Statistical tools and community resources for developing trusted models in biology and chemistry

Aidan C. Daly
Balliol College

Computational Biology Research Group
Computing Laboratory
University of Oxford

Trinity Term 2017

This thesis is submitted to the Computing Laboratory, University of Oxford, for the degree of Doctor of Philosophy. This thesis is entirely my own work, and, except where otherwise indicated, describes my own research.

Abstract

Mathematical modeling has been instrumental to the development of natural sciences over the last half-century. Through iterated interactions between modeling and real-world experimentation, these models have furthered our understanding of the processes in biology and chemistry that they seek to represent. In certain application domains, such as the field of cardiac biology, communities of modelers with common interests have emerged, leading to the development of many models that attempt to explain the same or similar phenomena. As these communities have developed, however, reporting standards for modeling studies have been inconsistent, often focusing on the final parameterized result, and obscuring the assumptions and data used during their creation. These practices make it difficult for researchers to adapt existing models to new systems or newly available data, and also to assess the identifiability of said models — the degree to which their optimal parameters are constrained by data — which is a key step in building trust that their formulation captures truth about the system of study.

In this thesis, we develop tools that allow modelers working with biological or chemical time series data to assess identifiability in an automated fashion, and embed these tools within a novel online community resource that enforces reproducible standards of reporting and facilitates exchange of models and data.

We begin by exploring the application of Bayesian and approximate Bayesian inference methods, which parameterize models while simultaneously assessing uncertainty about these estimates, to assess the identifiability of models of the cardiac action potential. We then demonstrate how the side-by-side application of these Bayesian and approximate Bayesian methods can be used to assess the information content of experiments where system observability is limited to "summary statistics" — low-dimensional representations of full time-series data.

We next investigate how *a posteriori* methods of identifiability assessment, such as the above inference techniques, compare against *a priori* methods based on model structure. We compare these two approaches over a range of biologically relevant experimental conditions, and highlight the cases under which each strategy is preferable. We also explore the concept of optimal experimental design, in which measurements are chosen in order to maximize model identifiability, and compare the feasibility of established *a priori* approaches against a novel *a posteriori* approach.

Finally, we propose a framework for representing and executing modeling experiments in a reproducible manner, and use this as the foundation for a prototype "Modeling Web Lab" where researchers may upload specifications for and share the results of the types of inference explored in this thesis. We demonstrate the Modeling Web Lab's utility across multiple modeling domains by re-creating the results of a contemporary modeling study of the hERG ion channel model, as well as the results of an original study of electrochemical redox reactions.

We hope that this works serves to highlight the importance of both reproducible standards of model reporting, as well as identifiability assessment, which are inherently linked by the desire to foster trust in community-developed models in disciplines across the natural sciences.

Contents

Acknowledgements

The completion of this thesis, and my time at Oxford in general, would not have been possible without help from many individuals and organizations, to whom I owe specific mentions of gratitude.

I would first like to thank my supervisors, Professor David J. Gavaghan, Professor Chris Holmes, and Dr. Jonathan Cooper, who were invaluable sources of knowledge and guidance, both personal and academic, over the last four years. I would particularly like to thank Jonathan for his patience working with me day-to-day and dedication to improving my ability to write both code and prose, Chris for advancing my knowledge of Bayesian and approximate Bayesian statistics, and David for being the architect of my project and guiding my growth as a researcher. I additionally owe great thanks to Professor Simon Tavener at Colorado State University, who helped me make great strides in my understanding of model identifiability, and who was a wonderful mentor and host throughout our collaboration.

I would next like to thank my previous research supervisors, including Professor Alán Aspuru-Guzik, Professor Ryan Adams, Dr. Roberto Olivares-Amaya and Dr. Sergio Boixo from Harvard University, Dr. Robert DeSalle, Dr. Sergios Orestis-Kolokotronis, and Dr. Angelica Cibrián-Jaramillo from the American Museum of Natural History, and Professor Dennis Shasha from NYU's Courant Institute, for their years of support and inspiration which have lead me to where I am today.

I owe tremendous gratitude to the Rhodes Trust, whose financial and administrative support have made this degree possible, and the Rhodes Scholar community at large, who have made the experience of earning the degree unforgettable.

I must also thank the Harvard-Radcliffe Kendo Club for exposing me to martial arts which have become a large part of my life, and without which I doubt I would have had the discipline to attend and complete this course of study. Furthermore, I cannot thank the Oxford University Kendo Club enough for the wonderful times over the last four years: furthering my personal growth, making life-long friends, and maintaining my sanity.

Finally, I would like to thank all my friends (both old and new) and family for their support and encouragement, especially my mother, Gloria Coruzzi, and father, Douglas Daly, who always believed in me and came to love Oxford at least as much as I did.

1

Introduction

The desire to understand the processes of the natural world has long driven scientists to attempt to re-create observed phenomena in a controlled environment. In recent decades, the scope of these environments has expanded beyond the traditional *in vitro* to the *in silico*: using computational models to explain and predict behavior in biology and chemistry. These models are attractive for many disciplines due to the low-risk and generally low-cost nature of computational simulations when compared to studies in living systems. As a result, fields such as neuronal and cardiac biology, which have direct medical applications, have benefited from the continued attention of modeling research since the inception of the field of mathematical cellular modeling in the 1940s (Hodgkin and Huxley, 1952; Noble, 1962). Since this time, advancements both in our understanding of biological and chemical processes and in the computational resources available to us have made realistic simulations more and more viable, bringing us closer to developing true *in silico* test-beds for experimentation in the natural sciences.

Beyond simply reproducing observed behavior (and predicting new behavior), models can help further our understanding of how the target system operates. For this reason, *parametric* models – those possessing a fixed mathematical structure – have distinct advantages over *phenomenological* models – those possessing little interpretable structure — for many applications in biology and chemistry (Wooley and Lin, 2005). Parametric models effectively posit that each component variable has a direct physiological interpretation, and thus that the equations relat-

ing model variables to each other dictate how components interact and evolve over time. If the parameterized model can be shown to be consistent with existing experimental data, then it supports the idea that the modeler has captured some essential truth about the target system in its formulation. Conversely, poor explanation of data may indicate that the system in question operates in a different manner to what was supposed, necessitating an alternative formulation and/or the gathering of additional data. In this manner, experimentation and parametric modeling may interact in an iterative fashion, and historically, such studies have furthered our understanding of fields such as cardiac cell biology (Noble and Rudy, 2001; Cooper et al., 2015).

If a model is to be adopted by a "user" — either a fellow modeler or an experimentalist, who may wish to leverage its predictions to make real-world decisions — they must be able to "trust" the model. Because these concerns are nearly universal in the field of modeling, we may borrow a means of evaluating trust from engineering and the physical sciences: the "verification, validation, and uncertainty quantification" (VVUQ) formalism (National Research Council, 2012). This formalism divides trust assessment into three steps. The first step, *verification*, involves ensuring that the computational model solves the underlying mathematical model with an acceptable degree of accuracy. In this stage, we seek to improve trust by minimizing the amount of programming error and numerical error that may result from software implementation of a complex mathematical model. The next stage, *validation*, involves assuring that the model reproduces phenomena observed in the target system with an acceptable degree of accuracy. In this stage, we improve trust by ensuring that the model output agrees with observations made over a range of experiments. It is worth noting that a model can never be fully validated — only validated over specific "contexts of use" (i.e., the finite set of experiments against which model predictions were found to be congruous with experimental data). The final stage, *uncertainty quantification* (UQ), involves assessing how uncertainties in model input are propagated to model output. This stage is particularly important for building trust in biological models, in which we expect to see variation in parameters (due to effects such as random noise and population variation), which must in turn produce alteration or conservation of predicted behavior

that is in line with system observation. We believe that such rigorous assessments of trust in models are important in this context, even though VVUQ in general, and UQ in particular, has yet to gain purchase in mature areas of biological modeling (Pathmanathan and Gray, 2013).

After building trust in a model, the best way to improve its impact is to disseminate it effectively to a community. Models in biology and chemistry are rarely developed in isolation, frequently requiring collaboration between multiple experimentalists and theoreticians. Even after formulation, adoption of a model by a community at large can aid in its validation on new data, or even its application to new systems of study. We posit that an effective community modeling resource should enable three main functions. First, it should enable users to enter a modeling domain and clearly see the state of the art therein. Not only the *provenance* of the model should be apparent — i.e., which assumptions were made, which data were employed to fit which components of the model — but additionally what trust metrics, using approaches such as VVUQ, were calculated. This would increase the transparency of how models were formulated for their target system, and aid in the selection of appropriate models to explain new data. Second, such a resource should make it easy to *reproduce* the results of a modeling study. Here, we define *reproduction* as "independent re-implementation of the essential aspects of a carefully described experiment," which we contrast with *replication*, defined as "re-running a simulation in exact detail" (Howe, 2012; Cooper et al., 2015). While reporting models to a replicable degree, such as making simulation code available, is important as a minimum investment in community development, such reporting practices on their own often limit our ability to apply them to new systems without substantial alterations (Drummond, 2009). As such, community modeling resources should enforce reproducible reporting standards, using high-level descriptions of models, as well as the processes used to construct and validate them. Finally, this resource should be made openly available online, in order to facilitate contributions from all members of the modeling community.

In building towards such an online community resource for sharing and curating models, we became interested in an aspect of model trust and validation that has profound implications in

the fields of biology and chemistry in particular. This was the concept of model *identifiability* — the degree to which the parameters that lead to optimal reproduction of some observed behavior are unique. As we mentioned with regards to uncertainty quantification, the presence of natural biological variation or observational noise means that there is rarely one "true" set of parameters to explain behavior in the system of study. Cases of extreme variation in optimal model parameters, however, may indicate the unsuitability of the model to represent the system, as it reduces faith in the direct biological interpretation of each parameter. When unbounded variation is present in one or more parameters, the model is deemed to be *unidentifiable*, and the modeler should consider either an alternative model formulation, experimental alterations, or both. Unidentifiability is commonly divided into two classes depending on its underlying cause. *Structural unidentifiability* describes a fundamental lack of information in the data about one or more model parameters and results in unbounded variation of the parameters even in the absence of noise. Addressing this requires an alternative model formulation, which may in turn require the collection of additional system observations. *Practical unidentifiability* describes extreme sensitivity of the recovered model parameters to noise. Unlike structural unidentifiability, this may be remedied by an increase in recording precision or experimental control (Raue et al., 2011). Concerns regarding identifiability, its causes, and means to address unidentifiability have been raised in multiple areas of biological significance, particularly in cardiac modeling (Milescu et al., 2005; Fink and Noble, 2009; Csercsik et al., 2012).

Cardiac cell models are interesting targets for both identifiability research and development of community resources, not only because of their importance and mathematical properties, but also because of the modeling practices currently employed in their construction. Motivated by applications such as drug screening and personalized medicine, the scope of cardiac modeling studies has expanded to cover a wide range of systems and experimental conditions (Clancy and Rudy, 2001; Romero et al., 2009; O'Hara et al., 2011; Yuan et al., 2015). Along with this increase in variety has come an expansion in complexity, as the community seeks to make models that reflect our increased knowledge of sub-cellular processes. While advancements in record-

ing techniques such as the patch clamp experiment (Neher and Sakmann, 1992) have allowed us to observe cells in greater detail, it is still difficult to gather sufficient data to fit these complex models during a single cellular observation. Furthermore, experiments on cardiomyocytes are largely designed without modeling in mind (Krogh-Madsen et al., 2016), instead relying on classical setups such as voltage step experiments that may be insufficient to constrain today's models (Beaumont et al., 1993; Wang and Beaumont, 2004). To compensate for the lack of sufficient data, cardiac modelers often borrow parameters or entire subunits from other models that may have been constructed for different systems or under differing experimental conditions (Cannon and D'Alessandro, 2006; Burger, 2011). While effort is taken to adjust model components for differences in species and/or experimental conditions, as well as to maintain certain macroscopic properties, such manual tuning practices are unlikely to locate truly optimal parameters (Krogh-Madsen et al., 2016), and have been shown to generalize poorly in comparative studies (Cherry and Fenton, 2006; ten Tusscher et al., 2006; Niederer et al., 2009). As a result of these practices, one of the most mature areas of biological modeling has paradoxically benefited very little from rigorous identifiability analysis. This can make it difficult for modelers or experimentalists to choose models to adopt from the myriad formulations that have been proposed to explain similar cardiac phenomena.

Despite this, the cardiac modeling domain boasts a large number of community resources and practices that can be built upon to promote the adoption of more rigorous model validation and uncertainty quantification methods. There already exist standardized, unambiguous languages like CellML (Lloyd et al., 2004; Garny et al., 2008; Miller et al., 2010) for representing model formulations, along with associated public databases to facilitate their exchange among modelers and experimentalists alike (Yu et al., 2011). Additionally, software packages such as Chaste (Mirams et al., 2013) exist that read models in these formats and provide numerical solutions, thus minimizing the amount of effort required to implement and verify models (Pathmanathan and Gray, 2014). Projects such as the functional curation extension to Chaste have eased the task of comparing simulation results by separating the notion of a *model* — the

underlying mathematics thought to represent the system — from the notion of a *protocol* — the manner in which the system is stimulated and recorded (Cooper et al., 2011). The custom protocol language of functional curation allows for the easy reproduction of experiments on new model formulations, and has been made available online as a prototype Cardiac Electrophysiology Web Lab to aid in the comparative assessment of models by members of the community (Cooper et al., 2016). These resources, and their adoption by the modeling community at large, are promising precursors for a more advanced online resource that extends the concept of the Web Lab to experiments in the modeling domain, facilitating the specification and exchange of "modeling protocols" detailing how models are constructed and validated from experimental data.

This inspired us to ask the question: can we develop online tools to tackle the twin problems of reproducible reporting and identifiability assessment in modeling experiments? While we were initially concerned with the field of cardiac cell models (for the reasons listed above), we felt that such tools could bring benefit to many related classes of models, and thus decided to broaden our investigation to general time series models in biology and chemistry, with a particular focus on differential equation models. Within this family of models, we sought to understand the following: what standards already exist, and which need to be developed, in order to support the development of reproducible differential equation models? Which methods of assessing identifiability are most appropriate to this class of models, and what are their limitations? Given that a model is identifiable, which methods for parameterizing models should we support? And finally, how do we make such resources available online in a way that eases uptake by existing modeling communities?

Our goal is to build towards an online resource where members of a given modeling community can perform, and share the results of, model fitting and validation assays with a particular focus on assessing identifiability. This resource will build upon existing standards of representation for models within the community, but increase transparency as to the manner by which models were constructed, linking parameters to data by means of reproducible protocols for

model fitting and identifiability analysis. In order to facilitate such analyses, we investigate Bayesian and approximate Bayesian techniques for assessing identifiability in general time-series models so that we may deploy an appropriate initial set of methods for this web resource and encourage their application to newly proposed models. While we have chosen cardiac action potential models as a flagship application domain for this modeling resource, and as such have focused much of our research into identifiability on these models, we demonstrate the general applicability of the abstractions used to create this web resource, as well as the algorithms implemented for it, to other problem domains.

In Chapter 2 we present background information that is necessary to interpret the models and statistical methods that we will be considering for the rest of the thesis. We present a basic introduction to cardiac cell biology, as well as to common mathematical formalisms used to capture essential features of said biology in quantitative models. We then discuss means by which the parameters for these models may be chosen, with a particular focus on Bayesian and approximate Bayesian methods and on the insight that these methods give into model identifiability. Finally, we discuss automated methods by which experiments may be designed to maximize model identifiability, and the feasibility of adopting these for cardiac and general time-series models.

In Chapter 3 we detail how approximate Bayesian computation (ABC) methods may be applied to cardiac cell models to simultaneously fit parameters and assess uncertainty about them. The output of cardiac action potential models is often restricted to "summary statistics" — low-dimensional simplifications of the true model output — either by experimental necessity or in order to simplify thinking about qualitative changes in model output. In these situations, we demonstrate how the comparative application of Bayesian and approximate Bayesian techniques can be used to assess the ability of a given set of summary statistics to constrain model parameters. Such analyses can aid the modeling community in both assessing trust in models — whether one can expect to identify a model from a given set of data — and in designing experiments with the goal of maximizing trust in the resulting model.

In Chapter 4 we perform a comparative study of a range of techniques for assessing model identifiability — notably, Fisher information-based methods, exact and approximate Bayesian methods, and a novel "measure theoretic" approach described by Daly et al. (2017b) — on relevant model problems in biology. We explore not only how factors such as model complexity, model structure, and assumptions about observational error can affect our ability to apply these techniques, but also how to interpret the results of each technique with a particular eye towards distinguishing between structural and practical identifiability. We additionally examine several automated approaches to the design of experiments that seek to maximize indices of identifiability, and compare the results of these methods to strategies involving expert intuition. As a result of this study, we are able to provide clear guidelines for how to select and apply an appropriate method for assessing and optimizing identifiability under a range of restrictions that may be present in biological systems, which can inform users of online modeling resources on how to perform and interpret similar methods themselves.

In Chapter 5 we present a framework for abstracting key components of parameter fitting and identifiability analysis techniques into classes for an object-oriented programming language. We have developed a Python implementation of this framework that has enabled our comparative analyses within this thesis, allowing us to easily plug-and-play components of modeling studies with minimal reimplementation. This framework serves as the basis for a language to describe reproducible modeling experiments within our online framework, allowing for the general application of techniques for fitting and validation across varying model formulations.

In Chapter 6 we present a prototype implementation of an online resource for model development and curation in cardiac models. We build upon the existing Cardiac Electrophysiology Web Lab for describing virtual experiments on cardiac models to allow the incorporation of data and the specification (and execution) of selected techniques for fitting and identifiability analysis. We demonstrate how this "Modeling Web Lab" may be used to reproduce, visually render, and share the results of a modeling study carried out by Beattie et al. (2017), who employed Bayesian techniques to fit a new model of the hERG potassium channel using original

experimental data. We finally show the extensibility of such a community modeling resource by presenting a version created for models of electrochemical redox reactions, where we highlight the increased ability of "alternating current" (AC) protocols to constrain parameters of a given reaction model when compared to simpler "direct current" (DC) protocols.

In Chapter 7, we present the conclusions from this research, and discuss the future steps that remain to be taken in order to further develop these community modeling resources for the cardiac modeling community, and beyond.

2

Background

Before detailing work that has been done to progress the development of a Modeling Web Lab, we will explore the biology of our primary application domain — cardiac cells — as well as the models used in their study and the techniques that have been used to parameterize said models. In Section 2.1 we discuss the biology of the cardiac action potential, two classes of mathematical models that are used to represent this behavior, and the types of data that may be used to fit these models. In Section 2.2, we introduce three basic classes of inference methods that are employed in this research, along with the conditions under which a modeler may choose to employ each. Finally, in Section 2.3, we discuss automated means of designing experiments to maximize model identifiability, which will be employed in Chapter 4 and compared against our own novel methods for doing the same. This informed approach to designing experiments promises to improve the quality of models being proposed in a given field, and we will detail how the techniques employed and results obtained from these studies may be made available on our prototype Modeling Web Lab in Chapter 6.

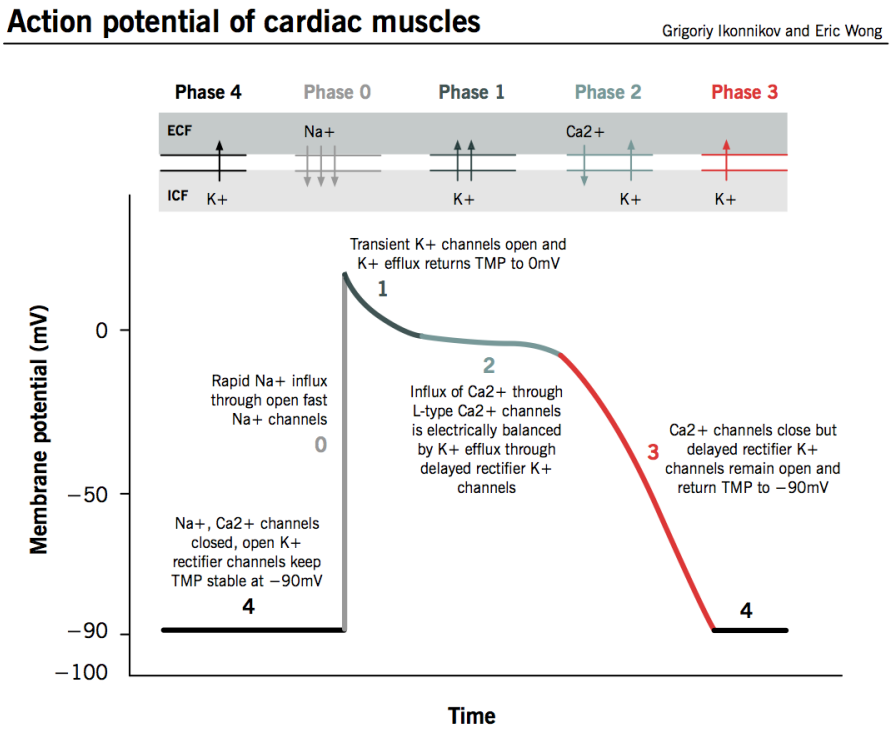
Given the broad scope of this research, in this chapter we present only an overview of methods used later in the thesis. Further detail and literature review will be provided at the beginning of subsequent chapters in order to introduce information as it becomes relevant to the discussion.

2.1 The Cardiac Action Potential

The heart relies on regular contractions in order to move blood through its chambers and into circulation in the body. Such contractions are possible because the heart is primarily muscular tissue composed of *myocytes* — long, rod-like cells containing proteins that allow them to compress along their long axis at the expense of energy. The contraction of these individual cells is triggered by a characteristic electrical impulse across the cell membrane known as an action potential (AP). We describe the fundamental operation of the action potential below, and the reader may refer to Mohrman and Heller (2010) for further details.

The action potential describes the change in the electrical potential across the cell membrane — the "transmembrane potential," or TMP — of a cardimyocyte over time. This change is accomplished through the movement of ions across the cell membrane. When there is a net outward movement of positive ions, the membrane potential becomes negative, and when there is a net inward movement of positive ions, the membrane potential becomes positive. The movement of ions across the cell membrane, which is normally impermeable to charged particles, is made possible by specialized proteins, one family of which are called *ion channels*. Each cell contains a variety of types of ion channels, each of which selectively allows the transport of a single type of ion, such as sodium (Na^{+}), potassium (K^{+}), or calcium (Ca^{2+}). Furthermore, each channel can adopt a range of physical conformations, or "states," only allowing the passage of ions when it is in an "open" state. Ion channels may alter conformation in response to changes in TMP, or automatically after a period of time spent in one conformation. It is through these voltage- and time-dependent conformational changes that the "open" activity of ion channels may be restricted temporally, allowing for the TMP to go through a series of defined changes that comprise the action potential (Figure 2.1).

Figure 2.1 A graphical representation of the action potential in cardiomyocytes. "ECF" and "ICF" labels on the top diagram denote the extracellular and intracellular faces of the cell membrane (respectively). Reproduced from `http://www.pathophys.org/physiology-of-cardiac-conduction-and-contractility/`.



The action potential begins and ends at a "resting" phase, where the TMP attains a constant, negative value known as the *resting potential*. This is maintained by a steady outflow of K^+ ions through inward rectifier channels, while all Na^+ and Ca^{2+} channels remain closed. When the cell is subjected to a *stimulus current*, which induces a small positive change in the TMP, fast Na^+ channels open, triggering a large influx of positive ions that *depolarize* the cell. These channels rapidly self-inactivate, leaving the cell at a slightly positive TMP referred to as *peak potential*. Immediately following this, transient outward K^+ channels open, enhancing the efflux of positive ions and bringing the TMP close to zero. This potential is maintained for a "plateau phase" by the balancing influx of Ca^{2+} ions through L-type (for "long-opening") calcium channels that begin to open during the initial depolarization. This influx of calcium ions triggers the cell to release further Ca^{2+} ions, sequestered internally in the sarcoplasmic

reticulum, which in turn triggers physical contraction. Finally, the cell undergoes *repolarization* when the Ca^{2+} channels close and the continued efflux of K^{+} ions causes the membrane to re-attain a negative potential. During this time, the initial concentration gradients of the ions (high extracellular Na^{+} and Ca^{2+} and high intracellular K^{+}) are restored by specialized membrane proteins called *ion pumps* and *ion exchangers* at the expense of energy, preparing the cell for another action potential.

The action potential propagates throughout cardiac tissue to trigger organ-level contraction via *gap junction* proteins, which link adjacent cardiomyocytes. These gap junctions electrically couple cells, allowing the depolarization of one cell to serve as a stimulus to trigger action potentials in its neighbors. Action potentials initiate from self-exciting "pacemaker" cells, which are localized to the sino-atrial node in the right atrium of the heart. The excitation then passes slowly through the atria to the atrio-ventricular node, allowing time for the atria to contract before the impulse moves quickly down the bundle of His and into the Purkinje fibers, causing ventricular contraction. The differential rates at which the action potential propagates through regions of the heart is achieved through the presence of many different physically localized cell types with differing conductive properties. This allows the heart, under normal conditions, to contract its various chambers in distinct, sequential steps, thus ensuring the orderly flow of blood through the organ, and consequently, the circulatory system. Cardiac tissue possesses an additional property, "refractoriness," conferred by ion channel kinetics, which prevents secondary activation for a period after returning to resting potential. This further aids in the propagation of orderly contraction by resisting "re-entrant" signals, which feed back upon themselves rather than completing their normal circuit.

While there is a great deal of research dealing with mathematical modeling of tissue- and organ-level behavior of the heart, the focus of this thesis will be largely on single-cell action potential models which serve as the fundamental building blocks of these larger models. In the subsequent sections, we discuss the two most common formulations for these models.

2.1.1 Mathematical modeling

The first mathematical model of an action potential was created by Hodgkin and Huxley to represent an axon action potential, which is shorter than a cardiac action potential and has a slightly different morphology (Hodgkin and Huxley, 1952). The formalisms of this model, however, were quickly adapted to the field of cardiac biology by Denis Noble, who created a similar model to represent the action potential of Purkinje fibers (Noble, 1962), and have been employed so extensively to this day that a class of eponymous "Hodgkin-Huxley style models" has emerged in the cardiac community.

Hodgkin and Huxley theorized that the squid axon acted like an electrical circuit, with transmembrane current being carried by either the capacitative effects of the membrane itself or by the mediated movement of ions across the membrane. This lead to the formulation of the following ordinary differential equation (ODE) to describe transmembrane potential V_m, a formulation which is conserved in practically every model of the action potential to this day:

$$\frac{dV_m}{dt} = \frac{-1}{C_m} I_{tot}, \tag{2.1.1}$$

where C_m denotes the capacitance across the cell membrane and I_{tot} denotes the total transmembrane ionic current. Both capacitance and current are often given in units per fixed area of cell membrane (e.g., $\mu F/cm^2$ or $\mu A/cm^2$, respectively) to control for variability in cell size.

Another feature of the Hodgkin-Huxley model that is conserved across most modern AP models is the manner in which individual ion channels and pumps contribute to the total ionic current. Each channel or pump has its own, independent current associated with it, and thus the total ionic current is calculated as the simple sum of these component currents using:

$$I_{tot} = \sum_{x=1}^{N_c} I_x, \tag{2.1.2}$$

where $\{I_1, \ldots, I_{N_c}\}$ is the set of distinct currents in the cell.

Due to the community consensus on this circuit interpretation, the great variety found in

AP models being proposed today stems from both the currents considered to comprise I_{tot} and the formalism used to model the evolution of each individual current. Hodgkin and Huxley considered three transmembrane ionic currents in explaining the neuronal action potential: I_K, the current carried by potassium ions, I_{Na}, the current carried by sodium ions, and I_l, a catch-all leakage current. While the Noble model did not account for any new ion species, a second, inward rectifying potassium current was added to account for the long plateau phase of the cardiac action potential, slowing the time until the resting potential was regained. Continuing in this manner, new currents were added, or existing currents subdivided, by successive models in order to reflect our evolving understanding of ion channel activity. A more complete history of the discovery and inclusion of the ion channel species used in most modern models is beyond the scope of this thesis, but can be found in (Noble and Rudy, 2001; Rudy and Silva, 2006; Noble, 2007; Noble et al., 2012). Instead, in the subsequent sections we choose to focus on the two families of models most commonly used to represent ion channel currents, and what implications their usage has for the validity and identifiability of the resulting action potential model.

2.1.2 Hodgkin-Huxley style ion channel models

The first class of ion channel models we will discuss are the Hodgkin-Huxley style models mentioned in the previous section. Following the interpretation of the original authors, the current passing through a given species of ion channel is calculated by Ohm's law (Equation 2.1.3), which depends on both the time-varying conductance g_x associated with that channel, as well as the membrane potential V_m and the unique "reversal potential" E_x associated with the ion species being transported, that is:

$$I_x = g_x(V_m - E_x). \tag{2.1.3}$$

The value of the reversal potential dictates an equilibrium point at which no net movement of ions is observed, and thus the difference between this reversal potential and the membrane

potential in Equation 2.1.3 creates a driving force to move ions across the membrane. The reversal potential is usually calculated using the Nernst equation,

$$E_x = \frac{RT}{zF} \log \frac{[\text{ion}]_o}{[\text{ion}]_i}, \tag{2.1.4}$$

which uses information on the chemical gradient of ions $[\text{ion}]_o/[\text{ion}]_i$. Here $[\text{ion}]_o$ is the extracellular concentration and $[\text{ion}]_i$ is the intracellular concentration, R is the ideal gas constant, T is the temperature, F is Faraday's constant, and z is the charge of the ion in question.

The true hallmark of a Hodgkin-Huxley style model, however, is in the manner in which the channel conductance is calculated. In these models, conductance — the permeability of the channels to electric current — is thought to be a function of membrane voltage and time. More specifically, it is thought to be a function of the fraction of channel proteins of the given type that are in an "open" configuration (allowing the passage of ions) and a constant maximum conductance G_x representing the maximum conductance if all such channels were open. This is represented in Equations 2.1.5 and 2.1.6, which we will discuss below:

$$g_x(V_m, t) = G_x \prod_{i=1}^{M_x} (s_{x,i}(V_m, t))^{k_{x,i}}, \tag{2.1.5}$$

$$\frac{ds_{x,i}}{dt} = f(V_m, t, s_{x,i}), \quad s_{x,i}(0) = s_{x,i,0}, \quad 0 \leq s_{i,x}(t) \leq 1 \quad \forall s_{x,i}, t. \tag{2.1.6}$$

This formalism treats each ion channel as a collection of M_x types of protein subunits, each with multiplicity $k_{x,i}$, that must all be in an "active" state in order to allow for the transport/exchange of ions across the channel as a whole. The activation probability of each of these subunits is modeled by a gating variable $s_{x,i}$, which can also be thought of as the active fraction of each subunit across the whole membrane. Thus, as shown in Equation 2.1.6, each gating variable evolves according to an ODE with initial condition $s_{x,i,0}$ and is bounded between 0 and 1, leading to a system of ODEs for the calculation of the ionic current.

For a concrete example, consider the formulation of the sodium current in the original

Hodgkin-Huxley model, reproduced below:

$$g_{Na} = G_{Na} m^3 h, \tag{2.1.7}$$

$$\frac{dm}{dt} = \alpha_m(1-m) - \beta_m m, \quad \frac{dh}{dt} = \alpha_h(1-h) - \beta_h h, \tag{2.1.8}$$

$$\begin{aligned} \alpha_m(V_m) &= \frac{0.1(V_m+25)}{\exp\left(\frac{V_m+25}{10}\right)-1}, \quad \beta_m(V_m) = 4\exp\left(\frac{V_m}{18}\right), \\ \alpha_h(V_m) &= 0.07\exp\left(\frac{V_m}{20}\right), \quad \beta_h(V_m) = \frac{1}{\exp\left(\frac{V_m+30}{10}\right)+1}. \end{aligned} \tag{2.1.9}$$

In Equation 2.1.7, we see that each sodium channel is theorized to be composed of four subunits — three of type "m" and one of type "h" — which must all be active in order to allow the passage of ions. The activation and inactivation of these subunits is described by Equation 2.1.8, which dictates that at each time point, each inactive subunit independently activates with some probability (α_m for "m" subunits and α_h for "h" subunits) and each active subunit independently inactivates with some probability (β_m for "m" subunits and β_h for "h" subunits). Each of these probabilities is in turn a function of transmembrane voltage (Equation 2.1.9). By solving these equations numerically from initial conditions associated with the resting state, we see that they produce a characteristic spiking pattern in the sodium conductance, which leads to the rapid membrane depolarization observed in the action potential.

The parameters of interest in these models are the maximum conductance parameters G_x, which are related to the expression level of the associated protein in the cell, and the constants of the rate equations (Equation 2.1.9), which control the kinetics of opening and closing. In order to obtain data to fit both conductance and kinetic model parameters, Hodgkin and Huxley made use of the voltage clamp experimental setup, which holds membrane potential at a constant value by means of a negative feedback circuit and measures the resulting ionic current (for a more detailed description of the experimental setup, see Halliwell et al. (1994)). Voltage

clamp experiments have enjoyed enduring popularity as data sources for fitting Hodgkin-Huxley models, despite concerns that the recordings may not contain enough information to fully constrain today's complex models (Wang and Beaumont, 2004). More recently, the development of the patch clamp technique, a variant of the voltage clamp technique allowing observation of membrane current across an area of the cell membrane containing one or a few ion channels (Neher and Sakmann, 1992), has revealed more information about individual ion channel dynamics. The availability of these data prompted re-examination of the formulations used to represent key currents in established Hodgkin-Huxley style models (Luo and Rudy, 1991, 1994), as well as the creation of a new class of models that were able to capture these dynamics more accurately, which we will discuss below.

2.1.3 Markov style ion channel models

Based on information gained from patch clamp recordings of the transmembrane current through single ion channels, as well as data from independent studies of channel structure, researchers in the early 1990s began proposing a class of models known as Markov models to represent cardiac ionic current. Unlike Hodgkin-Huxley models, which treat proteins as being composed of subunits that may independently attain different states, Markov models consider a fixed set of states for the protein as a whole. More importantly, the rates of transition between states of a Markov model are dependent on the current state of the model, with some states being unreachable from others.

If we define the states of a Markov model as $[S_1, \ldots, S_n]$, the probability of being in state S_i at time t as $P(S_i, t)$, and the probability of moving from state S_i to state S_j as $P(S_i \rightarrow S_j)$, then we can model the stochastic behavior of a single channel using the following "master equations":

$$\begin{aligned} \frac{dP(S_i,t)}{dt} &= \sum_{j=1}^{n} \left[P(S_j,t)P(S_j \rightarrow S_i) - P(S_i,t)P(S_i \rightarrow S_j)\right] \quad \forall i \neq n, \\ P(S_n,t) &= 1 - \sum_{j=1}^{n-1} P(S_j,t). \end{aligned} \tag{2.1.10}$$

For whole-cell models, however, the sheer number of channels present in the membrane makes it extremely impractical to model the state of each one independently. As a result, models at this level are generally concerned with predicting the *fraction* of channels in any given state, similar to the interpretation of gating variables in Hodgkin-Huxley style models (Equations 2.1.5, 2.1.6). If we consider s_i to be the fraction of channel proteins in state S_i at time t, and transition rates $r_{i,j}$ to represent the frequency with which channels move from state S_i to S_j, then the master equation becomes the following set of ODEs:

$$\begin{aligned} \frac{ds_i}{dt} &= \sum_{j=1}^{n} (s_i r_{i,j} - s_j r_{j,i}) \quad \forall i \neq n, \\ s_n &= 1 - \sum_{j=1}^{n-1} s_j, \end{aligned} \tag{2.1.11}$$

$$r_{i,j} = f_{i,j}(V_m, t), \tag{2.1.12}$$

where the occupancy of each channel state evolves based on the net movement of channels to/from connected states. This is controlled by the rate parameters $r_{i,j}$, which are thought in turn to be parametric functions of membrane potential and time (Equation 2.1.12). From the solution to these ODEs, one can calculate the conductance g_x through a given ion channel species in a manner almost identical to that of a Hodgkin-Huxley model:

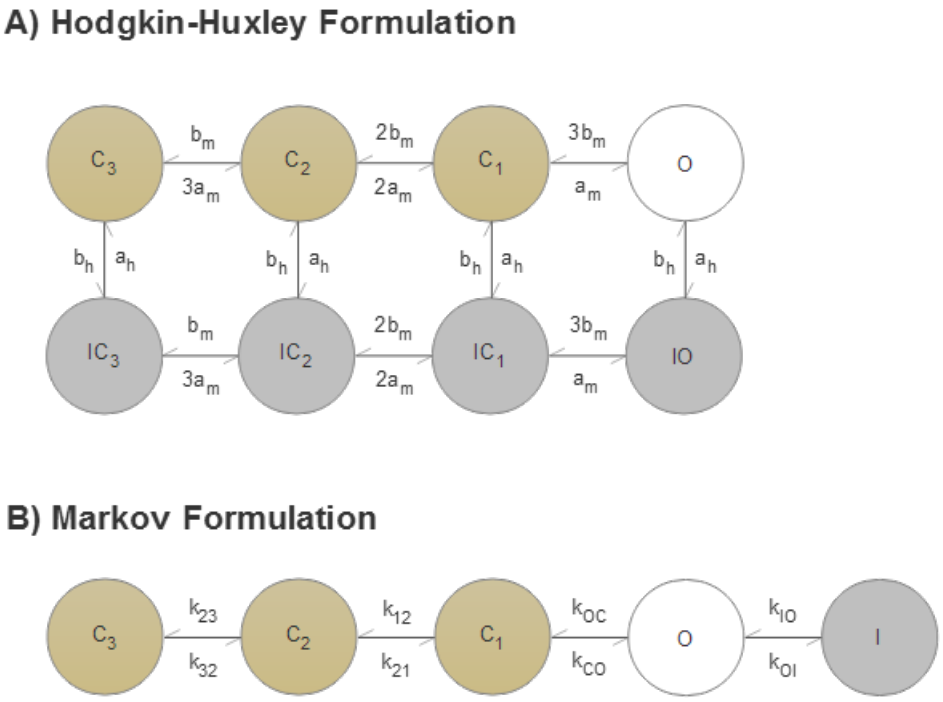
$$g_x(V_m, t) = G_x s_{x,O}, \tag{2.1.13}$$

where $s_{x,O}$ is the open fraction of channels of type x and G_x is the maximum conductance that can be achieved through said channels.

As with the Hodgkin-Huxley models in the previous section, the parameters of interest in these models are the maximum conductance parameters G_x, and the parameters controlling the transition rates $r_{i,j}$. These functions are often assumed to take a similar exponential form to those presented in Equation 2.1.9, and studies employing patch clamp data often seek to determine the parameters that best reproduce the observed kinetic behavior (Beattie et al., 2017).

The similarities between Markov and Hodgkin-Huxley models are more than superficial. Markov models are, in fact, a superset of Hodgkin-Huxley models, giving them an ability to represent a wider range of protein activity. In Figure 2.2A, we see a Markov state diagram of the four-subunit Hodgkin-Huxley model for sodium conductance (Equation 2.1.7), which effectively produces three "closed" states (distinguished by how many "m" subunits are closed), an "open" state, and an "inactive" state — a non-conducting state physically distinct from being "closed" that depends on the conformation of the "h" subunit. In Figure 2.2B, we see a Markov model possessing the same states, but distinct from the Hodgkin-Huxley formalism in that the "inactive" state can only be reached from the "open" state, thus preventing its representation by a set of Hodgkin-Huxley style equations. Certain ion channels, such as those responsible for the fast sodium current, have been shown to undergo state-dependent as well as voltage-dependent transitions, being more likely to enter an inactive state from an open state than a closed one (Aldrich et al., 1983). Wang et al. (1997) observed similar state-dependent inactivation in the hERG channel, a channel whose disruption by drug action has been linked to fatal arrhythmias (Roy et al., 1996; Suessbrich et al., 1996), leading them to formulate one of the first widely-adopted cardiac models employing Markov formalisms. Although it has been shown that modification of the standard Hodgkin-Huxley equations can mimic state-dependent behavior in ion channels (Marom and Abbott, 1994), Markov models are still generally preferred for describing such kinetic behavior.

Figure 2.2 Example state diagrams for Markov models of ion channels. Ion channel states are represented by circles, with double-ended arrows connecting states that can be sequentially traversed. States may be "inactive" (I), "closed" (C), or "open" (O), and the kinetic rates of (reversible) transition between connected states are indicated by the labels on the directional arrowheads. A) An equivalent Markov diagram for the original Hodgkin-Huxley sodium channel model (Equations 2.1.7–2.1.9). B) Markov diagram of a four-state model that cannot be represented in the Hodgkin-Huxley formalism.



Despite this, Hodgkin-Huxley models are often deployed instead of Markov models, largely due to concerns of computational complexity. Markov models have greatly expanded state spaces when compared to Hodgkin-Huxley models, leading to more parameters and (often) stiffness in the system of ODEs. These factors make Hodgkin-Huxley models more attractive for larger scale simulations (Cannon and D'Alessandro, 2006; Fink and Noble, 2009). While the use of Hodgkin-Huxley formalisms requires the simplification of ion channel dynamics, which can alter performance of the model under conditions such as simulated blocking of the channel by drugs (Liu and Rasmusson, 1997), the models may perform almost indistinguishably under normal physiological conditions. Thus, the context of use for whole-cell models should

be considered before making any simplifications. Indeed, Markov models are often used in conjunction with simpler Hodgkin-Huxley formulations when essential processes such as calcium dynamics cannot be safely simplified (ten Tusscher et al., 2003, 2006).

While Markov models have the potential to capture ion channel behavior more accurately, however, they are often proposed without modeling in mind. Firstly, the geometry of Markov models of ion channels — both the number of states and connectivity — are generally decided upon in an *ad hoc* manner based either on knowledge of protein structure (Burger, 2011) or qualitative properties of recorded signals. Without investigation into alternate formulations, one cannot be sure that our chosen model is superior to, or even distinguishable from other structures (Milescu et al., 2005; Beattie et al., 2017). Such uncertainty limits the amount of information we can gain by interpreting the model structure. Furthermore, with the increased number of variables comes an increased risk of the model containing unidentifiable parameters (Xuyan et al., 2012). Unidentifiability in model parameters throws the truth of the model formulation into question, and a review of 13 commonly employed cardiac cell models by Fink and Noble detected unidentifiabilities in Markov components of 9 of them (Fink and Noble, 2009). The authors recommended replacing these over-parameterized components with simpler Hodgkin-Huxley formulations to improve the interpretability of the overall model. Thus, while Markov models have the potential to provide great predictive power, they should only be employed over Hodgkin-Huxley formulations when demonstrated to be both identifiable and computationally worthwhile for the current context.

2.1.4 Summary statistics of the action potential

Because the output of cardiac action potential models is generally current or voltage evolving over time, the ideal data used to fit these models would be current or voltage time course data. As mentioned in the previous sections, these data are often recorded by voltage clamp or patch clamp experiments, which are capable of making very fine-grained recordings (on the order of 10 measurements per millisecond in Beattie et al. (2017) — see Polder et al. (2005) for a

full discussion on the technical capabilities of clamp experiments). Because such voluminous data are difficult to reason about, experimentalists may employ "summary statistics" — low-dimensional quantities that seek to describe essential features of the full data — to report on action potentials.

In Figure 2.3, we show a typical cardiac action potential annotated with five summary statistics commonly employed by cardiac modelers:

1. Peak potential - the largest positive membrane potential observed over the course of the action potential.

2. Resting potential - the steady-state membrane potential (approximated by the most negative membrane potential observed during the action potential).

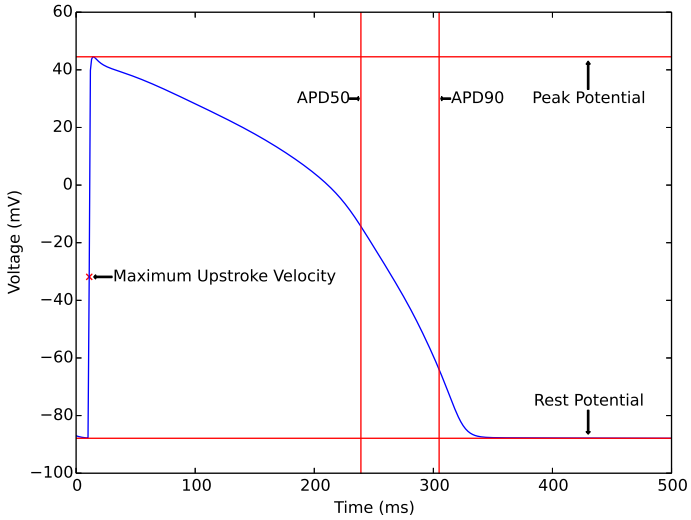
3. APD90 - the time between the membrane depolarizing and repolarizing to 90% of the difference between its resting and peak voltage.

4. APD50 - the time between the membrane depolarizing and repolarizing to 50% of the difference between its resting and peak voltage.

5. Maximum upstroke velocity - the largest value of dV_m/dt observed during the action potential.

These summary statistics are particularly used to reason about changes in AP morphology that may result in arrhythmias, or other potentially fatal disruption of cardiac activity. Values for quantities like action potential duration (APD) are thought to lie in a defined physiological range, and deviations from this are often associated with disease phenotypes (Corrias et al., 2010). As a result, summary statistic data are often used when assessing the *sensitivity* of proposed models — how perturbations in model parameters affect quantities of interest about the output, such as APD (Romero et al., 2009). If summary statistics of the action potential are essentially conserved when parameters are varied, it demonstrates robustness of the system to said parameters. Conversely, if parameter variation leads to deviation of simulated summary

Figure 2.3 Visual representation of five common summary statistics of the cardiac action potential. Vertical lines indicate the times (ms) corresponding to APD50 and APD90, while horizontal lines indicate the voltages (mV) corresponding to peak and rest potential. The red "x" indicates the point at which dV/dt (mV/ms) is greatest, i.e. the point of maximum upstroke velocity.



statistics from the physiological acceptable range, the model is deemed to be sensitive to the parameter(s) in question. These effects are of interest in both drug safety assays, where drug block of ion channels is simulated by the adjustment of relevant conductance or kinetic parameters (Corrias et al., 2010; Mirams et al., 2011; Yuan et al., 2015), or research in personalized medicine, where one may simulate the effects of inter-subject variability by varying ionic parameters over a physiological range (Romero et al., 2009; Britton et al., 2013).

Summary statistic data are often also employed by necessity, when *in vivo* conditions prevent the gathering of time course voltage data. Patch and voltage clamp experiments rely on electrodes that maintain a suction contact with, and puncture, the cell membrane, which is not possible *in vivo* due to the mechanical contractions of the heart. Instead, recordings done in live subjects rely on floating or non-suction contact electrodes mounted on catheters, which may only be able to record low-dimensional summaries such as APD (Corrado et al., 2015, 2016). While there exist techniques to record a "monophasic action potential" (MAP) from

non-suction, catheter-mounted electrodes, and this recording may be used to reconstruct the true action potential as one might record from a standard clamp technique (Franz et al., 1986; Ino et al., 1988), the quality of these recordings is highly dependent on experimental conditions and tissue type, and the interpretation of these signals is known to be an error-prone process (Franz, 1999; Kadish, 2004). In fact, the quality of MAP recordings is validated against traditional clamp recordings of the action potential by comparing summary statistics such as APD50 and APD90, further affirming the prevalence and usefulness of these simplified data (Ino et al., 1988). A final method for recording action potentials *in vivo* is optical mapping, which uses fluorescent probes to measure quantities such as membrane voltage and calcium dynamics, but currently requires that the heart be removed from the subject first, which understandably limits its appeal for human subjects (Liang et al., 2011).

Whether by convenience or necessity, we see that summary statistics of the action potential are commonly employed in cardiac modeling. In the subsequent sections, we will discuss the manner in which data are used to fit cardiac models, and the implications that employing summary statistic data during this process have for the fitting techniques that modelers may employ.

2.2 Inference Methods

After the form of a parametric model is chosen, a modeler must then determine the values of the parameters (in the case of cardiac modeling, either the maximum conductance parameters, ion channel kinetic parameters, or both) that reproduce some observed behavior to an acceptable degree. In the early days of Hodgkin and Huxley, due to the simplicity of the model and the data, this could be done by hand, plotting out solutions on graph paper that matched the observed traces by eye. As models have evolved in complexity, this is no longer feasible, and a host of automated methods for performing parameter inference have emerged. In the subsequent sections, we will detail three main classes of algorithms that may be used to parameterize

cardiac models, and more generally, models dealing with time series data. We also discuss the conditions under which they are best suited for application. Finally, we will explain how the application of certain methods may aid in the assessment of model identifiability, which we would like to encourage as a practice for users of our Modeling Web Lab.

2.2.1 Parameter optimization

The most basic approach to parametric model fitting is to choose the parameters that "best" reproduce some set of observed data. This "goodness of fit" is quantified by a distance function $\mathcal{D}(\mathbf{y}, f(\theta))$ chosen by the modeler to evaluate proximity between observed data $\mathbf{y}$ and simulated data $f(\theta)$, where θ is a set of all parameters to be chosen. An example of such a distance function for time series data with N points is the square error function:

$$\mathcal{SE}(\mathbf{y}, f(\theta)) = \sum_{i=1}^{N} (\mathbf{y}_i - f_i(\theta))^2, \tag{2.2.1}$$

which sums the square of differences between corresponding simulated output points $f_i(\theta)$ and observed output points $\mathbf{y}_i$ over the full time trace.

Once an appropriate distance function is chosen, the problem of finding optimal parameters boils down to solving the following minimization problem:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{D}(\mathbf{y}, f(\theta)), \tag{2.2.2}$$

which seeks to find the parameters θ^* that produce simulation output that is minimally distant from the observed data. The difficulty of this problem is influenced by many factors, including the number of free parameters, the structure of the model, and the character of the distance function. Because of this, there are many competing methods available, and members of the cardiac modeling community have employed a variety of these methods in their studies.

One class of methods that has found great purchase in the field of cardiac modeling are gradient descent methods. These methods employ information about the derivative of the distance function with respect to the free model parameters in order to find local optima in the distance

function — points which are defined in the following manner:

$$\theta^* \quad \text{s.t.} \quad \frac{\partial \mathcal{D}(\mathbf{y}, f(\theta^*))}{\partial \theta_i} = 0 \quad \forall i = 1, \ldots, P, \tag{2.2.3}$$

where $[\theta_1, \ldots, \theta_P]$ are the free parameters of the model. Beginning at some initial set of parameters, these algorithms iteratively update their guess as to the optimal parameter θ^* by taking steps in parameter space proportional to the negative of the gradient of the distance function $\mathcal{D}$ evaluated at the current guess. This ensures that each iteration updates the parameter estimate in the direction of steepest decrease in distance, ensuring a principled progression towards a local minimum (which can be distinguished from a local maximum by examining second-derivative information).

While these algorithms are appealing in their simplicity, there are several drawbacks to gradient-based approaches. Firstly, there are no guarantees that the local minima that are found are globally optimal solutions. Without rules for more advanced exploration, these algorithms give no indication of the potential presence of other optima, and risk becoming stuck on solutions that explain training data decently but will generalize poorly (Dokos and Lovell, 2004; Wang and Beaumont, 2004). Additionally, these methods may fail to converge when the function surface is particularly jagged or particularly flat, in which case steps chosen by gradient evaluation may be excessively large or small, respectively, causing the algorithm to jump around wildly or stagnate in a particular region of parameter space. Despite these concerns, gradient descent methods such as the Newton method, the Gauss-Newton method, and the Levenberg-Marquardt algorithm have been used to parameterize some of the most popular cardiac models employed today (Wang et al., 1997; O'Hara et al., 2011).

Alternatively, modelers may employ gradient-free methods for optimization, such as simulated annealing, genetic algorithms, or particle swarm optimization (PSO), which seek to improve global exploration. Simulated annealing algorithms employ ideas from thermodynamics, beginning from an initial estimate and proposing moves through parameter space according to a random function chosen by the modeler. Moves are accepted or rejected depending on

the difference in the quality of the parameters (as measured by the distance function) and a "temperature" parameter. During the execution of the algorithm, moves to worse parameter estimates are accepted with some probability, which decays over time as the system "cools." A well-chosen "cooling schedule" promises to balance between local optimization and global exploration, and as such, these algorithms have found purchase in the field of action potential modeling (Vanier and Bower, 1999). Genetic algorithms and PSO both attempt to guarantee global exploration by evaluating many parameter sets simultaneously, iteratively guiding an initially random "population" of parameter sets towards a global optimum using ideas from evolutionary or social biology, respectively. Both genetic algorithms (Syed et al., 2005; Bot et al., 2012; Kaur et al., 2014; Groendaal et al., 2015) and particle swarm optimization (Weber et al., 2008; Chen et al., 2012) have been applied extensively to cardiac models in recent years. These gradient-free methods avoid the dangers of being trapped in local minima by moving through parameter space without considering the function landscape, but may be slower than gradient descent methods in certain situations for this reason (Szekely et al., 2011).

A final issue with simple optimization of model parameters is that it is difficult to determine model identifiability from a single, optimal point in parameter space. While repeated execution of gradient-based techniques or the use of population-based methods such as genetic algorithms can improve our confidence in finding a global optimum, the results of these methods do very little to tell us about the "landscape" of our model output about that point. How much more fit is our global optimum than its neighbors in parameter space? How sensitive is this optimum to changes in each parameter? While we can attempt to answer these questions by using gradient information at the optimal parameters to calculate indices of uncertainty such as standard confidence intervals, these metrics are only accurate when the size of the data is large relative to the number of parameters, measurement noise is relatively small, and model output is roughly linearly dependent on model parameters (Raue et al., 2011). When one or all of these conditions are violated, as may frequently happen in biology, interpreting curvature-based sensitivity indices may lead us to make incorrect conclusions about model identifiability. In order to ad-

dress this, in the next two sections we will detail methods of *posterior inference* that may be used to perform parameter inference and identifiability analysis side-by-side, and which will be employed to demonstrate the functionality of our Modeling Web Lab.

2.2.2 Bayesian inference

We move now away from optimization techniques, which seek to find a single point in parameter space that best reproduces the observed data, and towards *posterior inference* techniques, which seek to produce a probability distribution over parameter space detailing the likelihood of each set of parameters reproducing the observed data. We can still determine the most optimal, or in our probabilistic terminology, "most likely," parameters by examining the mode of this distribution, while gaining additional information about model identifiability by examining the spread of the distribution. A posterior that is highly peaked about the mode indicates unique identifiability, while uniformity over a range of values may indicate unidentifiability. Interpretation of identifiability from posterior parameter distributions, including how to distinguish between practical and structural identifiability, will be discussed more fully in Chapters 3 and 4.

Bayesian inference requires two things to produce a posterior estimate over model parameters: a *prior distribution* and a *likelihood function* (Gelman et al., 2014). The prior distribution is a probability distribution over parameter space, which we will denote as $\pi(\theta)$, which describes the likelihood of parameters before observing any data. This can be used to incorporate previous information about the system to be modeled, such as the physiological range of parameters, or the manner in which parameters are believed to vary over a population. The likelihood function, which we denote by $\mathcal{L}(\mathbf{y}, f(\theta))$, is a metric of goodness of fit between the simulated data and the observed data. Unlike the distance metrics mentioned in the previous section, the likelihood function is a measure of *proximity* between the two sets of data. In fact, this metric quantifies $P(\mathbf{y}|\theta)$ — the probability of observing data $\mathbf{y}$ if the "true" parameters of the system are θ. This allows us to calculate the posterior likelihood of a given set of parameters using

Bayes rule:

$$P(\theta|\mathbf{y}) = \frac{P(\mathbf{y}|\theta)P(\theta)}{P(\mathbf{y})} \propto \mathcal{L}(\mathbf{y}, f(\theta))\pi(\theta), \tag{2.2.4}$$

where the "prior predictive" $P(\mathbf{y})$ in the denominator is calculated by integrating $P(\mathbf{y}|\theta)$ over all values of θ, using the Law of Total Probability.

As a concrete example, let us consider a common case in biology where we assume that a set of time-series observations $\mathbf{y}_i$, $i = 1, ldots, N$ are perturbed from the "true" values, predicted by the model $f_i(\theta)$, by Gaussian measurement noise. This gives us the following statistical model for our observations:

$$\mathbf{y}_i \sim \mathcal{N}(f_i(\theta), \sigma^2), \tag{2.2.5}$$

and thus our likelihood function $\mathcal{L}(\mathbf{y}, f(\theta))$ becomes the product of the probability density function of our Gaussian distribution $g(\mathbf{y}_i)$ evaluated at each time point:

$$\mathcal{L}(\mathbf{y}, \theta) = \prod_{i=1}^{N} g(\mathbf{y}_i). \tag{2.2.6}$$

For all but the simplest models, it is impossible to enumerate all possible parameter sets and form a posterior distribution by direct application of Equation 2.2.4. As a result, the most common approaches to Bayesian inference involve sampling strategies, where a population of estimates is drawn from the prior distribution and assigned a posterior likelihood by the application of Equation 2.2.4. The most popular approaches by far across most problem domains are Markov Chain Monte-Carlo (MCMC) algorithms, which propose a series of random steps through parameter space according to a defined proposal distribution. A proposed step is accepted or rejected probabilistically based on the *ratio* of the likelihood function evaluated at the proposed location to the likelihood function evaluated at the current position, biasing exploration towards regions of greater likelihood (and thus reducing the number of steps required to explore the most interesting regions of parameter space), but maintaining a chance of moving to a "worse" location to avoid becoming stuck in local minima. This has the added benefit of avoiding the calculation of the prior predictive $P(\mathbf{y})$ in Equation 2.2.4, which is expensive to

compute or estimate, since it is canceled out by examining the ratio of likelihoods for competing parameter sets:

$$\frac{P(\theta'|\mathbf{y})}{P(\theta|\mathbf{y})} = \frac{\mathcal{L}(\mathbf{y},\theta')\pi(\theta')}{\mathcal{L}(\mathbf{y},\theta)\pi(\theta)}. \tag{2.2.7}$$

MCMC algorithms operate for a fixed number of steps, some early portion of which are generally discarded as "burn-in" in order for the sampler to locate regions of high probability and begin exploring parameter space in an unbiased manner. The remaining samples are used to estimate the posterior distribution, and can be used to visualize parameter variation or calculate any relevant statistic thereof. There exist a wide array of MCMC techniques (see Bardenet (2013) for a more comprehensive review) which have been applied to problems in cardiac biology at scales ranging from full cell models down to single ion-channel kinetic models (Johnstone et al., 2016; Beattie et al., 2017).

There exist alternative strategies for Bayesian inference, such as sequential Monte-Carlo (SMC) algorithms, which maintain a population of simultaneously evolving chains in order to promote global exploration. We believe that SMC methods provide distinct benefits for the cardiac modeling community, and the biological and chemical modeling community at large (Daly et al., 2017a), and thus we will explore these samplers more fully in Chapter 3. While both MCMC and SMC methods are far more expensive to run than simple optimization techniques (due to the sheer number of samples required), they are better equipped to assess model identifiability than curvature-based metrics in certain situations, as we will detail in Chapter 4.

2.2.3 Approximate Bayesian inference

While Bayesian inference is attractive for the manner in which it reveals full distributions to characterize how model parameters are constrained by data, there are instances where the character of the data employed prevents the application of these techniques. When we consider the summary statistics of the cardiac action potential discussed in Section 2.1.4, we see that these quantities may be highly nonlinear in the underlying voltage measurements used to calculate them. Consider making an action potential recording, inevitably affected by measurement error,

and subsequently calculating summary statistics of this trace. The nonlinear functions relating voltage data to its summaries will affect how the error propagates, producing non-standard error distributions over summary statistics that are difficult to reason about (Daly et al., 2017b). Without a defined error model, we cannot calculate a likelihood function, and thus cannot employ Bayesian inference techniques. A similar inability to calculate likelihood can also prevent the application of Bayesian techniques to models with stochastic components.

In the absence of a calculable likelihood function, we may instead apply approximate Bayesian computation (ABC) methods, which use a distance function over model outputs as a substitute and produce an *approximation* of the "true" posterior (Csilléry et al., 2010; Marin et al., 2012). We will introduce the concept of ABC by means of a minimal example. Assume we are dealing with a model $f(\theta)$, data $\mathbf{y}$, and a distance function $\mathcal{D}(\mathbf{y}, f(\theta))$. We may approximate the posterior over model parameters θ given data $\mathbf{y}$ by finding all parameter sets for which the distance metric is less than some small amount ε. The simplest algorithm to perform such an inference is known as the ABC rejection sampler, and is described below:

Algorithm 2.1 ABC rejection sampler

```
Initialize an empty set S = ∅ of accepted parameters
while S contains fewer than n particles do
    Draw a sample θ* from a prior distribution π(θ)
    if D(y, f(θ*)) < ε then
        Add θ* to S
    end if
end while
return S
```

The result of this algorithm is a population of n parameter sets that produce simulations within ε of the observed data. Because we do not have enough information about the distribution of error on the model outputs, we assume that the distance from the observed data $\mathbf{y}$ is distributed uniformly over the range $[0, \varepsilon]$. Because this is rarely true in biology — parameters producing solutions more proximal to the observations are generally believed to be more likely than those producing solutions further away — this assumption results in an inflation of variance relative to

that of the true posterior. The manner in which this happens is highly problem-dependent, and there is very little theory that exists to explain this in general. This means that it is more difficult to draw hard conclusions about model identifiability from these methods, as the character of the posterior distributions produced is affected by factors other than model structure and data.

While the rejection sampler is appealing in its simplicity, it may require a large number of iterations to produce a posterior estimate if the prior distribution differs significantly from the posterior, due to the high rate of rejected samples. This has motivated development of MCMC- and SMC-based ABC samplers, which operate similarly to their Bayesian analogues and seek to improve performance using proposal distributions that are more likely to yield accepted parameters than a strategy of randomly sampling from the posterior (Sisson et al., 2007, 2009). ABC-SMC algorithms in particular have found application in areas of biological modeling (Toni et al., 2009; Sunnaker et al., 2013), and we will discuss the operation of these algorithms as well as their particular application to cardiac models in Chapter 3. A more complete review of approaches to efficient ABC sampling can be found in Marin et al. (2012).

Another area of focus for research in ABC techniques is "dimension reduction," or the selection of an optimal set of summary statistics to use during inference from a list of potential measurements. Because "sufficient" summary statistics — which represent the summarized data without loss of information — are rare outside of the simplest of models, modelers generally employ ensembles of summary statistics in the hope of capturing the most information possible. There is a great deal of interest in the automated construction of these sets, with algorithms weighing the benefits of including a new summary statistic, as measured by its effects on posterior entropy or maximum posterior likelihood, against potential negatives such as added computational cost or measurement expense (Fearnhead and Prangle, 2011; Blum et al., 2013; Prangle, 2015). Cardiac models employing AP summary statistics are unique in this regard, since modelers may sample from the "true" posterior distribution over model parameters by examining the underlying voltage trace. We discuss the implications of this knowledge for dimension reduction in Chapters 3 and 4.

2.3 Optimal Experimental Design

The desire to avoid unidentifiability when building and parameterizing models leads naturally to consideration of the design of experiments that are maximally informative about the parameters of a given model. Within biological and chemical modeling, optimal experimental design is generally concerned with determining the best times to observe the system and the best quantities to observe at those times, in order to minimize structural and practical unidentifiability in the model. One may additionally consider optimizing parameterized inputs to the system (such as voltage stimuli in recordings of cardiac cells) in order to drive behaviors that provide information on parameters of interest (Beattie et al., 2017). There are two main approaches to assessing model identifiability that naturally extend to the design of experiments. *A priori* methods generally focus on the spectral properties of the Fisher information matrix (FIM) which quantify the sensitivity of the model output to changes in system parameters. These are the same indices used in the calculation of classical confidence estimates, which will become very wide when the recorded model output is insensitive to parameter changes as a result of structural/practical identifiability. Because these indices are relatively inexpensive to compute, a family of methods to maximize FIM-based indices of identifiability have been proposed (Banks et al., 2011; Cintron-Arias et al., 2009). In Chapter 4, we will detail the calculation of the FIM as well as the operation of a prominent family of these experimental design algorithms.

When the observable model output is not differentiable with respect to the parameters of interest, as is the case when employing summary statistics like those described in Section 2.1.4, we cannot apply *a priori* methods based on the Fisher information matrix. We may instead employ *a posteriori* methods based on inference from experimental data, and use Bayesian or approximate Bayesian methods to draw conclusions about the identifiability of the model based on the size and character of the variation in the posterior distribution over the model parameters. The utility of inference techniques for assessing model identifiability is being increasingly recognized in the field of biochemical modeling, with many discussions and evaluations of com-

peting approaches being carried out (Chis et al., 2011; Kreutz et al., 2012; Miao et al., 2011; Raue et al., 2014; Villaverde and Banga, 2014). For example, Hines et al. (2014) demonstrate the use of posteriors produced by a Markov Chain Monte Carlo (MCMC) sampler to assess the identifiability of several competing models of reaction kinetics for a fixed set of data. In Chapter 4 we will detail how Bayesian and approximate Bayesian posteriors may be analyzed to determine and maximize model identifiability, and explore the experimental conditions under which these methods are preferable to FIM-based methods for biological and chemical models.

2.4 Summary

In this section, we have presented the necessary background information to interpret cardiac action potential models, including their mathematical structure, parameters of interest, and the biological interpretation of said parameters. We have also reviewed methods for parameterizing these models, as well as those for assessing their identifiability, and note that these methods may also be employed for general time-series models. In the subsequent chapters, we will explore the application of these modeling techniques to problems in cardiac biology in order to develop a set of methods for use in our Modeling Web Lab (which will be presented in Chapter 6), with the ultimate goal of promoting more rigorous validation of models by the community.

3

Posterior Inference Techniques

Now that we have covered necessary background information on cardiac models, which will be the flagship application domain for our Modeling Web Lab, we will demonstrate the application of posterior inference techniques to these models.

We will present two sequential Monte-Carlo sampling strategies that have been adapted to perform both Bayesian and approximate Bayesian inference, outline their benefits for applications in biology, and apply them to posterior inference on both the classical Hodgkin-Huxley neuronal action potential model and the modern O'Hara-Rudy cardiac action potential model. Through these analyses, we will demonstrate the importance of posterior inference for model validation. Additionally, we will demonstrate how the parallel application of Bayesian and approximate Bayesian techniques can aid in the design of informative experiments for models employing summary statistic data. The designs of the Modeling Web Lab (Chapter 6) and the underlying framework that will make this resource possible (Chapter 5) are inspired by these results, and a principle aim of our research is to encourage these types of analyses by modeling communities.

The work in Section 3.4 of this chapter has been published in Daly et al. (2015), and the work in Sections 3.2 and 3.5 has been published in Daly et al. (2017a).

3.1 Introduction

In building towards our online resource for modeling experiments, it is important to consider the classes of inference methods that should be supported in order to best serve the target community. While the cardiac modeling community, our initial area of application, has employed a wide range of optimization techniques to parameterize their models, we felt our resource would provide the greatest benefit to the community by focusing our initial efforts on posterior inference techniques. As we have mentioned in Chapter 2, these inference techniques have benefits over optimization techniques in that they provide estimates of parameter uncertainty, which are important for establishing faith in parametric models. Because cardiac models may be fit to either direct observation data or summary statistic data (which may preclude the application of Bayesian inference techniques), we feel that is is necessary for our resource to support both Bayesian and approximate Bayesian inference techniques in order to provide the greatest coverage for community usage cases. As an initial step towards the construction of our modeling resource, we decided to investigate the application of these inference techniques to cardiac models, as well as detailing the biological insight that the outputs of these methods can provide to those employing them.

For the purposes of this study, we have chosen to focus on Sequential Monte-Carlo (SMC) algorithms — a class of efficient sampling techniques that have been implemented in both Bayesian and approximate Bayesian contexts. Unlike MCMC methods, which draw samples from a single parameter distribution (the target posterior), SMC methods sample from a *sequence* of parameter distributions en route to a final target distribution. In a parameter inference context, this can be thought of as beginning from a prior distribution $\pi_0(\theta)$, proceeding through a series of artificial "intermediate distributions" $\pi_1(\theta), \ldots, \pi_{T-1}(\theta)$, and terminally sampling from the posterior distribution $\pi_T(\theta)$ using a process known as "importance sampling" to correct for differences between subsequent distributions (Neal, 2001). This has the effect of "smoothing" the difference between prior and posterior, and has been shown to have a beneficial effect

similar to that of simulated annealing approaches (Chopin, 2002). While these methods are generally more complex than MCMC approaches, often employing parallel MCMC chains to generate this series of distributions, SMC techniques are more resistant to being trapped in local minima for this very reason (Del Moral et al., 2006). Additionally, SMC algorithms allow for sequential Bayesian inference, which incorporates observations as the algorithm proceeds (instead of considering all observations at once as MCMC must do), which may save time when dealing with large data sets (Del Moral et al., 2006). For these reasons, SMC methods have recently been favored in both Bayesian and approximate Bayesian contexts (Chopin, 2002; Del Moral et al., 2006; Sisson et al., 2007, 2009; Toni et al., 2009; Del Moral et al., 2012).

Due to their generally differing spheres of use, little theoretical or practical literature exists comparing Bayesian and approximate Bayesian inference on a given class of models (although an example on a toy model may be seen in Sisson et al. (2007)). This might prove frustrating to biological modelers, who may wish to know how much uncertainty is introduced when fitting to summary statistic data, which requires the application of approximate Bayesian computation (ABC) methods, relative to fitting using time-course data with Bayesian methods. In order to enable comparative analyses within our modeling resource, we have taken two commonly employed ABC-SMC samplers — the "Toni" algorithm (Toni et al., 2009) and the "Del Moral" algorithm (Del Moral et al., 2012) — and generalized them to perform either exact or approximate Bayesian inference. This was done using a method originally proposed by Wilkinson (2013) for altering ABC rejection samplers and ABC-MCMC samplers, but extended by us here to the more powerful class of ABC-SMC samplers.

In this chapter, we will first present the generalized Toni and Del Moral algorithms, and discuss the adjustments that must be made in order to perform Bayesian or approximate Bayesian inference. We will validate the ability of these solvers to perform exact inference using a toy linear model, and examine their performance in high-dimensional parameter spaces using a polynomial model. We will then detail the application of the Toni algorithm to perform posterior inference on the classical Hodgkin-Huxley neuronal action potential model (Hodgkin and

Huxley, 1952) from original data reported by the authors, and investigate how adjustments in model complexity and the data employed in fitting affect posterior uncertainty. This will both serve as an affirmation of the quality of the work of the original authors, which was performed without the aid of modern computing technology, and will additionally demonstrate the utility of performing posterior inference and validating model behavior on a wide range of data. From there, we will proceed to the more modern and complex O'Hara-Rudy model of the cardiac action potential (O'Hara et al., 2011), which we will fit using both the exact Del Moral algorithm with time trace data and the approximate Del Moral algorithm with corresponding summary statistic data. By comparing the resulting posteriors, we can begin to assess the effects of employing summary statistic data on parameter uncertainty in cardiac action potential models — a problem that has yet to be addressed by the community. Through this study of the O'Hara-Rudy model, we will see the challenges that the complexity of modern action potential models bring to the application of posterior inference algorithms, as well as demonstrating a novel means of assessing the information content of summary statistics that can be enabled by comparative community modeling resources.

3.2 Sequential Monte-Carlo Inference

Before we present the two algorithms we will be considering in this section, we will discuss the concept of an acceptance kernel, which is the function that a Bayesian or approximate Bayesian algorithm uses to determine the probability of "accepting" a set of parameters, thereby adding them to the current estimate of the posterior distribution. It is through alteration of this acceptance kernel that we can interchange between the application of Bayesian and approximate Bayesian inference.

In the absence of a calculable likelihood function, many ABC methods employ the squared error distance metric $\mathcal{SE}(\mathbf{y}, \theta)$ to define the proximity between experimental data $\mathbf{y}$ and simulated data $f(\theta)$. Without any knowledge of the distribution of this error, ABC methods typically

assume the error to be uniformly distributed, and thus accept/reject points deterministically depending on whether they produce data within some range ε of the experimental observation. This yields the following binary acceptance kernel:

$$\rho_\varepsilon(\theta) = \mathbb{I}(\mathcal{SE}(\mathbf{y}, f(\theta)) < \varepsilon), \tag{3.2.1}$$

where $\mathbb{I}$ represents the indicator function, which evaluates to 1 when the enclosed condition holds and 0 otherwise. This kernel function is sometimes referred to as a "top-hat kernel" due to its shape.

As we have mentioned in the previous chapter, inference using such a top-hat kernel is exact only when model/system error is uniform over a sphere of radius ε. Because this is generally not the case for models of biological systems, employing a top-hat kernel may lead to an overestimation of posterior variance. Wilkinson (2013) suggests that by replacing the indicator function in Eq 3.2.1 with a probability density function (PDF) representing the true distribution of error, ABC rejection and ABC Markov Chain Monte Carlo samplers can be adjusted to perform exact inference.

If we are considering time course data $[\mathbf{y}_1, \ldots, \mathbf{y}_N]$ and our observation error is instead independently Gaussian with constant variance σ_e at each time point:

$$\mathbf{y}_i \sim \mathcal{N}(f(\bar{\theta}), \sigma_e^2), \quad i = 1, \ldots, N, \tag{3.2.2}$$

where $\bar{\theta}$ are the "true" system parameters, we should instead adopt the following smooth acceptance kernel to match that error distribution and allow for us to perform exact inference:

$$\rho_\varepsilon(\theta) \propto \exp\left(-\frac{SE(\mathbf{y}, f(\theta))}{2(\varepsilon\sigma_e)^2}\right)/c_\varepsilon, \tag{3.2.3}$$

where c_ε is a normalizing constant and ε is an adjustable "tolerance" controlling the width of the acceptance kernel. The simplest means of performing exact inference using such a kernel would be rejection sampling, the Bayesian analogue of the ABC rejection sampler (Algorithm 2.1) provided in the previous chapter:

Algorithm 3.1 Probabilistic rejection sampler

1: Initialize an empty set $\mathcal{S} = \emptyset$ of accepted parameters
2: **while** $\mathcal{S}$ contains fewer than n particles **do**
3: Draw a sample θ^* from a prior distribution $\pi(\theta)$
4: Add θ^* to $\mathcal{S}$ with probability $\rho_\varepsilon(\theta^*)$.
5: **end while**
6: **return** $\mathcal{S}$

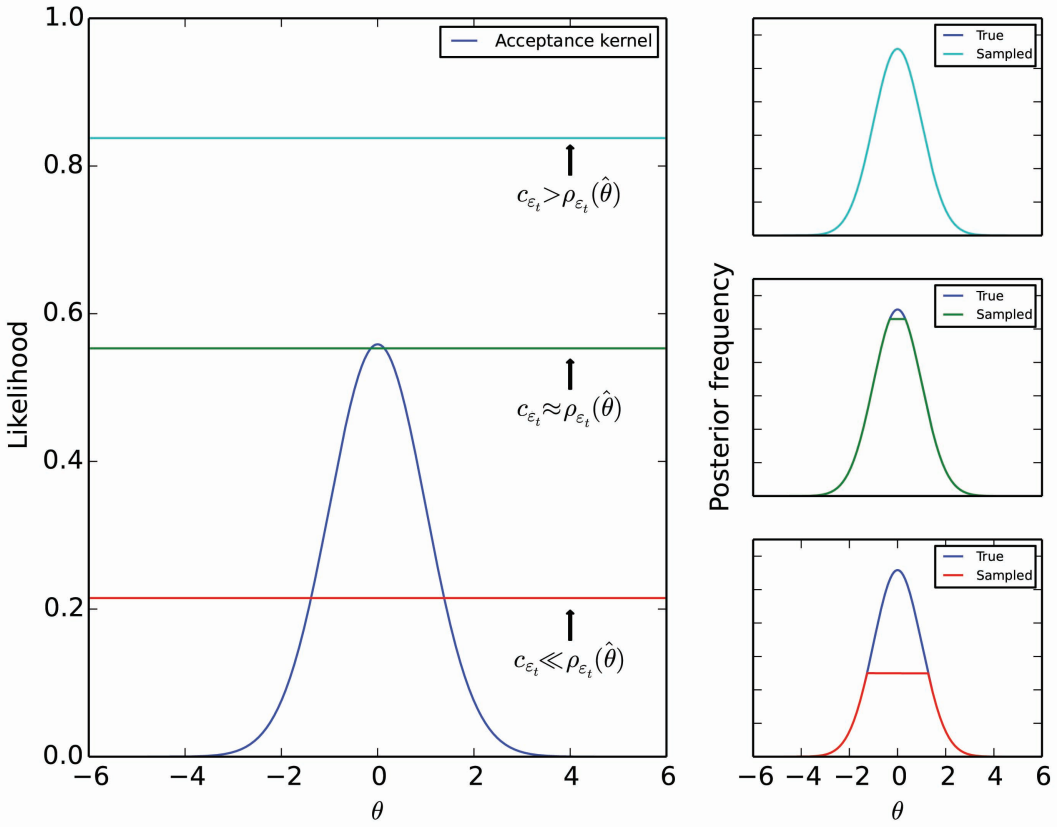
The efficiency of a rejection sampler employing the acceptance kernel in Equation 3.2.3 is highly dependent on the choice of c_ε. If the constant is set too high, the algorithm will require too many iterations, and if the constant is set too low, then the likelihood function will be "decapitated," exceeding a value of 1 over some range of parameter space and resulting in uniform sampling over this range regardless of the true distribution (see Figure3.1 for a graphical illustration).

ABC Sequential Monte Carlo samplers employ a sequence of decreasing thresholds $\varepsilon_0 > \varepsilon_1 > \ldots > \varepsilon_T$ to sample from a sequence of intermediate distributions, smoothing the difference between the prior and posterior. Adopting the probabilistic acceptance kernel in Eq 3.2.3 will allow us to adjust these algorithms to perform exact inference as the iteration proceeds, as $\rho_\varepsilon(\theta) \to \mathcal{N}(f(\theta), \sigma_e^2)$ when $\varepsilon \to 1$. In this case, the tolerance parameter ε can be analogized to the "temperature" parameter in simulated annealing. Thus, the adjusted algorithm similarly "heats" the likelihood to an initial value ε_0, at which point its surface is highly smooth, then slowly "cools" the system towards the true posterior through a sequence of intermediate distributions defined by $[\varepsilon_1, \ldots, \varepsilon_T]$.

3.2.1 The generalized Toni algorithm

The first sampler we consider is based on the one proposed by Toni et al. (Toni et al., 2009). This sampler maintains a constant number of particles n that, at each intermediate distribution, are updated by rejection sampling (along with local perturbation according to a fixed proposal distribution) from the previous posterior estimate, but with a lower error threshold. This is

Figure 3.1 Impact of the magnitude of the normalizing constant c_{ε_t} of Equation 3.2.3 on posteriors produced by a rejection sampler. The left-hand plot demonstrates a sample (Gaussian) acceptance kernel function with mode $\hat{\theta}$, with three potential values for the normalizing constant shown relative to an empirical estimate of the modal likelihood $\rho_{\varepsilon_t}(\hat{\theta})$. Right-hand plots show the true posterior (blue) along with empirical posteriors that would result from employing a sampler with an acceptance kernel normalized to a value greater than (top), slightly less than (middle), or much less than (bottom) its true modal value.



represented schematically in Figure3.2.

We propose a generalized version of the Toni sampler in Algorithm 3.2 that employs an arbitrary acceptance kernel $\rho_\varepsilon(\theta)$. Let $\pi(\theta)$ represent the prior distribution on parameters θ. We will draw a series of intermediate distributions each composed of n particles $\{\theta_t^{(1)}, \ldots, \theta_t^{(n)}\}$ estimating the posterior, where the subscript t denotes the index of the intermediate distribution and the superscript indicates the index of the particle *within said estimate*. We define a (series of) proposal distribution(s) $K_t(\theta'|\theta)$ which describe the probability of a particle moving from θ to θ' in parameter space between iterations (we have been using a Gaussian proposal distribution of fixed width, chosen to be 10% of the variance of the prior for each parameter, for all values of t). Finally, we define weights for these particles, $\{w_t^{(i)}\}$, that are updated after each iteration according to the following scheme:

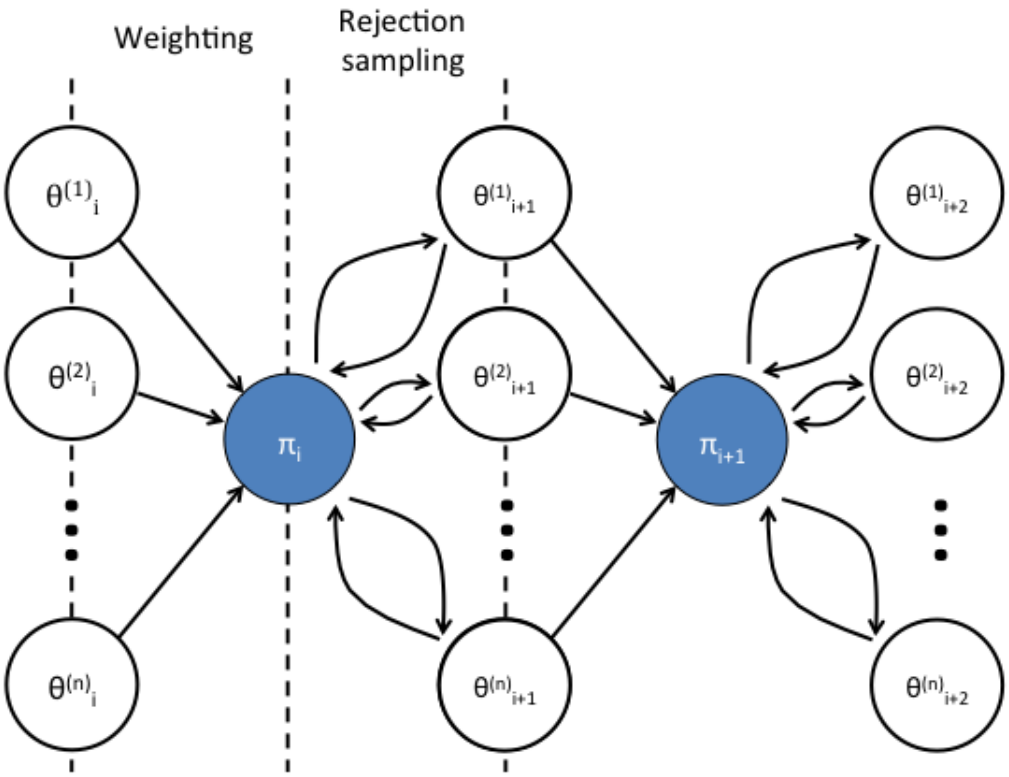
$$w_t^{(i)} \propto \frac{\pi(\theta_t^{(i)})}{\sum_{j=1}^{n} w_{t-1}^{(j)} K_t(\theta_t^{(i)}|\theta_{t-1}^{(j)})}. \tag{3.2.4}$$

These particle weights $\{w_t^{(i)}\}$ are calculated according to a scheme proposed by (Del Moral et al., 2006) that minimizes their variance with respect to those of the previous estimate $\{w_{t-1}^{(i)}\}$.

The α parameter in Algorithm 3.2 can be thought of as the parameter defining the "cooling schedule" in simulated annealing, as it controls the percent reduction of ε_t at each iteration. If the posterior estimate update does not succeed within a given number of iterations ($I_{max} = 5 \times 10^6$), we re-attempt with a more conservative reduction of ε_t for the next iteration. This adaptively adjustment of the error threshold allows the algorithm to be applied to a wide range of problems with little manual input.

A major advantage of this scheme is its amenability to parallelization. Because rejection sampling (lines 7-14) is performed independently for each particle, sampling can be divided between different threads/processes. The calculation of weights (lines 20-22) can also be parallelized once rejection sampling for the associated particles has completed. In our Python implementation, this parallelization was accomplished through the use of the Pathos multipro-

Figure 3.2 Schematic representation of the Toni SMC sampler. "Weighting" encompasses lines 20-22 of algorithm 3.2, where importance weights are assigned to each particle to allow for sampling the next estimate. "Rejection sampling" encompasses lines 7-14, where samples are drawn from the previous estimate, perturbed according to the proposal distribution, then accepted or rejected (and subsequently re-sampled) according to the kernel function.



Algorithm 3.2 The generalized Toni SMC sampler

Parameters: n: number of particles, k_{min}, I_{max}: termination conditions, α: cooling schedule, ε_0: initial error threshold

1: Set $t = 0$, $k_0 = \varepsilon_0$ $\{t$ is the iteration number, k_t controls reduction in error threshold for this iteration.$\}$
2: **while** $k_t > k_{min}$ **do**
3: **if** $t = 0$ **then**
4: Using Algorithm 3.1 with tolerance ε_0, draw n particles $\{\theta_0^{(i)}\}$
5: Set all $w_0^{(i)} = 1/n$
6: **else**
7: **for** $i = 1$ **to** n **do**
8: Draw θ^* from $\{\theta_{t-1}^{(i)}\}$ (with weights $\{w_{t-1}^{(i)}\}$)
9: Draw θ^{**} from the proposal distribution $K_t(\theta|\theta^*)$
10: **if** $\pi(\theta^{**}) = 0$ **then**
11: Repeat from line 8
12: **end if**
13: Set $\theta_t^i = \theta^{**}$ with probability $\rho_{\varepsilon_t}(\theta^{**})$ (roll a die). Otherwise, repeat from line 8.
14: **end for**
15: **if** Lines 7-14 did not succeed in populating $\{\theta_t^{(i)}\}$ within I_{max} draws from the previous posterior estimate **then**
16: $k_t \leftarrow \alpha k_t$
17: $\varepsilon_t \leftarrow \varepsilon_{t-1} - k_t$
18: Repeat from line 7
19: **else**
20: **for** $i = 1$ **to** n **do**
21: Set $w_t^{(i)} = \pi(\theta_t^{(i)}) / \sum_{j=1}^{n}(w_{t-1}^{(j)} K_t(\theta_t^{(i)}|\theta_{t-1}^{(j)}))$
22: **end for**
23: **end if**
24: **end if**
25: Normalize $\{w_t^{(i)}\}$
26: $k_{t+1} = \min(k_t, \alpha\varepsilon_t)$
27: $\varepsilon_{t+1} = \varepsilon_t - k_{t+1}$
28: $t \leftarrow t + 1$
29: **end while**
30: **return** $(\{\theta_t^{(i)}, w_t^{(i)}\})$

cessing module (McKerns and Aivazis, 2010; McKerns et al., 2011).

3.2.2 The generalized Del Moral algorithm

A potential drawback of the Toni algorithm is the complexity of its weighting scheme. We note that this weighting scheme (Eq 3.2.4) is asymptotically linear in the number of particles n for each particle, making the cost of updating the weights of all particles quadratic in n. While the cost of simulation is usually assumed to dominate the runtime of the sampler, this expensive weighting step may add noticeable overhead for large numbers of particles regardless of the model chosen.

This computational complexity in part motivated Del Moral et al. (2012) to propose an alternative ABC-SMC implementation based on Metropolis-Hastings updates, rather than rejection sampling, that allowed them to employ an approximation of the weighting scheme in Eq 3.2.4:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{\mathbb{I}(SE(\theta_{t-1}^{(i)}) < \varepsilon_t)}{\mathbb{I}(SE(\theta_{t-1}^{(i)}) < \varepsilon_{t-1})}, \tag{3.2.5}$$

which, when adapted to perform probabilistic inference, becomes:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{\rho_{\varepsilon_t}(\theta_{t-1}^{(i)})}{\rho_{\varepsilon_{t-1}}(\theta_{t-1}^{(i)})}. \tag{3.2.6}$$

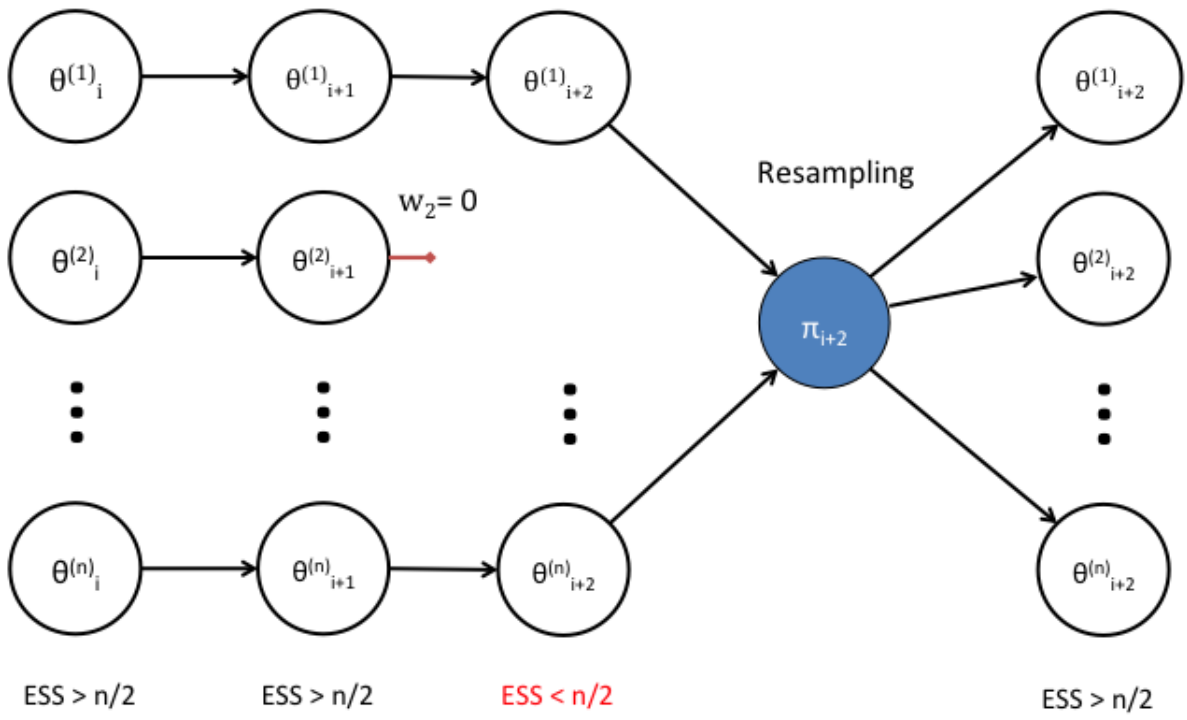
This scheme boasts constant-time calculation of particle weights, leading to only linear cost for a single distribution update. While this approximation overestimates the variance of the incremental weights relative to the scheme in Eq 3.2.4, Del Moral et al. (2012) believe the approximation should be sufficiently close as long as sequential posterior estimates are sufficiently close (i.e., $\pi_t(\theta) \approx \pi_{t-1}(\theta)$).

The Del Moral algorithm effectively maintains a set of n independent Markov chains that are periodically resampled when the effective sample size (ESS):

$$ESS(\{w_t^{(i)}\}) = 1/\sum_{i=1}^{n}((w_t^{(i)})^2) \tag{3.2.7}$$

falls below some threshold (generally taken to be $n/2$), indicating a significant drop in entropy. This implementation is represented schematically in Figure 3.3, and a generalized version employing an arbitrary acceptance kernel $\rho_\varepsilon(\theta)$ is described in Algorithm 3.3.

Figure 3.3 Schematic representation of the Del Moral SMC sampler. Between rounds, particles with non-zero weights evolve according to independent Markov Chains (lines 22-25, Algorithm 3.3) and weight calculation (lines 7-9) can happen concurrently. If effective sample size (ESS) falls below a critical fraction, "Resampling" occurs to repopulate the posterior estimate with uniformly weighted draws from the previous estimate (lines 16-21).



Algorithm 3.3 The generalized Del Moral SMC sampler

Parameters: n: number of particles, k_{min}: termination condition, α: cooling schedule, ε_0: initial error threshold

1: Set $t = 0$, $k_0 = \varepsilon_0$ $\{t$ is the iteration number, k_t controls reduction in error threshold for this iteration.$\}$
2: **while** $k_t > k_{min}$ **do**
3: **if** $t = 0$ **then**
4: Using Algorithm 3.1 with tolerance ε_0, draw n particles $\{\theta_0^{(i)}\}$
5: Set all $w_0^{(i)} = 1/n$
6: **else**
7: **for** $i = 1$ **to** n **do**
8: Set $w_t^{(i)} = w_{t-1}^{(i)} \rho_{\varepsilon_t}(\theta_t^{(i)}) / \rho_{\varepsilon_{t-1}}(\theta_t^{(i)})$.
9: **end for**
10: **if** all weights $\{w_t^{(i)}\}$ are 0 **then**
11: $k_t \leftarrow \alpha k_t$
12: $\varepsilon_t \leftarrow \varepsilon_{t-1} - k_t$
13: Repeat from line 7
14: **end if**
15: Normalize $\{w_t^{(i)}\}$
16: **if** $ESS(\{w_t^{(i)}\}) < n/2$ **then**
17: Resample n particles $\{\theta^{(i)*}\}$ from $\{\theta_{t-1}^{(i)}\}$ using the new weights $\{w_t^{(i)}\}$.
18: **for** $i = 1$ **to** n **do**
19: Set $\theta_{t-1}^{(i)} = \theta^{(i)*}$ and $w_t^{(i)} = 1/n$.
20: **end for**
21: **end if**
22: **for** $i = 1$ **to** n **do**
23: Draw θ^* from the proposal distribution $K_t(\theta | \theta_{t-1}^{(i)})$
24: Set $\theta_t^i = \theta^*$ with probability $\min\left(\frac{\rho_{\varepsilon_t}(\theta^*) K(\theta_{t-1}^{(i)} | \theta^*) \pi(\theta^*)}{\rho_{\varepsilon_t}(\theta_{t-1}^{(i)}) K(\theta^* | \theta_{t-1}^{(i)}) \pi(\theta_{t-1}^{(i)})}, 1 \right)$ (roll a die). Otherwise, set $\theta_t^{(i)} = \theta_{t-1}^{(i)}$.
25: **end for**
26: **end if**
27: $k_{t+1} = \min(k_t, \alpha \varepsilon_t)$
28: $\varepsilon_{t+1} = \varepsilon_t - k_{t+1}$
29: $t \leftarrow t + 1$
30: **end while**
31: **return** $(\{\theta_t^{(i)}, w_t^{(i)}\})$

The adaptive ε_t schedule (controlled by a cooling parameter α identically to our implementation of the Toni algorithm in Algorithm 3.2) remains the same as in the Toni implementation, but as this sampling scheme performs a fixed number of Metropolis-Hastings steps for each intermediate distribution update, there is no need to set a maximum number of iterations I_{max} for the posterior update step. Instead, since the weighting scheme is effectively counting the number of particles that would "survive" the ε_t update, the conditional on line 10 determines whether this update would result in all particles being rejected by the kernel ρ_{ε_t}. One might replace this check with a more general condition, where the update fails if the ESS drops below some critical threshold, which we expect would increase resistance of the posterior estimates to underdispersion resulting from an overly aggressive choice of cooling schedule.

As with the Toni algorithm, we have generally been using a Gaussian proposal distribution of fixed width, chosen to be 10% of the variance of the prior for each parameter. However, we have found that adaptively adjusting the width of the kernel (scaling the covariance matrix by a constant) based on the acceptance rate of draws, as is done for many MCMC samplers, has helped our Del Moral implementation. We use an update scheme, detailed in Algorithm 3.4, adopted from the PyMC (Patil et al., 2010) implementation of tuned MCMC that increases/decreases proposal variance proportionally to deviation of the acceptance rate from the 20%-50% range at fixed intervals (every 100 proposals).

3.3 Validation and Benchmarking of Samplers

Before employing the generalized Toni and Del Moral samplers for inference on action potential models, we sought to verify that the adjusted samplers were capable of performing exact inference, and to investigate the effects of the various algorithmic parameters on their performance. To do this, we examined the ability of the Toni and Del Moral samplers to recover the parameters of toy linear and polynomial models, the results of which we will detail in the following subsections.

Algorithm 3.4 Periodic update applied to the width of the proposal distribution K_t used by the the Del Moral algorithm.

$K_t \sim \mathcal{N}(0, s\Sigma)$
Let a be the fraction of accepted steps over the last 100 proposals
if $a < 0.001$ **then**
    $s \leftarrow 0.001s$
else if $a < 0.05$ **then**
    $s \leftarrow 0.5s$
else if $a < 0.2$ **then**
    $s \leftarrow 0.9s$
else if $a > 0.95$ **then**
    $s \leftarrow 10s$
else if $a > 0.75$ **then**
    $s \leftarrow 2s$
else if $a > 0.5$ **then**
    $s \leftarrow 1.1s$
end if

3.3.1 SMC Performance on a Linear Model

In order to verify the ability of the solvers to perform exact inference, we chose to examine the following linear model:

$$f(x, a) = ax, \quad y_i \sim \mathcal{N}(f(x_i, a), \sigma_e^2), \tag{3.3.1}$$

which has a known Gaussian posterior over the slope parameter a when a conjugate Gaussian prior $\pi(a)$ is assumed over the slope parameter:

$$\pi(a) \sim \mathcal{N}(0, \sigma_a^2) \rightarrow P(a|\mathbf{y}) \sim \mathcal{N}(\frac{\mathbf{x}^\intercal \mathbf{y}}{\mathbf{x}^\intercal \mathbf{x} + \sigma_e^2 \sigma_a^{-2}}, \frac{\sigma_e^2}{\mathbf{x}^\intercal \mathbf{x} + \sigma_e^2 \sigma_a^{-2}}). \tag{3.3.2}$$

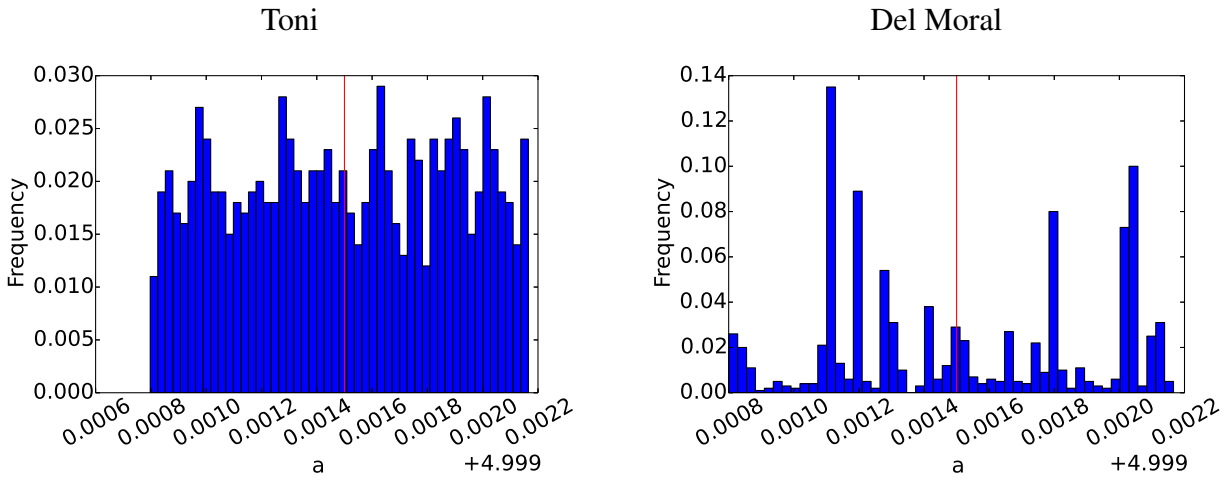
For each fitting experiment, experimental data were simulated as a single vector of n noisy observations $\mathbf{y}$ taken at fixed time points $\mathbf{x}$. n=301 time points were chosen, symmetrically about 0: $\mathbf{x} = \{-150, -149, ..., 0, ...149, 150\}$. The true value of a was chosen as $\bar{a} = 5$, while the width of the prior σ_a was set to 10, and the magnitude of Gaussian noise σ_e was set to 1. A Gaussian proposal distribution $K(\theta'|\theta) \sim \mathcal{N}(\theta, 1)$ was chosen for both samplers during all inferences. The width of this distribution was periodically updated in the Del Moral scheme as described in Algorithm 3.4 in order to heuristically attain a 20-50% acceptance rate of Metropolis-Hastings steps.

We began with a comparison of the ABC inference capabilities of both samplers on the model, performing posterior inference with the Toni and Del Moral samplers using the standard top hat kernel (Equation 3.2.1), which does not match the true (Gaussian) distribution of observation error (Equation 3.3.1). We first compared the two samplers with a common number of particles (1000) and $\alpha = 0.5$, aiming to halve the maximum error among particles in the population after each update. The initial tolerance ε_0 for both samplers was set to $\max_i(SE(\theta_0^i))$, the maximum error among n samples drawn from the prior. Both algorithms were set to terminate when $\varepsilon_t < 301$, the sample size of the data. This second criterion was introduced because the expected value of $SE(\theta)$ at the true solution, $\bar{a}$, is equal to the number of observations n, which thus sets a reasonable estimate for the final error ε_T and would ensure timely termination. In the presence of noisy observations, the acceptance probability of the top-hat kernel (Equation 3.2.1) approaches 0 for all θ as $\varepsilon_t \to 0$ when a finite number of observations are employed, and thus a positive limit is expected on ε_T.

The results of approximate inference on the linear model under the above settings are shown in Figure 3.4. We see that while both posteriors capture the theoretical mean (which differs from the generating value of $a = 5$ by less than 0.01%) and span the same range of values, the posterior produced by the Del Moral sampler shows significantly higher variance than the one produced by the Toni sampler, which shows the uniformity expected when employing a top-hat kernel.

We sought to investigate the effects of increased population size and relaxed cooling schedule on the uniformity of the Del Moral posterior estimates for the linear model. Both of these changes would increase the similarity between subsequent posterior estimates, and thus would be expected to improve the approximation of the true importance sampling weights employed by the Del Moral algorithm (Del Moral et al., 2012). In Figure 3.5 we see that either by increasing the number of particles from 1,000 to 10,000 or by decreasing the cooling schedule α below 0.3, the variance in the resulting Del Moral posterior estimate decreases dramatically and it approaches the uniformity of the Toni posterior. While the variance in the final posterior

Figure 3.4 Posteriors generated by the Toni and Del Moral samplers over the linear model when using a top hat acceptance kernel. Vertical bars represent frequencies of binned values of a among particles in the posterior estimate, while the vertical red line represents the theoretical posterior mean calculated with Eq 3.3.2.

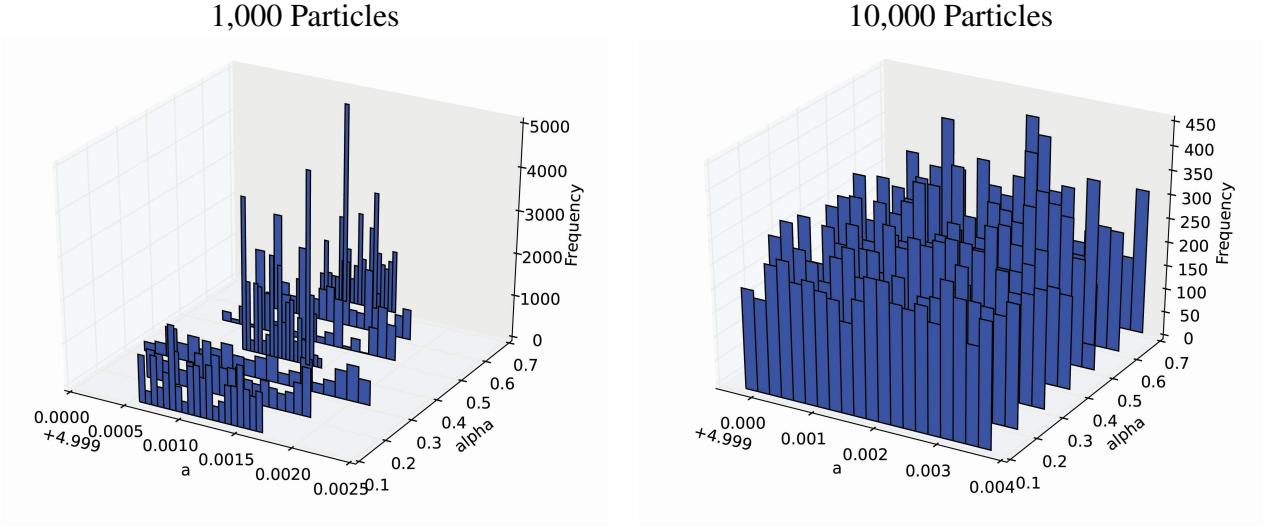


decreases proportionally to α for both 1,000 and 10,000 particle samplers, the effect is much more pronounced at the lower population size.

We next ran both ABC-SMC variants on the same linear model, but employing the probabilistic acceptance kernel that matches the observational error distribution (Eq 3.2.3). When employing a Gaussian kernel (Eq 3.2.3), the initial tolerance ε_0 for both samplers was set to 1,000, as it gave a reasonable acceptance rate for the first round of rejection sampling. Both algorithms were set to terminate when $k < k_{min} = 0.001$, or when $\varepsilon_t = 1$, as in the probabilistic kernel (Equation 3.2.3) the interpretation of ε is a multiplier of the true standard deviation of the error model, and thus decreasing below 1 would cause underdispersion in the final estimate. Informed by the results of the approximate study of the linear model, the cooling schedule parameter α was decreased to 0.25 for both samplers and the number of particles employed by the Del Moral sampler was increased to 10,000 to improve the quality of the weighting approximation. The population size of the Toni sampler was maintained at 1,000 due to its reasonable performance under these settings when performing approximate inference.

Applying the probabilistic Del Moral sampler, we see in Figure 3.6 that the resulting posterior estimate shows a high degree of homology with the theoretical posterior. The histogram of

Figure 3.5 Effect of population size and α in the Del Moral sampler on posteriors produced over the linear model. Posteriors produced using 1000 (left) and 10,000 (right) particles at varying values of the algorithmic parameter α. Histograms parallel to the "a" axis represent the frequencies of particles in the final posterior estimate generated under the corresponding value of α on the "alpha" axis.

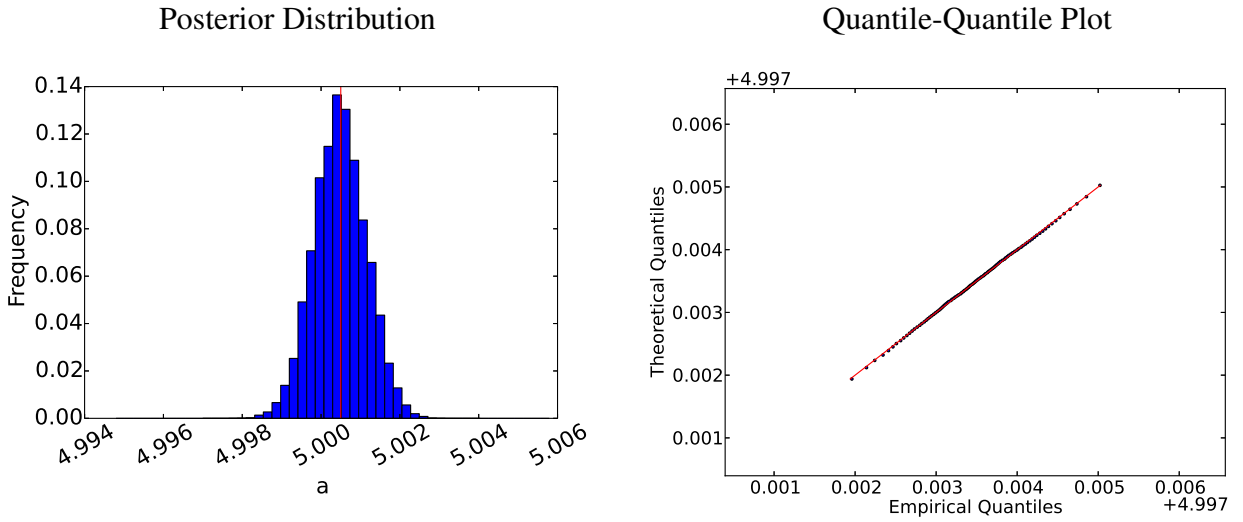


values of the slope parameter a appears normally distributed, and captures the theoretical mean almost exactly. Additionally, the quantile-quantile plot, which plots the percent point function (PPF) of the theoretical posterior distribution (Equation 3.3.2) against that of the empirical posterior, shows a linear relationship with slope 1, indicating that the two distributions are Gaussian with equal variance.

In order to similarly employ the probabilistic Toni sampler, however, we must additionally consider the value of the normalizing constant c_{ε_t} in Equation 3.2.3. In the Del Moral sampler, the constant is canceled by the Metropolis-Hastings update, but in rejection-based samplers like Algorithm 3.2, it controls the tradeoff between a reasonable acceptance rate (line 13) and correctness of the sampler. Wilkinson (2013) suggests choosing for c_{ε_t} the modal value of the acceptance kernel, where $SE(\theta) = 0$, which would improve the overall acceptance rate but still bound the function between 0 and 1. In the presence of noise, however, no single parameter set will generate such error-free predictions, so we instead employed the empirical mode $\rho_{\varepsilon_t}(\hat{\theta})$, where $\hat{\theta}$ is the maximum likelihood estimate of the parameters given the noisy data.

In Figure 3.7 we see the sensitivity of the Toni sampler to three values for c_{ε_t} chosen rel-

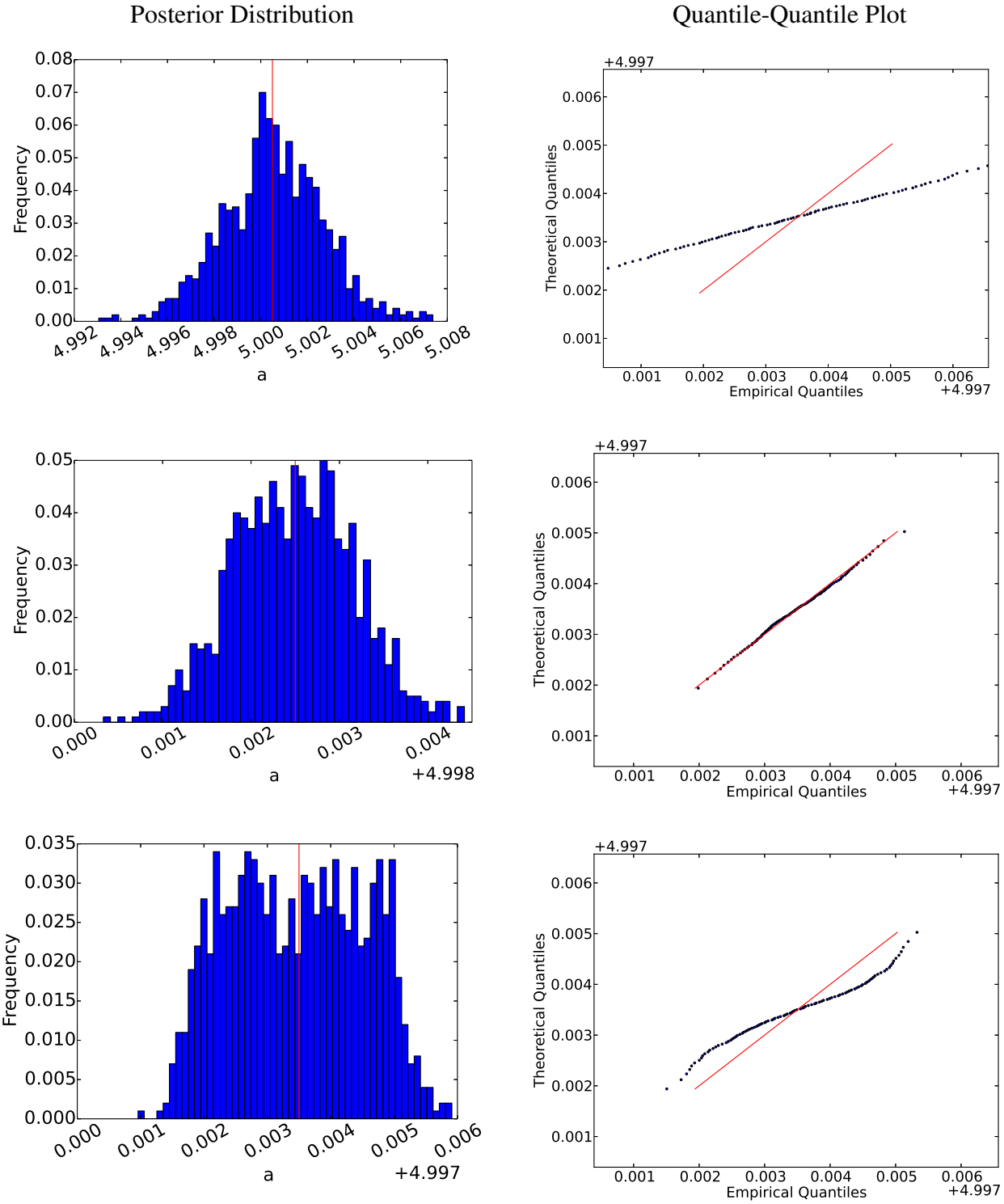
Figure 3.6 Posterior generated by the Del Moral sampler over the linear model when using a Gaussian acceptance kernel. (Left) Histogram of particle frequencies, with red vertical line indicating theoretical mean. (Right) Quantile-quantile plot comparing the PPF (percent point function) of the estimated posterior (x-axis) to that of the theoretical posterior (y-axis) by paired estimation at 100 equally spaced quantiles.



ative to this empirical mode. When $c_{\varepsilon_t} = 0.1\rho_{\varepsilon_t}(\hat{\theta})$, deliberately below the modal value, we see the posterior take on a nearly uniform character around the mean, with a high nonlinearity in the quantile-quantile relationship confirming disagreement with the theoretical Gaussian posterior. Conversely, when $c_{\varepsilon_t} = 10\rho_{\varepsilon_t}(\hat{\theta})$, the algorithm terminates early with final $\varepsilon_T > 3$ due to repeatedly exceeding the I_{max} criterion (line 15 of Algorithm 3.2), and the marginal and quantile-quantile plots reveal the final posterior estimate to be overdispersed relative to the theoretical posterior in Equation 3.3.2. Finally, when $c_{\varepsilon_t} = \rho_{\varepsilon_t}(\hat{\theta})$, we see overall agreement between the empirical posterior and the theoretical one, despite some leveling off near the mode. In both cases of $c_{\varepsilon_t} = 0.1\rho_{\varepsilon_t}(\hat{\theta})$ and $c_{\varepsilon_t} = \rho_{\varepsilon_t}(\hat{\theta})$ the value of the acceptance kernel was observed to take on values greater than 1, leading to inexact sampling due to "decapitation" of the likelihood function (see Figure 3.1). This occurred with far less frequency when $c_{\varepsilon_t} = \rho_{\varepsilon_t}(\hat{\theta})$, and only at later iterations.

As a result of the sensitivity of the Toni algorithm to this parameter of the probabilistic acceptance kernel, and the difficulty involved in estimating $\rho_{\varepsilon_t}(\hat{\theta})$, we believe the Del Moral algorithm should be favored for applications in probabilistic inference.

Figure 3.7 Effect of c_{ε_t} on posteriors generated by the Toni algorithm over the linear model with a Gaussian acceptance kernel. (Top) $c_{\varepsilon_t} = 10\rho_{\varepsilon_t}(\hat{\theta})$, (Middle) $c_{\varepsilon_t} = \rho_{\varepsilon_t}(\hat{\theta})$, (Bottom) $c_{\varepsilon_t} = 0.1\rho_{\varepsilon_t}(\hat{\theta})$, where $\hat{\theta}$ is the maximum likelihood estimate of the slope parameter. (Left) Histogram of particle frequencies, with red vertical line indicating theoretical mean. (Right) quantile-quantile plot comparing the PPF of the estimated posterior (x-axis) to that of the theoretical posterior (y-axis) at 100 equally spaced quantiles.



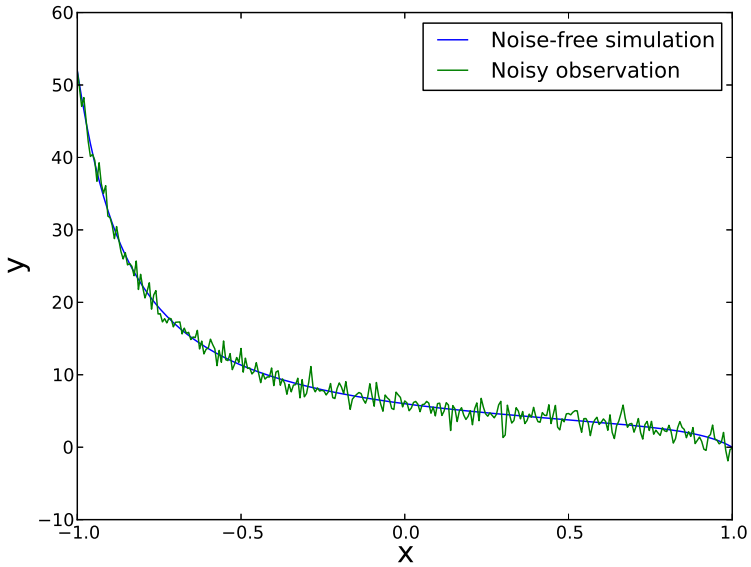
3.3.2 SMC Performance on a Polynomial Model

After comparing the performance of the two samplers on the simple linear model, we sought to investigate the efficacy of the two samplers in larger parameter spaces that more closely approximate those of modern action potential models. For this investigation, we chose the following polynomial model:

$$f(x) = \sum_{i=1}^{P} c_{i-1} x^{i-1}, \quad \mathbf{y}_t \sim \mathcal{N}(f(\mathbf{x}_t), \sigma_e^2), \quad t = \{0, ..., T\}, \tag{3.3.3}$$

which has P free parameters. We chose $P = 16$ to make the dimension of parameter space similar to that of the O'Hara-Rudy action potential model, and chose "true" parameters $c_0 = 6, c_1 = -6, c_2 = 5, c_3 = -5, c_4 = 4, c_5 = -4, c_6 = 3, c_7 = -3, c_8 = 2, c_9 = -2, c_{10} = 1, c_{11} = -1, c_{12} = 2, c_{13} = -2, c_{14} = 3, c_{15} = -3$. We generated data using 301 uniformly spaced points between -1 and +1 for $\mathbf{x}$, and set the magnitude of the experimental noise $\sigma_e = 1$. Output from this model is depicted graphically in Figure 3.8.

Figure 3.8 Model output from the 16-parameter polynomial model. The output of the model is shown in blue, alongside the data after the addition of noise (green) that was used for approximate Bayesian inference.



In order to assess the effects of increasing the dimension of parameter space, we performed

inference on the above model using both Toni and Del Moral samplers while varying the number of free parameters and holding all others to their true values. For a number of free parameters ranging from 3 to 16, we performed posterior inference using both the Toni and Del Moral sampler, and recorded properties of the intermediate distributions produced by each sampler. While there is a known error model for the data above, we chose to employ the approximate versions of both samplers, using the top-hat acceptance kernel (Equation 3.2.1) over the full trace, in order to investigate the operation of each sampler while avoiding the issue of choosing a normalizing constant for the acceptance kernel in the Toni sampler (see the previous section).

In each inference, independent uniform priors over the range (-10,10) were assigned to each free variable to remove any effects of the prior PDF from particle weighting in the SMC algorithms. An independent Gaussian proposal distribution $K(\theta'|\theta) \sim \mathcal{N}(\theta, 2)$ was set over each model parameter for both samplers. The width of this distribution was periodically updated in the Del Moral scheme as described in Algorithm 3.4. For both samplers, the number of particles was set to 1,000 and the cooling schedule α was set to 0.2. The initial tolerance ε_0 was set to the maximum error of the initial posterior estimate, and the algorithms were set to terminate when $k < k_{min} = 1$ or when $\varepsilon_t < 301$, the sample size of the data.

For each sampler, we investigated the variance of the weights assigned to each particle in successive intermediate distributions. Low variance in this quantity indicates that the distribution has high entropy, which promotes exploration of parameter space in the subsequent iteration. Conversely, high variance in this quantity indicates that a small number of particles have been severely upweighted, and exploration of parameter space will be severely biased in these areas. As we see in Figure 3.9, both samplers show an initial increase in posterior weight variance from the uniform prior across all values of P. In the Del Moral sampler, we see that regardless of the value of P, the sampler maintains a ceiling on weight variance as the importance sampling proceeds, thanks to resampling steps which redistribute weight across the posterior (indicated by sudden drops in weight variance). In the Toni sampler, we see that at $P \leq 6$, the variance of posterior weights largely decreases as the iterations proceed, yet at large values of

P the variance swings wildly between iterations, with effects becoming more pronounced at higher values of P. Upon termination, the posterior returned by the Toni sampler when $P = 16$ contained a single particle whose weight accounted for over 34% of the distribution.

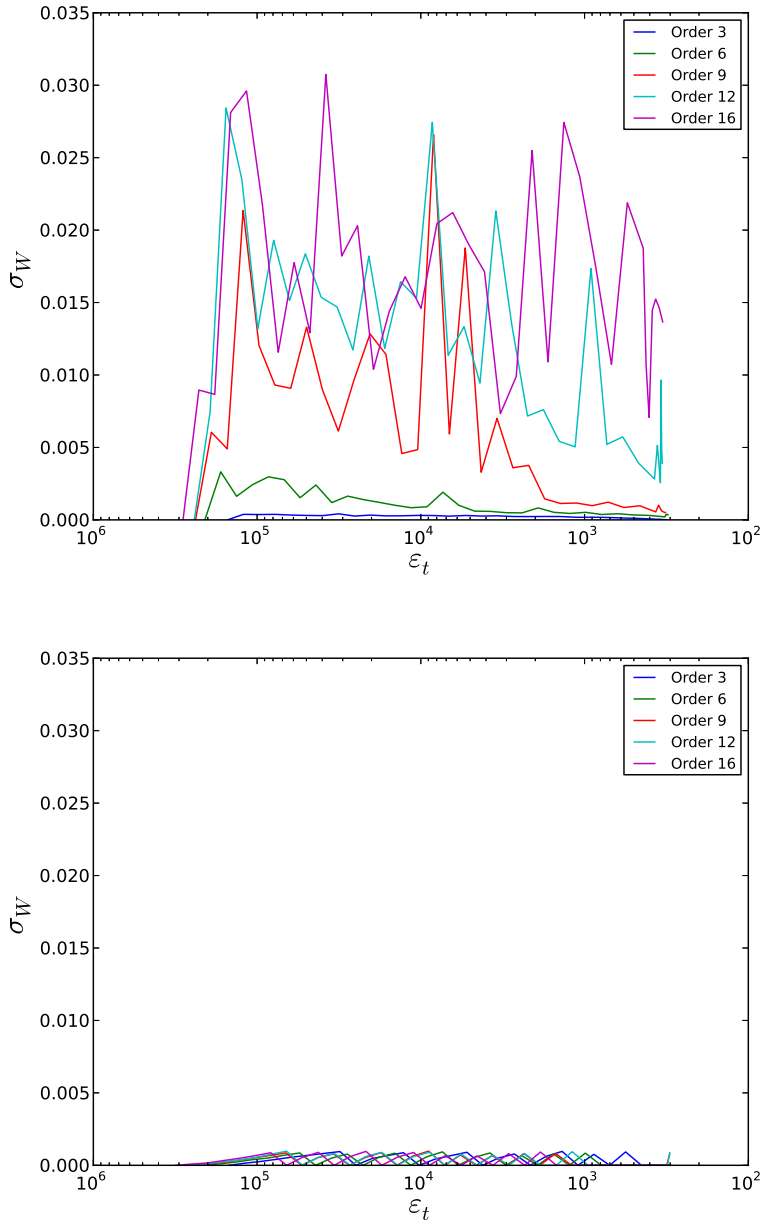
While both samplers are expected to struggle as the number of parameters approaches the number of particles employed, due to the difficulty of fully exploring parameter space, we have demonstrated the increased tendency of the Toni sampler (relative to the Del Moral sampler) to "collapse" to a single point in parameter space under conditions similar to those seen in modern action potential models. This may result from the denominator of the Toni weighting scheme (Eq 3.2.4) having high variance due to poor coverage of the proposal distribution $K(\theta|\theta^*)$ by the posterior particles. As such, even with a uniform prior, certain particles may become severely upweighted between iterations and the posterior estimate may collapse to one or several points. The Del Moral weighting scheme (Eq 3.2.5) is immune to this effect, and the resampling step in the algorithm itself (Algorithm 3.3, lines 16-21) effectively caps the variance that the particle weights may attain. Beyond simply increasing the number of particles, which comes at computational cost in the weighting scheme, the performance of the Toni sampler might be improved by employing an adaptive proposal distribution that can redistribute weight on distributions that become too peaked, or by explicit resampling as in the Del Moral scheme, but under current formulation the Del Moral sampler is preferable for high dimensional parameter spaces.

3.4 The Hodgkin-Huxley Neuronal AP Model

In order to demonstrate the application of ABC techniques to action potential models, we will begin with an examination of the original Hodgkin-Huxley neuronal AP model[1]. We chose to examine this model in particular for its simplicity as an initial test case, its seminal importance, and the fact that the authors fit the model entirely to personally collected data, much of which was made available in the original publication (Hodgkin and Huxley, 1952). As we have men-

[1]While we recognize that exact inference methods may be more appropriate for this deterministic model when time course data are available, we chose to employ ABC methods for this study as part of an exploration of the techniques.

Figure 3.9 Effect of model dimension on stability of Toni and Del Moral samplers. Variance of particles weights (σ_w) in the Toni sampler (top) and the Del Moral sampler (bottom), each with 1,000 particles, as they iterate over posterior estimates of the polynomial model with increasing numbers of parameters (indicated by their "order"). Iteration of the algorithm is marked on the x-axis (log-scale) by the value of the algorithmic threshold parameter (ε_t).



tioned in previous chapters, this is a rare occurrence in today's complex models, and as such we have chosen to investigate this early, exemplary case of model building in order to motivate what analyses may be possible in a modeling resource that explicitly links parameters to data.

3.4.1 The simplified Hodgkin-Huxley model

To begin, we present the following simplified version of the Hodgkin-Huxley model, which treats the membrane voltage as constant, mimicking the conditions of the voltage clamp experiment that the authors used to collect data on ionic conductance:

$$\begin{aligned} I_{tot} &= I_{Na} + I_K + I_l, \\ I_{Na} = g_{Na}(V_m - E_{Na}), \quad I_K &= g_K(V_m - E_K), \quad I_l = g_l(V_m - E_l), \\ g_{Na} = G_{Na}m^3h, \quad g_K &= G_K n^4, \quad g_l = G_l, \\ \frac{dm}{dt} &= \alpha_m(1-m) + \beta_m m, \\ \frac{dh}{dt} &= \alpha_h(1-h) + \beta_h h, \\ \frac{dn}{dt} &= \alpha_n(1-n) + \beta_n n. \end{aligned} \tag{3.4.1}$$

In this model, all the gate opening rates α and closing rates β, which are normally considered to be functions of membrane voltage V_m, can be simplified to constants. These six parameters will be treated as free parameters of interest for fitting, while the maximum conductance parameters G_{Na}, G_K, and G_l, as well as the initial conditions m_0, h_0, and n_0, will be treated as known.

In the original study, Hodgkin and Huxley performed a series of voltage clamp experiments, depolarizing the membrane from resting potential by some value ΔV_m, then holding it constant and observing the evolution of the sodium and potassium conductances at 11 fixed time points over a 12ms interval. For each voltage, they then sought to determine the values of the six rate parameters α_m, β_m, α_h, β_h, α_n, and β_n that best reproduced the observed behavior by eye. We define a more rigorous metric to evaluate distance between observed conductance data $\mathbf{g}$ and

simulated conductance data $g(\theta)$ at a fixed series of N time points in the following manner:

$$\mathcal{D}(\mathbf{g}, g(\theta)) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{g}_i - g_i(\theta))^2}. \tag{3.4.2}$$

This metric is generally referred to as the root mean squared error, or RMSE, and can be used within an ABC algorithm to perform posterior inference over the model parameters.

We employed the ABC variant of the Toni algorithm to perform posterior inference over these gating rate parameters. This inference is performed separately for sodium and potassium conductance parameters, as the authors made separate recordings of potassium and sodium conductance, which evolve independently under conditions of voltage clamp. For this study, the RMSE function outlined above was employed as a distance metric, and a "no improvement" threshold was set to cause the algorithm to terminate if successive rounds did not decrease the maximum error by more than 0.003. The population size was set to $n = 100$ and cooling parameter set to $\alpha = 0.5$. The number of particles was chosen due to computational limitations of the early, inefficient implementations of the sampler and simulation protocol, which were corrected prior to the work presented in Section 3.5. The α parameter was chosen empirically, so as to give a reasonable ($> 25\%$) acceptance rate for particles in each posterior estimate.

All Hodgkin-Huxley conductance data were obtained from figures 3 and 6 digitized from the original publication using Plot Digitizer.[2] Plot L of the potassium conductance data (figure 3, depolarization of -6mV) was eliminated due to the lack of a reliable scale on the y-axis. The prior distribution was assumed to be independently uniform over each parameter, with a width spanning the reported values under all voltage clamp experiments. A proposal distribution was employed for the Toni algorithm that was independently Gaussian for each variable, as detailed in Table 3.1. Forward simulations were carried out using the Python implementation of the functional curation extension to Chaste, which used a CellML file to specify the model and a custom protocol file to simulate the voltage clamp experiments described in the literature (see Daly et al., 2015, for more details). Default ODE solver settings were used for all analyses.

[2] http://plotdigitizer.sourceforge.net

Table 3.1 Specification of prior and proposal distribution employed over each parameter of the simplified Hodgkin-Huxley model (Equation 3.4.1).

Variables	Prior distribution	Proposal distribution
α_n, β_n	Uniform(0,1)	$\mathcal{N}(0, 0.1)$
α_m, β_m, α_h, β_h	Uniform(0,10)	$\mathcal{N}(0, 1)$

The results of our ABC posterior inference using the Toni algorithm are depicted in Figure 3.10. We see a high homology between the mean posterior estimates and the values of α_n and β_n reported in the original publication. We also see universally small standard deviations about the mean posterior estimates, indicating a high degree of confidence that they reflect a well-constrained optimal value. When the magnitude of the depolarization is smaller, we see a slight systematic deviation of the mean posterior estimates produced by ABC from the original reported values. We attribute this to the ability of our automated method to produce a better fit than the manual methods of Hodgkin and Huxley when fluctuations in the data ($\propto |\min(g_K) - \max(g_K)|$) are small. Indeed, we found the RMSE performances of all particles in these posterior estimates exceed that of the reported parameterization (not shown), indicating an improvement in fit to the experimental data by the ABC estimates.

Figure 3.10 shows a less perfect homology between ABC mean posterior values of α_m, β_m, α_h, β_h and those reported by Hodgkin and Huxley, as well as generally larger standard deviations, which is not unexpected given the increased complexity of the function for sodium conductance (Equation 3.4.1). For the activation term m, which controls the initial increase of sodium conductance, we see that the α_m shows lower homology with reported values for high depolarizations (as well as looser bounds) yet traces from the final ABC estimates were found to achieve better RMSE performance than traces with reported values. The higher uncertainty at large depolarizations despite improvement of fit may be explained by the low frequency of sodium conductance sampling during the action potential in Figure 6 of (Hodgkin and Huxley, 1952), which results in very few data points falling on the spike itself. Poor definition of this portion of the curve would lead to multiple, equally fit choices for α_m, and thus a larger variance

in the ABC posterior. ABC mean values for β_m are reasonably consistent with reported values with some exception towards low depolarizations, where the similar RMSE performance and loose bounds may indicate a non-unique parameterization for the complex function, resulting from low deviation in g_{Na} for $|\Delta V_m| < 10$mV.

For the inactivation term h, ABC posterior estimates for α_h and β_h are both tightly bounded and show high mean homology with reported values (particularly α_h, which is effectively 0 throughout) except for low-magnitude depolarizations. At $|\Delta V_m| < 10$mV, however, the sodium conductance inactivation gate h is effectively irrelevant, as the conductance never reaches a significant spike in activation. At these depolarization values, the shape of the sodium conductance curve begins to resemble that of potassium conductance, the model for which contains no decay term. This lack of impact of h on the observable quantity g_{Na} explains the large variations of the posteriors around α_h and β_h.

3.4.2 The complete Hodgkin-Huxley model

Having demonstrated the application and interpretation of posterior inference on the simplified Hodgkin-Huxley action potential model, we now consider inference in the more complex and realistic case where ion channel gating is affected by membrane voltage as well as time. This means replacing the α and β parameters previously considered with parametric functions of V_m, as described by the original authors (and shown in Equation 2.1.9):

$$\begin{aligned}
\alpha_m(V_m) &= \frac{k_{\alpha_m 1}(V_m + k_{\alpha_m 2})}{\exp\left(\frac{V_m + k_{\alpha_m 2}}{k_{\alpha_m 3}}\right) - 1}, \quad \beta_m(V_m) = k_{\beta_m 1}\exp(V_m/k_{\beta_m 2}), \\
\alpha_h(V_m) &= k_{\alpha_h 1}\exp(V_m/k_{\alpha_h 2}), \quad \beta_h(V_m) = \frac{1}{\exp\left(\frac{V_m + k_{\beta_h 1}}{k_{\beta_h 2}}\right) + 1}, \\
\alpha_n(V_m) &= \frac{k_{\alpha_n 1}(V_m + k_{\alpha_n 2})}{\exp\left(\frac{V_m + k_{\alpha_n 2}}{k_{\alpha_n 3}}\right) - 1}, \quad \beta_n(V_m) = k_{\beta_n 1}\exp(V_m/k_{\beta_n 2}).
\end{aligned} \tag{3.4.3}$$

This expanded model now has has nine free parameters controlling sodium conductance ($k_{\alpha_m 1}$, $k_{\alpha_m 2}$, $k_{\alpha_m 3}$, $k_{\beta_m 1}$, $k_{\beta_m 2}$, $k_{\alpha_h 1}$, $k_{\alpha_h 2}$, $k_{\beta_h 1}$, and $k_{\beta_h 2}$) and five free parameters for potassium

Figure 3.10 Visualizations of ABC posterior estimates for conductance gating rate parameters of the simplified Hodgkin-Huxley model. ABC mean estimates for α and β are depicted in solid lines, with error bars indicating one standard deviation above and below these mean estimates.

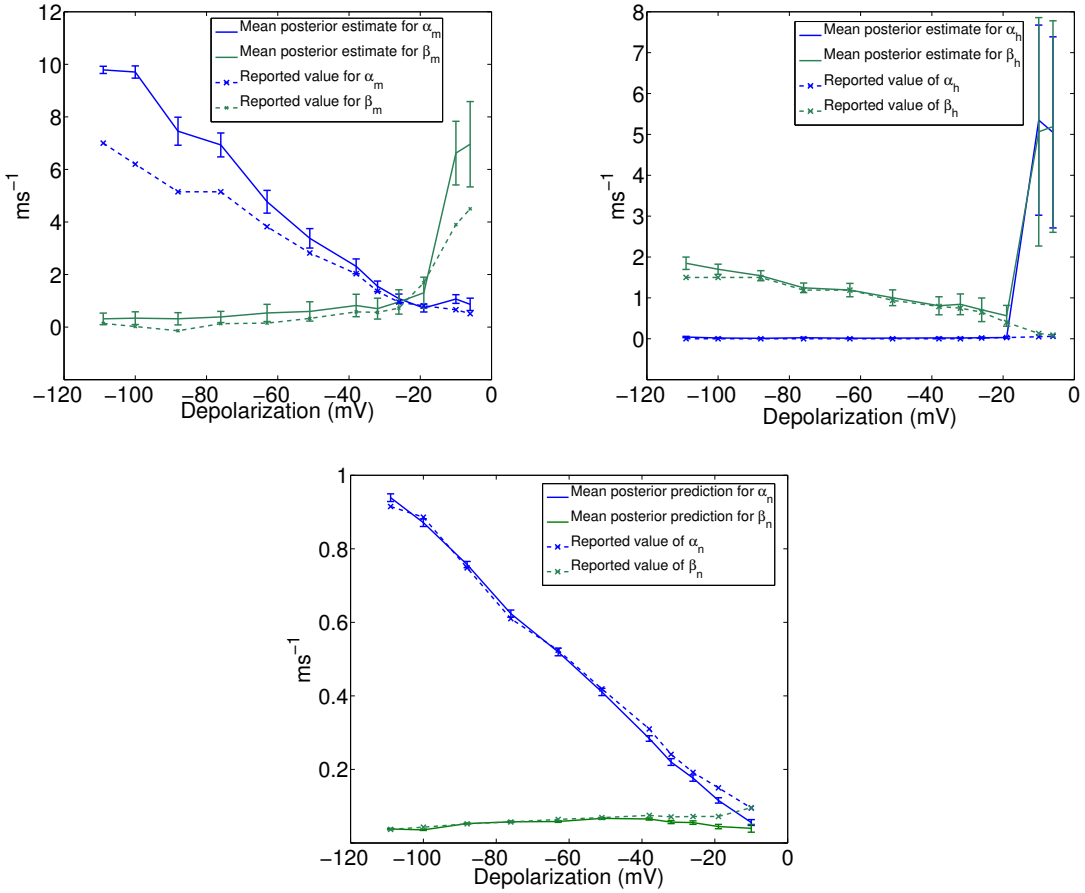


Table 3.2 Specification of prior and proposal distributions employed over each parameter of the complete Hodgkin-Huxley model (Equation 3.4.3).

Parameters	Prior distribution	Proposal distribution
$k_{\alpha_n 1}$, $k_{\beta_n 1}$, $k_{\alpha_m 1}$, $k_{\alpha_h 1}$	Uniform(0,1)	$\mathcal{N}(0, 0.1)$
$k_{\beta_m 1}$	Uniform(0,10)	$\mathcal{N}(0, 1)$
$k_{\alpha_n 3}$, $k_{\beta_n 2}$, $k_{\alpha_m 3}$, $k_{\beta_m 2}$, $k_{\alpha_h 2}$, $k_{\beta_h 2}$	Uniform(1,100)	$\mathcal{N}(10)$
$k_{\alpha_n 2}$, $k_{\alpha_m 2}$, $k_{\beta_h 1}$	Uniform(0,100)	$\mathcal{N}(10)$

conductance ($k_{\alpha_n 1}$, $k_{\alpha_n 2}$, $k_{\alpha_n 3}$, $k_{\beta_n 1}$, and $k_{\beta_n 2}$) for a total of 14 parameters to describe gating (we again assume the maximum conductance parameters and initial conditions to be known). Because the model was only reported by the authors in its final parameterized form, we had to assign the above free parameters to it ourselves in order to perform inference.

The form and parameterization of these equations was chosen by Hodgkin and Huxley to fit the values of α and β that the authors determined by fitting the simplified model to individual voltage clamp recordings. This effectively gives a two-stage fitting process, where voltage-dependence parameters were inferred based on the results of multiple independent fittings to time-dependent data. For our analysis, we instead choose to perform the fitting all at once, incorporating information on both time- and voltage-dependence by considering multiple voltage clamp traces simultaneously. Under this setup, we consider experimental data of the form $\mathbf{g} = [\mathbf{g}_1, \ldots, \mathbf{g}_M]$, where each $\mathbf{g}_i$ represents a voltage clamp reading at N fixed time points $\mathbf{g}_i = [\mathbf{g}_{i,1}, \ldots, \mathbf{g}_{i,N}]$. To measure distance between experimental and simulated data $g(\theta)$, we extend the previous RMSE function (Equation 3.4.2) to the following metric:

$$\mathcal{D}(\mathbf{g}, g(\theta)) = \frac{1}{M} \sum_{i=1}^{M} \sqrt{\frac{1}{N} \sum_{j=1}^{N} (\mathbf{g}_{i,j} - g_{i,j}(\theta))^2}. \tag{3.4.4}$$

Using this distance metric, and again considering sodium and potassium components separately, we performed ABC posterior inference using the Toni algorithm, again with $n = 100$ particles, cooling schedule $\alpha = 0.5$, and no-improvement criterion of 0.003. All other simulation settings were the same, and we again employed independent uniform priors and independent Gaussian proposal distributions over all parameters, as described in Table 3.2.

We report summaries of the ABC posterior estimates over sodium parameters in Table 3.3,

Table 3.3 Summary of the ABC posterior estimate for voltage dependency parameters of sodium conductance gating rates. "Mean" and "Variance" columns indicate the marginal posterior mean and variance for each parameter, while "5th, 95th Percentiles" denote the points at which the posterior cumulative density function (CDF) records 5% and 95% density, respectively (analogous to 90% confidence interval). The "RMSE" row contains the value of the distance metric attained by the reported parameters, as well as the minimum and maximum (bounded from above by the final ABC error constraint ϵ_T) values attained by particles in the final posterior estimate.

	Reported	ABC Posterior Estimates		
Parameter	Value	Mean	Variance	5th, 95th Percentiles
$k_{\alpha_m 1}$	0.1	0.110	$3.69 * 10^{-5}$	0.102, 0.121
$k_{\alpha_m 2}$	25	13.3	3.43	10.8, 16.1
$k_{\alpha_m 3}$	10	3.73	1.36	1.68, 5.66
$k_{\beta_m 1}$	4	2.59	0.0915	2.17, 3.12
$k_{\beta_m 2}$	18	95.9	9.87	91.1, 99.6
$k_{\alpha_h 1}$	0.07	0.414	0.0978	0.0315, 0.917
$k_{\alpha_h 2}$	20	5.73	17.3	1.08, 13.0
$k_{\beta_h 1}$	30	40.6	8.74	36.8, 45.6
$k_{\beta_h 2}$	10	13.6	3.01	11.5, 16.6
RMSE	1.93	Min 1.20, Max ($\leq \epsilon_T$) 1.22		

and potassium parameters in Table 3.4. We additionally visualize the voltage dependency of the gating rate parameters calculated for each particle in the posterior estimate in Figure 3.11, in order to demonstrate how variability in the 9 free parameters affects model behavior.

While Table 3.3 shows several parameters with high mean deviation from reported values ($k_{\alpha_m 2}$, $k_{\alpha_m 3}$, $k_{\beta_m 2}$, and $k_{\alpha_h 2}$), posterior estimates for all parameters show low relative variance and 90-percentile spread with the exception of $k_{\alpha_h 2}$. This, coupled with the substantial decrease in RMSE of all particles in the posterior when compared to the model under reported parameterization (indicated in the bottom row of Table 3.3), suggests that ABC has arrived at a relatively tightly bounded optimum exceeding that of the manual methods of the original paper. Figure 3.11 seems to support this, as posterior traces around α_m, β_m, and β_h noticeably deviate from the reported traces yet maintain a constrained form across all values of V. When considering the traces of α_h in Figure 3.11, we see that there is a considerable increase in spread at low values of ΔV_m. As we established in the posterior inference analysis on the simplified Hodgkin-Huxley

Figure 3.11 Visualizations of ABC posterior estimates for voltage dependency parameters in the complete Hodgkin-Huxley model. Each particle in the ABC posterior estimate for the voltage dependency parameters is used to generate a trace of α, β according to Equation 3.4.3 in a dotted line, while traces generated by reported parameterizations are represented by solid lines.

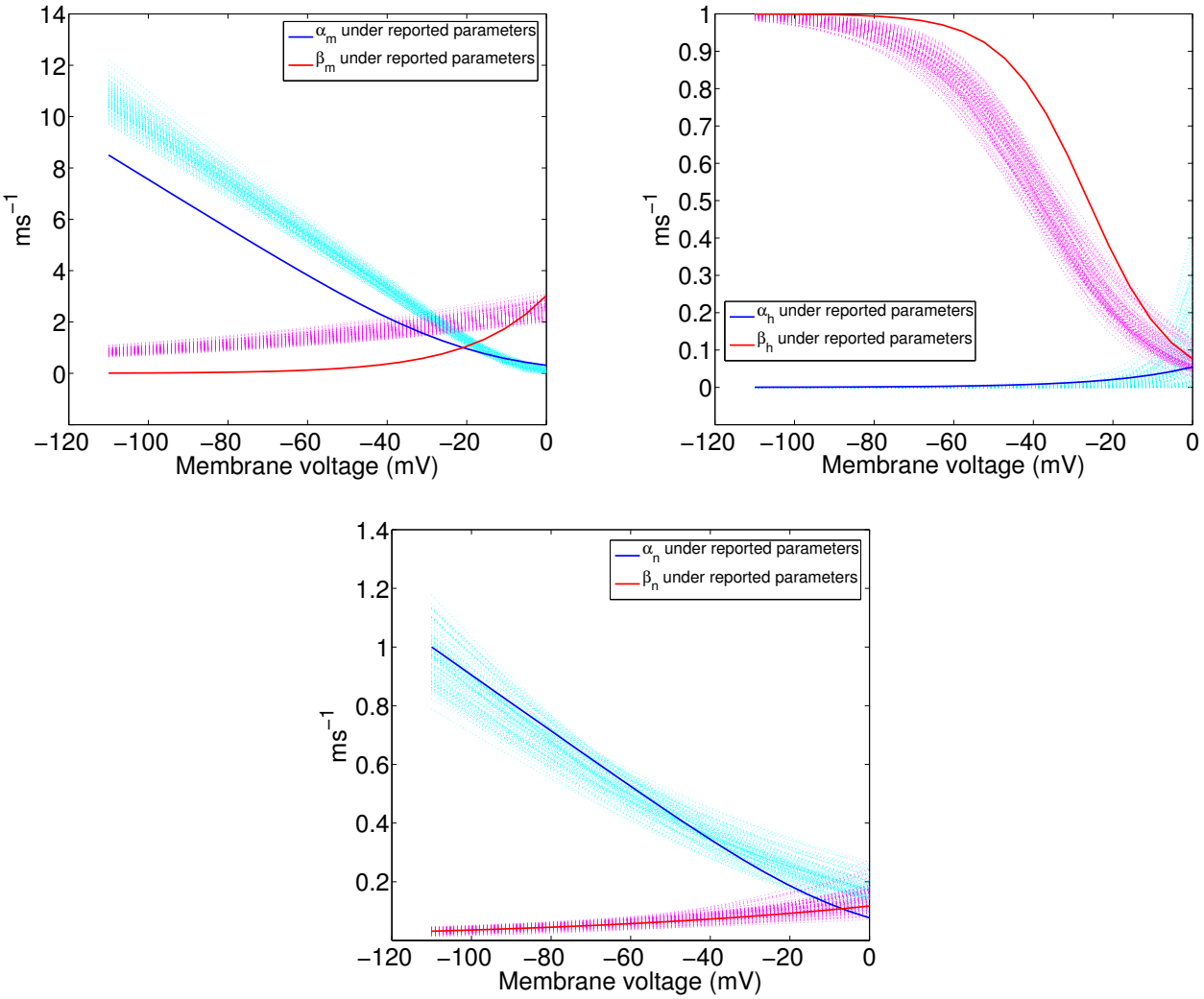


Table 3.4 Summary of ABC posterior estimate for voltage dependency parameters of potassium conductance gating rates. "Mean" and "Variance" columns indicate the marginal posterior mean and variance for each parameter, while "5th, 95th Percentiles" denote the points at which the posterior cumulative density function (CDF) records 5% and 95% density, respectively (analogous to 90% confidence interval). The "RMSE" row contains the value of the distance metric attained by the reported parameters, as well as the minimum and maximum (bounded from above by the final ABC error constraint ϵ_T) values attained by particles in the final posterior estimate.

	Reported	ABC Posterior Estimates		
Parameter	Value	Mean	Variance	5th, 95th Percentiles
$k_{\alpha_n 1}$	0.01	0.0128	$4.29 * 10^{-6}$	0.0103, 0.0177
$k_{\alpha_n 2}$	10	44.8	162	27.1, 62.1
$k_{\alpha_n 3}$	10	30.7	48.2	18.7, 37.4
$k_{\beta_n 1}$	0.125	0.177	$3.14 * 10^{-3}$	0.0986, 0.253
$k_{\beta_n 2}$	80	65.1	218	48.9, 95.2
RMSE	0.642	Min 0.559, Max ($\leq \epsilon_T$) 0.794		

model, the system is highly insensitive to the h gate under these conditions, allowing for large variation in both the gating rate α_h and its underling voltage dependency parameters $k_{\alpha_h 1}$ and $k_{\alpha_h 2}$.

Turning our attention to the posteriors on potassium conductance, we note that the mean posterior estimates of $k_{\alpha_n 2}$ and $k_{\alpha_n 3}$ demonstrate high deviation from the reported values. Additionally, we find the reported values of these parameters lie outside the 90-percentile range around this mean. The significance of this deviation, however, is negated by the large spread of posterior estimates for these parameters, quantified both by the variance about the mean and the width of the 90-percentile range, which suggests a lack of confidence in the mean posterior estimate. Similarly, while the reported value of $k_{\beta_n 2}$ lies within the 90-percentile range about the mean posterior estimate, the width of the distribution suggests the "recovered" estimate is not unique. This suggests that the data may be insufficient to constrain one or more of these parameters, and the model as posited may suffer from unidentifiability. While a full identifiability analysis is beyond the scope of this chapter, and the performance of such analyses will be discussed further in Chapter 4, we are comfortable ruling out a structural unidentifiability in the model. In the case of a structural unidentifiability, we would expect low variation in the

function output despite high variation in parameter space. Instead, in Figure 3.11 we observe a relatively wide distribution around α_n at all values of V, and a widening distribution around β_n at low values of V when k_{β_n} begins to dominate the exponential portion of the function.

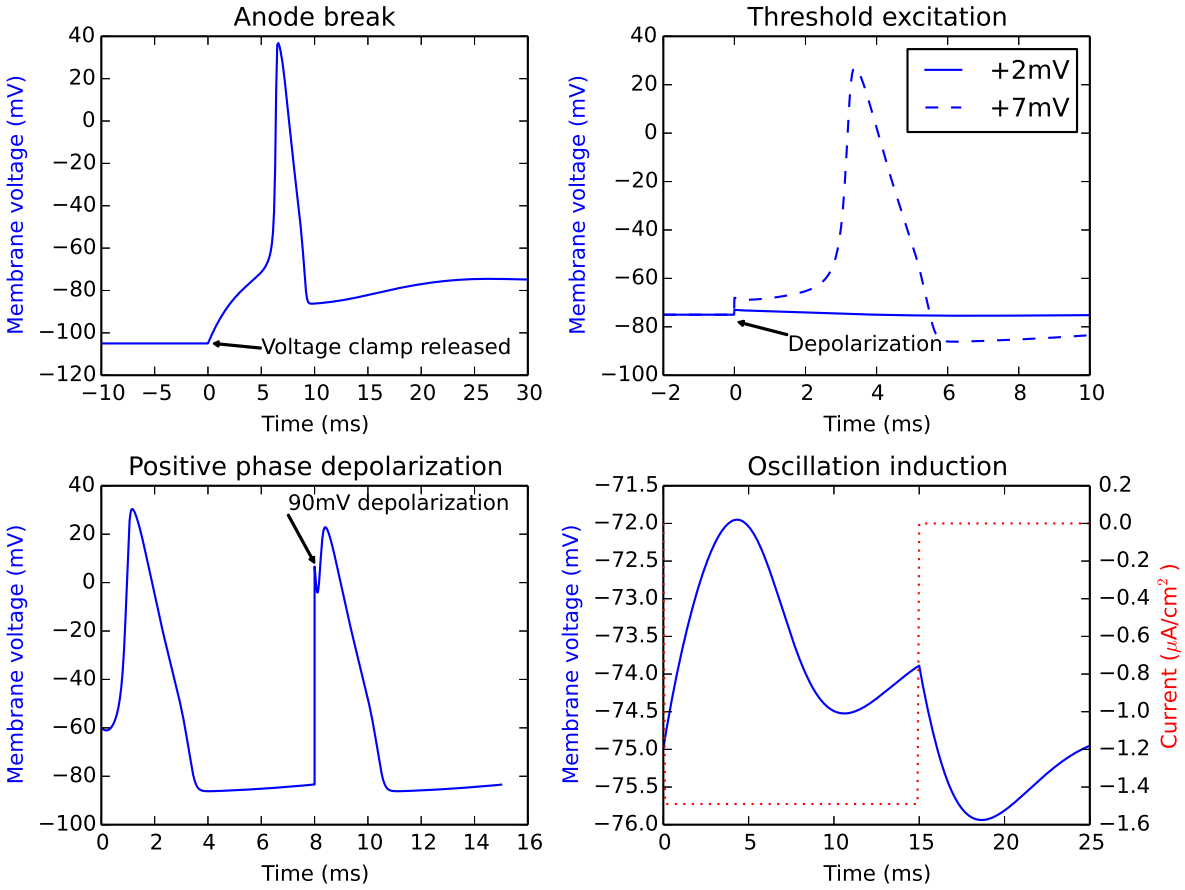
3.4.3 Validation against other voltage protocols

In order to fully understand the consequences of posterior variation in model parameters, and how this information can be used to inform the design of future experiments, we now turn our attention to model validation. In the analyses of the Hodgkin-Huxley model so far, we have only considered "training data" — data used in the parameterization of a model. While the uncertainty about model parameters that results from this training process can tell us about the suitability of the data used to fit the model, we can also employ "validation data" — data which have not been used in the fitting process — to assess the effects of this variation on the model's ability to predict new phenomena.

This process was considered by Hodgkin and Huxley, who validated their final, parameterized model by ensuring that it was able to capture behavior observed in cells subjected to four voltage protocols that differed greatly from the voltage clamp protocol that was used to fit the model. These protocols are concerned with the response of membrane voltage V_m to stimulus, and thus probe whole-cell behavior rather than the single ion conductance recordings carried out during voltage clamp experiments. The four protocols considered are represented visually in Figure 3.12, and described in greater detail in the following labeled subsections. We will investigate how posterior uncertainty in the model parameters from the full Hodgkin-Huxley model, as determined by ABC posterior analysis in the previous section, translates to uncertainty in response to these protocols.

Anode Break: This protocol introduced an anodal polarization, lowering potassium conductance activation and sodium conductance inactivation, reversing the membrane current at resting potential and causing full excitation after release of the clamp.

Figure 3.12 Stimulus-response representation of the four advanced voltage protocols proposed by Hodgkin and Huxley. Membrane voltage, the output quantity, is represented in blue. In the anode break (top left), threshold excitation (top right), and positive phase depolarization (bottom left) experiments, membrane voltage is manipulated directly as described in Section 3.3.3. The graph representing the oscillation induction experiment (bottom right) shows the induced stimulus current (dotted red) and the resulting fluctuation in membrane potential. In all graphs, the models are at steady-state by time $t = 0$ when stimulus is applied.



From steady state, the model was subjected to a 200ms voltage clamp to 30mV below resting potential. This clamp was then released and the membrane voltage recorded every 0.1ms for the duration of a 30ms time course simulation (Figure 3.12, top left).

Threshold Excitation: This protocol sought to assess behaviour of the membrane potential under depolarizations insufficient to trigger a full action potential.

After reaching steady state, the model was subjected to a membrane depolarization of either -10mV, 2mV, 5mV, 6mV, or 7mV, with only the latter being sufficient to trigger an action potential under reported values for gating rate voltage dependency parameters (Figure 3.12, top right). The membrane voltage was then recorded every 0.01ms for the duration of a 10ms time course simulation.

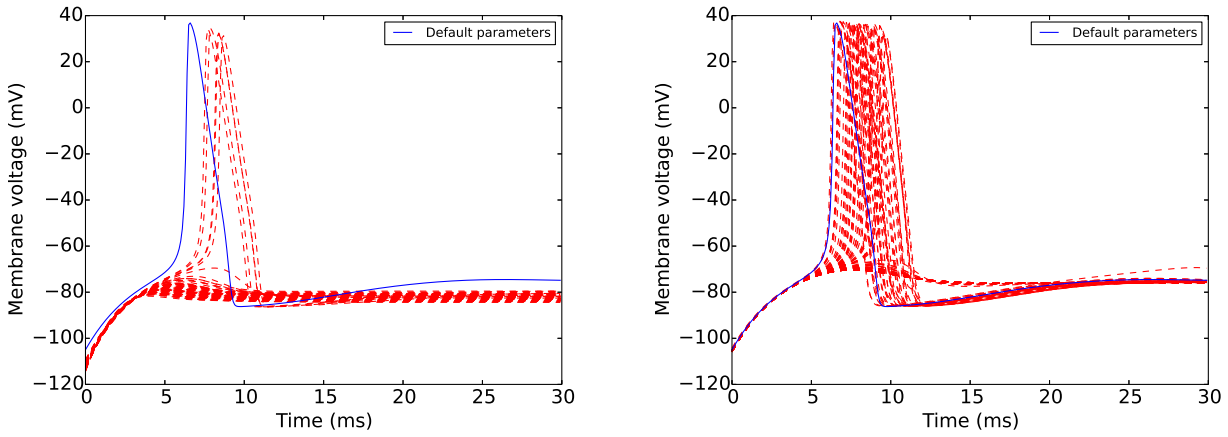
Positive Phase Depolarization: This protocol was designed to determine the degree to which a voltage stimulus during the recovery phase of the action potential could trigger a secondary activation.

After reaching steady state, the model was subjected to a 15mV membrane depolarization followed by a 5ms, 6ms, or 8ms timecourse simulation. After this, the model was subjected to an additional 90mV depolarization and the remainder of a 15ms total time course simulation (Figure 3.12, bottom left). Membrane voltage was recorded every 0.01ms for the total duration of the 15ms time course following the initial depolarization.

Oscillation Induction: This protocol probed the observed oscillation of the membrane potential in response to small, sustained current injections.

After reaching steady state, the membrane current of the model was clamped to -1.49mA/cm^2 for the duration of a 15ms time course simulation. The clamp was then released and the model subjected to an additional 10ms time course simulation, with membrane voltage being recorded every 0.1ms for the entirety of the 25ms time course (Figure 3.12, bottom right).

Figure 3.13 Membrane voltage response to the anode break excitation protocol. Models parameterized according to the posterior estimates for potassium (a) or sodium (b) parameters of Equation 3.4.3 are shown in red, while response under default parameters is shown in blue.



The particles composing the posterior estimates produced in the previous section were, by definition, equally able to reproduce the voltage clamp data used to train the model. For both sodium and potassium posteriors separately, we parameterized Equation 3.4.3 according to each particle in the posterior estimate, holding the parameters of the other ionic current to reported values, and applied the four voltage protocols outlined above to the resulting model. This allowed us to visualize the variation in cell-level behavior that results from variation allowed by the voltage clamp protocol, the results of which are presented in Figures 3.13–3.16.

When examining the effects of posterior variation in sodium conductance parameters, we see that across all four voltage protocols, a majority of the predictions agree with the observed behavior in a qualitative sense, with some showing near exact agreement. The range in behavior under the anode break protocol, for example (Figure 3.13b), suggests that the uncertainty about the sodium conductance parameters resulting from inference on voltage clamp data manifests itself as a temporal shift in the AP-like response to hyperpolarization. Knowing the correct timing of such a response in the current context of use, one could further constrain the model by incorporating this data in fitting, or at the very least, choose a parameterization from an existing posterior estimate that best suits the needs of a current simulation. It could be said of any of

Figure 3.14 Membrane voltage response to the threshold depolarization protocol for (from top to bottom) 7mV, 2mV, -10mV. Models parameterized according to the posterior estimates for either potassium (a-c) or sodium (d-f) parameters of Equation 3.4.3 are shown in red, while response under default parameters is shown in blue.

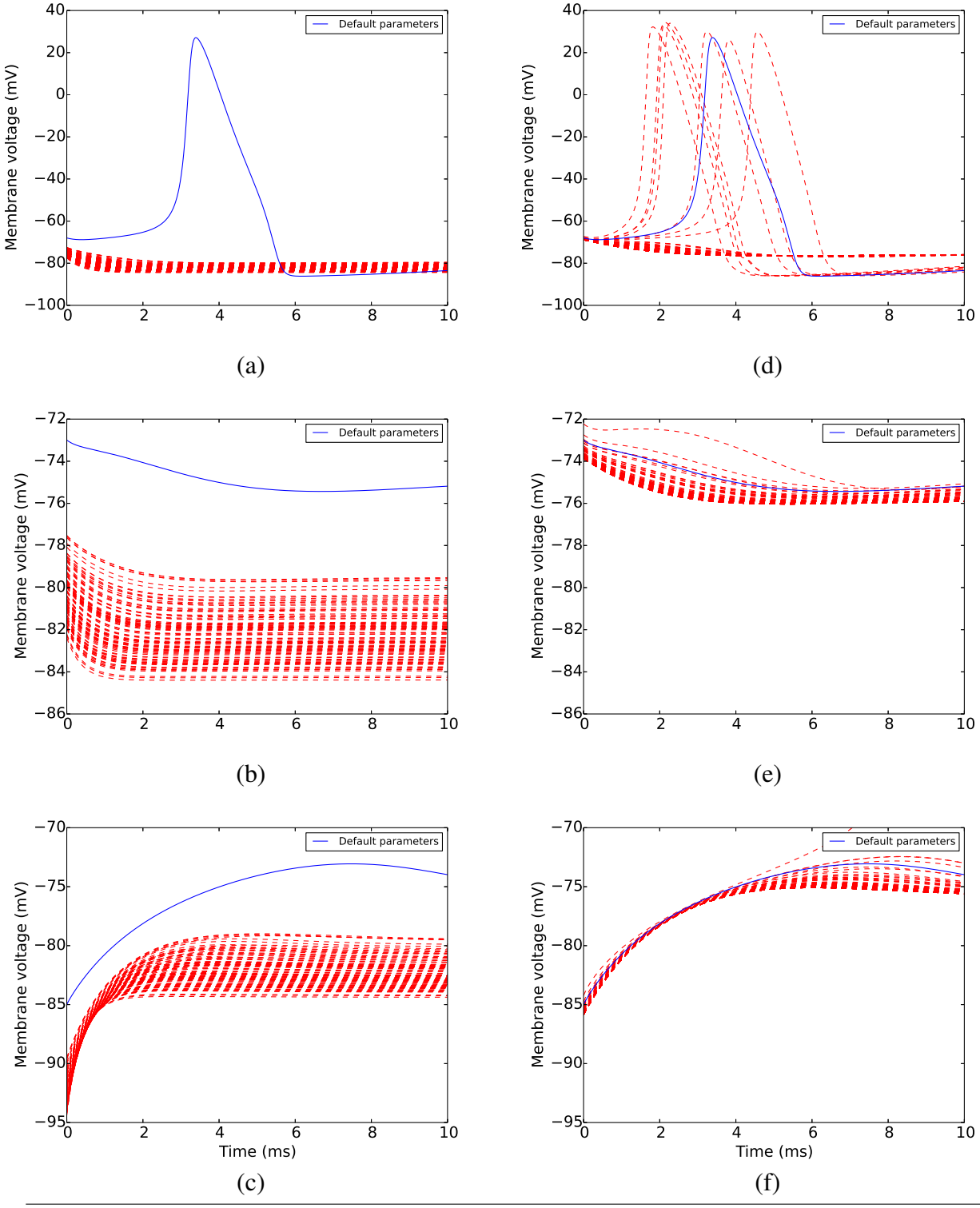


Figure 3.15 Membrane voltage response to the positive phase depolarization protocol at (from top to bottom) 5ms, 6ms, 8ms. Models parameterized according to the posterior estimates for either potassium (a-c) or sodium (d-f) parameters of Equation 3.4.3 are shown in red, while response under default parameters is shown in blue.

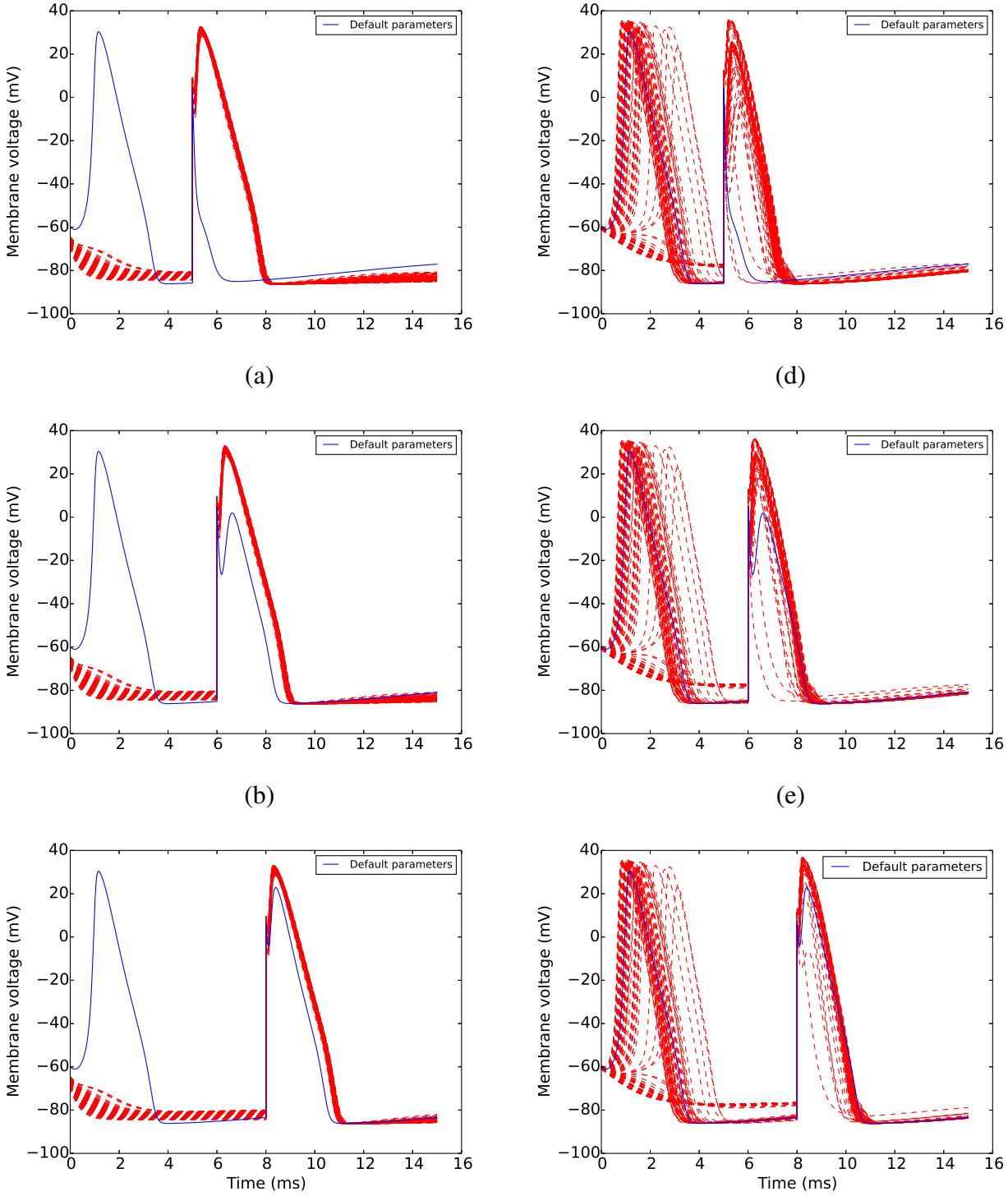
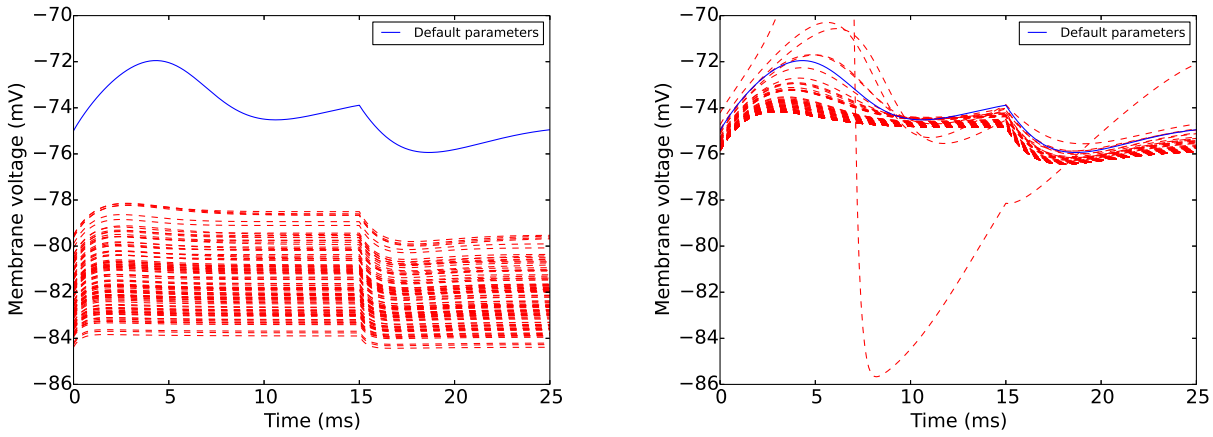


Figure 3.16 Membrane voltage response to the oscillation induction protocol. Models parameterized according to the posterior estimates for potassium (a) or sodium (b) parameters of Equation 3.4.3 are shown in red, while response under default parameters is shown in blue.



the four protocols presented here that data collected from such an experiment could be used to validate the behavior of, or further constrain, a model of sodium conductance fit to voltage clamp data.

When examining the effects of posterior variation in potassium parameters, however, we see that aside from the anode break protocol, the posterior parameterized models largely show qualitative deviation in response from the predicted behavior. Under the anode break excitation experimental protocol, the model's membrane voltage showed variable behaviour over the potassium posterior ranging from total inactivation to full excitation nearly matching the experimental trace in both magnitude and temporal placement (Figure 3.13(a)). This suggests that the experiment might be informative for further constraining one or more parameters that showed high posterior variation under voltage clamp data alone, which is not entirely surprising given that the hyperpolarizing current alters the potassium conductance at resting potential, potentially revealing ion channel kinetics that were absent under the simpler protocols.

Both the positive phase depolarization experiment and the 7mV (Figure 3.14a) threshold excitation experiment reveal the inability of the posterior parameterized models to trigger an action potential in response to the same membrane stimulus sufficient for the reported parame-

terization. The positive phase depolarization experiments (Figure 3.15a-c) also show a more exaggerated second activation response to a stimulation during restitution across all posterior parameterized models. At low depolarization (Figure 3.14b) or slight polarization (Figure 3.14c) of the membrane, as well as the oscillation induction experiment (Figure 3.16a), the posterior parameterized models showed the same qualitative behaviour, yet at a noticeable downward shift in voltage.

This inability to recreate the reported behaviour may be due to the fact that the authors set the parameters of the leakage current component of the model (I_l in Equation 3.4.1) after fitting the sodium and potassium conductances as a "catch-all" to ensure matching to experimental behaviour. As such, this value is likely not a biological truth, and employing it without a similar adjustment for our parameterizations may lead to this failure to capture the same cell-level behaviour. Regardless, it appears from these results that separately fitting the potassium conductance parameters to the voltage clamp experiments with ABC failed to produce a parameterized model capable of exhibiting all of the behaviour reported by the authors.

3.5 The O'Hara-Rudy Cardiac AP Model

Having considered the Hodgkin-Huxley model in great detail, we now turn our attention to performing inference on the O'Hara-Rudy model — a recent model of the cardiac action potential that employs the Hodgkin-Huxley formalism (O'Hara et al., 2011). The O'Hara-Rudy model contains 13 ionic conductances, carried by a mix of ion channels, pumps, and exchangers, as well as a stimulus current, giving it a complexity that far exceeds that of the Hodgkin-Huxley model. In this section, we will consider how the SMC algorithms handle models of this size, as well as how they may be used in the presence of summary statistic data.

In this study of the O'Hara-Rudy model, we will be concerned with fitting the parameters controlling maximum conductance rather than the kinetic parameters considered in our previous study of the Hodgkin-Huxley model. As a result, we will be employing data on membrane

Table 3.5 Reported values and prior distributions for parameters of the O'Hara-Rudy cardiac action potential model.

Parameter	Reported value ($mS/\mu F$)	Uniform prior range ($mS/\mu F$)
G_{Na}	75	[32, 150]
G_{NaL}	7.5×10^{-3}	[3.2×10^{-3}, 0.015]
G_{to}	0.02	[0.01, 0.04]
G_{CaL}	1×10^{-4}	[5×10^{-5}, 2×10^{-4}]
G_{Kr}	0.046	[0.023, 0.092]
G_{Ks}	3.4×10^{-3}	[1.7×10^{-3}, 6.8×10^{-3}]
G_{K1}	0.1908	[0.09, 0.4]
G_{NaCa}	4×10^{-4}	[2×10^{-4}, 8×10^{-4}]
G_{NaK}	30	[15, 60]
G_{Nab}	3.75×10^{-10}	[1.8×10^{-10}, 7.5×10^{-10}]
G_{Kb}	3×10^{-3}	[1.5×10^{-3}, 6×10^{-3}]
G_{pCa}	5×10^{-4}	[2.5×10^{-4}, 1×10^{-3}]
G_{Cab}	2.5×10^{-8}	[1.25×10^{-8}, 5×10^{-8}]

voltage rather than data on individual ionic conductance. In the Hodgkin-Huxley model, which had two types of ionic current (besides the leakage current which served as a correction for the effects of yet-unknown ion transporters), it was simple enough to collect data on individual currents and use these to fit the parameters controlling corresponding ion channel kinetics. When considering macroscopic data such as membrane voltage, these kinetic parameters are often taken as known quantities (often chosen by separate studies using patch clamp experiments), and modelers instead seek to fit the maximum conductance parameters G_x. Consequently, in this section we will consider the O'Hara-Rudy model to contain 13 free parameters, which we detail in Table 3.5, along with their reported values and assigned prior distributions (independently uniform between 50% and 200% of the reported value). Full details of the Hodgkin-Huxley-style specifications of ion channel kinetics for this model can be found in the original publication (O'Hara et al., 2011).

When fitting these conductance parameters, we employed two types of data: time-dependent voltage data collected during a single action potential, and five common summary statistics calculated from said action potential (APD50, APD90, maximum upstroke velocity, rest potential, and peak potential). When employing voltage data, we made the assumption that experimental

data was affected by independent, identically distributed Gaussian noise at each point, which is a standard assumption for most time-series data. This allows us to employ Bayesian inference with the Gaussian acceptance kernel described in Equation 3.2.3, where we set the width of the experimental noise to $\sigma_e = 0.25$ based on discussions with experimental collaborators (the width of this distribution can also be fit as a separate parameter). When employing summary statistic data, however, we were unable to derive a calculable likelihood function due to the nonlinear transformation of Gaussian error from the underlying trace. We were instead required to employ ABC methods with the top-hat kernel described in Equation 3.2.1, using square error between vectors of simulated and observed summary statistics. Because the five summary statistics were of similar scale under reported parameters, they would contribute roughly equally to the overall distance metric, though in other applications one may consider normalizing individual summary statistics by some nominal value in order to adjust this behavior.

For this study, we considered simulated experimental data in order to investigate the effects of employing summary statistic data on posterior uncertainty while the model was known to be correct. While the approach we will describe can also be applied to experimentally collected data, the use of simulated data allows us to assess the information content of summary statistics *a priori*, before spending time and resources on data collection. Action potentials were simulated using the functional curation extension to Chaste, using a CellML representation of the O'Hara-Rudy model obtained from the CellML model database[3] and a custom protocol file. The protocol simply recorded the membrane voltage every 0.25ms for a duration of 500ms after the application of the stimulus current. The voltage data used for fitting were generated by executing a simulation, then adding Gaussian noise ($\sigma_e = 0.25$, as described above) to each recorded time point. The summary statistic data used for fitting were calculated from this voltage trace using commands in the post-processing library of functional curation.

We chose to perform both Bayesian and approximate Bayesian inference on the O'Hara-Rudy model using the Del Moral sampler only. Through investigation of a simple linear model,

[3] `https://models.cellml.org/cellml`

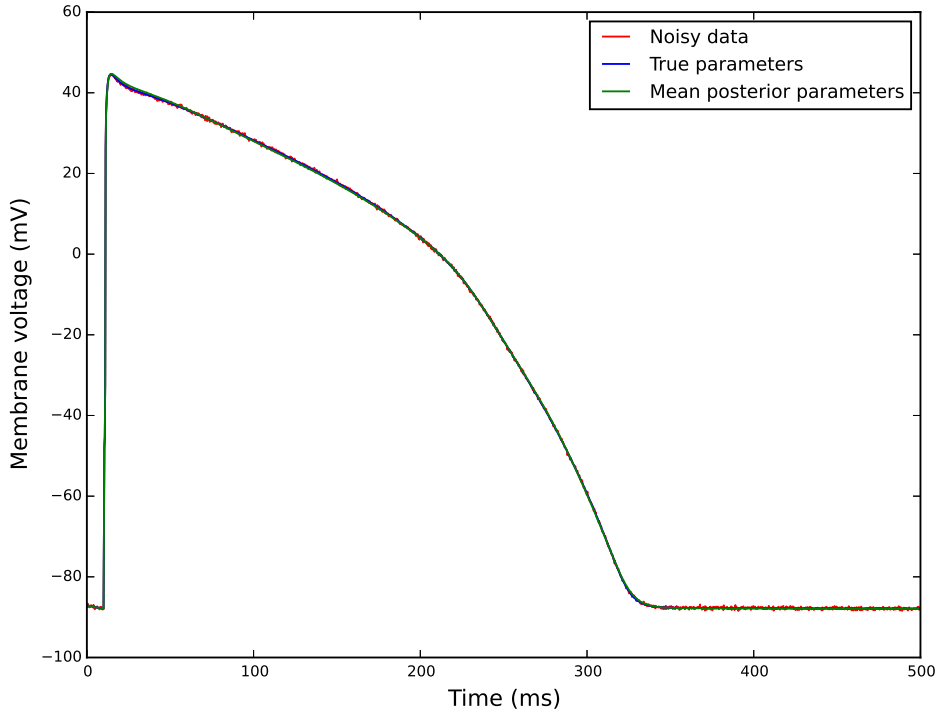
which we detailed in Section 3.3.1, we found that when performing exact inference, the Toni sampler was highly sensitive to the choice of the normalizing constant in the kernel function (c_ε in Equation 3.2.3). Many choices for this constant caused the algorithm to either produce truncated posteriors or to reject many samples — an effect which we had forseen and illustrated in Figure 3.1, and to which the Del Moral sampler is immune due to its use of Metropolis-Hastings steps. We additionally demonstrated in Section 3.3.2 that the Toni sampler suffered from convergence issues in high-dimensional parameter space, while the Del Moral sampler was more resistant to these effects.

Due to the computational complexity added by calculating summary statistics, the Del Moral sampler was limited to 1,000 particles for approximate inference, and this number of particles was maintained during exact inference for consistency. To ensure gradual evolution of the posterior, and consequently make Equation 3.2.5 a good approximation of the true importance sampling weights, the cooling schedule α was set to 0.2, and resampling was triggered at 60% ESS (as opposed to 50% on line 16 of Algorithm 3.3). An additional minimum of 30% ESS was set for all posterior updates (as opposed to the simpler nonzero requirement on line 10). The approximate Del Moral sampler was set to terminate when the threshold ε fell below the value of the objective function evaluated for data simulated under the generating parameters (Table 3.5). The probabilistic solver was set to terminate at $\varepsilon = 1$, the point at which exact posterior sampling was being performed.

We first consider the results of performing posterior inference over the full 13-parameter model using voltage data. From the marginal distributions presented in Figure 3.17, we see that the marginal posteriors are well-constrained for both major currents such as G_{Na} and minor currents such as G_{Nab}, but may show noticeable mean deviation from the values used to generate the data. We expect deviations of the posterior means from generating values due to the effects of random noise, as well as redundancies in the system, in which parameters may vary in a compensatory manner so that model output remains largely unchanged. The presence of the latter effect can be confirmed by Figure 3.18, in which we observe great consistency between action

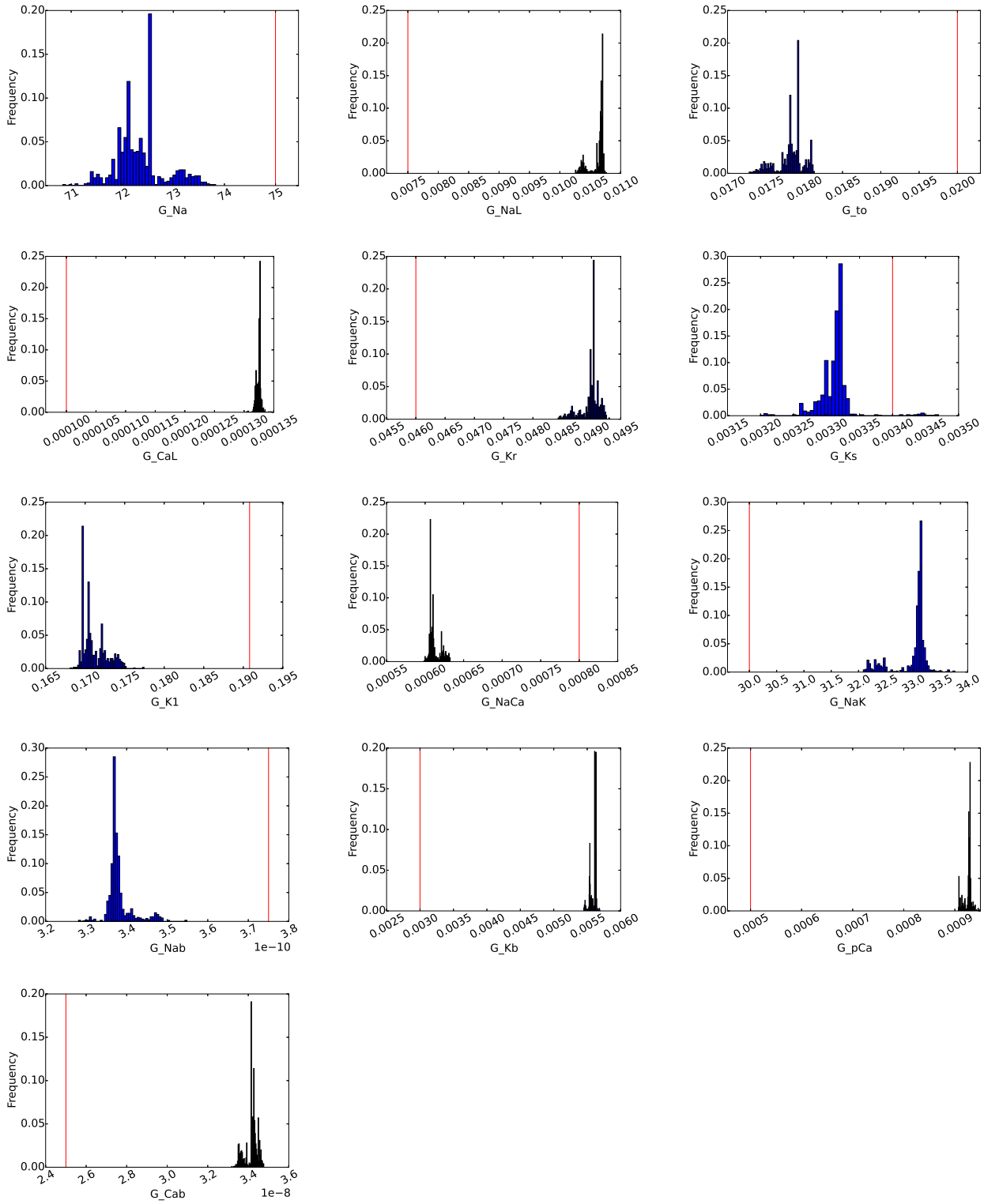
potentials simulated under the mean posterior values and the data used to fit the model despite their differing values. It is also worth noting that the width of these posterior distributions is a function of the number of sample points employed, and thus the location of the generating values outside of some "credible interval" of the posterior mean can be most accurately attributed to low variance in the estimator.

Figure 3.18 O'Hara-Rudy action potential trace generated by the mean posterior parameter values from the probabilistic Del Moral algorithm. The mean posterior trace is shown in green, while noisy time trace data used to fit the model are shown in red, and the noise-free trace using the generating parameter values is shown in blue.



To ensure that the solver was not becoming stuck in a local minimum, we repeated the fitting 10 times and reported the results in Table 3.6 (top). For nearly all model parameters, particularly the major currents $G_{Na}, G_{to}, G_{Kr}, G_{K1}$, the posterior mean averaged over these ten repeated inferences showed strong agreement with the generating values, as well as low standard deviation across the repeats. As such, the mean posterior deviations observed in single instances of in-

Figure 3.17 All marginal distributions from the posterior produced by the probabilistic Del Moral algorithm for the 13-parameter O'Hara-Rudy action potential model. Inference was performed using full time-course data. Generating values are shown as vertical red lines. All conductances are displayed in $mS/\mu F$.



ference are likely due to the difficulty of covering the 13-dimensional space with only 1,000 sampler particles. While greater consistency between the posterior mean and the generating values may be obtained by increasing the population size of the sampler, computational restrictions imposed by the calculation of summary statistics in our subsequent analyses limited us to 1,000 particles in the approximate sampler. As such, we maintained this number across both exact and approximate samplers to allow for maximal homology between the two during our comparisons.

We next consider the results of posterior inference using summary statistic data. In Figure 3.19, we see a general widening of the approximate marginal posteriors relative to those produced using full time course data, as well as multimodality approaching uniformity in the posteriors over parameters such as G_{Nab} and G_{NaCa}. In Table 3.6 we quantify the increase in uncertainty resulting from the use of summary statistics by presenting the ratio of posterior to prior standard deviation, both when full data are employed (σ_P/σ_0) and when only summary statistics are employed (σ_A/σ_0). These ratios normalize against the scale and initial spread of the parameters, allowing for the assessment of relative uncertainty about a parameter after inference. In Table 3.6 (top), we see that that G_{Na} and G_{Kr} remain the two best constrained parameters when employing summary statistic data, as indicated by the value of σ_A/σ_0, with nothing else even reaching 50% reduction in prior uncertainty. If we compare the reduction in uncertainty for these two parameters when using summary statistic data to the case when full time course data were used (σ_P/σ_0), however, we see that G_{Na} shows less than two-fold relative increase in uncertainty when summary statistics were employed, while G_{Kr} shows a nearly 10-fold increase in relative uncertainty under the same conditions. Thus, while we can tell simply by examining posterior-prior variance ratios that the five summary statistics contain the most information about G_{Na}, and to a lesser extent G_{Kr}, we can tell by comparing to the corresponding ratios from exact inference over time-course data that a disproportionate amount of information is being lost by some major (G_{Kr}, G_{to}, G_{K1}) and minor (G_{Nab}, G_{NaK}) currents when these summary statistic data are employed instead.

Figure 3.19 All marginal distributions from the posterior produced by the approximate Del Moral algorithm for the 13-parameter O'Hara-Rudy action potential. Inference was performed using summary statistic data. Generating values are shown as vertical red lines. All conductances are displayed in $mS/\mu F$.

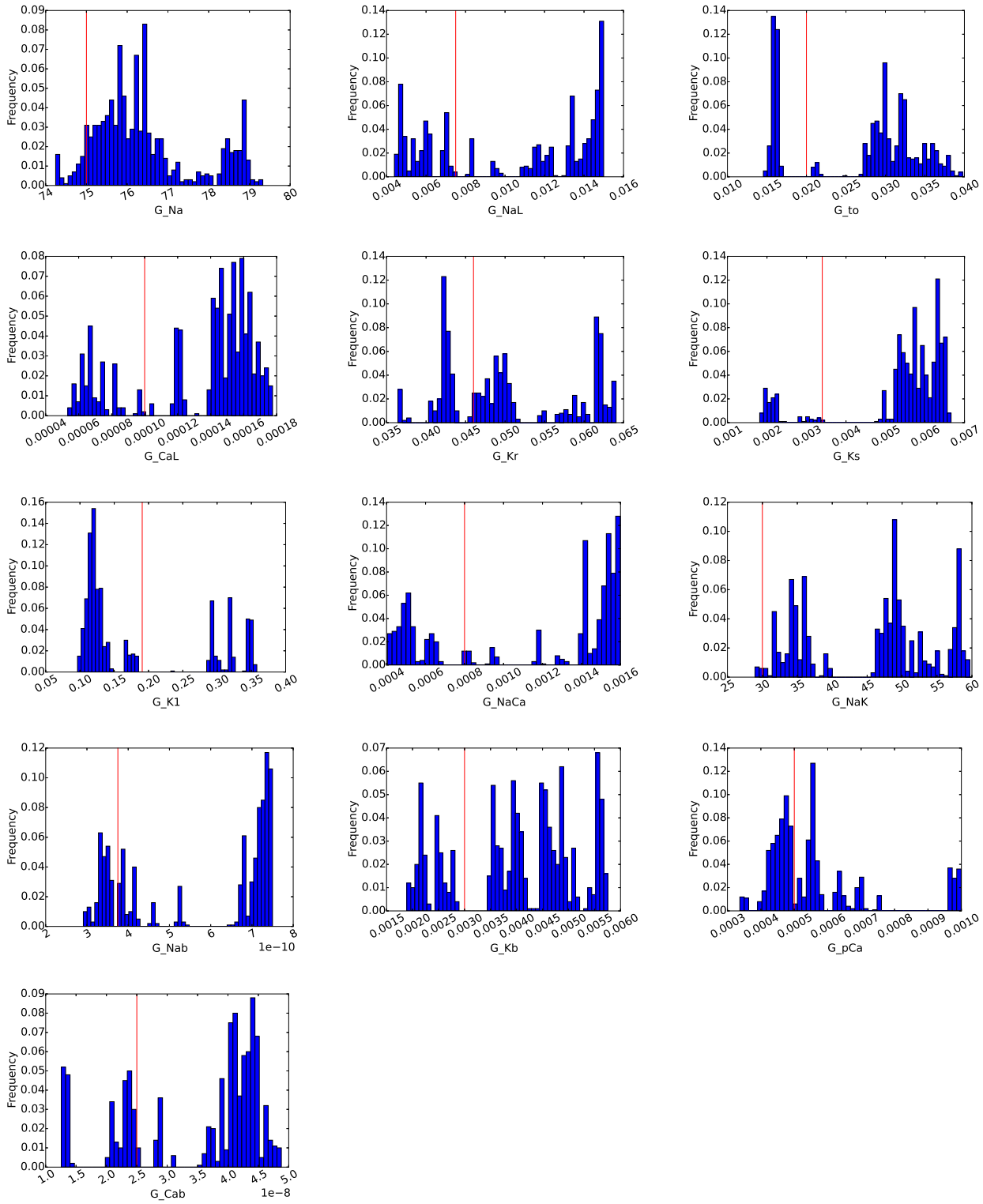


Table 3.6 Summaries of the marginal posteriors over free O'Hara-Rudy model parameters produced by the Del Moral algorithm.

13 Free Parameters

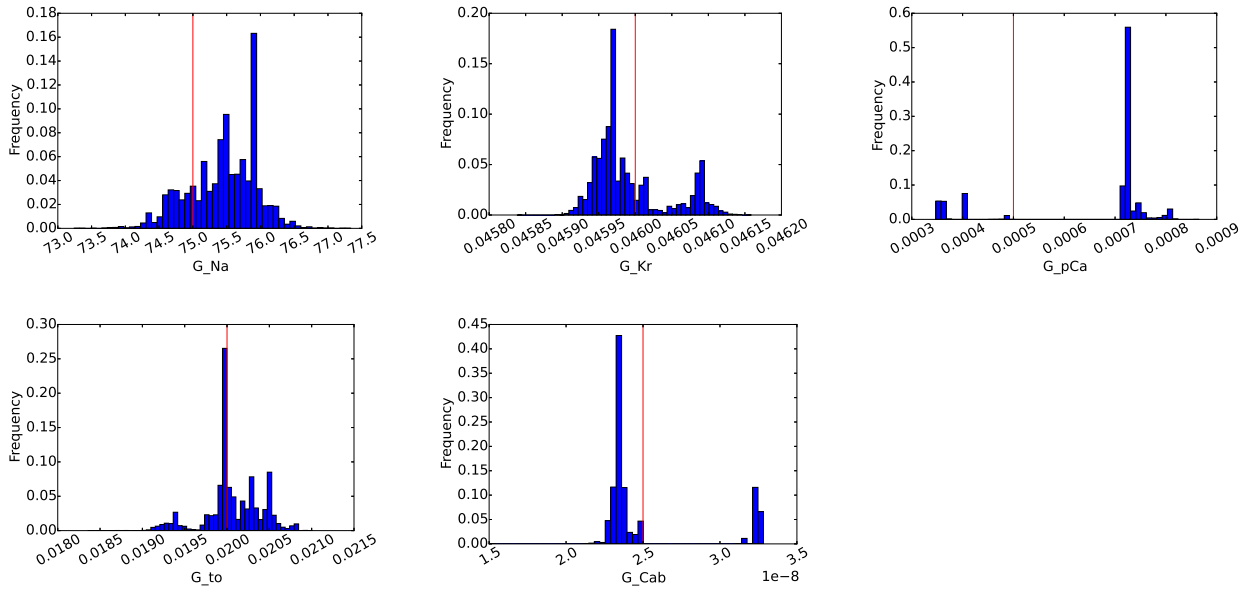
Name	True Value	μ_P	(std.)	σ_P/σ_0	(std.)	μ_A	(std.)	σ_A/σ_0	(std.)
G_{Na}	75	75.8	(3.90)	0.0173	(0.00855)	76.4	(0.752)	0.0292	(0.0115)
G_{NaL}	7.5×10^{-3}	8.06×10^{-3}	(2.80×10^{-3})	0.131	(0.252)	1.04×10^{-2}	(8.50×10^{-4})	0.729	(0.261)
G_{to}	0.02	0.0215	(2.72×10^{-3})	0.0453	(0.0725)	0.0252	(3.89×10^{-3})	0.658	(0.209)
G_{CaL}	1×10^{-4}	1.11×10^{-4}	(1.43×10^{-05})	0.101	(0.181)	1.22×10^{-4}	(2.39×10^{-05})	0.684	(0.238)
G_{Kr}	0.046	0.0467	(5.01×10^{-3})	0.0315	(0.0615)	0.0497	(4.11×10^{-3})	0.297	(0.0839)
G_{Ks}	3.4×10^{-3}	4.45×10^{-3}	(1.57×10^{-3})	0.0756	(0.110)	4.63×10^{-3}	(9.38×10^{-4})	0.729	(0.249)
G_{K1}	0.1908	0.192	(0.0171)	0.0400	(0.0501)	0.184	(0.0461)	0.765	(0.301)
G_{NaCa}	8×10^{-4}	1.04×10^{-3}	(2.98×10^{-4})	0.155	(0.251)	1.01×10^{-3}	(1.95×10^{-4})	0.779	(0.217)
G_{NaK}	30	39.8	(12.4)	0.0984	(0.167)	42.9	(4.95)	0.691	(0.117)
G_{Nab}	3.75×10^{-10}	5.04×10^{-10}	(1.38×10^{-10})	0.0564	(0.0804)	4.58×10^{-10}	(1.07×10^{-10})	0.804	(0.235)
G_{Kb}	3×10^{-3}	3.17×10^{-3}	(9.73×10^{-4})	0.189	(0.317)	3.72×10^{-3}	(8.18×10^{-4})	0.696	(0.158)
G_{pCa}	5×10^{-4}	6.664×10^{-4}	(2.01×10^{-4})	0.104	(0.122)	5.73×10^{-4}	(7.44×10^{-5})	0.642	(0.157)
G_{Cab}	2.5×10^{-8}	3.63×10^{-8}	(9.51×10^{-09})	0.111	(0.238)	3.31×10^{-8}	(4.22×10^{-9})	0.676	(0.212)

5 Free Parameters

Name	True Value	μ_P	(std.)	σ_P/σ_0	(std.)	μ_A	(std.)	σ_A/σ_0	(std.)
G_{Na}	75	75.0	(0.389)	0.0221	(4.05×10^{-3})	75.7	(0.0736)	0.0123	(7.32×10^{-4})
G_{Kr}	0.046	0.0461	(5.94×10^{-5})	3.14×10^{-3}	(1.31×10^{-3})	0.0458	(2.70×10^{-5})	0.0103	(2.37×10^{-4})
G_{pCa}	5×10^{-4}	6.07×10^{-4}	(1.46×10^{-4})	0.521	(0.216)	6.20×10^{-4}	(8.47×10^{-5})	1.04	(0.167)
G_{to}	0.02	0.0201	(2.45×10^{-4})	0.0449	(0.0210)	0.0204	(4.54×10^{-4})	0.193	(0.0184)
G_{Cab}	2.5×10^{-8}	2.92×10^{-8}	(5.17×10^{-9})	0.474	(0.267)	2.86×10^{-8}	(1.75×10^{-9})	0.978	(0.0887)

Columns marked "μ_P" and "σ_P/σ_0" denote the posterior mean and posterior-prior standard deviation ratio for each parameter when exact inference is performed, while columns marked "μ_A" and "σ_A/σ_0" denote corresponding quantities from the approximate sampler employing the five summary statistics described in Section 3.3. Each of these quantities is reported as the average over 10 repeated inferences, with corresponding standard deviations in parentheses.

Figure 3.20 All marginal distributions from the posterior produced by the probabilistic Del Moral algorithm for the five-parameter O'Hara-Rudy action potential model. Inference was performed using full time-course data. Generating values are shown as vertical red lines. All conductances are displayed in $mS/\mu F$.

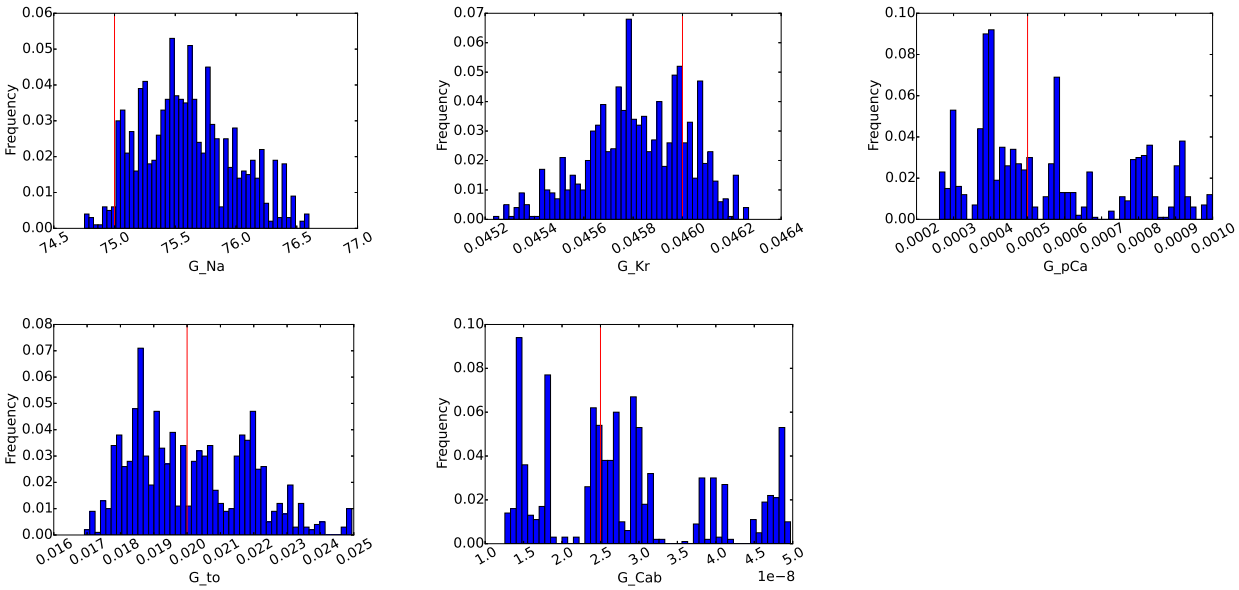


In order to control for under-determination in the system when summary statistics are employed (the number of parameters exceeding the number of data points), we repeated both fittings allowing only the five parameters best constrained by summary statistic data (as indicated by σ_A/σ_0 in Table 3.6 (top)) to vary, and holding all other parameters at their reported values. While G_{Na} and G_{Kr} were clearly the two best constrained, the next best three — G_{pCa}, G_{to}, and G_{Cab} — belonged to a class of parameters with roughly identical values of σ_A/σ_0, and thus their selection was somewhat arbitrary.

We see in Table 3.6 (bottom), as well as Figure 3.20 and Figure 3.21, that under these conditions there is better agreement between posterior mean and reported value for both exact and approximate inference, demonstrating the solver's increased ability to cover the parameter space in lower dimensions for a fixed number of particles.

In Table 3.6 (bottom), we see that relative to the posteriors on the 13-parameter model, the three major currents in this model – G_{Na}, G_{Kr}, G_{to} – show similar or decreased posterior uncertainty when time series data are employed, and dramatically reduced uncertainty when

Figure 3.21 All marginal distributions from the posterior produced by the approximate Del Moral algorithm for the five-parameter O'Hara-Rudy action potential model. Inference was performed using summary statistic data. Generating values are shown as vertical red lines. All conductances are displayed in $mS/\mu F$.



summary statistics are employed. This suggests that when compensatory interactions between model currents are removed, the approximate sampler in particular is better able to constrain these individual major currents. We notice, however, by comparing σ_A/σ_0 to σ_P/σ_0 in the five-parameter model, that G_{Kr} and G_{to} still exhibit a disproportionate increase in posterior uncertainty relative to G_{Na} when employing summary statistics, just as they did in the 13-parameter model. G_{Na} in fact exhibits a slight decrease in σ_A/σ_0 relative to σ_P/σ_0, though this is likely an artifact of the solver as the information contained in the summary statistics is necessarily a subset of the information contained in the full time trace.

Even when time trace data are available, however, it is relatively more difficult to estimate background currents such as G_{pCa} and G_{Cab} in the five-parameter model. In Table 3.6 (bottom), we see that the value of σ_A/σ_0 for these minor currents is much higher than that of the major currents, and actually shows an average-case increase in posterior uncertainty relative to the 13-parameter model. This may be due to the effects of noise added to the action potential, which causes the true maximally likely model parameters to differ slightly from the parameters used to

generate the data when only a finite number of observations are used during fitting. Thus, when we fixed 8 of the model parameters to the values from Table 3.5 (as opposed to the best values found when fitting to the full trace), we may have influenced the solver to explore a region of "flatter" likelihood for the parameters G_{pCa} and G_{Cab}, resulting in an increase in uncertainty. Because summary statistics were not shown to have significantly more information about these parameters than others in the 13-parameter model, it is unsurprising that the posteriors produced by approximate inference in this case are completely uninformative relative to the prior, as indicated by a σ_A/σ_0 of 1 in Table 3.6 (bottom). The degree of apparent constraint of these minor current parameters in the approximate posterior over the 13-parameter model was likely an artifact of under-determination of the system and an insufficient number of particles in the solver.

3.6 Discussion

In this chapter, we have presented two SMC samplers, generalized from popular ABC-SMC samplers to be capable of performing either exact or approximate inference on time-series models.

Through application of the approximate Toni sampler to the Hodgkin-Huxley action potential model, we demonstrated not only the process by which ABC techniques may be applied to cardiac models, but also the types of conclusions we can begin to draw from posterior inference. ABC posterior estimates for the parameters of the simplified Hodgkin-Huxley model were tightly constrained across most magnitudes of depolarization, indicating a high level of information content in the data collected. Not only this, but the mean estimates showed high homology with the values reported by Hodgkin and Huxley in their original publication. Such a well-constrained recovery of the original parameters highlights the remarkable degree of accuracy achieved by the authors without the aid of modern computational tools. Even when the posterior estimates of the ABC approach widened at low magnitude depolarizations, particu-

larly around the parameters controlling the sodium inactivation gate h, simulations employing the estimates of the original authors were able to reproduce the experimental data.

Performing ABC parameter inference on the *full* Hodgkin-Huxley model highlighted the value of the method's ability to quantify model uncertainty, and how such uncertainty may be propagated through the model in order to perform model validation. The wide posteriors around certain parameters in the potassium ($k_{\alpha_n 2}$, $k_{\alpha_n 3}$, and $k_{\beta_n 2}$) and sodium ($k_{\alpha_h 2}$) conductance components suggested an inability of the provided experimental data to constrain all model parameters, though a general inflation of posterior variance is expected when using ABC methods. This analysis suggests that additional data may be useful in further constraining the model, although without further experimentation we cannot rule out the large posterior variation being attributed to natural biological variability. To assess the effects of this variation, we examined the performance of the posterior parameterized model against several complex voltage protocols proposed by Hodgkin and Huxley at the end of their original paper. We noted that the particles of the sodium conductance posterior showed greater consistency in the response to these protocols, both internally and when compared to the response of the reported parameterization, than the particles of the potassium posterior. The failure of all particles in the posterior to produce the same behavior as the reported parameterization could be attributed to dependencies between the parameters of the two conductance models not captured by the separate fitting approach, over-fitting to voltage clamp data by our algorithm, or simply another constraint on the model not captured by the voltage clamp protocol. In either case, these results suggest additional experimentation performed to constrain the model might be focused on protocols which more fully probe the behavior of the potassium current.

Having demonstrated the application and interpretation of ABC posterior inference on time series data, we then moved to the more realistic case of performing posterior inference on summary statistic data, applying the approximate Del Moral sampler to the O'Hara-Rudy model. The modular specification of the Del Moral sampler additionally allowed us to perform exact inference on the corresponding time series data, which in turn allowed us to assess easily the

increase in posterior uncertainty resulting from employing said statistics. By examining the posteriors produced by approximate Bayesian inference over the summary statistics APD50, APD90, peak potential, resting potential, and maximum upstroke velocity, we found that these measurements contained the greatest amount of information about the conductances of the fast sodium (G_{Na}) and rapid delayed rectifier (G_{Kr}) currents, as measured by the relative width of their marginal posterior distributions after approximate inference. This makes intuitive sense, as fast sodium current activity is extremely localized to the rapid depolarization phase of the action potential, which is well characterized by maximum upstroke velocity and peak potential, while the rapid delayed rectifier current is involved in repolarizing the membrane in the later stages of the action potential, a phase which may be well characterized by ADP50, APD90, and resting potential. Other model currents whose phases of activity were not characterized well by the summary statistics, such as the slow delayed rectifier current G_{Ks} which describes a small repolarization of the membrane occuring after reaching peak potential but long before APD50 or APD90, showed very little decrease in posterior variability relative to the prior.

This suggests that a certain amount of domain specific knowledge can be employed by experimentalists when choosing the summary statistics to use in fitting currents within a model. However, as we have shown, once these summary statistics have been chosen, ABC methods should then be used to assess the extent to which these statistics can capture the information necessary to fit to all of the parameters of interest. In our case it is clear that the chosen set of five summary statistics contain sufficient information about only one of the parameters of interest, G_{Na}, as this was the only parameter that could be recovered to a precision similar to that achievable with the full data set. Care should therefore be taken not to over-interpret fluctuations in observed values of these statistics as being in any way indicative of changes in the underlying values of the conductance parameters other than G_{Na}, and to a more limited extent G_{Kr} and G_{to}.

Having presented posterior inference strategies that we believe are particularly suited to applications in cardiac modeling, and time-series modeling as a whole, we will next consider

how to interpret posterior distributions to draw conclusions about model identifiability. The techniques presented in the next chapter will not only aid the assessment of trust in model formulations given data, but also in the design of experiments that are maximally informative about a given model formulation.

4

Analyzing Identifiability

In the previous chapter we reviewed methods that may be used for posterior inference on biological models. We now discuss the manner in which the results of these methods may be interpreted to gain insight into model identifiability — a key step in building trust in a given formulation.

We introduce a systematic means of interpreting the results of posterior inference to determine model identifiability, as well as distinguishing between structural and practical unidentifiabilities. We additionally present two alternative means of assessing identifiability — an *a priori* approach based on Fisher information indices, and a novel *a posteriori* "measure theoretic" approach developed in part by our collaborator Simon Tavener at Colorado State University. We explore two models under a range of realistic experimental conditions, outline the situations for which each approach to identifiability analysis is most appropriate, and compare the results of applicable methods to make sure their findings are consistent. We conclude with a brief examination of automated methods to design experiments that maximize model identifiability, highlighting the challenges that these methods face in realistic biological and chemical contexts. This study serves as both an exploration of techniques that will be deployed in our Modeling Web Lab, as well as a guide on their correct application and interpretation.

The work presented in this chapter is under peer review in Daly et al. (2017b).

4.1 Introduction

One important aspect of modeling that we would like to improve through development of community resources is the standardization of identifiability assessment. As we have covered in Chapter 1, unidentifiability, which amounts to an inability to constrain one or more model parameters about a unique optimal value, weakens support for the biophysical interpretation of the model, making it harder to draw meaningful conclusions from its study. In order to avoid this problem, recent efforts in model development have been focused on automated methods to design experiments (controlling variables such as recording times, measurement types, and input stimuli) that maximize the identifiability of the model (see Section 2.3). In order to both aid in the assessment of model identifiability and the maximization thereof, we must provide the means to calculate metrics of model identifiability that are viable over a range of model formulations. Because biological models in particular are faced with numerous sources of error — from observation error to sample variation to random individual effects — some degree of variation over model parameters is expected, and thus simple metrics for quantifying variation like standard confidence intervals may not serve our purposes well. The metrics we employ must be able not only to detect the two types of unidentifiability, structural and practical, but *distinguish* between them, as different approaches may be required to address the arising issues (e.g., altering model formulation as opposed to refining experimental precision, as discussed in Chapter 1).

As we recall from Section 2.3, approaches to assessing identifiability can be divided into two classes: *a priori* methods based on analysis of model structure, and *a posteriori* methods based on the results of posterior inference. When model outputs are differentiable with respect to the parameters, the Fisher information matrix (FIM) can be used to quantify the amount of information about unknown model parameters one can expect to gain from a proposed experiment. As such, a wide body of literature exists detailing how functionals of the FIM can be used in the automated design of experiments whose outputs will be maximally informative about underlying

model parameters (Asprey and Macchietto, 2002; Cintron-Arias et al., 2009; Banks et al., 2011, 2013). In biological models, however, we may often encounter cases where the observable quantities are not differentiable with respect to model parameters, as in the case of the summary statistics of the cardiac action potential introduced in Section 2.1.4 and investigated in the previous chapter. Without access to gradient information, model identifiability can instead be determined by investigating the results of posterior inference methods like the ones described in the previous chapter. While these methods are generally more costly than *a priori* approaches, their generality and integration with the task of parameter fitting have lead to their proliferation in the field of biological modeling (see Section 2.3), and they may be employed as an alternative to FIM-based methods for identifiability analysis and experimental design (Chakrabarty et al., 2013).

In biological (or chemical) models, application of any of the approaches for identifiability analysis and model fitting may be complicated by the facts that, (i) experimental data are often very noisy, (ii) measurements may only be possible at a limited set of time points, (iii) only summary statistics of the data may be available (these are often termed "biomarkers" in physiological system modelling). Further, (iv) the model may contain many unknown parameters and (v) the full biological model may be expensive to compute. Any or all of the above may limit our ability to calculate the FIM or perform parametric inference. In the face of these concerns, it may be difficult to decide upon an appropriate method of assessing model identifiability, or to interpret the results of an identifiability assay in order to improve future modeling studies.

In this chapter, we explore two biologically relevant models under a range of observational conditions and demonstrate the application of specific *a priori* and *a posteriori* approaches for assessing (and distinguishing) model identifiability and designing principled experiments. The two models considered are the logistic growth model commonly used to represent bacterial proliferation, and the Hodgkin-Huxley action potential model described in Chapter 3. In contrast to previous approaches, we focus on gaining insight into the causes of unidentifiability from a geometric perspective, and introduce a novel method to this end based on a "measure-theoretic"

approach to inverse sensitivity analysis. This new approach can be considered as part of the family of posterior inference strategies, and we will compare its utility with that of two other *a posteriori* approaches, Markov Chain Monte Carlo (MCMC) and Approximate Bayesian computation (ABC), and with the perhaps more familiar *a priori* FIM-based approaches. We explore the ability of the Monte Carlo and measure theoretic inference techniques to determine which parameters of the Hodgkin-Huxley model can be effectively identified from a set of potential measurements, and to select the measurements that will maximize the identifiability of these parameters. Finally, we will explore the recommendations of FIM-based optimal experimental design algorithms (Banks et al., 2011; Cintron-Arias et al., 2009), and highlight problems that may arise when applying these techniques to realistic biological models. The inferential methods applied here consider simply the map from the input (parameter) space to the output space and make no assumption on the form of that map, giving them a wide range of applicability to different model systems. Our aim is for this study to serve as a guide to the selection and correct interpretation of identifiability metrics for nonlinear biological models, as well as an exploration of *a priori* and *a posteriori* methods for the design of maximally informative experiments.

4.2 Methods

In this section we will cover the mathematical background necessary to understand the calculation of several metrics of model identifiability that we will use throughout the chapter.

In general throughout our analyses we will be considering time-dependent systems $\boldsymbol{x}(t,\theta)$ with the state vector $\boldsymbol{x} \in \mathbb{R}^{N_x}$ and the system parameters $\theta \in \mathbb{R}^{N_\theta}$. The evolution of these systems will be described by systems of ordinary differential equations of the form:

$$\left.\begin{aligned} \frac{d\boldsymbol{x}}{dt} &= \boldsymbol{f}(t,\boldsymbol{x},\theta), \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0. \end{aligned}\right\} \tag{4.2.1}$$

We will also consider cases where we observe "quantities of interest" (QOIs) of the system state, rather than observing the state directly. These QOIs will be functions of both the solution

and the parameters, i.e.,

$$\boldsymbol{q}(t) = \boldsymbol{q}(t, \boldsymbol{x}(t, \theta), \theta) \tag{4.2.2}$$

where $\boldsymbol{q} \in \mathbb{R}^{N_q}$.

4.2.1 A priori methods: Fisher information

In this section, we detail the calculation of the FIM before we examine its utility in assessing model identifiability (Section 4.3) and guiding optimal experimental design (Section 4.4).

The sensitivity matrix. Before we can calculate the FIM, we must be able to calculate the *sensitivity matrix* of the model we are considering. The sensitivity matrix $\boldsymbol{M}$ defines the rate of change of the solution or the rate of change of quantities of interest derived from the solution, with respect to the parameters. In general, the sensitivity matrix is defined as

$$\boldsymbol{M} = \frac{\partial \boldsymbol{q}}{\partial \theta} \in \mathbb{R}^{N_M \times N_\theta}, \tag{4.2.3}$$

where

$$N_M = \begin{cases} N_x \text{ if using direct model outputs,} \\ \\ N_q \text{ if using quantities of interest.} \end{cases} \tag{4.2.4}$$

In general, in practice we will observe the system outputs y_i at a discrete set of times $t_i,\ i = 1, \ldots, N$, and we can solve the system equations (4.2.1) at each of these times to obtain $x(t_i)$. The sensitivity matrix $\boldsymbol{M}_N$ based on variables or quantities of interest at these discrete times, $\boldsymbol{t}_{obs} = \{t_1, \ldots, t_N\}$ is then of size $(NN_M \times N_\theta)$, where

$$\boldsymbol{M}_N = \begin{pmatrix} \left.\frac{\partial q_1}{\partial \theta_1}\right|_{t_1} & \cdots & \left.\frac{\partial q_1}{\partial \theta_{N_\theta}}\right|_{t_1} \\ \vdots & \vdots & \vdots \\ \left.\frac{\partial q_{N_M}}{\partial \theta_1}\right|_{t_1} & \cdots & \left.\frac{\partial q_{N_M}}{\partial \theta_{N_\theta}}\right|_{t_1} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \left.\frac{\partial q_1}{\partial \theta_1}\right|_{t_N} & \cdots & \left.\frac{\partial q_1}{\partial \theta_{N_\theta}}\right|_{t_N} \\ \vdots & \vdots & \vdots \\ \left.\frac{\partial q_{N_M}}{\partial \theta_1}\right|_{t_N} & \cdots & \left.\frac{\partial q_{N_M}}{\partial \theta_{N_\theta}}\right|_{t_N} \end{pmatrix}. \tag{4.2.5}$$

The Fisher Information Matrix. Once we have access to the sensitivity matrix, we may define the Fisher Information Matrix as

$$\boldsymbol{F} = \boldsymbol{M}^\top \boldsymbol{M} \in \mathbb{R}^{N_\theta \times N_\theta}, \tag{4.2.6}$$

which, like $\boldsymbol{M}$, is a function of model parameters θ.

As we will demonstrate in Section 4.3 the spectral properties of the corresponding $\boldsymbol{F}$ — namely, its rank, eigenvalues, and eigenvectors — can be interpreted to gain insight into the identifiability of a model with "true" parameters $\bar{\theta}$. In practice, however, we may access these properties without calculating $\boldsymbol{F}$ directly, saving computational time by performing all operations on the sensitivity matrix $\boldsymbol{M}$. Consider the singular value decomposition

$$\boldsymbol{M} = USV^\top, \quad U \in \mathbb{R}^{N_M \times N_M},\ S \in \mathbb{R}^{N_M \times N_\theta},\ V \in \mathbb{R}^{N_\theta \times N_\theta}. \tag{4.2.7}$$

We need to consider situations in which $N_M \geq N_\theta$ and $N_M < N_\theta$ and define N_S, the maximal possible rank of $\boldsymbol{M}$, as

$$N_S = \min\{N_M, N_\theta\}. \tag{4.2.8}$$

This is the maximum number of non-zero singular values of $\boldsymbol{M}$.

A simple calculation shows that the FIM and the $N_\theta \times N_\theta$ diagonal matrix $S^\top S$ are unitarily similar, meaning they share the same rank and eigenvalues of $\boldsymbol{F}$, typically labelled $\sigma_i,\ i = 1, \ldots, N_S$, are related to the singular values of the sensitivity matrix $\boldsymbol{M}$, labeled $s_i,\ i = 1, \ldots, N_S$ by

$$\sigma_i = s_i^2, \quad i = 1, \ldots, N_S. \tag{4.2.9}$$

Using these relations, we see that we may access the rank, eigenvalues, and eigenvectors of the FIM simply by performing singular value decomposition on the sensitivity matrix.

For the remainder of the chapter, we will assume that, once provided with a sensitivity matrix $\boldsymbol{M}$, we have access to the spectral properties of the corresponding FIM using the methodology described above.

4.2.2 A posteriori methods

For the purposes of this study, we examined two approaches to *a posteriori* inference to compare against the FIM-based *a priori* approaches: Monte-Carlo inference and the novel "measure theoretic" approach. As we have discussed the operation of Monte-Carlo methods for both Bayesian and approximate Bayesian inference in the previous two chapters, in this section we will simply detail the operation of the algorithms considered. When discussing the measure theoretic approach, we provide both an algorithm and a short theoretical background for its operation.

All code pertaining to ABC and MCMC analysis in this chapter was written (in Python) and executed by myself, while all code pertaining to the measure theoretic approach was written (in Matlab) and executed by Simon Tavener.

Monte-Carlo methods

When considering a deterministic model with a known, calculable likelihood function over its outputs, we may apply a Markov Chain Monte-Carlo (MCMC) algorithm for posterior inference over model parameters (see Section 2.2.2 for a review of these methods). For this study we employed the MCMC algorithm described in (Bardenet, 2013) that proposes steps in parameter space according to a multivariate normal (MVN) distribution Σ that is adaptively adjusted to attain an acceptance rate of $\approx 25\%$. Given a prior distribution over parameter space $\pi(\theta)$, a likelihood function $P(D|\theta)$ of the data D given parameters θ, and a starting point θ_0, it proceeds according to Algorithm 4.1. Within this algorithm, T_{burn} defines a "burn-in" time that allows the sampler to converge before samples are recorded, and is generally taken to be equal to $n/2$ where n is the total number of steps.

The parameters controlling the adaptation of the proposal distribution are the initial covariance matrix of the MVN, Σ_0, and the target acceptance rate. In this instance, we chose Σ_0 to be a diagonal matrix with (diagonal) elements defined to be 10% of the standard deviation of each corresponding parameter. This causes the algorithm to initially propose independent

Algorithm 4.1 Adaptive MCMC sampling

Parameters: n: number of steps, T_{burn}: burn-in length, θ_0: starting point, Σ_0: covariance matrix of the initial proposal distribution
Initialize $\mathcal{S} \leftarrow \emptyset$ and $\mu_0 \leftarrow \theta_0$.
for i from 1 to n **do**
  Sample θ^* from $\mathcal{N}(\theta_{i-1}, \sigma_{i-1}\Sigma_{i-1})$
  Calculate the acceptance probability $\rho = \min\left(1, \frac{P(D|\theta^*)\pi(\theta^*)}{P(D|\theta_{i-1})\pi(\theta_{i-1})}\right)$
  Sample u from $\mathcal{U}(0,1)$
  if $u < \rho$ **then**
    $\theta_i \leftarrow \theta^*$
  else
    $\theta_i \leftarrow \theta_{i-1}$
  end if
  if $i > T_{burn}$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \theta_i$
  end if
  $\mu_i \leftarrow \mu_{i-1} + (i+1)^{0.7}(\theta_i - \mu_{i-1})$
  $\Sigma_i \leftarrow \Sigma_{i-1} + (i+1)^{0.7}\left((\theta_i - \mu_{i-1})(\theta_i - \mu_{i-1})^\top - \Sigma_{i-1}\right)$
  $\sigma_i \leftarrow \exp\left[\log(\sigma_{i-1}) + (i+1)^{0.7}(\rho - 0.25)\right]$
end for
return $\mathcal{S}$

steps in each dimension of parameter space, with magnitude proportional to the magnitude of the corresponding parameter. The target acceptance rate was set to 25%, which is close to the asymptotically optimal acceptance rate for Metropolis-Hastings algorithms (Bedard, 2008). As the algorithm progresses, it maintains an internal "reference point" μ_i, which is initially defined to be the starting point ($\mu_0 = \theta_0$, which is in turn chosen randomly from the prior distribution in this instance but may alternatively be set to a known point of high likelihood), and gradually evolves to better reflect the current sampling point θ_i as the iteration proceeds. Differences between this reference point and the current proposed sample cause the covariance matrix Σ_i to change shape, biasing exploration in directions of high deviation. The algorithm additionally maintains a scale parameter for the MVN covariance matrix σ_i, initially set to 1, which grows as the Metropolis-Hastings ratio ρ differs from the target acceptance rate, thus promoting exploration of parameter space until a region of sufficiently high probability is found.

Because there are recognized concerns for the convergence of Monte Carlo methods in the

face of model unidentifiability (Raue et al., 2016), we have verified that all our MCMC chains terminate in "well mixed" conditions (i.e., demonstrating unbiased exploration about a mean value), and have repeated all our inferences to ensure consistency.

When a likelihood function is not known or calculable, as in the case of inference over summary statistics of the cardiac action potential model, we must instead employ an algorithm for approximate Bayesian inference. For the purposes of this study, we employ the approximate Del Moral algorithm described in Chapter 3, with all settings maintained unless otherwise specified in the text.

Measure theoretic method

An alternative approach to Monte Carlo samplers for inference on model parameters given probability distributions on the outputs, is the measure theoretic inverse sensitivity approach that was developed in Breidt et al. (2011) and Butler et al. (2012) for scalar valued functions (maps) and extended to vector valued maps in Butler et al. (2014a). Given a functional map from parameter space to output space, the measure theoretic algorithm seeks to create an inverse map, allowing one to propagate the values of a likelihood function (or uniform acceptance kernel, in the case of approximate inference) back to regions of parameter space called "equivalance classes." The result is a posterior distribution estimate like those produced by Monte-Carlo methods, though the algorithm operates in a deterministic manner by partitioning both parameter space and output space and repeatedly evaluating the forward map. Here we present a theoretical discussion of the measure theoretic approach, excerpted from Daly et al. (2017b), followed by an algorithmic description.

The Measure Theoretic Inverse Sensitivity approach assumes a (measurable) vector-valued map Q from an input (parameter) domain $\Theta \subset \mathbb{R}^n$ onto a range $\mathcal{G} = Q(\Theta) \subset \mathbb{R}^m$ where $m < n$. The inverse Q^{-1} is a map from $\mathcal{G}$ to a space $\mathcal{L}$ of *equivalence classes*, where each equivalence class is a set $x \in \Theta$ such that $Q(x) = y,\ y \in \mathcal{G}$. Under mild assumptions, $Q^{-1}(y),\ y \in \mathcal{G}$, is a manifold or *generalized contour* in Θ, the analog of an elevation contour on a topographic

map (approximation and convergence of $\mathcal{L}$ appears in Breidt et al. (2011); Butler et al. (2012, 2014a), but it is a theoretical device and need not be explicitly constructed).

The forward stochastic propagation problem assumes Θ is a measurable space $(\Theta, \mathcal{B}_\Theta)$, where $\mathcal{B}_\Theta$ is a σ-algebra, and poses a probability measure P_Θ on $(\Theta, \mathcal{B}_\Theta)$. This induces a probability measure on the measurable space $(\mathcal{G}, \mathcal{B}_\mathcal{G})$, where $\mathcal{B}_\mathcal{G}$ is a σ-algebra, as

$$P_\mathcal{G}(A) = P_\Theta(Q^{-1}(A)), \quad A \in \mathcal{B}_\mathcal{G},$$

which is commonly constructed through Monte Carlo sampling in Θ.

The inverse of the forward stochastic propagation problem assumes $\mathcal{G}$ is a measurable space and poses a probability measure $P_\mathcal{G}$ on $(\mathcal{G}, \mathcal{B}_\mathcal{G})$. The map Q induces a *unique* inverse probability measure $P_\mathcal{L}$ as $P_\mathcal{L}(A) = P_\mathcal{G}(Q(A))$ for $A \in \mathcal{B}_\mathcal{L}$. A Disintegration Theorem then provides a decomposition of any probability measure on $(\Theta, \mathcal{B}_\Theta)$ in terms of the unique inverse probability measure on $(\mathcal{L}, \mathcal{B}_\mathcal{L})$ and measures on each generalized contour. An Ansatz of uniform probability measures on each generalized contour (in the absence of further information), provides a unique inverse probability measure on $(\Theta, \mathcal{B}_\Theta)$.

An algorithmic representation of the measure theoretic approach to estimating posterior density is presented in Algorithm 4.2. We consider a deterministic map $f : \Theta \to \mathcal{G}$ from the input (parameter) space $(\Theta, \mathcal{B}_\Theta)$ to the output space $(\mathcal{G}, \mathcal{B}_\mathcal{G})$, and a known probability distribution $P_\mathcal{G}$ on $(\mathcal{G}, \mathcal{B}_\mathcal{G})$. The algorithm seeks to determine the corresponding probability distribution on $(\Theta, \mathcal{B}_\Theta)$.

For all analyses in this paper, we independently divided each dimension of parameter space into 100 equally sized partitions ($M = 10^{2N_\theta}$) and divided the output space $\mathcal{G}$ into 20 equal partitions across the observable range. A detailed convergence analysis of this algorithm is provided in Butler et al. (2014b).

Algorithm 4.2 Measure theoretic inverse sensitivity

Partition Θ into sets $\Theta_i, i = 1, \ldots, M$
Let $\theta_i \in \Theta_i, i = 1, \ldots, M$ be the centroid of Θ_i
Partition $\mathcal{G}$ in to sets $Y_j, j = 1, \ldots, N$
Let A be an $M \times N$ zero matrix
for i from 1 to M **do**
  Evaluate the map f at θ_i
  $A_{ij} = A_{ij} + 1$ iff $f(\theta_i) \in Y_j$
end for
Let $q_j = P(Y_j)$ for $j = 1, \ldots, N$
Then $p_i = A_{ij} * q_j / (\sum_{i=1}^{M} A_{ij})$
return $\{\Theta_i, p_i\}$

4.3 Identifiability Assessment

Having introduced three metrics that can be used to assess model identifiability, we will now apply these metrics to two model problems. We will detail in particular how these metrics may be used to distinguish between structural and practical unidentifiability, and how the conclusions derived under each metric compare against each other (when applicable).

4.3.1 Logistic model

We begin our investigation into model identifiability by examining the logistic growth model given below:

$$\left.\begin{aligned} \frac{dx}{dt} &= rx\left(1 - \frac{x}{K}\right) \quad t \in (0, T], \\ x(0) &= x_0\,. \end{aligned}\right\} \tag{4.3.1}$$

For the purposes of our study, we treat the initial population size x_0 as known, leaving us with two free parameters: the growth rate r and the carrying capacity K. In order to generate simulated data for study, we select "true" parameters $x_0 = 0.1$, $r = 0.7$ and $K = 17.5$, and set the final time $T = 30$. The solution to the model under these conditions is shown in Figure 4.1, along with properties of its sensitivity matrix which we will be discussing in the next section.

A priori identifiability

Before we can calculate the FIM, we must first formulate the sensitivity matrix for the logistic growth model (see Equation 4.2.6). Given that there are two free parameters, the sensitivity matrix evaluated at any single time point is the (1×2) matrix given below:

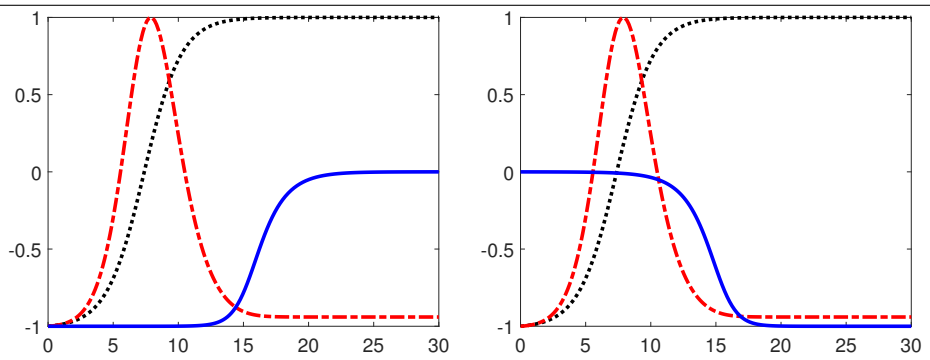
$$\boldsymbol{M} = \begin{pmatrix} x\left(1 - \frac{x}{K}\right) & \frac{rx^2}{K^2} \end{pmatrix}. \tag{4.3.2}$$

We can gain some of insight into the model by examining the properties of this sensitivity matrix. As we discussed in Section 4.2.1, the singular values and singular vectors of $\boldsymbol{M}$ are trivially related to the eigenvalues and eigenvectors of the FIM, which are often used to assess the *sensitivity* of the model — or, how changes in parameters correspond to changes in model output. The lead eigenvalue of the FIM is used to measure overall system sensitivity, while the corresponding eigenvector indicates the direction (in parameter space) along which the system is most sensitive.

In Figure 4.1, we plot the singular value of $\boldsymbol{M}$, along with the components of the associated singular vector, when the matrix is evaluated at times between 0 and 30. We see a peak in the leading (and only) singular value of $\boldsymbol{M}$ around $t = 7.9$, indicating that the system is maximally sensitive to parametric fluctuation near the point of maximal population growth. When measurements are taken at times $t \leq 10$, the singular vector (which is also the eigenvector corresponding to the single non-zero eigenvalue of the FIM) is oriented in the direction of the growth rate r in parameter space. For $t \leq 10$ the system is therefore sensitive to changes in the growth rate r, but largely insensitive to changes in the carrying capacity K. Conversely, for measurements taken at times $t \geq 20$, the singular vector of the sensitivity matrix is oriented in the direction of the growth rate K, and so the system is sensitive to changes in the carrying capacity K but largely insensitive to changes in the growth rate r. Both these conclusions are physically intuitive. Between $t = 10$ and $t = 20$ the singular vector rotates through $\pi/2$ and contains components of both parameters.

We next consider the identifiability of the logistic model under more realistic conditions,

Figure 4.1 Logistic solution, singular value and singular vectors of the sensitivity matrix. Left: Solution (black), singular value of the sensitivity matrix $\boldsymbol{M}$ (Equation 4.3.2) (red) and r-component of the corresponding singular vector (blue) Right: Solution (black), singular value (red) and K-component of the corresponding singular vector (blue). The logistic solution and singular value in both graphs have been re-scaled between [-1,1] for clarity of presentation alongside singular vector components.



where we collect multiple observations throughout the time trace. As we discussed in Section 4.2.1, when the system is measured at N discrete times $t_i,\ i = 1, \ldots, N$, the sensitivity matrix attains dimension $(N \times 2)$ and is given by:

$$\boldsymbol{M}_N = \begin{pmatrix} x\left(1-\frac{x}{K}\right)\Big|_{t_1} & \frac{rx^2}{K^2}\Big|_{t_1} \\ x\left(1-\frac{x}{K}\right)\Big|_{t_2} & \frac{rx^2}{K^2}\Big|_{t_2} \\ \vdots & \vdots \\ x\left(1-\frac{x}{K}\right)\Big|_{t_N} & \frac{rx^2}{K^2}\Big|_{t_N} \end{pmatrix}. \tag{4.3.3}$$

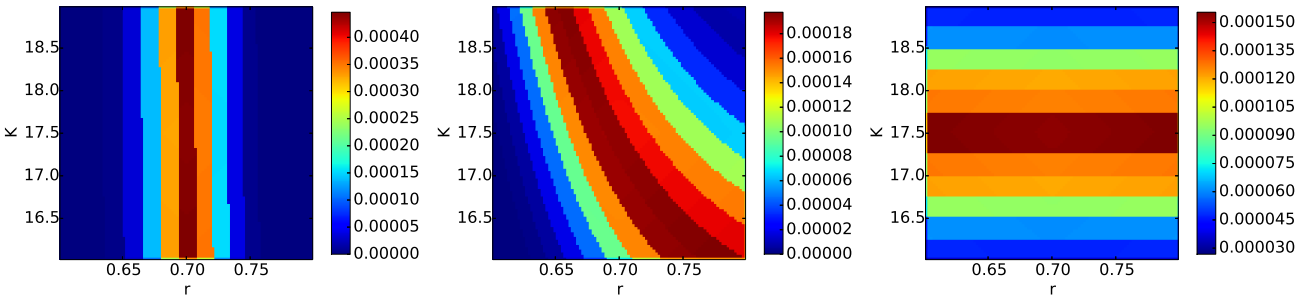
As the matrix has grown in dimension, it is more difficult to represent the spectral properties graphically. Instead, in Table 4.1 we present the singular values of the sensitivity matrix $\boldsymbol{M}_N$ for an increasing number n of uniformly-spaced sample points in the range $[0, 30]$. Since the singular value of the sensitivity matrix has a strong maximum at $t \approx 7.9$, indicating high system sensitivity, we also included the singular values based on the solution at three points chosen close to this maximum; this mirrors choices that may arise when using the optimal experimental design algorithms discussed in Section 4.4.

Table 4.1 Singular values of the sensitivity matrix M_N based on the solution at points $t_i,\ i = 1, \ldots, N$. Here s_1 and s_2 represent the first and second singular values. The ratio of these two singular values κ is the square root of the condition number of the corresponding Fisher Information Matrix.

N	$t_i,\ i = 1, \ldots, N$ (ms)	s_1	s_2	κ
1	12	7.6656	0	∞
2	12,24	7.6667	0.9921	7.7277
3	8,16,24	33.3612	1.4054	23.7378
6	4,8,12,16,20,24	34.6636	1.9094	18.1541
12	2,4,6,8,10,12,14,16,18,20,22,24	45.5863	2.5975	17.5501
3	7.899,7.900,7.901	57.8538	0.0028	20662.0714

Using this table, we may now investigate the effects of sampling point choice on two properties of the FIM that are often used to assess model identifiability: rank, and condition number. The rank of the FIM, which corresponds to the number of non-zero eigenvalues (or equivalently, the number of non-zero singular values in $\boldsymbol{M}_N$), indicates the structural identifiability of the system, with rank-deficient matrices indicating structural unidentifiability (Raue et al., 2009). The FIM is rank deficient, and thus the system is structurally unidentifiable, when fewer than two observations are taken. Given two or more observations, the FIM is full rank and the condition number of said matrix may inform us as to the *practical* identifiability of the system, with large values indicating practical unidentifiability (Dufresne et al., 2016). In Table 4.1, the ratio κ of the largest to smallest singular values is the square root of the condition number of the corresponding Fisher Information Matrix, and thus can be used to explore practical identifiability. In general, we see that κ decreases as the number of uniformly-spaced measurements in $[0, T]$ increases, indicating that more measurements generally increase practical identifiability. The small value of κ when the system is measured exclusively at $t_i = 12, 24$ is anomalous and may be explained by the fact that neither of these measurements fall near the point of inflection ($t \approx 7.9$), where the system is maximally sensitive. We also notice that clustering observations near $t = 7.9$, the time at which the singular value of the sensitivity matrix is maximized, results in a FIM with a condition number several orders of magnitude greater than that using the same number of uniformly spaced measurements, and suggests practical unidentifiability may result

Figure 4.2 Recovered input parameter distributions for the logistic model corresponding to an output that is normally distributed, centered at the population for the nominal parameter values and with a standard deviation equal to 5% of the nominal solution value for $t = 3, 10, 23$ (from left to right).



from exclusively measuring in this small area (this phenomenon will be discussed further in Section 4.4).

A posteriori identifiability

Having explored the use of FIM-based approaches for considering parameter identifiability for the logistic equation, we now consider the measure-theoretic and MCMC inferential techniques for doing the same. Throughout this section we will be operating in the presence of experimental noise which introduces uncertainty in the measured outputs.

We first consider application of the measure theoretic algorithm (Algorithm 4.2) on single measurements, where each simulated value of $x(t)$ is perturbed by Gaussian noise with mean zero and standard deviation equal to 5% of $x(t)$. In Figure 4.2, we present the resulting posterior distributions over K and r when these measurements were collected near the beginning of the trace ($t = 3$), the middle of the trace ($t = 10$), and the end of the trace ($t = 23$).

The conclusions we may draw about sample point placement from examining Figure 4.2 are consistent with those obtained from the previous *a priori* examination of the singular vector of the sensitivity matrix. At small times, the sets of equal posterior probability are parallel to the K axis, indicating that a single observation at small times enables us to assign different probabilities to different growth rates, but is completely uninformative about the carrying capacity K. At large times, the sets of equal posterior probability are parallel to the r axis, indicating

that a single observation at large times enables us to assign different probabilities to carrying capacities K, but is completely uninformative about the growth rate r. A single solution at intermediate times enables us to assign different probabilities to different combinations of r and K, and we see for the first time the notion of parameter compensation, whereby interactions between the parameters can lead to identical changes in the measured outputs, leading to functional relationships between K and r in regions of equal posterior likelihood.

We may construct similar visualizations by applying the MCMC algorithm (Algorithm 4.1) to compute the posterior distribution. We have assumed the following Gaussian error model for observations $y(t)$ for the logistic model:

$$y(t) = x(t) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 0.4), \tag{4.3.4}$$

and assumed uniform priors on both model parameters

$$\pi_K \sim \mathcal{U}(x_0, 200), \quad \pi_r \sim \mathcal{U}(0, 1). \tag{4.3.5}$$

We note that this error model, which assumes a independent Gaussian noise with fixed width, differs slightly from the proportional Gaussian noise model assumed in Figure 4.2. Both are reasonable to assume in biological systems, however, and the magnitude of noise added under each observation model were comparable in this instance.

Visualizations of posteriors produced by an MCMC sampler using the noise model above are presented in Figure 4.3. Multiple samples were collected from either the beginning ($t_i = 1, 2, 3$), middle ($t_i = 6, 7, 8$), or end ($t_i = 22, 23, 24$) of the trace, as well as when samples are collected from all three locations ($t_i = 2, 7, 23$). Here, we employed three time points for all sampling schemes so that we may compare the identifiability of the system when measured at three distinct regions in ensemble to the identifiability of the system when measured at each region exclusively while maintaining a consistent sample size. For all analyses, we used $n = 6,000$ steps for each sampler with a burn-in of $3,000$. These settings were chosen due to the fact that the algorithm showed unbiased sampling by the time 3,000 steps had been taken, and that

several thousand post-burn-in samples were needed to provide adequate visual characterization of the posterior.

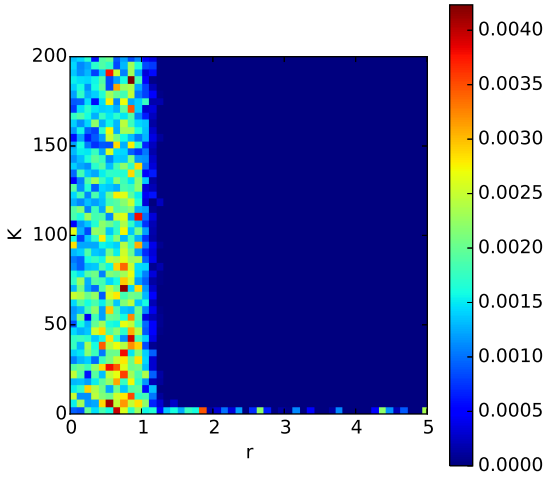
The MCMC posteriors in Figure 4.3 again show vertical banding when all measurements are taken at small values of t and horizontal banding when all measurements are taken at large values of t. When $t = \{6, 7, 8\}$, a set of values near the maximum of the singular value of the sensitivity matrix, we observe a distinct maximum of the posterior distribution near the true parameter values. However, we also observe that the posterior distribution is highly curved and extends to two extreme regions of parameter space. This suggests that collecting data only at this point allows for compensatory interactions between parameters, with different combinations of K and r producing outputs that are equally likely under the observation model (this mirrors our results from the measure theoretic approach when only considering data points in this span). Thus, while the model parameters could be uniquely identified under this observational scheme in the absence of noise, the presence of noise causes confidence regions to become increasingly large and non-elliptical. If we were concerned with the 90% confidence region of these parameters, we would conclude them to be practically unidentifiable in this instance. It is only when we observe the system at points taken from disparate phases of the curve ($t = \{2, 7, 23\}$) that we see an elliptical posterior indicating unique identifiability even in the presence of noise.

4.3.2 Hodgkin-Huxley model

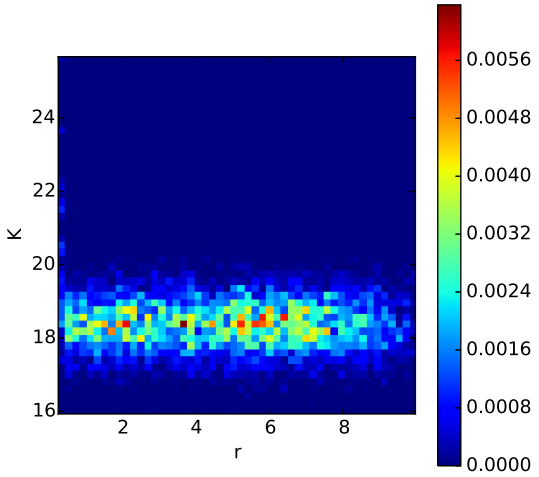
We next consider identifiability analysis of the Hodgkin-Huxley action potential model, originally introduced in Section 3.3.1. and summarized below:

$$\left.\begin{aligned} \frac{dV_m}{dt} &= \frac{-1}{C_m}\left(I_{Na} + I_K + I_l + I_{Stim}\right) \\ \frac{dm}{dt} &= \alpha_m(V_m)(1-m) - \beta_m(V_m)m \\ \frac{dh}{dt} &= \alpha_h(V_m)(1-h) - \beta_h(V_m)h \\ \frac{dn}{dt} &= \alpha_n(V_m)(1-n) - \beta_n(V_m)n \end{aligned}\right\}, \tag{4.3.6}$$

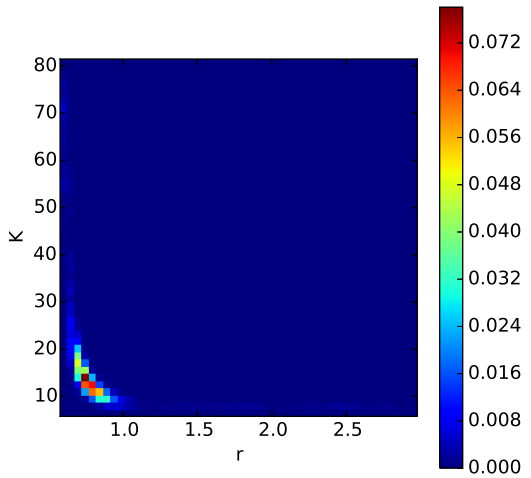
Figure 4.3 Posterior distributions of parameters of the logistic model recovered using MCMC when observed at $t_i,\ i = 1, \ldots, 3$ as indicated below each plot. Color bars to the right of each plot indicate the magnitude of posterior probability at each point. Differing scales are used for each axis to highlight the essential geometry of the posterior under differing identifiability conditions.



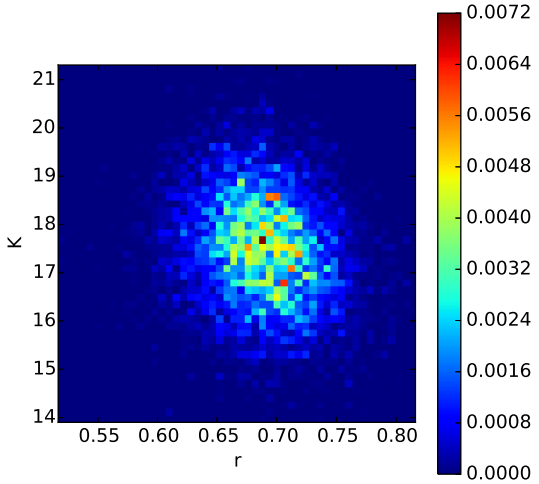
$t = \{1, 2, 3\}$



$t = \{22, 23, 24\}$



$t = \{6, 7, 8\}$



$t = \{2, 7, 23\}$

where

$$I_{Na} = G_{Na} m^3 h (V_m - E_{Na}),\ I_K = G_K n^4 (V_m - E_K),\ I_l = G_l (V_m - E_l),\ I_{Stim} = -20.$$

Unlike our original investigation of this model, where we fit the kinetic rate parameters using data on individual ionic conductances, we consider the rate parameters to be known and use action potential voltage recordings to fit the three maximum conductance parameters G_{Na}, G_K, and G_l (similar to our analysis of the O'Hara-Rudy model in the previous chapter). For the purposes of this study, we consider the "true" values of these parameters when generating data to be their originally reported values (given in mS/cm^2):

$$G_{Na} = 120, \quad G_K = 36, \quad G_l = 0.3, \tag{4.3.7}$$

and the known initial conditions are chosen to be

$$(V_0, m_0, h_0, n_0)^\top = (-75, 0.05, 0.6, 0.325)^\top. \tag{4.3.8}$$

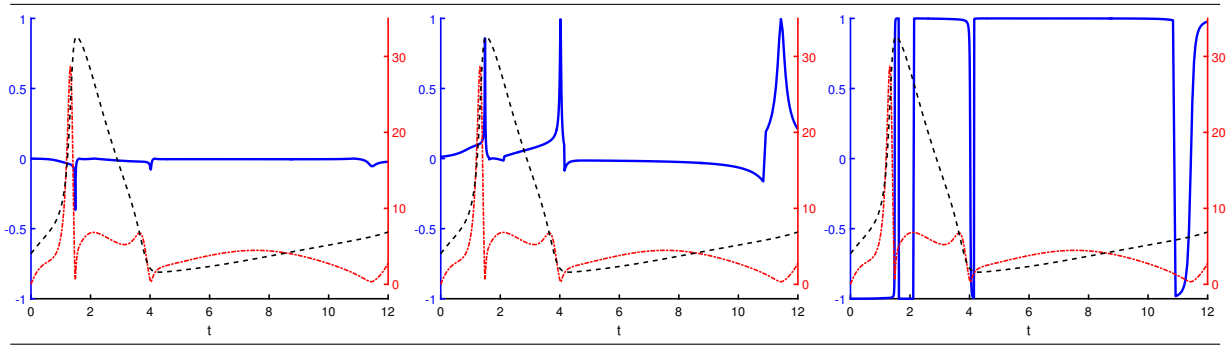
with all other parameters set to the values reported in Hodgkin and Huxley (1952). The solution of the membrane voltage using the nominal parameter values between 0 and 12 milliseconds, after which the membrane has become re-polarized, is shown in Figure 4.4.

A priori identifiability

We again start our analysis by considering the sensitivity matrix of the system. Whilst four state variables exist in the Hodgkin-Huxley model, the membrane voltage V_m is (generally) the only one that can be observed in practice, and we therefore consider only the effects of the parameters on this single output measure in this section.

As with the logistic model, we begin by exploring the solution, singular value, and singular vector of the sensitivity matrix when evaluated at single points $V_m(t)$ across the trace. In Figure 4.4, we note that the leading singular value of the sensitivity matrix is highest during the upstroke, indicating high system sensitivity at this point. Examining the components of the

Figure 4.4 Solution, singular value of the sensitivity matrix based on the membrane potential only and corresponding singular vector for the Hodgkin-Huxley model. The black dashed line in each figure is the membrane potential, the red chained line in each figure is the singular value of the sensitivity matrix, and the blue lines are the three components of the singular vector corresponding to the singular value, ordered as sodium, potassium and leakage currents from left to right. Scale of the singular vector components is indicated by the left (blue) axes, scale of the singular value is indicated by the right (red) axes, and membrane voltage is presented scale-free to highlight qualitative properties.



singular vector, we observe spikes in the components associated with sodium and potassium conductance parameters during the upstroke, repolarization, and recovery, but the singular vector is almost always dominated by the component associated with the leakage current. This initially surprising result can be understood by constructing the FIM and noting $m, n, h < 1$. Because the G_{Na} and G_K components of the FIM are high-ordered polynomial functions of these small variables, the constant-valued G_l component is dominant over most values, and thus controls the direction of the leading eigenvector. Phenomena such as this highlight the difficulties of interpreting the FIM directly even in simple systems such as this.

As with the logistic model, we now consider the identifiability of the model when multiple observations are collected. In Table 4.2, we record the values of the singular values of the sensitivity matrix, as well as its condition number, for an increasing number of N uniformly spaced measurements in the range $[0, 12]$. As in the case of the logistic model, we notice that the Hodgkin-Huxley model is structurally unidentifiable when the number of measurements is smaller than the number of parameters ($N = 1, 2$), as indicated by rank-deficiency of the sensitivity matrix, and we also note that the condition number κ decreases as the number of measurements increases for $N > 3$, indicating increasing practical identifiability.

Table 4.2 Singular values of the sensitivity matrix M_N for the Hodgkin-Huxley model taking measurements of the membrane potential at $t_i,\ i = 1, \ldots, N$.

N	$t_i,\ i = 1, \ldots, N$ (ms)	σ_1	σ_2	σ_3	κ
1	6	3.7904	0	0	∞
2	6,12	4.6819	0.5115	0	∞
3	4,8,12	5.2083	0.6541	0.0013	4006.3846
6	2,4,6,8,10,12	9.7274	0.7199	0.0465	209.1913
12	1,2,3,4,5,6,7,8,9,10,11,12	14.3055	0.9989	0.1123	127.3865

A posteriori identifiability: voltage measurements

We now turn our attention back to the inferential measure theoretic and MCMC methods for assessing identifiability. As it is easy to collect frequent measurements during an action potential recording, we begin right away with a study considering multiple data points simultaneously. For this analysis, we will be applying the MCMC algorithm to investigate the effect of the location of these data points on resulting model identifiability.

The prior distributions for each parameter are taken to be uniform between 50% and 200% of their nominal values (again in mS/cm^2),

$$\pi_{G_{Na}} \sim \mathcal{U}(60, 240), \quad \pi_{G_K} \sim \mathcal{U}(18, 72), \quad \pi_{G_l} \sim \mathcal{U}(0.15, 0.6). \tag{4.3.9}$$

Noisy observations of the membrane voltage were simulated by adding Gaussian noise to the computed voltage according to

$$y(t) = V_m(t) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 0.2). \tag{4.3.10}$$

We considered observations of the model employing three data points whose locations were chosen under two different schemes: (a) uniformly spaced points, $t_i = \{3.0, 6.0, 9.0\}$, and (b) local maxima of the singular value shown in Figure 4.4, $t_i = \{1.2, 2.3, 7.2\}$. One local maximum of the singular value occurs at the same time as the maximum rate of increase in membrane potential, or "maximum upstroke" ($t \approx 1.2$ms). Two other local maxima of the singular value occur during the "repolarization" period after peak membrane potential is reached ($t \approx 2.3$ms), and during the "recovery" period during which the membrane recovers its initial

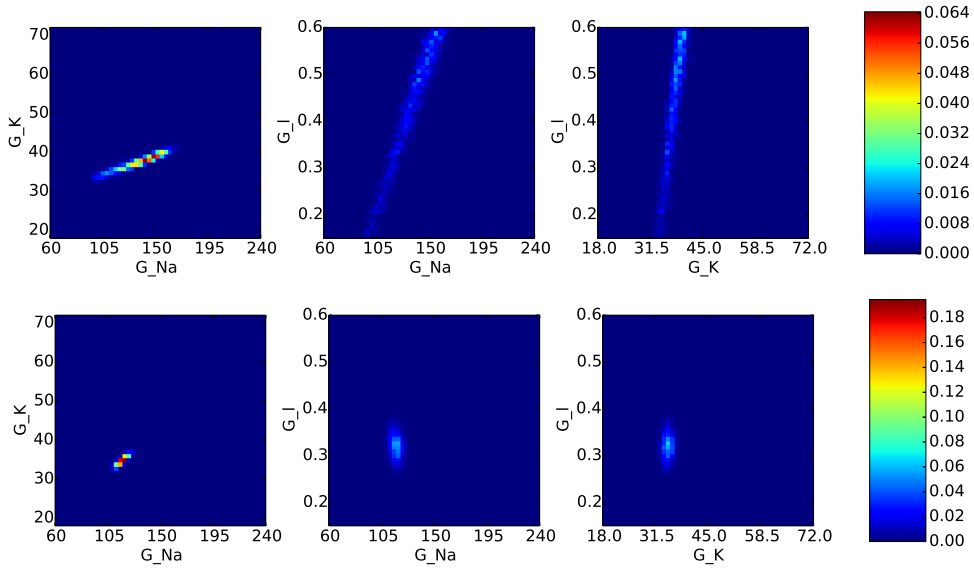
potential ($t \approx 7.2$ms). The sodium current is most active during the upstroke, when the cell rapidly depolarizes, while the potassium current is involved mainly in repolarizing the cell, and the leakage current is a catch-all for ion exchanges occurring close to resting potential. As such, we would expect measurements taken at upstroke, repolarization, and recovery to constrain G_{Na}, G_K, and G_l, respectively, and an ensemble of these measurements to constrain the system as a whole.

Posterior distributions, shown in Figure 4.5, were produced for each observation scheme using 40,000 MCMC steps with 30,000 discarded as burn-in. As with the logistic model, burn-in was chosen at a point after which MCMC samples were unbiased, and the number of samples taken beyond this point was chosen based on clarity of the visualized posterior. These settings were used for all subsequent MCMC analyses in this chapter. When the system is observed naively at uniform time intervals, the leakage current is unidentifiable as indicated by bands of nearly equal posterior likelihood spanning the entire prior range of the parameter. The posterior distributions for G_{Na} and G_K are broad due to the lack of a measurement during the upstroke period of the system. Using the measurements at $t_i = \{1.2, 2.3, 7.2\}$, which includes a measurement at the maximum upstroke, produces a significant decrease in the uncertainty of G_{Na}. Additionally, under this measurement scheme, G_l becomes identifiable, as indicated by a bounded elliptical posterior distribution in all parameter dimensions, although it does still exhibit greater posterior uncertainty than either of the other conductances.

A posteriori identifiability: summary statistics

In realistic experimental scenarios, however, the frequency with which we can record measurements is so great (multiple recordings within a single millisecond) that this large number of uniformly spaced measurements should be able to uniquely identify the system, as shown by the trend in Table 4.2. As such, it is less important to choose the location of these measurements in time. A more realistic case of limited observability, where we would be concerned with the design of an observation vector to maximally constrain the system, occurs when one

Figure 4.5 Biplot visualizations of MCMC posterior distributions for parameters of the Hodgkin-Huxley model when noisy observations are made at times $t_i = 3.0, 6.0, 9.0$ (top) or $t_i = 1.2, 2.3, 7.2$ (bottom). Axis ranges span the full uniform prior for each conductance parameter in mS/cm^2, and posterior likelihood is indicated by the corresponding color legend.



is forced to employ summary statistic data of the kind discussed in Section 2.1.4 and Chapter 3. If we consider the same five summary statistics: APD50, APD90, maximum upstroke velocity, peak potential and rest potential, we can no longer employ Fisher information indices, or indeed any Jacobian-based sensitivity metrics, because they are all non-differentiable functions of the membrane voltage V_m. We may still, however, employ both the measure theoretic inverse sensitivity approach and Bayesian or approximate Bayesian inference strategies (depending on the assumptions about model error) in this situation.

We begin by assuming a situation under which we are able to observe any or all of these summary statistics with a known noise distribution. For each statistic, we take this noise to be independent, zero-centered, and normally distributed with standard deviation equal to 1% of its value when calculated under reported parameters (see Equation 4.3.7). This relative noise model was chosen to correct for the differing magnitudes of the summary statistics in the calculation of the likelihood function, giving equal weighting to each observation. We compare the results from both measure theoretic and MCMC methods.

In Figures 4.6 and 4.7, we see the results of posterior inference using data from each sum-

mary statistic individually. As expected for such an underdetermined system, the posteriors produced by both methods demonstrate structural unidentifiability in the system, as evidenced by the unconstrained relationships between model parameters in the likelihood surface. Despite this, we can ascertain the information each summary statistic contains about each model parameter by examining the shape of the likelihood surface. The geometric relationships between parameters in the presence of parameter compensation can be seen very clearly, greatly aiding the intuitive understanding of the nature of the unidentifiability. We see that APD50, APD90, and peak potential each define a nearly linear relationship between G_{Na} and G_K, but do not provide enough information to uniquely identify either of these parameters or G_l. The maximum upstroke velocity provides a good deal of information about G_{Na}, as we observe tight banding in the likelihood along this axis, but very little information about the other two parameters. This matches our intuition, as the sodium current is the dominant factor controlling the upstroke.

By considering the geometry of the inverse sets corresponding to individual outputs, the methods provide insight into which combinations of outputs might be most informative. The timing/placing of any additional experimental measurements can then be tested *in silico* to determine whether, based on these geometric structures, they may potentially provide information to improve identifiability. This approach exploits the notion of *geometrically distinct* quantities of interest developed in Butler et al. (2014a). Geometrically, we can consider the contours corresponding to a two-dimensional quantity of interest as the intersection of the two surfaces corresponding to each of the (independent) scalar-valued quantities of interest. We may use this information in constructing a maximally informative vector of summary statistics to observe. In Figures 4.8 (measure theoretic inverse sensitivity) and 4.9 (MCMC), we see that when we combine summary statistics whose confidence regions were roughly overlapping in parameter space when observed separately, as with APD50 and APD90, we observe negligible decreases in uncertainty relative to independent observation. When we instead combine statistics whose individual confidence regions were more orthogonal, such as APD50 and maximum upstroke velocity, we see reduction in uncertainty to the point where we can uniquely identify G_{Na} and

Figure 4.6 Biplot visualizations of input (parameter) probability distributions for the Hodgkin-Huxley model calculated by the measure theoretic inverse sensitivity method employing a single summary statistic (indicated by labels above each triptych) observed with a known error model. Recovered probabilities for the input parameters are indicated by the corresponding color legend. All conductances are given in mS/cm^2.

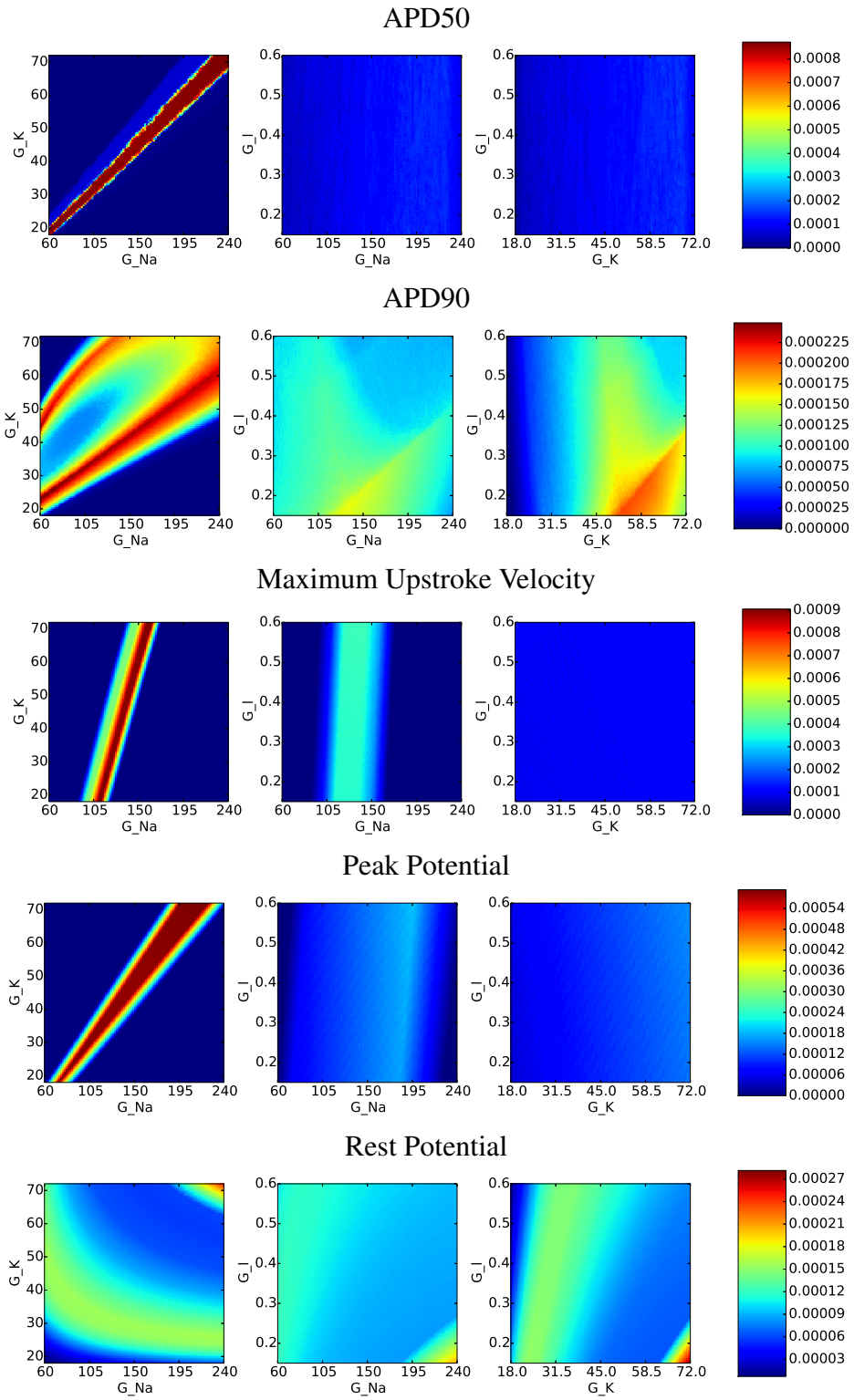


Figure 4.7 Biplot visualizations of input (parameter) probability distributions for the Hodgkin-Huxley model calculated by MCMC employing a single summary statistic (indicated by labels above each triptych) observed with a known error model. Axis ranges span the full uniform prior for each conductance parameter, and the posterior likelihood is indicated by the color legend. All conductances are given in mS/cm^2.

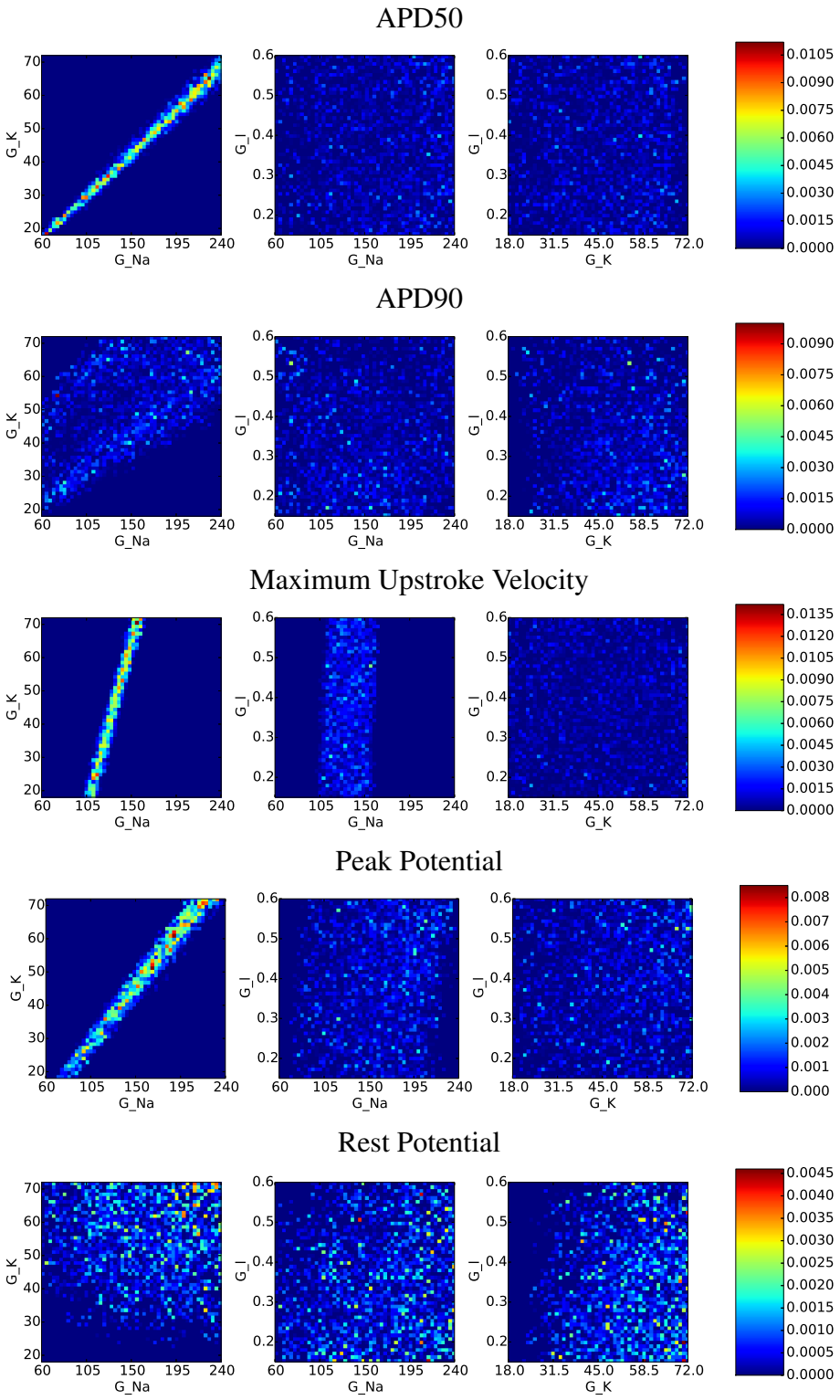
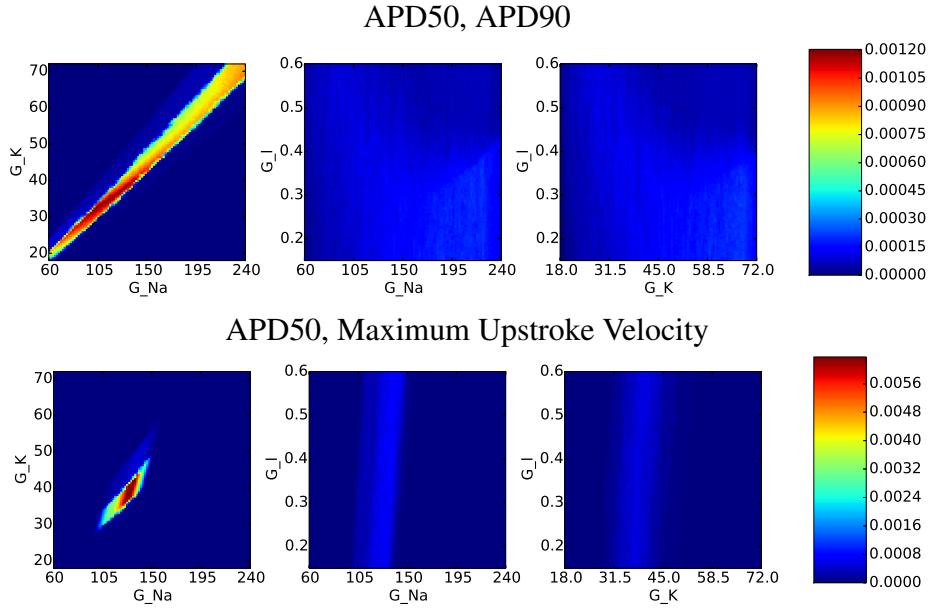


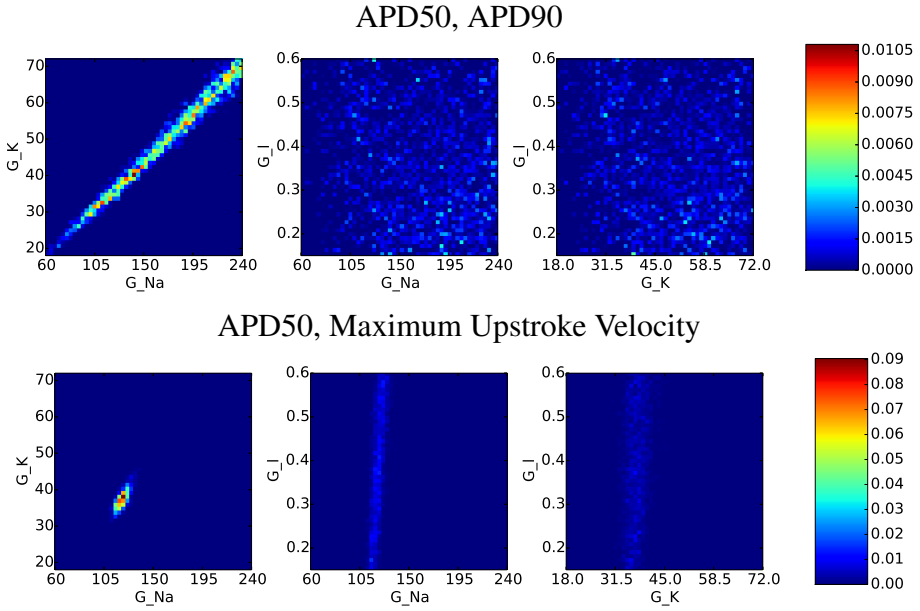
Figure 4.8 Biplot visualizations of input (parameter) probability distributions for the Hodgkin-Huxley model calculated by the measure theoretic inverse sensitivity method employing two summary statistics (indicated by labels above each triptych) observed with a known error model. Recovered probabilities for the input parameters are indicated by the corresponding color legend. All conductances are given in mS/cm^2.



G_K, as indicated by the elliptical posterior likelihood. Note however, that if the contours corresponding to the two quantities of interest considered independently are each only weakly dependent upon the parameter G_l, their intersection will remain only weakly dependent upon the parameter G_l, and employing an ensemble measurement will not improve the identifiability of this parameter.

When considering three quantities of interest, we might expect the intersection of the contours associated with each measurement to produce a zero-dimensional object in parameter space, i.e., a uniquely identified point, provided the Jacobian of the input-output map is full rank. However, as we observed when employing two summary statistics, if the contours resulting from posterior inference on each individual summary statistic are weakly dependent on G_l, then the location of their intersection is highly sensitive to error. As a result, we found that no combination of three summary statistics from this set is able to constrain the leakage current, as these descriptors separately failed to constrain the parameter past its uniform prior likeli-

Figure 4.9 Biplot visualizations of MCMC posterior distributions for parameters of the Hodgkin-Huxley model when employing two summary statistics (indicated by labels above each triptych) observed with a known error model. Axis ranges span the full uniform prior for each conductance parameter, and the posterior likelihood is indicated by the corresponding color legend. All conductances are given in mS/cm^2.



hood. The posteriors produced when observing all possible combinations of three parameters are shown in Figures 4.10 and 4.11.

We conclude this section by an examination of these techniques when the error model on the observables is unknown. As we have mentioned in Chapters 2 and 3, summary statistics are often calculated from a voltage trace that has its own observational error, and it is difficult to predict how the (point-wise) error distribution over the voltage trace (generally assumed to be Gaussian) will propagate to the error distribution of the summary statistics. Figure 4.12 provides the distributions of the five summary statistics calculated from 1,000 independent noisy realizations of a Hodgkin-Huxley action potential (simulated using the noise model in Equation 4.3.10). These are clearly non-Gaussian and it is difficult to know the correct error model (likelihood function) to use when employing either the deterministic measure theoretic inverse sensitivity algorithm or MCMC algorithms for inference.

As in Chapter 3, under these conditions we may employ ABC methods to perform posterior

Figure 4.10 Biplot visualizations of input (parameter) probability distributions for the Hodgkin-Huxley model (Equation 4.3.6) calculated by the measure theoretic inverse sensitivity method employing three summary statistics (indicated by labels above each triptych) observed with a known error model. Recovered probabilities for the input parameters are indicated by the corresponding color legend. All conductances are given in mS/cm^2.

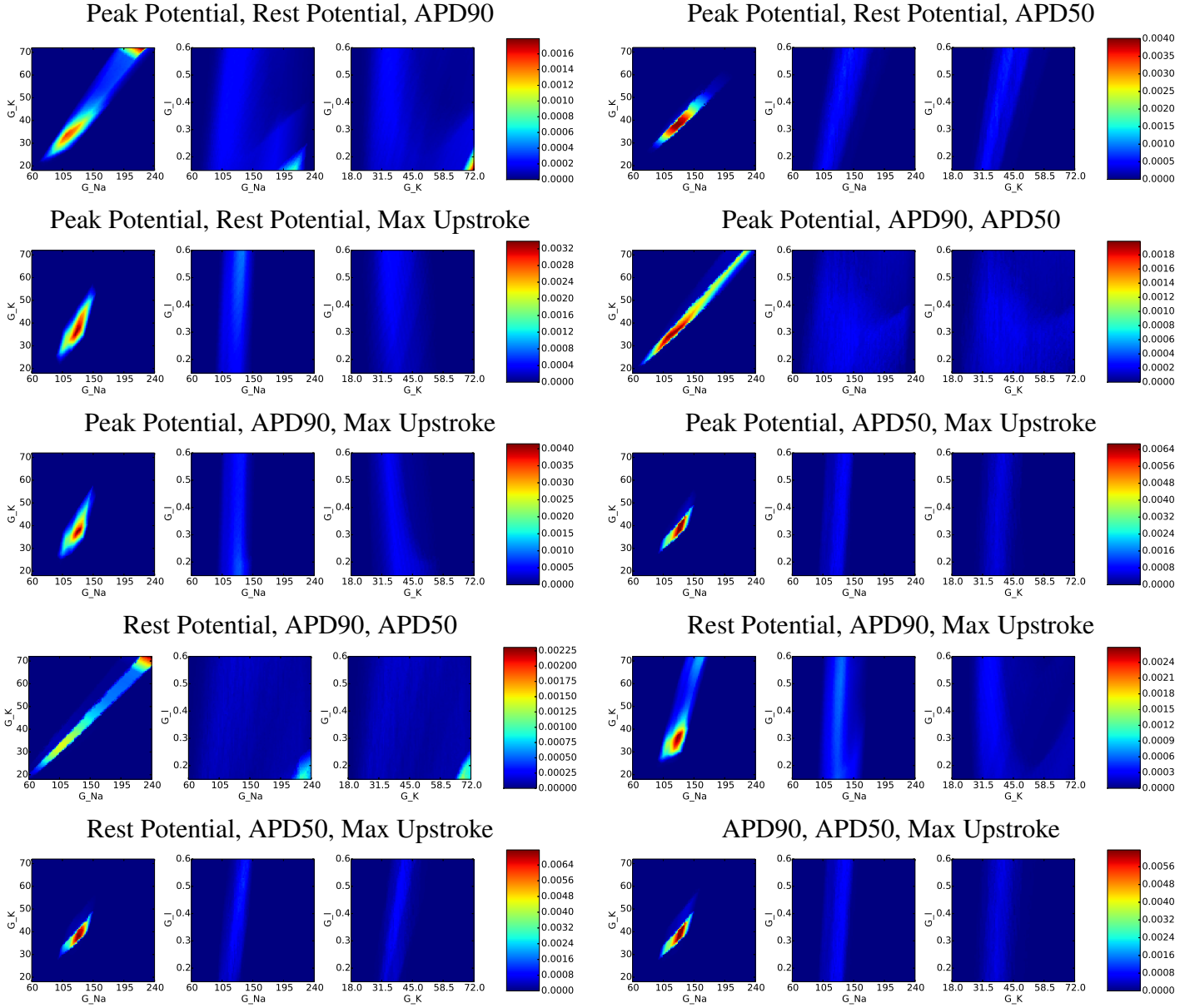
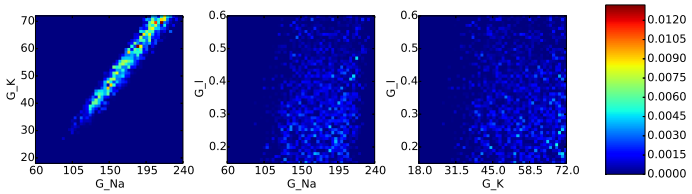
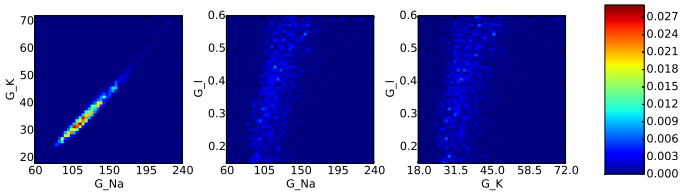


Figure 4.11 Biplot visualizations of MCMC posterior distributions for parameters of the Hodgkin-Huxley model (Equation 4.3.6) when employing three summary statistics (indicated by labels above each triptych) observed with a known error model. Axis ranges span the full uniform prior for each conductance parameter, and the posterior likelihood is indicated by the corresponding color legend. All conductances are given in mS/cm^2.

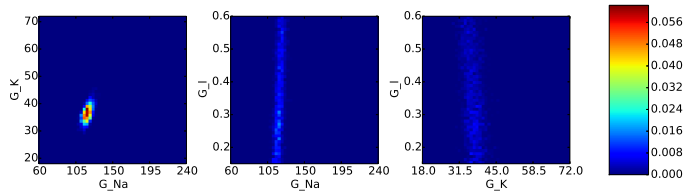
Peak Potential, Rest Potential, APD90



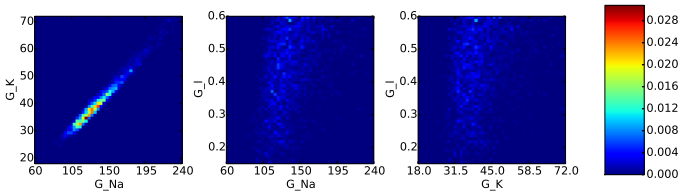
Peak Potential, Rest Potential, APD50



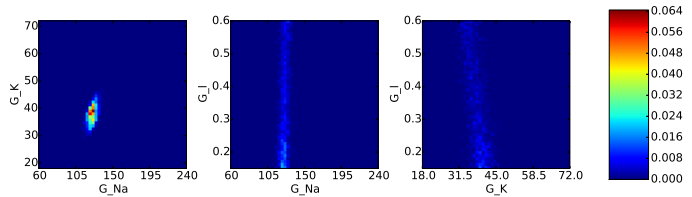
Peak Potential, Rest Potential, Max. Upstroke Velocity



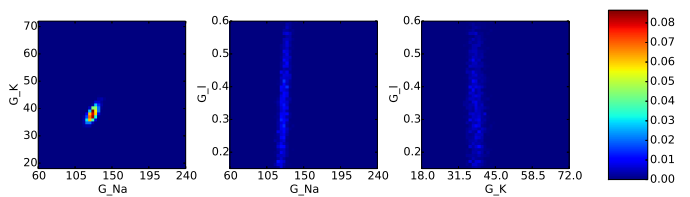
Peak Potential, APD90, APD50



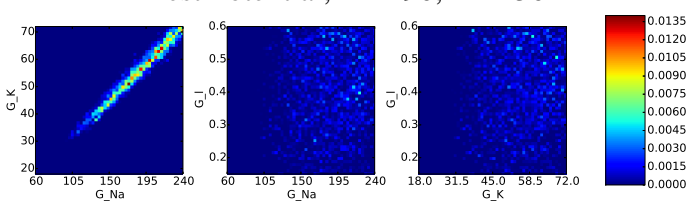
Peak Potential, APD90, Max. Upstroke Velocity



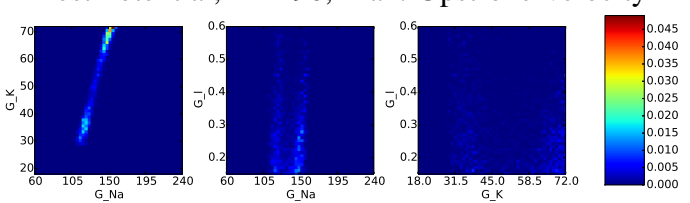
Peak Potential, APD50, Max. Upstroke Velocity



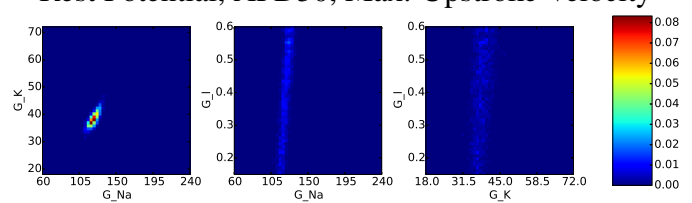
Rest Potential, APD90, APD50



Rest Potential, APD90, Max. Upstroke Velocity



Rest Potential, APD50, Max. Upstroke Velocity



APD90, APD50, Max. Upstroke Velocity

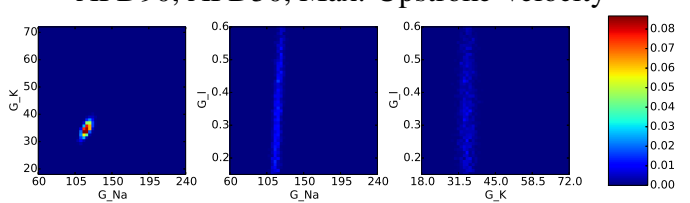
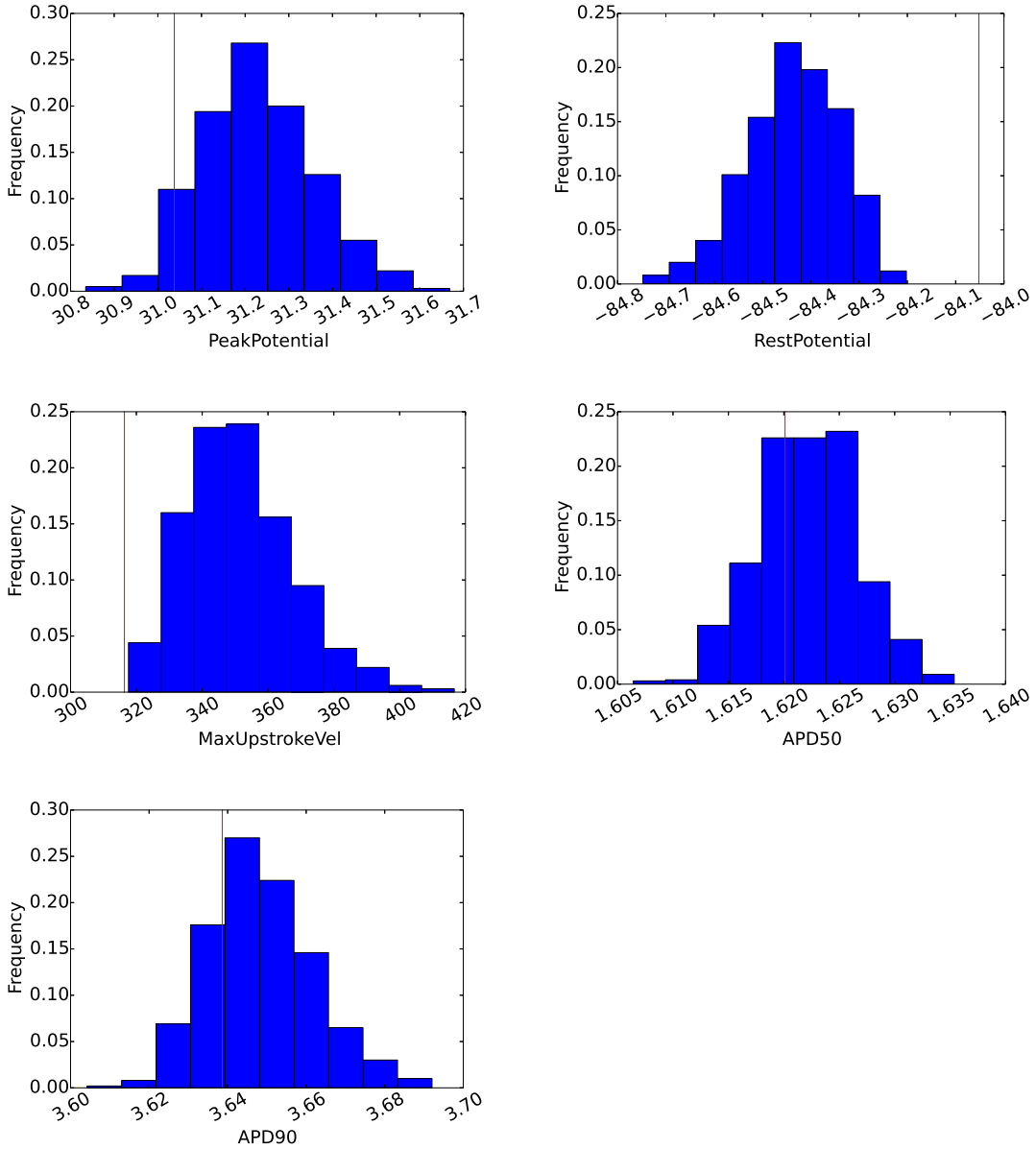


Figure 4.12 Distributions of the five summary statistics for the Hodgkin-Huxley model (Equation 4.3.6) computed using 1,000 independent realizations of an action potential trace with added Gaussian noise. Vertical red lines indicate the "true" value of the summary statistic when calculated from the voltage trace in the absence of noise.



inference, employing the following distance function:

$$\mathcal{D}(\mathbf{y}, f(\theta)) = \frac{1}{n_{ss}} \sum_{i=1}^{n_{ss}} \sqrt{\sum_{j=1}^{n_{obs}} \left(\frac{f(\theta)_i - \mathbf{y}_{ij}}{\mathbf{y}_{ij}} \right)^2}, \tag{4.3.11}$$

where $f(\theta)_i,\ i = 1, \ldots, n_{ss}$ are the values of n_{ss} summary statistics simulated using parameters θ, and $\mathbf{y}_{ij},\ i = 1, \ldots, n_{ss},\ j = 1, \ldots, n_{obs}$ are n_{obs} repeated experimental observations of the same statistics. The value of distance metric (4.3.11) fits naturally within the measure theoretic inverse sensitivity framework as a single new quantity of interest with a uniform distribution.

Figures 4.13 (measure theoretic) and 4.14, (ABC) provide the results of measure theoretic inference and ABC-SMC inference (Algorithm 3.3 using 1,000 particles as we did when performing summary statistic inference in the previous chapter), respectively, when using data from 25 repeated observations of APD50, peak potential, and rest potential (each calculated from an independent, noisy observation of an action potential) to infer model parameters. The tolerance for the above distance function within the ABC algorithm was arbitrarily set to $\varepsilon = 0.075$. Relative to the posteriors produced by exact inference using the same measurements, we see an increase in posterior uncertainty to the point where G_{Na} and G_K are once again unidentifiable. This indicates that, under these limited observation conditions, we would be able to gain very little insight into the "true" parameter values of the system, and should consider either an alternative observation strategy or a simplified model.

Figure 4.13 Biplot visualizations of input probability distributions for the Hodgkin-Huxley model determined using the measure theoretic inverse sensitivity approach using the distance metric in Equation 4.3.11 as a single quantity of interest. The distance metric incorporates APD50, Peak Potential, and Rest Potential. Recovered probabilities for the input parameters are indicated by the corresponding color legend. All conductances are given in mS/cm^2.

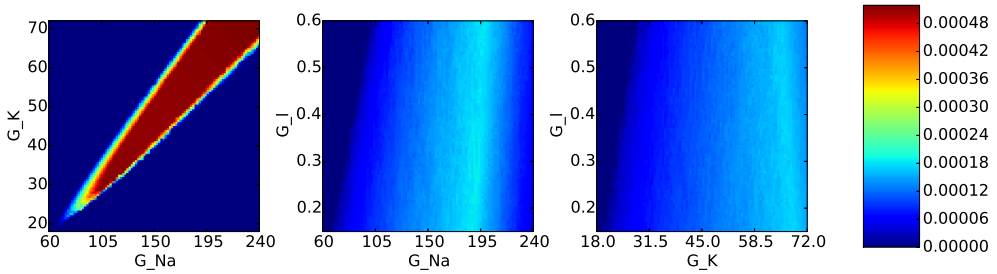
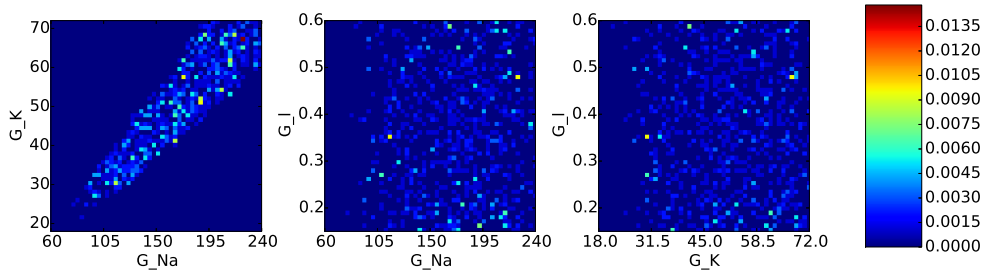


Figure 4.14 Biplot visualizations of ABC posterior distributions for parameters of the Hodgkin-Huxley model when employing the ensemble of APD50, Peak Potential, and Rest Potential calculated from *noisy* action potentials. Axis ranges span the full uniform prior for each conductance parameter, and the posterior likelihood is indicated by the color legend. All conductances are given in mS/cm^2.



4.4 Automated Experimental Design

Having discussed the means by which one may examine *a priori* and *a posteriori* identifiability metrics to choose optimal observation times by hand, we now turn our attention briefly to automated methods for doing so. Because these strategies generally rely on the repeated calculation of identifiability metrics under differing measurement scenarios, FIM-based metrics are often employed due to their low computational cost. For example, Banks et al. (2011) describe a number of strategies which seek to select observation times $t_i,\ i = 1, \ldots, N$ so as to maximize the sensitivity of the experimental observations to the parameter values. These are all based on the spectrum of the Fisher Information Matrix calculated about *assumed* values of the true parameters $\bar{\theta}$. SE-optimal design, D-optimal design and E-optimal design choose observation times $t_i,\ i = 1, \ldots, N$ so as to maximize the (normalized) trace, determinant and minimum eigenvalue of the FIM respectively. As a result, each of these schemes seeks to maximize identifiability as quantified by a slightly different metric: SE-optimal design minimizes the relative asymptotic error on each parameter, D-optimal design minimizes the volume of the confidence ellipsoid, and E-optimal design minimizes the principal axis of the confidence ellipsoid.

For the purposes of this study, we will consider the application of SE-optimal design to the logistic growth model (Equation 4.3.1). As such, we will be seeking observation points that

minimize the trace of the associated FIM, which is given by:

$$\boldsymbol{F} = \begin{pmatrix} x^2\left(1-\frac{x}{K}\right)^2 & \frac{rx^3}{K^2}\left(1-\frac{x}{K}\right) \\ \frac{rx^3}{K^2}\left(1-\frac{x}{K}\right) & \frac{r^2x^4}{K^4} \end{pmatrix}. \tag{4.4.1}$$

The method presupposes prior knowledge of the underlying model parameters $\bar{\theta}$, which are generally unknown in real-world models and must be constructed from an initial, minimally informed set of measurements. As such, for a fixed number of measurements ($N = 9$ in this case), we must first collect N measurements — uniformly spaced, as we are minimally informed about the system to begin with — then perform parameter inference using these measurements to construct an estimate θ^* of the "true" system parameters. We may finally employ this estimate to minimize the FIM over the N-dimensional space of potential measurements. In this study, optimization calculations were performed using the COBYLA constrained minimization routine in Python's SciPy module (Jones et al., 2001). Output least squares recovery of the model parameters was performed using the least squares minimization routine from the same module.

A final consideration when employing this algorithm is the enforcement of minimum distance between measurements. In realistic experimental scenarios, measurements cannot be taken arbitrarily close to each other, and thus the results of an unconstrained experimental design algorithm may be unrealistic. For the purposes of our study, we assume a minimal inter-point distance of $\varepsilon = 0.1$, and enforce this in one of two ways: either treating observations within this range as a single measurement, or adding constraints to the SE-optimal minimizer. As a result, observation times chosen according to the following strategies are shown in Table 4.3.

(a) Uniform: times chosen uniformly over $t \in (0, 25]$.

(b) SE-optimal: times chosen by SE-optimal design, then points within $\varepsilon = 0.1$ of each other treated as a single observation.

(c) Constrained SE-optimal: time chosen by SE-optimal design with an additional constraint requiring points to be at least $\varepsilon = 0.1$ apart.

Table 4.3 (a) Uniform, (b) SE-optimal, and (c) Constrained SE-optimal placement of nine sampling points for the logistic model.

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
(a)	0.000	3.125	6.250	9.375	12.500	15.625	18.750	21.875	25.000
(b)	7.717	7.717	7.717	7.717	20.611	20.611	20.933	20.933	24.998
(c)	7.567	7.667	7.767	7.867	20.955	21.103	21.374	22.840	24.958

The unconstrained SE-optimal strategy places four overlapping observation points close to the point of inflection, effectively reducing the number of measurements taken by three. This placement is a result of the strong local maximum of the singular value of M_N at $t \approx 7.7$, the inflection point of the growth curve. The constrained SE-optimal strategy places four minimally separated observations around this point. As we have seen in Table 4.1, highly clustered samples, even at a point of maximal sensitivity, can lead to practical unidentifiabilities, highlighting the dangers of using these automated methods without well-considered constraints.

In order to test the robustness of these sampling strategies in the presence of experimental error, we sought to recover the generating parameters using data collected at time points suggested by each approach. We generated simulated data with added independent Gaussian noise (as described in Equation 4.3.4) at the time points chosen under each sampling strategy, and recovered the nominal parameter values using ordinary least squares (OLS) minimization. The mean and standard deviations of the recovered parameters over 10,000 independent repeats of this recovery experiment appear in Table 4.4.

Table 4.4 Mean and standard deviation (in parentheses) of OLS inference applied to 10,000 noisy realizations of the logistic growth model sampled at time points chosen under three different schemes.

Sampling scheme	K_{OLS}	r_{OLS}
Uniform	17.50 (0.1815)	0.7000 (0.01178)
SE optimal	17.50 (0.2020)	0.7000 (0.0123)
Constrained SE optimal	17.50 (0.1790)	0.7000 (0.006266)

The unconstrained SE optimal method performs slightly worse than uniform sampling in reliably recovering both parameters, as quantified by the increased standard deviation about

the recovered values, which we might expect due to the effective reduction in sample size that results from the SE-optimal algorithm placing multiple samples at the same point in time. The constrained SE optimal method performs slightly better than uniform sampling for inferring K and much better at inferring r (nearly a 50% reduction in uncertainty about the mean estimate). This brief study highlights the potential usefulness of such algorithms for experimental design, but emphasizes the pitfalls that may result without constraining the algorithms to ensure realistic experimental conditions.

4.5 Discussion

By examining the logistic growth and Hodgkin-Huxley action potential models, we have demonstrated the ability of measure theoretic inverse sensitivity and Monte Carlo inference methods to capture identifiability information contained in the Fisher information matrix. Structural unidentifiability is indicated by rank-deficiency in the FIM, and may lead to unbounded regions of equal likelihood in the probability distributions on the input space. Practical unidentifiability, indicated by ill-conditioning of the FIM, is typically manifested as a functional (geometric) relationship between large regions of equal probability in parameter space. Two-dimensional projections of the probability distributions on the input space produced by measure theoretic inverse sensitivity and MCMC methods easily inform us as to which parameters are well-constrained and which are poorly constrained. While this can also be achieved by examining the direction of the eigenvectors of the FIM, such an examination becomes difficult in the high-dimensional systems we would expect in modern biological and chemical modeling. Thus, while posterior inference strategies incur additional cost relative to operations on the FIM, they do offer benefits in interpretability, and do not require knowledge of the "true" model parameters.

While the spectrum of the FIM can be examined to inform the collection of data that will maximally constrain the model, care must be taken to control against unrealistic recommendations. After we showed that choosing to measure both the logistic and Hodgkin-Huxley models

at points where they were maximally sensitive reduced overall uncertainty in the resulting posteriors, we also demonstrated the dangers of sampling exclusively at these locations. When automated means of experimental design such as the SE-optimal strategy sample aggressively at points of maximal sensitivity, subsequent efforts to recover model parameters were prone to greater error than when the model was sampled uniformly. Even though this can be corrected for by enforcing minimum distances between sampling points, these methods still presuppose knowledge of the true parameters of the system, which is unrealistic for most modeled systems. The congruence between these points of maximal system sensitivity and known regions of physiological interest (such as the point of inflection on the logistic growth curve and the point of maximum upstroke velocity on the action potential) suggests that system-specific knowledge may be employed both in the manual design of informative experiments, as well as in the adjustment of "optimal" sampling strategies to conform to realistic experimental limitations (such as minimum inter-sample distances).

When the model output is not differentiable and the FIM is not available, we have shown how both measure theoretic inverse sensitivity and Monte Carlo methods can still be applied to assess model identifiability, allowing them to be applied over a great range of model problems. When observation of the Hodgkin-Huxley model was limited to five summary statistics with known noise distributions, we demonstrated that, as expected, no one such statistic from the five we chose to examine (APD50, APD90, maximum upstroke velocity, peak potential, and rest potential) was sufficient to constrain the system. Ensembles such as ADP50 and maximum upstroke velocity were able to constrain both the sodium (G_{Na}) and potassium (G_K) conductance parameters. Furthermore, we demonstrated that these ensembles may be designed in a principled manner. Combining measurements such as APD50 and maximum upstroke that individually constrain the system in differing manners (as indicated by the shapes of the posteriors produced by measure theoretic or MCMC approaches when employing the measurements separately) could produce an overall decrease in uncertainty. Alternatively a poor choice such as APD50 and APD90, which individually provided very similar information, did not significantly

reduce uncertainty. This demonstrates the utility of inference-based methodologies, such as the measure theoretic and MCMC approaches, not only for assessing identifiability prior to experimentation, but also for the principled design of future experiments. While such design was performed in this study by manually constructing subsets of observations based on the geometry of the posterior distribution, and would be an intuitive and valuable process for any modeler, this could easily be automated by choosing subsets of observations with maximally divergent posterior information, as quantified by an inter-distributional distance metric such as Kullback-Leibler divergence, and optionally weighted by the relative difficulty (in terms of cost, time, or invasiveness) of making each recording. Finally, by demonstrating that no combination of three summary statistics was able to uniquely identify the leakage current conductance G_l, we have shown how a system may be unidentifiable in some parameters even when when employing a theoretically sufficient number of measurements. This suggests that, if limited to these five summary statistics, we cannot expect to gain meaningful information on G_l, and exemplifies how these computational tools can help to set and refine expectations of modeling studies.

We also examined the case in which the distribution of observation error for model outputs (summary statistics) is unknown. We demonstrated how the use of ABC, or the use of the measure theoretic inverse sensitivity algorithm when assuming a uniform distribution of output error, leads to an inflation of uncertainty about the parameters of the Hodgkin-Huxley model relative to when the noise model on its summary statistics is known. Under these conditions, observations such as the ensemble of APD50, peak potential, and rest potential were no longer able to uniquely identify any parameters of the model, suggesting that different or additional measurements would be required to constrain the model under these conditions.

Having detailed the application and interpretation of both *a priori* and *a posteriori* investigations into model identifiability to biologically relevant models, we will next consider a means by which these experiments, as well as the results of the posterior inference methods introduced in Chapter 3, may be described in a reproducible manner. The next chapter will present a framework language that is capable of representing a core set of the modeling techniques discussed

thus far, and is used as the basis for describing, executing, and disseminating the results of these analyses on a prototype version of the Modeling Web Lab.

5

Representing Reproducible Modeling Experiments

In the last two chapters, we have introduced and examined methods that can be used to parameterize models and assess identifiability — a key component in building trust in their formulation.

We now shift our attention to the manner in which such analyses may be described in a modular, reproducible manner that may be disseminated throughout a modeling community. In this chapter, we will present a framework we have developed for representing and executing modeling tasks within a Web Lab environment. We outline what we believe to be the essential entities within a modeling experiment, describe in detail the way in which we have abstracted these into components of our framework, and demonstrate how the framework may be employed to represent the types of analyses we have presented in the previous chapters.

5.1 Introduction

The methods we have introduced thus far for parameterizing and building trust in models (through the assessment of identifiability) are only truly useful in a community context if they can be represented in a reproducible manner. Models should rarely remain static, but rather evolve with the availability of new data, development of new algorithms, or the evolution of our understanding of the system. Thus, to get the most out of a model, it is important to document

not only a final parameterized form, but the process by which it was generated. Sufficiently unambiguous reporting should not only provide enough information to reproduce results in exact detail ("replication"), but also enough to independently deploy similar techniques in a potentially modified system ("reproduction") (Cooper et al., 2015). This would allow for both the independent verification of results and the reapplication of existing fitting strategies to new models or data (or vice versa) by other researchers working in the same problem domain.

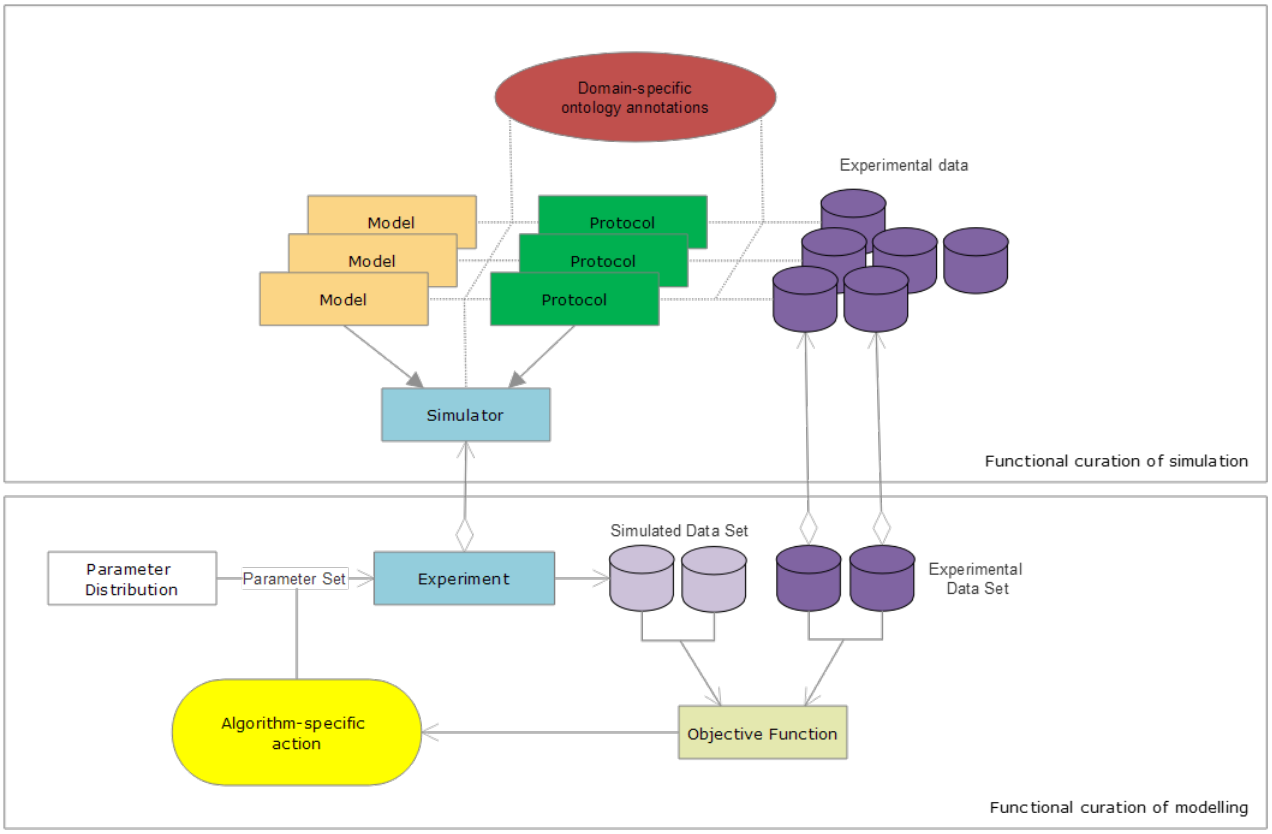
As we have noted in Chapter 1, such reporting practices could provide great benefit in the field of cardiac modeling in particular, where the devolving link between models and original experimental data has made it difficult to choose or adapt an appropriate model to represent a new system. Re-establishing this link by documenting how data are employed in fitting would allow for increased interpretability of the final model, as well as the application of uncertainty quantification methods such as Bayesian posterior estimation, which can inform us as to our degree of faith in the model's predictions.

While standards for reporting models and simulations such as CellML (Miller et al., 2010), SBML (Hucka et al., 2003), and SED-ML (Waltemath et al., 2011) have emerged, often with associated repositories to facilitate exchange thereof, the uptake of these standards is slow and far from universal, and these languages are primarily concerned with reporting parameterized models and single hard-baked simulations (Cooper et al., 2015). The functional curation extension to the Chaste simulation environment has taken this a step further, by separating the concepts of "model" (the underlying mathematical equations thought to represent the system, which may be specified in a standard format like CellML) and "protocol" (the manner in which the system is stimulated and recorded, which is defined in a custom language), whose interactions are mediated by an ontology as depicted in Figure 5.1 (top) (Cooper et al., 2011). This allows for the standardized representation of simulations, and the abstraction between components supports plug-and-play interaction, as well as reproduction of results in new experimental systems.

We aim to build upon the abstractions of functional curation, extending them from the sim-

ulation domain to the modeling domain. If functional curation gives us the ability to describe an "experiment" — the complete instructions for gathering data (real or simulated) from a system — in a modular and standardized manner, then we seek to analogously describe "modeling experiments", which we define as the complete instructions necessary to perform a variety of modeling tasks — such as parameter fitting, identifiability analysis, optimal experimental design, or model selection — for a system given observed data. Maintaining modularity in the components of a modeling experiment can be a difficult task, as differing model domains may have differing standards of representation, and the wide range of available algorithms that make use of model-specific information (such as Jacobian matrices) may hinder re-application to new systems.

Figure 5.1 Illustration of modelling framework entities, indicating data flow and interaction with the functional curation virtual experiment paradigm.



Some attempts have been made at creating standardized methods for representing and executing modeling experiments, particularly within the Python programming language. The "scipy.optimize" library (Jones et al., 2001) provides a variety of algorithms for minimizing arbitrary functions, which might easily wrap calls to simulation environments such as Chaste. Since this library only provides techniques that output a single optimal parameter set, however, it is incapable of characterizing uncertainty about these estimates beyond indices based on local curvature. Results are also difficult to interpret as routines of this library require the representation of simulation inputs as arrays. This may be confusing for biological systems with many parameters that may not be intuitively ordered, but may adhere to naming ontologies that can index parameters less ambiguously. The "lmfit" library (Newville et al., 2014) addresses the latter by allowing for dictionary-like representation of parameters, but still only provides only optimizing routines which output single points, which suffer from the same limitations as those in scipy.optimize. The "PyMC" (Patil et al., 2010) and "Stan" (Carpenter et al., 2017) libraries address these concerns, both allowing for the incorporation of prior information and uncertainty quantification through Bayesian inference (with Stan additionally allowing for approximate Bayesian inference). Both, however, require simulation code to be re-written in their own custom modeling language, which limits inter-operability with existing standards of representation, for which there already exist compliant, optimized simulation environments such as Chaste. Our goal is to address the limitations of these libraries for the biological modeling context, creating a framework for representing modeling experiments that can interface with existing ontologies and simulation environments and provide a means for Bayesian and approximate Bayesian inference over model and/or protocol parameters.

In this chapter, we present a Python framework that provides abstraction over key components of modeling studies: parameters and distributions, experiments, data, objective functions, and algorithms themselves (represented graphically in Figure 5.1, bottom). While the primary focus of this chapter is the description of parameter fitting experiments using this framework, as representing these types of experiments is the initial goal for our Modeling Web Lab, we will

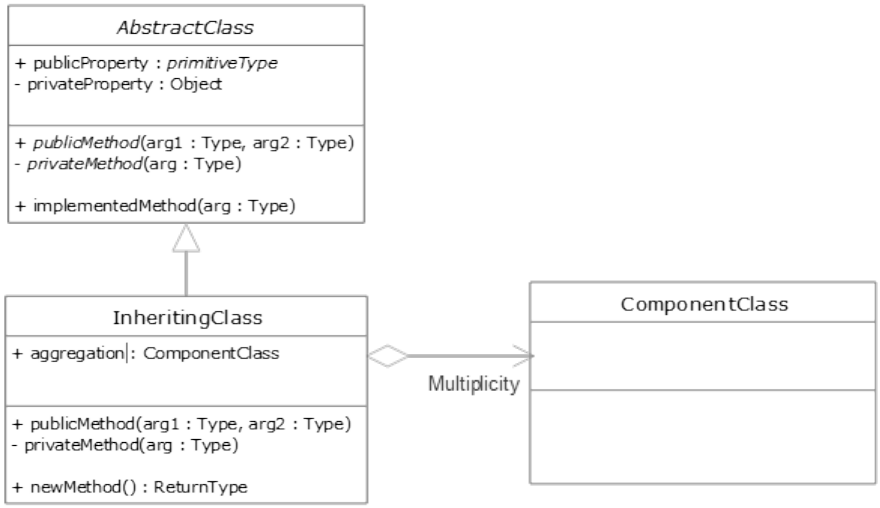
highlight how these abstractions may be applied to experiments in model selection and experimental design. For each entity, we begin with a qualitative discussion on its responsibilities and interactions with other framework entities, then present and discuss a technical specification making use of Unified Modeling Language (UML) diagrams (introduced in Section 5.2). As with the functional curation paradigm, if the abstractions presented in this chapter are respected, one can easily substitute components of a modeling experiment leading to highly re-applicable parameter fitting experiments. We believe that this framework establishes clear delineations of entity responsibilities, and thus serves as the basis for description and execution of modeling experiments in the prototype of our Modeling Web Lab, which will be discussed in the next chapter. We hope that this framework may one day serve as the basis for an implementation-independent language for representing general modeling experiments.

5.2 Reading UML diagrams

In this chapter, UML diagrams (Rumbaugh et al., 2004; OMG, 2015) will be used to graphically illustrate the classes used by the modeling framework, as well as the relationships between them. This graphical language, originally formulated in 1997, has become an industry standard, and will serve both as a clear and unambiguous representation of the framework architecture, and a launching off point for the justification of design choices.

While UML is capable of representing a wide range of computer science concepts and applications through the use of various diagram types (a full specification can be found in OMG (2015)), the primary focus of this chapter is to represent the classes that compose the modeling framework and their interaction, so we will largely limit our usage and discussion of UML to the UML Class Diagram. This diagram represents the structure of, and relation between, uninstantiated classes within an application. Figure 5.2 demonstrates the graphical and textual representation of several key concepts in class diagrams, which all subsequent figures in this chapter will adhere to:

Figure 5.2 Example UML diagram detailing the representation several key concepts in UML representation: abstract classes, inheritance, private/public properties/methods of a class, and aggregation.



- Each class is represented as a box composed of three rows, each containing specific information about the class. The top row lists only the class name. The second row lists the *properties* of the class — i.e., what an instance of the class "has." The third row lists the *methods* of the class — i.e., what an instance of the class "does."

- Class properties, and indeed all variables, are indicated by a name and a type, separated by a colon. Primitive types — those for which support is built in to Python, such as `int`, `float`, `bool`, `dict` — are italicized, while all other object types are not.

- Class methods are indicated by a name, a list of type-specified arguments in parentheses, and finally an optional return type, separated by a colon. If no arguments are listed, the function takes no arguments. If no return type is listed, the function returns `None`.

- A "+" preceding a class property/method declaration indicates it is "public," meaning it can be accessed by any object that has a reference to an instance of said class, while a "–" preceding a class property/method declaration indicates it is "private," and should only be accessed or changed by class methods. While Python does not strictly support

private and public properties, we indicate the functionality a class is meant to provide for other classes as public, while we indicate internal representations and helper methods as private.

- Abstract classes, which contain one or more unimplemented methods (abstract methods) or uninstantiated properties (abstract properties), are denoted by an italic class name. Abstract methods and properties are similarly denoted by italicized names. Abstract classes cannot be instantiated, but can provide full or partial implementations of some methods that will be inherited by classes that generalize them.

- Inheritance (or "generalization") is indicated by an open, triangularly-headed arrow pointing from the inheriting class to the base class. An inheriting class receives a copy of all the properties and methods of the base class, though it can choose to override them. A non-abstract class that inherits from an abstract class must provide an implementation of all the abstract properties and methods of the base class, as indicated by non-italicized declarations of the corresponding property/method.

- "Aggregation" — when one class contains one or more instances of another class as a property — is indicated by a double-ended connector between the participating classes. The container class is indicated by a diamond, while the component class is indicated by an arrowhead. The multiplicity of the aggregation — i.e., the number of instances of the component class within the containing class — is indicated by text near the connector arrow in the form `min .. max`, with "`*`" indicating "one or more."

Having presented this primer on interpreting UML diagrams, we will now proceed to discussions on the design and implementations of the essential entities of the modeling framework.

5.3 Experiments and Data

As illustrated in Figure 5.1, the concept of an "experiment", as we define it, encapsulates the concepts of "model" and "protocol" from the functional curation framework. Thus, its primary function is as a simulator — a means of producing structured data given an input set of parameters. In this section we will discuss the `Experiment` class, which serves as the basis for all simulators within our framework, and the `DataSet` class which is used to represent both the output of said simulations and any experimental data that may be provided by the user.

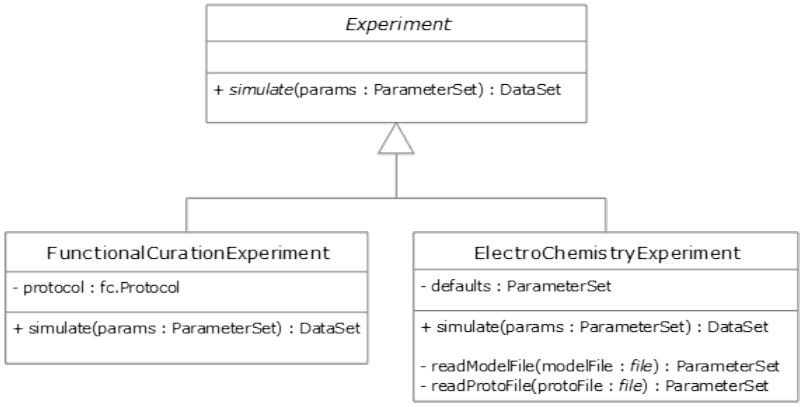
5.3.1 The `Experiment` Class

The sole function of the `Experiment` class is to accept parameters, in the form of a `ParameterSet` object, and output data, in the form of a `DataSet` object. For each modeling domain, such as cardiac modeling, a sub-class of `Experiment` should be created that is supplied with specifications of a model and protocol in a standardized form, and performs simulations under user-specified parameters using either custom code or calls to existing simulation libraries.

While classes extending `Experiment` may maintain an internal separation between model and protocol in the style of functional curation, they only advertise a "black box" simulation routine to other classes. The decision to seemingly "break" the abstraction of functional curation comes from a desire to maximize interoperability of the components of a modeling experiment. By revealing minimal system information, we allow for easy reproduction of modeling studies in new systems, under new conditions, or even in new domains of study. The implications of this decision on the types of algorithms that may be implemented in our framework will be discussed later in Section 5.6.2.

In Figure 5.3, we present the abstract `Experiment` class, which requires only a single method, `simulate()`, that produces a structured `DataSet` object for a given set of named parameters represented by a `ParameterSet` object. We additionally present examples of

Figure 5.3 UML diagram of the Experiment abstract class used as a base for all simulators in the framework, along with two implementing classes in the cardiac and electrochemical domains.



two implementing classes: the `FunctionalCurationExperiment` class, which handles simulations in the cardiac domain using existing standards of representation and simulation libraries, and the `ElectroChemExperiment`, which handles simulations in the domain of electrochemical redox reactions (which we will introduce in the next chapter).

The `FunctionalCurationExperiment` class provides an implementation of `Experiment` that interfaces with the Python implementation of functional curation. This class is instantiated with both a CellML file describing the model and a plain text file describing the functional curation protocol, which are parsed and maintained through a private instance of the functional curation `Protocol` class. The `Protocol` object maintains state for the model and protocol, which includes the values of all model parameters and protocol inputs. This allows for the partial specification of parameters (via a `ParameterSet` object) by the user, with all unspecified parameters defaulting to values specified in the model/protocol file. This in turn simplifies the fitting of subsets of model parameters, as the user need only specify priors over the parameters that are allowed to vary freely.

As we discuss in Section 5.4, our conception of a `ParameterSet` consists of a set of all parameters required for simulation, each associated with a unique name. Thus, the `FunctionalCurationExperiment` class needs to differentiate between model-specific

and protocol-specific parameters within the single set passed to it. Following the pattern used for references to model variables within functional curation protocols, which was in turn inspired by XML namespaces, parameter names within this set are formatted as colon-delimited strings with prefix and local name parts:

- Protocol parameters use the reserved prefix "`proto`," and should be named according to the form `proto:inputName`, where `inputName` corresponds to a variable declared in the "inputs" section of the protocol file.
- Parameters to the objective function are denoted by the reserved prefix "`obj`," and will be discussed in Sections 5.4.2 and 5.5.
- Other prefixes refer to model parameters, and by default use the same prefixes declared within the protocol file. These prefixes, as in the case of XML namespaces, abbreviate full URIs for the ontology term from which the parameter names are taken.

  Model variables mapping to these parameters are annotated with the corresponding full ontology terms using RDF. For example, the maximum conductance of the rapid delayed rectifier current would typically be referred to as `oxmeta:membrane_rapid_delayed_rectifier_potassium_current`, where `oxmeta` is declared in the protocol by:

  "`namespace oxmeta = 'https://...oxford-metadata#'`"

These naming conventions, in addition to unambiguously specifying the entities within the model and protocol specifications that are being altered, ease the reapplication of `ParameterSet` and `ParameterDistribution` objects to models employing the same naming ontology. While there is no comprehensive ontology for all of cardiac modeling, and indeed some model variables will be specific to certain formulations, the ontology developed for the cardiac electrophysiology Web Lab covers a broad range of common concepts, and should be respected when possible to give maximum generalizability to parameter fitting experiment specifications.

We additionally demonstrate the `ElectroChemistryExperiment` class, which implements `Experiment` for the domain of electron transfer reactions occurring at an electrode surface. While details of the formulation of these models and the protocols they are subjected to are discussed in Chapter 6, as electrochemistry experiments serve as an example domain for the Modeling Web Lab built upon this framework, we present the specification of the simulator class to show generalization of `Experiment` beyond the functional curation domain. Unlike `FunctionalCurationExperiment`, the `ElectroChemistryExperiment` class does not maintain model state between calls to `simulate()`, instead wrapping a call to custom ordinary and partial differential equation solvers for the electrochemical domain. To allow for the partial specification of parameters, as in the functional curation domain, we allow for the specification of a private set of default parameters whose values are employed for the solver unless otherwise specified in the input `ParameterSet`. This class of experiments additionally defines its own file formats for the specification of models and protocols, and thus provides private parsing support that is called upon instantiation.

5.3.2 The `DataSet` Class

In order to avoid ambiguity in modeling experiments, we must associate each simulation input or output with a unique identifier. As we are working in a primarily biological context, where entities are either direct measurements or simplifying approximations of biophysical entities, we may refer to such entities through the use of descriptive names. Thus, we have chosen to represent a `DataSet`, which aggregates all model outputs relevant to a fitting experiment, as a dictionary that maps names to arbitrary values. The names of these entities can distinguish between multiple outputs recorded from a single experiment (such as "membrane_voltage" and "membrane_sodium_current"), multiple repetitions of a single recording (such as "membrane_voltage1" and "membrane_voltage2"), or both, although no formal naming conventions in this regard have been adopted.

While most data considered will be numerical (either single values or n-dimensional arrays),

we do not make assumptions about the structure of the data, as experimental output may consist of an aggregation of measurements of different types. The implications of this general definition of data will be discussed in the later section on objective functions (Section 5.5), which operate over `DataSet` objects.

5.4 Parameters and Distributions

Within this framework, we define a "parameter" as any variable datum about a fixed-form model or protocol that influences the result of a simulation. We further define a "parameter set" to be a collection of all the parameters required for a simulation. These sets, represented by the `ParameterSet` class, will be the base units of input to the `Experiment` class. In this section, we will discuss not only the structure of these objects, but also introduce the corresponding concepts of "distributions," which represent variation about a single value and "parameter distributions," which represent variation over a set of parameters simultaneously. These concepts, represented by the `Distribution` and `ParameterDistribution` classes, respectively, are highly important for the representation of prior and posterior uncertainty in Bayesian and approximate Bayesian algorithms, and thus will serve as both our input to and output from parameter fitting algorithms.

5.4.1 The `ParameterSet` Class

As we have mentioned previously, we assume that all simulation inputs (and outputs) in our primarily biological context have a unique physical interpretation, and thus should be associated with names for clarity. Thus, as we have done for `DataSet`, we have chosen to represent a `ParameterSet` as a dictionary mapping parameter names to arbitrary values. The vast majority of model and protocol parameters in the cardiac domain may be represented as single numbers at a given time point (such as voltage, current, or kinetic rate); however, time-dependent or otherwise ordered inputs to a protocol, such as a sequence of voltage steps to apply during a clamp, may be better represented by vectors. Additionally, non-numerical

parameters such as the "model type" of the O'Hara-Rudy cardiac action potential model in the CellML repository may be better represented by informative strings, such as "endocardial"/"midmyocardial"/"epicardial", rather than arbitrary numerical indices. Thus, to allow for maximal interpretability and structural freedom, we have chosen to allow the representation of parameter values by arbitrary data types, and relegate the interpretation thereof to the simulator. This burden of interpretation does not break any abstraction between the concepts of `ParameterSet` and `Experiment`, as parameters are entirely simulation-dependent by definition.

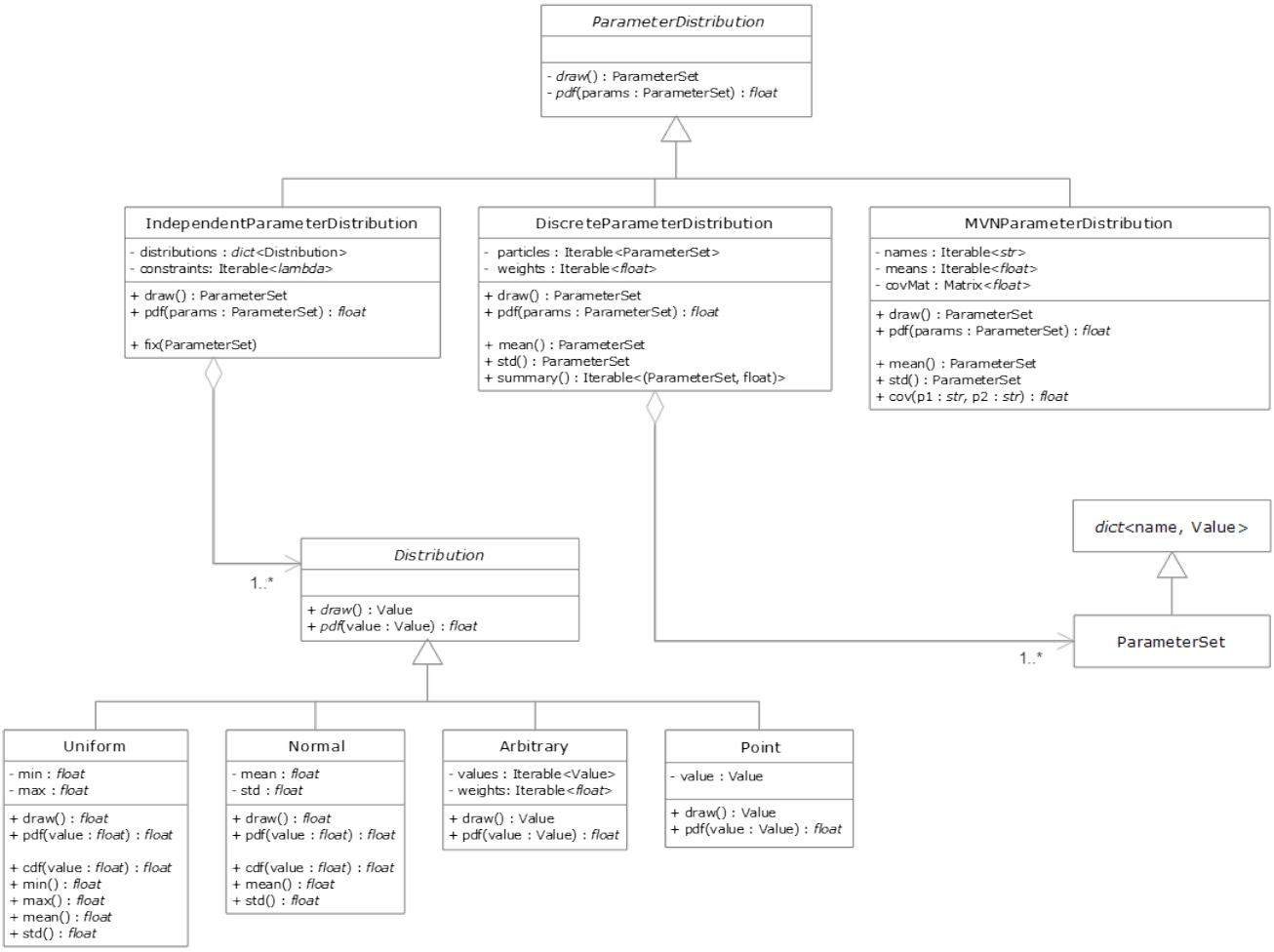
In Figure 5.4, we present the `ParameterSet` class, which extends a basic dictionary mapping names to arbitrary values. While we restrict indexing of a `ParameterSet` to the `string` type (the built-in Python `dict` type supports indexing by any immutable type, including numbers) in order to enforce unique and informative naming of parameters, we place no restrictions on the types of parameter values themselves.

5.4.2 The `Distribution` and `ParameterDistribution` Classes

The `Distribution` and `ParameterDistribution` classes provide methods that allow the user to fully characterize and quantify the variance about a value, and only differ in their scope of application. Instances of the `ParameterDistribution` base class operate on instances of `ParameterSet`, while those of the `Distribution` base class operate on instances of arbitrary values.

For each class, only two methods are required: drawing a sample (`ParameterSet` for a `ParameterDistribution` or arbitrary value for a `Distribution`), and evaluating the probability density/mass function (PDF/PMF) of a sample under the distribution. The "draw" method is sufficient to fully characterize the distribution with an appropriate number of calls, and the PDF/PMF is required for the evaluation of likelihoods during modeling studies. We have chosen this minimal interface to allow maximal freedom in specification of the underlying distribution. Requiring additional properties or methods such as "mean" would

Figure 5.4 UML diagram of the ParameterDistribution abstract class used to represent prior and posterior distributions in the modeling framework, the Distribution abstract class often aggregated by ParameterDistribution implementations, and concrete classes implementing each that cover common usage cases.



implicitly prevent the use of distributions for which they are not defined (e.g, the Cauchy distribution), or the use of non-numerical parameter values such as the previously mentioned "model type", which cannot be summed over. Classes implementing either `Distribution` or `ParameterDistribution` may provide additional properties or methods for algorithms to leverage, such as "min" and "max" for bounded optimization, but these must be strictly optional to ensure proper abstraction of algorithms from `ParameterDistribution` and `ParameterSet`.

In Figure 5.4, we represent the abstract base classes for both `Distribution` and `ParameterDistribution` objects, which contain the public methods:

- `draw():Object` — return a sample `Object` from the distribution,
- `pdf(o:Object):float` — evaluate the likelihood (from 0 to 1) of a given `Object` under the distribution according to the probability density function (or probability mass function, in the case of discrete distributions).

In Figure 5.4, we additionally show selected implementations of both `Distribution` and `ParameterDistribution` that cover common usage cases in the biochemical domain. In order to represent prior distributions over parameters, in which correlation is rarely assumed, we provide the `IndependentParameterDistribution` class, which contains a dictionary of `Distribution` objects and produces/operates on `ParameterSet` objects that are indexed by the same set of parameter names. We currently provide four implementations of the `Distribution` base class:

- `Uniform` — uniform variation of a single `float` value between `min` and `max`,
- `Normal` — Gaussian variation of a single `float` value, centered about `mean` with standard deviation `std`,
- `Arbitrary` – discrete variation over a list of (optionally) weighted possibilities,

- `Point` — a single value without variation.

The `IndependentParameterDistribution` class only provides `min()`, `max()`, `mean()`, `std()` methods if they are defined for each of its component distributions.

The `IndependentParameterDistribution` class also supports the specification of constraints between parameters as a list of boolean-valued functions over a `ParameterSet`:

- `Iterable<lambda(params:ParameterSet):bool>`

which are enforced both in `draw()` (by rejection sampling) and `pdf()` (by evaluating to zero if any constraints are not met). These constraints, along with the `fix()` method, which fixes the values of one or more parameters, allow the user to easily incorporate updated knowledge of the system into the modeling process.

In order to represent the output of samplers, which is generally a population of particles approximating the true posterior distribution over simulation parameters, we provide the `DiscreteParameterDistribution` class. This class simply aggregates one or more identically structured `ParameterSet` objects, with optional float-valued weights denoting relative importance. For ease of summarizing these distributions under the common case of fully numerical parameters, `mean()` and `std()` are provided, but these methods will fail in the presence of other value types. When these methods are inapplicable, the `summary()` method gives full information about the structure of the distribution in the form of tuples associating `ParameterSet` objects with floating point weights. This information can be leveraged to determine the empirical marginal distribution over parameter sets of any structure.

Finally, in order to encompass perhaps the most common case of structured correlation between simulation parameters — multi-variate normal — we provide the `MVNParameterDistribution` class. While this class maintains a matrix-based representation of parameter covariance internally, thus inherently limiting its use to fully numerical parameter spaces, it only interacts with the user via `ParameterSet` objects, maintaining abstraction from the underlying structure of parameter space. Thus,

while it is not as widely applicable as the previously presented implementations, an `MVNParameterDistribution` can still be easily substituted with another compatible generalization of `ParameterDistribution`.

These three implementations cover the most common cases of variation in parameter spaces within biological modeling, and we believe will be sufficient for most applications. Distributions over unusually formulated parameter spaces may still be handled by custom implementations of the `ParameterDistribution` abstract base class.

5.5 The `Objective` Class

An "objective function" quantifies the similarity (or dissimilarity) between two sets of data. Based on this definition, we introduce the `Objective` class of functions, which operate on two `DataSet` objects (plus any additional arguments) and return a single floating-point number as output. The scale and interpretation of this number depends on the function one is implementing (e.g, squared error vs. Gaussian likelihood) and the interpretation thereof is relegated to the algorithm employing this objective. While these functions will be used in order to evaluate the "quality" of parameters (by comparing data simulated under them to experimental data), we chose to have `Objective` perform direct comparisons between `DataSet` objects (rather than indirectly between `ParameterSet` and `DataSet`) to minimize coupling with the `Experiment` class.

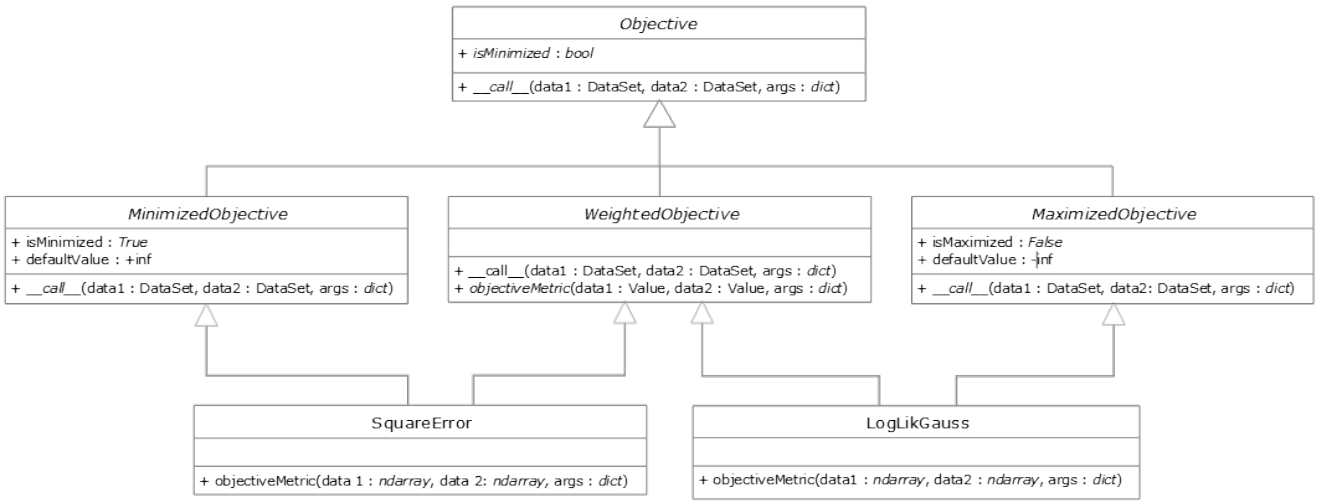
It is worth noting that the base `Objective` class places no restrictions on the structure of `DataSet` objects passed to it. The two `DataSet` objects may possess different numbers or types of entries, or even parallel entries (indexed by the same name in both objects) with differing value types. Despite this, all implementations of the `Objective` class that we have provided assume that parallel entries are of identical type and structure (e.g., two N-dimensional arrays of the same shape). In the functional curation framework, which served as our primary test bed, the protocol language provides a range of post-processing functions that

can transform simulation output to make it completely analogous to experimental data. For example, the cardiac-specific post-processing library provides functionality to calculate summary statistic data of the kind considered in Chapters 3 and 4 from full time-dependent voltage data. This highlights a decision to make the protocol responsible for exactly replicating experimental measurement, which we take into account by assuming identical structure between objects representing *in silico* and *in vitro* or *in vivo* data. Beyond this, custom implementations of `Objective` can be designed to compare inherently different quantities. One such example might be Mahalanobis distance (De Maesschalck et al., 2000), which quantifies the distance between a single quantity and a distribution over that quantity, as might be done between a single deterministic simulation and a repeated recording, or between a single recording and a repeated stochastic simulation. Such instances of variably repeated data would demand their own subclass of `Objective`, but do not break any of the paradigms outlined above due to the minimal assumptions about the structure of `DataSet` objects.

In Figure 5.5, we see our definition of the `Objective` abstract base class, which declares an abstract `__call__()` method and an abstract `isMinimized` property. Within Python, the presence of a `__call__()` method allows instances of a class to be called as functions, thus allowing for the conceptual maintenance of `Objective` as a function over two `DataSet` objects (and optional arguments). Defining `Objective` as a callable class rather than a traditional function allows for the inheritance of properties and methods that are common across families of objective functions, thus minimizing the amount of code that needs to be rewritten when defining a new function.

The advantages of inheritance can be seen in the classes extending `WeightedObjective`, a partial implementation of `Objective` that operates over identically structured `DataSet` objects. `WeightedObjective` implements `__call__()` as a weighted sum of comparisons between corresponding entries in `DataSet`, performed using the same abstract function `objectiveMetric()`, which operates on data of arbitrary type and returns a `float`. Weighting can be specified within the optional arguments in one of several ways:

Figure 5.5 UML diagram depicting the abstract base `Objective` class, three partial implementations thereof, and finally implementations of the commonly used square error (`SquareError`) and logarithm of Gaussian likelihood (`LogLikGauss`).



- `dict<outputName,float>` — Specify an individual weight for each output,
- `lambda(data:Value):float` — Define a common functional over all entries in the data set, such as inverse mean or inverse variance,
- Use a reserved keyword, such as "uniform" to specify common system-independent weighting schemes.

This allows for inheriting classes such as `SquareError` and `LogLikGauss`, which implement squared error and logarithm of Gaussian likelihood, respectively, to employ common weighting schemes while simply overriding the means by which data are compared (defined by the `objectiveMetric()` class method).

The callable class paradigm also allows for `Objective` functions to maintain properties such as `isMinimized`, which can be used by algorithms to automatically determine whether a given instance should be minimized or maximized. In Figure 5.5, `MaximizedObjective` and `MinimizedObjective` provide partial implementations of `Objective` that define this property, and thus extending these rather than the abstract base class directly removes the need for repetitive code in custom objective functions. These classes additionally define

a `defaultValue` property, which provides a function-specific "failure" value which might warrant special handling by the algorithm.

5.6 Tasks and Algorithms

We define two final concepts that are required for the execution of modeling experiments: a "task", which we define as the collection of all information required to specify a modeling experiment (analogous to a "Task" object in SED-ML (Waltemath et al., 2011)), and an "algorithm", which is a function that accepts a task (along with any additional arguments) and produces a final output.

Both the components required for specifying a task and the nature of algorithmic output depend on the type of modeling experiment being considered. As we defined it earlier in the chapter, a "modeling experiment" may encapsulate studies in parameter fitting, model selection, or experimental design. While these classes of studies are often related (a model selection study may involve repeated parameter estimation, for example), we have chosen to define specialized tasks and algorithms to represent each of the aforementioned types of modeling experiments due to their fundamentally distinct inputs and goals.

For our initial version of the Modeling Web Lab, we have chosen to limit our scope to the description of parameter fitting experiments, with a focus on posterior inference techniques. As such, we restrict our discussion in this section to the `ParameterFittingTask` and `ParameterFittingAlgorithm` classes that have been created for these types of modeling studies. In the concluding section of this chapter (Section 5.7), we will discuss how the framework as we have presented it thus far may be used to describe model selection and experimental design tasks in future iterations of the Modeling Web Lab.

5.6.1 The `ParameterFittingTask` Class

In order to specify a parameter fitting experiment, a `ParameterFittingTask` must contain the following information:

- `ParameterDistribution` — used to define prior knowledge of parameter space,
- `Experiment` — used to simulate data for a given `ParameterSet`,
- `DataSet` — used to represent fixed experimental data,
- `Objective` — used to compare experimental data to data simulated under a parameter set that is being considered.

Such collection of objects is equally capable of representing Bayesian/ABC inference, where the `ParameterDistribution` represents an informative prior, or simple optimization, where the `ParameterDistribution` represents bounds on each parameter (or $\pm\infty$ for unbounded optimization) through uniform distributions.

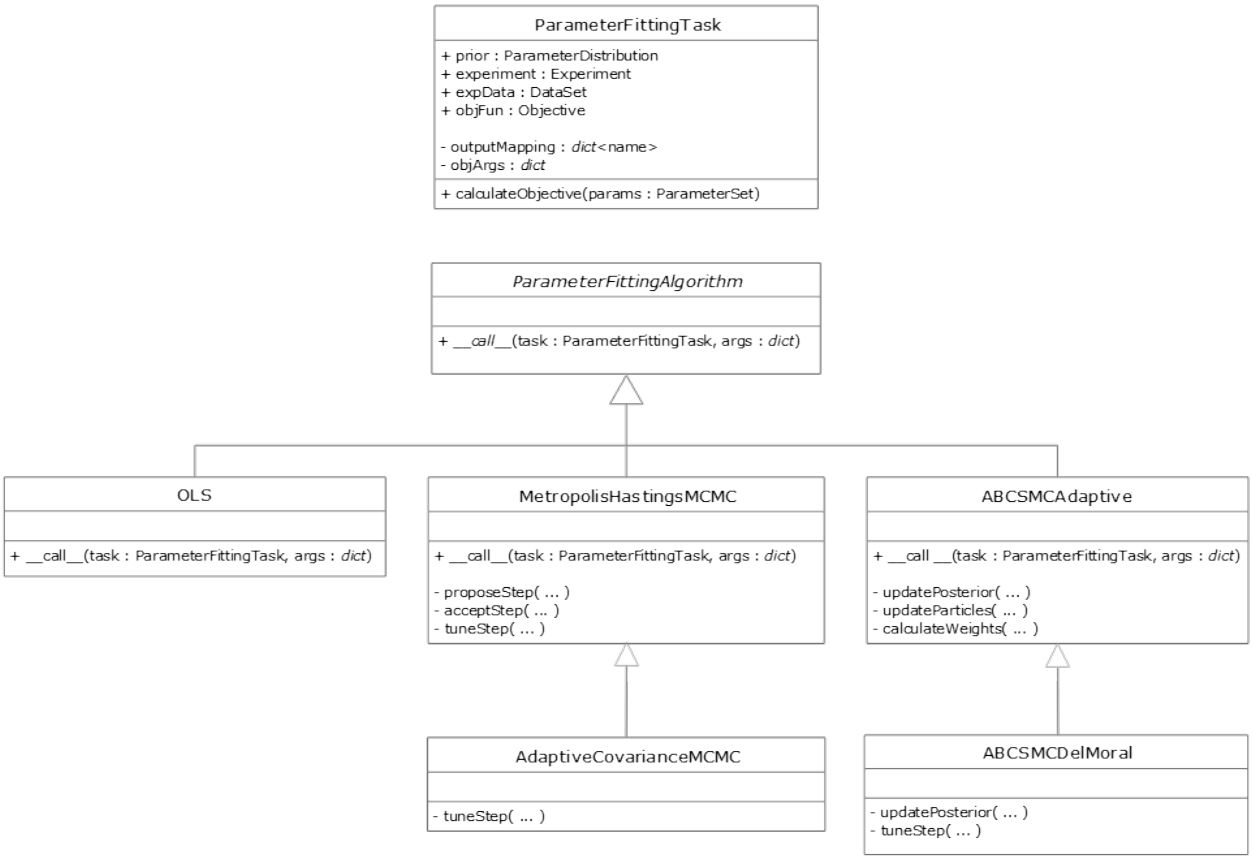
In addition to simply containing these objects, our implementation provides functions to handle basic interactions between these components. For example, the class contains a function "calculate objective," which takes a `ParameterSet`, generates simulated data using an `Experiment`, and evaluates the `Objective` function between corresponding simulated and experimental data. Creating such pipelines between interacting components removes the burden of re-implementing these interactions, making it easier to develop and deploy new algorithms.

In Figure 5.6, we see our definition of the abstract base class for `ParameterFittingTask`, which contains instances of `ParameterDistribution`, `Experiment`, `DataSet`, and `Objective`.

As stated above, the main functionality implemented by `ParameterFittingTask` is the `calculateObjective()` function, which simplifies the interaction between `Experiment` and `Objective` by allowing the algorithm to directly evaluate the fitness of a given set of parameters against the stored experimental data. In addition to simply linking the outputs of `task.experiment.simulate()` and `task.expData` to `task.Objective`, this method provides some additional functionality:

- If simulation fails under the specified parameters, the function defaults to the value spec-

Figure 5.6 UML depiction of the ParameterFittingAlgorithm abstract base class and several provided implementations, as well as the wrapper class ParameterFittingTask which serves as input.



ified by `Objective.defaultValue` (if specified), or $\pm\infty$, depending on the value of `Objective.isMinimized`. This sanitizes against problematic inputs which may be inadvertently allowed under the prior.

- Parameters designated for the objective function – denoted by a reserved namespace prefix "obj," (e.g., `obj:parameterName`) – are separated out prior to simulation and passed to the objective. This allows for the fitting of objective parameters, such as the magnitude of Gaussian noise, in tandem with model/protocol parameters without requiring interaction between `Experiment` and `Objective` objects.

- Additional fixed parameters to the objective may be specified in `task.objArgs`, and are combined with any objective parameters specified in the input `ParameterSet` prior to evaluation.

- Mapping between the names of simulated outputs and experimental outputs may be optionally specified in `task.outputMapping`. This may also be used to specify subsets of outputs to consider for the objective function, as either simulated or experimental `DataSet` objects may contain irrelevant data.

These steps provide convenient functionality beyond the basic class interactions, but do not change the fundamental responsibilities of each class. Future versions of `ParameterFittingTask` might easily replace the manner in which objective function parameters are specified during fitting, for instance, without altering the manner of input expected by `Experiment.simulate()`. By localizing nearly all component class interactions within the `ParameterFittingTask` class, we can easily adapt to unanticipated demands of future simulation domains.

5.6.2 The `ParameterFittingAlgorithm` Class

The `ParameterFittingAlgorithm` class provides functions that operate on `ParameterFittingTask` objects in order to produce `ParameterDistribution`

objects, which represent the results of posterior inference. Because single points in parameter space can be represented as `ParameterDistribution` objects (see Section 5.4.2), this definition of an algorithm encapsulates both optimization algorithms and posterior inference algorithms. For our initial deployment of the Modeling Web Lab, we have provided implementations of all posterior inference algorithms discussed this far: exact and approximate rejection sampling (Algorithms 2.1 and 3.1), the exact and approximate Toni SMC algorithm (Algorithm 3.2), the exact and approximate Del Moral SMC algorithm (Algorithm 3.3), and the adaptive covariance MCMC algorithm (Algorithm 4.1) — all of which make use of the "calculate objective" functionality of the `ParameterFittingTask` class in order to simplify sampling from prior and proposal distributions.

Other commonly used algorithms, however, may put explicit or implicit restrictions on the structure of parameter space. Ordinary least squares (OLS) fitting, for example, implicitly assumes a continuous, numerical input space and differentiable outputs, as it leverages derivative information from the Jacobian matrix (either provided analytically or estimated numerically) to update parameter estimates. Such a strategy would be fundamentally incompatible with some parameter spaces within our broad definition, such as those taking on non-numerical values (such as `string`) or having discrete prior distributions. While leveraging this type of system-specific information allows OLS and other optimization strategies to explore parameter space efficiently, it comes at the cost of generalizability, and future modelers may find difficulty repeating fitting experiments in alternative formulations of the system. While these restrictions do limit our interest in these types of methods, they can be implemented in our framework. We have provided an `OLS` class that wraps the implementation found in SciPy's "optimize" module[1], and performs type-checking to ensure numerically valued inputs. We have also wrapped other optimizers such as CMA-ES (Hansen and Ostermeier, 2001), which we will employ alongside posterior inference methods in the Modeling Web Lab prototype discussed in the next chapter.

In Figure 5.6, we present our definition of the `ParameterFittingAlgorithm`

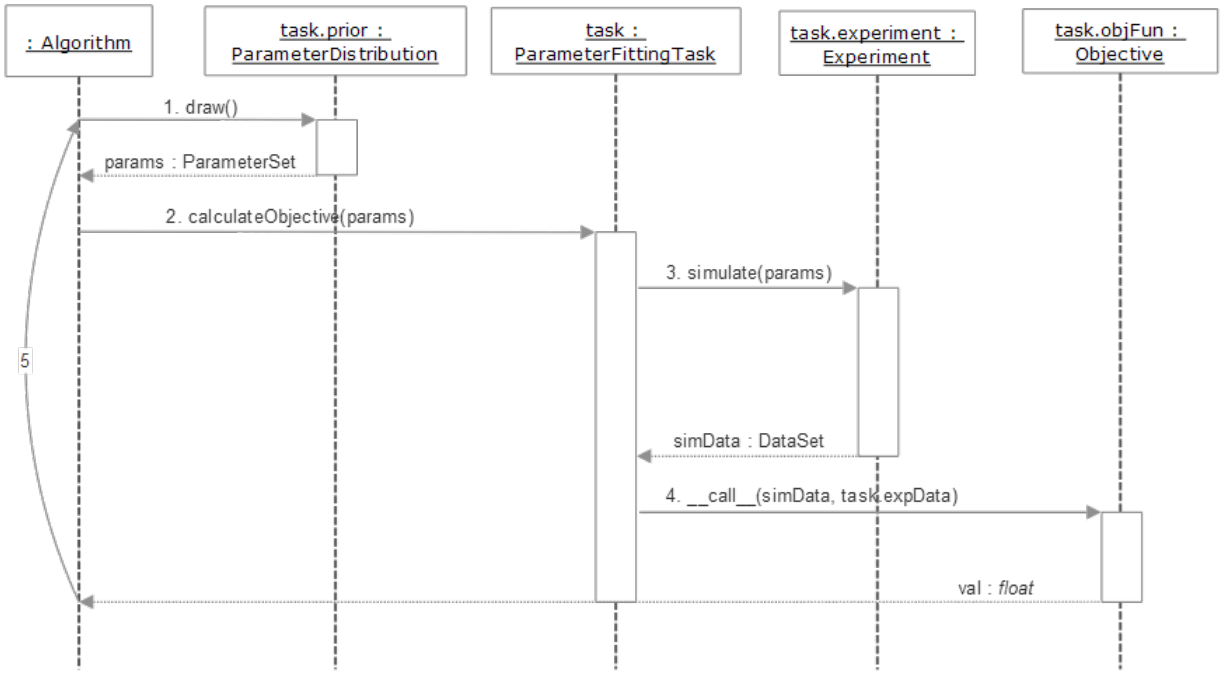
[1] http://docs.scipy.org/doc/scipy/reference/optimize.html

class, which provides a single function, `__call__()`, operating over a `ParameterFittingTask` and a dictionary of optional algorithmic arguments and returning a `ParameterDistribution` object.

Similarly to the `Objective` class, we have chosen to define `ParameterFittingAlgorithm` as a callable class rather than a standard function to allow for the inheritance of sub-routines. This allows for incremental development of algorithms by inheritance and overriding of class-specific subroutines. As an example, we see in Figure 5.6 the definition of a basic Metropolis-Hastings MCMC algorithm (Hastings, 1970) in `MetropolisHastingsMCMC` (see Section 2.2.2). By inheriting from this, we may easily define the variant implementation employed in Chapter 4, `AdaptiveCovarianceMCMC`, which moves differently through parameter space by redefining the private method `tuneStep()` (Haario et al., 2001; Bardenet, 2013). Similarly, we provide two variant implementations of the approximate Bayesian computation sequential Monte Carlo (ABC-SMC) sampler (Toni et al., 2009; Del Moral et al., 2012; Daly et al., 2017a), which are discussed extensively and compared in Chapter 3, by overriding the `updatePosterior()` method of `ABCSMCAdaptive` and providing an additional helper method to define `ABCSMCDelMoral`. These inheritance patterns demonstrate the potential for a "genealogy" of parameter fitting methods to emerge by creatively factoring common methods.

To demonstrate the basic flow of algorithmic execution within our framework, we present a UML sequence diagram demonstrating the interaction between a `ParameterFittingAlgorithm` and its target `ParameterFittingTask` for the simple rejection sampling algorithm (see Algorithm 2.1). We see in Figure 5.7 that rejection sampling is accomplished by repeatedly drawing samples from the prior and evaluating with `calculateObjective`. Other, more complicated sampling techniques can be developed from this methodology by sampling from distributions other than the prior: Metropolis-Hastings sampling performs similar draw-update steps, but with each draw (after the first)

Figure 5.7 UML sequence diagram representing the execution flow of basic rejection sampling steps within the execution of an `Algorithm`. Labeled boxes at the top of the diagram indicate objects involved in parameter fitting, with class name and optional variable name separated by a colon. The vertical dimension represents execution time, with numbered solid arrows indicating transfer of control flow by one class calling another's method, and corresponding dashed lines indicating returned values.



centered about the previous draw, and importance sampling draws from a sequence of intermediate distributions with perturbation. Each of these sampling techniques, which form the basis of many Bayesian and approximate Bayesian inference strategies, follows a basic "proposal-evaluation-update" sequence that exposes no information about the underlying structure of the parameter space to the `ParameterFittingAlgorithm` class.

5.7 Future Development

While the framework we have presented in this chapter is sufficient for representing the types of parameter inference algorithms that may be applied in various biological and chemical domains (as we will detail in the next chapter), we believe it is important to continue development on the framework so that it may additionally represent studies in model selection and experiment design. This will give our framework the capability for a broad range of expression, covering the majority of modeling studies performed so far in biological domains and enabling modelers to perform advanced techniques in a reproducible manner.

Within our framework, the ability to perform model selection will be inherently limited to experiments respecting the separation of model and protocol as defined in the functional curation paradigm. As described in Section 5.3, we have removed this notion of separation within the parameter fitting framework to allow for modeling experiments to take place in the case of general, black-box simulation, giving great generality to our framework. For the purposes of supporting model selection within this paradigm, we may maintain internal separation between these concepts by creating a subclass of `Experiment` that requires a "set model" method, or by adding support into existing subclasses (such as `FunctionalCurationExperiment`) to change the model as one would a parameter. These alterations would allow for "explicit" model selection, where we choose a model from a fixed set that maximizes a metric of model appropriateness such as the Akaike or Bayes information criteria (Akaike, 1973).

In addition to explicit model selection, we may support "implicit" model selection methods such as sparsity-induced selection, in which all potential model structures are contained within the parameter space (e.g., a fully-connected Markov model) and selection is performed via optimizing parameters while penalizing complexity. Under such a scheme, model selection is identical to parameter fitting, but with a slightly altered objective function. In order to represent this type of model selection, we have created a prototype implementation called `RegularizedObjective`, a subclass of `Objective` whose "call" method accepts an ad-

ditional `ParameterSet` argument used to calculate a complexity penalty term. The passing of this additional argument would be handled by the `ParameterFittingTask` class, allowing the other components of parameter fitting to remain unchanged.

Finally, we consider the class of algorithms concerned with experimental design. As discussed in the previous chapter, experiment design algorithms generally seek to parameterize a protocol so that its output minimizes uncertainty about the parameters of a given model. In order to allow for this, we would have to introduce the concept of a `DataGenerator` — analogous to the SED-ML concept of a "DataGenerator" (Waltemath et al., 2011) — which produces `DataSet` objects in response to changing `ParameterSet` inputs, as optimizing over experimental parameters means that the "recorded" data will no longer be static between iterations of the algorithm. The algorithms currently proposed for experiment design are perhaps even more varied than parameter fitting algorithms, with many employing system-specific information such as the Jacobian and Fisher information matrices (see Chapter 4). Some methods, such as the Asprey-Macchieto method, even seek to design protocols that guarantee worst-case or average case performance over a range of model parameters, which requires simultaneous optimization over two parameter domains (Asprey and Macchietto, 2002). While such actions are inherently possible in our framework, the lack of consensus on approach and variety of information required by these algorithms makes it difficult to standardize a form for "`ExperimentDesignAlgorithm`" as we did for `ParameterFittingAlgorithm`. As such, inclusion of these methods in our framework might be considered on an algorithm-by-algorithm basis, with each distinct strategy requiring its own definition of an associated "task".

5.8 Summary

Having discussed the paradigms of the framework for modeling experiments, with a particular eye towards parameter estimation experiments, we will next present two initial prototype versions of the Modeling Web Lab employing this framework to represent the results of studies in

cardiac cells and electrochemical reactions.

6

The Modeling Web Lab

Having described a framework capable of representing parameter fitting and identifiability analysis experiments of the type used to analyze biological models throughout this thesis, we now present a prototype implementation of an online community modeling resource — the Modeling Web Lab — built upon this framework.

We describe example problems in two modeling domains, cardiac electrophysiology and electrochemistry, and demonstrate how parameter fitting and identifiability analysis studies using experimental data may be specified, executed, and shared using the Modeling Web Lab platform.

6.1 Introduction

In the introductory chapter of this thesis, we posited that communities of modelers in the natural sciences could benefit from improved standards of model reporting, particularly in regard to uncertainty quantification and reproducibility. Current practices in which only a final, parameterized form is reported give no indication as to the uniqueness of these parameters in their ability to explain the data (i.e., the identifiability of the model), let alone the data employed in deriving their values. This discards a significant amount of biological information and makes models difficult to trust. Additionally, if insufficient information about the process by which the

model is created is reported, it will be difficult or impossible to independently deploy similar techniques, impeding our ability to adapt models to reflect our changing understanding of a system of study.

We proposed to address these problems through the development of online community resources for model development, which would promote the application of identifiability analysis techniques, as well as the specification and sharing of modeling experiments in formats that are transparent and reproducible. We are now ready to present a prototype implementation of such a resource, the Modeling Web Lab, which is built upon the framework for representing modeling experiments described in Chapter 5. As with the "Experiment" class in said framework, each instance of this Modeling Web Lab will be specific to a class of models, such as cardiac cell models, but the paradigms employed by the Modeling Web Lab can be easily applied to new modeling domains. While the underlying modeling framework is capable of representing a range of modeling experiments, as discussed in the previous chapter, our initial implementation is focused on parameter fitting experiments, primarily Bayesian fitting techniques, along with *a posteriori* methods for assessing identifiability that follow from the results of these methods (see Chapter 4).

We focus initially on the cardiac cell implementation of the Modeling Web Lab, as these models have been the primary target of the parameter fitting and identifiability techniques developed in this thesis. Within this modeling domain, we build on existing community tools for the representation and exchange of models and simulation results, such as the CellML language for the representation and exchange of parameterized models (Lloyd et al., 2004; Garny et al., 2008; Miller et al., 2010), and the Cardiac Electrophysiology Web Lab for the specification of simulation experiments (Cooper et al., 2016). The use of accepted standards, when possible, minimizes the barrier of entry to the Modeling Web Lab for members of the cardiac modeling community.

In Section 6.2, we introduce a model of the hERG ion channel, which will be the flagship model used to demonstrate the functionality of the cardiac Modeling Web Lab. This model,

which is similar in structure to the Hodgkin-Huxley style models we have considered so far in this thesis, was recently constructed by Beattie et al. (2017) using original voltage clamp experimental data. We then detail the implementation of the cardiac Modeling Web Lab, and demonstrate its ability to reproduce the results of this contemporary fitting experiment within our rigorous and transparent framework. This section also serves to highlight the functionality that the Modeling Web Lab provides for easy visualization and exchange of results from these kinds of fitting experiments.

In Section 6.3, we demonstrate the extensibility of the Modeling Web Lab paradigm to other domains by introducing the electrochemistry Modeling Web Lab, which deals with models of electron transfer reactions. These models take the form of systems of partial differential equations (PDEs), thus demonstrating the ability of the underlying modeling framework to handle time-series data beyond the ordinary differential equation (ODE) models considered in Chapters 3 and 4. After providing background on these reaction models, we introduce the implementation of the electrochemistry Modeling Web Lab, highlighting differences from the cardiac Modeling Web Lab resulting from the differences in the domain. We then demonstrate the ability of this Modeling Web Lab to reproduce the results of two Bayesian modeling studies of ferricyanide electron transfer reactions we have conducted employing data provided by our experimental collaborators in the Bond group (Gavaghan et al., 2017). Finally, we will use the Modeling Web Lab to visually compare the results of the two aforementioned studies, revealing the effect of changes to the experimental protocol on the resulting posterior uncertainty.

We have chosen to employ these models, rather than the Hodgkin-Huxley and O'Hara-Rudy models considered earlier in the thesis, due to the abundance of original experimental data about these systems that have been made available to us through our collaborations with Beattie et al. (2017) and Alan Bond's group at Monash University. Like cardiac modeling, electrochemical modeling is a field that is studied by a sizeable community, but has not benefited to date from advanced modeling techniques such as identifiability analysis, let alone standards for the reproducible documentation of such studies. Furthermore, both the voltage clamp experiments used

to study the hERG channel (and ion channels in general) as well as the cyclic voltammetry experiments used to study electrochemical reactions accept parametric inputs that until now have been chosen without rigorious consideration of model identifiability, making both systems ideal targets for studies in experimental design.

6.2 The Cardiac Modeling Web Lab

The first Modeling Web Lab prototype we will introduce is targeted towards cardiac models such as the ones we have been examining in this thesis. In this section, we will consider the a Hodgkin-Huxley style model of the hERG ion channel and detail how a parameter fitting study on this model may be represented in our prototype.

The protein encoded by the KCNH2 gene, also known as hERG, is the principal component of the ion channel responsible for carrying the delayed rectifier potassium current I_{Kr}. This current is primarily active during the repolarization phase of the action potential, when the cell returns to its initial excitable state, and is a necessary step for achieving a regular heart rate (see Section 2.1). As one might expect from such an essential protein, disruption in its activity has been linked to dangerous arrhythmias such as Torsades de Pointes, which may have fatal consequences (Curran et al., 1995). Moreover, the channel is highly susceptible to binding and blockage by blood-borne pharmaceutical compounds, making it a frequent concern in the design and testing of drugs (Hancox et al., 2008). As a result of these concerns, both the wild-type and drug-blocked behavior of the hERG channel are important targets of study, and have lead to tremendous interest in the development of models that may accurately capture its behavior under a range of conditions (Beattie et al., 2017). The prevalence of competing model formulations and interest in assessing model trust make this an ideal target for deployment in our Modeling Web Lab.

In the subsequent sections, we will introduce a proposed model for the hERG channel, as well as a novel voltage protocol proposed by Beattie et al. (2017) to fit the parameters of said

model, and demonstrate how the results of this parameter fitting study may be reproduced and disseminated using our prototype Modeling Web Lab. A live version of this Modeling Web Lab, along with public example files, can be found at `https://muck.cs.ox.ac.uk/FunctionalCuration`, and the reader is encouraged to visit this web site while reading Section 6.2.2. It is worth noting that this is a prototype implementation, and as such only essential functionality of the Modeling Web Lab will be discussed below, with details of navigation and specific user documentation left to descriptive text on the live web site.

6.2.1 Modeling the hERG channel

As with most ion channel models, the main observable of interest in studying the hERG channel is the current associated with said channel, I_{Kr}. No matter the underlying model chosen to represent the channel kinetics, this current can be calculated by Ohm's law as:

$$I_{Kr} = G_{Kr} \cdot O(V_m, t) \cdot (V_m - E_K), \tag{6.2.1}$$

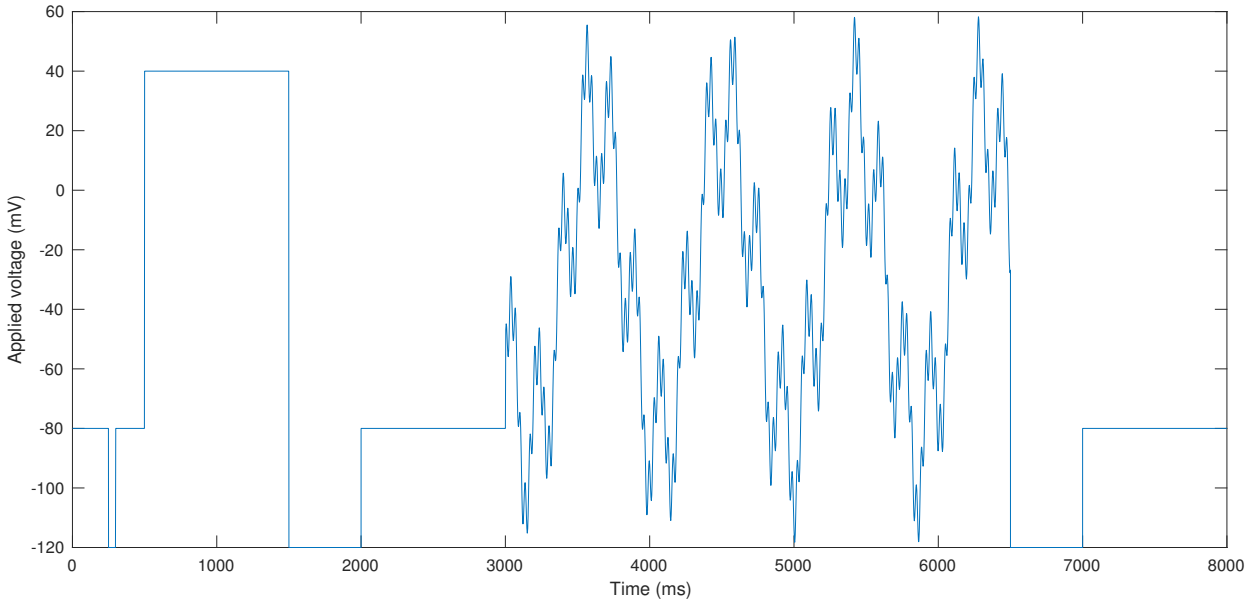
where G_{Kr} is the maximum ionic conductance, O is the probability of the channel being in the "open" state (in turn a function of membrane voltage and time), V_m is the transmembrane voltage, and E_K is the reversal potential associated with potassium ions. As detailed in Section 2.1.2, this reversal potential — the potential at which no net flow of ions is observed — is given by:

$$E_K = \frac{RT}{zF} \ln \left(\frac{[K]_o}{[K]_i} \right), \tag{6.2.2}$$

where R is the ideal gas constant, T is the temperature in Kelvin, z is the valency of the ion in question (1, in this case), F is Faraday's constant, and $[K]_o$ and $[K]_i$ are the extracellular and intracellular concentrations of potassium ions, respectively.

As detailed in Chapters 2 and 3, modelers have traditionally relied on voltage clamp experiments where V_m is varied according to "square wave" patterns over time (held at differing constant levels for differing durations) in order to elicit a current response to constrain the model. Recently, however, there has been interest in developing condensed voltage protocols that probe

Figure 6.1 Visual representation of the mix of sine wave voltage protocol proposed by Beattie et al. (2017) to probe the dynamics of the hERG channel.



system dynamics more efficiently, including voltage signals based on sinusoidal waves. The hope for these types of experiments is that they may be used to constrain model parameters while lowering expenditure in time and resources during data collection, and may be optimized to maximize model identifiability.

Motivated by these interests, which tie in to our discussions of optimal experimental design in Chapter 4, we will be considering the following original voltage protocol (represented visually in Figure 6.1) proposed by Beattie et al. (2017) to probe behavior of the hERG ion channel using a mix of square and sine wave voltage inputs:

$$V_m(t) = \begin{cases} -80 & \text{if } t \leq 250 \\ -120 & \text{if } 250 < t \leq 300 \\ -80 & \text{if } 300 < t \leq 500 \\ 40 & \text{if } 500 < t \leq 1500 \\ -120 & \text{if } 1500 < t \leq 2000 \\ -80 & \text{if } 2000 < t \leq 3000 \\ -30 + \sum_{i=1}^{3} A_i \sin(2\pi\omega_i(t - 2500)) & \text{if } 3000 < t \leq 6500 \\ -120 & \text{if } 6500 < t \leq 7000 \\ -80 & \text{if } 7000 < t \leq 8000, \end{cases} \tag{6.2.3}$$

where t is the time in milliseconds, giving a total duration of 8 seconds. We see that the sinusoidal portion of the protocol is composed of a mix of three waves controlled by six parameters: $A_1 = 54$mV, $A_2 = 26$mV, and $A_3 = 10$mV dictating the amplitudes of the component waves, and $\omega_1 = \frac{0.007}{2\pi}$, $\omega_2 = \frac{0.037}{2\pi}$, and $\omega_3 = \frac{0.19}{2\pi}$ controlling their periods. Since the design of the protocol, including the number, duration, and amplitudes of the square waves preceding and following the sine wave, was chosen by expert intuition rather than automated experimental design, a discussion thereof is beyond the scope of this application study, and can instead be found in Beattie et al. (2017).

The main source of variety across models proposed for the hERG channel is in the equations chosen to represent O. For the purposes of this study, we will focus on a simple 4-state Hodgkin-Huxley style ion channel model which was shown to be identifiable under the protocol outlined in Equation 6.2.3, in addition to showing good generalization over a range of validation protocols (Beattie et al., 2017).

The four states of this model — open (O), closed (C), inactive-open (IO) and inactive-closed (IC) — evolve according to the following system of differential equations:

$$\frac{dO}{dt} = k_O C + k_A IO - (k_C + k_I)O, \tag{6.2.4}$$

$$\frac{dC}{dt} = k_C O + k_A IC - (k_O + k_I)C, \tag{6.2.5}$$

$$\frac{dIO}{dt} = k_I O + k_O IC - (k_A + k_C)IO, \tag{6.2.6}$$

$$IC = 1 - (O + C + IO), \tag{6.2.7}$$

which may also be represented as a set of Hodgkin-Huxley style equations with two gating variables, due to the independence of opening-closing and activating-inactivating transitions. The rate constants k associated with these equations are in turn controlled by the following

parametric functions:

$$k_O = k_{O1} \exp(k_{O2} V_m), \tag{6.2.8}$$

$$k_C = k_{C1} \exp(-k_{C2} V_m), \tag{6.2.9}$$

$$k_I = k_{I1} \exp(k_{I2} V_m), \tag{6.2.10}$$

$$k_A = k_{A1} \exp(-k_{A2} V_M). \tag{6.2.11}$$

Since we assume the initial state to be known, with all channels fully in state C as a result of experimental manipulation (Beattie et al., 2017), there are 8 parameters controlling the kinetics of the hERG channel: k_{O1}, k_{O2}, k_{C1}, k_{C2}, k_{I1}, k_{I2}, k_{A1}, and k_{A2}. Since the observable being considered is I_{Kr} and not O directly, we must additionally consider G_{Kr} to be a free parameter, giving a total of 9 parameters for the ionic current model.

We will next discuss the implementation of the cardiac Modeling Web Lab and the manner in which a fitting experiment over these nine model parameters may be specified and executed.

6.2.2 Using the Cardiac Modeling Web Lab

As stated in the chapter introduction, the cardiac Modeling Web Lab builds on the existing cardiac electrophysiology Web Lab, which adheres to the functional curation paradigm separating the notion of a model (the underlying mathematics representing the system) from the protocol (the manner in which the system is stimulated and observed) (Cooper et al., 2016). Users of the cardiac electrophysiology Web Lab upload separate model and protocol files, represented respectively in CellML and functional curation protocol syntax, in order to specify and share the results of simulations. Model parameters in the CellML files should be tagged with metadata annotations that allow them to be externally read or adjusted by simulation protocols, and thus there exists a notion of "compatibility" between model and protocol, which requires that

all variables referenced in the protocol exist as tagged entities in the model. The exact structure and interpretation of these files is beyond the scope of this thesis, and the reader is referred to Cooper et al. (2011) for further details.

Our Modeling Web Lab expands upon this paradigm to specify parameter fitting experiments. Users are still required to upload separate files specifying a CellML model and a functional curation protocol, which, when compatible, are internally combined to create an `Experiment` entity for simulation (see Section 5.3). When model and protocol specifications are incompatible, they cannot be paired to run a fitting experiment.

What differs in the Modeling Web Lab is that, in order to specify a parameter fitting experiment, simulation protocols must be paired with two additional files: a file containing experimental data, and a "fitting protocol" that specifies the information necessary to execute the fitting experiment. We will delineate the structure and expressive capabilities of these files by way of example using the fitting experiment specified by Beattie et al. (2017) using the model and simulation protocol specified in Section 6.2.1. The files that will be discussed are available as a public fitting protocol on the cardiac Modeling Web Lab (`https://muck.cs.ox.ac.uk/FunctionalCuration`), and can be viewed by clicking the "Sine Wave Fitting" entry on the "Experiments" sub-page.

The data file is used to specify named experimental data collected from the system of study. It takes the form of a comma-separated value (CSV) file, where the first row specifies names for each variable, and associated columns specify the corresponding data. We note that this structure currently expects 0- or 1-dimensional data for each named variable (although higher-dimensional arrays may be specified in flattened form), as these were the only types of data currently being employed for our application problems. The exact manner of representing data could easily be changed to a more expressive format such as NuML (`https://github.com/NuML/NuML`) without any effect on the rest of the system, due to the abstractions employed by the underlying parameter fitting framework. In the hERG fitting experiment, the data file contains two columns of equal length representing time and the

Table 6.1 Entries in the fitting protocol for the hERG ion channel model. The value associated with the "algorithm" entry is a string of characters, and is represented as is, while all other value entries are nested JSON objects, and are presented in "key=value" format for clarity. This is also true for the prior specification, which is represented separately in Table 6.2 due to its size.

Fitting Protocol Entity	Value
algorithm	AdaptiveMCMC
arguments	cmaOpt=5, cmaMaxFevals=20000, burn=50000, numIters=100000
output	IKr=IKr
input	exp_times=t
prior	(see Table 6.2)

associated I_{Kr} current.

The fitting protocol takes the form of a JSON-formatted[1] text file. These types of files contain a series of named values (i.e., a dictionary), where each value may be either a string of characters, a numerical value, a list of values, or a nested JSON object. We represent the contents of the fitting protocol for the hERG fitting experiment in Table 6.1, and will discuss the interpretation of each required entry below.

The first entry in the fitting specification is the "algorithm" to be used, which is specified by a unique identifier. In the case of the hERG fitting experiment, this value is "AdaptiveMCMC", which maps to the MCMC algorithm specified in Algorithm 4.1. While the underlying modeling framework outlined in the previous chapter maintains a separation between the algorithm being used and the objective function it employs, the current implementation does not expose this functionality to the user in order to facilitate the application of common modeling experiments. As such, in the Modeling Web Lab, the adaptive MCMC algorithm currently assumes a Gaussian likelihood function, which is commonly assumed for time-series data. If future users demand a range of available likelihood functions for each algorithm, the fitting protocol language may be extended to support the specification of an optional "objective" entry.

The next entry we consider is a dictionary of "arguments," specific to the execution of the algorithm. In the hERG example, these include the standard arguments for MCMC — the total number of iterations "numIters" and the number of iterations discarded as burn-in "burn" —

[1] `www.json.org`

as well as two new arguments “cmaOpt” and “cmaMaxFevals” which deal with the selection of a starting point for MCMC. These arguments tell the back end to first run a series of 5 random restarts of a global optimizer, the “Covariance Matrix Adaptation Evolution Strategy”, to choose a starting point for the MCMC chain. This algorithm, which was employed by Beattie et al. (2017), employs a “population of solutions” to explore parameter space, similar to the genetic algorithm and particle swarm optimizer mentioned in Section 2.2.1, yet differs from these methods in that it uses an evolving covariance matrix to propose moves in parameter space rather than random processes inspired by biology. The reader is referred to Hansen and Ostermeier (2001); Hansen and Kern (2004) for further details. In the final version of the Modeling Web Lab, a full list of available named algorithms, along with details of their operation and adjustable parameters, will be made available on the web site.

The next two sections, “inputs” and “outputs”, deal with matching experimental and simulated data. The “inputs” section details a list of named inputs to the simulation protocol (“exp_times”, in this instance) which are matched to named entries in the data file (“t”, in this instance). Here this tells the simulator to generate outputs at the times specified in the data file, and removes the need to alter the functional curation protocol when new data are collected. The “outputs” section tells the objective function which named outputs from the simulation protocol are to be considered (“IKr” in this instance), and which named entries in the data file (also called “IKr”) they are to be compared against using the objective function. This allows data files to be used that do not adhere to the same naming conventions as the associated simulation protocol, again avoiding the need to alter simulation protocols when new data are acquired.

Finally, we consider the prior distribution, which is specified as a nested JSON object identified by “prior” (and represented separately in Table 6.2). The Modeling Web Lab currently supports only uniform priors (specified by a two-membered list containing a lower and upper bound) or point distributions (a single fixed value), as these are the only cases demanded by our application models. As with the data set,we may easily change the way these are represented in order to support a broader range of use cases if demanded by users, as the underlying framework

Table 6.2 Prior distribution specified within the fitting protocol for the hERG model. Parameter names are shortened from their true metadata annotations to the names presented in Section 6.2.1 for clarity. "obj:std" refers to the magnitude of the Gaussian noise, and is passed as an argument to the objective function as denoted by the "obj:" ontology prefix (see Section 5.3).

Parameter Name	Prior Range
k_{O1}	Uniform(1e-7,0.1)
k_{O2}	Uniform(1e-7,0.1)
k_{C1}	Uniform(1e-7,0.1)
k_{C2}	Uniform(1e-7,0.1)
k_{I1}	Uniform(1e-7,0.1)
k_{I2}	Uniform(1e-7,0.1)
k_{A1}	Uniform(1e-7,0.1)
k_{A2}	Uniform(1e-7,0.1)
G_{Kr}	Uniform(0.0612,0.612)
obj:std	0.00463

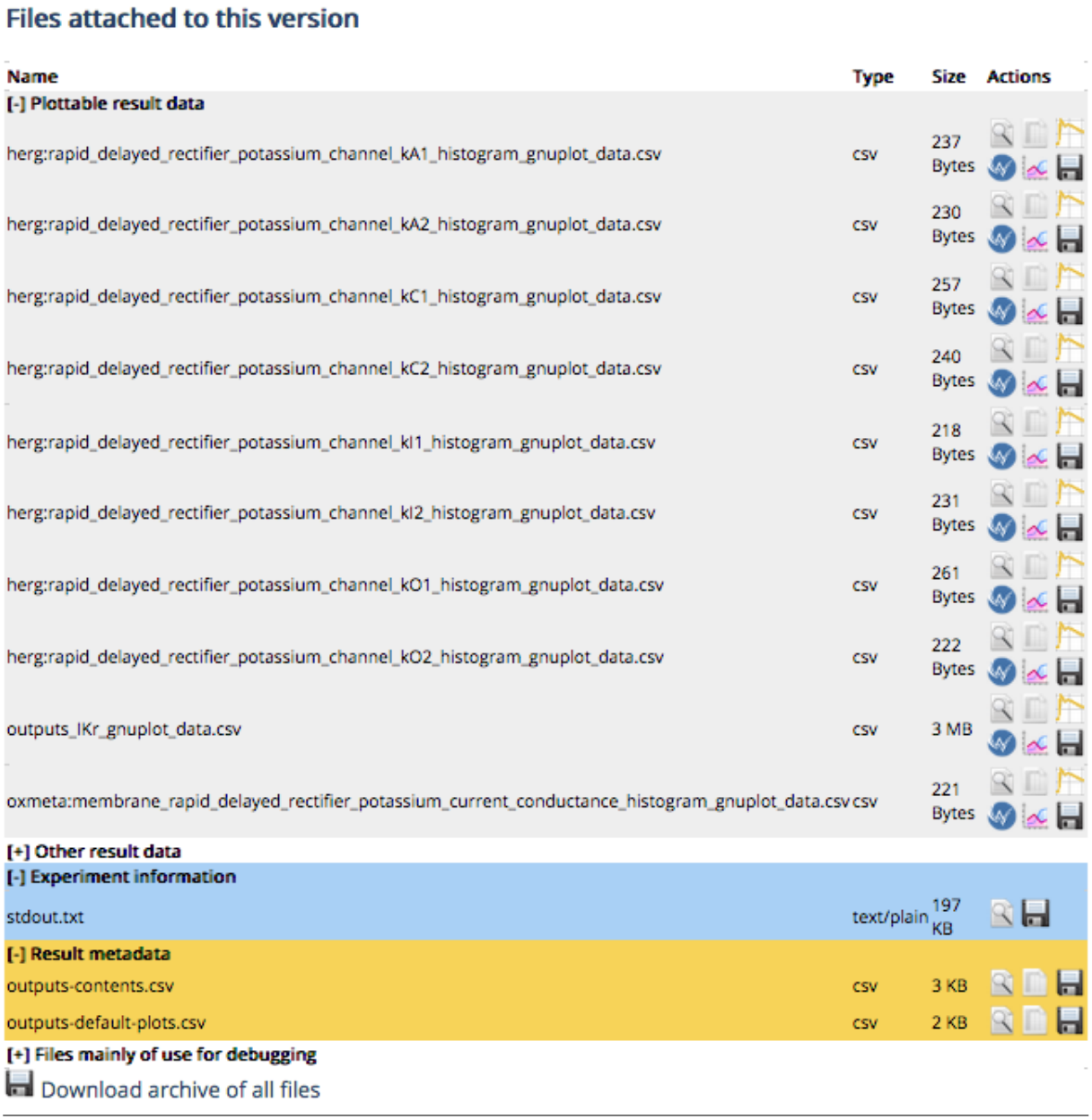
is capable of representing any type of parameter distribution (see Section 5.4).

We may now present the results of the fitting experiment specified by Table 6.1. By matching said specification with the hERG model in the "Experiment" sub-page of the Modeling Web Lab, users may execute this fitting experiment and view the results when it has completed. The results of fitting experiments are presented as a list of output files, all of which are available for download, and some of which can be viewed as interactive plots within the browser (see Figure 6.2).

In Figure 6.3, we see the results of plotting the experimental data against the corresponding data simulated under the mean posterior parameters. The two traces show strong agreement, indicating a successful fit. This is a good means of sanity-checking the results of a fitting experiment.

In Figure 6.4, we see the marginal posterior distribution over each model parameter produced by this inference strategy. These plots were generated from data obtained by the "export data" feature of the Modeling Web Lab, which provides descriptions of all marginal posterior distributions, as well as time-dependent traces like those shown in Figure 6.3. While similar plots may be visualized in-browser from the results page, we chose to present independently visualized data rather than screen shots for the sake of clarity within this thesis. In this example the

Figure 6.2 Files returned by the cardiac Modeling Web Lab after posterior inference over the hERG model as specified in Table 6.1. Files listed under "Plottable result data" can be viewed as interactive plots by clicking the yellow graph icon, while all files may be downloaded by clicking the "save" icon or viewed within the browser by clicking the magnifying glass icon.

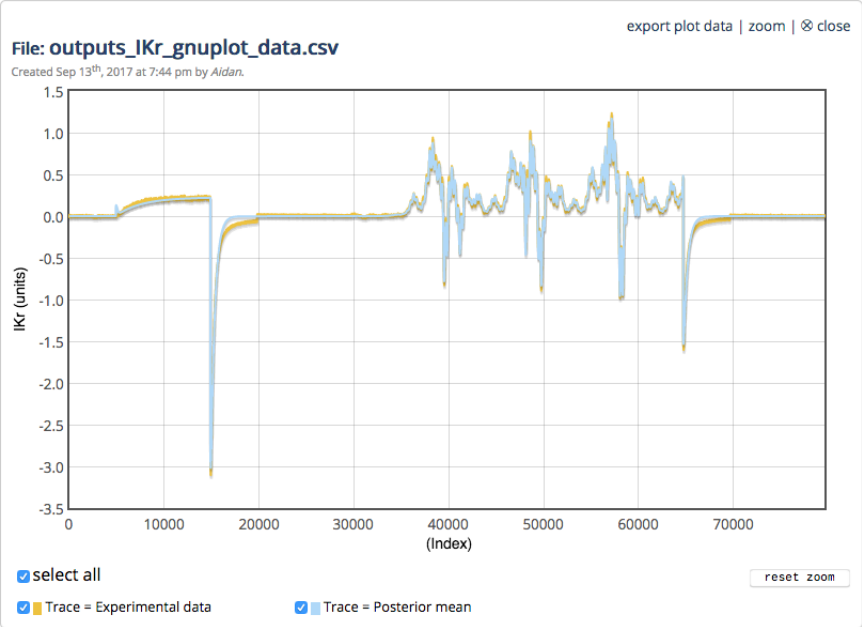


Files attached to this version

Name	Type	Size	Actions
[-] Plottable result data			
herg:rapid_delayed_rectifier_potassium_channel_kA1_histogram_gnuplot_data.csv	csv	237 Bytes	
herg:rapid_delayed_rectifier_potassium_channel_kA2_histogram_gnuplot_data.csv	csv	230 Bytes	
herg:rapid_delayed_rectifier_potassium_channel_kC1_histogram_gnuplot_data.csv	csv	257 Bytes	
herg:rapid_delayed_rectifier_potassium_channel_kC2_histogram_gnuplot_data.csv	csv	240 Bytes	
herg:rapid_delayed_rectifier_potassium_channel_kI1_histogram_gnuplot_data.csv	csv	218 Bytes	
herg:rapid_delayed_rectifier_potassium_channel_kI2_histogram_gnuplot_data.csv	csv	231 Bytes	
herg:rapid_delayed_rectifier_potassium_channel_kO1_histogram_gnuplot_data.csv	csv	261 Bytes	
herg:rapid_delayed_rectifier_potassium_channel_kO2_histogram_gnuplot_data.csv	csv	222 Bytes	
outputs_IKr_gnuplot_data.csv	csv	3 MB	
oxmeta:membrane_rapid_delayed_rectifier_potassium_current_conductance_histogram_gnuplot_data.csv	csv	221 Bytes	
[+] Other result data			
[-] Experiment information			
stdout.txt	text/plain	197 KB	
[-] Result metadata			
outputs-contents.csv	csv	3 KB	
outputs-default-plots.csv	csv	2 KB	
[+] Files mainly of use for debugging			

Download archive of all files

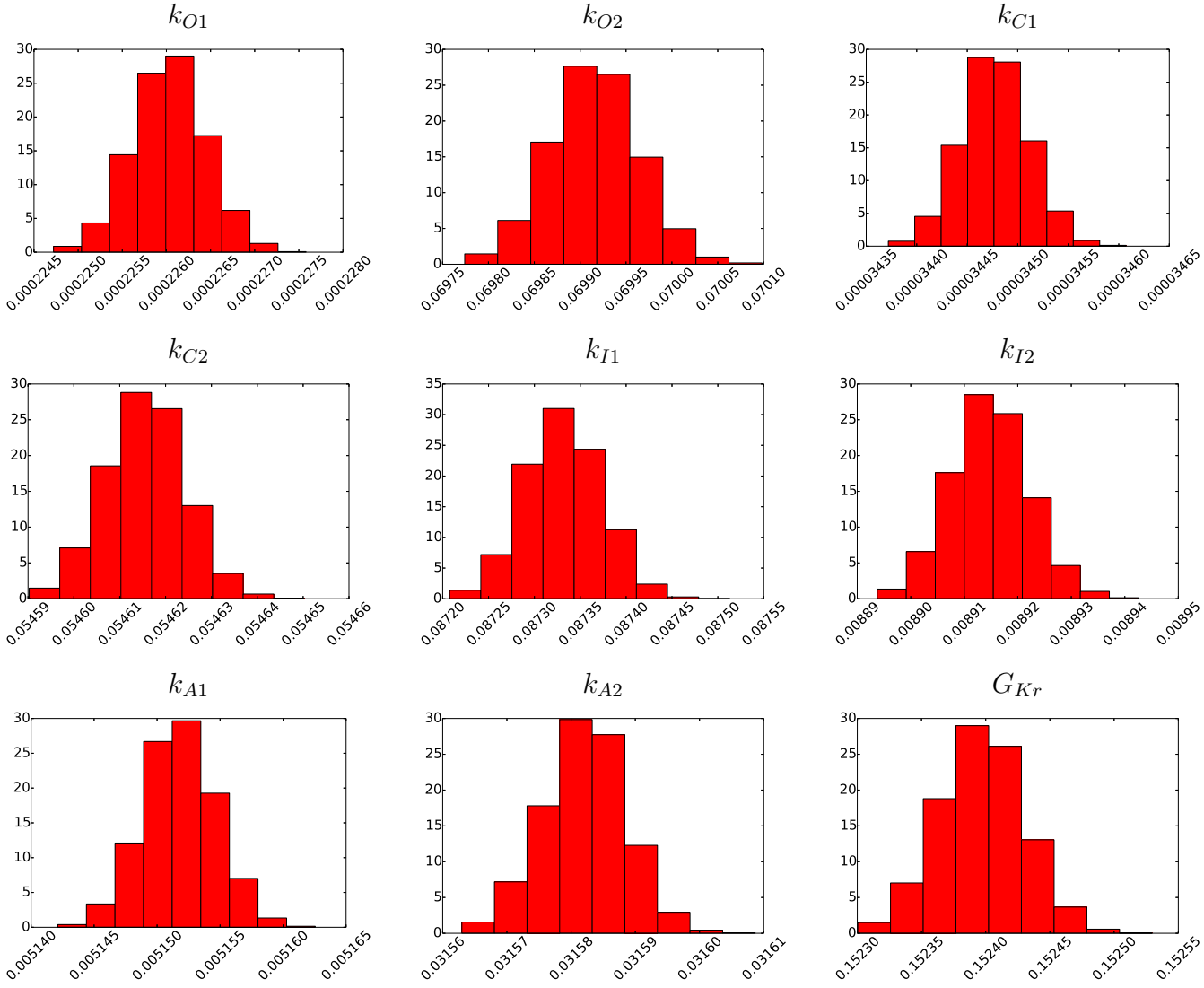
Figure 6.3 Experimental recording of I_{Kr} overlaid with I_{Kr} simulated under mean posterior parameters returned by fitting.



posteriors are all well-constrained Gaussian distributions, indicating that there is a clear, unique set of optimal parameters, and thus the model is fully identifiable. The results additionally show a strong homology with those presented in Beattie et al. (2017), indicating successful reproduction of the fitting experiment. While biplots of the posterior distributions like those shown in Figures 4.6-4.14 are not currently available, the full posterior estimate can be downloaded under the "Other result data" file section and visualized by the user on their desktop. Despite the currently limited means of visualizing the posterior distributions, which are being addressed for upcoming iterations of the Modeling Web Lab, this study demonstrates the capabilities of this resource to reproduce the results of current modeling studies using complex data and protocols.

Finally, we note that the results of these fitting experiments can be made publicly available by adjusting the "visibility" settings associated with each model and protocol. This allows users to develop models in private, then share results with either a set of fellow users or the community at large when certain milestones have been reached in the study.

Figure 6.4 Marginal posterior histograms for all nine parameters of the hERG model presented in Section 6.2.1. The marginal posterior shown in each panel is indicated by the label above the graph.



6.3 The Electrochemistry Modeling Web Lab

In order to demonstrate the applicability of the Modeling Web Lab paradigm to other systems using time-series data, we now present a variant of the prototype introduced in the previous section that instead operates on models of electrochemical "redox" reactions.

Redox reactions are chemical reactions involving the transfer of electrons between two species. These reactions are vital components of biological processes such as cellular respiration and photosynthesis, as well as chemical processes such as corrosion and combustion. Due to their importance, these reactions have been extensively studied, with the field of voltammetry emerging as one of the most popular means to probe their rates and mechanisms (Bard and Faulkner, 1980). Voltammetry studies are generally conducted on a compound in solution, where the kinetics of its reduction (gaining an electron) and/or oxidation (losing an electron) are probed by applying electrical potential through the use of an electrode (generally metallic) and recording the resulting current. Since redox reactions occur only at the electrode-solution interface, many factors such as the rate of diffusion of the compound, and the composition and shape of the electrode can affect the rate of reaction, in addition to the reactivity of the compound and the potential that is applied. As such, a great variety of voltammetry protocols are possible, and different settings may be necessary to elicit behavior in different systems.

While the possible mechanisms for redox reactions are well understood and limited to a few types of PDE models depending on how many electrons are involved in the transfer, it remains a challenge to provide accurate estimates of model parameters given voltammetry data, or even to select between competing reaction models in an automated fashion (Bieniasz and Speiser, 1998; Bieniasz and Rabitz, 2006). Current approaches have been limited to providing point estimates of reaction parameters, which give little indication of model identifiability (Gavaghan et al., 2017). Coupled with the broad range of possible voltammetry protocols, this makes it difficult to gauge the efficacy of an experiment in constraining model parameters, or to design protocols to maximize this information content as we have done for other systems in Chapter

4. As such, electrochemical reactions are another system of study that may benefit greatly from the Bayesian modeling techniques and identifiability assessment tools made available through our Modeling Web Lab.

In the subsequent sections, we will introduce a model of single-electron transfer reactions that we will fit to data gathered on the reversible reduction of $[\mathrm{Fe}(\mathrm{CN}_6)]^{3-}$ to $[\mathrm{Fe}(\mathrm{CN}_6)]^{4-}$ under two competing "cyclic" voltammetry protocols — alternating current (AC) and direct current (DC) — using our Modeling Web Lab. As with the cardiac Modeling Web Lab, we will demonstrate the expressive capabilities of the Modeling Web Lab in this domain, which makes use of custom languages to describe redox models and voltammetry protocols, as well as its ability to visually demonstrate the information content of competing protocols. A live version of this Modeling Web Lab, including publicly viewable examples, can be found at `https://lofty.cs.ox.ac.uk/FunctionalCuration`, and the reader is encouraged to visit this web site while reading Section 6.3.2. As with the cardiac Modeling Web Lab prototype, only essential functionality of the web site is discussed, with specifics on use and navigation left to instructions on the evolving web site.

In order to conserve the scope of this thesis, we provide a limited description of the kinetics of redox reactions under voltammetry experiments, with a focus on understanding the nature of the free parameters involved. Further discussion and derivations of electron transfer dynamics under these conditions can be found in (Bard and Faulkner, 1980; Fisher, 1996).

6.3.1 Electrochemical Reaction Models

For the purposes of this study, we will be considering single electron redox reactions of the form:

$$\mathrm{A} \leftrightarrow \mathrm{B} + \mathrm{e}^{-}. \tag{6.3.1}$$

In order to trigger these types of reactions and investigate their properties, we will be employing cyclic voltammetry, which applies a changing potential to the system that, beginning at an initial value E_{start}, varies linearly with time until a target potential E_{rev} is attained, then

switches direction until E_{start} is reattained. The simplest and most common form of cyclic voltammetry is the DC mode, which applies the following potential signal:

$$E_{dc}(t) = \begin{cases} E_{start} + vt & \text{if } 0 \leq t \leq \frac{E_{rev}-E_{start}}{v} \\ E_{rev} - vt & \text{if } \frac{E_{rev}-E_{start}}{v} < t \leq 2\frac{E_{rev}-E_{start}}{v}, \end{cases} \tag{6.3.2}$$

where v is the voltage sweep rate and t is time.

Despite the ubiquity of this DC approach, another variant of cyclic voltammetry, AC voltammetry, may provide more information about the underlying reaction mechanism (Bard and Faulkner, 1980). This technique overlays an alternating potential of the following form on top of a DC signal:

$$E_{ac}(t) = \Delta E \sin(\omega t + p), \tag{6.3.3}$$

where ΔE is the amplitude of the AC signal, ω is the angular frequency, and p is the phase shift. Because of this relation, we may express the applied potential E_{app} for either DC or AC voltammetry as the sum of two component signals:

$$E_{app}(t) = E_{dc}(t) + E_{ac}(t), \tag{6.3.4}$$

where DC signals are generated by setting $\Delta E = 0$.

In response to this applied potential, a current is generated in the system with two components: a Faradiac component I_f, which is generated by the flux of electrons at the electrode surface, and a capacitive component I_C, which results from the potential drop across the electrode-solution interface. Thus, we may represent the total system current I_{tot}, which is our main output of interest, with the following equation:

$$I_{tot} = I_f + I_C. \tag{6.3.5}$$

Using this, we may now examine the parameters that control the reaction, and, by extension, the observed current.

The first parameter of interest is the resistance, R_u, which modifies the applied potential of the voltammetry experiment to produce an "effective potential" E_{eff}, as shown in the equation

below:

$$E_{\mathit{eff}}(t) = E_{app}(t) - I_{tot}R_u. \tag{6.3.6}$$

With this in hand, we may calculate the capacitive current in the following manner:

$$I_C(t) = C_{dl}\frac{dE_{\mathit{eff}}}{dt}. \tag{6.3.7}$$

Finally, we consider the calculation of the Faradiac current. We assume the two ionic species A and B have respective concentrations $c_A(x,t)$ and $c_B(x,t)$ at distance x from the electrode surface. We are primarily concerned with c_A, which we will use as shorthand for $c_A(0,t)$, the concentration of species A at the electrode surface. Assuming equal diffusion constants ($D_A = D_B$), we need only solve for the concentration of c_A, as the two concentrations are related by $c_A = c_\infty - c_B$, where c_∞ is the bulk concentration of species A. We may model the evolution of c_A using Fick's second law:

$$\frac{\partial c_A}{\partial t} = D_A\frac{\partial^2 c_A}{\partial x^2}, \tag{6.3.8}$$

with initial and boundary conditions given by:

$$c_A(x,0) = c_\infty, \quad c_A \to c_\infty \text{ as } x \to \infty, t > 0. \tag{6.3.9}$$

Using conditions of conservation and flux at the electrode surface, we may relate the solution to the above equations to the Faradaic current as:

$$\frac{I_f}{FA} = D_A\frac{\partial c_A}{\partial x}, \tag{6.3.10}$$

Which, along with the Butler-Volmer condition, allows us to formulate the Faradiac current in the following manner:

$$I_f(t) = FA(c_A(t)k_{ox} - c_B(t)k_{red}), \tag{6.3.11}$$

where F is Faraday's constant, A is the surface area of the electrode, and k_{ox} and k_{red} are the rates for the oxidizing and reducing reactions, which are in turn calculated by:

$$k_{ox} = k_0 \exp\left((1-\alpha)\frac{F}{RT}(E_{\mathit{eff}} - E_0)\right) \tag{6.3.12}$$

Table 6.3 Parameters controlling single-electron transfer reactions described in Section 6.3.1.

Parameter	Interpretation	Units
R_u	Uncompensated resistance	Ω
C_{dl}	Double-layer capacitance	F/cm^2
E_0	Equilibrium potential	V
k_0	Equilibrium rate constant	s^{-1}
α	Transfer coefficient	dimensionless

Table 6.4 Parameters controlling DC and AC cyclic voltammetry protocols.

Parameter	Interpretation	Units
T	Absolute temperature	K
A	Electrode surface area	cm^2
E_{start}	Initial potential	V
E_{rev}	Reversal potential	V
v	Sweep rate	V/s
ΔE	AC signal amplitude	V
ω	AC signal angular frequency	Hz
p	AC signal phase shift	dimensionless

$$k_{red} = k_0 \exp\left(-\alpha \frac{F}{RT}(E_{\mathit{eff}} - E_0)\right), \tag{6.3.13}$$

where T is absolute temperature, R is the ideal gas constant, k_0 is the equilibrium rate constant (the value of both k_{ox} and k_{red} when $E_{\mathit{eff}} = E_0$), and α is the "transfer coefficient" that controls whether the reaction favors oxidation or reduction.

We note that, due to the spatial terms involved in the Faradaic current, calculating the total current I_{tot} will involve the solution of a system of *partial* differential equations, rather than the ODEs we have been accustomed to in cardiac models. While these are more difficult to solve numerically, we will demonstrate in the following section that this does not hinder the application of modeling techniques described using the framework presented in the previous chapter.

In summary, the model for single electron transfer we consider in this section is controlled by the free parameters presented in Table 6.3, while both DC and AC voltammetry protocols are controlled by the parameters presented in Table 6.4.

In the next section, we will demonstrate how the electrochemistry Modeling Web Lab may be

used to perform Bayesian inference over the five model parameters of interest, and investigate the comparative ability of DC and AC signals to constrain these parameters.

6.3.2 Using the Electrochemistry Modeling Web Lab

We will now introduce the prototype electrochemistry Modeling Web Lab by way of highlighting differences from the cardiac Modeling Web Lab introduced in Section 6.2.2, then detail the specification and results of fitting experiments conducted using two competing voltammetry protocols within this system.

The main difference from the cardiac Modeling Web Lab is the manner in which simulations are specified and carried out. As we recall, the cardiac Modeling Web Lab makes use of the functional curation protocol language to specify simulations on models represented in the CellML format. While there exist languages such as CML (`http://www.xml-cml.org/`) to represent chemical reactions, there does not yet exist a standardized functional curation-like language for specifying voltammetry protocols (or other general stimuli) on these model formats. This necessitated the development of a basic protocol language for these experiments, which we structured as a JSON file with required fields for all parameters specified in Table 6.4. Similarly, rather than employing an existing modeling language to specify chemical reactions, we opted to use a JSON file with fields specifying the five parameters in Table 6.3. Simulations were then carried out using instances of the `ElectroChemExperiment` class (see Section 5.3), which employs a custom ontology (reserving parameter names specified in Tables 6.3 and 6.4) to match prior distributions and inputs specified in the fitting protocol to entities in the model and simulation protocol. The simulator wraps a fully implicit finite difference scheme (see the θ-method in Morton and Mayers (2005)) implemented by Martin Robinson, a member of our group, which solves Equations 6.3.5-6.3.13 and outputs I_{tot} at specified times.

The benefits of employing this minimal, custom language for representing electrochemical models — rather than a broader language for general chemical reactions — is that we explicitly limit our users to the class of reactions currently supported by this Modeling Web Lab. The

Modeling Web Lab, as we have defined it, is a paradigm that aids in the community development of models centered around a specific system, and thus, clear limits of scope must be set to allow for maximal inter-operability between models and protocols. Despite this, the system may easily be updated to allow for a broader class of models (such as multi-electron transfer) without hindering the application of paradigms from other Modeling Web Labs. This will be demonstrated through the direct re-application of major system components from the cardiac Modeling Web Lab, which was constructed for an entirely different modeling domain.

In this vein, the electrochemistry Modeling Web Lab employs exactly the same language to specify fitting experiments as the cardiac Modeling Web Lab (see Section 6.2.2). Since the layer of abstraction provided by the `Experiment` class (in this case, `ElectroChemExperiment` instead of `FunctionalCurationExperiment`) and associated naming ontology, we may specify algorithms, algorithmic arguments, prior distributions, and protocol inputs/outputs without knowledge of the underlying model structure. The files used to represent data are identically structured to those in the cardiac Modeling Web Lab.

We will now provide details of the specification of Bayesian fitting of parameters controlling the following reaction:

$$[\mathrm{Fe(CN_6)}]^{3-} + \mathrm{e}^{-} \leftrightarrow [\mathrm{Fe(CN_6)}]^{4-}, \tag{6.3.14}$$

to DC and AC voltammetry protocols within our Modeling Web Lab. As in the cardiac Modeling Web Lab, we currently support only MCMC fitting with Gaussian likelihood and uniform prior distributions over model parameters, as these are the only usage cases demanded by our collaborators thus far. In Table 6.5, we detail the prior distributions over model parameters, as well as the assumed magnitude of Gaussian observational error σ_e, that were arrived at through discussion with our experimental collaborators (see Gavaghan et al. (2017)). In Table 6.6 we list specifications of DC and AC cyclic voltammetry protocols (with identical underlying DC signals) used by our experimental collaborators to probe the mechanisms of Ferricyanide redox reactions in the same study.

In Figures 6.5, 6.6, and 6.7, we demonstrate the results of posterior inference using the

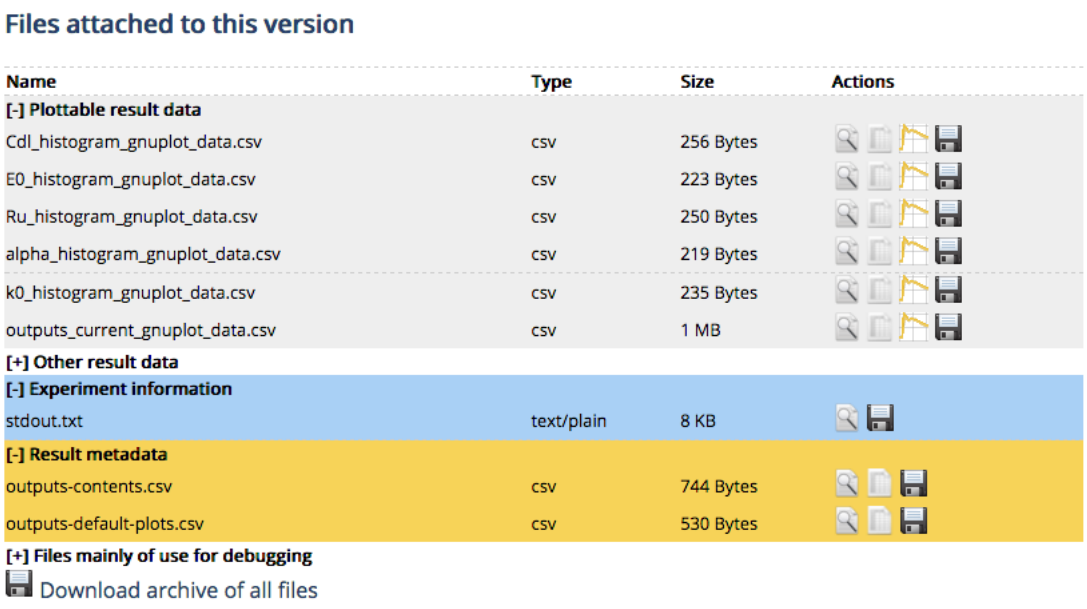
Table 6.5 Prior distributions over model parameters controlling Ferricyanide redox reactions (Equation 6.3.14), along with width of Gaussian observational error σ_e, used during fitting to voltammetric data.

Parameter	Prior Distribution
R_u	Uniform(0,755)
C_{dl}	Uniform(0,2×10^{-4})
E_0	Uniform(0.02,0.38)
k_0	Uniform(0,5)
α	Uniform(0,1)
σ_e	6.61×10^{-8}

Table 6.6 Settings for DC and AC voltammetry protocols used to collect experimental data from redox reactions of Ferricyanide compounds (Equation 6.3.14)

Parameter	DC Protocol Value	AC Protocol Value
T	297	297
A	0.07	0.07
E_{start}	-0.1	-0.1
E_{rev}	0.5	0.5
v	-0.08941	-0.08941
ΔE	0	0.08
ω	9.0152	9.0152
p	0	0

Figure 6.5 Files returned by the electrochemistry Modeling Web Lab after posterior inference over parameters in Table 6.3. Files listed under "Plottable result data" can be viewed as interactive plots by clicking the yellow graph icon, while all files may be downloaded by clicking the "save" icon or viewed within the browser by clicking the magnifying glass icon.



adaptive covariance MCMC algorithm (Algorithm 4.1) using 60,000 steps, with an initial point chosen by CMA-ES optimization and the initial 40,000 steps discarded as burn-in. In Figure 6.5, we see that, as in the cardiac Modeling Web Lab, the "Experiment" view page of the fitting experiment provides viewable representations of the marginal posterior distribution over each model parameter, as well as each protocol output (only I_{tot}, in this case). A full representation of the posterior estimate is also available as a CSV file under "Other result data."

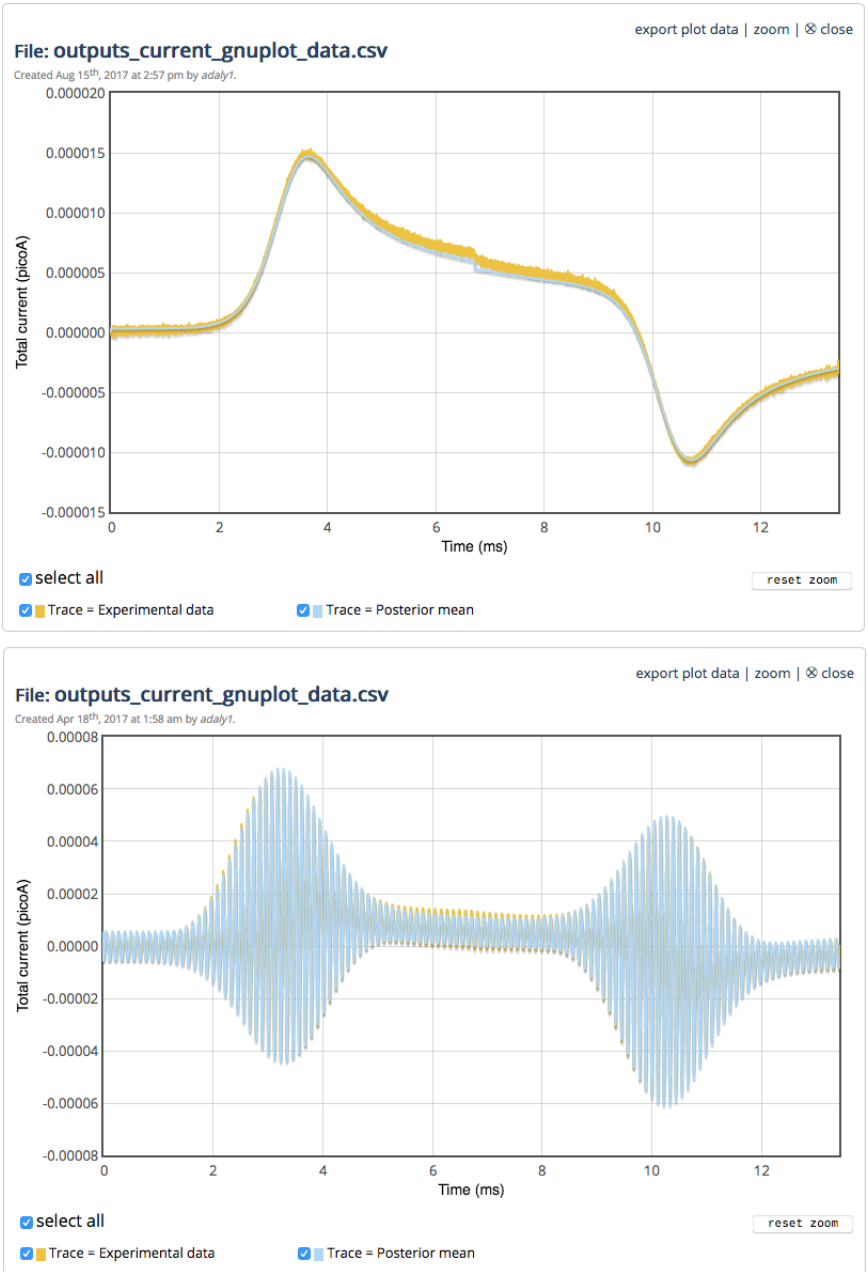
In Figure 6.6, we see the result of viewing the protocol output plot, which overlays the experimental data used to fit the model with the data simulated under the mean posterior parameters. For both DC and AC voltammetry experiments, we see that the mean predictive trace agrees closely with the experimental data, with a slight exception at the mid point of the trace when the protocol has attained E_{rev} and reverses the direction of the voltage sweep. This behavior produces a capacitative spike in experimental recordings at the switching point, which is considered by our experimental collaborators to be an artifact. Though it does not seem to affect the quality of fit for the remainder of the trace, this region can be excluded from future fitting exper-

iments by removing the data points that fall within some window of the switching time from the input data file, which would prompt the `ElectroChemExperiment` class to exclude these time points from simulated data as well.

We now turn our attention to the posterior distributions over the model parameters themselves, which are presented in a comparative view between DC and AC inference results in Figure 6.7. We have constructed these plots using data obtained from the "export data" functionality of the Modeling Web Lab for the sake of clear presentation within this thesis, although similar comparative plots can be viewed on the web site itself (an example of which is shown in Figure 6.8). We see in Figure 6.7 that with the exception of R_u, both DC and AC voltammetry experiments produce well-constrained Gaussian distributions over all model parameters. Although the Gaussian nature of the AC posteriors is more clearly visible in standalone plots due to the difference in scale of the DC posteriors, we have presented these comparative plots to demonstrate the relative locations of the two posteriors in parameter space. The asymmetric behavior of posteriors over R_u is a result of the CMA-ES minimization finding 0, the lower bound of the prior distribution, to be optimal for both experiments, leading the MCMC algorithm to explore a "half-Gaussian" region in parameter space bounded by this minimum value. As this behavior has been observed in multiple experimental repeats for both AC and DC data, we may safely assume that resistance in the system is truly negligible, as the results suggest.

While both experimental setups show posteriors with a clear, unique maximum, confirming the "correctness" of the model and sufficiency of the output to identify its parameters, we notice that the posteriors over all model parameters fit to AC data are shifted and noticeably more narrow when compared to those fit to DC data. The shift can be explained by unavoidable variability between experiments, which may be introduced by alterations to the electrode surface resulting from polishing between experiments (see McCreery (2008); Patel et al. (2012)), as well as random fluctuations in protocol conditions. The noticeable narrowing of the posteriors when employing AC data instead of DC data confirms the increased information content that is expected from these experiments. These comparative views, similar to those available on the

Figure 6.6 Experimental traces of I_{tot} overlaid with simulated data under mean posterior parameters for both DC (top) and AC (bottom) fitting experiments.



Modeling Web Lab itself (see Figure 6.8), clearly display this behavior to the user, showing approximately an order of magnitude of difference between the posterior uncertainty in all model parameters when fit to AC data instead of DC data.

Although data gathered from DC voltammetry protocols were sufficient to identify all model parameters, despite a lower information content relative to data from AC protocols, there are documented instances in which this is not the case. Gavaghan et al. (2017) detail how at extreme "true" values of k_0 (i.e., either highly reversible or highly irreversible reactions), DC voltammetry data may be insufficient to identify all model parameters, as evidenced by both wide, flat posteriors on k_0 and correlations between k_0 and other model parameters (notably R_u and α) indicating compensatory parameter interactions. In these cases, employing AC voltammetry data was shown to substantially decrease posterior uncertainty about k_0 and remove correlations, indicating that the complex protocol was able to identify the model by probing the system dynamics more completely. While parameter compensation effects were not observed in our study, due to the properties of the ferricyanide reaction considered, future iterations of the Modeling Web Lab will include tools for generating biplot visualizations (like those shown in Chapter 4) to aid in the assessment of model identifiability.

A final feature that we have developed for the electrochemistry Modeling Web Lab is the "Fit data" page, which further facilitates comparisons between modeling experiments. In this page, a user may enter any publicly available fitting specification into a template, adjust any and all of its variables via a graphical user interface, and save the result as a new fitting specification. This will allow users to perform alterations to prior specifications (such as adjusting ranges or fixing model parameters), upload new versions of experimental data, or even change algorithmic settings, with minimal interaction with the underlying file format. This feature can facilitate the update and adaptation of currently published models in response to newly available information, which is an ultimate goal of community development methods, and reproducible reporting standards in general.

Figure 6.7 Marginal posterior distributions produced over all redox reaction parameters when fitting to either DC or AC voltammetric data. The marginal posterior shown in each panel is indicated by the label above the graph, and the panel containing R_u provides an inset detail view so that the behavior under AC voltammetry is visible. Note that axis breaks have been included in most panels for clarity.

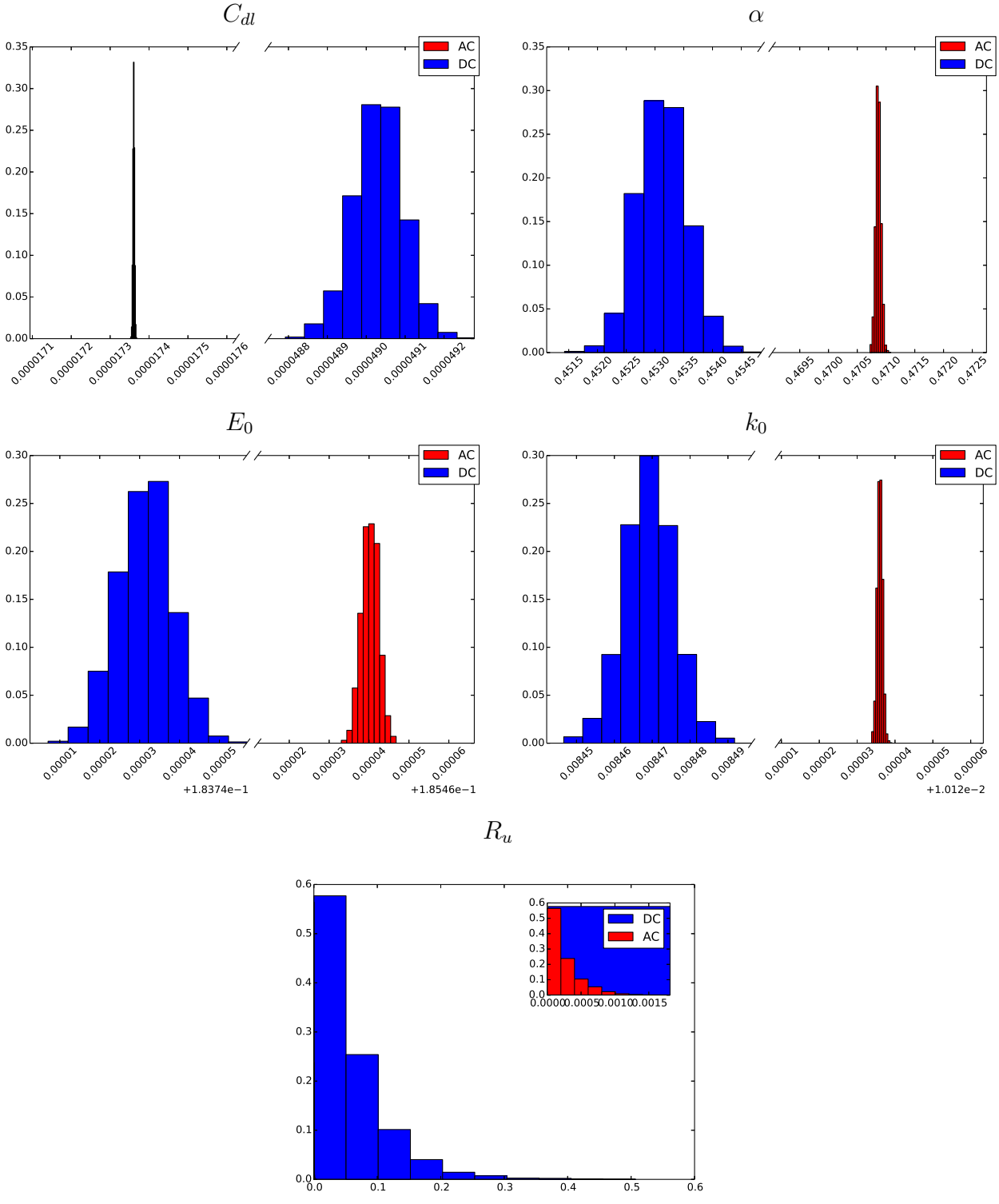


Figure 6.8 Modeling Web Lab comparison view between marginal posterior distributions over α parameter resulting from fitting to DC and AC voltammetry data.

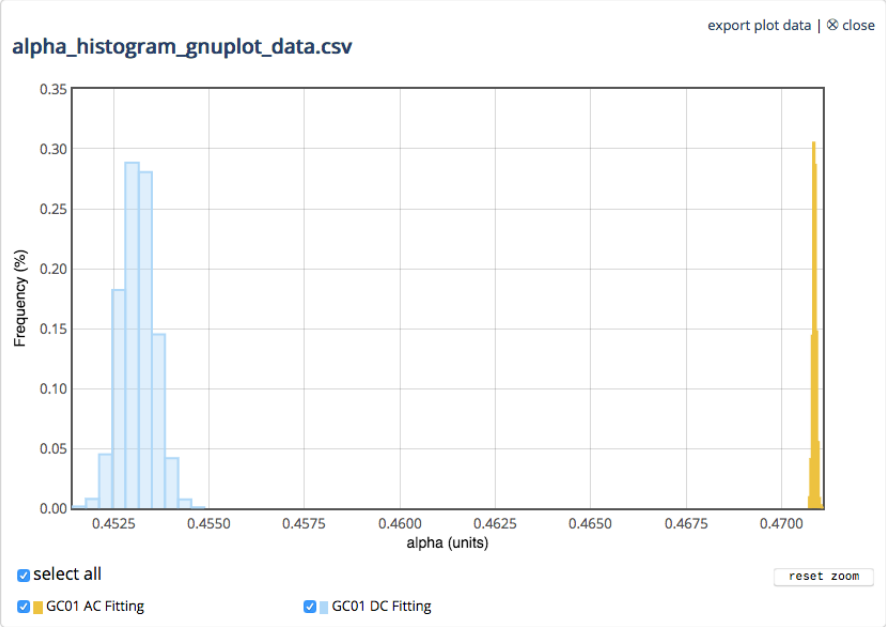
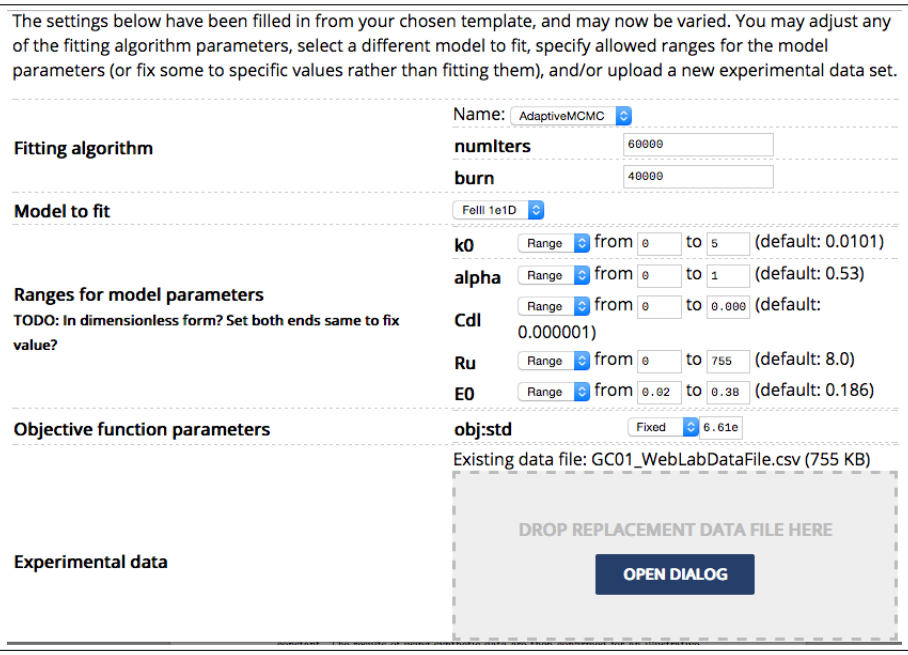


Figure 6.9 Interface for adaptively generating a fitting specification from an existing template in the electrochemistry Modeling Web Lab.



6.4 Discussion and Future Directions

In this chapter, we have presented a prototype implementation of the Modeling Web Lab, a community resource built upon the modeling framework we have developed for representing experiments in parameter fitting and identifiability analysis. We have demonstrated the ability of this tool to reproduce the results of a recent fitting experiment conducted on a Hodgkin-Huxley style hERG ion channel model, and highlighted the benefits that the Modeling Web Lab provides for visualization and sharing of results. We additionally demonstrated that this tool, as designed, can be easily adapted to new modeling domains, such as electrochemical redox reactions. Within the electrochemical Modeling Web Lab, we used the comparative visualization tools provided by the web site to demonstrate the higher information content of AC voltammetry experiments when compared to DC voltammetry, which validates a theoretically predicted result.

We note that, while the current prototype versions of the Modeling Web Lab presented in this chapter are limited in functionality (such as the number of algorithms made available), the degree of control given to the user over said algorithms, and the types of data that may be uploaded, the framework upon which they are built is capable of a wide range of expression, as we have detailed in Chapter 5, and thus extending functionality to meet user demand should be highly feasible. Future work will improve the visualization capabilities of both Modeling Web Lab versions, beginning with the goal of presenting enhanced renderings of posterior distributions (like the biplot visualizations presented in Chapter 4) in order to help in the *a posteriori* assessment of model identifiability. The manner in which modeling experiment results are presented will also be adjusted, as the current matrix view we have carried over from the cardiac electrophysiology Web Lab, which matches protocols to models, does not allow for variation of data or fitting protocols without re-uploading a full fitting specification. We have begun to address this with our prototype "fitting view" in the electrochemistry Modeling Web Lab, which allows users to create versions of fitting specifications in-browser, but future work in this depart-

ment will likely need to be guided by testing alternative designs on a focus group of modelers. Subsequent iterations of the Modeling Web Lab may separate the fitting protocol specification into model-specific (e.g., prior specifications), protocol-specific (e.g., input specifications) and algorithm-specific components to increase generalizability of a fitting specification across varying simulation conditions.

Finally, we note that there is room for development in the scope of models considered by both Modeling Web Labs. Work is currently being undertaken towards presenting all alternative model formulations considered by Beattie et al. (2017) in the Modeling Web Lab, and using comparative views of posterior inference results to determine which models were best identified by particular protocols. This will necessitate the development of comparative visualization tools for cases when competing models do not share common parameters, such as Hodgkin-Huxley or Markov models with varying numbers or configurations of states. In the electrochemistry Modeling Web Lab, the simple modeling language currently employed will likely be altered in order to allow for the specification of reaction modes other than the one-dimensional, one-electron models we are currently limited to. This would allow modelers to compare between competing formulations in a rigorous manner, and perform automated model selection based on properties such as goodness of fit and parameter identifiability. The range of data that the user may supply to fit the model may also be expanded, allowing for the use of either summary statistics or Fourier transforms of voltammetry data, which have been employed in previous modeling studies in this domain (Bard and Faulkner, 1980; Sher et al., 2004; Morris et al., 2013).

Having presented these prototypical resources that exemplify our goals for community modeling resources and incorporate the statistical tools we have developed earlier in this thesis, we will use the next chapter to review our main contributions and outline future directions for the progression of this research.

7

Conclusions

In this chapter, we will review the main contributions of this thesis, then discuss ways in which they may be extended through future research.

7.1 Summary

Predictive *in silico* models have become an undeniably important part of research in biological and chemical sciences. Not only do they promise to offer low-cost, low-risk alternatives to *in vivo* or *in vitro* experimental environments, but the iterated interaction between real-life experimentation and model construction has frequently given us insights into the operation of systems, particularly those that are too complex to observe directly. As the scope of these models expands, along with the number of researchers taking part in their development and use, it is important to reproducibly document the process of their creation, as well as providing data-driven metrics of trust in said models, so that they may be adapted to new systems or newly available data in an easy and informed manner.

For these reasons, we felt it was important to investigate and improve the way models were constructed in established fields of modeling such as cardiac electrophysiology. We were motivated to develop statistical tools for the assessment of model identifiability, a key indicator of the suitability of a given model formulation for a system of study, and to build online community resources that will make these types of tools available to a wide user base, promoting their

reproducibly documented application to contemporary systems of study. Within the last four chapters of this thesis, we have presented what we feel to be our main contributions towards this goal, which we will summarize below, considering their place in the context of current research, before discussing the manner in which they may be extended by future work.

We began in Chapter 3 by investigating the use of Bayesian and approximate Bayesian methods for parameterizing cardiac action potential models, due to the ability of these methods to simultaneously assess uncertainty about model parameters. Since in some circumstances only summary statistics of the raw model output are available for fitting, we began by presenting two sequential Monte Carlo samplers which we had generalized to perform either Bayesian or approximate Bayesian inference, allowing them to be applied over a wide range of observability conditions. We next demonstrated the ability of one of these solvers to recover the parameters of the classic Hodgkin-Huxley action potential, which both confirmed the precision of the original landmark study and demonstrated how these techniques may be used to validate model behavior in response to new data. This portion of the study has been published in Daly et al. (2015). We then performed a novel study on the modern O'Hara-Rudy action potential model using the side-by-side application of exact and approximate inference methods to assess the amount of information lost by employing summary statistic data. This is the first instance we are aware of in which the consequences of employing summary statistic data on the identifiability of an action potential model have been investigated statistically, and we believe that the methodology we presented in this paper, and published in Daly et al. (2017a), should inform the design of future modeling studies in this area.

In Chapter 4, we continued our investigation into the utility of Bayesian and approximate Bayesian methods for identifiability analysis by comparing their effectiveness against standard *a priori* approaches over a range of biologically relevant observational conditions. Through the investigation of a toy logistic model and the Hodgkin-Huxley model, we determined that, while *a priori* methods based on Fisher information had advantages in computational efficiency, their results were difficult to interpret in higher dimensional models, and they could not be employed

when observations were limited to complex summary statistic data. Moreover, we demonstrated the tendency of optimal experimental design strategies based on Fisher information indices to lead to practical unidentifiability in models fit to those data. Conversely, we showed the broad applicability of Bayesian and approximate Bayesian approaches, and outlined a means of designing maximally informative experiments based on an intuitive geometric interpretation of the results of posterior inference methods like those introduced in Chapter 3. The results of this study, currently under peer review in Daly et al. (2017b), help modelers to choose appropriate methods for assessing identifiability based on model structure experimental conditions, and present visual methods for optimal experimental design that are accessible to researchers with a wide range of statistical backgrounds.

Finally, in Chapter 5, we presented a framework for describing these types of modeling studies in a transparent, reproducible manner, and in Chapter 6 described a prototype community resource — the Modeling Web Lab — in which researchers may execute such experiments as well as interpret and share their results. We described the manner in which our framework is constructed to be capable both of a wide range of expression, as well as interfacing with existing standards of representation (such as the CellML modeling language for representing cardiac cell models) in order to present a low barrier to entry for members of a research community while being capable of presenting a wide range of functionality to them as demanded. We then detailed two variants of the Modeling Web Lab, both built upon this framework but catering to the differing modeling domains of cardiac electrophysiology and electrochemistry. We demonstrated the ability of the cardiac Modeling Web Lab to reproduce the results of a Bayesian study of the hERG ion channel model carried out by Beattie et al. (2017), which verifies the ability of this paradigm to represent complex, contemporary studies while also highlighting the visual and communal functionalities of the site. We further explored these capabilities using the electrochemistry Modeling Web Lab, which showed the ability of the Modeling Web Lab to assess relative information content between two competing voltammetry protocols (the results of which were published in Gavaghan et al. (2017)) as well as affirming the generality of the Mod-

eling Web Lab paradigm. We believe that these Web Labs, as they are continually developed, will serve as vital media for the storage and exchange of modeling studies, and additionally will promote the application of data-driven identifiability analyses by members of a variety of research communities.

7.2 Future Directions

Extensive work is already underway within our group to extend and build upon the research described in this thesis. As such, we will conclude with a discussion of current efforts and potential future avenues to build upon the research presented thus far.

The most immediate area for advancement of this research would be the optimization of algorithms currently implemented within the framework. Since we want this framework to be available as a library with wide support, it will be important to provide general implementations that will work efficiently on a variety of systems, but for the Modeling Web Lab back end implementation in particular, a machine-optimized version should be created for the server we choose to deploy it on. The Toni SMC algorithm in particular may benefit from an implementation that supports deployment on graphical processing units (GPUs), due to the algorithm's amenability to parallelization (discussed in Chapter 3). While the cost of running a simulation is generally what dominates the runtime of a modeling study, and as such the onus is placed on users to provide efficient implementations of simulation protocols, lowering the operational overhead of SMC algorithms in particular would allow users of the Modeling Web Lab to explore larger parameter spaces with greater efficiency, and to more easily perform the repeated inferences required for optimal experimental design algorithms such as those laid out in Chapter 4.

In addition to the improvements in functionality detailed at the end of Chapter 6, our research group will seek to develop the Modeling Web Lab in response to feedback from experimental collaborators across several modeling domains. In the cardiac domain, collaboration with the authors of Beattie et al. (2017) will continue with the goal of representing the full range of

competing models considered in their hERG study, and work towards improvements in the user interface and documentation of the Modeling Web Lab will be undertaken based on their feedback so that experimentalists working on similar studies can reproduce their results without our assistance. In the electrochemical domain, work will be undertaken with Dr. Alison Parkin at the University of York and Professor Alan Bond at Monash University to develop an environment akin to Chaste for representing simulations in electrochemistry, which will supplant our current operational language for representing such simulations in the Modeling Web Lab. Using this updated simulation environment, the electrochemistry Modeling Web Lab will be developed further with the goal of being able to present the results of modeling studies of enzyme electrochemical reactions fit to original experimental data. Finally, a collaboration with Dr. Lucia Sivolotti at UCL will be established in order to adapt the Modeling Web Lab paradigm to the domain of neuronal electrophysiology, with a focus on supporting inference over stochastic models such as the ones employed by their group to model the behavior of single ion channels within the cell membrane. From this diverse set of collaborators, our group hopes to gain insights into the range of algorithms, objective functions, and other components of parameter inference that will be most helpful to end users, as well as updating our design choices based on the results of controlled focus group studies, as we have done in the past for the cardiac electrophysiology Web Lab.

In future iterations of the Modeling Web Lab, we would also like to support the notion of rigorous model selection and experimental design. While approaches to model selection based on maximal posterior likelihood — such as Bayes information criteria, Akaike information criteria, or Bayes factors (see Akaike (1973) and Kass and Raferty (1995) for more details) — would be easy enough to provide alongside the results of parameter fitting experiments in the Modeling Web Lab, we believe that validation against new protocols, similar to our study of the Hodgkin-Huxley model in Section 3.4.3, provides a more robust and practical means of assessing a model's correctness. To this end, it will be important to support the propagation of posterior uncertainty from inference studies through new protocols, and comparison against

corresponding experimental data, to allow modelers to assess whether or not models generalize well to new conditions.

Since experimental design studies require iteration between data collection and identifiability assessment, with the goal of adjusting protocol settings to maximize identifiability, automated experimental design within the Modeling Web Lab requires that "experimental" data be updated periodically. To this end, we must either employ simulated data, which can be generated easily within our current framework, or connect our framework to a physical device that is capable of collecting experimental data. This would be difficult in domains such as cardiac electrophysiology, due to limitations in the lifetime of single cells *in vitro* and the number of protocols they may subjected to before death, but we could envision such an automated device for fields like electrochemistry, where, in theory, solutions and reagents may be automatically replaced (and electrodes automatically cleaned) for an arbitrary number of iterations. Development of such a device would require extensive collaboration between mechanical and software engineers to ensure that optimal design methods propose protocols that are within the limits of the physical system, and that both experiments and inference studies are carried out in a timely manner.

As with the work presented in this thesis as a whole, these research directions have been provided with both the present and future of community-based modeling studies in mind. We not only seek to improve cooperation and transparency in modeling studies, but also to develop a standard of trust in the results of these assays by promoting the application of validation and uncertainty quantification techniques. We are entering an era of research where cooperation is paramount to the advancement of our knowledge of the natural world, and hope that this work, and all work that stems from it, continues to serve the greater goal of promoting understanding through connection and collaboration.

Bibliography

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second international symposium on information theory*, 1973.

R.W. Aldrich, D.P. Corey, and C.F. Stevens. A reinterpretation of mammalian sodium channel gating based on single channel recording. *Nature*, 306:436–441, 1983.

S.P. Asprey and S. Macchietto. Designing robust optimal dynamic experiments. *J. Process Control*, 12:545–56, 2002.

H.T. Banks, K. Holm, and F. Kappel. Comparison of optimal design methods in inverse problems. *Inverse Probl.*, 27:1–31, 2011.

H.T. Banks, D. Rubio, N. Saintier, and M.I. Troparevsky. Optimal design techniques for distributed parameter systems. In *Proceedings of the Conference on Control and its Applications*, San Diego, California U.S.A., 2013. SIAM.

A.J. Bard and L.R. Faulkner. *Electrochemical Methods*. Wiley, 1980.

R. Bardenet. Monte Carlo methods. In T. Delemontex and A. Lucotte, editors, *EDP Sciences*, volume 55. IN2P3 School of Statistics, 2013.

Kylie A Beattie, Adam P Hill, Rémi Bardenet, Yi Cui, Jamie I Vandenberg, David J Gavaghan, Teun P de Boer, and Gary R Mirams. Sinusoidal voltage protocols for rapid characterization of ion channel kinetics. *bioRxiv*, 2017. doi: 10.1101/100677. URL `http://biorxiv.org/content/early/2017/03/13/100677`.

J. Beaumont, F.A. Roberge, and D.R. Lemieux. Estimation of the steady-state characteristics od the Hodgkin-Huxley model from voltage clamp data. *Math. Biosci.*, 1993.

Mylene Bedard. Optimal acceptance rates for Metropolis algorithms: moving beyonf 0.234. *Stoch. Process. Their Appl.*, 118(12):2198–2222, 2008.

L.K. Bieniasz and H. Rabitz. Extractions of parameters and their error distributions from cyclic voltammograms using bootstrap resampling enhanced by solution maps: computational study. *Anal. Chem.*, 78(24):8430–37, 2006.

L.K. Bieniasz and B. Speiser. Use of sensitivity analysis methods in the modelling of electrochemical transients: Part 3. Statistical error/uncertainty propagation in simulation and in nonlinear least-squares parameter estimation. *J. Electroanal. Chem.*, 458(1):209–29, 1998.

M.G.B. Blum, M.A. Nunes, D. Prangle, and S.A. Sisson. A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci.*, 28(2):189–208, 2013.

C.T. Bot, A.R. Kherlopian, F.A. Ortega, and T. Christini, D.J. Krogh-Madsen. Rapid genetic algorithm optimization of a mouse computational model: benefits for anthropomorphization of neonatal mouse cardiomyocytes. *Front. Physiol.*, 3:421, 2012.

J. Breidt, T. Butler, and D. Estep. A measure-theoretic computational method for inverse sensitivity problems I: Method and analysis. *SIAM J. Numer. Anal.*, 49(5):1836–1859, 2011.

O.J. Britton, Bueno-Orovio A., K. Van Ammel, H.R. Lu, and R. Towart. Experimentally calibrated population of models predicts and explains intersubject variability in cardiac cellular electrophysiology. *PNAS*, 2013.

M. Burger. Inverse problems in ion channel modeling. *Inverse Probl.*, 27:1–34, 2011.

T. Butler, D. Estep, and J. Sandelin. A computational measure theoretic approach to inverse sensitivity problems II: a posteriori error analysis. *SIAM J. Numer. Anal.*, 50(1):22–45, 2012.

T. Butler, D. Estep, S.J. Tavener, C. Dawson, and J. Westerink. A measure-theoretic computational method for inverse sensitivity problems III: Multiple quantities of interest. *SIAM J. Uncertain. Quantif.*, pages 1–27, 2014a.

T. Butler, D. Estep, S.J. Tavener, T. Wildey, C. Dawson, and L. Graham. Solving stochastic inverse problems using sigma-algebras on contour maps. *arXiv preprint arXiv:1407.3851*, 2014b.

R.C. Cannon and G. D'Alessandro. The ion channel inverse problem: neuroinformatics meets biophysics. *PLoS Comput. Biol.*, 2(8):862–869, 2006.

B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M.A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: a probabilistic programming language. *J. Stat. Softw.*, 76 (1), 2017.

A. Chakrabarty, G.T. Buzzard, and A.E. Rundell. Model-based design of experiments for cellular processes. *WIREs Syst. Biol. Med.*, 5:181–203, 2013. DOI: 10.1002/wsbm.1204.

F. Chen, A. Chu, X. Yang, Y. Lei, and J. Chu. Identification of the parameters of the Beeler-Reuter ionic equation with a partially perturbed particle swarm optimization. *IEEE Trans. Biomed. Eng.*, 59:3412–21, 2012.

E.M. Cherry and F.H. Fenton. A tale of two dogs: analyzing two models of canine ventricular electrophysiology. *Am. J. Physiol. Heart Circ. Physiol.*, 292:H43–H55, 2006.

O.T. Chis, J.R. Banga, and E. Balsa-Canto. Structural identifiability of systems biology models: a critical comparison of methods. *PloS ONE*, 6(11):e27755, 2011.

N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–551, 2002.

A. Cintron-Arias, H.T. Banks, A. Capaldi, and A.L. Lloyd. A sensitivity matrix based methodology for inverse problem formulation. *J. Inverse Ill-Posed Probl.*, 17:545–564, 2009.

C.E. Clancy and Y. Rudy. Cellular consequences of HERG mutations in the long QT syndrome: precursors to sudden cardiac death. *Cardiovasc. Res.*, 50(2):301–313, 2001.

J. Cooper, G.R. Mirams, and S.A. Niederer. High-throughput functional curation of cellular electrophysiology models. *Prog. Biophys. Mol. Biol.*, 107:11–20, 2011.

J. Cooper, J.O. Vik, and D. Waltemath. A call for virtual experiments: accelerating the scientific process. *Prog. Biophys. Mol. Biol.*, 117:99–106, 2015.

J. Cooper, M. Scharm, and G.R. Mirams. The cardiac electrophysiology web lab. *Biophys. J.*, 110(2):292–300, 2016.

C. Corrado, S. Williams, H. Chubb, M. O'Neill, and S.A. Niederer. *Personalization of Atrial Electrophysiology Models from Decapolar Catheter Measurements*, pages 21–28. Springer International Publishing, Cham, 2015.

C. Corrado, J. Whitaker, H. Chubb, S. Williams, M. Wright, , J. Gill, M. O'Neill, and S.A. Niederer. Personalized models of human atrial electrophysiology derived from endocardial electrograms. *IEEE Trans. Biomed. Eng.*, 2016.

A. Corrias, L. Romero, M.J. Bishop, M. Bernabeau, E. Pueyo, and B. Rodriguez. Arrhythmic risk biomarkers for the assessment of drug cardiotoxicity: from experiments to computer simulations. *Philos. Trans. A Math. Phys. Eng. Sci.*, 368(1921):3001–3025, 2010.

D. Csercsik, K.M. Hangos, and G. Szederkenyi. Identifiability analysis and parameter estimation of a single Hodgkin-Huxley type voltage dependent ion channel under voltage step measurement conditions. *Neurocomputing*, 77(1):178–88, 2012.

K. Csilléry, M.G.B. Blum, O.E. Gaggiotti, and O. François. Approximate Bayesian computation (ABC) in practice. *Trends in Ecology & Evolution*, 25(7):410–418, 2010.

M.E. Curran, I. Splawski, K.W. Timothy, G.M. Vincen, E.D. Green, and M.T. Keating. A molecular basis for cardiac arrhythmia: HERG mutations cause long qt syndrome. *Cell*, 80: 795–803, 1995.

A.C. Daly, D.J. Gavaghan, C. Holmes, and J. Cooper. Hodgkin-huxley revisited: reparameterization and identifiability analysis of the classic action potential model with approximate Bayesian computation. *J. R. Soc. Open Sci.*, 2(12), 2015.

A.C. Daly, J. Cooper, D.J. Gavaghan, and C. Holmes. Comparing two sequential Monte-Carlo samplers for exact and approximate bayesian inference on biological models. *J. R. Soc. Int.*, 14(134), 2017a.

A.C. Daly, D.J. Gavaghan, J.C. Cooper, and S.J. Tavener. Geometric assessment of parameter identifiability in nonlinear biological models. *Math. Biosci.*, 2017b. In Submission.

R. De Maesschalck, D. Jouan-Rimbaud, and D.L. Massart. The Mahalanobis distance. *Chemometrics Intell. Lab. Syst.*, 50(1):1–18, 2000.

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Statist. Soc. B*, 68(3):411–436, 2006.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential monte carlo method for approximate bayesian computation. *Stat. Comput.*, 22(5):1009–1020, 2012. ISSN 0960-3174.

S. Dokos and N.H. Lovell. Parameter estimation in cardiac ionic models. *Prog. Biophys. Mol. Biol.*, 85(2-3):407–31, 2004.

D.C. Drummond. Replicability is nor reproducibility: nor is it good science. 2009. Available: `http://cogprints.org/7691/`. Accessed 11 May 2017.

E. Dufresne, H.A. Harrington, and D.V. Raman. The geometry of sloppiness. 2016.

P. Fearnhead and D. Prangle. Constructing summary statistics for approximate Bayesian computation: semi-automatic ABC. 2011. arXiv:1004.1112.

M. Fink and D. Noble. Markov models for ion channels: versatility versus identifiability and speed. *Phil. Trans. Royal Soc. A*, 367:2161–2179, 2009.

A.C. Fisher. *Electrode dynamics*, volume 34 of *Oxford Chemistry Primers*. Oxford University Press, 1996.

M.R. Franz. Current status of monophasic action potential recording: theories, measurements and interpretations. *Cardiovasc. Res.*, 41(1):25–40, 1999.

M.R. Franz, D. Burkhoff, M.L. Spurgeon, M.L. Weisfeldt, and E.G. Lakatta. In vitro validation of a new cardiac catheter technique for recording monophasic action potentials. *Eur. Heart J.*, 7(1):34–41, 1986.

A. Garny, D. Nickerson, J. Cooper, R. Weber dos Santos, A.K. Miller, S. McKeever, P. Nielsen, and P.J. Hunter. CellML and associated tools and techniques. *Phil. Trans. R. Soc. A*, 366 (1878):3017–43, 2008.

D.J. Gavaghan, J. Cooper, A.C. Daly, C. Gill, K. Gillow, M. Robinson, A.N. Simonov, J. Zhang, and A.M. Bond. Use of Bayesian inference for parameter recovery in DC and AC voltammetry. *ChemElectroChem*, 2017. Online: http://dx.doi.org/10.1002/celc.201700678.

A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian data analysis*. CRC Press, Taylor and Francis Group, Florida, U.S.A., 3rd edition, 2014. ISBN: 978-1-4398-4095-5.

W. Groendaal, F.A. Ortega, A.R. Kherlopian, A.C. Zygmunt, T. Krogh-Madsen, and D.J. Christini. Cell-specific cardiac electrophysiology models. *PLoS Comput. Biol.*, 11, 2015. e1004242.

H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis Hastings algorithm. *Bernoulli*, 7(2):223–242, 2001.

J. V. Halliwell, T. D. Plant, J. Robbins, and N. B. Standen. *Microelectrode Techniques: The Plymouth Workshop Handbook*, chapter Voltage clamp techniques, pages 17–35. The Company of Biologists Limited, Cambridge, 1994.

J.C. Hancox, M.J. McPate, A. El Harchi, and Y.H. Zhang. The hERG potassium channel and hERG screening for drug-induced torsades de pointes. *Pharmacol. Ther.*, 119(2):118–32, 2008.

N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In X. Yao et al., editors, *Parallel Problem Solving from Nature PPSN VIII*, volume 3242 of *LNCS*, pages 282–291. Springer, 2004.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195, 2001.

W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

K.E. Hines, T.R. Middendorf, and R.W. Aldrich. Determination of parameter identifiability in nonlinear biophysical models: A Bayesian approach. *J. Gen. Physiol.*, 143(3):401–416, 2014.

A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117(4):500–544, 1952.

B. Howe. Virtual appliances, cloud computing, and reproducible research. *Comput. Sci. Eng.*, 14:36–41, 2012.

M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, , the rest of the SBML Forum:, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novre, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

T. Ino, H. Karagueuzian, K. Hong, M. Meesmann, W.J. Mandel, and T. Peter. Relation of monophasic action potential recorded with contact electrode to underlying transmembrane action potential properties in isolated cardiac tissues: a systematic microelectrode validation study. *Cardiovasc. Res.*, 22(4), 1988.

R.H. Johnstone, E.T.Y. Chang, R. Bardenet, T.P. de Boer, D.J. Gavaghan, P. Pathmanathan, R.H. Clayton, and G.R. Mirams. Uncertainty and variability in models of the cardiac action potential: can we build trustworthy models? *J. Mol. Cell. Cardiol.*, 96:49–62, 2016.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`. [Online].

A. Kadish. What is a monophasic action potential? *Cardiovasc. Res.*, 63(4):580–81, 2004.

R.E. Kass and A.E. Raferty. Bayes factors. *J Am. Stat. Assoc.*, 90(430):773–795, 1995.

J. Kaur, A. Nygren, and E.J. Vigmond. Fitting membrane resistance along with action potential shape in cardiac myocytes improves convergence: application of multi-objective parallel genetic algorithm. *PLoS One*, 9, 2014. e107984.

C. Kreutz, A. Raue, and J. Timmer. Likelihood based observability analysis and confidence intervals for predictions of dynamic models. *BMC Sys. Biol.*, 6:120, 2012.

T. Krogh-Madsen, E.A. Sobie, and D.J. Christini. Improving cardiomyocyte model fidelity and utility via dynamic electrophysiology protocols and optimization algorithms. *J. Physiol.*, 594 (9):2525–36, 2016.

D. Liang, M. Sulkin, Q. Lou, and I.R. Efimov. Optical mapping of action potentials and calcium transients in the mouse heart. *J. Vis. Exp.*, 55:3275, 2011.

S. Liu and R.L. Rasmusson. Hodgkin-Huxley and partially coupled inactivation models yield different voltage dependence of block. *Am. J. Physiol.*, 272:H2013–22, 1997.

C.M. Lloyd, M.D.B. Halstead, and P.F. Nielsen. CellML: its future, present, and past. *Prog. Biophys. Mol. Biol.*, 84(2-3):433–50, 2004.

C.H. Luo and Y. Rudy. A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction. *Circ. Res.*, 1991.

C.H. Luo and Y. Rudy. A dynamic model of the cardiac ventricular action potential. I. Simulations of ionic currents and concentration changes. *Circ. Res.*, 1994.

J.-M. Marin, P. Pudlo, C.P. Robert, and R.J. Ryder. Approximate Bayesian computational methods. *Stat. Comput.*, 22(6):1167–1180, 2012.

S. Marom and L.F. Abbott. Modeling state-dependent inactivation of membrane currents. *Biophys. J.*, 67(2):515–20, 1994.

R.L. McCreery. Advanced carbon electrode materials for molecular electrochemistry. *Chem. Rev.*, 108(7):2646–87, 2008.

M.M. McKerns and M. Aivazis. pathos: a framework for heterogeneous computing, 2010. http://trac.mystic.cacr.caltech.edu/project/pathos.

M.M. McKerns, L. Strand, T. Sullivan, A. Fang, and M.A.G. Aivazis. Building a framework for predictive science. In *Proceedings of the 10th Python in Science Conference*, 2011.

H. Miao, X. Xia, A. S. Perelson, and H. Wu. On identifiability of nonlinear ODE models and applications in viral dynamics. *SIAM Review*, 53(1):3–39, 2011.

L.S. Milescu, G. Akk, and F. Sachs. Maximum likelihood estimation of ion channel kinetics from macroscopic currents. *Biophys. J.*, 2005.

A. K. Miller, J. Marsh, A. Reeve, A. Garny, R. Britten, M. Halstead, J. Cooper, D. P. Nickerson, and P. F. Nielsen. An overview of the CellML API and its implementation. *BMC Bioinf.*, 11 (178), 2010.

G. Mirams, Y. Cui, M. Fink, J. Cooper, B.M. Heath, N.C. McMahon, D.J. Gavaghan, and D. Noble. Simulation of multiple ion channel block provides improved early prediction of compounds' clinical torsadogenic risk. *Cardiovasc. Res.*, 91(1):53–61, 2011.

G.R. Mirams, C.J. Arthurs, M.O. Bernabeau, R.F. Bordas, J. Cooper, A. Corrias, Y. Davit, S.J. Dunn, A.G. Fletcher, D.G. Harvey, M.E. Marsh, J.M. Osborne, P. Pathmanathan, and D.J. Gavaghan. Chaste: an open-source C++ library for computational physiology and biology. *PLoS Comput. Biol.*, 9(3), 2013.

D.E. Mohrman and L.J. Heller. *Cardiovascular Physiology*. McGraw-Hill, 7th edition, 2010. ISBN: 978-0-07-176652-4.

G.P. Morris, A.N. Simonov, E.A. Mashkina, R. Bordas, K. Gillow, R.E. Baker, D.J. Gavaghan, and A.M. Bond. A comparison of fully automated methods of data analysis and computer assisted heuristic methods in an electrode kinetic study of the pathologically variable $[\mathrm{Fe}(\mathrm{CN}_6)]^{3-/4-}$ process by AC voltammetry. *Anal. Chem.*, 84(24):11780–87, 2013.

K.W. Morton and D.F. Mayers. *Numerical Solutions of Partial Differential Equations: An Introduction*. Cambridge University Press, 2nd edition edition, 2005.

National Research Council. *Assessing the reliability of complex models: mathematical and statistical foundations of verification, validation, and uncertainty quantification.* The National Academic Press, Washington D.C., 2012.

R. Neal. Annealed importance sampling. *Statist. Comput.*, 11:125–39, 2001.

E. Neher and B. Sakmann. The patch clamp technique. *Sci. Am.*, 266(3):44–51, 1992.

M. Newville, T. Stensitzki, D.B. Allen, and A. Ingargiola. LMFIT: nonlinear least-square minimization and curve fitting for Python. DOI: 10.5281/zenodo.11813, 2014.

S.A. Niederer, M. Fink, D. Noble, and N.P. Smith. A meta-analysis of cardiac electrophysiology models. *Exp. Physiol.*, 64(5):486–495, 2009.

D. Noble. A modification of the Hodgkin-Huxley type equations applicable to Purkinje fibre action and pace-maker potentials. *J. Physiol.*, 1962.

D. Noble. From the Hodgkin-Huxley axon to the virtual heart. *J. Physiol.*, 580(1):15–22, 2007.

D. Noble and Y. Rudy. Models of cardiac ventricular action potentials: iterative interactions between experiment and simulation. *Phil. Trans. R. Soc. Lond. A*, 359(1793):1127–1142, 2001.

D. Noble, A. Garny, and P.J. Noble. How the Hodgkin-Huxley equations inspired the Cardiac Physiome Project. *J. Physiol.*, 590(1):2613–28, 2012.

T. O'Hara, L. Virag, A. Varro, and Y. Rudy. Simulation of the undiseased human cardiac ventricular action potential: model formulation and experimental validation. *PloS Comput. Biol.*, 7(5), 2011.

OMG Unified Modeling Language. OMG Object Management Group, 2.5 edition, 2015. http://www.omg.org/spec/UML/2.5/PDF/.

A.N. Patel, M.G. Collignon, M.A. O'Connell, W.O.Y. Hung, K. McKelvey, J.V. Macpherson, and P.R. Unwin. A new view of electrochemistry and highly oriented pyrolytic graphite. *J. Am. Chem. Soc.*, 134(49):20117–30, 2012.

P. Pathmanathan and R.A. Gray. Ensuring reliability of safety-critical clinical applications of computational cardiac models. *Front. Physiol.*, 2013.

P. Pathmanathan and R.A. Gray. Verification of computational models of cardiac electrophysiology. *Int. J. Meth. Biomed. Engng.*, 30:525–44, 2014.

A. Patil, D. Huard, and C.J. Fonnesbeck. PyMC: Bbayesian stochastic modeling in Python. *J. Stat. Soft.*, 35(4):1–81, 2010.

H.R. Polder, M. Weskamp, K. Linz, and R. Meyer. *Practical Methods in Cardiovascular Research*, chapter Voltage-Clamp and Patch-Clamp Techniques, pages 272–323. Springer, 2005. ISBN 978-3-540-26574-0.

D. Prangle. Summary statistics in approximate Bayesian computation. 2015.

A. Raue, C. Kreutz, T. Maiwald, J. Bachmann, M. Schilling, U. Klingmuller, and J. Timmer. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(15):1923 –1929, 2009.

A. Raue, C. Kreutz, T. Maiwald, U. Klingmuller, and J. Timmer. Addressing parameter identifiability by model–based experimentation. *IET Syst. Biol.*, 5(2):120–130, 2011.

A. Raue, J. Karlson, M.P. Saccomani, M. Jirstrand, and J. Timmer. Comparison of approaches for parameter identifiability analysis of biological systems. *Bioinformatics*, 30(10):1440–1448, 2014.

A. Raue, C. Kreutz, F.J. Theis, and J. Timmer. Joining forces of Bayesian and frequentist methodology: a study for inference in the presence of non-identifiability. *Phil. Trans. R. Soc. A*, 371(20110544), 2016.

L. Romero, E. Pueyo, M. Fink, and B. Rodriguez. Impact of ionic current variability on human ventricular electrophysiology. *Am. J. Physiol. Heart Circ. Physiol.*, 297(4):H1436–1445, 2009.

M. Roy, R. Dumaine, and A.M Brown. hERG, a primary human ventricular target of the nonsedating antihistamine terfenadine. *Circulation*, 94():817–823, 1996.

Y. Rudy and J.R. Silva. Computational biology in the study of cardiac ion channels and cell electrophysiology. *Q. Rev. Biophys.*, 39(1):57–116, 2006.

J Rumbaugh, I Jacobsen, and G Booch. *The Unified Modeling Language Reference Manual*. Pearson Higher Education, 2 edition, 2004.

A.A. Sher, A.M. Bond, D.J. Gavaghan, K. Harriman, S.W. Feldberg, N.W. Duffy, S.X. Guo, and J. Zhang. Resistance, capacitance, and electrode kinetic effects in Fourier-transformed large-amplitude sinusoidal voltammetry: emergence of powerful and intuitively obvious tools for recognition of patterns of behavior. *Anal. Chem.*, 76(21):6214–28, 2004.

S.A. Sisson, Y. Fan, and M. Tanaka. Sequential Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci.*, 2007.

S.A. Sisson, Y. Fan, and M. Tanaka. Sequential Monte Carlo without likelihoods: Errata. *Proc. Natl. Acad. Sci.*, 2009.

H. Suessbrich, S. Waldegger, F. Lang, and A.E. Busch. Blockage of hERG channels expressed in xenopus oocytes by the histamine receptor antagonists terfenadine and astemizole. *FEBS Lett.*, 385:77–80, 1996.

M. Sunnaker, A.G. Busetto, E. Numminen, J.C. Corander, M. Foll, and C. Dessimoz. Approximate Bayesian computation. *PLoS Comput. Biol.*, 9(1), 2013.

Z. Syed, E. Vigmond, S. Nattel, and L.J. Leon. Atrial cell action potential parameter fitting using genetic algorithms. *Med. Biol. Eng. Comput.*, 43:561–71, 2005.

D. Szekely, J.I. Vandenberg, S. Dokos, and A.P. Hill. An improved curvilinear gradient method for parameter optimization in complex biological models. *Med. Biol. Eng. Comput.*, 49(3): 289–96, 2011.

K.H.W. ten Tusscher, D. Noble, P.J. Noble, and A.V. Panafilov. A model for human ventricular tissue. *Am. J. Physiol. Heart Circ. Physiol.*, 2003.

K.H.W. ten Tusscher, O. Bernus, R. Hren, and A.V. Panfilov. Comparison of electrophysiological models for human ventricular cells and tissues. *Prog. Biophys. Mol. Bio.*, 90:326–345, 2006.

T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. R. Soc. Interface*, 6(31):187–202, 2009.

M.C. Vanier and J.M. Bower. A comparative survey of auromated parameter-search methods for compartmental neural models. *J. Comput. Neurosci.*, 7, 1999.

A.F. Villaverde and J.R. Banga. Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *J. R. Soc. Interface*, 11(91):20130505, 2014.

D. Waltemath, R. Adams, F.T. Bergmann, M. Hucka, F. Kolpakov, A.K. Miller, I.I. Moraru, D. Nickerson, J.L. Snoep, and N. Le Novere. Reproducible computational biology experiments with SED-ML - the Simulation Experiment Description Markup Language. *BMC Sys. Biol.*, 5(198), 2011.

G.J. Wang and J. Beaumont. Parameter estimation of the Hodgkin-Huxley gating model: an inversion procedure. *SIAM J. Appl. Math.*, 64(4):1249–67, 2004.

S. Wang, S. Liu, M.J. Morales, H.C. Strauss, and R.L. Rasmusson. A quantitative analysis of the activation and inactivation kinetics of HERG expressed in Xenopus oocytes. *J. Physiol.*, 502(1):45–60, 1997.

F.M. Weber, S. Lurz, D. Keller, D.L. Weiss, G. Seemann, C. Lorenz, and O. Dossel. Adaptation of a minimal four-state cell model for reproducing atrial excitation properties. In *Computers in Cardiology*, volume 35, pages 61–64, Piscataway, N.J., U.S.A., September 2008.

R.D. Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Stat. App. Gen. Mol. Bio.*, 12:129–141, 2013.

J.C. Wooley and H.S. Lin, editors. *Catalyzing inquiry at the interface of computing and biology*. National Academies Press, 2005. ISBN: 0-309-54937-X.

X. Xuyan, Y. Yingchun, and Y. Xiangqun. Markov chain inversion approach to identify the transition rates of ion channels. *Acta. Math. Sin.*, 32(5):1703–18, 2012.

T. Yu, C.M. Lloyd, D.P. Nickerson, M.T. Cooling, A.K. Miller, A. Garny, J.R. Terkildsen, J. Lawson, R.D. Britten, P.J. Hunter, and P.M. Nielsen. The Physiome Model Repository 2. *Bioinformatics*, 27(5):743–44, 2011.

Y. Yuan, X. Bai, C. Luo, K. Wang, and H. Zhang. The virtual heart as a platform for screening drug cardiotoxicity. *Br. J. Pharmacol.*, 172(23):5531–5547, 2015.