MethodsX 3 (2016) 577–588

Contents lists available at ScienceDirect

MethodsX

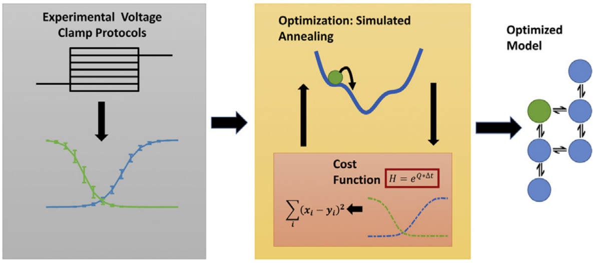
journal homepage: www.elsevier.com/locate/mex

A computationally efficient algorithm for fitting ion channel parameters

Zachary R. Teed, Jonathan R. Silva*

Department of Biomedical Engineering, Washington University, St Louis, MO, United States

GRAPHICAL ABSTRACT



ABSTRACT

Continuous time Markov models have been widely used to describe ion channel kinetics, providing explicit representation of channel states and transitions. Fitting models to experimental data remains a computationally demanding task largely due to the high cost of model evaluation. Here, we propose a method to efficiently optimize model parameters and structure. Voltage clamp channel protocols can be decomposed into a series of fixed steps of constant voltage resulting in a set of linear systems of differential equations. Given the linear systems, ODE integration can be swapped for the faster matrix exponential routine. With our parallelized implementation, optimized models are able to reproduce a wide range of experimentally collected data within one minute, a 50 times speedup over ODE integration.

- The cost of the objective function is reduced by employing the matrix exponential
- The likelihood of convergence is improved by applying synchronous start simulated annealing
- The approach was tested by optimizing parameters for a model of the cardiac voltage-gated Na^+ channel, $Na_V1.5$, and the KCNQ1 K^+ channel.

* Corresponding author at: Washington University, Campus Box 1097, 1 Brookings Dr., St. Louis, MO 63130, United States.
E-mail addresses: jonsilva@wustl.edu, jonsilva@gmail.com (J.R. Silva).

http://dx.doi.org/10.1016/j.mex.2016.11.001

ARTICLE INFO
Method name: The matrix exponential method for ion channel parameterization
Keywords: Cardiac action potential, Ion channels, Computer modeling
Article history: Available online 16 November 2016

Method

Following the seminal work of Hodgkin and Huxley in 1952 [1], mechanistic ordinary differential equation (ODE) models have been used to simulate dynamics of excitable systems including neurons, myocytes and pancreatic beta cells [2,3]. A persistent challenge in creating these models is the identification of optimal parameters. This difficulty arises from the multi-dimensionality of the search, the exponential form of the rate equations, and the existence of many local minima. Global search methods such as simulated annealing and genetic algorithms have been applied to identify optimal sets of parameters [4–6]. However, these approaches require significant computational resources.

The primary bottleneck for optimization is the numerical solution of ODEs needed to simulate experimentally observed dynamics. When evaluating a large set of models with global searches, this step becomes rate-limiting, and is particularly difficult when the equations become stiff, requiring very small time-steps or complicated implicit schemes [7].

We propose to overcome the ODE barrier by solving these differential equations with the matrix exponential, an approach that has been used to analyze channel kinetics and improve the efficiency of action potential simulations [8,9]. We tested this approach on experimental voltage pulse protocols, optimizing a two discrete-state Markov models of the cardiac Na^+ channel and the KCNQ1 K^+ channel. Optimization was performed using multiple chain simulated annealing [10]. Code has been made publically available at, where demos and https://github.com/silvalab/MMOptimizer instructions on model fitting can be found. Example model graphs and fits to Na^+ experimental data are shown in Fig. 1; K^+ fits are shown in Fig. 2.

We used the following voltage pulse protocols for Na^+ model fitting:

Voltage Dependent Activation (Fig. 1B)*:* A series of depolarizing pulses were applied from −120 mV to 20 mV in 10 mV increments from a resting potential of −100 mV. The voltage-dependence of channel conductance was found from this protocol by finding the peak current during the pulse and dividing by the driving force ($V_m - E_{Na}$), where V_m is the membrane potential and E_{Na} is the reversal potential for Na^+.

Steady State Inactivation (Fig. 1B)*:* Cells were held at −120 mV before being exposed to conditioning pulses ranging from −120 mV to 20 mV in 10 mV increments for 200 ms. Voltage was then stepped to −20 mV and peak current was recorded and normalized by dividing by the maximum current.

Recovery from Inactivation (Fig. 1C)*:* Cells were held at −120 mV before being exposed to a −20 mV depolarizing pulse for 200 ms. Cells were then allowed to recover at −120 mV for durations ranging from 1 ms to 1000 ms before a final depolarizing pulse at −20 mV where peak current was recorded and normalized across samples.

Rise/Fall Time (Fig. 1D)*:* This protocol replicated *Voltage Dependent Activation*. Rise (10%-90%) measured the amount of time it took for current to rise from 10% to 90% of the maximum recorded current. Fall (90%-20%) characterized fast inactivation kinetics by measuring the time it took for current to fall from 90% to 20% peak current.

Conductance Trace (Fig. 1E): In the final protocol we exposed the cell to a single depolarizing pulse and recorded normalized conductance over 5 ms.

Similarly, we used the following voltage pulse protocols for K^+ model fitting:

Voltage dependent activation (Fig. 2B): A series of depolarizing pulses were applied from −100 mV to 60 mV in 20 mV increments from a resting potential of −100 mV. The voltage-dependent channel

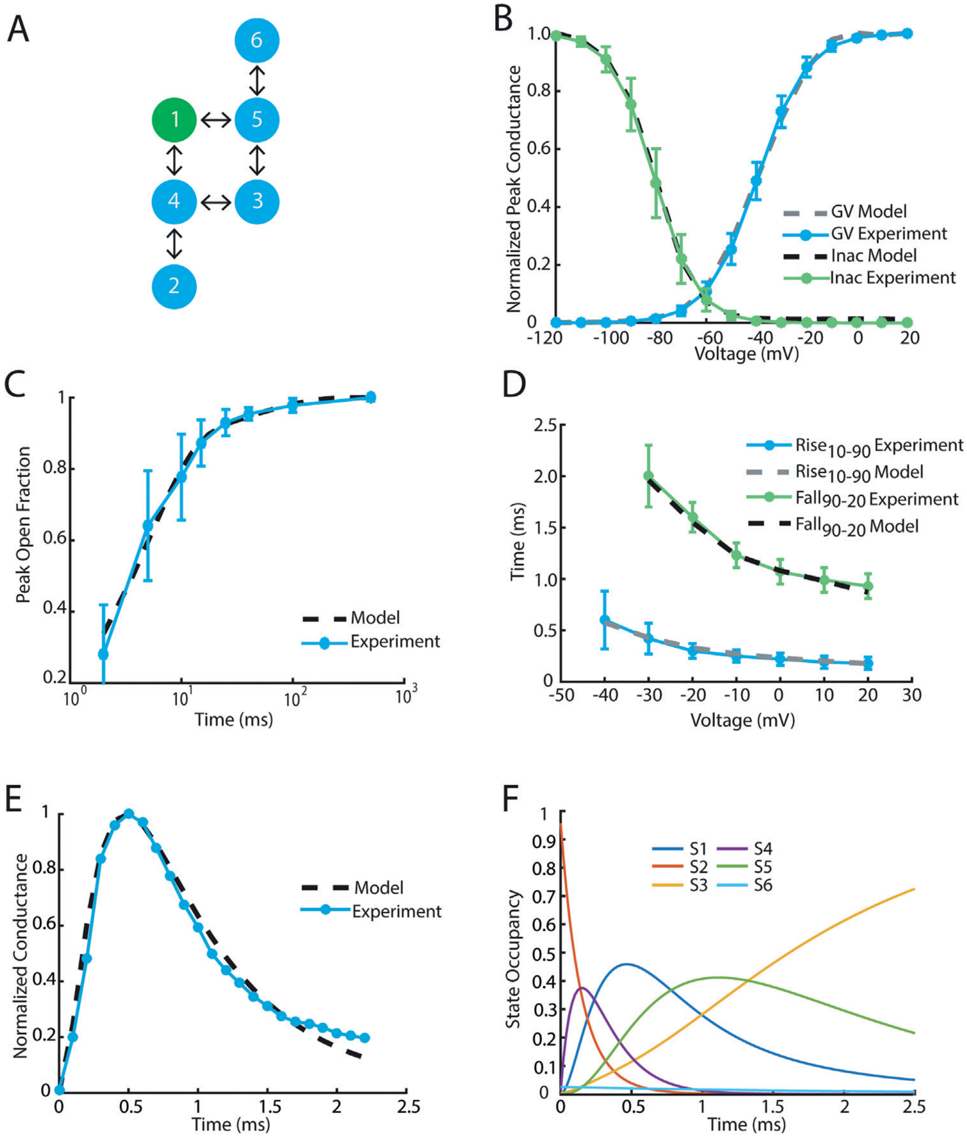


Fig. 1. Overview of Na^+ model fitting results. (A) Architecture of the fitted model, open state is shown in green. (B) Voltage dependent activation (G-V) and steady state inactivation (SSI) model simulations compared with experiment using protocols as described in reference [27]. (C) Recovery from single depolarizing pulse model simulation compared with experiment. (D) Rise time (10–90%) and fall time (90–20%) of normalized conductance from model and experiment. (E) Normalized conductance trace −10 mV depolarizing pulse. (F) Simulated state occupancy from a −10 mV pulse. According to the model, a depolarizing voltage pulse causes the channel to transition from the closed state (S2) through S4 to the conducting state (S1). The channel inactivates by transitioning from the conducting through S5 to the inactive state (S3).

conductance was calculated by dividing the peak current by the driving force ($V_m - E_K$), where V_m is the membrane potential and E_K is the reversal potential for K^+.

Deactivation (Fig. 2C): Cells were held at −80 mV before exposure to a depolarizing pulse at 60 mV for 2 ms. Voltage was then stepped to conditioning pulses ranging from −120 mV to 0 mV for 4 ms.

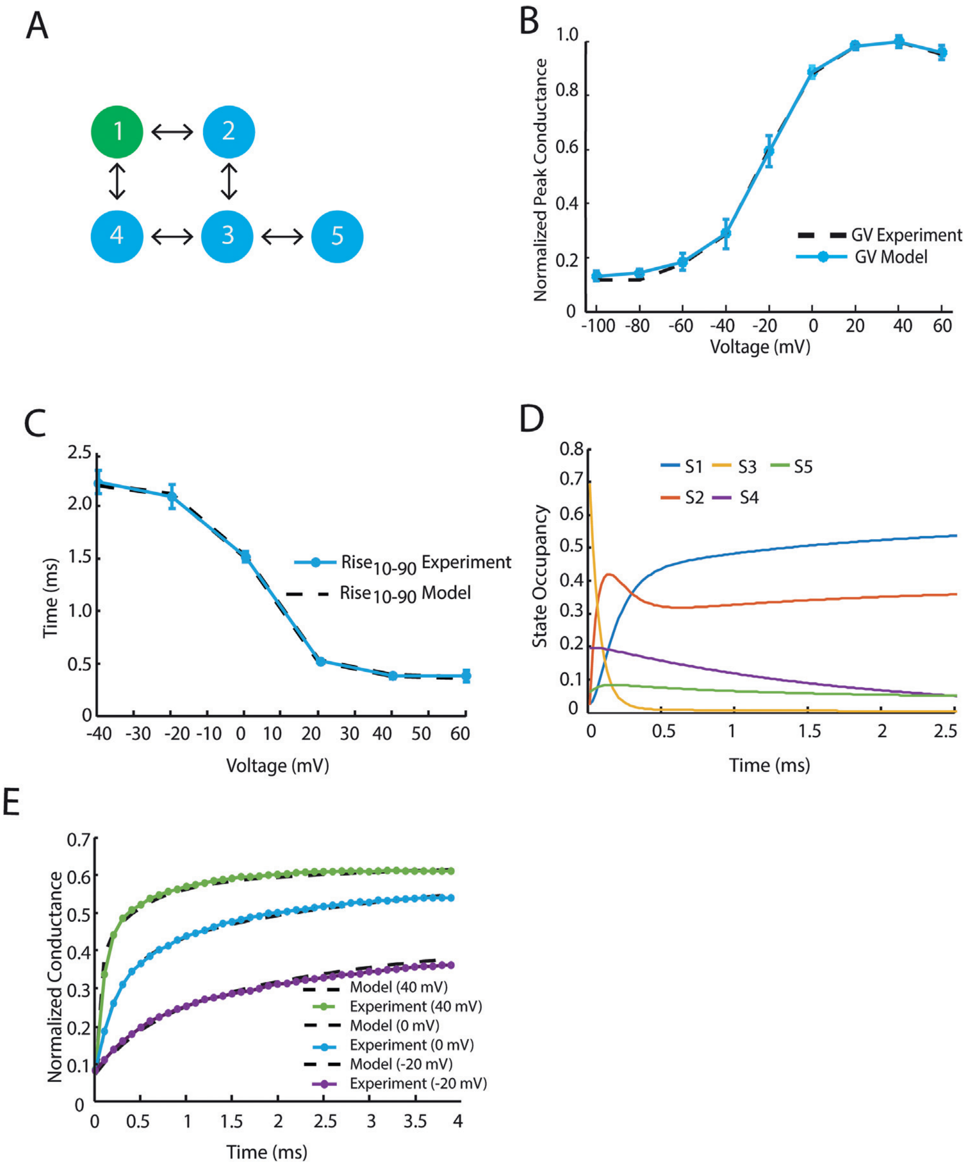


Fig. 2. Overview of K^+ model fitting results. (A) Architecture of the fitted model, open state is shown in green. (B) Voltage dependent activation model simulations compared with experiment. (C) Deactivation simulation compared to experiment (D) Rise time (10–90%) of conductance from model and experiment. (E) Normalized conductance trace of −20 mV, 0 mV, and 40 mV depolarizing pulses from a resting potential of −80 mV. (F) Simulated state occupancy from a −10 mV pulse. When exposed to a depolarizing pulse, the model transitions from the closed state (S3) through S4 to the open state (S1). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Voltage dependent-channel conductance was calculated by dividing the minimum absolute current by the driving force.

Rise Time (Fig. 2D)*:* This protocol replicated *Voltage Dependent Activation*. Rise (10%-90%) measured the amount of time it took for the current to rise from 10% to 90% of the peak.

Conductance Traces (Fig. 2E): Cells were exposed to −20 mV, 0 mV, and 40 mV depolarizing pulses and normalized conductance was recorded over a 4 ms interval.

Model structure

Models are implemented as a connected graph, a sequence of rate parameters, and a boolean conductivity vector. Each node in the graph represents a discrete model state, with transitions between model states denoted by graph edges. The conductivity vector gives the relative conductance for each state. Models are initialized from the space of random connected graphs using an approach described by Wilson et al. [11]. We begin by constructing a minimum spanning tree by performing a random walk over the graph until all nodes have been visited. Every time an unvisited node is encountered, the edge connecting this node to the previous node is added to the graph. Afterwards, new edges are added with constant probability. Initial rate parameters are drawn from independent normal distributions.

If we let $e \in E$ denote the set of edges and $r \in R$ the set of rates, the transition matrix Q can be generated as follows:

$$Q_{ij} = \begin{cases} r_{ij}, (i,j)\varepsilon E \\ 0, otherwise \end{cases} for i \neq j \tag{1}$$

$$Q_{ii} = -\sum_{(i,k)\varepsilon E} r_{ik} \tag{2}$$

Note that each edge is bidirectional, so $(i,j)\varepsilon E \Leftrightarrow (j,i)\varepsilon E$. With the transition matrix, the model kinetics can be defined by a single differential equation

$$\frac{d\boldsymbol{x}}{dt} = Qx \tag{3}$$

where $\vec{x}$ is the vector of state occupancies. The solution to this ODE can be approximated by using standard solvers such as Runge-Kutta methods [12].

If we limit ourselves to the unconstrained case, we can optimize over all $r \in R$ rate parameters. However, most ion channels are observed to obey microscopic reversibility, or the reversibility of flow in closed loops [13,14], so we instead limit the search space to the set of models that satisfy this principle. Rather than attempting to solve the constrained optimization problem, we follow the approach outlined by Menon et al. [4], where we optimize over a set of independent rate parameters which can be used to generate models that observe microscopic reversibility.

Microscopic reversibility requires that in equilibrium, the transition rates between any two connected states must balance each other [15]. Thus, if $(i,j) \in E$, the following must hold:

$$r_{ij}s_i = r_{ji}s_j \tag{4}$$

where s_i and s_j are the steady state occupancies for states i and j respectively. Taking the log of both sides, Eq. (4) can be written in matrix form

$$\ln(r_{ij}) - \ln(r_{ji}) = \ln(s_j) - \ln(s_i) \tag{5}$$

$$Dln(\boldsymbol{r}) = I_e^{\mathrm{T}}\ln(\boldsymbol{s}) \tag{6}$$

where I_e^T is the transpose of the even columns of the graph's incidence and D is a E × 2E difference matrix with

$$D_{ij} = \begin{cases} 1, j = 2i \\ -1, j = 2i + 1 \\ 0, otherwise \end{cases} \tag{7}$$

This forms a system of equations with 2E unknowns and E equations. To solve for *R*, we need an additional E independent equations. A simple choice is to define a new variable *k* to be the product of forward and backward rates

$$k_{ij} = r_{ij} r_{ji} \tag{8}$$

$$\ln(k_{ij}) = \ln(r_{ij}) + \ln(r_{ji}) \tag{9}$$

or again written in matrix form

$$abs(D)\ln(\boldsymbol{r}) = \ln(\boldsymbol{k}) \tag{10}$$

Combining Eq. (9) with Eq. (6) gives an independent linear systems of equations with 2E variables and 2E equations.

$$\begin{bmatrix} D \\ abs(D) \end{bmatrix} \ln(\boldsymbol{r}) = \begin{bmatrix} I_e^T \ln(\boldsymbol{s}) \\ \ln(\mathbf{k}) \end{bmatrix} \tag{11}$$

$$\ln(\boldsymbol{r}) = \begin{bmatrix} D \\ abs(D) \end{bmatrix}^{-1} \begin{bmatrix} I_e^T \ln(\boldsymbol{s}) \\ \ln(\mathbf{k}) \end{bmatrix} \tag{12}$$

Of course, we expect the elements of the rate vector to be voltage-dependent, but microscopic reversibility must also hold for all values of voltage. Eq. (12) shows that $\ln(\boldsymbol{r})$ is linear in terms of $\ln(\boldsymbol{s})$ and $\ln(\boldsymbol{k})$. Thus, we can define $\boldsymbol{s}(v)$ and $\boldsymbol{k}(v)$ to the exponent of a linear operation

$$s_i(v) = \exp(\boldsymbol{f}(v) \cdot \boldsymbol{\alpha_i}) \tag{13}$$

$$k_{ij}(v) = \exp\left(\boldsymbol{f}(v) \cdot \boldsymbol{\beta_{ij}}\right) \tag{14}$$

where $\boldsymbol{f}(v)$ is a vector valued function of voltage, and $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are vectors of rate parameters which are subject to optimization. Substituting Eqs. (13) and (14) into Eq. (12)

$$\ln(r) = \begin{bmatrix} D \\ abs(D) \end{bmatrix}^{-1} \begin{bmatrix} I_e^T \alpha f(v)^T \\ \beta f(v)^r \end{bmatrix} = \begin{bmatrix} D \\ abs(D) \end{bmatrix}^{-1} \begin{bmatrix} I_e^T \alpha \\ \beta \end{bmatrix}^{-1} f(v)^T \tag{15}$$

Substituting the voltage constant part of Eq. (15) for θ

$$\theta = \begin{bmatrix} D \\ abs(D) \end{bmatrix}^{-1} \begin{bmatrix} I_e^T \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \tag{16}$$

$$r(\boldsymbol{v}) = \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{v})^{\boldsymbol{T}}\right) \tag{17}$$

The results obtained in Eq. (17) can be used to generate the transition matrix as shown in Eq. (2). Because the rate vector is the result of linear operations on $\boldsymbol{s}$ and $\boldsymbol{k}$ it is of the same form as $\boldsymbol{s}$ and $\boldsymbol{k}$.

Voltage dependence

In the model, transition rates are given as the exponent of the inner product of a vector valued voltage function and the model parameters.

$$r(v) = e^{\boldsymbol{f}(v) \cdot \boldsymbol{\theta}} \tag{18}$$

Any rate function that can be written in this form can be made to satisfy microscopic reversibility by the method previously described. The simplest voltage function is a linear model motivated by the constant temperature Eyring rate theory [16], with $\boldsymbol{f}(v) = [1 \quad v]$. The transition rate then becomes

$$r(v) = e^{\theta_1 + v\theta_2} \tag{19}$$

In practice, the linear model worked well. We were able to obtain good fits for all provided voltage clamp protocols. Nonetheless, the linear model is prone to stiff differential equations. As membrane voltage becomes increasingly large (for $\theta_2 > 0$) or increasingly small (for $\theta_1 < 0$), the $r(v)$ becomes exponentially larger, often resulting in physically implausible transition rate. A simple solution would be to bound $r(v)$; however, in models where the rates are not linearly independent microscopic reversibility could be violated.

Another proposed voltage function extends the linear model by adding higher order polynomial terms [17], with $f_i(v) = v^{i-1}$ and $r(v) = e^{\theta_1 + \theta_2 v + \ldots + \theta_n v^{n-1}}$. We test both second order expansions of this form. The higher order terms gave the model more expressive power and resulted in significantly reduced fitting error. Regardless, the polynomial model showed signs of severe overfitting. Even slight modifications of the protocol parameters resulted in very different behavior, making models ill-suited for larger scale simulations.

Finally, we attempted a third voltage dependence function in attempt to circumvent some of the limitations of the linear model by bounding transitions rates while not violating microscopic reversibility. The proposed function is in the form

$$f(v) = \left[1 \quad sig(\frac{v-a}{b}) \right] \tag{20}$$

where sig is the sigmoid function and a and b are fixed parameters shared across all rates. In regions where $v \approx a$, the sigmoid model closely approximates the linear model; however, at large or small voltages, the sigmoid function becomes bounded limiting the maximum rate and reducing ODE stiffness. Additionally, this function provided better fits than the linear model.

Cost function evaluation

Each of the fitted voltage clamp protocols consist of a sequence of discrete voltage steps. The cost function is computed by simulating each of the protocols, recording the results, and using squared error as a metric to evaluate model performance. In this paper, we look at two different ways to simulate the model: traditional ODE solvers and the matrix exponential. A flow chart illustrating the cost function evaluation for both approaches is show in Fig. 3.

As the execution time profiling results in Fig. 4A show, ODE integration is the clear bottleneck, accounting for as much as 99% of execution time. ODE solvers were designed to solve a broad class of problems

$$\dot{x} = f(x,t), x(0) = x_0 \tag{21}$$

Yet, voltage clamp protocols commonly consist of a series of discrete voltage steps, reducing the system from $\dot{x} = Q(v(t))x$ to$\dot{x} = Q(v)x$ where $Q(v)$ is constant over the time $t_0[k]$ to $t_1[k]$. We expect that standard ODE solvers are not able to take full advantage of the linear nature of the constant voltage steps. In contrast, using the matrix exponential, the solution is given as $x(t) = e^{tQ}x_0$ [23]. The matrix exponential can further be used to sample state occupancy over a series of fixed time steps

$$x(t + n\Delta t) = Q^n x(t) = Qx(t + (n-1)\Delta t) \tag{22}$$

showing that it can effectively replace ODE solvers in our cost function evaluation routine. In our implementation we used ExpoKit's Padé approximation routine [19,23]

The resulting speedup is substantial—using a single thread on an Intel Xeon E5-2630 processor, the matrix exponential (EXPM) implementation converges in significantly less time than ode integration (ODE) as shown in Fig. 4B. With a single thread, the EXPM implementation was able fit models in the order of five minutes compared to five hours, a speed-up of more than 50×. Synchronous chain simulated annealing can easily be modified to use more than a single thread. Using 12 threads, model fitting shown in Fig. 1 can be performed in under a minute.

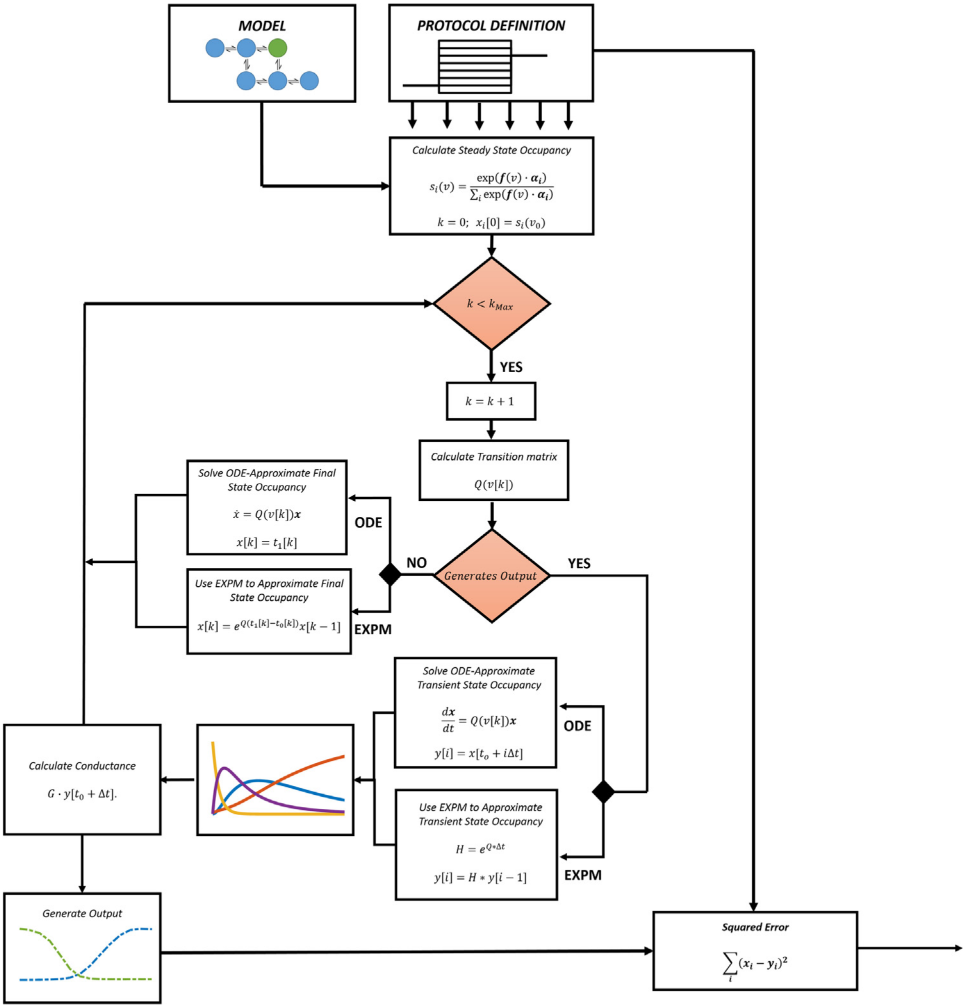

MODEL
PROTOCOL DEFINITION
Calculate Steady State Occupancy
$s_i(v) = \frac{\exp(\boldsymbol{f}(v) \cdot \boldsymbol{\alpha}_i)}{\sum_i \exp(\boldsymbol{f}(v) \cdot \boldsymbol{\alpha}_i)}$
$k = 0;\ x_i[0] = s_i(v_0)$
$k < k_{Max}$
YES
$k = k + 1$
Calculate Transition matrix
$Q(v[k])$
Solve ODE-Approximate Final State Occupancy
$\dot{x} = Q(v[k])\boldsymbol{x}$
$x[k] = t_1[k]$
ODE
NO
Generates Output
YES
Use EXPM to Approximate Final State Occupancy
$x[k] = e^{Q(t_1[k]-t_0[k])}x[k-1]$
EXPM
Solve ODE-Approximate Transient State Occupancy
$\frac{d\boldsymbol{x}}{dt} = Q(v[k])\boldsymbol{x}$
$y[i] = x[t_o + i\Delta t]$
ODE
Calculate Conductance
$G \cdot y[t_o + \Delta t]$.
Use EXPM to Approximate Transient State Occupancy
$H = e^{Q*\Delta t}$
$y[i] = H * y[i-1]$
EXPM
Generate Output
Squared Error
$\sum_i (x_i - y_i)^2$

Fig. 3. Cost function evaluation using matrix exponential/ODE solver. The cost function takes the model and protocol definition as input and outputs the squared error between the model and experimental outputs. Detailed information and instructions regarding protocol encoding can be found https://github.com/silvalab/MMOptimizer. The main procedure is repeated for each voltage trace in each protocol. The error for each trace is computed by comparing the simulation outputs to experimentally collected values; whether or not a step produces any output is encoded in the protocol design. First, the steady state occupancy is computed for the resting membrane potential. Next, we iterate through each step in the voltage clamp protocol, beginning at $t_0[k]$ and ending with $t_1[k]$ with step voltage denoted by $v[k]$. $Q(v[k])$ is calculated using Eq (2). If the current step does not produce output, we simply compute the state occupancy $x[k]$ at $t_1[k]$. Using ODE integrators we can approximate $x[k]$ by solving $\dot{x} = Q(v[k])\boldsymbol{x}$ with initial state $\boldsymbol{x}(0) = x[k-1]$. Using the matrix exponential we simply compute $x[k] = e^{Q(t_1[k]-t_0[k])}x[k-1]$. On the other hand, if the step does produce output, we are forced to use a small step size Δt to sample state occupancy over the range $t_0[k]$ to $t_1[k]$. Using the ODE solver, intermediate state occupancy is recorded for each time step for all $is.t.i\Delta t < t_1 - t_0$. When using the matrix exponential, H is precomputed as $H = e^{Q*\Delta t}$. Each step is then computed by 'powering up' the matrix exponential $y[i] = H \cdot y[i-1]$. Both methods then share the next step in which normalized conductivity is computed by performing the dot product of the state occupancy matrix y with the conductivity vector G. This giving the normalized channel conductivity over the timespan $t_0[k]$ to $t_1[k]$, providing the raw data necessary to output derived features such as peak conductance, rate constants, and normalized traces.

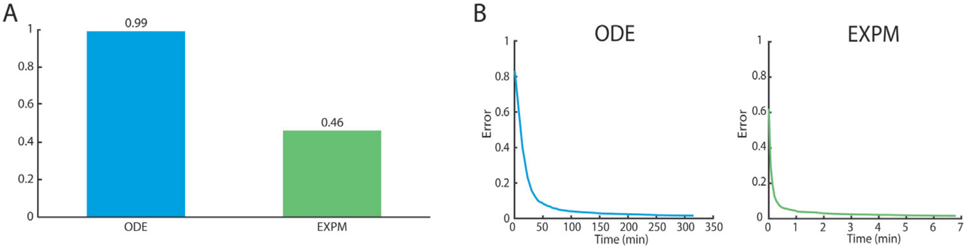


Fig. 4. (A) Execution time profile of model fitting. When using ode solvers, channel simulation accounts for 99% of execution time, compared with 46% in the EXPM implementation. (B) Average model error as a function of execution time for both EXPM and ODE implementations using a single process. Error converges in approximately 300 min when using ode integration; this time is reduced to five minutes with matrix exponentiation. These results agree with [23] where ODE methods were shown to take significantly longer and obtain lower overall accuracy.

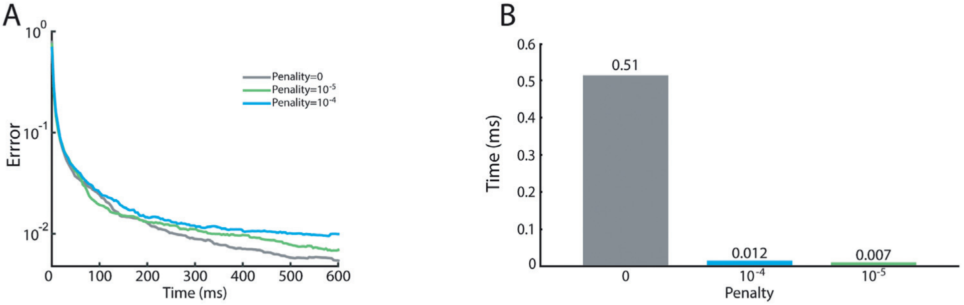


Fig. 5. (A) Impact of stiffness penalty on fitting error. The stiffness penalty imposes an additional cost on models proportional to the largest eigenvalue of the rate matrix for a range of voltages. (B) Impact of stiffness penalty on ODE integration time. Following the experiment shown in C, the fitted models were again run over the protocols except using an ODE solver. The values show the average per model execution time.

Finally, we evaluated a stiffness penalty on the models proportional to the model's largest eigenvalue over a range of voltages (−120 mV to +20 mV). This penalty constrains the search space and resulting models have slightly higher cost (Fig. 5A). Nonetheless, adding a small penalty proved to be important. Following optimization, we reran the fitted models over the protocols, using ODE solvers rather than the matrix exponential (Fig. 5B). Models with no added penalty took over 50 x longer due to their stiff differential equations. In whole cell simulations, membrane voltage is no longer a series of discrete steps, but rather a continuous variable requiring ODE integration. This penalty ensures that fitted models can be efficiently used in larger scale simulations, while constraining transition rates to physically plausible values.

Optimization

Similar to Menon et al., we optimize over model architecture as well rate parameters, maintaining microscopic reversibility by encoding rates as a set of independent parameters. Instead of a genetic algorithm, we use simulated annealing to explore the combinatorial search space. Genetic algorithms involve a recombination step, where individuals are selected to form offspring for the next generation [22]. With like structured models, new model rates can simply be drawn from a random distribution parametrized by the parents' rates. When models have different structures (i.e. different numbers of nodes and connectivity), recombination is less obvious. One option is to only perform recombination over models with the same structure. However, we observed this leads to a collapsed structural search space, and leads to unsatisfactory local optima. We obtained better convergence results in much less time using simulated annealing.

Specifically, we used synchronous chain simulated annealing for model optimization, considering the tradeoff between execution time and model error. Simulated annealing is an iterative stochastic search algorithm which combines local Monte Carlo search with an annealing schedule [10]. It probabilistically guarantees a globally optimal solution provided an infinitely long cooling schedule [20]; however, in practice, we wish to fit good suboptimal models quickly.

At each iteration we generate a new model proposal by perturbing the current model $m' \sim m$**.**

Perturbations are reversible and can alter both model structure and rate parameters taking one or more of the following forms:

1. State Addition: A new state is added to the model. A random edge is formed connecting the new state to a random state drawn from the existing model. The new rate parameters governing the edge and state are drawn from a normal distribution.
2. State Removal: A state is randomly chosen from the existing model and removed; all edges connecting this state are also removed. If removal results in a disconnected graph, the graph is reconnected by forming a random minimum spanning tree over the disconnected components.
3. Edge Addition: Two model states are randomly selected, if these states are not already connected a new edge is formed. Rate parameters are drawn from a normal distribution.
4. Edge Removal: A random model edge is selected and removed. If removal results in a disconnected graph, the graph is reconnected similar to (2).
5. Rate Perturbation: Each rate parameter has probability p of being updated. Updates are performed by adding a random variable drawn from a zero mean normal distribution to the current parameter. Thus, for rate parameter θ_i

$$\theta_i = \theta_i + x_i y_i \quad where \quad x_i \sim Ber(p), y_i \sim N(0, \sigma) \tag{23}$$

The locality of the perturbation can be modified by adjusting hyper-parameters p and σ. Every perturbation performs a rate update—other structural modifications are performed with a predefined probability.

Following perturbation, the cost function for the new model is computed. The new model is accepted with probability

$$P(e, e', T) = \max\left(\exp\left(-\frac{e' - e}{T}\right), 1\right) \tag{24}$$

where e is the previous model error, e' is the perturbed model error, and T the annealing temperature [21]. As the temperature decreases, $T \to 0$, the probability of making an uphill transition, $e' > e$, becomes increasingly small. In our implementation, we adopt the simple exponential annealing schedule [21] with $T(t) = T_0\gamma^t$ or equivalently $T(t) = T(t-1)\gamma$, where γ governs the rate of annealing and initial temperature T_0.

To avoid getting trapped in local optima and increase the probability of convergence, we run multiple synchronous chains [10]. Rather than a single model, we maintain a set of models, each with their own search path. Model perturbation and acceptance are both performed individually for each chain.

The optimization procedure attempts to fit all the voltage clamp protocols while minimizing model complexity. We found that in many cases, several models of similar complexity were able to achieve approximately the same fitting error. Hence, different runs often converged to different model structures. Since model structure is an optimization variable, we make no attempt to distinguish the types of channel states, rather such states naturally emerge as a product of the optimization. Open states can easily be identified by examine the conductivity vector. Similarly, closed and inactivated states can be identified by exposing models to a depolarizing test pulse (Fig. 1F), and looking at the transition of state occupancy.

Conclusions

Identification of continuous Markov model parameters is a challenging problem due to a large search space and a high cost of function evaluation, often requiring significant computational resources. We propose to overcome these limitations by efficiently encoding the cost function as a sequence of discrete voltage steps, solvable using the matrix exponential. Our implementation successfully captures channel kinetics without requiring excess resources or time.

Overall, we observed a 50× speedup when using the matrix exponential implementation of the cost function. One a single processor, optimization could be performed in only 10 min—extended to 12 cores, convergence time dropped to under a 2 min. The encoding of the cost function can be generalized to other voltage clamp protocols, allowing our implementation to be easily adapted to fit other channels types and data.

Modern biophysics methods are beginning to reveal molecular details of channel function [24–27], the nanoscale consequences of inherited mutations [27,28], and the precise mechanisms of small-molecule drug regulation [29]. To rigorously describe these molecular details, the complexity of models and the burden of optimization will increase substantially, necessitating advanced optimization methods and computer hardware. As such we expect that much future work will be needed to develop the methods required to create these models.

The matrix exponential method for ion channel parameterization

Acknowledgments

The authors are thankful to Taylor Voelker and Angela Schubert for help with figure composition, and the laboratory of Prof. Jianmin Cui for providing K$^+$ channel data. This work was supported by a James M. McKelvey Undergraduate Research Award in Engineering (Z.T.) and a Burroughs Wellcome Fund Career Award at the Scientific Interface 101029 and an American Heart Association grant 16GRNT31200025 (J.S.).

MethodsX thanks the reviewers of this article for taking the time to provide valuable feedback.

References

[1] Alan L. Hodgkin, Andrew F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. 4 (1952) 500.

[2] Yoram Rudy, Jonathan R. Silva, Computational biology in the study of cardiac ion channels and cell electrophysiology, Q. Rev. Biophys. 39 (01) (2006) 57–116.

[3] Jonathan R. Silva, Paige Cooper, Colin G. Nichols, Modeling K, ATP-dependent excitability in pancreatic islets, Biophys. J. 107 (9) (2014) 2016–2026.

[4] Vilas Menon, Nelson Spruston, William L. Kath, A state-mutating genetic algorithm to design ion-channel models, Proc. Natl. Acad. Sci. U. S. A. 106 (39) (2009) 16829–16834.

[5] Meron Gurkiewicz, Alon Korngreen, A numerical approach to ion channel modelling using whole-cell voltage-clamp recordings and a genetic algorithm, PLoS Comput. Biol. 3 (8) (2007) e169.

[6] Lorin S. Milescu, Gustav Akk, Frederick Sachs, Maximum likelihood estimation of ion channel kinetics from macroscopic currents, Biophys. J. 88 (4) (2005) 2494–2515.

[7] David Garfinkel, Carl B. Marbach, Norman Zalmon Shapiro, Stiff differential equations, Annu. Rev. Biophys. Bioeng. 6 (1) (1977) 525–542.

[8] David Colquhoun, Alan G. Hawkes, A Q-matrix cookbook, Single-Channel Recording, Springer, US, 1995, pp. 589–633.

[9] Tomáš Starý, Vadim N. Biktashev, Exponential integrators for a Markov chain model of the fast sodium channel of cardiomyocytes, IEEE Trans. Biomed. Eng. 62 (4) (2015) 1070–1076.

[10] Soo-Young Lee, Kyung Geun Lee, Synchronous and asynchronous parallel simulated annealing with multiple Markov chains, IEEE Trans. Parallel Distrib. Syst. 7 (10) (1996) 993–1008.

[11] David Bruce Wilson, Generating random spanning trees more quickly than the cover time, Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, ACM, 1996.

[12] Uri M. Ascher, Linda R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-algebraic Equations, vol. 61, Siam, 1998.

[13] David Colquhoun, et al., How to impose microscopic reversibility in complex reaction mechanisms, Biophys. J. 86 (6) (2004) 3510–3518.

[14] Brad S. Rothberg, Karl L. Magleby, Testing for detailed balance (microscopic reversibility in ion channel gating, Biophys. J. 80 (6) (2001) 3025.

[15] Frederick P. Boynton, Detailed balance and microscopic reversibility, J. Chem. Phys. 40 (10) (1964) 3124.

[16] Peter Hänggi, Peter Talkner, Michal Borkovec, Reaction-rate theory: fifty years after Kramers, Rev. Mod. Phys. 62 (2) (1990) 251.

[17] Alain Destexhe, John R. Huguenard, Nonlinear thermodynamic models of voltage-dependent currents, J. Comput. Neurosci. 9 (3) (2000) 259–270.

[19] Roger Sidje, Expokit: a software package for computing matrix exponentials, ACM Trans. Math. Softw. (TOMS) 24 (1) (1998) 130–156.

[20] Lester Ingber, Simulated annealing: practice versus theory, Math. Comput. Model. 18 (11) (1993) 29–57.

[21] Scott Kirkpatrick, C. Daniel Gelatt, Mario P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[22] Kalyanmoy Deb, et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[23] Cleve Moler, Charles Van Loan, Nineteen dubious ways to compute the exponential of a matrix, SIAM Rev. 20 (4) (1978) 801–836.

[24] Wandi Zhu, Zoltan Varga, Jonathan R. Silva, Molecular motions that shape the cardiac action potential: insights from voltage clamp fluorometry, Prog. Biophys. Mol. Biol. 120 (1) (2016) 3–17.

[25] Joshua B. Brettmann, et al., Role of protein dynamics in ion selectivity and allosteric coupling in the NaK channel, Proc. Natl. Acad. Sci. U. S. A. 112 (50) (2015) 15366–15371.

[26] Michael W. Rudokas, et al., The Xenopus oocyte cut-open vaseline gap voltage-clamp technique with fluorometry, JoVE 85 (2013).

[27] Zoltan Varga, et al., Direct measurement of cardiac Na+ channel conformations reveals molecular pathologies of inherited mutations, Circ. Arrhythmia Electrophysiol. 8 (5) (2015) 1228–1239.

[28] Jonathan R. Silva, Steve A.N. Goldstein, Voltage-sensor movements describe slow inactivation of voltage-gated sodium channels II: a periodic paralysis mutation in NaV1. 4 (L689I), J. Gen. Physiol. 141 (3) (2013) 323–334.

[29] Yukiko Muroi, Baron Chanda, Local anesthetics disrupt energetic coupling between the voltage-sensing segments of a sodium channel, J. Gen. Physiol. 133 (1) (2009) 1–15.