{3.12}$$

and

$$\frac{\partial \omega_{ij}(t)}{\partial t} = \mathbf{G}_{ij}\big(\Omega_{ij}(t), \mathbf{y}_j(t), \mathbf{y}_i(t-\tau_{ij}), \Lambda_{ij}\big) \tag{3.13}$$

In the limit as $\Delta t \to 0$, equations (3.5) and (3.9) can be written as $\mathbf{y}_j(t+dt)$ and $\Omega_{ij}(t+dt)$ and equations (3.12) and (3.13) apply. However, from a practical experimental perspective time measurements will always be finite and the discrete forms need be considered. As such, in this paper we do not pursue further the interesting theoretical implications of the continuous forms given by equations (3.12) and (3.13).

Finally, we define an observation set $\mathbf{z}_j(t)$ composed of M variables, that operates directly on the state set $\mathbf{y}_j(t)$:

$$\mathbf{z}_j(t) = \mathbf{F}_j\big(\mathbf{y}_j(t), \Phi_j\big) \tag{3.14}$$

The observation function $\mathbf{F}_j$ is vector valued:

$$\mathbf{F}_j(t) = \cup_{m\in\mathbb{N};m\leq M} F_{m,j}\big(\mathbf{y}_j(t), \Phi_{m,j}\big) \tag{3.15}$$

with parameter set Φ_j given by

$$\Phi_{m,j} = \{\phi_{l,m} : l, l \in \mathbb{N}; l \leq L_m; m \leq M\} \tag{3.16a}$$

$$\mathbf{\Phi_j} = \cup_{m\in\mathbb{N};m\leq M} \Phi_{m,j} \text{ for any given vertex } j \tag{3.16b}$$

The framework presented here is general, as it allows for communication between any two state variables between any two vertices. Transition functions and their parameters are defined specific to vertex $\mathbf{H}_j(\cdot)$ or communication between vertex pairs $\mathbf{G}_{ij}(\cdot)$. This produces a large set of functions and parameters, though in practice one or two different functions are applied to all cells or combinations. The weighing set Ω_{ij} can operate on all state variables of the connecting vertex i into target j, though usually one state of i is transmitted to one state in j. In the next section, we will describe how several dynamics and communication models used in cellular networks fit within this framework to reproduce observable quantities similar to experimentally measured data.

3.4 Results

The framework can accommodate essentially all models of both neuronal and astrocytic dynamics. Independent of the specifics of any single cell model chosen, the framework provides a compact mathematical structure that quantitatively describes signaling and information propagation and flow in geometrically defined networks. The geometry and physical connectivity topology of the network can be simulated (e.g. random, scale free, or small world) or measured from experimental data such using methods such as optical imaging. Regardless of how one choses to set up the network, the framework provides a description of information flow through the network given knowledge of temporal signaling delays and chosen single cell models, or can be used to identify and map unknown functional connectivities and parameters in real neural circuits and networks. In all cases, the framework is able to provide an estimate of the complete description of the functional network and the interaction between all observable and hidden state variables and parameters. Figure 3.1 illustrates a simple five vertex example that summarizes everything that is needed to describe the functional dynamics of information flow through the network. Figure 3.2 provides a specific example of the network from figure 3.1 using a Hodgkin-Huxley model and simulating one second worth of data. Note how the framework provides experimentally measurable variables (calcium and membrane voltage) for every cell in the network in the temporal sequence dictated by the geometry and connectivity of the network.

3.4.1 Individual Cell Dynamics

In this section we discuss how neuronal models of single cell dynamics, synaptic connections, plasticity, and observation fit within the framework. We begin by showing how to construct the state transition of an isolated (unconnected) vertex, and then build the full transition function by connecting multiple vertices into a network. In an isolated vertex case with no incoming connections, the state transition reduces to

$$\mathbf{y}(t + \Delta t) = \mathbf{H}(\mathbf{y}(t), \mathbf{u}(t), \Theta) \tag{3.17}$$

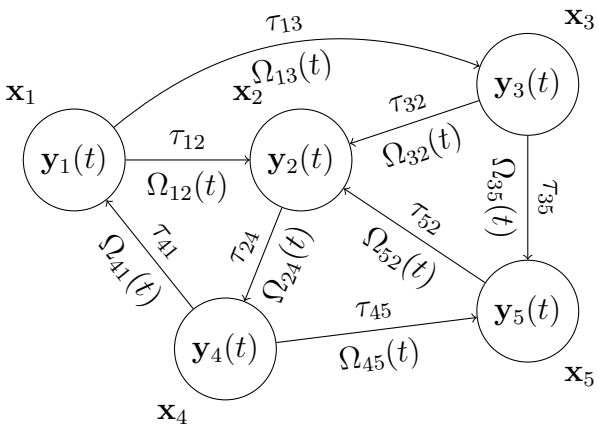

Figure 3.1: A five-vertex dynamic network. Each vertex j has a position in physical Cartesian space denoted by vector $\mathbf{x}_j$, and a dynamic state set $\mathbf{y}_j(t)$. A vertex's dynamic state varies in discrete time steps, and is influenced by its own previous state and the states of other vertices connecting into it, with a delay τ and a functional connection weight Ω. The time delays between vertices are a function of their positions in space. The magnitude of the connection weights, Ω_{ij}, are estimated based from vertices' known positions and the observed dynamics.

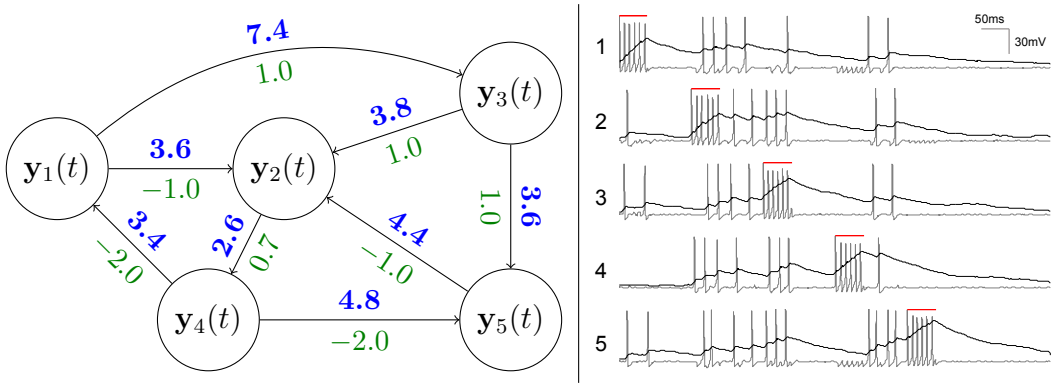

Figure 3.2: Signaling dynamics of the network from figure 3.1. A Hodgkin-Huxley model is used for single-cell dynamics in a one second simulation. The delays (written in **bold blue**) are in milliseconds and are based on the cartesian distances between cells. Functional weights, shown as relative strengths in green, are chosen arbitrarily for the purposes of this example. The right panel shows the network dynamics for each of the five cells following a sequential pulse stimulation (red horizontal bar) at each cell. Experimentally observable calcium traces are shown as solid black lines on an arbitrary vertical scale. Voltage, a hidden variable, is shown as a gray line and constitute the neurons' action potentials.

This generic form encompasses neuronal models described in differential equation form, as well as those with a state reset based on some threshold value. Most neuronal models are expressed in differential form as

$$\frac{d\mathbf{y}}{dt} = h(\mathbf{y}(t), \mathbf{u}(t), \Theta) \tag{3.18}$$

Converting 3.18 into the state transition form given by 3.17 is a matter of numerical integration with an integration method of choice. Using Euler's method, for example, the state transition function $\mathbf{H}_j$ of the system in 3.18 becomes

$$\mathbf{H}_j(\mathbf{y}_j(t), \mathbf{u}_j(t), \Theta_j) = \mathbf{y}_j(\Theta, t) + \Delta t \cdot h(\mathbf{y}_j(t), \mathbf{u}_j(t), \Theta_j) \tag{3.19}$$

Here we used the Euler method of integration for its simplicity and clarity, but other, but more complex integration methods like trapezoidal or Runge-Kutta can also be used to generate the next time step from the current step.

As an example, consider the simple Fitzhugh-Nagumo oscillator used to model neurons. In its differential form the model is given by the pair of equations

$$\begin{aligned} \frac{dV(t)}{dt} &= aV(t) - bV(t)^3 - cW(t) + S(t) + U(t) \\ \frac{dW(t)}{dt} &= e(V(t) + f - gW(t)) \end{aligned} \tag{3.20}$$

The state set $\mathbf{y}(t)$ is comprised of two state variables $\mathbf{y}_j(t) = \{V_j(t), W_j(t)\}$. The experimental control set is composed of only one variable, affecting the $V(t)$ state variable, so $\mathbf{u}_j(t) = \{U_j(t)\}$. This system can be expressed in state transition form as $\mathbf{H}_j = \{H_{V,j}, H_{W,j}\}$, with

$$\begin{aligned} H_{V,j}(t) &= V_j(t) + \Delta t \cdot \big(aV_j(t) - bV_j(t)^3 - cW_j(t) + U_j(t)\big) \\ H_{W,j}(t) &= W_j(t) + \Delta t \cdot \big(e(V_j(t) + f - gW_j(t))\big) \end{aligned} \tag{3.21}$$

Note that we index the state transition functions based on the state variables they operate on; for example, $H_V(t)$ advances $V(t)$. The parameter set for this system is

composed of the parameters for each of the state transition equations in 3.21:

$$\begin{aligned}\Theta_{V,j} &= \{a, b, c\} \\ \Theta_{W,j} &= \{e, f, g\} \\ \Theta_j &= \Theta_{1,j} \cup \Theta_{2,j}\end{aligned} \tag{3.22}$$

This system has two state variables and six parameters.

Another class of neuronal models are those with a hard reset. These models are also described in differential equation form, but contain a hard reset when a state variable reaches a certain value. As an example, consider the Izhikevitch simple model, written in differential form as

$$\begin{aligned}\frac{dV(t)}{dt} &= \frac{1}{C}\Big[k(V(t) - V_r)(V(t) - V_t) - W(t) + S(t) + U(t)\Big] \\ \frac{dW(t)}{dt} &= a\big(b(V(t) - V_r) - W(t)\big) \\ &\left.\begin{aligned} V(t+) &= c \\ W(t+) &= W(t) + d \end{aligned}\right\} \text{ if } V(t) > V_{peak} \text{ (spike event)}\end{aligned} \tag{3.23}$$

This model consists of a voltage and amplifying currents ($V(t)$) and a resonant gating variable ($W(t)$). The system has up to 9 parameters, and resets both state variables when a certain voltage threshold (V_{peak} parameter) is surpassed. Mapped onto our framework, the model and its transition functions are

$$H_{V,j} = \begin{cases} V_j(t) + \Delta t \cdot \frac{1}{C}\Big[k(V_j(t) - V_r)(V_j(t) - V_t) - W_j(t) + U_j(t)\Big] & \text{if } V_j(t) < V_{peak} \\ V_{reset} & \text{otherwise} \end{cases} \tag{3.24}$$

$$H_{W,j} = \begin{cases} W_j(t) + \Delta t \cdot \big[r\big(b(V_j(t) - V_r) - W_j(t)\big)\big] & \text{if } V_j < V_{peak} \\ W_j(t) + d & \text{otherwise} \end{cases}$$

Here, the parameter set is $\Theta_j = \{C, k, V_r, V_t, V_{reset}, V_{peak}\} \cup \{r, b, d, V_r, V_{peak}\}$, and just like the Fitzhugh-Nagumo model, $\mathbf{y}_j(t) = \{V_j(t), W_j(t)\}$, $\mathbf{u}_j(t) = \{U_j(t)\}$ and $\mathbf{H}_j = \{H_{V,j}, H_{W,j}\}$. By changing the parameter values of the individual models, different classes of neurons can be simulated with the same transition function.

Similarly, any model can be accommodated and fit into the framework, from the simplest to the most complex. Traditionally, all neuronal models have membrane voltage as a state variable and propagate a discrete signal in the form of an action potential when the membrane voltage rises past some threshold value at the axon hillock in response to depolarizing and hyperpolarizing currents in dendrites mediated by spatial and temporal summation of presynaptic currents. The simplest neuronal model, the leaky integrate and fire (LIF) has voltage as a single state variable that decays to a target value and is perturbed by incoming currents. If the voltage rises past a threshold value it is reset at the next time step to a reset value. One of the most complex and realistic single-cell models is the Hodgkin-Huxley (HH) model which relies on four state variables $\{v, m, n, h\}$ to describe the dynamics responsible for the generation of action potentials. The number of parameters increases with the number of state variables, from 4 in the LIF model to 22 for the HH model. Additionally, the required time step is shorter for HH models, being on the order of 0.03 milliseconds compared to roughly 5 milliseconds for the LIF model. The increased number of state variables and parameters along with shorter time steps puts a significant computational burden on any simulation or mapping algorithm. The question of which model and how much complexity is required to best describe real-world data is not trivial and depends on the purpose and intent of the modeling.

Astrocyte models are expressed in differential forms similar to equation 3.18. Further research into astrocytic models is important though because astrocytes have been shown to play a direct role in the bidirectional communication between themselves and neurons via intracellular calcium transients and intercellular calcium waves under controlled experimental conditions [FSG02, APM$^+$08, Ver06, CCP$^+$03, MYBS08, SG06] and more recently physiologic conditions in the neural retina cerebellum [KNMN09, HKG$^+$09]. Pathophysiologically intercellular calcium waves in astrocytes independent of neuronal hyperactivity have recently been shown to occur spontaneously *in vivo* in the APP/PS1 transgenic mouse model of Alzheimers disease [KLHB09]; and amyloid beta has been shown to be sufficient to trigger complex temporally delayed intercellular calcium waves in isolated astrocyte networks [CYM$^+$09].

The state transition framework handles all single-cell dynamic models, as

well as heterogenous systems of different cell types, either by different parameter sets or state transition function sets or both. Whatever the dynamics of individual neurons or astrocytes, all perform the same general task whereby processes and inputs generate outputs to other cells in a connected network.

3.4.2 Cellular Network Signaling

There are three components to cellular signaling: how long it takes for information from one cell to reach another, what are the effects of one cell on another and how do those effects change through time given the relative dynamics of the two connected cells. In this section we describe how signaling delays, functional connectivity and plasticity are accomodated by the framework.

Signaling Delays

Neurons and astrocytes form signaling networks that pass and process information between cells, and the state transition function must be extended to include signal propagation between functionally connected cells. In biological cellular networks, signal propagation occurs at a finite and relatively slow speed (i.e. compared to electronic circuit networks). Information in cell networks propagates on the order of microns per second for astrocytes to meters per second for myelinated axons in neurons. Thus the influence of the dynamics of one cell is felt by another cell after some delay τ_{ij}. While the general form for the delay is given by 3.11 above, the simplest form it can take is when signaling is a geodesic between vertices (or between the centers of cell bodies or centers of other cellular compartments as needed in morphologic models) and the transmission speed s is constant:

$$\tau_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{s} \tag{3.25}$$

i.e.

$$\tau_{ij} = D(\Gamma, \mathbf{x}_i, \mathbf{x}_j),\ \Gamma_{ij} = \{s\}\ (c.f. 3.11 \text{ above})$$

Here the delay is simply the Euclidean distance between the cell centers di-

vided by the transmission speed. For a diffusive network, as is the case with astrocytes, delays are proportional to the square of the distance between vertices. A more complex delay function may take into account knowledge about the particular physiology of the network, curved paths between cells, non-uniform speeds, etc. The dependency of the framework on the delays is critical to its ability to describe how and when information within the network is processed, ultimately to a significant degree dictating the intercellular dynamics of the overall neural circuit or network.

Figure 3.3 illustrates the dependency of network dynamics on signaling speed and delay times in a 100-vertex three dimensional network by varying the intercellular signaling speed, with everything else, including its geometry (i.e. its physical connectivity), the functional connectivity and input stimulus, remaining the same. We stimulated all cells with 500 ms of depolarizing current. The delays are inversely proportional to the signal propagation speed. We illustrate the effects of three signaling speeds, 2, 20, and 200 pixels/ms. At the lowest signaling speed, 2 pixels/ms, a low-frequency periodic activity was produced that qualitatively resembles a central pattern generator. At a speed of 20 pixels/ms fewer cells exhibited low frequency oscillations, and signaling became more sporadic. For both the 2 and 20 pixel/ms propagation speeds however, signaling continued past the period of stimulation. At a propagation speed of 200 pixels/ms however, there was no signaling past the stimulus period. In fully recurrent networks such as the one illustrated here, delays serve as a form of signal storage, essentially giving cells time to recover from a refractory period between activations, which in turn maintains recurrent signaling propagation well beyond an initial stimulus. For some appropriate range of signaling speeds, and therefore delay times, this recurrent signaling can settle into a repeatable pattern. If however, the signaling speeds are too fast, incoming signaling from upstream cells never have an opportunity to activate downstream cells because they are still refractory and do not respond. This leads to signaling in the network quickly dying away and not being sustained without it being driven by an external stimulus, as is the case with speeds of 200 pixels/ms in this example. A full discussion of the dependency of the network dynamics on the variables that govern it is very involved and beyond the scope of this paper. However, this example serves to illustrate that along with a network's connectivity topology and individual vertex dynamics, sig-

naling delays play a crucial role in its overall response and dynamics and must be part of any network simulation or modeling framework that attempts to capture the inherent behaviors of neurobiological networks.

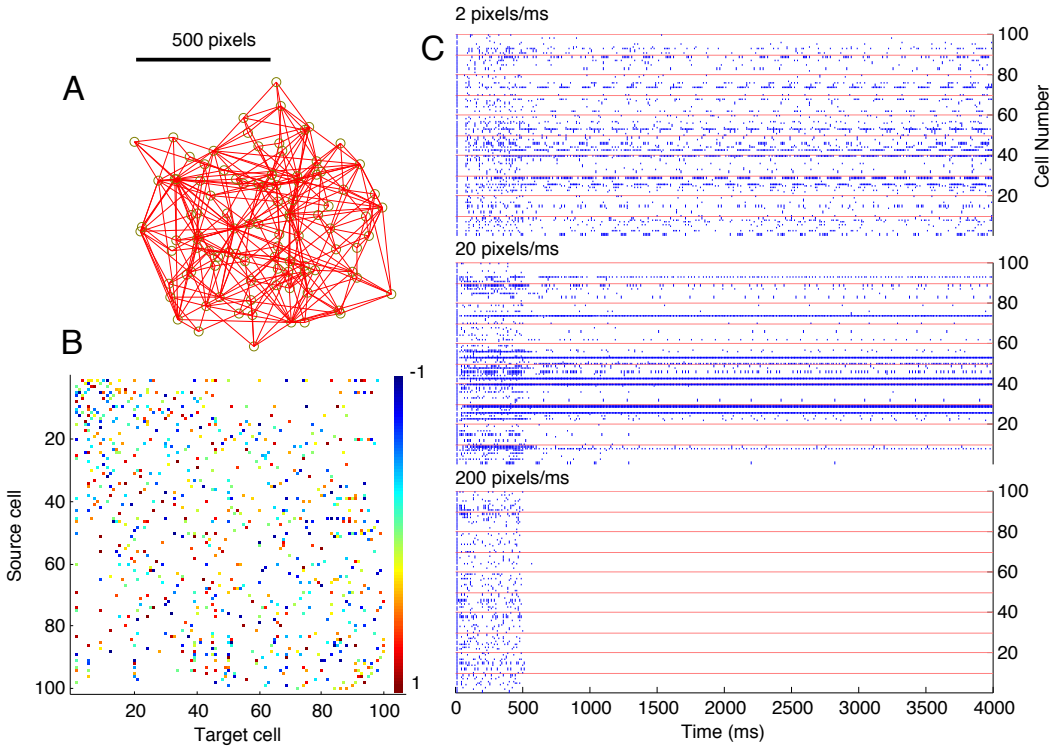

Figure 3.3: Effects of signaling speed on network dynamics.The network with spatial locations and physical connections shown in panel **A**, is assigned random weights uniformly distributed between -1 and 1 on each physical edge, panel **B**. An Izhikevitch simple model of bursting neurons was used to model the individual vertex dynamics. **C.** By varying the speed of signal propagation, the delay distributions are scaled, having a substantial impact on the spike dynamics (see text).

Functional Connectivity

The other component of signaling is the functional connectivity of the network, or how the state of one vertex influences the state of another. In equation 3.2 the set $\mathbf{Y}_j$ collects all the states of all the vertices in the network except j, delayed by a time value relative to j. When $\mathbf{Y}_j$ is passed into the transition function $\mathbf{H}_j$, the information contained in every state of every other vertex is made available to all

state variables in vertex j. Within the framework, this is the broadest possible scope of connectivity, though in practice typically information from one variable affects one or more variables in another vertex.

As an example, consider a neuron and its pre-synaptic (chemical) inputs that induce post-synaptic currents. The post-synaptic current (PSC) is modeled with additional state variables in the neuron's state vector $\mathbf{y}_j(t)$. There are two PSC models in wide use which we have tested within the framework, although it is in no way limited to these two examples. In general, a pre-synaptic neuron causes a PSC in a post-synaptic neuron that ultimately affects membrane voltage. Modeling the effect of an incoming signal on a target cell is key to establishing connectivity based on the observed cell dynamics. Post-synaptic current is represented as a function of the form

$$s(t) = g_{max} \cdot r(t) \cdot (V(t) - E_{rev})$$

where the resultant signals $s_j(t)$ are summed and passed to the voltage state variable in connected vertex i. g_{max} is the maximum conductance, and in the case of a specific synapse, can be expressed as a product of the maximum allowable conductance and the (instantaneous) functional connection weight ω_{ij}. $r(t)$ describes the time course of the current, and is generally one of two forms, depending on the neuron type and neurotransmitter release [DMS98, DMS94]:

$$r(t) = e^{-at} \qquad \text{simple exponential}$$
$$r(t) = at \cdot e^{-at} \qquad \alpha\text{-function}$$

where a is the time decay constant. In both cases, time starts at the moment of activation, in this case the time of the activation of the pre-synaptic cell plus the delay to the post-synaptic cell, τ_{ij}. For either case, the expressions for $r(t)$ can be written as linear differential systems, with the spike as the impulse. For the simple

exponential the differential equation is

$$\begin{aligned}
\frac{dr_j(t)}{dt} &= -ar_j(t) \\
r_j(t+) &= r_j(t) + \omega_{ij} \quad \text{upon arrival of spike event from vertex } i
\end{aligned} \tag{3.26}$$

For the α-function, another state variable $p(t)$ is used:

$$\begin{aligned}
\frac{dr_j(t)}{dt} &= p_j(t) \\
\frac{dp_j(t)}{dt} &= -a^2 r_j(t) - 2ap_j(t) \\
p_j(t+) &= p_j(t) + \omega_{ij} \quad \text{upon arrival of spike event from vertex } i
\end{aligned} \tag{3.27}$$

Shifting the arrival delays is simply a matter of shifting the spike detection function of the pre-synaptic neuron, so the arrival time from vertex i to vertex j is effectively the time shifted function of the voltage of $i : V_i(t - \tau_{ij})$. The synaptic current is a decreasing exponential with rate a, incremented by a weight value ω_{ij} upon arrival of a spike occuring τ_{ij} time units ago at another cell i.

For example, extending the state set for the Izhikevitch model given by 3.24 for an arbitrary vertex to include an α-function PSC model of post-synaptic currents produces $\mathbf{y}_j(t) = \{V_j(t), W_j(t), r_j(t), p_j(t)\}$ and $\mathbf{H}_j = \{H_{V,j}, H_{W,j}, H_{r,j}, H_{p,j}\}$ where,

$$H_{V,j} = \begin{cases} V_j(t)+ & \\ \frac{\Delta t}{C}\Big[k(V_j(t) - V_r)(V_j(t) - V_t) - W_j(t) + r(t) + U_j(t)\Big] & \text{if } V_j(t) < V_{peak} \\ V_{reset} & \text{otherwise} \end{cases} \tag{3.28}$$

$$H_{W,j} = \begin{cases} W_j(t) + \Delta t \cdot \big[r\big(b(V_j(t) - V_r) - W_j(t)\big)\big] & \text{if } V_j < V_{peak} \\ W_j(t) + d & \text{otherwise} \end{cases}$$

$$H_{r,j} = r_j(t) + \Delta t \cdot p_j(t) \tag{3.29}$$

$$H_{p,j} = p_j(t) + \Delta t \cdot \left(-a^2 r_j(t) - 2ap_j(t) + \sum_{k=1,k\neq j}^{J} \omega_{kj} \mathrm{spd}(V_k(t - \tau_{kj}))\right) \tag{3.30}$$

The spike detection function is given by

$$\text{spd}\left(V(t)\right) = \begin{cases} 1 & \text{if } V(t) \geq V_{peak} \\ 0 & \text{otherwise} \end{cases} \tag{3.31}$$

Here, the spike inputs from other neurons are passed into the state variable $p(t)$ with transition function $H_{p,j}$, integrated by variable $r(t)$, and passed into the voltage variable $V(t)$. The functional weights only operate on the voltage spikes from other neurons, so only information from the voltage state variable $V(t)$ is passed to other neurons. A similar set can be constructed with the PSC models in 3.26.

Plasticity

Up to this point we have described models with fixed functional connectivity strengths. This assumption is valid for networks observed over short periods where connection strengths can be assumed to be constant for the purposes of mapping or simulating since the plasticity mechanisms that modulate connection strengths operate on longer time scales. If the weights change as a function of the activities of the cells it connects, the framework can be used to modulate connective strengths (equations 3.9 and 3.10). Just as with single cell dynamics and network connectivity, there are many models of plasticity and we will not attempt to list or review them all here. Rather, we describe how a simple spike-time dependent plasticity model reviewed by Bi and Poo in [BP01] that incorporate long-term potentiation (LTP) and long-term depression (LTD) can be easily implemented within the proposed framework.

In equation 3.9 we described the functional strength or weight Ω_{ij} with transition function $\mathbf{G}_{ij}$ analagous to the state set for individual neurons. When the weight is constant, the set contains only one variable so $\Omega_{ij} = \{\omega_{ij}\}$ and there is no temporal change in ω_{ij} and no transition function or parameters. However, if the weight is modulated by the activitiy of cell i on j (the connection is directional, so ω_{ij} modulates information flowing from i to j), then the states $\mathbf{y}_i(t - \tau_{ij})$ and $\mathbf{y}_j(t)$ affect how $\Omega_{ij}(t)$ changes in time.

The neuronal plasticity model in [BP01] describes strengthening and weaken-

ing of synaptic conductance based on the timing of spikes in pre and post-synaptic neuronal spiking. If a post-synaptic spike occurs immediately after a pre-synaptic spike, the synaptic conductance is increased and the functional connection is effectively strengthened. If the post-synaptic neuron spikes before the pre-synaptic neurons, the connection is weakened. Other conditions like spike coincidence or long times between the spikes of pre and post-synaptic neurons have no affect on the synaptic conductance.

To incorporate this model into our framework we first augment the set of cell states with another state variable $s_j(t)$, that stores the time from the last spike. This state variable has transition function $H_{s,j}$:

$$H_{s,j} = (1 - \text{spd}(V_j(t)) \cdot (s_j(t) + \Delta t) \tag{3.32}$$

This model uses two exponential curves to describe changes in synaptic conductance based on the spike times. By defining the function $q_{ij}(t)$ as

$$q_{ij}(t) = (s_i(t - \tau_{ij}) - s_j(t)) \cdot \max(\text{spd}(V_i(t - \tau_{ij}), \text{spd}(V_j(t))$$

we can reconstruct the plasticity model within the framework as

$$\frac{d\omega_{ij}(t)}{dt} = \alpha \cdot \text{sign}(q_{ij}(t)) \cdot \exp\left(-\beta |q_{ij}(t)|\right)$$

Thus, the transition function for dynamic synaptic weights is

$$G_{ij} = \omega_{ij}(t) + \Delta t \cdot (\alpha \cdot \text{sign}(q_{ij}(t)) \cdot \exp\left(-\beta |q_{ij}(t)|\right)) \tag{3.33}$$

The parameter set for connection ij is $\Lambda_{ij} = \{\alpha, \beta\}$. This way, a static weight ω_{ij} can be converted into a dynamical one $\omega_{ij}(t)$, with behaviors governed by any arbitrary plasticity model and its associated parameter set Λ_{ij}. It is important to note that while here we describe only one scalar weight between two vertices, the framework as defined can accomodate as many weights as there are state variables for a given vertex, thereby describing different classes of intercellular signaling between vertex pairs. For example, in networks of neurons signaling may occur via gap junctional

mediated electrical synapses in addition to chemical synapses, while in astrocyte networks intercellular signaling is typically mediated by diffusional processes (e.g. vesicularly released adenosine triphosphate, ATP).

3.4.3 Experimental Observability Through Calcium Observation

Typically only one or a few of the state variables in the state vector are available for observation (i.e. are experimentally measurable). This is certainly the case with cellular networks, especially in neurons where voltage is measured as an indicator of signaling activity. But simultaneous voltage measurements are difficult for networks of many neurons where the geometry of the network may be important to the analysis or interpretation of the data. While high density planar multi-electrode arrays can record from a few hundred cells at once, it is typically not possible to correlate recorded activity with the native geometry of the network. (There is one notable exception to this that we are aware of, which is the ganglion cell monolayer in the peripheral retina. Chichilnisky and colleagues are able to computationally infer the geometry and functional activity of these retinal ganglion cells due to their unique planar arrangement- see [Pil05, SRC08, CB99]. However, in the brain and even in the retina where multilayered ganglion cells receive incoming macular inputs it is not possible to do with electrode arrays.) Given the challenges associated with direct electrophysiological measurements of large neuronal ensembles, calcium fluorescence imaging has been used as an indirect measure of the dynamics of large neuronal networks. The dynamics of calcium, often influenced by voltage spiking, are modeled as state variables and associated transition functions added to the state set. There are a few models describing the time course of calcium as it is driven by changes in membrane voltage, and attempts to develop more refined and robust calcium dynamics models specifically for the study neural microcircuits and networks is a very active area of research. Our intent here is not to describe on-going efforts or the state of the art but simply to illustrate the integration of one such model within our framework.

The simplest model of calcium dynamics can be expressed as a linear system,

with a spike input

$$\tau_c \frac{dc(t)}{dt} = -\frac{1}{\tau_c} c(t) + \text{spd}(V(t)) \tag{3.34}$$

This model is integrated into the framework by the addition of another state variable $c(t)$ describing the cytosolic calcium concentration to the state set, with transition function:

$$H_{c,j} = c_j(t) + \Delta t \left(-\frac{1}{\tau_c} c_j(t) + \text{spd}(V_j(t)) \right)$$

This model has one parameter τ_c which describes the removal rate of calcium, after an input caused by a spike. This is the simplest model used for calcium dynamics based on neuronal spiking and is often used to extract spikes from calcium [**?**]. More complex and non-linear models of calcium dynamics have been developed and these can also readily be integrated within the framework .

Finally, there is the issue of observable variables. In equation 3.14 we defined an observable set $\mathbf{z}_j(t)$ as some function of the current state set $\mathbf{y}_j(t)$. When using fluorescent calcium indicators, the sole observation variable is the recorded intensity $I_j(t)$ for cell j at a particular pixel reflecting some linear multiplier of the cytosolic calcium concentration at that point in the visual field, based on the dye loading in a cell, as well as the camera, microscope, and illumination setup. Defining $\mathbf{z}_j(t) = \{I_j(t)\}$ with observation function $\mathbf{F}_j$,

$$I_j(t) = \mathbf{F}_j(c_j(t), \mathbf{\Phi}_j) = nc_j(t) + b + \mathcal{N}\left(0, \gamma^2\right)$$

The parameter set $\mathbf{\Phi}_j = \{n, b, \gamma\}$ represents the scaling, offset, and noise standard deviation of the observation function. The function $\mathcal{N}(0, \gamma^2)$ generates a normally distributed, random value with zero mean and γ standard deviation. The noise term models the type of frame-to-frame variation typically seen in the amplification of the CCD signal prior to digitization. The size of γ is proportional to the magnitude of the noise, itself affected by camera type and gain settings.

Since the framework is defined at cellular resolution, we are making the simplifying assumption that the recorded intensity represents the average intensity for the region of interest demarcating a specific cell j within a larger field. Additionally, if the camera records at a slower frequency than the transition dynamics $f_{camera} > 1/\Delta t$,

where f_{camera} is the camera recording frequency and Δt is the time incriment (*c.f.* equation 3.5), one is averaging intensity values for the duration that the camera shutter stays open. If this is the case then multiple sequential calcium concentration values would be averaged to produce a single intensity reading.

3.4.4 GPU Implementation and Benchmarks

The practical application of the theoretical framework both for simulation and, as will be described in a subsequent paper, for mapping the unknown functional connectivity of experimentally observed cellular networks necessitates its implementation in an appropriate computing environment. We have taken advantage of emerging high performance general purpose-graphics processing unit (GPU) parallel computing, although it can run as serial code on a normal central processing unit (CPU), which we have also tested. Within the GPU environment, the code has been designed to run on nVIDIA graphics cards equipped with the CUDA interface (see http://www.nvidia.com/cuda). In this way, we can parallelize vertex dynamics, signaling dynaimcs, and observation integrations over many processor cores, achieving significant speedup over CPU or cluster computations. The framework and associated single cell dynamic and network connectivity models have been coded as compact MATLAB-callable libraries. All graphics user interface (GUI) and input/output (I/O) operations are handled through MATLAB and the code has been written in both MATLAB and plain C libraries that communicate through MATLAB. The libraries offer direct control over all parameters for all vertices. Using plain C language, any model that can be analytically described within the framework can be easily coded into a simulation library. From a practical experimental perspective GPU computation offers unprecedented scalability to larger systems with full access to all state variables and parameters, enabling rapid parallel simulation when the framework is run in the forward direction, and real-time dynamic mapping when the framework is applied to the inverse problem of mapping unknown functional connectivities of cellular neural networks. Speed and parallelization are critical for statistical simulation based identification methods such as particle filtering to operate in real time or near real time.

Benchmarks in figure 3.4 show the relative speeds of the CPU and GPU implementations of two different dynamic cell models within the framework, an Izhikevitch model and a Hodgkin-Huxley model, simulated in 40 test networks (see section 3.4.5 below regarding the test networks). The parallel GPU implementation performed anywhere from 8 to 200 times faster than the same code executed on a single CPU process. Performance was measured as a slowdown or speedup factor relative to real time. It is important to note that identical code ran on both CPU and GPU, with the former emulating the latter. The Izhikevitch model had six state variables and a timestep of Δt 1 millisecond. We simulated 10 seconds worth of data equivalent to 10,000 time steps. The more computationally intensive Hodgkin-Huxley model had eight state variables and a time step of 0.03 milliseconds. One second worth of data was simulated for the HH model, corresponding to about 33,333 time steps. Benchmarks were calculated as the dimensionless ratio of the actual period of time simulated (i.e. real time) to the amount of physical computational time it took the GPU or CPU to carry out the simulations. Because of the parallel implementation on the GPU, the performance falloff was much slower than on the CPU with increasing network size. In both CPU and GPU cases, performance decreased with increasing network density (total number of edges/vertices square), since in more dense networks more information is transferred between edges. This is more apparent in the random networks that have higher density than other network classes, producing relatively slower processing speeds.

In general, the constant time step in the framework makes parallelization easy on GPU architectures, delivering network simulation performance near or even faster than real-time. Because of the parallel threading on GPUs, performance is only modestly decreased when going from a 100 to a 1000-vertex network. For the HH model, the GPU was able to carry out the computations in essentially real time for all of the networks tested except for the largest random networks, which were about 10 to15 times slower then real time for random networks with over a thousand vertices (upper left panel in figure 3.4). For computationally simpler models such as the Izhikevitch model the GPU computations were actually faster then the period being simulated (i.e. faster than real time), ranging from roughly 10 to 20 times faster for most networks and about real time for the largest random networks

(upper right panel in figure 3.4). In contrast, CPU computations were always slower than real time, from 15 to 800 times slower for the HH model depending on the network class and size (lower left panel in figure 3.4) and from just under near real time for small 10 vertex networks to about 15 times slower for the largest random networks for the Izhikevitch model (lower right panel figure 3.4). Additional GPU cards can further improve performance by splitting the task of advancing temporal cell dynamics; however transfer of information between cells is still limited by memory and bus speeds, so dense networks will run slower than sparse networks. It is important to appreciate that the ability to carry out such forward simulations or to solve the inverse problem of mapping unknown functional connectivity topologies of networks in near real time or faster than real time using GPU computing is due to the mathematical construction of the framework and its efficient algorithmic implementation. It is impossible to do such network simulations or mappings in real time using biophysical compartmental simulation environments. The MATLAB and CUDA code for the framework are available for download from the authors' website (http://www.silva.ucsd.edu/Silva_Lab/Links.html).

3.4.5 Standardized Tests for Connectivity Estimation

Lastly, we propose a standardized basis test set to evaluate the effectiveness of mapping algorithms. A standard test set is well accepted in the field of non-linear optimization, providing a standard measure of different algorithms [HS80]. There are two benefits to having a standard set of networks to use for mapping. First, multiple algorithms can be evaluated against the same network and model, providing relative performance benchmarks. Second, data generated for a network using one dynamic model can be mapped using another dynamic model and comparisons can be made between the original topology and the mapped topology. This latter approach helps answer questions about model fitness, which are especially useful when trying to map data with multiple models or uncertainty in models. We emphasize that the test set we propose here should in no way be interpreted as implying that the full complexity and variability of real neural circuits and networks is captured or even described by the set. But we argue that any mathematical method or algorithm that

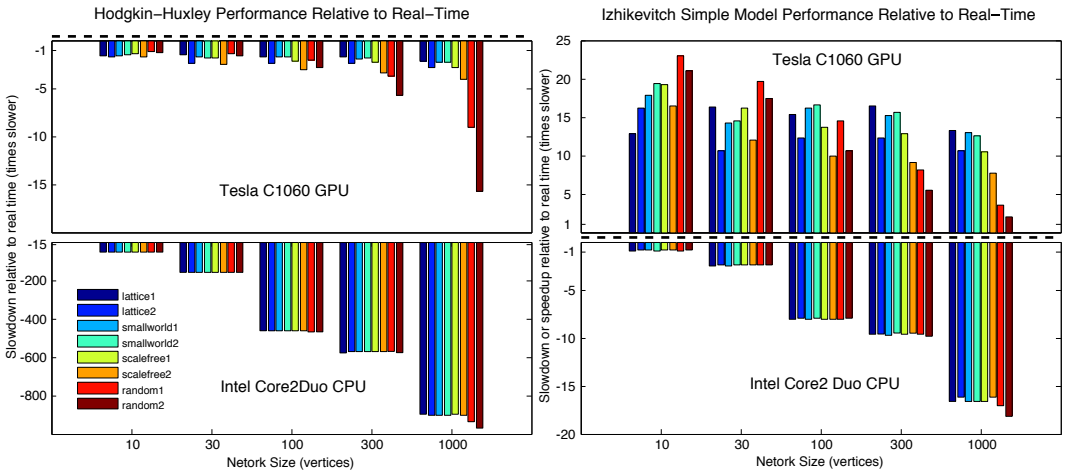

Figure 3.4: CPU and GPU benchmark results for framework simulations using Hodgkin-Huxley and Izhikevitch models of single cell dynamics. Two different network topologies for four different classes of networks were simulated (lattice, small world, scale free, and random), with network sizes (i.e. number of vertices) of 10, 30, 100, 300, and 1000 vertices simulated for each of the eight different networks. The top graphs show performance on a single nVIDIA Tesla C1060 GPU, expressed relative to the real time simulation period for each single cell dynamic model (see text), while the data for the bottom two graphs show the performance on a single core of the an Intel Core2Duo 2.5GHz processor. Negative values represent a slowdown relative to real time, while positive values represent a speedup. All forty networks were tested on both. See section 3.4.5 below for details regarding the test networks and network classes.

claims to be able to deal with real cellular neural networks of any meaningful size (e.g. on the order of tens to hundreds of cells or larger) must at the very least be able to effectively and efficiently map functional networks derived by this test set, which offers a first order approximation to the dynamics and complexity displayed by such networks.

Here we offer the foundation for such a test set: The location of vertices in physical space and their physical connections based on different connective classes. The choice of dynamic model, parameters, and functional weights is left open and up to the discretion of the individual investigator, since they are specific to the network being studied and the mapping algorithms being designed, but can be directly implemented within the framework we have developed. Test networks vary in size from 10 to 1000 vertices, covering the range of cells that can be imaged simultaneously with fluorescence microscopy. At the small end of the scale, networks on the order of 10 vertices is about the limit of existing connectivity estimation methods [MPdF05, EZJO10]. The upper end of the scale at a 1000 vertices was chosen largely due to limits of computational power available at present.

Each graph is composed of N interconnected vertices located in two or three dimensional physical space, with minimum distance constraints and other dynamic parameters as described below for each network class. The physical connectivity between vertices follow one of four different graph theoretical classifications, since there is no measurable and "real" network spatial geometry defined in this case; this is also discussed below. Vertices were positioned in geometric space randomly, but with a prescribed minimum distance between neighbors. We developed a simple algorithm to populate a physical space with N cells or nodes:

$X_1 \leftarrow$ random position in space range
$i \leftarrow 2$
while $i \leq N$ **do**
  $X_i \leftarrow$ random position in space range
  $found \leftarrow 1$
  for $j = 1$ to $i - 1$ **do**
    if $||X_j - X_i|| < d$ **then**

(i.e., if cell X_i is less than d units from cell X_j)
$found \leftarrow 0$
break for loop
end if
end for
if $found = 1$ **then**
$i \leftarrow i + 1$
end if
end while

In this algorithm, the space range establishes the dimensions of the space occupied by vertices in the network. For a two dimensional network this could be a square area of 500×500 distance units. Distance and position units are non-dimensional and can be scaled as needed. The minimum distance between vertices is a parameter, but can also be expressed as a function of the number of vertices N and the dimension of the space the graph occupies:

$$d_{min} \propto LN^{-1/D}$$

Here the minimum distance d_{min} is proportional to the length of the space L times the number of vertices N raised to the negative inverse of the dimension D. For example, a three dimensional space of length $L = 100$ could fit $N = 1000$ vertices with minimum distance $d_{min} = 100 \times 1000^{-1/3}$. This implies a minimum distance of 10. While this is the absolute minimum distance with cubic packing, when placing vertices at random this minimum distance is reduced to allow for some variability in placement. Figure 3.5 shows examples of vertex placement in two and three dimensions for three different size networks. It is important to note that with this formula, the minimum distance can be prescribed for fractional dimensions, as may be the case for some neural tissues where cell arrangement is neither flat nor fully three dimensionally filling. Vertices are numbered from the center outwards, so mapping can be performed on a subset of vertices that interact with the complete network. This is a more difficult case but a more realistic scenario, as vertices would be receiving inputs from unobserved vertices in real cellular networks due to

experimentally limited windows of observability.

Once the vertices are placed in physical space, connections are made using established graph theoretic classes. We included lattice, small world, scale free, and random classes in the test set. For each connectivity class, we generated two networks of different edge densities, one with fewer and one with more edges. We chose these four classes because they represent the major graph theoretic topologies, but of course any other algorithmically defined class can be used. The physical connectivity of a graph intuitively represents a constrained phase space on which dynamic signals propagate in both space and time (i.e. the functional connectivity topology) as determined by the network signaling framework and chosen model of single cell dynamics. Specifically, we considered the following classes and specific parameters for each:

Lattice networks. This class of networks has only nearest neighbor connections with no long distance connections. The number of nearest local connections or total number of edges can be specified before construction. In our case we limited connections for each vertex to its closest 3 and 8 neighbors.

Small world networks. This is a modification of a lattice network which includes a specified probability of long-range connections [BA99]. The probability of long range connections ranges between 0 (lattice network) and 1 (random network), but typical values are around 10 percent, meaning that 10 percent of all edges are randomly chosen. In our case we built networks of 5 and 15 percent probability of random re-wiring.

Scale free networks. These networks follow a power law connection (edge) degree distribution, with many cells having few connections and few cells having many connections. When positional aspects are taken into account, scale free networks take on some small world properties and are essentially scale free geometric graphs called apollonian networks [AHAS05].

Random networks. The study of random graphs extends all the way back to the

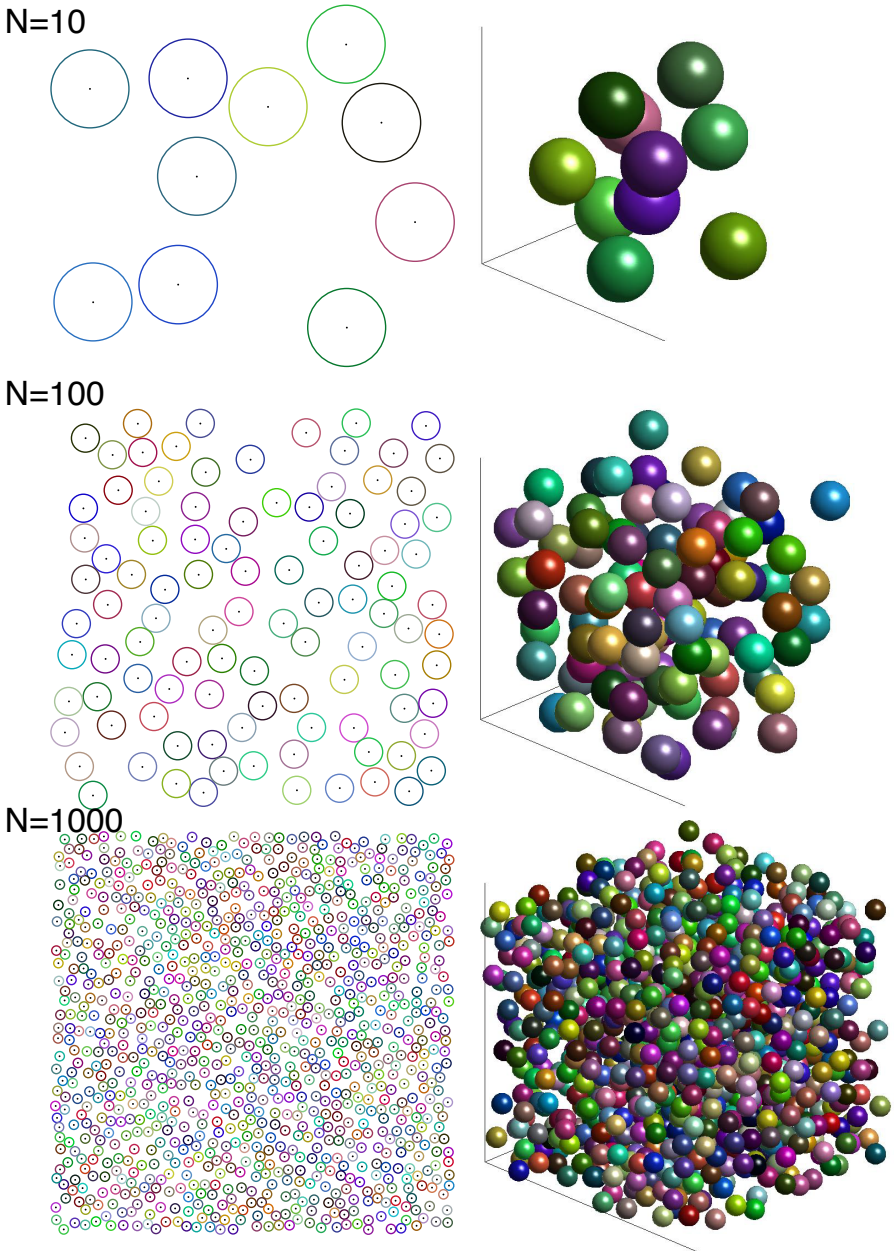

Figure 3.5: Random placement of vertices with a minimum distance constraint. Examples of groups of 10, 100, and 1000 vertices placed in two (left column) and three (right column) dimensions are shown here. Vertices were colored randomly for clarity. The minimum distance is a function of both the number of vertices and the spatial dimension. For the same number of vertices, a two-dimensional network will have a smaller minimum distance and thus be packed tighter than a three-dimensional network. See text for details.

original work of Erdös and Rényi. In a random graph, a specified number of edges are placed between randomly chosen vertices, without regards for vertex position, which has no meaning. We built random networks of 10 and 20 percent densities, meaning about $(N^2 * 10\%)$ or $(N^2 * 20\%)$ number of edges where N is the number of vertices in the network. A 100% dense network connects every vertex to every other vertex.

The different classes and densities are shown in figure 3.6, along with graph-theoretic statistics on connectivity and wiring lengths. The connections establish the physical connection between vertices or the edges along which functional connections are possible. The magnitude of the functional weights should be chosen based on the dynamic model and the specifics of the system studied. A mapping algorithm must identify the functional connections without knowledge of the physical connectivity class. Delays are defined according to the cartesian distances between connected vertices. Three-dimensional spaces generally have narrower and smaller distance distributions than two-dimensional packings. The formula presented in equation 3.25 is used to establish delays, with the speed parameter chosen arbitrarily based on the system being studied. The complete test set is 80 networks, combinations of two dimensions, five sizes, and four connectivity classes, as outlined in table 3.1. Figure 3.7 shows an example of simulated calcium response raster plots for the networks shown in figure 3.6. A mapping algorithm should be able to identify the dynamics parameters and the functional connectivity of each test network, given a chosen single cell dynamic model. Ultimately, the only kind of measured experimental data available to any such algorithm would be imaged calcium responses such as those simulated in figure 3.7 or some equivalent data for another marker of functional cellular activity. These are the practical experimental constrains that any theoretical methods aimed at mapping functional activity in cellular neural circuits and networks with single cell resolution must face. The test networks are also available for download from the authors' website (http://www.silva.ucsd.edu/Silva_Lab/Links.html).

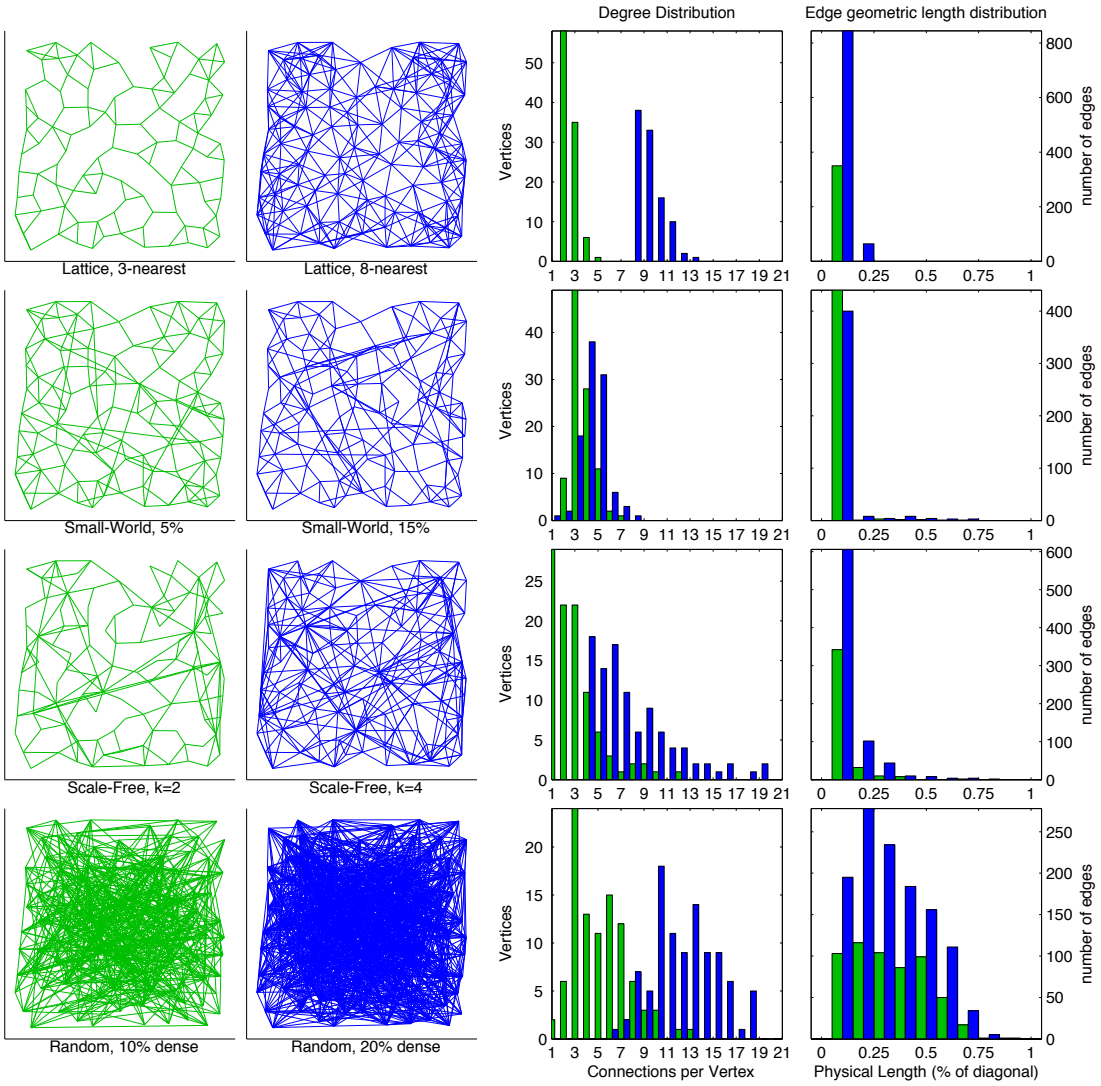

Figure 3.6: Test network topologies for a 100-vertex two dimensional network. For each of the four topologies included in the test set, two edge densities were considered, one with fewer edges or randomness (green), and a more complex one with more edges or randomness (blue; left two columns). The third column and fourth columns from the left show the degree and geometric length distributions of each network, respectively. For each network, the total number of edges and total wiring lengths can be computed from the areas under the degree and geometric length distribution plots. The higher edge density networks (colored in blue) have both higher total number of edges and total wiring length than the lower edge density networks (colored in green). For a constant signal propagation delays are proportional to wiring lengths in a network.

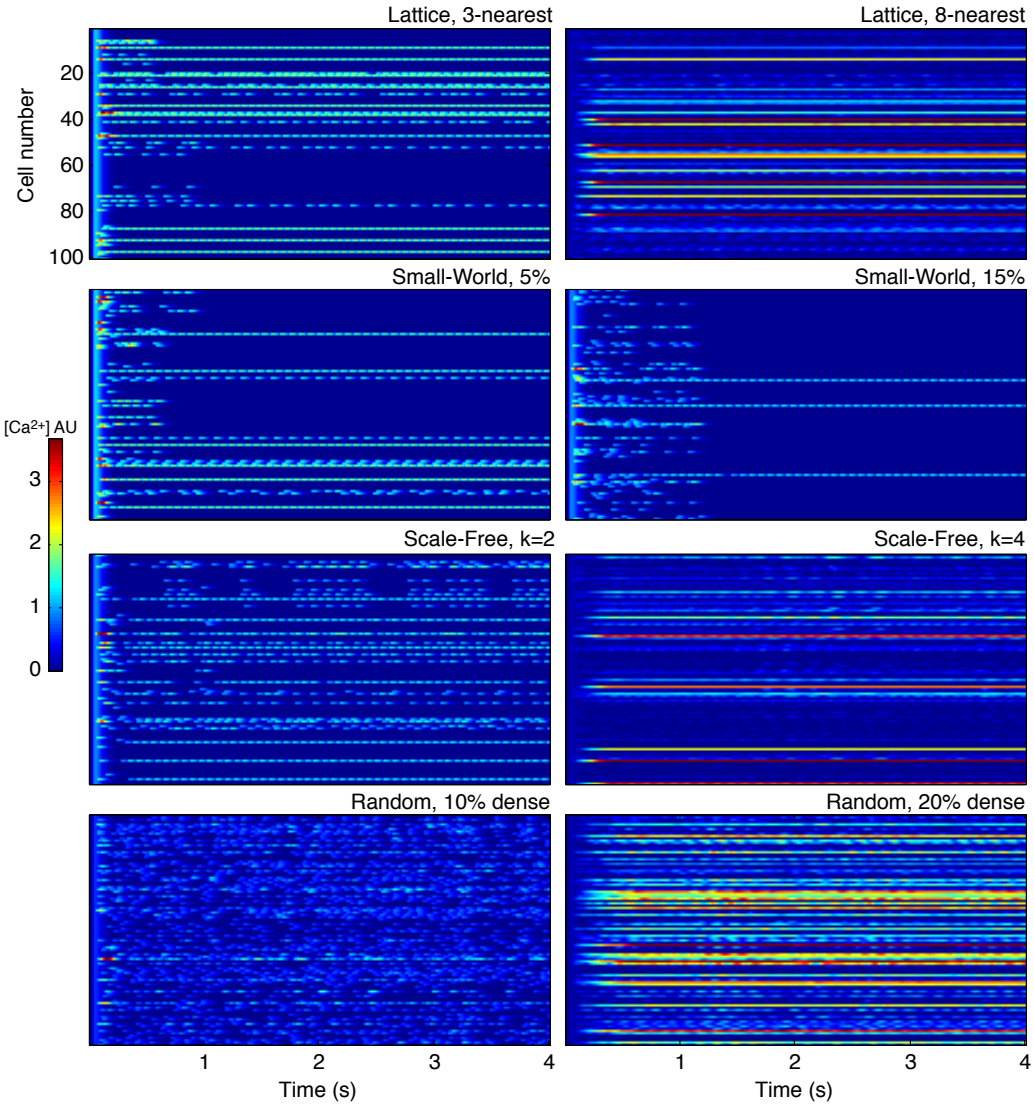

Figure 3.7: Computed (i.e. simulated) dynamics of experimentally observable calcium signaling for each of the eight networks shown in figure 3.6, arranged in the same order. Each network was simulated for four seconds using the Izhikevitch simple model, with a constant current input to generate activity. The colors of the maps encode calcium concentration, representing the approximate instantaneous spike rate of individual neurons in the network.

Table 3.1: Range of parameters specified by the test set. The current set is composed of eighty test networks spanning ranges in network size, geometric dimension, connectivity type and edge density. The choice of dynamical cell model, parameters, functional weights, observation variable, user inputs and noise levels are left to the individual user.

Parameter	**Range**
Network Size N	10, 30, 100, 300, 1000
Geometric Dimension	2D, 3D (may even be fractal, i.e. 2.5D)
Connectivity Type	lattice, small world, scale-free, random
Edge Density	low, high

3.5 Discussion

Within the study of networks, there are two opposing yet deeply interrelated processes: Simulation and estimation. Simulation of networks deals with the forward problem of making predictions using an established model and measured parameters and connectivities. The reverse problem, estimation or mapping, uses the result of actual collected data to infer, estimate, or map the parameters of a model or, for networks, functional connectivity. The framework we introduce here can be used both for simulating signal propagation in physically realistic networks, and for the reverse process of estimating or mapping unknown functional connectivities of networks. The framework is bounded by a set of rules and constraints imposed by the experimental reality of cellular neurobiological methods: Complex non-linear dynamics, limited observability, noise and uncertainty, and experimental control. It is designed around current observation and experimental capabilities, which are shifting from single neuron multiple trials, to multiple neuron single trial experiments [QP09].

Within the framework, the mathematical construction for the dynamic model describes the time course of each vertex. A general state transition representation encompasses different model types, from ordinary differential equations to state machines and Markov models, to simplified neuronal models like LIF and Izhikevitch. The choice of model will certainly affect the estimation of the dynamic parameters from the collected data; how much the single-cell dynamic model affects the estimation of functional weights is is still an open question. The proposed test set should

help address this by simulating artificial data with one model and estimating with another.

Mapping a complete functional topology is ultimately a reverse process, and will involve some combination of estimation, filtering, and optimization. While some approaches exist for estimating parameters and dynamics of single neurons [ACFK09] or small groups of neurons [MPdF05, EDS03, EZJO10], mapping large networks within biologically realistic constraints remains a challenge, and we are still a long way from establishing a complete functional connectivity map of even simple processes and tasks. Indeed, from a neurophysiological perspective, it is not even entirely clear 'what' we should be mapping or how to properly interpret such data from the perspective of deciphering the neural code. The proposed framework attempts to unify both theoretical and practical considerations as an "open standard" for the development of large scale functional topology reconstruction algorithms.

Since the goal of mapping is to identify both the dynamic parameters of individual vertices as well as the connectivity between vertices, a well-designed input control should be used to make the observation as informative as possible, provided it does not alter the parameters and connectivities of interest. Experimentally, input control can take on many forms. The dynamics of individual cells can be perturbed using methods such as optogenetics, pharmacologically using appropriate agonists and antagonists, and electrophysiology. For single cells there are different input functions that can be used and there are a few approaches describing input function design to extract the most amount of information [LBP09, BGMH07]. At the network level, the set of input functions for each cell must be designed in parallel and coordinated with observed activity in order to provide the most amount of information to the mapping algorithm.

3.6 Appendix: Test Network Generation

In the appendix we describe several neuronal and astrocytic models that we have implemented within the proposed framework. For both cases, the state dynamics (both the signaling and observation models) are described, along with the parameter sets and typical value ranges. The last section of the appendix describes

in more detail the construction of the test networks.

3.6.1 Neuronal Models

Voltage Dynamics

Linear Leaky Integrate and Fire. The simplest model that captures most (but not all) of the membrane dynamics of an idealized neuron is a linear leaky integrate and fire model (for a review see [Bur06]). For each neuron, it has one state variable $V(t)$ and four parameters modeled as

$$C\frac{dV(t)}{dt} = V_0 - V(t) + S(t) + U(t) + \mathcal{N}(0, \sigma)$$
$$V(t+) = V_{reset} \text{ if } V(t) > V_{th} \quad \text{(spike event)}$$

With parameters and typical values outlined in table 3.2. Note that in the above equations we have dropped the i notation for clarity, but each neuron has it's own state $V_i(t)$.

Table 3.2: Typical linear leaky integrate and fire model parameters. These are found in the standard literature.

Parameter	Value	Description
C	100	membrane capacitance
V_0	0	resting potential, reference
V_{reset}	10	reset potential
V_{th}	20	spike threshold voltage ($V_{th} > V_{reset}$)

$S(t)$ is the synaptic inputs into the network. This model is typical of most neuronal simulation environments and represents the simplest model of membrane voltage. It is a hybrid system in that it contains both the internal dyanmics of the cell and external input, determined from both interacting synapses $S(t)$ and user control $U(t)$. Since the voltage recovery is a linear system, the resting voltage V_0 is set to 0 and reset and threshold voltages are set relative to V_0. This model therefore has only 4 parameters: V_{th}, V_{reset}, τ, and noise σ. The time step can be very high, up to 1 millisecond per integration step, as the non-threshold dynamics are generally

linear and stable.

Fitzhugh-Nagumo Oscillator. The Fitzhugh-Nagumo oscillator is the simplest non-linear oscillator that mimics neuronal dynamics without a thresholding and re-setting step. It has two state variables, voltage and recovery, and six parameters (*c.f.* [Izh05])

$$\begin{aligned}\frac{dV(t)}{dt} &= aV(t) - bV(t)^3 - cW(t) + S(t) + U(t)\\ \frac{dW(t)}{dt} &= e(V(t) + f - gW(t))\end{aligned}$$

Typical parameter values are $a = 1, b = 0.333, c = 1, e = 0.08, f = 0.7$, and $g = 0.8$. A signaling event (i.e. spike) occurs when $V(t)$ becomes greater than 1.

Hodgkin-Huxley. The original Hodgkin-Huxley model is a membrane conductance model with four state variables and up to 22 parameters. It is the most complex and computationally intensive of the four models we explored, but widely recognized as the most realistic neuronal model (*c.f.* [Izh05]). The state variables are given by $V(t)$, $n(t)$, $m(t)$, and $h(t)$, representing membrane voltage and three gating variables, respectively. The equations are as follows and for clarity we omit the (t) notation i.e. $V = V(t)$ and denote $dV(t)/dt$ as $\dot{V}$.

$$\begin{aligned}C\dot{V} &= -g_K n^4(V - E_K) - g_{Na} m^3 h(V - E_{Na}) - g_L(V - E_L) + S(t) + U(t)\\ \dot{n} &= \alpha_n(V)(1 - n) - \beta_n(V)n\\ \dot{m} &= \alpha_m(V)(1 - m) - \beta_m(V)n\\ \dot{h} &= \alpha_h(V)(1 - h) - \beta_h(V)h \quad ,\end{aligned}$$

where

$$\alpha_n(V) = \alpha_{n1}\frac{\alpha_{n2} - V}{\exp(\frac{\alpha_{n2}-V}{\alpha_{n3}}) - 1} \qquad \beta_n(V) = \beta_{n1}\exp\left(\frac{-V}{\beta_{n2}}\right)$$

$$\alpha_m(V) = \alpha_{m1}\frac{\alpha_{m2} - V}{\exp(\frac{\alpha_{m2}-V}{\alpha_{m3}}) - 1} \qquad \beta_m(V) = \beta_{m1}\exp\left(\frac{-V}{\beta_{m2}}\right)$$

$$\alpha_h(V) = \alpha_{h1}\exp\left(\frac{-V}{\alpha_{h2}}\right) \qquad \beta_h(V) = \frac{\beta_{h1}}{\exp(\frac{\beta_{h2}-V}{\beta_{h3}}) + 1} \quad .$$

The Hodgkin-Huxley model models the kinetics of the individual currents that affect membrane voltage. It can have as many as 22 parameters, depending on how many are defined, and the list includes the Nernst potentials (E_K, E_{Na}, E_L) and maximal conductances (g_K, g_{Na}, g_L) for the potassium (K), sodium (Na), and leakage (L) currents respectively. The other parameters are found in the activation variables n, m, and h and their expressions for transition rates $\alpha(V)$ and $\beta(V)$. The input current $S(t)$ represents synaptic input current as a results of a spike in a presynaptic neuron, modulated by the strength of the synapse. This is the most complex model, and the most computationally intensive. A summary of the definitions and values of the parameters we used in modeling it are given in table 3.3.

Models of experimentally observable calcium in neurons

A more complex, non-linear model of calcium adds two additional state variables m_{Ca} and h_{Ca} and, unlike the linear model which uses spikes as inputs, this operated directly on the voltage variable [FM91]:

$$\frac{dc(t)}{dt} = \alpha\Big[g_{Ca}m_{Ca}(t)^2 h_{Ca}(t)(V(t) - E_{Ca}(c))\Big] - \beta\big(c(t) - [Ca^{2+}]_{min}\big)$$

$$\frac{dm_{Ca}(t)}{dt} = \frac{m_\infty(V(t)) - m_{Ca}(t)}{\tau_m(V(t))}$$

$$\frac{dh_{Ca}(t)}{dt} = \frac{h_\infty(V(t)) - h_{Ca}(t)}{\tau_h(V(t))}$$

Table 3.3: Hodkin-Huxley model parameters. From [Izh05]. These are the standard parameters for a cortical pyramidal neuron.

Parameter	Value	Description
C	1	capacitance
g_K	36	potassium conductance
E_K	-12	potassium reversal potential
g_{Na}	120	sodium conductance
E_{Na}	120	sodium reversal potential
g_L	0.3	leak conductance
E_L	10.6	leakage potential
α_{n1}	0.01	n-gate alpha parameter 1
α_{n2}	10	n-gate alpha parameter 2
α_{n3}	10	n-gate alpha parameter 3
β_{n1}	0.125	n-gate beta parameter 1
β_{n2}	80	n-gate beta parameter 2
α_{m1}	0.1	m-gate alpha parameter 1
α_{m2}	25	m-gate alpha parameter 2
α_{m3}	10	m-gate alpha parameter 3
β_{m1}	4	m-gate beta parameter 1
β_{m2}	18	m-gate beta parameter 2
α_{h1}	0.07	h-gate alpha parameter 1
α_{h2}	20	h-gate alpha parameter 2
β_{h1}	1	h-gate beta parameter 1
β_{h2}	30	h-gate beta parameter 2
β_{h3}	10	h-gate beta parameter 3

where:

$$
\begin{aligned}
E_{Ca}(c) &= 12.5 \ln \frac{[Ca]_{ext}}{c} \\
m_{\infty}(V(t)) &= \frac{1}{1 + \exp -\frac{V(t)+56}{6.2}} \\
\tau_m(V(t)) &= .204\left(1 + \frac{1.64}{\exp(-.06V(t) - 7.8) + \exp(.055V(t) - .86)}\right) \\
h_{\infty}(v) &= 1 - \frac{1}{1 + \exp(-\frac{V(t)+80}{5})} \\
\tau_h(V(t)) &= .5(1 - \tanh(V(t) + 81))(.333 \exp(0.15V(t)) + 7) \\
&\quad + .5(1 + \tanh(V(t) + 81))(.333 \exp(-.95V(t) - .2) + 9.32)
\end{aligned}
$$

This model is continuously differentiable, and should be paired with a non-linear neuronal dynamic model, like a Hodgkin-Huxley.

3.6.2 Astrocyte models

Simplified model of astrocytic dynamics. In the context of the current paper and in testing the framework, we introduce the simplest astrocyte intracellular signaling model that we were able to construct that still captures the salient features of intercellular astrocyte network signaling measured experimentally. We leave the use of the much more detailed model we are developing to future work. This dynamic intracellular model follows a similar framework described for neurons, consisting of internal dynamics, signaling, delays, noise, and calcium observation. We began by constructing a model of internal dynamics with only two state variables, C and D. C is the observed state variable and describes the time course of the calcium signal, acting as an integrator of the principal (and unobserved) dynamical variable D. The

system is described by

$$\begin{aligned}
\frac{dC_i(t)}{dt} &= D_i(t) - bC_i(t) + \mathcal{N}(0,\sigma^2) \\
\frac{dD_i(t)}{dt} &= -a^2 C_i(t) - 2aD_i(t) + \sum_{j=1}^{N} \omega_{ji} \cdot \max(D_j(t-\tau_{ji}),0) + U_i(t) + \mathcal{N}(0,\sigma^2) \\
\omega_{ij} &= k_1 \exp(-k_2 \|\mathbf{x}_i - \mathbf{x}_j\|) \\
\tau_{ij} &= k_3 \|\mathbf{x}_i - \mathbf{x}_j\|^2
\end{aligned}$$

The dynamical parameters $a, b > 0$ are positive, non-zero values, and the signaling global parameters $k_1, k_2, k_3 > 0$ are also positive, non-zero values. The delays τ_{ij} are a function of the square of the geometric distance $\|\mathbf{x}_i - \mathbf{x}_j\|$, as would be the case for a diffusion-based signal. The weights ω_{ij} are modeled as negative exponentials, also a function of the distance between cells, with cells farther apart having a weaker influence on neighboring cells than cells closer together. In this example, both the weights and delays are symmetric, so $\omega_{ij} = \omega_{ji}$ and $\tau_{ij} = \tau_{ji}$. The signaling function passes only positive values of the state of the $D(t)$ variable from one cell to a connected one, with the appropriate delay and weight scale. The noise term $\mathcal{N}(0,\sigma^2)$ is added to both state variables. While this simplified model fits the same general framework of equation (3.17), astrocyte signaling models are different from neuronal models in two key respects: There is a graded rather than thresholded response, and calcium is not only an observation variable, but rather directly mediates the adenosine triphosphate (ATP) and inositial triphosphate (IP_3 signaling mechanisms that represent the actual intercellular astrocyte signal. In some respects, this makes astrocyte networks easier to map, as knowledge of the calcium state is directly linked to predicting the state of the hidden variable at the next time step.

We illustrate this with a specific example (Fig. 3.8). We simulated a lattice network of 100 astrocytes, placed on a 500 x 500 arbitrary unit two dimensional grid with a minimum distance of 40 units. We stimulated a single cell for a brief time period, simulating a transient mechanical or pharmacological perturbation of the cell typical of controlled experiments [YBC$^+$09, MYBS08], and tracked the signal propagation through the network, using the above intracellular model, and compared

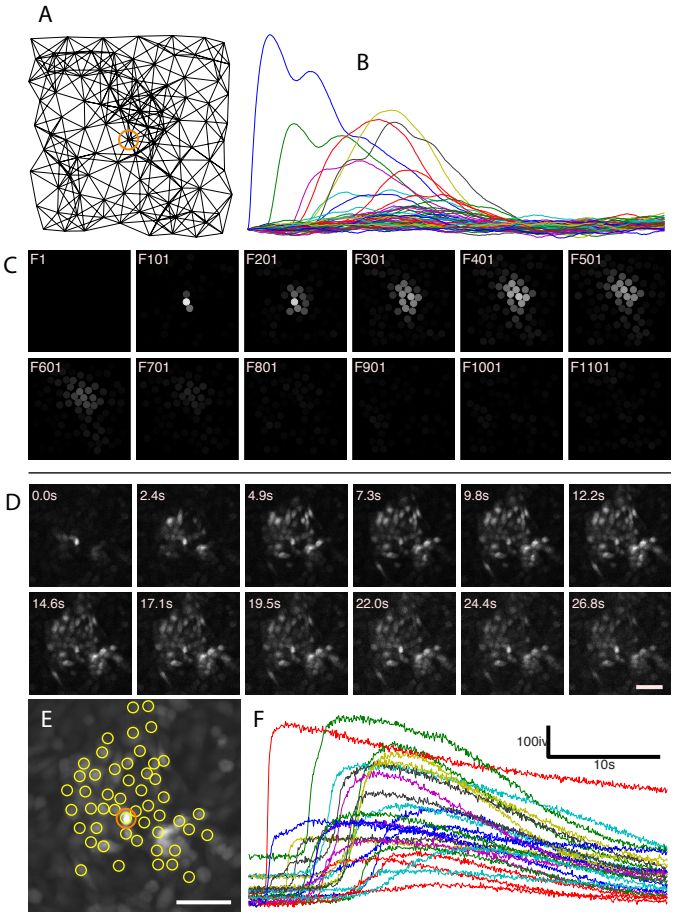

Figure 3.8: Astrocyte network model simulation and *in-vitro* data. **A.** A two dimensional, 100 vertex lattice network was modeled using a simple model of astrocyte dynamics (see text above), to produce the simulated calcium intensity traces in **B**. **C**. Following the spatiotemporal evolution of the signal through the network simulated calcium concentrations from selected frames are shown. **D**. Selected frames from a recorded movie of experimental fluorescence data collected from a spontaneously forming dissociated network of rMC-1 glial cells *in-vitro* loaded with a calcium indicator dye following selective stimulation of an arbitrarily chosen cell in the network (indicated by the orange cell in panel E). **E**. Composite average of 100 frames with manually selected cell regions of interest (ROIs) representing the physical locations of individual cells in the real cell network, shown as yellow circles. The site of initial stimulation is circled orange. **F**. Intensity profile calcium responses from selected ROIs for the experimental data. Compare to the simulated data produced by the model in panel A. The scale bar in **D** and **E** is $40\mu m$. The vertical scale bar in F. labeled as iv refers raw camera intensity values.

successive frames of the simulation with an experimental data we measured *in-vitro* using the r-MC1 glial cell line (Fig. 3.8). The simulation showed similar behaviors and activation patterns to the experimental data. The units are non-dimensional, and for simplicity the parameters were tuned manually to mimic some of the observed behavior. For example, both the simple intracellular dynamic model and the real data have rapid increases and slow decreases in calcium following activations. In both cases, early activations had rise times than later activations, which is typical of diffusive networks. Despite differences between the model and the real data in particular in the shape of the decrease of the calcium responses and less noise then the experimental data, this simple model was able to reproduce the key intercellular dynamics just by considering a finite signaling speed, temporal delays, and state noise.

3.6.3 Network size, geometry, dimension, and connectivity

Each graph is composed of N interconnected vertices located in two or three dimensional physical space, with minimum distance constraints and other dynamic parameters as described below. The physical connectivity between vertices follow one of four different graph theoretical classifications, since there is no measurable and "real" network spatial geometry defined in this case; this is also discussed below.

Vertices were positioned in geometric space randomly, but with a prescribed minimum distance between neighbors. We developed a simple algorithm to populate a physical space with N cells or nodes:

```
pick a random position for first cell X1
set i=2
while i<=N
   pick a random position Xi in space range
   if Xi is at least d units from cells X1 to Xi-1 then
      accept Xi as next cell
      increment i
   end if
```

```
end while
```

The space range describes the dimensions of the space occupied by vertices in the network. For a two dimensional network this could be a square area of 500 x 500 distance units. Distance and position units are non-dimensional and can be scaled as needed. The minimum distance between vertices is a parameter, but can also be expressed as a function of the number of vertices N and the dimension of the space the graph occupies:

$$d_{min} \propto LN^{-1/D}$$

Here the minimum distance d_{min} is proportional to the length of the space L times the number of vertices N raised to the negative inverse of the dimension D. For example, a three dimensional space of length $L = 100$ could fit $N = 1000$ vertices with minimum distance $d_{min} = 100 \times 1000^{-1/3}$. This implies a minimum distance of 10. While this is the absolute minimum distance with cubic packing, when placing vertices at random this minimum distance is reduced to allow for some variability in placement. Figure 3.5 shows examples of vertex placement in two and three dimensions. It is important to note that with this formula, the minimum distance can be prescribed for fractional dimensions, as may be the case for some neural tissues where cell arrangement is neither flat nor fully 3D filling.

3.6.4 Network Connectivity Classes

The connectivity of a derived network takes into account both graph theoretical concepts, which are being increasingly used in systems neuroscience (see for example [Spo02, BS09]), and the spatial positions of the individual cells. As such, we considered a continuum of connectivity topologies that connect vertices with a specified direction ($\rightarrow$). Since this represents functional connectivity, along each edge we encoded a relative connection strength that describes how much of one vertex's signal is propagated to another through a connected edge. We follow the usual convention that excitatory connections are positive and inhibitory connections are negative. Resultant networks will have different distributions of connectivity, and each connected edge will be a non-zero value describing the strength and polarity of

that connection.

In this paper we considered four different physical connectivity topologies, representing the major graph theoretic topologies, but of course any other algorithmically defined topology can be used. It is on these physical connectivity topologies that multiple functional connectivity topologies exist, determined by network connectivity model and the chosen single node dynamic models. In other words, the physical connectivity of a graph intuitively represents a constrained phase space on which dynamic signals propagate in both space and time (i.e. the functional connectivity topology). Specifically, the four classes of physical graphs we considered were lattice networks, small world networks, scale free networks, and random networks (Fig. 3.6).

Lattice networks. This class of networks has only nearest neighbor connections with no long distance connections. The number of nearest local connections or total number of edges can be specified before construction. In our case we limited connections for each vertex to its closest 3 and 8 neighbors.

Small world networks. This is a modification of a lattice network which includes a specified probability of long-range connections [BA99]. The probability of long range connections ranges between 0 (lattice network) and 1 (random network), but typical values are around 10 percent, meaning that 10 percent of all edges are randomly chosen. In our case we built networks of 5 and 15 percent probability of random re-wiring.

Scale free networks. These networks follow a power law connection (edge) degree distribution, with many cells having few connections and few cells having many connections. When positional aspects are taken into account, scale free networks take on some small world properties and are essentially scale free geometric graphs called apollonian networks [AHAS05].

Random networks. The study of random graphs extends all the way back to the original work of Erdös and Rényi. In a random graph, a specified number of edges

are placed between randomly chosen vertices, without regards for vertex position which has no meaning. We built random networks of 10 and 20 percent densities, meaning about $(N^2 * 10\%)$ or $(N^2 * 20\%)$ number of edges where N is the number of vertices in the network. A 100% dense network connects every vertex to every other vertex.

Chapter 3, in full, is a reprint of the material as it appears in Buibas M, and Silva GA. "A Framework for Simulating and Estimating the State and Functional Topology of Complex Dynamic Geometric Networks" *Neural Computation.*, pp. 183-214 vol. 23 (1), 2011. The dissertation author was the primary investigator and author on this paper.

Chapter 4

Parametric Functional Connectivity Mapping

4.1 Abstract

Although there are currently significant efforts to map the physical connectivity of the brain, it is less obvious how to identify and map dynamic functional networks mediated by cellular signaling, which transiently recruit only a fraction of the cells that make up the physical network. There is as yet no method that can estimate and map the functional connectivity of geometrically defined biological neural networks given the experimental limitations of current methods for interrogating such networks. Previous theoretical attempts have required access to experimentally unobservable variables, rendering them impractical as experimental tools, or have been limited to the identification of very small numbers of connected cells, typically no more than five, providing limited physiological relevance. Here, we introduce and validate a theoretical method that can accurately map functional biological neural circuits and networks with single-cell resolution in either two or three dimensions, i.e. in a plane or volume, for networks in the tens of cells by using only experimentally imaged calcium implemented using graphics processing unit (GPU) computing. The method is simple, works with virtually any cell model, and performs computations at a fraction of the cost and energy usage of traditional clusters, making mapping possible within the laboratory. The ability to map functional cellular neural net-

works provides completely new ways to think about how the brain functions under normal conditions and how it breaks down in disease.

4.2 Background and Motivation

Several newly developed methods are attempting to tackle the problem of functional connectivity connectivity parameters using multiunit cellular recordings. As is the case with most system identification or inverse problems, these new methods perform optimization of a biophysical model's parameters so that differences between simulated and observed data are minimized. One such method, partial correlation analysis [EDS03], is more statistically oriented and can provide, for acyclic or non-recurrent networks, information on direction and polarity (inhibitory or excitatory) of functional connections from spike trains. Another method [MPdF05] optimizes both neuronal and connectivity parameters of a simple linear integrate and fire deterministic model by comparing simulated output to experimental data, in this case being artificially generated spike trains. The method was validated for small networks of five neurons or less, and can handle feedback connections, though the estimated connection strengths are only reliable for indicating polarity and not relative synaptic strength. A different approach involves designing a network by modifying structural features to fit a periodic pattern of spikes [MT06]. They illustrate their design approach by optimizing a network of linear oscillators to a given signal with additional constraints for wiring length. A more recently published paper [Feo08] uses three successive optimization steps to estimate a full state-space reconstruction of a measured signal, fitting of a local nonlinear dynamical model to the reconstructed signal, and then estimating a linear model to the interactions of the individual local models. Recent work from Mishchenko, Vogelstein, and Paninski [ref not yet admitted] uses a Bayesian approach for connectivity estimation using only calcium fluorescence and a generalized linear model of spike rates. Finally, work by Eldawlatly, Zhou, Jin and Oweiss [EZJO10], employs dynamic Bayesian networks for identification of connections of small (N=10) networks using a Poisson spiking model. While identification algorithms such as these perform reasonably well on the specific test networks presented in the individual publications, the broad applicabil-

ity and usability of these methods is unproven, to a significant degree because no standardized test networks exist that are biologically realistic and incorporate all the constraints and controls in a laboratory setting. Another important aspect of a unified mapping framework and test set is model selection validation, namely how well neuronal models interpret real-world data and provide estimates of functional connectivity. The approach we present is more general because it works with a variety of single cell signaling and dynamic models not limited to spiking neurons, is portable to the laboratory that makes provisions for real-world considerations like limited observability annoys, works on a sufficiently small timescale so that affects like plasticity that modulate functional weights are not apparent during mapping, and are modular enough to make use of different mathematical optimization algorithms that are used for the weight estimation part of the mapping. Alongside the previously presented framework we hope to add structure to the field of functional network mapping by providing a unified approach and standardized test that to compare the efficiencies of different algorithms and approaches across a common standard.

4.3 Mapping Functional Connectivity

There is an important distinction between a physical network and a functional network that is analogous to the differences between a road map and a traffic map. The former presents a relatively static picture of all possible connections, while the latter is dynamic and reflects the specific flow patterns at a given instance in time. Just like cars on roads, information flows dynamically in a cellular neural network, forming functional connections that are spatially and temporally transient subsets of the set of possible functional connections dictated by the physical connectivity of the network. Since many patterns of network function can occur within the same physical network, experimental observation and quantitative mapping is inherently difficult, both from the point of producing and observing transient patterns of activity. For example, the only brain to date physically mapped in its entirety with cellular resolution is 302-neuron C. Elegans brain [WSTB86]. Despite having accomplished this in the mid 1980's, there is almost no understanding, quantitative analysis, or theory of how the underlying physical connectivity mediates functional networks that are

ultimately responsible for the behaviors in the organism. Recent progress on both observation methods like two-photon microscopy [NPY07] and network-scale perturbation techniques like optogenetics [GYB$^+$10] and glutamate uncaging [BCH10] provide important cornerstones for functional network reconstruction. However, while these methods have been used in largely qualitative investigations, no unified framework has existed to integrate both the capabilities of network observation and perturbation with existing dynamical models of neurons and glia for the purposes of functional network reconstruction. If we are to develop an insightful understanding of brain function with any predictive power, such a framework is critical to analysis and interpretation of network-scale activity data, in terms of already established individual cell-scale models and knowledge.

Building on the theoretical framework for simulating geometric dynamical networks [BS11], we present a novel technique to map the functional connectivity of cellular neural networks using only experimentally measurable data. This method works with virtually any dynamic cell model, and can make use of any type of data that is informative of cell dynamics, like calcium concentration or membrane voltage. We outline assumptions, data collection requirements, describe how to map using any model, estimate its performance, and briefly highlight some of the numerical and mathematical limitations of the method. Unlike some previous efforts [EZJO10, MPdF05, Feo08], this approach encompasses a broader set of models, and should have broad applicability to the many experimental efforts towards understanding network function across different organisms.

We define a dynamic network to be a collection of connected vertices whose individual state vectors evolve temporally based on their current state $\mathbf{y}_j(t)$, external inputs $\mathbf{s}_j(t)$, and experimental controls $\mathbf{u}_j(t)$, according to the transition rule $H_j(\cdot)$:

$$\mathbf{y}_j(t + \Delta t) = H_j(\mathbf{y}_j(t), \mathbf{s}_j(t), \mathbf{u}_j(t)) \tag{4.1}$$

External inputs represent the sum of the weighted (ω_{ij}) and delayed (τ_{ij}) states from

all other vertices connecting into j:

$$\mathbf{s}_j(t) = \sum_{i=1, i\neq j}^{N} \omega_{ij}\mathbf{y}_i(t - \tau_{ij}) \tag{4.2}$$

From some limited observation of at least one of the components of the state vectors $\mathbf{y}_j$ the goal is to estimate the function-transmitting weight set ω_{ij} and possibly the delays τ_{ij} for all i, j combinations in a network of N vertices. All neuronal models published to date can be expressed in the above framework, with the rule $H_j(\cdot)$ either being a state transition map or a differential equation flow (framework REF). Because the full state vector $\mathbf{y}_j(t)$ is not completely observable, the transition rule $H_j(\cdot)$ is required to make any estimates of the hidden variables, including the functional weights. As such, the rule $H_j(\cdot)$ must be informative of the entire state vector $\mathbf{y}_j$. Otherwise state, given enough measurements of a part of $\mathbf{y}_j$, the entire complete vector $\mathbf{y}_j(t)$ can be estimated within some certainty for a finite observation period $t \in \{0 \dots T\}$. We refer the reader to [BS11] for full mathematical details on the mathematical framework.

Given this state based framework, in order to map the unknown functional connectivity of a network, a minimum observation period of at least N time steps for each cell in a network of N cells is required. This is a minimum requirement and not a guarantee for successful estimation, but it does mandate that the total number of measurements scales with the square of the number of cells in the network. The frequency of measurement is governed by sampling rules from control theory and depends on the choice of model, with more complex models generally requiring faster sampling. Another requirement is that the signals are well mixed, meaning that recordings from any two vertices do not resemble each other. The condition number of the observation matrix that has as rows the observation vectors for each vertex in the network is a simple test determining the suitability of the data. If this condition number is high, then the observations are likely to be very similar and making it difficult or impossible to discern between contributions of individual vertices to the dynamics of a target vertex. Experimental control $\mathbf{u}_j(t)$ may be used to add variety to the observation signals and lower the condition number of the observations. With

the use of optogenetics or uncaging, this level of individual cell control is now possible for small cell networks, and use of these techniques will serve to make the observed activity more informative, ensuring that not all cells in the network have the same temporal activity patterns. In summary, these requirements simply state that the observed activity of cells has to be informative for any sort of estimation or mapping to be meaningful. While not guaranteeing success, these are necessary conditions for mapping of virtually any neuronal, astrocytic, or mixed network, using virtually any model of cell dynamics from non-invasive observations of many cells' activities.

If the states of each cell can be independently observed, the network can be simplified from a multiple-input multiple-output (MIMO) system to a series of multiple-input, single-output systems (MISO). This decomposition, made possible by the observability of the dynamics of many cells using fluorescence microscopy, greatly simplifies the estimation task reducing the dimensionality of the problem from $O(N^2)$ to a $O(N)$ parameters. More importantly, it splits the mapping process into N separate and independent estimation steps that can be executed in parallel, speeding up computation. To achieve this simplification, we need to estimate the output of each cell from its observed activity. In the case of neurons, this involves estimating spikes from calcium; for astrocytes, released ATP from observable calcium. Relatively independent of the cell's dynamic model, Monte Carlo-based techniques like particle filtering or path integrals compute a probabilistic estimate of a hidden quantity from an observable quantity using the transition rule $H_j(\cdot)$ and calcium observations[VWP$^+$09, CIY05]. While other techniques like deconvolution [YF06] may be less computationally intensive, the particle filtering method is generic enough to work with virtually any transition rules of $H_j(\cdot)$ including non-linear models, and thus any cell model, including heterogenous networks made of differently modeled cell types. Figure 4.1AB outlines the output estimation and decomposition steps, the first in the mapping process.

Once the outputs of all observed cells are estimated, we consider each cell in the network individually, seeking the weights that modulated the outputs of the other cells in such a way that, when summed as in equation 4.2, and simulated through the transition rule in equation 4.1, most closely reproduce the observed dynamics of the individual cell. Alternatively stated, for each cell in the network and each time point

of observation, we seek to minimize the differences between the actual observed dynamics, and the expected observed dynamics for set of weights modulating incoming signals to a given cell. As illustrated in Figure 4.1B, the problem becomes a data fitting exercise, where the parameters are the weight set for a given cell, the model is the transition rule and the data are the observations for that cell and outputs of all other cells.

Because the data and the models are both nonlinear and non-Gaussian, some form of numerical optimization is required to 'fit the curve' or find the weight set that, when simulated, can best reproduce the observations. The no-free-lunch theorem [WM97] states that there is no optimal optimizer for all problem classes, so some optimizers will work better with some models than others. From a practical and computability standpoint, we have developed a modified parallel stochastic search algorithm and found it to be the easiest to program and most reliable across different cell models and network sizes. This data-fitting approach does not exclude other optimizers: since the problem is framed in terms of similarity to observation, the problem can be tackled by multiple optimizers in parallel with the best solution being the closest to the observed data. This is shown in figure 4.3FG, where two optimization algorithms map a 30-neuron network. The combined performance graph in 4.3H selects the best fitness scores for each vertex incoming weight estimates, resulting in an improved match with actual connection weights.

To evaluate this mapping approach, we consider the twin problem of mapping simulated networks of known connectivity using only observed calcium dynamics and comparing functional weight estimates with actual weights. Since most somatic neuronal calcium dynamics are only indicative of spike events, with subthreshold potential changes typically not manifested as calcium concentration changes, information available to the mapping algorithm is significantly truncated, so functional weights can only be estimated within a certain tolerance. Here we test several cases to evaluate some of the factors affecting mapping performance: network connectivity class, network size, choice of dynamic model and optimizer, and prior knowledge of physical connectivity.

We first start with a network of 100 cells of a known physical connectivity, and modulate the connective weights to produce 3 functional networks of random,

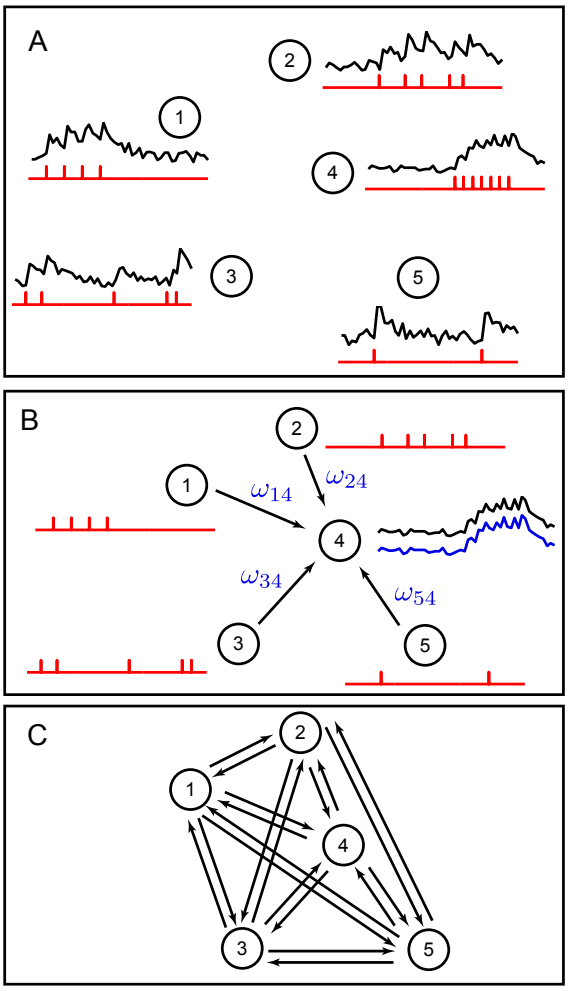

Figure 4.1: Functional Connectivity Mapping. A. The observable dynamics (black lines) of a group of cell is used to estimate the possible output of each cells (red lines), using the dynamical model $H_j(\cdot)$. In the case of neurons, we estimate spike outputs from calcium. B. For a given cell in the network, we search for the combination of incoming weights (blue lines) that, when modulating the outputs of the other cells, simulate an output that is closest to the actual observed dynamics of the target cell. This is typically achieved through some optimization process, which seeks to minimize the differences between data and simulation through a series of successive steps. C. This process is repeated for every cell in the network, producing the full, bi-directional connectivity matrix for the entire observable network.

scale-free/small world, and random connectivity patterns, shown in Figure 4.2ABC. The physical network is shown in gray, with the functional connections shown in red and randomly selected in both amplitude and polarity (inhibitory and excitatory). We model vertices as Fitzhugh-Nagumo neurons with observed calcium, and for each network, we provide a random stimulus to generate activity and produce a calcium signal. Given only the observed calcium signals in shown 4.2DEF and knowledge of the physical network (gray map in 4.2ABC), we estimate the complete functional connectivity map of each network, estimating the functional weights and polarities of each possible connection in the physical network. We compare the estimated 1280 weights with the actual values and use the correlation coefficient R-value as a measure of mapping accuracy. As shown in figure 2GHI, we see that R-values greater than 0.72 indicate a reasonable reconstruction of the original functional weights. Stated otherwise, an R-value of 0.7 can reconcile 50% of the variance of actual values. Since weights range from -1 to 1, an R-value of 0.7 means that the vast majority (99% in the Gaussian distribution case) of weights can be estimated with correct polarity, since 50% of the variance is effectively half the weight range. Therefore, a R-value of 0.7 can be established as a minimum threshold for any sort of weight mapping, meaning that at least the polarity of the connection (inhibitory or excitatory) can be reliably established for most weights.

The mapping algorithm can reasonably reconstruct all three functional network architectures in figure 4.2. The lattice network of figure 4.2A, the least dense (fewest total functional edges) of all three, was the easiest to map with the highest R-value. The random network (4.2C), was the densest, but not as difficult to map as the small-world/scale-free [AHAS05] in figure 4.2B, which had the lowest R-value. The network in 4.2B had the widest degree distribution of the three networks, meaning that there was a wide range of number of connections into a given cell. Overall, the mapping performance is similar across the three architectures and, as expected, the inhibitory portion (left part) of the performance graphs is the least accurate as most calcium models do not reflect hyper-polarization of neuronal membranes as a result of inhibitory post-synaptic currents.

In addition to reconstructing functional network architecture, the mapping algorithm performs evenly across different dynamic models. In figure 4.3AB, we

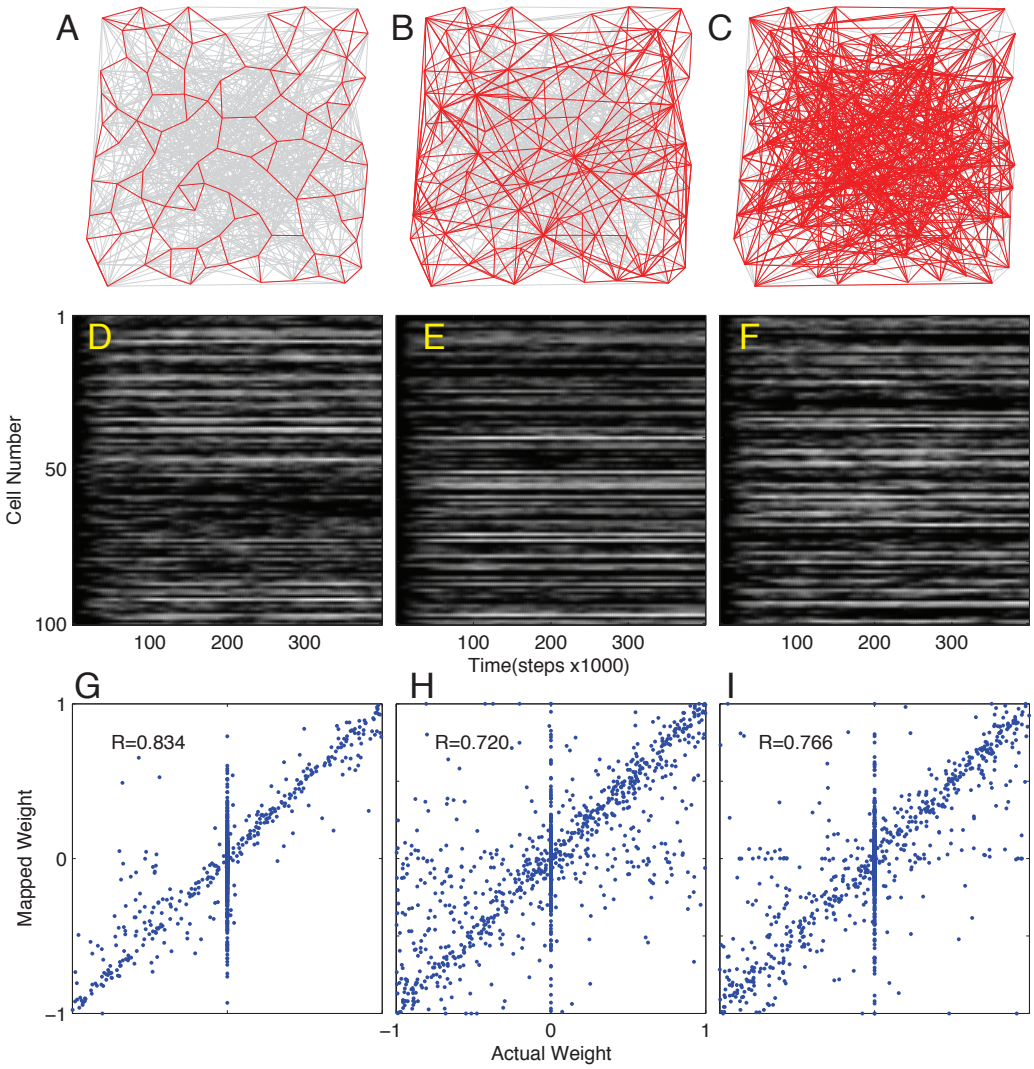

Figure 4.2: Mapping of different functional networks within the same physical network. Three functional networks of different topologies (A-lattice, B-smallworld/scalefree (Apollonian), C-Random) are shown in red embedded within the same physical network (gray lines). Functional weights of known values were used to generate the observable calcium dynamics shown in D, E, F, for each network above. Using knowledge of the physical connectivity (gray network), we mapped the functional connectivity from the computed calcium signal, and compared it with the known values that were used to generate the calcium signal. Plots of the comparisons are shown in G, H, I, with the correlation coefficients measuring the accuracy of the mapping. A maximum possible R value of 1 means that the estimate equals the actual weight value.

mapped the network of figure 4.2B using a linear integrate and fire (LIF) and Hodgkin-Huxley (HH) dynamical model. Interestingly mapping using the simplest neuronal dynamical model, LIF, was marginally worse than mapping with the most complex neuronal dynamical model, HH. We suspect the thresholds and resets in the LIF model add difficult hard boundaries in the cost function or similarity to observed data, where small changes in weights have no effect on observed dynamics. While the HH model is considered a stiff system of differential equations, the firing threshold is a softer boundary and the optimization path is somewhat smoother, leading to better overall weight estimates. Being able to work with a variety of models, this approach can be used with virtually any cell type, as well as networks of mixed cell types and dynamic models, provided the observations are informative as defined above.

The biggest factor affecting model performance is the size of the observed network. The more connections into a cell, the more combinations of weights can produce the same observed dynamics. The same relationship holds for any nonlinear parameter or state estimation algorithm, where larger systems become more difficult to estimate as multiple combinations of parameters or states can explain observed data. This is illustrated in figure 4.3CDE, where mapping performance of test networks with completely unknown physical connectivity is shown for networks of 10, 30, and 100 vertices. As expected, mapping performance drops off with increasing network size, and as with the networks of figure 4.2, the inhibitory connections are more difficult to estimate (lower left corner of the correlation charts). Additionally, with increasing network size, the slope of the correlation graph is reduced as optimization seems to favor many weaker connections over fewer stronger connections to explain the observed data. This effect may be interpreted as the effect of the concentration of measure theorem [vitaly milman, gromov, talagrand], the dual of the curse of dimensionality. In the context of dynamic networks, an interpretation of this theorem is that as the number of functioning inputs to a given cell increases, the cell's dynamic behavior ceases to be dynamic and tends to remain static. With a large set of inputs of possible inputs, it not only becomes more difficult to search the solution space, but the likelihood of a unique solution for a input weight set is also reduced, and the estimate will likely converge to zero as the network size increases. It is not mathematically feasible for a cell to receive meaningful functional input

from large numbers of cells, and it is likely that functional input patterns to any given cells are low-dimensional. This result suggests that functionally, a cell cannot make sense of too many inputs; its outputs won't change if it receives inputs from too many cells at a time.

When knowledge of the physical connectivity is available, the mapping can be dramatically improved, as shown in figures 4.3IJ. Here we provide the algorithm with the physical connectivity of the 100 and 300 vertex networks and the performance is dramatically improved as a result of the decreased dimensionality of the weight search space. In an experimental setting, the physical connectivity can be established after the dynamical data is recorded, and that information can be used in estimating functional connectivity.

The mapping algorithm uses and builds on existing knowledge to estimate the higher-order network-level causal connective parameters that are key to understanding brain function and disease, and, if successful, should validate not only the many hypotheses of network architectures but also the dynamical models of individual neurons and glia. This approach incorporates a broad swath of current knowledge and techniques in neuroscience. Being simulation-based, the mapping algorithm can accommodate any type of recorded data and model and estimate the hidden quantities of interest, much like fitting a line or distribution through a set of points to obtain a slope or average and standard deviations. We've designed the mapping and simulation algorithms to work with any neuronal or glial dynamical model. The mapping algorithm can incorporate useful knowledge like physical connectivity and experimental control and work with virtually any optimization algorithm to provide estimates of functional weights. Our approach has proven superior to past network estimation algorithms, being able to handle multiple dynamical models and larger networks with improved accuracy [OWB05, Ioa07, EZJO10, EDS03, MPdF05]. Looking forward, we plan on probing real neural networks and validating functional estimates from this method with voltage measurements from a few neuron pairs [Cal04, SGL$^+$05].

From a theoretical standpoint, improved optimization algorithms for larger dimensional problems of more than 30 parameters, and the design of optimal inputs to maximize information in observed calcium will be key areas of research. Search in non-linear high-dimensional spaces is a difficult enterprise and there is no one

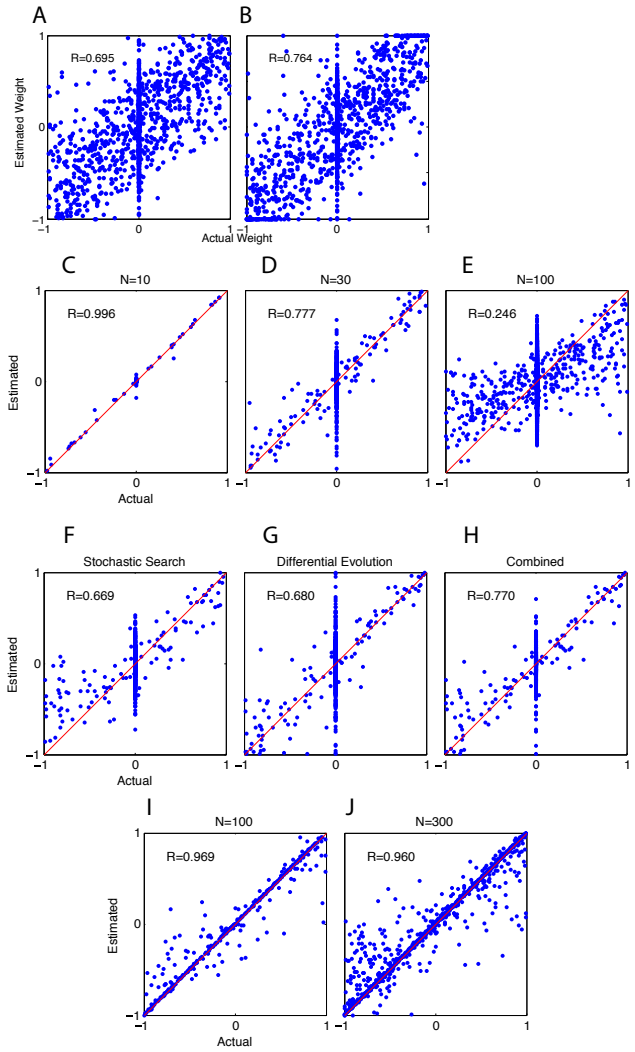

Figure 4.3: Factors affecting mapping accuracy. Two models spanning the complexity range of neuronal models were used to simulate and map: A-linear integrate and fire, B-Hodgkin-Huxley. Little substantive difference is observed between the two, indicating that mapping is relatively model agnostic. Network size or dimensionality of the estimation problem is the largest factor affecting mapping performance, as networks greater than 30 neurons with unknown physical connectivity become increasingly difficult to estimate (C, D, E). However, multiple optimizers can be used in parallel to improve estimates (F, G, H). Knowledge of the physical connectivity in a network can dramatically improve mapping accuracy, as the dimensionality of the optimization problem significantly reduced (I, J). Networks larger than 100 can be mapped with good accuracy, given knowledge of the physical connectivity.

optimizer that will handle all mapping tasks. More importantly, the flattening of the weight estimates in larger networks suggest that mathematical considerations like concentration of measure are affecting not only estimates, but may also provide insight into functional connectivity patterns themselves. It may be that cells in a dynamic network may only exhibit patterns of relatively few functional connections at any given time, so as not to overwhelm and make the cell static. These simulation results, coupled with the mathematics of high-dimensional spaces and functions, may form the basis for principle of stable functional networks, with a constraint of relatively few active functional inputs into a cell at any given time.

More formal proofs of the requirements and mapping algorithm are available as supplemental material. All computations were performed using a combination of MATLAB and nVIDIA Tesla C1060 computing processors. All code and figure examples are available upon request from the corresponding author's website at silva.ucsd.edu.

Chapter 4, in part is currently being prepared for submission for publication of the material. Buibas M and Silva GA. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Non-parametric Mapping Approaches

The parametric modeling framework and mapping approaches of chapters 3 and 4 work if there exist state-space models of the cells that form the networks. However, this is not always feasible, as parameter estimation techniques are often difficult to implement and pose challenging optimization problems. Additionally, a full state-space model is not always required to reproduce the observed dynamics of and infer the functional can causal connectivity between cells. In this chapter, I am reproducing an early version of a paper that demonstrates a nonlinear, autoregressive modeling approach for estimating input-output relationships for systems with exogenous outputs (NARX). Here, I present an enhanced version of a technique often use in black-box systems to estimate applied current from neuronal voltage measurements, as well as estimating spikes or spike rates from fluorescent calcium measurements. Using experimental data from mouse cortex pyramidal neurons, we can estimate with about 90% accuracy the applied current using only voltage measurements, effectively providing the equivalent of a current measurement from a voltage clamp, without silencing the cell, as is the case with a voltage clamp. Calcium data is currently being collected and, based on simulated studies, I expect that we will be able to estimate with high ($> 95\%$) accuracy, spike rates from somatic calcium, with same or higher accuracy than existing methods like particle filters or deconvolution, both of which require some form of dynamic model. As the data is still being collected,

this chapter is very much a work in progress.

5.1 Abstract

We develop techniques for estimating hidden or difficult to measure time-series quantities from observable and easier to measure ones using non-parametric and non-model based methodologies. We estimate input current from observed membrane voltage and spikes from calcium in both simulated and actual neurons. We base our estimation entirely on previously collected data of current/voltage and spike/calcium pairs, which we use as the basis for estimating current and spikes from voltage and calcium, respectively. We can achieve accuracies in excess of 90% in most cases, making this technique useful for functional connectivity estimation in larger cell networks, especially when dynamical models are not available or standard parameter estimation techniques are not feasible. We show performance of the technique and the requirements for data collection for these typical neuroscience experimental measurements.

5.2 Introduction

Functional or effective connectivity between cells in neuronal networks is fundamental to understanding how information is processed and propagated in neuronal networks, and how behaviors emerge from external stimuli and cell-cell interactions. Not directly observable, functional connectivity describes the effect of one cell's function onto another; its estimation requires observation of the two cells' activities over some period of time. Time-series quantities that describing the inputs to and outputs of a cell within a network are fundamental to the estimation of functional or effective connectivity between the cells in that network.

For the case of neuronal networks, *inputs* are typically post-synaptic applied currents that drive the membrane dynamics and can result in the generation of action potentials or spikes that are the *outputs* of a cell, propagated to other connected cells. Direct measurements of applied currents and spikes at the cell and network

level respectively pose significant experimental challenges that ultimately undermine the ability to estimate functional connectivity.

The membrane dynamics of a neuron are driven by the sum of post synaptic currents reaching the soma and leading to the generation of action potentials. Current measurements of neurons in networks are disruptive, since voltage clamps would effectively silence the neuron by holding it at a specific sub-threshold voltage, and prevent it from generating action potentials. Without propagating action potentials, the observed neuron does not propagate any signals into the network, affecting connectivity estimates. Thus, a method of estimating current from voltage, without silencing the observed neuron, is needed for unperturbed network activity observation.

Fluorescent calcium indicators allow relatively non-invasive network-level observation of spikes in neuronal circuits with cellular resolution, something that is not possible with electrodes that measure membrane voltages. In two-photon microscopy, the calcium activity of awake and aware mice is observable as the animal performs a given task [SGHK03, GH07]. While current measurements of individual neurons measure inputs to that neuron, calcium is a good indicator of spikes, the outputs of many individual neurons. Cytosolic calcium increases rapidly as voltage-gated Ca^{2+} channels are opened during action potentials, and decreases slowly as it is pumped out into stores or outside of the cell [HIS96]. The time course of calcium, typically described by an exponential decay, is not always the same for all neuronal types, and not always linear in nature. There are many models of calcium, with different parameters for different cell types. For connectivity estimates, the focus is not the specific calcium dynamics, but more on the recovery of the individual spikes or output signals from a particular neuron.

Both of the tasks present the experimenter with a filtering problem: extracting a quantity of interest from a more easily observable quantity. In many cases, filtering achieved through the use of model-based techniques, like Kalman, Wiener filters in the case of linear systems with Gaussian noise, and sequential Monte-Carlo methods like particle filters, for non-linear, possibly non-Gaussian cases [VWP$^+$09, YF06]. These have been used previously on both voltage-to-current and calcium-to-spikes filtering problem and, like any other model-based technique, require a description of

the transition rules that take the system from one time point to the next. Model-based techniques pose several challenges when applied to filtering problems. First, there choice of which model to use is not a trivial one, and there are no obvious algorithms for choosing a model over another for a given cell. There are dozens of neuronal models, each with many functional forms and parameters, and each developed for a specific cell and experimental protocol. Second, there is the problem of estimating model parameters, which inevitably involves some form of optimization that chooses the parameter set that best matches the observed data. Optimizer choice is not a trivial problem, especially for an experimentalist, requiring expertise in the field (there are over 200 optimization algorithms posted at optimization.org). Additionally, for complex models requiring many parameters, the curse of dimensionality plagues (See bellman ref) both the optimization algorithm and the quality of the solution: it takes a long time to converge to an acceptable parameter region and there is no guarantee that the optimized parameter set will yield a model that will accurately filter new data, or be robust enough to handle data from many different cells. When applied to biological data, model-based methods may not be the ideal approach to filtering the complex non-linear systems that are living cells.

Given these challenges, a non-model, non-parametric approach may be better suited for the filtering problem of estimating an input from an observable output. For the task of estimating connectivity, we are less interested in the internal dynamics of a cell that a full model-based estimation would provide, than in the input output relationship of signals into and out of a cell. Non-parametric approaches have been used in the past in systems that were not modeled, but whose input-output behavior is repeatable and relatively stable. Borrowing from control systems, we will use non-linear autoregressive models with exogenous inputs (NARX). These models achieve the same filtering goals as model based approaches, while being completely independent of choice of model, optimizer, or needing much expert adjustment. Sometimes referred to as a black-box model, NARX models have been successfully in many applications where state-space modeling is difficult or cannot be realized concisely in equation form, like chemical plant dynamics, nonlinear control systems, for example [SH03, BS98] NARX models have the advantage of being entirely and uniquely data dependent, without the need for any other external assumptions or expertise. They

completely avoid the problem of optimization, or tuning a model to fit the data, requiring only three parameters that set up the estimation algorithm. In the context of biological and neuronal data, the NARX approach can achieve similar or better performance than current approaches, specifically for problems related to network activity and connectivity.

To use a NARX approach for estimating current from voltage and spikes from calcium, a calibration period is needed, when both inputs and outputs are measured for the cell of interest. During this calibration period, a known injected current is added to drive the membrane voltage of a neuron and both current and voltage are measured, typically with a current clamp electrophysiology setup. After the calibration period, the data is used to build a table of input-output pairs. Then, while measuring only the output variable, newly acquired data is compared to table values to estimate input, the quantity of interest. For calcium recordings, calibration is achieved by measuring both voltage or spikes and calcium concentration or fluorescence. After calibration, only fluorescence is required to estimate spikes or spike rates. We illustrate this in figure 5.1. This method works remarkably well for these two problems, requiring relatively little calibration data and setup of the input-output table. Using both experimental and simulated data, we show that we can recover at least 90% of the original input signal for both voltage-to-current and calcium-to-spikes filtering problems. We believe that this method has broad applicability to many time-series data analysis applications in neuroscience, and is especially well suited for functional connectivity estimation.

5.3 Results

The nonparametric approach is applied to two common types of estimation in neuroscience, specific to modern imaging and network reconstruction applications: applied current from observed voltage and spikes from cytosolic calcium. For both problems, we illustrate the method on both simulated and experimental data, with similar results.

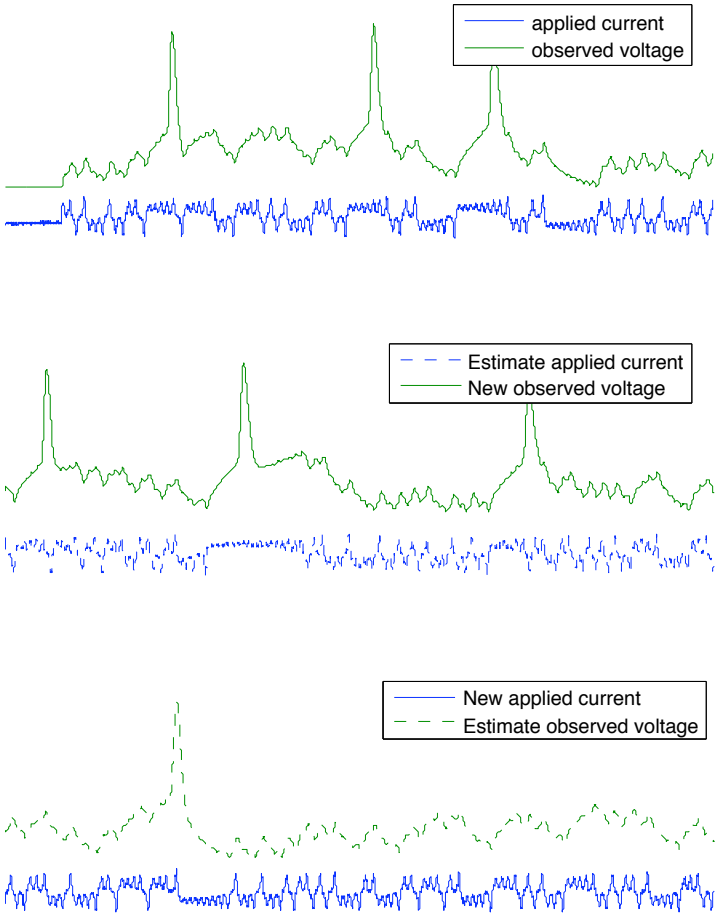

Figure 5.1: Nonparametric general problem statement. Top) Given two measurements of a known and chosen control current and observed voltage, we want to build a model that Middle) can estimate the unknown control from voltage measurements and Bottom) estimate voltage from a known control current.

5.3.1 Applied current from recorded voltage

This is a common task for electrophysiology recordings, as applied current is the sum of all post synaptic currents reaching the cell body. By observing spike activity in the entire network and the voltage of one cell in the network, functional connectivity can be inferred. Results are encouraging, and a brief discussion is offered in the figure captions.

Choice of stimulus current can affect performance, and should be selected based on expected stimulus of that neuron within the circuit containing it. Step functions are typically not informative, though noise is probably worst case. The stimulus must exercise the neuron, covering the entire range of possible stimuli the neuron would be subjected to in normal operation. Here we use a noisy stimulus and a chaotic current.

Figure 5.2 shows the performance for a simulated HH neuron driven by a noisy input stimulus. This input contains the most varied possible signal (highest entropy) received by a neuron and represents the lower bound of performance using this method. Figure 5.3 shows a chaotic but of lower entropy signal driving the same neuron, with much higher performance. The more ordered the driving current the better the prediction.

Figure 5.4 shows the effect of embedding dimension (how many past values to look when making a prediction), and the delay between past values when making the prediction. The choice of embedding dimension and delay is made to maximize R-values, a measure of estimation accuracy. Figure 5.5 shows the effect of collected data on measurement accuracy. As expected, the more data we use, the better our estimates. In this case, roughly 30,000 data points were used to build the lookup table and obtain accuracies of 90% or higher in the noisy current case. For voltage measurements this represents a few seconds of data.

Data for a real neuron, shown in figure 5.6, shows similar predictive performance as in the simulated neuron of figure 5.4, with only a slight degradation in the R-value, likely due to the actual experimental noise measurements.

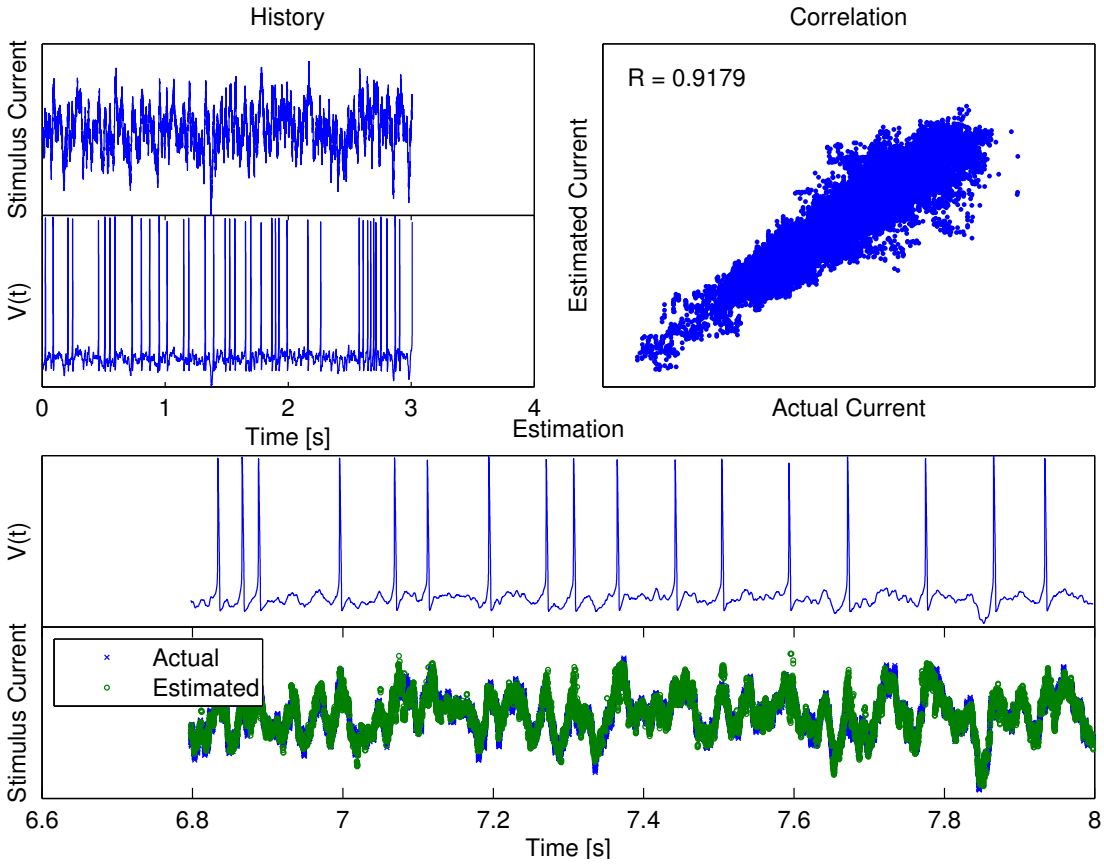

Figure 5.2: The noisy stimulus current is estimated from the voltage. The data was simulated with the Hodgkin-Huxley model with a noisy stimulus current. Then the first 3 seconds (75,000 time steps) of current and voltage data was used as history to build the input-output table (top left). Then an estimation of the current from voltage was done over 1.2 seconds (30,000 time steps) (bottom) and compared to the actual current by calculating the correlation coefficient R (top right). In this example the time delay was $\tau = 1.2$ ms, and the embedding dimension was 10.

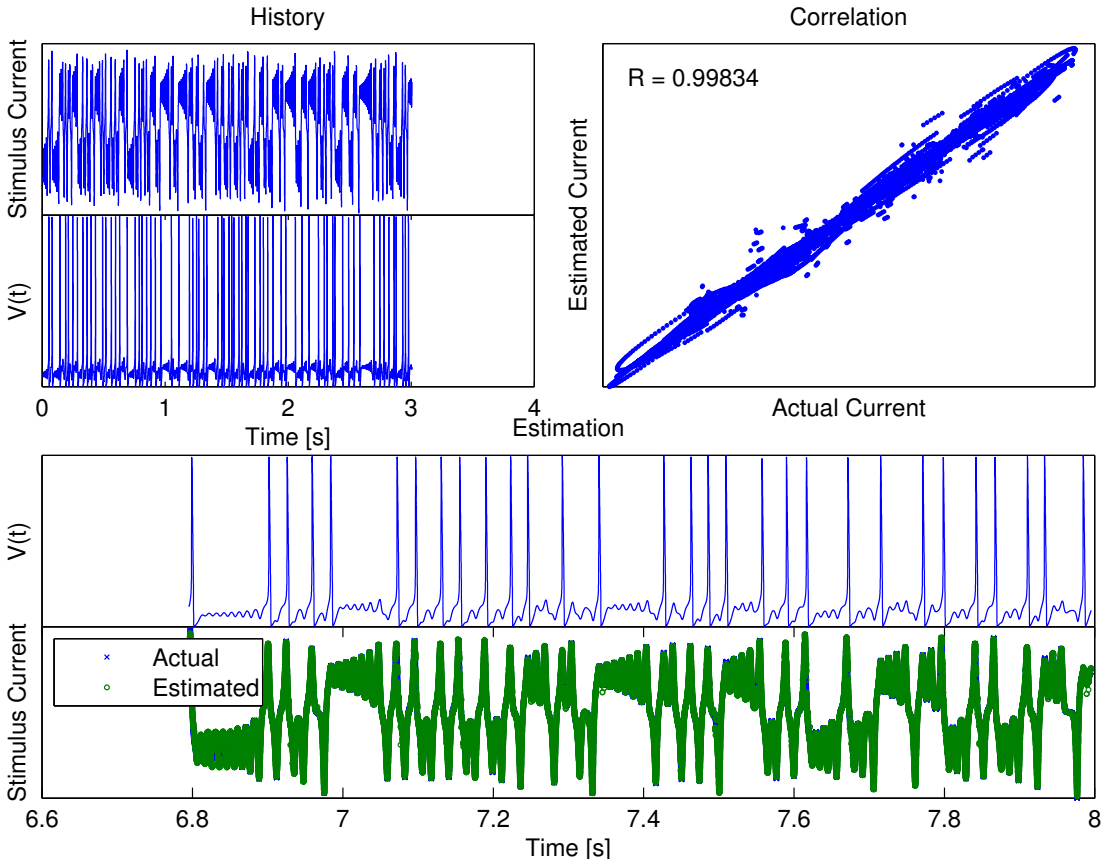

Figure 5.3: The Lorenz stimulus current is estimated from the voltage. The data was simulated with the Hodgkin-Huxley model with a Lorenz stimulus current. Then the first 3 seconds (75,000 time steps) of current and voltage data was used as history to build the input-output table (top left). Then an estimation of the current from voltage was done over 1.2 seconds (30,000 time steps) (bottom) and compared to the actual current by calculating the correlation coefficient R (top right). In this example the time delay was $\tau = 2.4$ ms, and the embedding dimension was 6. $\{-3\tau, -2\tau, -\tau, 0, \tau, 2\tau\}$

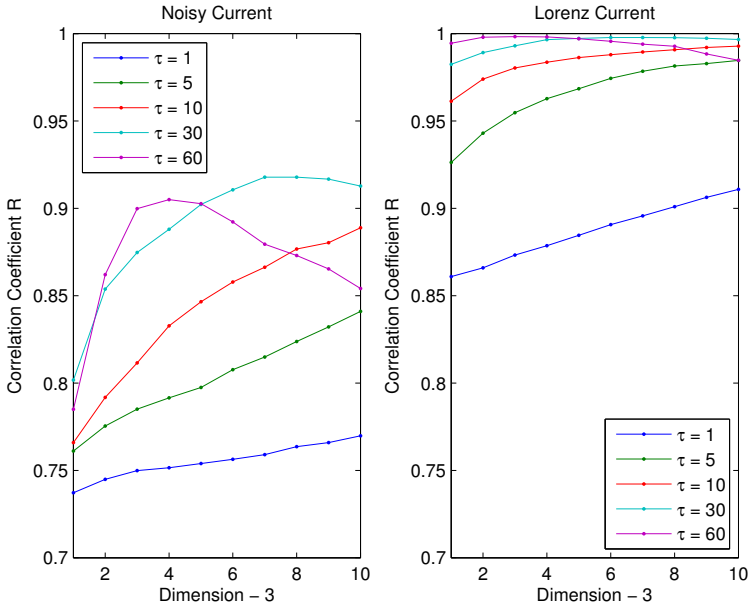

Figure 5.4: The correlation coefficient R comparing the estimated and actual current for a variety of choices of time delay τ and embedding dimension. The history was 3 seconds, and the estimation period was 1.2 seconds for both the noisy current (left) and the Lorenz current (right).

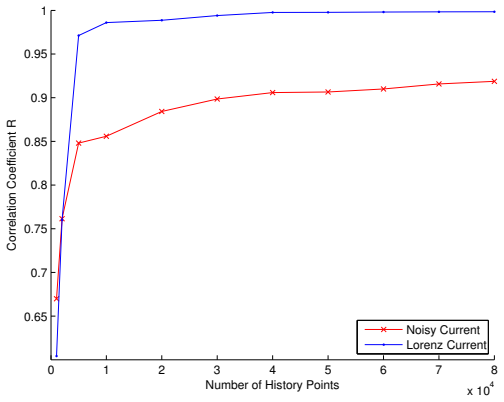

Figure 5.5: The correlation coefficient R which compares the estimated and actual currents as a function of the amount of history. For the noisy current $\tau = 1.2$ ms and embedding dimension was 10. For the Lorenz current $\tau = 2.4$ ms and the embedding dimension was 6.

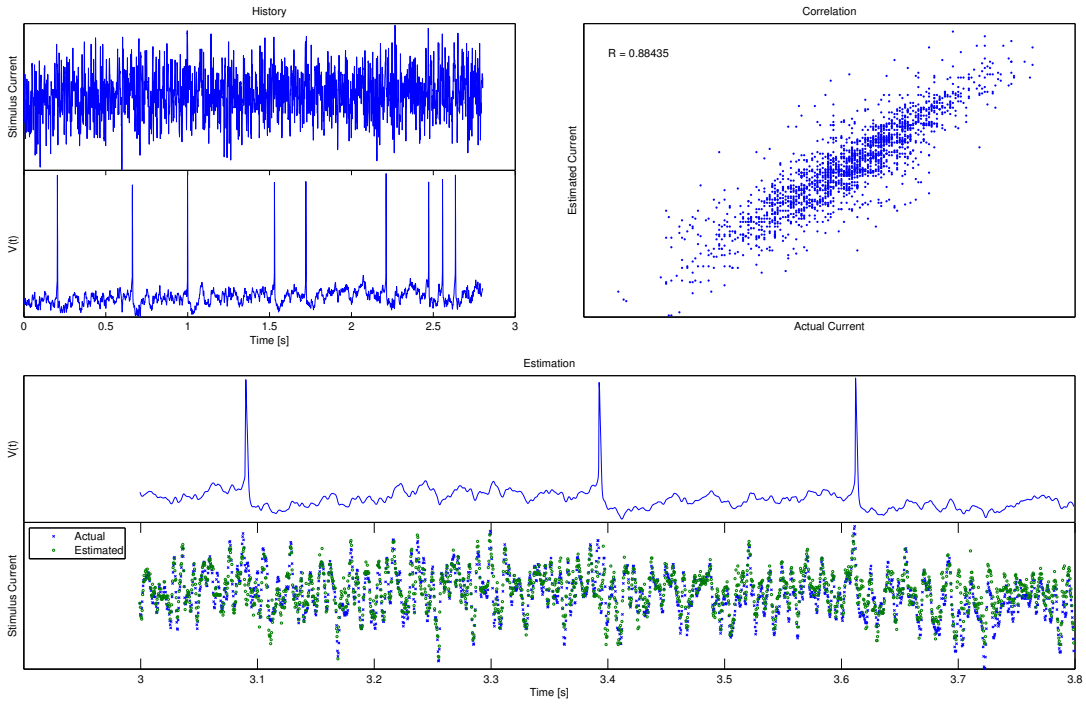

Figure 5.6: Mouse pyramidal neuron current from voltage estimation. A noisy current input was used to drive the dynamics of a mouse pyramidal neuron in slice. A calibration period of 70,000 points was used for current prediction of a new voltage sequence. A delay embedding of $\tau = 1$ms and embedding dimension of 5 historical voltage values $(-3\tau, -2\tau, -\tau, 0, \tau)$ were used to reconstruct current. Every 10th value was plotted for ease of display.

5.3.2 Spikes from calcium fluorescence

As with current from voltage, a common task in today's neuroscience laboratory is to reconstruct the spike train or spike rate from a calcium fluorescence signal. The same method as for voltage-to-current applies here, and we obtain similarly good results for this task, in both experimental and simulated conditions.

We estimate both spikes and average spike rates, depending on the frequency of calcium fluorescence capture. The only difference between the two is the convolved gaussian that is used to smooth the spikes and improve performance.

As actual data is acquired, this section will be updated. I expect to have similar performance between simulated and actual data, with only a slight degradation of performance when using experimental data.

5.4 Discussion

Our results demonstrate that a single non-parametric approach is sufficient for most time-series estimations of neuron biophysical data. No parameter estimation is required, no optimization, and no model estimation is required.

When compared to particle filter, path integral, or deconvolution methods, this approach is superior in that no additional assumptions or tests beyond the recorded data are necessary. Additionally, the method is self evaluating, in that we can estimate only using data, how well a novel output may be mapped to an input estimate.

This approach is best suited for connectivity estimation in neuronal networks, using calcium to estimate spiking activity, and voltage to estimate current. One approach is to make the assumption that the applied current is the sum of all post synaptic currents, which themselves are the results of weighted spike responses into the target neuron.

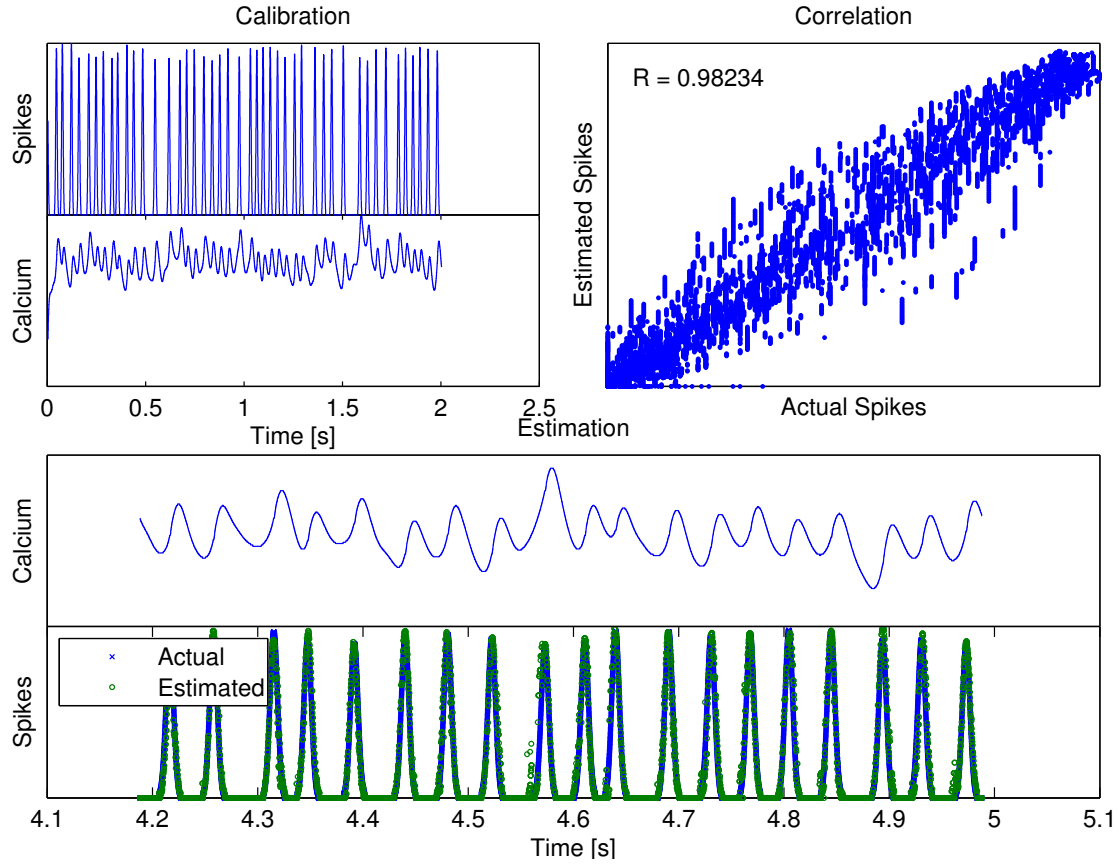

Figure 5.7: The spike times are estimated from simulated calcium florescence imaging data. The $V(t)$ output of the model was then turned into wide Gaussian peaks centered at the time of each spike. A calibration period of two seconds (50,000 time steps) of the calcium data and the spike data was used to build the tables (top left). Then an estimation of the spikes from the calcium signal was done over 0.8 seconds (20,000 time steps) and compared to the actual spikes (bottom). The correlation coefficient comparing the actual and estimated spikes was $R = 0.98$ (top right). In this example the time delay was $\tau = 4$ ms and the embedding dimension was 5. $\{-\tau, 0, \tau, 2\tau\, 3\tau\}$

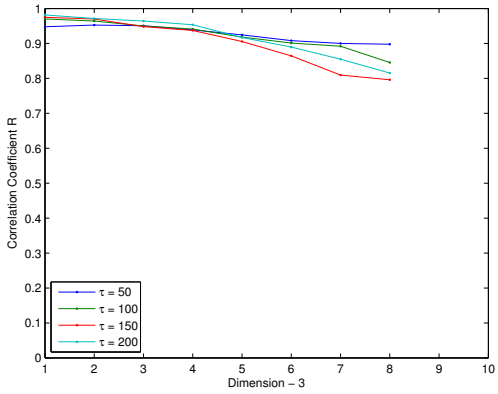

Figure 5.8: The correlation coefficient R comparing the estimated and actual spikes for a variety of choices of time delay τ and embedding dimension. A calibration period of 2 seconds and an estimation period of 0.8 seconds was used in all cases.

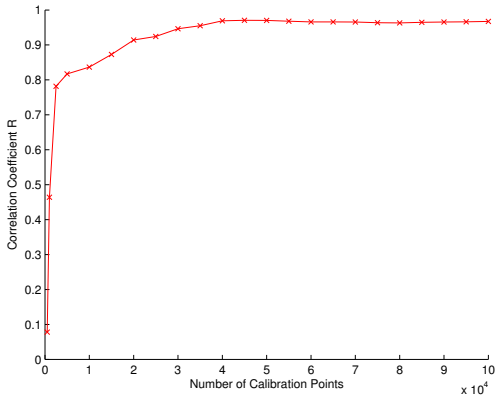

Figure 5.9: The correlation coefficient R which compares the estimated and actual spikes as a function of the amount of history. The embedding dimension was 4 and the time delay was $\tau = 4$ ms. $\{-\tau, 0, \tau, 2\tau\}$

5.5 Methods

5.5.1 Experimental Data Collection

Preparation

All experimental procedures were approved by the University of California at San Diego Institutional Animal Care and Use Committee. Sprague Dawley rats (9-23 d old) were anesthetized intraperitoneally with 0.1ml pentobartital sodium and perfused with 100 ml of cold (0-1°C) physiological solution containing the following (in mM): 125 NaCl, 2.5 KCl, 25 glucose, 25 $NaHCO_3$, 1.25 NaH_2PO_4, 2 $CaCl_2$, and 1 $MgCl_2$ (bubbled with 95% O_2 and 5% CO_2), after absence of eye blink reflex and the withdraw reflex in response to paw/tail pinch, the animal was decapitated and brain was taken out.

Cortical area containing the somatosensory cortex will be isolated and sliced cut in cold physiological solution using a vibratome (DTK-1000; Dosaka, Kyoto, Japan). Slices were cut at 300 μm thickness and incubated for about an hour in the physiology solution aerated with 95% O_2 and 5% CO_2 till the temperature of the solution reached room temperature (22-24°C).

Electrophysiological Recordings

The recording chamber was mounted on an upright microscope stage (Zeiss Axioskop 2 FS plus), maintained a constant temperature of 35C using a temperature control unit and continuously perfused with physiological solution. Whole cell patch recordings were performed under visual control using infrared differential interference contrast (IR-DIC)optics. Whole-cell voltage recordings from neurons were made using patch pipettes of 5-10 MOmega resistance pulled of thick borosilicate glass capillaries with filament (Hilgenberg GmbH, outer diameter, 1.5 mm; inner diameter, 1.0 mm, filament). The pipettes were filled with intracellular solution containing the following (in mM): 135 K-methylsulfonate, 10 HEPES, 10 Na-phosphocreatine, 4 $MgCl_2$, 4 Na-ATP, and 0.4 Na-GTP (adjusted to pH 7.25 with KOH). Neurobiotin (Sigma) was often added to the intracellular solution in a concentration of 1% for intracellular staining. The patch pipettes were pulled on a Narishige PP-830

puller and had a DC resistance of 1015 MΩ. The composition of the pipette (intracellular) solution was as follows (in mM): 135 K-methylsulfonate, 10 HEPES, 10 Na-phosphocreatine, 4 $MgCl_2$, 4 Na-ATP, and 0.4 Na-GTP (adjusted to pH 7.25 with KOH). The seal between the electrode tip and the cell membrane was higher than 1 GΩ. Recordings were made by Axoclamp 2B amplifiers (Axon Instruments) in current clamp mode. Electrical signals were recorded (Neurocorder DR- using the LabVIEW data acquisition and programming system (National Instruments).

Slices were placed in the recording chamber under an upright microscope (fitted with 2.5x, 0.075 numerical aperture and 40x, water immersion, 0.8 numerical aperture and a 3.3 mm working distance, Zeiss, Germany). The barrel field was visualized at low magnification under bright-field illumination and can be identified in layer 4 as narrow dark stripes with evenly spaced, light "hollows." Individual L4 and L5A neurons were identified at 40x magnification using infrared differential interference contrast (IR-DIC) microscopy. Somatic whole-cell recordings were performed at 34-36C. Signals were amplified using an Axoclamp amplifier (Axon Instruments, Union City, CA), filtered at 3 kHz, and sampled at 2-10 kHz using a program written in LABVIEW.

5.5.2 Simulated Data from Neuronal Dynamical Models

The standard HH model was used for to generate simulated data.

A non-linear calcium neuronal model, similar to the one presented in [FM91] is used to generate test values.

5.5.3 Nonparametric Estimation Algorithm

Given a series of measurements of a dynamical system like a neuron, consisting at least one observation variable $y(t)$, and one perturbing or control variable $u(t)$, recorded at many time points, the goal is to build a black box model that will be able to answer the following questions:

1. From new recordings of the observation variable $y(t)$, can we estimate the driving variable $u(t)$ responsible for the series of observations. This is the

filtering problem.

2. Given a new sequence of controls $u(t)$, along with a starting observation state $y(t_0)$, can we estimate the resultant observations.

3. From a series of novel observation from two or more entities $y_1(t), y_2(t), \ldots, y_m(t)$ of unknown inputs but of possibly trained black box models, can we estimate the similarity between each? In other words, do two measurements belong to the same model, even if driven by different inputs?

4. What kind of input function $u(t)$ is best suited to identify or classify members of a family, when performing coupled measurements of $u(t)$ and $y(t)$?

Problems 1 and 2 are shown in figure 5.1.

From a controls perspective, a system of the type as presented in the framework, where the forms of $\mathbf{H}()$ and its parameters are unknown, may be modeled using a nonlinear autoregressive model with an exogenous variable (NARX). That system can be approximated with the following relationship:

Since sampling is performed at constant and sufficiently small values of Δt, we can assume the regression function f to be reasonably smooth and continuous. The goal is to build a function f that uses some combination of l past values of y and m past values of u to predict the next value of y. That prediction, $y(t + \Delta t)$, is then placed as an argument in f,to predict the next point, $y(t + 2\Delta t)$, and so on. The value of $l + m$ is called the model order, and it's choice is critical in the construction of f. This is the prediction problem (#1 in the problem statement).

A nonlinear autoregressive model with exogenous input (NARX) is first and foremost a discrete time system, meaning that prediction and estimation take place in fixed time steps, mirroring most data collection situations. Depending on the configuration, a NARX model uses prior combined measurements of the observable state and the exogenous variable to estimate either the observable or the state variable from a set of past time observations. Denoting the state variable as $y(t)$ and the exogenous variable as $u(t)$, a NARX model can be used to predict a future state

from past $l+1$ states and $m+1$ inputs:

$$y(t+\Delta t) = f(y(t), y(t-\Delta t), \ldots, y(t-l\Delta t), u(t), u(t-\Delta t), \ldots, u(t-m\Delta t) \quad (5.1)$$

or can be used to estimate the exogenous variable $u(t)$ from a series of observable measurements around the time point of interest:

$$u(t) = g(y(t-k\Delta t), \ldots, y(t), \ldots, y(t+j\Delta t), u(t-\Delta t), \ldots, u(t-n\Delta t)) \quad (5.2)$$

In the above, we use k past values of the state $y(t)$, and $j+1$ values of the future state $y(t)$ and $n+1$ past values of the exogenous variable $u()$ for the estimation of the subsequent $u(t)$.

For both cases, we denote the vector $X(t)$ as the entire operand of the functions f and g. For the function g in equation 5.2,

$$X(t) = [y(t-k\Delta t) \quad \ldots \quad y(t) \quad \ldots \quad y(t+j\Delta t) \quad u(t-\Delta t) \quad \ldots \quad u(t-n\Delta t)] \quad (5.3)$$

and thus

$$u(t) = g(X(t))$$

Nearest Neighbor Interpolation

We may put these in the supplemental material.

Some prelimiaries on interpolation, linear or otherwise. The goal is to estimate a function mapping: $f : X \to Y$, using past measurement of X and Y . The simplest method is the nearest neighbor estimate

$$\hat{f}(X) = Y_j,\ j = \underset{i \in \{1,\ldots,N\}}{\arg\min}\, d(X, X_i)$$

where $\langle X_i, Y_i \rangle \in \{\langle X_1, Y_1 \rangle, \langle X_2, Y_2 \rangle, \ldots, \langle X_N, Y_N \rangle\}$ are the series of N measurements made previously. The choice of distance metric $d(a, b)$ may depend on the shape of X, with a Euclidean distance norm probably being sufficient in most cases.

nD nearest neighbor linear interpolation

A line equation can be written as $ax + b = y$. From known values of x and y, we want to estimate the coefficients a and b. In matrix form this becomes:

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & 1 \\ x_{n+1} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n+1} \end{bmatrix} \quad \text{or} \quad \mathbf{Xa} = \mathbf{y}$$

Where n, at a minimum, is the number of dimensions. For a 1D line, we need at least two points, so $n \geq 2$. Solving for the coefficients a and b, we simply invert the augmented X matrix using the pseudo inverse:

$$\mathbf{a} = pinv(\mathbf{X})\mathbf{y} \tag{5.4}$$

. Within this linear space, any value of x can be interpolated/extrapolated linearly using the coefficients in $\mathbf{a}$. Generalizing to larger dimensions and multi-dimensional mappings of the type: $\mathbf{y} = f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, with known $\mathbf{x}$ and $f(\mathbf{x})$ values, we seek to find the coefficients matrix of that linear space:

$$\begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^n & 1 \\ x_2^1 & x_2^2 & \cdots & x_2^n & 1 \\ \vdots & \vdots & \ddots & \vdots & 1 \\ x_{n+1}^1 & x_{n+1}^2 & \cdots & x_{n+1}^n & 1 \end{bmatrix} \begin{bmatrix} a_1^1 & a_2^1 & \cdots & a_m^1 \\ a_1^2 & a_2^2 & \cdots & a_m^2 \\ a_1^3 & & \ddots & \vdots \\ \vdots & & & \vdots \\ a_1^{n+1} & a_2^{n+1} & \cdots & a_m^{n+1} \end{bmatrix} = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^m \\ y_2^1 & y_2^2 & & y_2^m \\ \vdots & & \ddots & \vdots \\ y_{n+1}^1 & y_{n+1}^2 & \cdots & y_{n+1}^m \end{bmatrix} \tag{5.5}$$

Again, we invert $\mathbf{X}$ to solve for the coefficient matrix $\mathbf{a}$, as in equation 5.4. As long as we have $n+1$ points for n dimensions, we have a neccessary condition for invertibility. The rest is up to the condition number of $\mathbf{X}$.

nD polynomial interpolation

This is a linear interpolation, pretty standard. Polynomial interpolation can also be obtained, thougth the number of coefficients in **a** increased with the square(??) of the number of dimensions. Let's consider the 2D case $f(x, y)$:

$$\begin{bmatrix} x_1 & (x_1)^2 & y_1 & (y_1)^2 & x_1y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n(n+1)} & (x_{n(n+1)})^2 & y_{n(n+1)} & (y_{n(n+1)})^2 & x_{n(n+1)}y_{n(n+1)} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ \vdots \\ a_{n(n+1)+1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{n(n+1)+1} \end{bmatrix} \quad (5.6)$$

Now, the coefficients in **a** interpolate using the 2nd order polynomial:

$$\hat{f}(x, y) = a_1x + a_2x^2 + a_3y + a_4y^2 + a_5xy + a_6$$

Thus for 2D function, we need 6 coefficients. Let's verify for 3D:

$$\hat{f}(x, y, z) = a_1x + a_2x^2 + a_3y + a_4y^2 + a_5z + a_6z^2 + a_7xy + a_8xz + a_9yz + a_{10}$$

So it is possible (need to check), in general that the formula for the number of coefficients $C_{n,k}$ in **a** for n dimensions of polynomial of order k goes like:

$$C_{n,k} = \frac{1}{k!} \prod_{k=1}^{K} (n + k)$$

It is obvious that the larger dimensions n or polynomial order k, the larger the

number of coefficients required, so the larger the number of points required to estimate parameters. For now, I think a 2nd order polynomial suffices. Either way, once the **a** vector/matrix is found, then interpolating $\hat{f}(\mathbf{x})$ is relatively straightforward:

$$\hat{f}(\mathbf{x}^*) = \mathbf{x}^* \cdot \mathbf{a} \quad \text{or whatever basis of } \mathbf{x}$$

Parallel Implementation

Along with Jack Quinn, we've implemented the lookup and interpolation in a fully parallel fashion to run on GPU cards. In this way, nearest-neighbor lookups are performed in parallel, so the full estimation of an unknown signal runs in near real time. This technique is ideal for GPUs, as it can be parallelized quite easily. It is thus superior to other methods like deconvolution or particle filtering since it does not require any prior assumptions of the state-space model of the system.

Chapter 5, in part is currently being prepared for submission for publication of the material. Buibas M*, Cui J*, Quinn J*, Abarbanel H, Devor A, and Silva, GA. Buibas, Cui, and Quinn contributed equally to this paper, though the original idea is the thesis author's.

Chapter 6

Math and Consequences

6.1 Parameter Space Shape

This is a parameter space sweep of the cost function of the weight space of two neurons coupled into a third. There is extreme dependence on connectivity parameters in a coupled network of HH neurons. In figure 6.1, we plot the log of cost function (sum error integrals over the data range) as a function of the two functional non-zero weights into neuron one of a 30-neuron network. We see a very narrow, steep, and somewhat jagged valley, where the known global minimum lies at (0.2907,0.3908). The global minimum represents the known parameters, and a cost of 0. As the parameters get close and closer to 0, the slope of the valley walls increases, so small changes in parameter guesses have larger impacts on the cost function. A step towards a lower cost value that is based on the derivative would have to be very small so as not to result in an increase in the cost function, or a climb up the wall.

This shape of the parameter space makes it very difficult for any optimizer to find the global minimum, especially in higher dimensions. It is reminiscent of the Rosenbrock banana function, but with additional jaggedness. The jaggedness of the parameter space is likely caused by the rounding errors of the float variables used to represent states of neurons. I would expect that with higher precision arithmetic, the jaggedness goes away.

In higher dimensions, the problem is further compounded. Vast swaths of

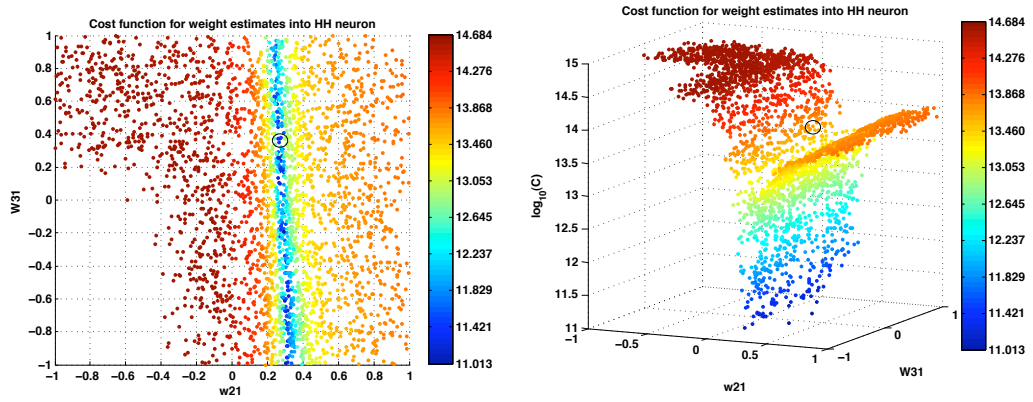

Figure 6.1: Parameter space of input weights around one HH neuron. The known functional weight set is circled black, and the log_{10} of the cost function is color coded as the Z-axis.

the parameter space is uninformative, meaning that there is no direction to take that lowers the cost function. This makes this space harder to search. Once a lower cost function is reached, the space is likely valley-shaped, and possibly curved, likely growing steeper and steeper as we get closer to the global minimum. Steep and curved spaces are difficult to traverse, since the scale of the step has to be very small so as not to climb out of the valley. I've illustrated this with the 2-D banana function. As is shown in figure 6.2, a parabolic-shaped valley is where the global minimum resides. While finding the valley might be relatively easy, traversing it is difficult as most straight directions within the valley will yield a higher cost function. For cases like these, what is ideally needed is a generalized description of the shape of the space at a macro (non-local, non-Euclidean) scale, so the next step taken in the optimization problem is both meaningful (not infinitesimally small), and results in finding a lower cost point.

6.2 Curse of Dimensionality

I've shown within the mapping paper that high-dimensional weight or parameter spaces are difficult, if not impossible to work with. Not only is the solution difficult to search for and find, but the space or set of possible solutions is increased, as more combinations of parameters may yield an equivalent cost function or re-

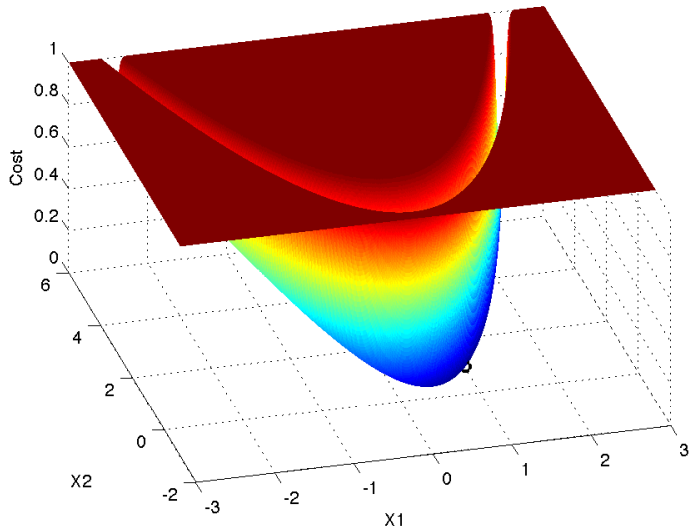

Figure 6.2: The Rosenbrock banana function is likely similar to the parameter space of functional connections into a neuron. For most of the space, the derivative is zero, meaning it is uninformative. Lower costs are likely to be found in valleys which are very difficult to traverse, using standard derivative methods. The global minimum is at $(1, 1)$, indicated by the partially obstructed black circle.

sult. This effect is seen in normal probability spaces, where it is often said that any hypothesis is valid with enough parameters. Intuitively this is valid, but more importantly two factors come into play in high-dimensional spaces, as in functional weight estimation, that make the problem not only intractable but make the answer relatively meaningless. Together they've been coined as the curse of dimensionality REF.

The first effect of high dimensional spaces is that everything is almost equally far away. In two dimensions we may consider two points close or far based on some distance metric like the Euclidean distance. Some points are close and can be considered similar, while others that are far are dissimilar. In high dimensional spaces, however, almost all points are equally far apart, so the notion of similarity based on geometric closeness is no longer valid. High dimensional space is mostly empty and everything is almost equally far apart.

The second effect is on probability distribution functions. In one dimension a Gaussian distribution is 99.7% contained within 3 standard deviations of its mean. However, as the number of independent dimensions is increased, the portion of the

distribution contained within a distance of 3 standard deviations from the mean tends to zero. As shown in figure 6.3, with more than about 20 dimensions, almost none of the distribution is within 3 standard deviations of the mean. This means that the concept of locality, or saying that the mean is representative of the population has no meaning when the mean describes many independent dimensions.

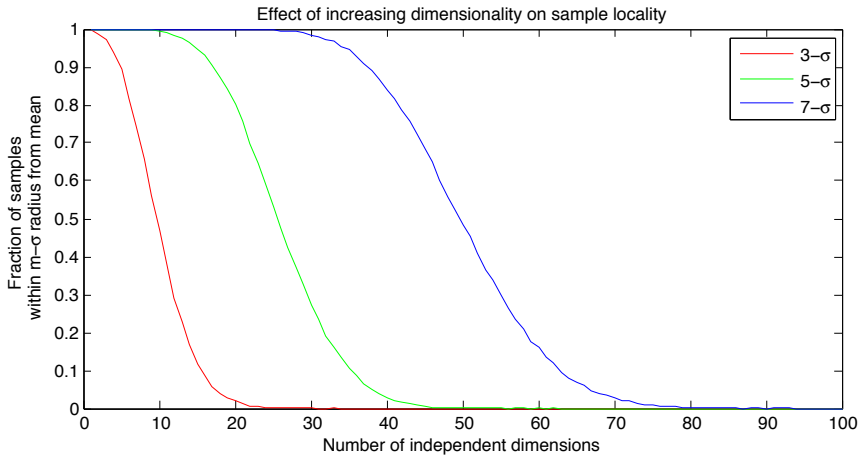

Figure 6.3: The effect of increasing dimensionality on the proportion of the distribution within a given distance from the mean. With increasing dimensionality less of the distribution lies close to the mean. Even with a distance of 7 standard deviations from the mean, with enough dimensions all data is far away from the mean.

6.3 Concentration of Measure

Another, perhaps more important consequence of mathematics in high dimensional spaces is the concentration of measure. Stated informally: continuous functions that depend on too many parameters tend to be constant. In a formal definition, we let (X_n, d, μ) denote a metric space of n dimensions, distance function $d(\cdot, \cdot)$, and probability measure μ. Letting $\kappa > 0$ be a small threshold measure, the *observable diameter* D is the smallest real number with the property that two elements $x, y \in X_n$, drawn with respect to probability measure μ, and for any given continuous function $f : X_n \to \mathbb{R}$ to denote some feature of members X_n, the proba-

bility that the values of f at x and y differ by D is below threshold κ:

$$P\big(|f(x) - f(y)| \leq D\big) < \kappa$$

Denoting C as characteristic size or the median value of distances between two elements in X_n, the *concentration of measure* phenomenon is the effect that as the dimensionality n is increased, the observable diameter D decreases dramatically with respect to C:

$$D \ll C \quad \text{as } n \to \infty.$$

This definition, as presented by Pestov in [Pes08], is relatively broad and is a profound effect of high dimensional spaces on functions. Functions are simply rules that transform one thing into another. *Concentration of measure* means that functions that depend on too many independent things tend to be constant, and have little resolving power, meaning that they cannot distinguish meaningfully between inputs, and all tend to average out to a constant value.

6.4 Consequences

6.4.1 Stable Network Models

Given the highly nonlinear shape of the parameter space and the extreme sensitivity of data similarity relative to functional connective weights, I have to raise questions on the validity of the models and approaches to network modeling. It is likely that networks of neurons with transmission delays exhibit chaotic behavior under many circumstances. If this is the case then the principle of extreme dependence on initial conditions holds, so any input-output mapping of the network are not likely to be repeatable or stable. I have reservations that simply connecting multiple neuronal models in a network is all that is required to simulate a network of neurons. There are likely other emergent phenomena that have to be modeled to stabilize the network so that they have repeatable input-output patterns and can accommodate significant variance in input patterns. As I've seen with the networks of the test suite, small perturbations in the delay or weight parameters can have

significant effects on the resultant dynamics, given our models. This is highly unlikely in a real-world neuronal network that can function reasonably well even when perturbed, so it is very likely that modeling a network will not be just modeling a collection of collected neurons.

6.4.2 Low Dimensional Representations

A statistician once mentioned that in high dimensional spaces any hypothesis is valid, and virtually any conclusion can be drawn. Because there is no concept of near or far in high dimensions, the idea of partitioning a high-dimensional space into clusters of related points, as is required for classification, does not work: everything is equally far apart from everything else. More importantly for a class of objects, if represented with many independent attributes, the average has no relation to and is not representative of the class. Given this, I suspect that our internal representation of objects, at a network level has to be low-dimensional. We have no problems recognizing items, words, and other people, and we can describe with relative ease, and with few words, typical objects of a given class. While everyone's description of a particular object may be different, internally we rely on a few features to classify and recognize stimuli, filtering out all other superfluous inputs. Using few features in classification and recognition ensures stability and robustness to variability, and allows us to have a relevant mean or typical member of a class. I would venture to guess that the embedding dimension is somewhere around 7, though the shape of the feature space is likely to be far from linear, complicating our analysis of it, which is typically based on Euclidean and orthogonal spaces.

6.4.3 Few Incoming Functional Connections

From the phenomenon of concentration of measure, I propose that cells in a network receive relatively few functional inputs at any given time, for any given task. While physically cells may be connected to tens of thousands of other cells, and even if 1% of synapses were potentiated, one hundred active inputs to any given cell would not likely present the soma with any distinguishable signal. Thus, I propose that at any given time, a cell must receive relatively few independent inputs for

any meaningful information processing to occur in the cell body. Receiving inputs from many independent cells at the time would effectively eliminate the information content departing the target cell body. This principle, grounded in the mathematics of large dimensional measure spaces, if true, would have a profound effect in the many theoretical and practical strategies aimed establishing functional connectivity. As a consequence, it would also be consistent with the optimal utilization of neural tissue, where multiple behaviors can be executed in the same tissue, with different information flow and thus functional connective patterns. Again, I speculate that any activity will likely recruit no more than about 7 active incoming connections to any given neuron. This constraint, if true, would greatly help improve the algorithmic efficiency of any mapping algorithm, but also provide novel directions and strategies for observation and experimentation. Knowing that only 7 or so independent inputs can be active at any given time, can significantly reduce the search space for the mapping algorithm. From an experimental standpoint, more effective observation strategies can be devised to maximize information content when whole networks are observed.

These hypotheses, resulting from the fundamental mathematics of high dimensional spaces, can form the basis for novel research ideas, spanning neuronal simulation, machine learning, optimization and representation, and biological networks. Within my research I've treated these disciplines in a connected manner, and I believe that combined use within these mathematical constraints will yield fruitful results and advance our understandings of biological neuronal systems.

Appendix A

Weaver Suite for Simulation and Mapping

The complete suite of simulation and estimation, both parametric and non-parametric has been implemented on Graphics Processor Unit cards. GPUs are becoming extremely powerful and affordable
($1/GFLOP GPU vs $7/GFLOP CPU). Designed for simulation and parallel processing. Second generation of software tools just available (OpenCL), though low-level programming of individual architecture is needed to achieve optimal performance.

I've attached the documentation for the first version of the weaver software suite, which is a collection of source code files used in the framework and mapping papers. The code is actively undergoing improvements, and is used by several labs doing simulation and functional connectivity estimation. The code is available at www.silva.ucsd.edu.

A.1 Requirements

The following requirements represent what was tested and found to work in the lab. Since this software is very much alpha-phase, little testing has been done outside our Mac laptops and Linux workstation.

Computer Hardware: An intel Core2Duo or Xeon 55xx multi-core processor is

highly recommended. At least 4GB of ram are needed to store the larger networks or longer simulation times.

Operating System: Mac OS X 10.5 or higher, Linux - Tested on Ubuntu 9.0 and Higher.

GPU Hardware Any nVIDIA video card with compute capability 1.1 or higher. (GeForce 8600M and Tesla C1060 have been tested).

CUDA Software/Compilers: CUDA drivers 3.0 and higher. The nvidia and gcc compilers are needed to make the executable that is called from MATLAB. See www.nvidia.com for the drivers and development kit.

MATLAB: Version 2008b. MATLAB has changed the compile flags and extensions of compiled mex files for 2009 and later versions. The current Makefile only seems to work for 2008b, but can be changed to accommodate newer version of MATLAB.

A.2 Installation

Upon unpacking the distribution, a single directory will be created, containing all required files. The fastest way to start is by opening and running the `benchmarks.m` script from MATLAB, from that directory. This will run the benchmarks on the local GPU, using both IZ and HH neuronal models.

To generate geometric networks, run the `built-test-nets.m` script from the same directory. It will place vertices and generate 80 networks of different connectivity classes.

The next sections complement the NECO article, describing the generation and simulation programs. The mapping program will be added upon publication of that manuscript.

A.3 Network Generation

The first step in generating a network is the placement of the vertices. The `positiongraph.m` function accomplishes this for two or three dimensions, and any number of vertices. Note that large, dense networks may be challenging to this algorithm, taking a long time to complete.

Of the four types of networks described in the NECO article, 3 can be generated with the `biosmallworldnet.m` function, just by varying the randomness parameter. A randomness parameter of 0 will generate a lattice network, values around 0.10 produce small-world networks with 10% random connections, and a random network is generated by setting the randomness to 1.

The scale-free network class is generated using the `biosf.m` function, generates a geometric scale-free network, also called an Apollonian network, which has both scale-free and small-world properties.

Both network generation scripts require pre-positioned vertices as input. From MATLAB, typing `>>help biosf` will show the help for that function.

A.4 Network Simulation

Entry to the network simulation routines is through a MATLAB shared library `cunetsim`, that is compiled from source using the provided Makefile. Parameters, initial conditions, network connectivity, delays, and external controls are all initialized in MATLAB and are passed to `cunetsim` as pointers to mex arrays, where they are copied to the GPU, and the individual vertex threads are started to perform the simulation. Upon completion of the simulation, the observable results are copied back to the CPU memory as a pointer that the MATLAB environment can access, plot and save as needed. We will briefly highlight some relevant functions, files, and features of the network simulation

A.4.1 cunetsim

Called from within MATLAB, with the following matlab arrays as inputs:

psE partial sparse connectivity matrix, specifies the incoming connections to every vertex, stored as uint32, generated from `nettosparse.m` MATLAB file.

psD partial sparse delay matrix, stored as uint32, generated from `nettosparse.m`

psW partial sparse weight matrix, stored as VARTYPE, generated from `nettosparse.m`. `psE, psD, psW` all specify incoming connections, delays, and weights to every vertex in the network. They are generated within MATLAB from the physical, delay, and functional connectivity matrices.

U external input control, a $N \times T$ matrix for a network of N vertices and T time points. Specifies injected current at each time point for each cell.

ptrans transition parameters. They are the parameters for the dynamic model, with the first parameter in the array being the time step used for integration.

pobs observation parameters, organized as follows for a linear model with noise:

state which state variable to observed

scale how much to scale

offset how much to offset

noise sigma how much noise ($z = y \cdot scale + offset + \mathcal{N}(0, \sigma^2)$)

y0 initial conditions, same for each vertex

`cunetsim` returns the following arrays to MATLAB:

Z the observable variable for each vertex at each time point ($N \times T$ matrix)

X output raster (spike raster for neurons) the output of each cell (used for transmission)

See comments in `benchmarks.m` and `cunetsim.cu` for further details.

A.4.2 izcamodel.cuh hhcamodel.cuh

The Izhikevitch and Hodgkin-Huxley models. These files contain the thread kernel functions, transitions, and transmitters for the network simulation. The model is selected at compile time, with the appropriate compile flag (see Makefile).

A.4.3 integrators.cuh

Header file with all integrator functions. We currently support one-step Euler, Runge-Kutta 2nd and Runge-Kutta 4th order methods. The integrator functions are designed to operate on a model, chosen at compile time. Multi-step methods will be added later.

A.4.4 observers.cuh

Observer functions that provide observation functions of state variables. Currently only linear observation models have been coded.

Bibliography

[AB05] Nicola J Allen and Ben A Barres. Signaling between glia and neurons: focus on synaptic plasticity. *Current Opinion in Neurobiology*, 15(5):542–8, Oct 2005.

[ACFK09] Henry D. I Abarbanel, Daniel R Creveling, Reza Farsian, and Mark Kostuk. Dynamical state and parameter estimation. *Siam J Appl Dyn Syst*, 8(4):1341–1381, Jan 2009.

[AHAS05] José S Andrade, Hans J Herrmann, Roberto F. S Andrade, and Luciano R Da Silva. Apollonian networks: Simultaneously scale-free, small world, euclidean, space filling, and with matching graphs. *Physical Review Letters*, 94(1):4, Jan 2005.

[APK+07] C Agulhon, J.-C Platel, B Kolomiets, V Forster, S Picaud, J Brocard, P Faure, and P Brulet. Bioluminescent imaging of ca2+ activity reveals spatiotemporal dynamics in glial networks of dark-adapted mouse retina. *The Journal Of Physiology*, 583(3):945–958, Jul 2007.

[APM+08] Cendra Agulhon, Jeremy Petravicz, Allison B McMullen, Elizabeth J Sweger, Suzanne K Minton, Sarah R Taves, Kristen B Casper, Todd A Fiacco, and Ken D McCarthy. What is the role of astrocyte calcium in neurophysiology? *Neuron*, 59(6):932–46, Sep 2008.

[BA99] AL Barabasi and R Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, Jan 1999.

[BBM99] A Bondarenko, D Bortz, and J More. Cops: Large-scale nonlinearly constrained optimization problems. *Mathematics and Computer Science Division*, Jan 1999.

[BCH10] Tiago Branco, Beverley A Clark, and Michael Häusser. Dendritic discrimination of temporal input sequences in cortical neurons. *Science*, Aug 2010.

[Ben05] M Bennett. A quantitative model of purinergic junctional transmission of calcium waves in astrocyte networks. *Biophysical Journal*, 89(4):2235–2250, Oct 2005.

[BFB94] JL BARRON, DJ FLEET, and S Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, Jan 1994.

[BGMH07] J Benda, T Gollisch, C Machens, and A Herz. From response to stimulus: adaptive sampling in sensory physiology. *Current Opinion in Neurobiology*, Jan 2007.

[BHP08] R K P Benninger, M Hao, and D W Piston. Multi-photon excitation imaging of dynamic processes in living cells and tissues. *Rev Physiol Biochem Pharmacol*, 160:71–92, Jan 2008.

[BM04] S Baker and I Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, Jan 2004.

[BP01] GQ Bi and MM Poo. Synaptic modification by correlated activity: Hebb's postulate revisited, Jan 2001.

[BS98] JD Bomberger and DE Seborg. Determination of model order for narx models directly from input-output data. *Journal of Process Control*, 8(5-6):459–468, Jan 1998.

[BS09] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, Mar 2009.

[BS11] Marius Buibas and Gabriel A Silva. A framework for simulating and estimating the state and functional topology of complex dynamic geometric networks. *Neural Computation*, 23(1):183–214, Jan 2011.

[BSL+96] P Baraldi, A Sarti, C Lamberti, A Prandini, and F Sgallari. Evaluation of differential optical flow techniques on synthesized echo images. *Ieee T Bio-Med Eng*, 43(3):259–272, Jan 1996.

[Bur06] A Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics*, Jan 2006.

[BYNS10] Marius Buibas, Diana Yu, Krystal Nizar, and Gabriel A Silva. Mapping the spatiotemporal dynamics of calcium signaling in cellular neural networks using optical flow. *Annals of biomedical engineering*, Mar 2010.

[Cal04] E Callaway. Feedforward, feedback and inhibitory connections in primate visual cortex. *Neural Networks*, Jan 2004.

[CB99] E J Chichilnisky and D A Baylor. Receptive-field microstructure of blue-yellow ganglion cells in primate retina. *Nature Neuroscience*, 2(10):889–93, Oct 1999.

[CBFCS90] A H Cornell-Bell, S M Finkbeiner, M S Cooper, and S J Smith. Glutamate induces calcium waves in cultured astrocytes: long-range glial signaling. *Science*, 247(4941):470–3, Jan 1990.

[CCP+03] Silvia Coco, Federico Calegari, Elena Pravettoni, Davide Pozzi, Elena Taverna, Patrizia Rosa, Michela Matteoli, and Claudia Verderio. Storage and release of atp from astrocytes in culture. *J Biol Chem*, 278(2):1354–62, Jan 2003.

[CDZ06] M Canepari, M Djurisic, and D Zecevic. Dendritic signals from rat hippocampal ca1 pyramidal neurons during coincident pre- and postsynaptic activity: a combined voltage- and calcium-imaging study. *The Journal Of Physiology*, 580(2):463–484, Dec 2006.

[CIY05] R Cossart, Y Ikegaya, and R Yuste. Calcium imaging of cortical networks dynamics. *Cell Calcium*, 37(5):451–457, May 2005.

[CJA08] Daniel R Creveling, James M Jeanne, and Henry D. I Abarbane. Parameter estimation using balanced synchronization. *Physics Letters A*, 372(12):2043–2047, Jan 2008.

[CYM+09] Siu-Kei Chow, Diana Yu, Christopher L Macdonald, Marius Buibas, and Gabriel A Silva. Amyloid-beta directly induces spontaneous calcium transients, delayed intercellular calcium waves, and gliosis in rat cortical astrocytes. *ASN neuro*, Dec 2009.

[DA09] Peter Dayan and L F. Abbott. Theoretical neuroscience: computational and mathematical modeling of neural .... page 460, Jan 2009.

[DMS94] A DESTEXHE, Z F Mainen, and Terry Sejnowski. An efficient method for computing synaptic conductances based on a kinetic-model of receptor-binding, Jan 1994.

[DMS98] A DESTEXHE, Z Mainen, and T Sejnowski. Kinetic models of synaptic transmission. *Methods in neuronal modeling*, Jan 1998.

[EDS03] M Eichler, R Dahlhaus, and J Sandkuhler. Partial correlation analysis for the identification of synaptic connections. *Biological Cybernetics*, 89(4):289–302, Jan 2003.

[ES03] J Ebling and G Scheuermann. Clifford convolution and pattern matching on vector fields. *Visualization, 2003. VIS 2003. IEEE*, pages 193 – 200, Oct 2003.

[EZJO10] Seif Eldawlatly, Yang Zhou, Rong Jin, and Karim G Oweiss. On the use of dynamic bayesian networks in reconstructing functional neuronal networks from spike train ensembles. *Neural Computation*, 22(1):158–89, Jan 2010.

[Feo08] Oscar De Feo. Estimating interdependences in networks of weakly coupled deterministic systems. *Phys. Rev. E*, 77(2):15, Feb 2008.

[FM91] Douglas D Fraser and Brian A MacVicar. Low-threshold transient calcium current in rat hippocampal lacunosum-moleculare interneurons: Kinetics and modulation by neurotransmitters. *Journal of Neuroscience*, 11(9):2812–2820, Jan 1991.

[FOM+07] S Feldt, H Osterhage, F Mormann, K Lehnertz, and M Zochowski. Internetwork and intranetwork communications during bursting dynamics: applications to seizure prediction. *Phys Rev E Stat Nonlin Soft Matter Phys*, 76(2 Pt 1):021920, Aug 2007.

[FSG02] R Douglas Fields and Beth Stevens-Graham. New insights into neuron-glia communication. *Science*, 298(5593):556–62, Oct 2002.

[GGL+04] Yu Guo, Biao Gong, Sebrina Levesque, Thomas Manfredi, and Ying Sun. Automated detection and delineation of mitochondria in electron micrographs of human skeletal muscles. *Microsc Res Tech*, 63(3):133–9, Feb 2004.

[GH07] W Gobel and F Helmchen. In vivo calcium imaging of neural network function. *Physiology*, 22(6):358–365, Dec 2007.

[GKH07] Werner Göbel, Björn M Kampa, and Fritjof Helmchen. Imaging cellular network dynamics in three dimensions using fast 3d laser scanning. *Nat Meth*, 4(1):73–79, Jan 2007.

[GMG+06] Olga Garaschuk, Ruxandra-Iulia Milos, Christine Grienberger, Nima Marandi, Helmuth Adelsberger, and Arthur Konnerth. Optical monitoring of brain function in vivo: from neurons to networks. *Pflugers Arch - Eur J Physiol*, 453(3):385–396, Nov 2006.

[GOU95] N GOULD. Cute: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, Jan 1995.

[GYB+10] Lisa A Gunaydin, Ofer Yizhar, André Berndt, Vikaas S Sohal, Karl Deisseroth, and Peter Hegemann. Ultrafast optogenetic control. *Nature Neuroscience*, 13(3):387–92, Mar 2010.

[HBJ+09] Ryota Homma, Bradley J Baker, Lei Jin, Olga Garaschuk, Arthur Konnerth, Lawrence B Cohen, and Dejan Zecevic. Wide-field and two-photon imaging of brain activity with voltage- and calcium-sensitive dyes. *Philos Trans R Soc Lond, B, Biol Sci*, 364(1529):2453–67, Sep 2009.

[HBS08] Mahboubeh Hashemi, Marius Buibas, and Gabriel A Silva. Automated detection of intercellular signaling in astrocyte networks using the converging squares algorithm. *Journal of Neuroscience Methods*, 170(2):294–299, Jan 2008.

[HIS96] F Helmchen, K Imoto, and B Sakmann. Ca2+ buffering and action potential-evoked ca2+ signaling in dendrites of pyramidal neurons. *Biophysical Journal*, 70(2):1069–81, Feb 1996.

[HKG+09] Tycho M Hoogland, Bernd Kuhn, Werner Goebel, Wenying Huang, Junichi Nakai, Fritjof Helmchen, Jane Flint, and Samuel S. H Wang. Radially expanding transglial calcium waves in the intact cerebellum. *P Natl Acad Sci Usa*, 106(9):3496–3501, Jan 2009.

[HS80] W Hock and K Schittkowski. Test examples for nonlinear programming codes. *Journal of Optimization Theory and Applications*, 30(1):127—129, Jan 1980.

[HS81] BKP Horn and BG Schunck. Determining optical flow(distribution of apparent movement velocities of image brightness .... *Techniques and applications of image understanding*, 21-23:319–331, Jan 1981.

[Ioa07] Andreas A Ioannides. Dynamic functional connectivity. *Current Opinion in Neurobiology*, 17(2):161–70, Apr 2007.

[Izh05] Eugene M. Izhikevich. Dynamical systems in neuroscience. 2005.

[Jah03] Jahne. Digital image processing. page 724, Jan 2003.

[KLHB09] Kishore V Kuchibhotla, Carli R Lattarulo, Bradley T Hyman, and Brian J Bacskai. Synchronous hyperactivity and intercellular calcium waves in astrocytes in alzheimer mice. *Science*, 323(5918):1211–1215, Jan 2009.

[KNMN09] Zeb L Kurth-Nelson, Anusha Mishra, and Eric A Newman. Spontaneous glial calcium waves in the retina develop over early adulthood. *Journal of Neuroscience*, 29(36):11339–11346, Jan 2009.

[KPW+02] M-G Knuttinen, T B Parrish, C Weiss, K S LaBar, D R Gitelman, J M Power, M-M Mesulam, and J F Disterhoft. Electromyography as a recording system for eyeblink conditioning with functional magnetic resonance imaging. *Neuroimage*, 17(2):977–87, Oct 2002.

[KTF+06] Nicolaos B Karayiannis, Guozhi Tao, James D Frost, Merrill S Wise, Richard A Hrachovy, and Eli M Mizrahi. Automated detection of videotaped neonatal seizures based on motion segmentation methods. *Clin Neurophysiol*, 117(7):1585–94, Jul 2006.

[LA07] Keith Langley and Stephen J Anderson. Subtractive and divisive adaptation in visual motion computations. *Vision Res*, 47(5):673–86, Mar 2007.

[LAG05] S Lim, J Apostolopoulos, and A Gamal. Optical flow estimation using temporally oversampled video. *Image Processing, IEEE Transactions on*, 14(8):1074 – 1087, Aug 2005.

[LBP09] Jeremy Lewi, Robert Butera, and Liam Paninski. Sequential optimal design of neurophysiology experiments. *Neural Computation*, 21(3):619–687, Jan 2009.

[LH08] Maxim Lavrentovich and Sheryl Hemkin. A mathematical model of spontaneous calcium(ii) oscillations in astrocytes. *J Theor Biol*, 251(4):553–60, Apr 2008.

[Miu05] Kota Miura. Tracking movement in cell biology. *Adv Biochem Eng Biotechnol*, 95:267–95, Jan 2005.

[MNS09] Eran A Mukamel, Axel Nimmerjahn, and Mark J Schnitzer. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron*, 63(6):747–60, Sep 2009.

[MPdF05] VA Makarov, F Panetsos, and O de Feo. A method for determining neural connectivity and inferring the underlying network dynamics using extracellular spike recordings. *Journal of Neuroscience Methods*, 144(2):265–279, Jan 2005.

[MT06] RM Memmesheimer and M Timme. Designing complex networks. *Physica D: Nonlinear Phenomena*, 224(1-2):182–201, 2006.

[MYBS08] Christopher L Macdonald, Diana Yu, Marius Buibas, and Gabriel A Silva. Diffusion modeling of atp signaling suggests a partially regenerative mechanism underlies astrocyte intercellular calcium waves. *Frontiers in neuroengineering*, 1:1, Jan 2008.

[NJL08] Suhita Nadkarni, Peter Jung, and Herbert Levine. Astrocytes optimize the synaptic transmission of information. *PLoS Comput Biol*, 4(5):e1000088, May 2008.

[NPY07] Volodymyr Nikolenko, Kira E Poskanzer, and Rafael Yuste. Two-photon photostimulation and imaging of neural circuits. *Nat Meth*, 4(11):943–950, Jan 2007.

[NZ97] E A Newman and K R Zahs. Calcium waves in retinal glial cells. *Science*, 275(5301):844–7, Feb 1997.

[OGK07] Nils Ostlund, Björn Gerdle, and J Stefan Karlsson. Location of innervation zone determined with multichannel surface electromyography using an optical flow technique. *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology*, 17(5):549–55, Oct 2007.

[OWB05] M Okatan, M Wilson, and E Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking .... *Neural Computation*, Jan 2005.

[PB98] C Pagano and G Bingham. Comparing measures of monocular distance perception: Verbal and reaching errors are not correlated. *Journal of Experimental Psychology-Human Perception and ...*, Jan 1998.

[Pes08] Vladimir Pestov. An axiomatic approach to intrinsic dimension of a dataset, Jan 2008.

[PEW+08] R Madelaine Paredes, Julie C Etzler, Lora Talley Watts, Wei Zheng, and James D Lechleiter. Chemical calcium indicators. *Methods*, 46(3):143–51, Nov 2008.

[Pil05] J. W Pillow. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25(47):11003–11013, Nov 2005.

[PVLBJ09] Maurizio De Pittà, Vladislav Volman, Herbert Levine, and Eshel Ben-Jacob. Multimodal encoding in a simplified model of intracellular calcium signaling. *Cogn Process*, 10 Suppl 1:S55–70, Feb 2009.

[QP09] Rodrigo Quian Quiroga and Stefano Panzeri. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience*, 10(3):173–185, Mar 2009.

[SA07] Jessica R Sudbury and Massimo Avoli. Epileptiform synchronization in the rat insular and perirhinal cortices in vitro. *Eur J Neurosci*, 26(12):3571–82, Dec 2007.

[SFMT98] G A Silva, C Feeney, L R Mills, and E Theriault. A novel and rapid method for culturing pure rat spinal cord astrocytes on untreated glass. *Journal of Neuroscience Methods*, 80(1):75–9, Mar 1998.

[SG06] Eliana Scemes and Christian Giaume. Astrocyte calcium waves: what they are and what they do. *Glia*, 54(7):716–25, Nov 2006.

[SGHK03] C Stosiek, O Garaschuk, K Holthoff, and A Konnerth. In vivo two-photon calcium imaging of neuronal networks. *Proceedings of the National Academy of Sciences*, Jan 2003.

[SGL+05] G Silberberg, S Grillner, FEN LeBeau, R Maex, and H Markram. Synaptic pathways in neural microcircuits, Jan 2005.

[SH03] L Sragner and G Horváth. Improved model order estimation for nonlinear dynamic systems. *... of the Second IEEE International Workshop ...*, Jan 2003.

[SM06] Michail Stamatakis and Nikos V Mantzaris. Modeling of atp-mediated signal transduction and wave propagation in astrocytic cellular networks. *J Theor Biol*, 241(3):649–68, Aug 2006.

[SMY99] D Smetters, A Majewska, and R Yuste. Detecting action potentials in neuronal populations with calcium imaging. *Methods*, Jan 1999.

[Spo02] O Sporns. Graph theory methods for the analysis of neural connectivity patterns. *Neuroscience Databases. A Practical Guide*, Jan 2002.

[SRC08] Jonathon Shlens, Fred Rieke, and Ej Chichilnisky. Synchronized firing in the retina. *Current Opinion in Neurobiology*, 18(4):396–402, Aug 2008.

[TAT+05] Guo-Feng Tian, Hooman Azmi, Takahiro Takano, Qiwu Xu, Weiguo Peng, Jane Lin, NancyAnn Oberheim, Nanhong Lou, Xiaohai Wang, H Ronald Zielke, Jian Kang, and Maiken Nedergaard. An astrocytic basis of epilepsy. *Nat Med*, 11(9):973–81, Sep 2005.

[TL08] L Tian and L Looger. Genetically encoded fluorescent sensors for studying healthy and diseased nervous systems. *Drug discovery today Disease models*, 5(1):27–35, Jan 2008.

[Tra10] Thomas P. Trappenberg. Fundamentals of computational neuroscience. page 390, Jan 2010.

[Ver06] Alexei Verkhratsky. Glial calcium signaling in physiology and pathophysiology. *Acta Pharmacol Sin*, 27(7):773–80, Jul 2006.

[VWP+09] Joshua T Vogelstein, Brendon O Watson, Adam M Packer, Rafael Yuste, Bruno Jedynak, and Liam Paninski. Spike inference from calcium imaging using sequential monte carlo methods. *Biophysical Journal*, 97(2):636–655, Jan 2009.

[WE05] D Weiskopf and G Erlebacher. 12 overview of flow visualization. *The visualization handbook*, Jan 2005.

[WM97] David H Wolpert and William G Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67 – 82, 1997.

[WSD08] Jonathon Wetherington, Geidy Serrano, and Ray Dingledine. Astrocytes in the epileptic brain. *Neuron*, 58(2):168–78, Apr 2008.

[WSEE05] D Weiskopf, F Schramm, G Erlebacher, and T Ertl. Particle and texture based spatiotemporal visualization of time-dependent vector fields. *Visualization, 2005. VIS 05. IEEE*, pages 639 – 646, Oct 2005.

[WSTB86] JG WHITE, E SOUTHGATE, JN THOMSON, and S BRENNER. The structure of the nervous-system of the nematode caenorhabditis-elegans, Jan 1986.

[YBC+09] Diana Yu, Marius Buibas, Siu-Kei Chow, Ian Y Lee, Zakary Singer, and Gabriel A Silva. Characterization of calcium-mediated intracellular and intercellular signaling in the rmc-1 glial cell line. *Cellular and Molecular Bioengineering*, 2(1):144–155, Mar 2009.

[YF06] E Yaksi and RW Friedrich. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved ca2+ imaging. *Nat Meth*, 3(5):377–383, Jan 2006.