Algorithmic Parameter Estimation and Uncertainty Quantification for Hodgkin–Huxley Neuron Models

Y. Curtis Wang^{1*}, Nirvik Sinha², Johann Rudi^{3*}, James Velasco¹, Gideon Idumah⁴, Randall K. Powers⁵, Charles J. Heckman², Matthieu Chardon^{2*}

*For correspondence:

ywang11@calstatela.edu (YCW);

jrudi@anl.gov (JR);

m-chardon@northwestern.edu (MC)

¹Department of Electrical and Computer Engineering, California State University, Los Angeles, CA, USA; ²Interdepartmental Neuroscience, Northwestern University, Chicago, IL, USA; ³Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL, USA; ⁴Department of Mathematics, Case Western Reserve University, Cleveland, OH, USA; ⁵Department of Physiology and Biophysics, University of Washington, Seattle, WA, USA

Abstract Experimental data-based parameter search for Hodgkin–Huxley-style (HH) neuron models is a major challenge for neuroscientists and neuroengineers. Current search strategies are often computationally expensive, are slow to converge, have difficulty handling nonlinearities or multimodalities in the objective function, or require good initial parameter guesses. Most important, many existing approaches lack quantification of uncertainties in parameter estimates even though such uncertainties are of immense biological significance. We propose a novel method for parameter inference and uncertainty quantification in a Bayesian framework using the Markov chain Monte Carlo (MCMC) approach. This approach incorporates prior knowledge about model parameters (as probability distributions) and aims to map the prior to a posterior distribution of parameters informed by both the model and the data. Furthermore, using the adaptive parallel tempering strategy for MCMC, we tackle the highly nonlinear, noisy, and multimodal loss function, which depends on the HH neuron model. We tested the robustness of our approach using the voltage trace data generated from a 9-parameter HH model using five levels of injected currents (0.0, 0.1, 0.2, 0.3, and 0.4 nA). Each test consisted of running the ground truth with its respective currents to estimate the model parameters. To simulate the condition for fitting a frequency-current (F-I) curve, we also introduced an aggregate objective that runs MCMC against all five levels simultaneously. We found that MCMC was able to produce many solutions with acceptable loss values (e.g., for 0.0 nA, 889 solutions were within 0.5% of the best solution and 1,595 solutions within 1% of the best solution). Thus, an adaptive parallel tempering MCMC search provides a “landscape” of the possible parameter sets with acceptable loss values in a tractable manner. Our approach is able to obtain an intelligently sampled global view of the solution distributions within a search range in a single computation. Additionally, the advantage of uncertainty quantification allows for exploration of further solution spaces, which can serve to better inform future experiments.

38 Introduction

39 Systematic parameter exploration: an unmet need for model building in neuro- 40 science

41 Can we measure everything needed to build a model?

42 Complete specification of models in neuroscience or systems physiology requires identification of
43 several parameters. For single neurons, these typically include geometric and electrical proper-
44 ties of the cell body, dendrites, and axons. At the network level, one needs to additionally spec-
45 ify the synaptic connectivity patterns along with the temporal and spatial properties of individual
46 synapses. Even with the sophistication of modern experimental techniques, however, measuring
47 all the necessary parameters is almost always impossible. Additionally, many single-neuron and
48 network-level properties exhibit remarkable context-dependent variability, even within the same
49 animal. For example, changes in the neuromodulatory environment alter the spiking dynamics of
50 spinal motoneurons (*Heckman et al., 2009*) and thalamocortical neurons (*Pape and McCormick,*
51 *1989*) by influencing their ionic conductances. Likewise, changes mediated by long-term potenti-
52 ation or depression at individual synapses propagate selectively throughout the network (*Fitzsi-*
53 *monds et al., 1997*). Thus, any single measurement at one time instance would be insufficient to
54 inform all single-neuron/network properties because it cannot account for the wide spectrum of
55 behaviors observed in vivo.

56 To complicate matters further, redundancy in biological systems leads to similar activity profiles
57 that can be produced by many different neurons or neuronal networks with dissimilar properties
58 (*Schulz et al., 2007; Roffman et al., 2012; Swensen and Bean, 2005*). For example, lateral pyloric
59 neurons in the crab stomatogastric ganglion exhibit as much as two- to fourfold interanimal vari-
60 ability in three different ion channel densities and their corresponding mRNA levels (*Schulz et al.,*
61 *2006*). The stomatogastric ganglia can also exhibit identical network activity from widely disparate
62 synaptic strengths and neuron properties (*Prinz et al., 2004*). Therefore, even averaging a param-
63 eter from multiple experimental preparations may fail to generate the desired behavior in the
64 computational models constructed from them (*Golowasch et al., 2002*). Systematic exploration of
65 the parameter space is obligatory to fit a neuron or a neuronal network model to experimental
66 data. Additionally, this systematic exploration may reveal deeper insights and motivate future ex-
67 periments by unraveling undiscovered parameter combinations that might reproduce the same
68 experimentally observed behavior.

69 Is manual tuning of model parameters still practicable?

70 Fitting experimental data to neuron models has been a major challenge for neuroscientists. In fact,
71 the parameters of the first biologically realistic quantitative description of the neuronal membrane
72 were hand fitted by *Hodgkin and Huxley (1952)*. This approach has remained popular; for exam-
73 ple, the peak conductances of ion channels in a model of an elemental leech heartbeat oscillator
74 were hand tuned by *Nadim et al. (1995)*. Likewise, both the single compartment and network pa-
75 rameters of a small group of neurons used to model the crustacean pyloric network were hand
76 tuned by *Soto-Treviño et al. (2005)* to reproduce a variety of experimentally observed behaviors.
77 However, the ever-increasing dimensionality and complexity of the neuron models (Hodgkin and
78 Huxley's description contained only three membrane currents), accompanied by the concomitant
79 increase in computational resources available at the disposal of neuroscientists, have made hand
80 tuning practically obsolete. Besides, hand tuning of parameters also introduces experimenter bias,
81 because different parameters are assigned preconceived roles during the tuning process (*Van Geit*
82 *et al., 2008*). Nevertheless, hand tuning to a certain extent may be unavoidable because the initial
83 range of values of different parameters over which the automated search is to be performed must
84 be determined by the experimenter based on physiological constraints.

Computational inference for neural models

The body of research on parameter estimation for models of neural dynamics for single cells or circuits spans across various scientific communities, approaches, and neuron models. A gap exists, however, between researchers with rich physiological knowledge, on the one hand, and researchers working on new solution algorithms for inference, on the other hand. This gap presents opportunities to create a bridge between physiologists, engineers, computational scientists, and mathematicians by exploring existing and developing new inference techniques for neuron dynamics of single cells and circuits.

Multiple realizability: the dilemma of many truths

Single-neuron and network-level activities that are amenable to easy measurement (e.g., spike trains, voltage traces, local field potentials) can often be identical for different parameter combinations. Just as this situation renders experimental determination of parameter values infeasible, it also poses an incredible challenge to the computational neuroscientist trying to systematically infer a model from the experimental data. For example, *Prinz et al. (2004)* found indistinguishable network activity from widely disparate deterministic models. Similarly, *Amarasingham et al. (2015)* reported statistically indistinguishable spike train outputs for different statistical processes that model the firing rate of networks. *Hartoyo et al. (2019)* stressed that, for models of dynamical systems, very different parameter combinations can generate similar predictions/outputs. Additionally, the authors showed that sensitivities of predicted parameters can exhibit wide variability, leading to the conclusion that modeling in neuroscience is confronted by the major challenge of identifiability of model parameters. The demonstrative model used in our present work shows a bursting neuron producing similar neural activities from multiple sets of conductance densities (*Alonso and Marder, 2019*).

Previous approaches

Computational approaches for estimating parameters of neuron models based on ordinary differential equations (ODEs) include brute-force grid search (*Prinz et al., 2003*) as well as more advanced techniques using heuristics or trial-and-error approaches, which may consist of intricate sequences of regression steps (*Achard and De Schutter, 2006; Van Geit et al., 2007, 2008; Buhry et al., 2011*). The latter include simulated annealing, differential evolution, and genetic algorithms. These approaches have the disadvantage of slowly converging to an optimal set of parameters and thus being computationally expensive. Using gradient-based optimization to accelerate convergence (*Doi et al., 2002; Toth et al., 2011; Meliza et al., 2014*), on the other hand, can suffer from the nonconvexity and the strong nonlinearities in the objective manifold and require good initial guesses in order not to get stuck in local minima (thereby resulting in suboptimal parameter sets). Another shortcoming of recovering only one set of parameters (also called point estimates) is the lack of knowledge about the uncertainties or error bounds for the inferred parameters.

Attempts at using machine learning techniques based on artificial neural networks (ANNs) have recently shown promising results. For instance, in the context of the FitzHugh–Nagumo model (*Rudi et al., 2021*), an ANN was constructed and optimized to generate an inverse map that is able to predict model parameters from observational data. *Bittner et al. (2019)* developed generative models from deep learning. Retrieving associated uncertainties for inferred parameters has been performed by *Gonçalves et al. (2020)*, using normalizing flows with Gaussian mixtures in order to generate approximations of posterior distributions for Hodgkin–Huxley-based inverse problems. However, machine learning techniques still suffer from the cost of generating large sets of data for training the artificial neural networks, where each training sample requires the numerical solution of a differential equation.

Theoretical neuroscientists have been developing and utilizing statistical methods for inference and uncertainty quantification (*Van Geit et al., 2007; Vavoulis et al., 2012*) and have relied on Bayesian inference frameworks (*Ahmadian et al., 2011; Doruk and Abosharb, 2019*) in order to

estimate parameters and quantify uncertainties of the recovered parameters. The uncertainties in recovered parameters with Bayesian inference frameworks are represented by posterior density functions, which describe a “landscape” of more likely parameters (“peaks in the landscape”) and less likely parameters (“valleys in the landscape”; see *Methods* for details). Bayesian likelihoods or their approximations are constructed, which in turn enables direct maps to posterior densities without numerical solutions of neuron models (i.e., differential equations). *Chen (2013)* provides an overview of Bayesian methods for neural spike train analysis. Recently, *René et al. (2020)* and *Schmutz et al. (2020)* have utilized Markov chain Monte Carlo for inference from spike train data of population models.

Using Markov chain Monte Carlo algorithms for inference

MCMC algorithms play a critical role in our approach for parameter estimation. For a brief introduction and background information about MCMC algorithms, we refer the reader to the *Methods* section.

Benefits and limitations of using MCMC in parameter searches

One of the main benefits of using MCMC is the detailed picture of the posterior “landscape” that it can provide. By analyzing the posterior, the uncertainty in the inferred parameters can be quantified, and the dependencies between different parameters in the model can be identified. Another benefit is that MCMC does not require derivatives of neuron models or the loss function for misfits of the data and model output. This is important in the case of the complex model and nondifferentiable loss function proposed by *Alonso and Marder (2019)*, which we also use in the present work. Furthermore, the key advantage of parallel tempering MCMC is its ability to expose multimodality in the posterior (see *Methods* section for more details).

Arguably, possible limitations of MCMC may arise if the search space for the parameters becomes too large; that is, if too many parameters in the model need to be inferred. With increasing dimensions of the parameter space, increasingly more iterations of MCMC may be required, eventually leading to the algorithm becoming computationally prohibitive. In the inference that we are targeting in this work, the number of inferred parameters remains at amounts where these limitations do not occur.

Applications of MCMC in biological modeling

MCMC-based methods have generally been a popular choice for solving inverse problems in a statistical framework, where one is interested in uncertainties in addition to optimal solutions (*Smith, 2013*). In the context of dynamic systems in biology, MCMC techniques have been successfully employed, as described in review articles by *Ballnus et al. (2017)* and *Valderrama-Bahamo (2019)*. Moreover, tempering MCMC methods have been shown to recover the multimodality of solutions in systems biology (*Caranica et al., 2018; Gupta et al., 2020*). However, in the context of neural dynamics with complex Hodgkin–Huxley models, such as in *Alonso and Marder (2019)*, which we target in this work, MCMC methods have not been attempted. The reason may be the computational challenges described above, for instance, multimodal posteriors.

Objectives and contributions of this work

In this study we address the issue of multiple realizability of models in neuroscience. Specifically, we target the computational inference for HH-based neuron models with a state-of-the-art MCMC algorithm. Our goal is to estimate the parameters in such models and quantify their uncertainties.

We design the parameter estimation problem in a Bayesian framework: (i) parameter sets in a model are treated as probability distributions rather than one single optimal set; and (ii) the distributions of parameter sets can exhibit dependencies between individual values of the set, rather than assuming each parameter to be independent from the others. This Bayesian framework for parameter estimation does not assume that a single independent optimum can be reached. The

likelihood is given implicitly in the form of a loss function between observational data and model output, hence requiring the numerical solution of the model for each new set of parameters. As a solution algorithm we utilize an extension MCMC, the parallel tempering MCMC method, which is critical for recovering distributions of parameter sets that are multimodal (i.e., multiple sets of parameters of a neuron model are recovered for a single set of observational data). Further, we visualize solution maps of the multimodal posterior that enable us to quantify which parameters can reproduce the observational data more closely.

Results

The results of our study show that our design of the inverse problem combined with parallel tempering MCMC for solving the inverse problem is able to successfully overcome the inference challenges by using a systematic approach. This MCMC algorithm can recover multiple parameter sets where data and model output are consistent. The algorithm requires only an initial setup of the prior information (i.e., assumptions on the distributions of the parameter values), and it especially does not require manual intervention, such as for widening or narrowing bounds for parameter values, during the search. Parallel tempering MCMC returns distributions of sets of parameters, which we refer to as *solution maps* of the posterior. With these solution maps physiologists can investigate the results from MCMC in order to quickly decide which parameter sets are indeed relevant from a physiological perspective. This is a main advantage of *solution maps* compared with *solution points* (i.e., a single set of “optimal” parameters) obtained from optimization algorithms.

Inference proof of concept with a Hodgkin–Huxley model

We illustrate the results of the parallel tempering MCMC method in a simple inverse problem setup, which nevertheless exhibits the difficulties of more complex inference setups in neural dynamics. We use a relatively simple and well-known HH model (*Hodgkin and Huxley, 1952; Mainen et al., 1995*) and the commonly used mean squared error (MSE) as the loss function to measure the fit between data and model outputs. To set up the inverse problem, we simulate the voltage trace for a given set of sodium and potassium conductances, denoted as the “true” parameter set, which is $g_{Na} = 200 \text{ pS}/\mu\text{m}^2$ for sodium and $g_K = 50 \text{ pS}/\mu\text{m}^2$ for potassium. This voltage trace serves as the data for our inverse problem.

The first goal of the inverse problem is to recover the true sodium and potassium conductances. To demonstrate the challenges of the problem, we visualize the loss function that would have to be minimized to find the optimum in *Figure 1A*, where the colors indicate the (positive) loss value. The true parameter set is located in a “valley” in the loss “landscape,” whereas the loss is large for parameters that produce large discrepancies between model output and data such as shown in graph S1 corresponding to point S1 in *Figure 1A*. However, multiple sets of parameters exist that are local minima in this landscape, and these multiple local minima pose major challenges because optimization algorithms may present one of them as the “optimal” solution. These local minima will not give a sufficiently good fit of model outputs vs. data, as is illustrated in graph S2 of *Figure 1* that corresponds to point S2 in the landscape (*Figure 1A*). Such local minima are known to be problematic for numerical optimization algorithms (*Nocedal and Wright, 2006*).

The second goal of the inverse problem is to, additionally, quantify the uncertainty with respect to these parameters and understand the sensitivities of the parameters. For instance, in *Figure 1*, the graphs S3 and S4 illustrate how different parameters produce voltage traces similar to the data, while at the same time this uncertainty in the parameters is visible in the landscape (*Figure 1A*) as a valley containing points S3 and S4. In order to recover these uncertainties, the formulation of the inverse problem in a Bayesian framework becomes critical.

The numerical solution of the Bayesian inverse problem with parallel tempering MCMC successfully provides a probability density spanned by the two-dimensional parameter space. The density is large, where the loss function in *Figure 1A* has its major valley. We show the progress of the MCMC algorithm in *Figure 1B* as the number of collected samples increases from 2,000 to 8,000.

Figure 1. Loss function corresponding to the inverse problem with a simple HH model and how MCMC is able to successfully recover optimal parameters and quantify uncertainties and parameter trade-offs. A: Loss function landscape (colors) over the parameters, sodium (g_{Na}) and (g_K) potassium conductances. The “true” parameter set represents a global minimizer of the loss; the valley of the loss along points S3 and S4 (in dark blue color of loss) translates to parameter uncertainties (or trade-offs). Graphs S1–S4 depict voltage traces at corresponding points in A. B: Densities of MCMC samples, where the dark purple color of highest density overlaps with the true parameter set. As the number of MCMC samples increases, the method recovers the valley of the loss landscape A and hence quantifies uncertainties in the parameters.

As the sample count (i.e., iteration count of MCMC) grows, the algorithm generates a longer tail along the valley, showing a clearer picture of the parameter uncertainties. The true parameter values are clearly visible in the high-density region as a dark blue area around 200 and 50 $pS/\mu m^2$ for sodium and potassium conductances, respectively. Furthermore, the tail to the upper right of the true parameter set shows the trade-off between sodium and potassium, when model outputs keep being consistent with data even though the values of the parameters deviate from the truth. These results serve as a proof of concept that the parallel tempering MCMC algorithm can successfully tackle multimodal losses. Next we transition to a more complex model for neural dynamics, which will be the main result of this paper.

Inference with the complex Alonso-Marder (AM) model

AM model – Posterior distribution

MCMC samples the posterior distribution to generate a solution map of the posterior. Since we have nine uncertain parameters that we want to infer, the posterior is a distribution in a nine-dimensional space. To visualize the nine-dimensional space, we consider one or two parameters at a time, where the remaining parameter dimensions are marginalized (i.e., summed up). The plots in **Figure 2** show the solution map of the posterior. The denser regions within **Figure 2** are parameter value sets that return lower loss function values; therefore they represent better fits between data and model outputs.

Along the diagonal of **Figure 2** are the histograms for the (1D marginal) distribution for each parameter. For the 2D marginals in the lower triangle of **Figure 2**, the points have the opacity set to darker where their density is higher. This setup allows for easier visualization of trends within the posterior distribution with regard to parameter values. Note that the denser points for a particular parameter (g_{Na} , for example) are not necessarily in the denser regions for another parameter (g_{Kd} , for example), if one considers different plots. Such links between two parameters can be established only when g_{Na} , g_{Kd} are plotted along the two axes of the same 2D marginal.

AM model – Individual constraint

We ran MCMC sampling on the AM model using individualized currents (0.0 to 0.4 nA at increments of 0.1 nA). This dataset is called the “Individual Constraint.” We also ran the MCMC search as a single aggregated search with all currents at once, which is discussed in a subsequent section. Below we present results from the individual constraints.

The MCMC search produced solution maps of the posteriors for each of the individual currents. Within each of the maps is a parameter set with the lowest loss value **Table 1**. In the traditional thinking of optimization, this set can be considered to be the optimal solution.

In addition to producing a series of parameter sets with low loss values, the MCMC algorithm is able to produce the loss function map (posterior distribution) for all 9 parameters of the model within the search range and injected currents (0.0 to 0.4 nA, 0.1 nA increments).

Table 1. Parameter sets for the lowest loss values calculated by the MCMC algorithm. The range of the parameter priors is also provided in the first row. The units for the conductances are in $pS/\mu m^2$.

Injected current (nA)	g_{Na}	g_{CaT}	g_{CaS}	g_A	g_{KCa}	g_{Kd}	g_H	g_L	τ_{ca}	Loss
uniform priors	[800, 2000]	[1, 10]	[1, 15]	[1, 200]	[1, 20]	[1, 200]	[0.01, 0.5]	[0.01, 0.5]	[100, 900]	-
0.0	1484.10	9.54	11.07	78.73	18.13	144.03	0.027	0.119	767.54	0.0008
0.1	1543.74	9.90	14.27	73.17	19.74	171.16	0.067	0.149	768.77	0.0002
0.2	1768.69	6.64	13.52	118.36	18.01	68.42	0.096	0.075	739.20	0.0348
0.3	1461.32	6.63	10.14	86.07	10.41	62.39	0.425	0.193	452.63	0.0013
0.4	1973.44	4.06	9.93	168.56	1.24	29.14	0.221	0.183	206.94	0.0047
aggregate	1335.84	9.18	14.97	102.72	18.84	142.91	0.363	0.163	802.44	8.8109
ground truth	1076.39	6.41	10.05	8.04	17.58	124.09	0.113	0.176	653.50	-

Figure 2. Final solution map of the posterior for the 9 parameters of the AM model for the **individual constraint** of the 0.0 nA injected current. The diagonal plots show the 1D marginals of each parameter. The x-axis is the search range of the parameter, and the y-axis is the normalized distribution. The remaining plots are the 2D marginals of pairs of parameters. The x- and y-axes are the range of the pairs of parameters. Each point within each plot is a sample found by the MCMC algorithm. The colors indicate the associated loss value of this sample, where the purple color indicates a lower loss value and the yellow color indicates a higher loss value. Density of points will contribute to the darkness of an area. The more pronounced a color signifies that more points are overlaid on top of each other. For example, in the top left 2D marginal [g_{CaT} , g_{Na}] the dark purple line around [g_{CaT} , g_{Na}] $\sim$ [9, 1200] is an accumulation of many overlaid points. This density can also be traced back to the 1D marginals to the top and the left side.

We summarized these findings in a large arrangement of plots in **Figure 2** for 0.0 nA infected current and for currents 0.1–0.4 in the Appendix. The rows and columns of subplots are each associated with one of the parameters (g_{Na} , g_{CaT} , g_{CaS} , g_A , g_{KCa} , g_{Kd} , g_H , g_L , and τ_{Ca}). As such, **Figure 2** resembles a triangular matrix, where the upper right triangle is redundant because it is symmetric to the lower left triangle.

1D Marginals: The diagonal portion of the subplot matrix shows the one-dimensional marginals (distribution histograms) of the solutions for each of the individual parameters. The x-axis represents the range of values, and the y-axis represents the probability of that value producing a good fit between data and model outputs. These distributions provide both the likeliest solutions (i.e., the peaks) as well as their uncertainty (distribution around the peaks). For example, the top left 1D marginal of the sodium conductance g_{Na} has 4 peaks. From those peaks, one can quickly quantify the uncertainty of each of those probable solutions.

2D Marginals: The remaining subplots are the two-dimensional marginals of the posterior distributions. Each row and column represent a parameter θ of the model, and each of the points within the subfigure is a solution found by the MCMC algorithm. To show the probability distribution within the solution space, we set the alpha value (the transparency of the color) of each solution to $\alpha = 10^{-3}$ (where opaque is $\alpha = 1$). A highly probable region has many more points and therefore will appear more opaque. The color of each point represents the loss value ranging from blue to yellow (low to high).

These plots show the dependence of each parameter on each other. Some of these dependencies are linear (for example, [row, col]: [g_{Kd} , g_{Na}]), and some are nonlinear (for example, [row, col]: [g_{Kd} , g_{CaT}]). Some of the linear dependencies are vertical and horizontal, signifying that one parameter is independent from the other (for example, vertical [g_{Kd} , g_{CaT}], [g_H , g_{Na}] and horizontal [τ_{Ca} , g_{CaS}]).

AM model – Aggregate constraint

To test the versatility of the MCMC algorithm at solving the inverse problem for HH-type equations, we combined all the input currents into a single analysis called “aggregate constraint” analysis (more details given in **Methods**).

Just as shown for the individual constraints, the aggregate constraint also have a solution map of the posterior. This map is shown in **Figure 3** where the 1D marginals and 2D marginals are illustrated. The same analysis can be performed to determine the peaks and variations of individual parameters using the 1D marginals and the dependence of two parameters using the 2D marginals. Compared with the individual constraint, the loss values are higher; hence the yellow colors dominate in these maps.

We found that the aggregate constraint solution map encompasses a subset of the individual constraint solution maps. To illustrate this, we show the intersection (multiplication of the posterior) of the kernel density estimates (KDEs) of the solution maps from the aggregate constraint and the two individual constraints 0.0 nA and 0.2 nA as an example in **Figure 4**. First we calculated the intersection between the two individual constraints to extract the common set of solutions between the two. This first intersection can be interpreted as the solution map if only 0.0 nA and 0.2 nA were aggregated. We then calculated the intersection of the aggregate map with the latter solution.

One major result flows from this analysis. The distinctive features (lines, clumps, ...) found in the individual constraints are found in the intersection even though they were indiscernible in the aggregate constraint results. This underlines the efficacy of the MCMC algorithm in solving for a solution map under the aggregate constraint.

In the next section we formalize these intersections using the Wasserstein distance metric.

Figure 3. Solution maps of the posterior for the 9 parameters of the AM model for the **aggregate constraint**. The diagonal plots show the 1D marginals of each parameter. The x-axis is the search range of the parameter, and the y-axis is the normalized distribution. The remaining plots are the 2D marginals of pairs of parameters. The x- and y-axes are the range of the pairs of parameters. Each point within each plot is a sample found by the MCMC algorithm. The colors indicate the associated loss value of this sample, where purple color indicates a lower loss value and yellow color indicates a higher loss value. The density of the points will contribute to the darkness of an area. More pronounced color signifies that more points are overlaid on top of each other. This density can also be traced back to the 1D marginals to the top and the left side.

Figure 4. Two-dimensional (g_{Na} and g_{Kd}) kernel density estimates (KDEs) of the solution maps of different cases, for visualization of the relationship between the aggregate constraint and the individual constraints. The arrows indicate the flow of the operation. The KDEs of the 0.0 nA and 0.2 nA individual constraints are on the left. The KDE of the aggregate constraint is shown on the right. The intersection is the elementwise multiplication of two or more KDEs. The final intersection between 0.0 nA, 0.2 nA, and aggregate constraint shows the contribution of the 0.0 and 0.2 nA individual constraints to the aggregate constraint. The colormap scales have been adjusted so that values are visible.

AM model – Distances between posterior distributions

To demonstrate the differences between the posterior distributions of the AM model between an individual constraint and the aggregate constraint, we computed the approximate Wasserstein distance for each individual injected current's posterior distribution taking into account all nine parameters as well as individual parameters against the aggregate constraint's posterior distribution. The approximate Wasserstein distance provides a quantitative metric of the total cost required to transform one probability distribution to another probability distribution (*Givens and Shortt, 1984*). The results are displayed in **Table 2**, with the rightmost column showing the Wasserstein distance of the entire posterior distribution (all nine parameters used) for a specific injected current. Overall, the posterior distribution of no injected current (0.0 nA) yields the closest posterior distribution to the aggregate constraint, followed by 0.2 nA, then 0.1 and 0.3 nA.

The parameter's posterior distributions that are the most different between an individual constraint and the aggregate constraint are the distributions for the parameter g_{Na} and to a lesser extent for g_A , g_{Kd} , and τ_{Ca} . This observation for the Wasserstein distances is consistent with the 1D marginal solution maps of the posterior when comparing the aggregate constraint in **Figure 3**

Table 2. Wasserstein distances for each posterior (individual parameters and all parameters) of individually sampled injected currents against the aggregated constraint's posterior.

Injected current (nA)	g_{Na}	g_{CaT}	g_{CaS}	g_A	g_{KCa}	g_{Kd}	g_H	g_L	τ_{Ca}	All parameters
0.0	44.644	0.818	0.272	4.588	0.505	6.093	0.057	0.018	11.961	99.183
0.1	45.854	0.686	0.447	15.687	1.077	8.696	0.059	0.019	26.163	110.521
0.2	42.552	1.025	0.343	18.422	1.373	6.009	0.059	0.026	34.762	104.880
0.3	59.637	0.523	0.384	19.405	0.558	5.315	0.043	0.053	17.832	109.133
0.4	35.752	0.734	0.397	18.584	0.367	11.825	0.035	0.053	33.112	104.520

Figure 5. Each row of this figure shows the voltage traces for the ground truth (top orange), for the best solution sets given the individual constraints (middle blue) and the best solution set given the aggregate constraint (bottom grey). Each column shows the traces for different injected currents. Loss values for these parameter solution sets are 0.0008 for the 0.0 nA individual constraint, 0.0348 for the 0.2 nA individual constraint, and 8.8109 for the aggregate constraint.

with **Figure 2** and the additional solution maps of the posterior for other individual currents in the Appendix.

Discussion

Alonso and Marder (2019) presented a complex and realistic neurological model and proposed visualization techniques in order to help understand how different parameter sets can have similar model outputs. They identified the challenge to find multiple parameter sets that are optimal in the sense that they generate good fits between data and model outputs. Consequently, this raises the need to develop algorithmic approaches in order to find these multiple sets. In the present work we design the inverse problem in a Bayesian framework, where multiple optimal parameters are part of a single multimodal posterior distribution. Furthermore, we utilize parallel tempering MCMC in order to recover the multiple modes in the posterior and visualize them in the solution maps. As a result, we have filled an important need in the field of neuroscience.

Interpreting solution maps and choosing parameter sets

The solution space that the MCMC algorithm provides can be overwhelming especially when one is familiar with the classical gradient approach to an inverse problem (i.e., a single solution). Here, the algorithm instead provides a space of solutions from which one can pick a solution.

Single-Solution Approach: As detailed in the Results, a parameter set exists that will produce the lowest loss value for each of the individual and aggregate constraints (see **Table 1**). Many sets of solutions also will provide extremely low loss values. For instance, we found, for the solution space for the injected current of 0.0 nA, a total of 889 solutions within 0.5% of the best solution and a total of 1,595 solutions with 1% of the best solution (see **Table 3**).

Even though the loss values are small for many parameter sets, the voltage traces produced by these sets are different. To illustrate these differences, in **Figure 6** we plotted the first five voltage traces in order of the loss value from best to worse with respect to the ground truth shown at the top in orange. The change in loss value is smaller than 10^{-2} and nearly identical for the last two traces d and e. These traces are visibly different, however, thus illustrating that MCMC is able to produce different parameter sets for a nearly identical loss value (posterior distribution). This result highlights how this ODE neuron model can produce multiple solutions as shown by **Alonso and Marder (2019)**.

Figure 6. First 5 solutions for the injected current of 0 nA. These solutions are ranked from “best” to “worst” (a–e) according to their loss. Even a nearly identical loss can produce a qualitatively different voltage trace, as shown by traces d and e

Injected current (nA)	0.5%	1.0%
0.0	889	1595
0.1	916	1618
0.2	21	118
0.3	120	233
0.4	149	265

Table 3. Number of solutions given the injected current as a function of percentage difference from the best solution. For instance, for a 0.0 nA injected current, 889 solutions are 0.5% away from the best 0.0 nA solution. MCMC is able to find these multiple acceptable solutions. **Figure 6** visualizes an example set of voltage traces generated from these solution parameter sets.

357 **Multimodal Approach:** Many more nearly-optimally fitting solutions exist, however. One can
 358 look for multiple modes or sets of possible solution regions: this is illustrated in the 1D and 2D
 359 marginals described in the Results. In the 1D marginals one can quickly gauge the important values
 360 within a range by the different peaks observed within the distribution as well as its sensitivity by the
 361 width/spread under peaks. Furthermore, the size of the range can be determined if distributions
 362 appear to abruptly stop at the boundaries. For instance, in the 1D marginal for parameter g_{KCa}
 363 found in **Figure 2**, it appears that the range was too small as the distribution begins a descent
 364 at the upper bound. Such insight can lead the modeler to reconsider the size of the range of
 365 the parameters. The 2D Marginals provide a fast method to determine which parameters are
 366 highly correlated. Taken together, one can begin to look at solution areas that previously were left
 367 unexplored, perhaps opening new avenues for investigation.

368 **Impact of loss function on recovered parameters**

369 The impact of the choice of loss function becomes apparent in **Figure 5**. There we see the ground
 370 truth voltage trace generated from the AM model with the ground truth set of parameters (orange
 371 color). This trace serves as the data in our inverse problem. To measure the fit between data
 372 and model outputs, we consider a particular loss function that is derived from the loss function
 373 employed by **Alonso and Marder (2019)**. After running the MCMC algorithm to recover the param-
 374 eters of the model, we select the set of parameters with the lowest loss value and plug them into
 375 the model to generate the voltage traces, shown in blue color in **Figure 5**. Comparing the orange
 376 (ground truth) and the blue trace graphs, we observe discrepancies between the traces that may
 377 appear as inadequate fits from a physiological perspective. These discrepancies are caused by the
 378 loss function, which defines a distance metric between traces. In essence, a possible limitation of
 379 the chosen loss function is that it quantifies disparate voltage traces as too similar by giving it a
 380 low loss value. This demonstrates the difficulty of finding appropriate losses for inverse modeling
 381 in neuroscience and is one main reason for manual and heuristic approaches for solving inverse
 382 problems governed by neural dynamics models.

383 To address this issue, **Alonso and Marder (2019)** invested efforts to refine the loss to specific
 384 inference setups that are targeted. This approach can be prone to human bias and is labor in-
 385 tensive. An alternative approach is taken in the present work because we aim at automated and
 386 algorithmically guided techniques. The idea is to augment the measure for fitting data and model
 387 outputs with additional information. Here we propose to augment the loss function with additional
 388 voltage traces that are generated at different input currents. This approach results in the inverse
 389 problem setup with aggregated current constraint, and we observe better-behaved solutions from
 390 an inference perspective, shown as the gray voltage traces in **Figure 5**. An alternative approach to
 391 the aggregate current constraint would be to design new loss functions (for individual currents),
 392 which is out of the scope of this work.

393 **Physiological meaning**

394 Extracting physiological information from the solution maps is the ultimate goal of this exercise.
 395 For instance, one can quickly rank the importance of a parameter by looking at the 2D marginals.
 396 Looking at the bottom row of **Figure 2**, we observe that τ_{Ca} is highly correlated to g_{Kd} while g_{CaT}
 397 and g_{CaS} are more independent of τ_{Ca} for the injection current constraint 0.0 nA. Therefore this
 398 MCMC analysis shows that the calcium component of the afterhyperpolarization (AHP) needed to
 399 replicate the ground truth spiking traces is dominated by the time constant τ_{Ca} . To be able to reach
 400 such a conclusion with traditional optimization techniques would require a sensitivity analysis.

401 Alonso and Marder showed that multiple solutions exist for solving for the same neural out-
 402 come. MCMC complements this finding by showing that a much larger solution set can be found.
 403 We believe that this multimodal solution set is the norm rather than the exception. As models in-
 404 crease in complexity and therefore increase in parameters, the likelihood of a multimodal solution
 405 will increase. For instance, although studies of variability of neuronal behavior have concentrated

on the role of ion channel density (e.g., *Marder and Goaillard (2006)*), changes in the voltage dependence of channels resulting from changes in phosphorylation (*Park et al., 2006*) or the binding of accessory proteins (*Bosch et al., 2015*) are likely to be equally important. The power of Bayesian inference together with MCMC is that these additional variables can be easily taken into account.

MCMC sampling operates within the bounds based on our prior assumptions such as our general knowledge, literature review, or colleagues. The MCMC algorithm “tests” these prior assumptions based on data and model by looking at the shape of the posterior, as mentioned above. Peaks at the boundaries of the ranges are likely to indicate too narrow a range of sampled parameter values (see parameters g_A and g_{KCa} in *Figure 2*). This could lead experimentalists to look beyond their prior assumptions.

Methods

Details of the AM model

For the simulations governed by the AM model in *Results*, the bursting neuron ODE system from *Alonso and Marder (2019)* was used. The parameters used for the ground truth were from Model A of *Alonso and Marder (2019)* for consistency (conductances in uS, time constants in ms): $g_{Na} = 1076.392$, $g_{CaT} = 6.4056$, $g_{CaS} = 10.048$, $g_A = 8.0384$, $g_{KCa} = 17.584$, $g_{Kd} = 124.0928$, $g_H = 0.11304$, $g_L = 0.17584$, and $\tau_{Ca} = 653.5$.

The BDF and LSODA solvers for ODEs from the SciPy library (*SciPy 1.0 Contributors et al., 2020; Petzold, 1983*) were utilized for the stiff ODE systems of the AM model. BDF and LSODA were used because of their overall convergence performance compared with that of other ODE solvers for stiff biological ODE models. LSODA has been shown to perform better in biological systems than many modern ODE solvers (*Städter et al., 2021*). The voltage trace generated by the ground truth parameters is shown in *Figure 5* in orange color.

Loss function for measuring fit between data and model outputs

The main loss function used in the present study involving the AM model is the loss in Equation 4 in *Alonso and Marder (2019)*, where we use the following coefficients: $\alpha = 1000$, $\beta = 1000$, and $\gamma = \delta = \nu = 0$. This design of the loss has been highly informed by domain expertise, because it targets specific properties of voltage traces. We purposefully eliminate several components of the loss (i.e., $\gamma = \delta = \nu = 0$). Rather than separating between bursters and non-bursters, our study uses the same loss function with fixed coefficient values throughout all experiments, because this allows us to compare inference results and it avoids possible human bias. Our approach is different from *Alonso and Marder (2019)*, where the coefficients in the loss are adjusted depending on the inference setup. Our goal is to develop inference methods that can be carried out in an automated fashion. Here we focus on the two components of the loss function that we observed to be dominant, associated with α , the burst frequency mismatch, and with β , the duty cycle mismatch. Our objective is to test the multimodality of the inverse problem with parallel tempering MCMC methods.

Loss function for individual and aggregate constraint

We use the loss function (detailed in *Loss function for measuring fit between data and model outputs*) to solve inverse problems, where one particular injected current is present; these studies are referred to as the individual current constraint. The individual current constraint (or “individual constraint” in short) is when the loss function value for a single current injection value (0.0 nA, 0.1 nA, ...) is used for measuring the fit between data and model outputs.

In addition, we carry out experiments with multiple injected currents, referred to as the aggregate current constraint. The aggregate current constraint (or “aggregate constraint” in short) is when the maximal loss function value for the entire set of injected currents ($\{0.0, 0.1, 0.2, 0.3, 0.4\}$ nA) is used for the misfit between data and model outputs. The aggregate current constraint is

motivated by its similarity to fitting against a frequency-current (F-I) curve (*Hultborn and Pierrot-Deseilligny, 1979*).

Introduction and background to MCMC

In this work we choose the Bayesian formulation (also called Bayesian framework), which is centered around Bayes' theorem:

$$P(\theta | y) = \frac{P(y | \theta)P(\theta)}{P(y)}, \quad (1)$$

where θ denotes the parameters of the ODE model and y is the data. On the left-hand side of (1), $P(\theta | y)$ is the unknown *posterior* (i.e., conditional probability for the model parameters θ given the data y). On the right-hand side of (1), $P(y | \theta)$ is the known but often intractable *likelihood* (i.e., conditional probability for the data y given the model parameters θ). The likelihood term is responsible for measuring the fit between data and model outputs. Hence the loss function, which we detailed above, is evaluated each time the likelihood is computed. Further on the right-hand side, $P(\theta)$ is the known and typically tractable *prior* (i.e., probability of the model parameters θ). The term $P(y)$ is called *evidence* (i.e., total probability of the data y); and since it is constant with respect to θ , it simply scales the posterior by a constant and can in practice remain unknown.

The first goal of MCMC algorithms, such as the Metropolis–Hastings algorithm (*Metropolis et al., 1953; Hastings, 1970*), is to find the most likely model parameters, θ , that will produce the greatest posterior (the highest probability of the model parameters to reproduce the data). The second goal is to quantify parameter uncertainties. To this end, the algorithm visits smaller locations in the solution map of the posterior (i.e., lower probabilities of the model parameters to reproduce the data). However, these visits to lower-probability locations of the posterior happen adaptively and automatically at smaller frequencies. The concept of MCMC visiting locations of high probability at greater frequencies and locations of lower probability at smaller frequencies is a key property of the algorithm.

In detail, at each iteration of MCMC, the algorithm first proposes a new value for (one or multiple) parameters, θ_{proposal} , by randomly choosing from a proposal distribution (e.g., a normal distribution centered at the value of the previous iteration θ_{previous}). The likelihood $P(y | \theta_{\text{proposal}})$ and prior $P(\theta_{\text{proposal}})$ are evaluated at the proposed value and multiplied to obtain a new value of the posterior. In the second MCMC iteration, the algorithm includes the proposed θ_{proposal} in a sequence of visited points or discards it, where the aim is to more frequently keep points of higher probability. To determine whether θ_{proposal} is accepted or rejected relative to the previous θ_{previous} , the algorithm takes the ratio of the proposed posterior to the current posterior. If the ratio is greater than one, θ_{proposal} must have a higher probability, and therefore this new parameter is saved because it is producing a better outcome. If the ratio is below one, then the algorithm applies a randomized rejection criterion such that some θ_{proposal} is stored at lower frequencies. Subsequently, this cycle repeats, where θ_{proposal} becomes the new θ_{previous} , if acceptance was successful. Note that in this algorithm, dividing the new value by the previous value of the posterior, $P(y)$ cancels out, thus showing that constant scaling factors are irrelevant for MCMC.

The Metropolis–Hastings algorithm constructs a sequence, also called a chain, of parameters (also called samples) that it visits during the iterations. This method is suitable for unimodal posterior distributions. However, it is not designed to capture multimodal posteriors (i.e., solution maps of the posterior with multiple peaks). Therefore, extensions of MCMC have been developed to deal with multimodal distributions. Early work by *Marinari and Parisi (1992)*; *Geyer and Thompson (1995)* employed simulated tempering, which is akin to simulated annealing algorithms from the field of optimization. More recently, parallel tempering MCMC (*Łqcki and Miasojedow, 2016*; *Vousden et al., 2016*) was proposed. This builds on ideas from simulated tempering and, additionally, uses multiple chains of single-chain methods, such as Metropolis–Hastings, in parallel. The key idea of tempering, which enables MCMC to discover multimodalities in posteriors, is that the posterior is taken to a power with an exponent, γ , between zero and one: $0 < \gamma < 1$. Hence we

501 obtain $P'(\theta | y)$. This has the effect that smaller locations in the solution map of the posterior are
502 elevated and can be visited more frequently by Metropolis–Hastings.

503 **MCMC method for estimating parameters in AM model**

504 This section provided details about our implementation for estimating parameters and their uncer-
505 tainties in AM models. Our AM model and MCMC methods constitute the following components:

- 506 • Neuron model component: Computes voltage traces for the AM model and a given set of
507 candidate parameters
- 508 • Likelihood component: Computes the Bayesian likelihood using the loss function based on
509 the voltage trace obtained by neuron model component
- 510 • Prior component: Evaluates the prior based on the candidate set of parameters
- 511 • Estimator component: Proposes candidate parameters at random (the proposal sampler)

512 The sampler used was the adaptive parallel tempering sampler (*Miasojedow et al., 2013*) within
513 the PyPESTO library (*Stapor et al., 2018*). Parallel tempering runs independent Markov chains at
514 various temperatures and performs swaps between the chains. Each individual chain runs adaptive
515 Metropolis–Hastings MCMC.

516 The loss functions for individual and aggregate constraints are evaluated for each candidate
517 parameter in the MCMC algorithm within the likelihood term. The prior term is defined as a uniform
518 distribution (i.e., unbiased prior) within predetermined boundaries, which were chosen sufficiently
519 wide to permit physiologically relevant model parameters.

520 **Experimental setup**

521 To investigate the robustness of parallel tempering MCMC on the inverse problem governed by the
522 AM model, we use model A from *Alonso and Marder (2019)*, which is defined by the parameters
523 that are listed in *Details of the AM model*, as ground truth data. We use this model to generate five
524 voltage traces each using different injected currents: 0.0, 0.1, 0.2, 0.3, and 0.4 nA. Noise to simulate
525 physical measurement noise was not added to the voltage trace of the ground truth, for several
526 reasons. To have an experimental setup similar to *Alonso and Marder (2019)*, we also did not alter
527 the ground truth trace data. Furthermore, MCMC methods are typically not detrimentally affected
528 by noisy data. Further, the Bayesian framework together with MCMC as a solution algorithm allows
529 noise to be modeled using an additional parameter (and prior) within the inverse problem. This is
530 a possible direction for future research.

531 The conductances for the ion channels and the time constant for the calcium channel are the
532 nine parameters that we target for inference. Each inference with individual current constraint
533 consisting of running the MCMC algorithm for a predetermined number of iterations/samples. Ad-
534 ditionally, for the inference with aggregate current constraint, a single run of MCMC was performed.
535 Each MCMC run used 24 to 32 parallel chains and 10,000 to 14,000 samples per chain. Geweke’s
536 convergence diagnostic was used to determine the burn-in of each chain (*Stapor et al., 2018*). The
537 burn-in is an initial phase of MCMC where the gathered samples do not satisfy certain statistical
538 properties to be considered adequate samples of the posterior. In certain cases, despite parallel
539 tempering, there are MCMC chains that can remain static through the parameter space. Results
540 from chains that did not move sufficiently (ratio of standard deviation to mean) were filtered out
541 in postprocessing.

542 **Software improvements for accelerating parallel tempering MCMC**

543 We reduced the computational time of the MCMC inference by carrying out various optimizations
544 of the source code. Increasing the number of Markov chains benefits the detection and exploration
545 of multimodal posteriors; however, it also increases the computation time. To shorten the com-
546 putation time, we parallelized the MCMC algorithm to run on each CPU core concurrently using
547 multiprocessing packages (*conda-forge community, 2015*). This approach significantly decreased

computation time because of the parallel nature of the chains. The total runtime for each MCMC run was between 24 and 48 hours. We utilized three different machine architectures: (1) the Broadwell nodes (36-core, 128 GB RAM) on the parallel cluster Bebop at Argonne National Laboratory, (2) a workstation equipped with a dual 16-core Intel Xeon 4216 CPU (total of 32 cores and 64 threads) with 128 GB of RAM, and (3) a workstation with a 32-core (64-thread) AMD Ryzen 3970x CPU with 128 GB of RAM.

Acknowledgments

We gratefully acknowledge the computing resources provided on Bebop, a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory.

The efforts of M.C., C.J.H., R.K.P, and N.S. was supported in part by the NIH R01 NS109552. The effort of G.I. was supported by the NSF Mathematical Sciences Graduate Internship. The effort of J.R. is based in part upon work supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357. The effort of Y.C.W. was supported in part by the Center for Energy and Sustainability at California State University, Los Angeles, and the National Science Foundation under Award No. HRD-1547723.

References

- Achard P**, De Schutter E. Complex Parameter Landscape for a Complex Neuron Model. *PLoS Computational Biology*. 2006 Jul; 2(7):e94. <https://dx.plos.org/10.1371/journal.pcbi.0020094>, doi: 10.1371/journal.pcbi.0020094.
- Ahmadian Y**, Pillow JW, Paninski L. Efficient Markov Chain Monte Carlo Methods for Decoding Neural Spike Trains. *Neural Computation*. 2011 Jan; 23(1):46–96. <https://direct.mit.edu/neco/article/23/1/46-96/7628>, doi: 10.1162/NECO_a_00059.
- Alonso LM**, Marder E. Visualization of currents in neural models with similar behavior and different conductance densities. *eLife*. 2019 Jan; 8:e42722. <https://elifesciences.org/articles/42722>, doi: 10.7554/eLife.42722.
- Amarasingham A**, Geman S, Harrison MT. Ambiguity and nonidentifiability in the statistical analysis of neural codes. *Proceedings of the National Academy of Sciences*. 2015 May; 112(20):6455–6460. <http://www.pnas.org/lookup/doi/10.1073/pnas.1506400112>, doi: 10.1073/pnas.1506400112.
- Ballnus B**, Hug S, Hatz K, Görlitz L, Hasenauer J, Theis FJ. Comprehensive benchmarking of Markov chain Monte Carlo methods for dynamical systems. *BMC Systems Biology*. 2017 Dec; 11(1):63. <http://bmcsystbiol.biomedcentral.com/articles/10.1186/s12918-017-0433-1>, doi: 10.1186/s12918-017-0433-1.
- Bittner SR**, Palmigiano A, Piet AT, Duan CA, Brody CD, Miller KD, Cunningham JP. Interrogating theoretical models of neural computation with emergent property inference. *Neuroscience*; 2019.
- Bosch MK**, Carrasquillo Y, Ransdell JL, Kanakamedala A, Ornitz DM, Nerbonne JM. Intracellular FGF14 (iFGF14) is required for spontaneous and evoked firing in cerebellar Purkinje neurons and for motor coordination and balance. *Journal of Neuroscience*. 2015; 35(17):6752–6769. Publisher: Soc Neuroscience.
- Buhry L**, Grassia F, Giremus A, Grivel E, Renaud S, Saïghi S. Automated Parameter Estimation of the Hodgkin-Huxley Model Using the Differential Evolution Algorithm: Application to Neuromimetic Analog Integrated Circuits. *Neural Computation*. 2011 Oct; 23(10):2599–2625. https://www.mitpressjournals.org/doi/abs/10.1162/NECO_a_00170, doi: 10.1162/NECO_a_00170.
- Caranica C**, Al-Omari A, Deng Z, Griffith J, Nilsen R, Mao L, Arnold J, Schüttler HB. Ensemble methods for stochastic networks with special reference to the biological clock of *Neurospora crassa*. *PLOS ONE*. 2018 May; 13(5):e0196435. <https://dx.plos.org/10.1371/journal.pone.0196435>, doi: 10.1371/journal.pone.0196435.
- Chen Z**. An Overview of Bayesian Methods for Neural Spike Train Analysis. *Computational Intelligence and Neuroscience*. 2013; 2013:1–17. <http://www.hindawi.com/journals/cin/2013/251905/>, doi: 10.1155/2013/251905.
- conda-forge community**, The conda-forge Project: Community-based Software Distribution Built on the conda Package Format and Ecosystem. Zenodo; 2015. <https://doi.org/10.5281/zenodo.4774216>, doi: 10.5281/zenodo.4774216.

- 595 **Doi S**, Onoda Y, Kumagai S. Parameter estimation of various Hodgkin–Huxley-type neuronal models using a
596 gradient-descent learning method. In: *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, vol. 3 IEEE;
597 2002. p. 1685–1688. doi: [10.1109/SICE.2002.1196569](https://doi.org/10.1109/SICE.2002.1196569).
- 598 **Doruk RO**, Abosharb L. Estimating the Parameters of FitzHugh–Nagumo Neurons from Neural Spiking Data.
599 *Brain Sciences*. 2019; 9(12):364. doi: [10.3390/brainsci9120364](https://doi.org/10.3390/brainsci9120364), publisher: Multidisciplinary Digital Publishing
600 Institute.
- 601 **Fitzsimonds RM**, Song Hj, Poo Mm. Propagation of activity-dependent synaptic depression in simple neural
602 networks. *Nature*. 1997 Jul; 388(6641):439–448. <https://doi.org/10.1038/41267>, doi: [10.1038/41267](https://doi.org/10.1038/41267).
- 603 **Geyer CJ**, Thompson EA. Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. *Journal*
604 *of the American Statistical Association*. 1995 Sep; 90(431):909–920. [http://www.tandfonline.com/doi/abs/10.](http://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476590)
605 [1080/01621459.1995.10476590](http://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476590), doi: [10.1080/01621459.1995.10476590](https://doi.org/10.1080/01621459.1995.10476590).
- 606 **Givens CR**, Shortt RM. A class of Wasserstein metrics for probability distributions. *Michigan Mathematical*
607 *Journal*. 1984 Jan; 31(2). [https://projecteuclid.org/journals/michigan-mathematical-journal/volume-31/](https://projecteuclid.org/journals/michigan-mathematical-journal/volume-31/issue-2/A-class-of-Wasserstein-metrics-for-probability-distributions/10.1307/mmj/1029003026.full)
608 [issue-2/A-class-of-Wasserstein-metrics-for-probability-distributions/10.1307/mmj/1029003026.full](https://projecteuclid.org/journals/michigan-mathematical-journal/volume-31/issue-2/A-class-of-Wasserstein-metrics-for-probability-distributions/10.1307/mmj/1029003026.full), doi:
609 [10.1307/mmj/1029003026](https://doi.org/10.1307/mmj/1029003026).
- 610 **Golowasch J**, Goldman MS, Abbott LF, Marder E. Failure of Averaging in the Construction of a Conductance-
611 Based Neuron Model. *Journal of Neurophysiology*. 2002 Feb; 87(2):1129–1131. [https://doi.org/10.1152/jn.](https://doi.org/10.1152/jn.00412.2001)
612 [00412.2001](https://doi.org/10.1152/jn.00412.2001), doi: [10.1152/jn.00412.2001](https://doi.org/10.1152/jn.00412.2001), publisher: American Physiological Society.
- 613 **Gonçalves PJ**, Lueckmann JM, Deistler M, Nonnenmacher M, Öcal K, Bassetto G, Chintaluri C, Podlaski WF, Had-
614 dad SA, Vogels TP, Greenberg DS, Macke JH. Training deep neural density estimators to identify mechanistic
615 models of neural dynamics. *eLife*. 2020; 9:e56261. doi: [10.7554/eLife.56261](https://doi.org/10.7554/eLife.56261), publisher: eLife Sciences Publi-
616 cations, Ltd.
- 617 **Gupta S**, Lee REC, Faeder JR. Parallel Tempering with Lasso for model reduction in systems biology. *PLOS*
618 *Computational Biology*. 2020 Mar; 16(3):e1007669. <https://dx.plos.org/10.1371/journal.pcbi.1007669>, doi:
619 [10.1371/journal.pcbi.1007669](https://doi.org/10.1371/journal.pcbi.1007669).
- 620 **Hartoyo A**, Cadusch PJ, Liley DTJ, Hicks DG. Parameter estimation and identifiability in a neural population
621 model for electro-cortical activity. *PLOS Computational Biology*. 2019 May; 15(5):e1006694. [https://dx.plos.](https://dx.plos.org/10.1371/journal.pcbi.1006694)
622 [org/10.1371/journal.pcbi.1006694](https://dx.plos.org/10.1371/journal.pcbi.1006694), doi: [10.1371/journal.pcbi.1006694](https://doi.org/10.1371/journal.pcbi.1006694).
- 623 **Hastings WK**. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. 1970
624 Apr; 57(1):97–109. <https://academic.oup.com/biomet/article/57/1/97/284580>, doi: [10.1093/biomet/57.1.97](https://doi.org/10.1093/biomet/57.1.97).
- 625 **Heckman CJ**, Mottram C, Quinlan K, Theiss R, Schuster J. Motoneuron excitability: the importance of
626 neuromodulatory inputs. *Clinical neurophysiology : official journal of the International Federation of*
627 *Clinical Neurophysiology*. 2009 Dec; 120(12):2040–2054. <https://pubmed.ncbi.nlm.nih.gov/19783207>, doi:
628 [10.1016/j.clinph.2009.08.009](https://doi.org/10.1016/j.clinph.2009.08.009), edition: 2009/09/27.
- 629 **Hodgkin AL**, Huxley AF. A quantitative description of membrane current and its application to conduction and
630 excitation in nerve. *The Journal of physiology*. 1952 Aug; 117(4):500–544. [https://pubmed.ncbi.nlm.nih.gov/](https://pubmed.ncbi.nlm.nih.gov/12991237)
631 [12991237](https://pubmed.ncbi.nlm.nih.gov/12991237), doi: [10.1113/jphysiol.1952.sp004764](https://doi.org/10.1113/jphysiol.1952.sp004764).
- 632 **Hultborn H**, Pierrot-Deseilligny E. Input-output relations in the pathway of recurrent inhibition to motoneu-
633 rones in the cat. *The Journal of Physiology*. 1979 Dec; 297(1):267–287. [http://doi.wiley.com/10.1113/jphysiol.](http://doi.wiley.com/10.1113/jphysiol.1979.sp013039)
634 [1979.sp013039](http://doi.wiley.com/10.1113/jphysiol.1979.sp013039), doi: [10.1113/jphysiol.1979.sp013039](https://doi.org/10.1113/jphysiol.1979.sp013039).
- 635 **Mainen ZF**, Joerges J, Huguenard JR, Sejnowski TJ. A model of spike initiation in neocortical pyramidal neu-
636 rons. *Neuron*. 1995 Dec; 15(6):1427–1439. <https://linkinghub.elsevier.com/retrieve/pii/0896627395900209>, doi:
637 [10.1016/0896-6273\(95\)90020-9](https://doi.org/10.1016/0896-6273(95)90020-9).
- 638 **Marder E**, Goaillard JM. Variability, compensation and homeostasis in neuron and network function. *Nature*
639 *Reviews Neuroscience*. 2006; 7(7):563–574. Publisher: Nature Publishing Group.
- 640 **Marinari E**, Parisi G. Simulated Tempering: A New Monte Carlo Scheme. *Europhysics Letters (EPL)*.
641 1992 Jul; 19(6):451–458. <https://iopscience.iop.org/article/10.1209/0295-5075/19/6/002>, doi: [10.1209/0295-](https://doi.org/10.1209/0295-5075/19/6/002)
642 [5075/19/6/002](https://doi.org/10.1209/0295-5075/19/6/002).
- 643 **Meliza CD**, Kostuk M, Huang H, Nogaret A, Margoliash D, Abarbanel HDI. Estimating parameters and predicting
644 membrane voltages with conductance-based neuron models. *Biological Cybernetics*. 2014 Aug; 108(4):495–
645 516. <http://link.springer.com/10.1007/s00422-014-0615-5>, doi: [10.1007/s00422-014-0615-5](https://doi.org/10.1007/s00422-014-0615-5).

- 646 **Metropolis N**, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of State Calculations by Fast
647 Computing Machines. The Journal of Chemical Physics. 1953 Jun; 21(6):1087–1092. [http://aip.scitation.org/](http://aip.scitation.org/doi/10.1063/1.1699114)
648 [doi/10.1063/1.1699114](http://aip.scitation.org/doi/10.1063/1.1699114), doi: 10.1063/1.1699114.
- 649 **Miasojedow B**, Moulines E, Vihola M. An Adaptive Parallel Tempering Algorithm. Journal of Computational and
650 Graphical Statistics. 2013 Jul; 22(3):649–664. [http://www.tandfonline.com/doi/abs/10.1080/10618600.2013.](http://www.tandfonline.com/doi/abs/10.1080/10618600.2013.778779)
651 [778779](http://www.tandfonline.com/doi/abs/10.1080/10618600.2013.778779), doi: 10.1080/10618600.2013.778779.
- 652 **Nadim F**, Olsen O, De Schutter E, Calabrese R. Modeling the leech heartbeat elemental oscillator I. Interac-
653 tions of intrinsic and synaptic currents. Journal of computational neuroscience. 1995 Oct; 2:215–35. doi:
654 10.1007/BF00961435.
- 655 **Nocedal J**, Wright SJ. Numerical optimization. 2nd ed ed. Springer series in operations research, New York:
656 Springer; 2006. OCLC: ocm68629100.
- 657 **Pape HC**, McCormick DA. Noradrenaline and serotonin selectively modulate thalamic burst firing by enhancing
658 a hyperpolarization-activated cation current. Nature. 1989 Aug; 340(6236):715–718. [https://doi.org/10.1038/](https://doi.org/10.1038/340715a0)
659 [340715a0](https://doi.org/10.1038/340715a0), doi: 10.1038/340715a0.
- 660 **Park KS**, Mohapatra DP, Misonou H, Trimmer JS. Graded regulation of the Kv2. 1 potassium channel by variable
661 phosphorylation. Science. 2006; 313(5789):976–979. Publisher: American Association for the Advancement
662 of Science.
- 663 **Petzold L**. Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equa-
664 tions. SIAM Journal on Scientific and Statistical Computing. 1983 Mar; 4(1):136–148. [http://epubs.siam.org/](http://epubs.siam.org/doi/10.1137/0904010)
665 [doi/10.1137/0904010](http://epubs.siam.org/doi/10.1137/0904010), doi: 10.1137/0904010.
- 666 **Prinz AA**, Billimoria CP, Marder E. Alternative to Hand-Tuning Conductance-Based Models: Construction and
667 Analysis of Databases of Model Neurons. Journal of Neurophysiology. 2003 Dec; 90(6):3998–4015. [https://](https://www.physiology.org/doi/10.1152/jn.00641.2003)
668 www.physiology.org/doi/10.1152/jn.00641.2003, doi: 10.1152/jn.00641.2003.
- 669 **Prinz AA**, Bucher D, Marder E. Similar network activity from disparate circuit parameters. Nature Neuroscience.
670 2004 Dec; 7(12):1345–1352. <http://www.nature.com/articles/nn1352>, doi: 10.1038/nn1352.
- 671 **René A**, Longtin A, Macke JH. Inference of a Mesoscopic Population Model from Population Spike Trains. Neural
672 Computation. 2020 Aug; 32(8):1448–1498. <https://direct.mit.edu/neco/article/32/8/1448-1498/95629>, doi:
673 10.1162/neco_a_01292.
- 674 **Roffman RC**, Norris BJ, Calabrese RL. Animal-to-animal variability of connection strength in the leech heartbeat
675 central pattern generator. Journal of neurophysiology. 2012 Mar; 107(6):1681–1693. [https://pubmed.ncbi.](https://pubmed.ncbi.nlm.nih.gov/22190622)
676 [nlm.nih.gov/22190622](https://pubmed.ncbi.nlm.nih.gov/22190622), doi: 10.1152/jn.00903.2011, edition: 2011/12/21 Publisher: American Physiological
677 Society.
- 678 **Rudi J**, Bessac J, Lenzi A. Parameter Estimation with Dense and Convolutional Neural Networks Applied to the
679 FitzHugh-Nagumo ODE. In: *Proceedings of Mathematical and Scientific Machine Learning (MSML21)*; 2021. p.
680 1–27.
- 681 **Schmutz V**, Gerstner W, Schwalger T. Mesoscopic population equations for spiking neural networks with
682 synaptic short-term plasticity. The Journal of Mathematical Neuroscience. 2020 Dec; 10(1):5. [https://](https://mathematical-neuroscience.springeropen.com/articles/10.1186/s13408-020-00082-z)
683 mathematical-neuroscience.springeropen.com/articles/10.1186/s13408-020-00082-z, doi: 10.1186/s13408-020-
684 00082-z.
- 685 **Schulz DJ**, Goaillard JM, Marder E. Variable channel expression in identified single and electrically coupled
686 neurons in different animals. Nature Neuroscience. 2006 Mar; 9(3):356–362. <https://doi.org/10.1038/nn1639>,
687 doi: 10.1038/nn1639.
- 688 **Schulz DJ**, Goaillard JM, Marder EE. Quantitative expression profiling of identified neurons reveals
689 cell-specific constraints on highly variable levels of gene expression. Proceedings of the National
690 Academy of Sciences. 2007 Aug; 104(32):13187. <http://www.pnas.org/content/104/32/13187.abstract>, doi:
691 10.1073/pnas.0705827104.
- 692 **SciPy 1.0 Contributors**, Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski
693 E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ,
694 Jones E, Kern R, Larson E, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature
695 Methods. 2020 Mar; 17(3):261–272. <http://www.nature.com/articles/s41592-019-0686-2>, doi: 10.1038/s41592-
696 019-0686-2.

697 **Smith RC**. Uncertainty quantification: theory, implementation, and applications. Computational science and
698 engineering series, Philadelphia: Society for Industrial and Applied Mathematics; 2013.

699 **Soto-Treviño C**, Rabbah P, Marder E, Nadim F. Computational model of electrically coupled, intrinsically distinct
700 pacemaker neurons. *Journal of neurophysiology*. 2005 Jul; 94(1):590–604. <https://pubmed.ncbi.nlm.nih.gov/15728775>, doi: 10.1152/jn.00013.2005, edition: 2005/02/23.

702 **Stapor P**, Weindl D, Ballnus B, Hug S, Loos C, Fiedler A, Krause S, Hroß S, Fröhlich F, Hasenauer J. PESTO: Param-
703 eter EStimation TOolbox. *Bioinformatics*. 2018 Feb; 34(4):705–707. [https://academic.oup.com/bioinformatics/](https://academic.oup.com/bioinformatics/article/34/4/705/4562504)
704 [article/34/4/705/4562504](https://academic.oup.com/bioinformatics/article/34/4/705/4562504), doi: 10.1093/bioinformatics/btx676.

705 **Städter P**, Schälte Y, Schmiester L, Hasenauer J, Stapor PL. Benchmarking of numerical integration methods for
706 ODE models of biological systems. *Scientific Reports*. 2021 Dec; 11(1):2696. [http://www.nature.com/articles/](http://www.nature.com/articles/s41598-021-82196-2)
707 [s41598-021-82196-2](http://www.nature.com/articles/s41598-021-82196-2), doi: 10.1038/s41598-021-82196-2.

708 **Swensen AM**, Bean BP. Robustness of Burst Firing in Dissociated Purkinje Neurons with Acute or Long-
709 Term Reductions in Sodium Conductance. *The Journal of Neuroscience*. 2005 Apr; 25(14):3509. [http://](http://www.jneurosci.org/content/25/14/3509.abstract)
710 www.jneurosci.org/content/25/14/3509.abstract, doi: 10.1523/JNEUROSCI.3929-04.2005.

711 **Toth BA**, Kostuk M, Meliza CD, Margoliash D, Abarbanel HDI. Dynamical estimation of neuron and network
712 properties I: variational methods. *Biological Cybernetics*. 2011 Oct; 105(3-4):217–237. [http://link.springer.](http://link.springer.com/10.1007/s00422-011-0459-1)
713 [com/10.1007/s00422-011-0459-1](http://link.springer.com/10.1007/s00422-011-0459-1), doi: 10.1007/s00422-011-0459-1.

714 **Valderrama-Bahamo GI**. MCMC Techniques for Parameter Estimation of ODE Based Models in Systems Biol-
715 ogy. *Frontiers in Applied Mathematics and Statistics*. 2019; 5:10.

716 **Van Geit W**, De Schutter E, Achard P. Automated neuron model optimization techniques: a review. *Bio-*
717 *logical Cybernetics*. 2008 Nov; 99(4-5):241–251. <http://link.springer.com/10.1007/s00422-008-0257-6>, doi:
718 10.1007/s00422-008-0257-6.

719 **Van Geit W**, Achard P, De Schutter E. Neurofitter: a parameter tuning package for a wide range of electro-
720 physiological neuron models. *Frontiers in Neuroinformatics*. 2007; 1:1. [https://www.frontiersin.org/article/](https://www.frontiersin.org/article/10.3389/neuro.11.001.2007)
721 [10.3389/neuro.11.001.2007](https://www.frontiersin.org/article/10.3389/neuro.11.001.2007), doi: 10.3389/neuro.11.001.2007.

722 **Vavoulis DV**, Straub VA, Aston JAD, Feng J. A Self-Organizing State-Space-Model Approach for Parameter
723 Estimation in Hodgkin-Huxley-Type Models of Single Neurons. *PLoS Computational Biology*. 2012 Mar;
724 8(3):e1002401. <https://dx.plos.org/10.1371/journal.pcbi.1002401>, doi: 10.1371/journal.pcbi.1002401.

725 **Vousden WD**, Farr WM, Mandel I. Dynamic temperature selection for parallel tempering in Markov chain Monte
726 Carlo simulations. *Monthly Notices of the Royal Astronomical Society*. 2016 Jan; 455(2):1919–1937. <https://academic.oup.com/mnras/article-lookup/doi/10.1093/mnras/stv2422>, doi: 10.1093/mnras/stv2422.

728 **Łacki MK**, Miasojedow B. State-dependent swap strategies and automatic reduction of number of tem-
729 peratures in adaptive parallel tempering algorithm. *Statistics and Computing*. 2016 Sep; 26(5):951–964.
730 <http://link.springer.com/10.1007/s11222-015-9579-0>, doi: 10.1007/s11222-015-9579-0.

Appendix 1

MCMC solution maps of posteriors for individual constraints

Appendix 1 Figure 1. 0.1 nA results: Diagonal shows the 1D marginals for each of the parameters. The remaining plots show the 2D marginal relating one parameter with an other. The color represents the loss value from blue to yellow (low to high). The density is represented by the opacity (i.e., light to dark shows low to high densities). Each point is set to a transparency level of $\alpha = 1e-3$ where $\alpha = 1$ is opaque. As such, darker regions have more dots and indicated higher probabilities for good fits between data and model outputs.

Appendix 1 Figure 2. 0.2 nA results: Diagonal shows the 1D marginals for each of the parameters. The remaining plots show the 2D marginal relating one parameter with an other. The color represents the loss value from blue to yellow (low to high). The density is represented by the opacity (i.e., light to dark shows low to high densities). Each point is set to a transparency level of $\alpha = 1e - 3$ where $\alpha = 1$ is opaque. As such, darker regions have more dots and indicated higher probabilities for good fits between data and model outputs.

Appendix 1 Figure 3. 0.3 nA results: Diagonal shows the 1D marginals for each of the parameters. The remaining plots show the 2D marginal relating one parameter with an other. The color represents the loss value from blue to yellow (low to high). The density is represented by the opacity (i.e., light to dark shows low to high densities). Each point is set to a transparency level of $\alpha = 1e - 3$ where $\alpha = 1$ is opaque. As such, darker regions have more dots and indicated higher probabilities for good fits between data and model outputs.

Appendix 1 Figure 4. 0.4 nA results: Diagonal shows the 1D marginals for each of the parameters. The remaining plots show the 2D marginal relating one parameter with an other. The color represents the loss value from blue to yellow (low to high). The density is represented by the opacity (i.e., light to dark shows low to high densities). Each point is set to a transparency level of $\alpha = 1e - 3$ where $\alpha = 1$ is opaque. As such, darker regions have more dots and indicated higher probabilities for good fits between data and model outputs.