

Generalized Leaky Integrate-And-Fire Models Classify Multiple Neuron Types

Corinne Teeter, Ramakrishnan Iyer, Vilas Menon, Nathan Gouwens, David Feng, Jim Berg, Nicholas Cain, Christof Koch, Stefan Mihalas

January 31, 2017

Abstract

In the mammalian neocortex, there is a high diversity of neuronal types. To facilitate construction of system models with multiple cell types, we generate a database of point models associated with the Allen Cell Types Database. We construct a series of generalized integrate-and-fire (GLIF) models of increasing complexity aiming to reproduce the spiking behaviors of the recorded neurons. We test the performance of these GLIF models on data from 771 neurons from 14 transgenic lines, with increasing model performance for more complex models. To answer how complex a model needs to be to reproduce the number of electrophysiological cell types, we perform unsupervised clustering on the parameters extracted from these models. The number of clusters is smaller for individual model types, but when combining all GLIF parameters 18 clusters are obtained, while 11 clusters are obtained using 16 electrophysiological features. Therefore, these low dimensional models have the capacity to characterize the diversity of cell types without the need for a priori defined features.

1 Introduction

The problem of understanding the complexity of the brain has been central to neuroscience. Cell types are a conceptual simplification intended to reduce this complexity. A large scale effort at the Allen Institute for Brain Science has focused on characterizing the diversity of cell types in primary visual cortex of the adult mouse using electrophysiology, morphological reconstructions, connectivity, and modeling in one standardized effort. This has resulted in the Allen Cell Type Database that is publicly and freely available at <http://celltypes.brain-map.org>. It includes both morphological and electrophysiological data of genetically identified neurons mapped to the common coordinate framework. Morphologically and biophysically detailed as well as simple generalized linear integrate-and-fire point neuron models have been generated (Allen Institute for Brain Science, 2016a) to reproduce the highly standardized data.

In this study, we address the problem of cell types using point models to provide a dimensionality reduction to the most relevant aspects of neuronal input/output transformations. We show the ability of simple generalized leaky integrate and fire point models to both reproduce the spike times of biological data and effectively reduce the biological space to a set of useful parameters. Beyond the benefits of clarifying mechanisms for single neuron behavior, single neuron models are used in larger network models that attempt to explain network computation. Associated with this study we provide a large scale database of point neuron models.

Many neuron models have been developed to describe and recreate various aspects of neuronal behavior (Koch, 2004). At one end of the spectrum are the morphologically and biophysically realistic Hodgkin-Huxley like models (Hodgkin and Huxley, 1952; Hay et al., 2011). Their strength relies in their capacity to map between multiple observables: morphology, electrophysiology, intra-cellular calcium concentration and levels of expression and patterns of distributions of ionic currents. At the other end are highly simplified models, in particular the integrate-and-fire (IF) family of models (Lapicque, 1907). Although adding complexity to a model may increase the ability of that model to recreate certain behavior, finding the right parameters for complex models becomes a challenge (Prinz et al., 2003), and the computational power needed to simulate sophisticated neural models can be quite large (Markram et al., 2015). Therefore, ideally one would use a computationally minimal model adequate to recreate and to understand the desired behavior (Gollisch and Meister, 2010). In order to both facilitate network modeling efforts and understand the complexity of cell types, models many levels of sophistication have been generated (Allen Institute for Brain Science, 2016a) to recreate biological data. However, the level of complexity needed to meaningfully describe biological data is not understood.

One simplification which significantly reduces model complexity is to represent the entire dendritic tree, soma and axonal hillock by a single compartment, while maintaining the dynamics of the individual conductances (Koch, 2004). This approximation is especially warranted when characterizing neurons via somatic current injection and voltage recording.

Several models can reproduce a wide variety of spiking behavior, while reducing the number of dimensions to two (FitzHugh, 1961; Morris and Lecar, 1981). In 2003, Izhikevich also created a simple model that recreated several stereotypical realistic firing patterns by reducing the dimensionality of the Hodgkin-Huxley model (Izhikevich et al., 2003; Izhikevich, 2007). Although low-dimensional (Izhikevich, 2004), this model is non-linear which creates difficulties optimizing the model parameters to achieve biological spiking behavior. Similar firing patterns can be obtained using the adaptive exponential (AdEx) neuronal model (Brette and Gerstner, 2005).

Threshold-based models, which represent the spike initiation as a threshold rather than a dynamical system, are based on the simple leaky integrate and fire (LIF) (Lapicque, 1907) model. Their parameters can more easily be fit to biological data (Paninski et al., 2004; Rossant et al., 2010; Mihalas and Niebur, 2009; Dong et al., 2011; Mensi et al., 2012; Pozzorini et al., 2015). For a more in-depth characterization of the diversity of neuron models, see the review (Herz et al., 2006), and for their capacity to reproduce spike times see (Gerstner and Naud, 2009).

Here, we use the leaky integrate-and-fire (LIF) framework, due to the versatility of adding multiple mechanisms and the capacity to rapidly find unique parameters for them. A standard LIF model was the starting point, progressing to more generalized leaky integrate-and-fire (GLIF) models schematized in Figure

2a. In these models, the mechanisms are separated by time scale: none of the fast processes associated with the action potential itself are included directly in the dynamics. Some of these are, however, accounted for in the reset rules which map the state before the spike to the state after (i.e. voltage and threshold). Unlike the simple LIF which assumes all spikes are stereotyped and reset to the same state, the generalized reset rule (R) assumes that the spikes are sufficiently similar such that a mapping between the state before and after the spike can be found. In this study, we search only for linear mappings. The rapid changes of the action potentials are followed by slower dynamics which affect a neuron's state: e.g. slow inactivation of voltage-dependent currents which activate during the spike, or Ca^{2+} -dependent potassium currents. We assume that they have a stereotyped activation following a spike, and we bundle all these effects into a set of afterspike currents at different time scales (ASC), and into a change of the instantaneous threshold. Finally, slow depolarization can lead to partial inactivation of the voltage-dependent sodium current which generates a spike. We incorporate this mechanism into a change of the threshold which is dependent on the membrane potential (AT). The GLIF framework (Mihalas and Niebur, 2009) is sufficiently general such that multiple additional mechanisms can be added. How such a framework can be used to rapidly fit neuronal models is known (Pozzorini et al., 2015). The current work uses similar methods to (Pozzorini et al., 2015), with a set of differences: 1. More complex reset rules are used which map the state of the neuron before the spike to that after. These are essential for differentiating between neuron types. 2. We include a membrane potential dependent adaptation of threshold. 3. Exponential bases functions are used for the time dependencies. These allow a much sparser representation which is essential for clustering. 4. We use a direct measurement of noise in the membrane potential of a neuron rather than a probabilistic threshold. 5. The variance in the input current is smaller, closer to that observed in vivo.

When classifying neuron types based on a selection of features, one question which is difficult to address is the importance of these features to the function of the circuit. By doing the clustering on parameters which link somatic input of currents to spike train, we construct classes which are relevant for simulations where spike times are a desired observable. One of the advantages of GLIF neurons is their capacity to obtain unique parameters for diverse mechanisms (Mensi et al., 2012; Dong et al., 2013), which we show here, allow for classifications on model parameters.

Here we demonstrate the great power of computationally simple GLIF models to both reproduce the spike times of biological data and reduce the vast biological parameter space to a low dimensional space which can classify neurons with a specificity equal to that of common electrophysiological features. We characterize the parameters of the GLIF models from the different transgenic lines and illustrate the neuron that best recreates the biological spike train for each of the transgenic lines. In addition, we characterize common electrophysiological features associated with the different GLIF clusters.

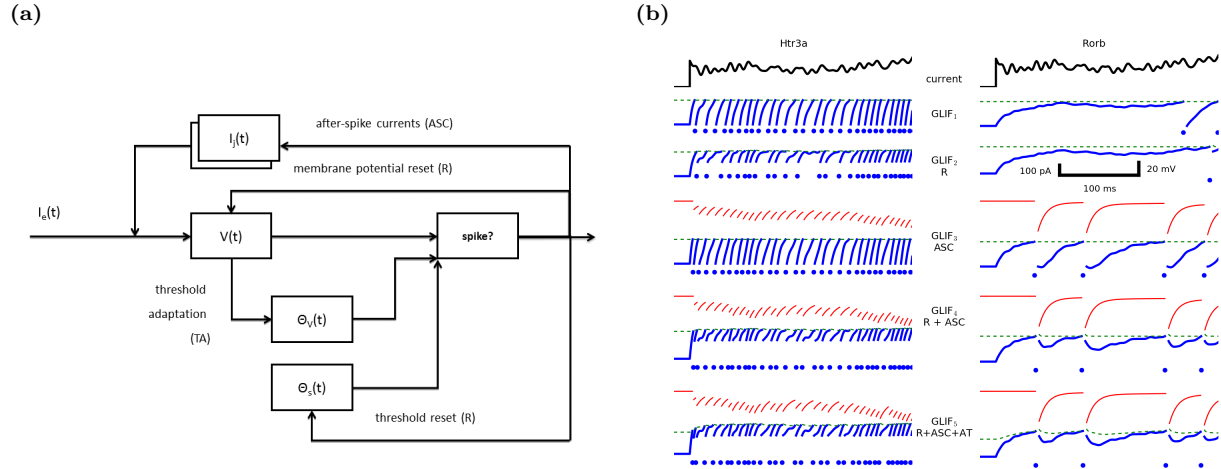


Figure 1: (a) Schematic describing the mechanisms included in our family of generalized leaky integrate-and-fire (GLIF) models used to describe the evolution of the trans-membrane potential and spiking of close to a thousand neurons recorded under highly standardized conditions. The input is a current, $I_e(t)$, injected via the patch electrode. The output is the membrane potential, $V(t)$. 1D model: $GLIF_1$ model contains one variable, $V(t)$, which is reset when it reaches a fixed threshold. 2D model: $GLIF_2$ models include a second variable, spike induced threshold $\Theta_S(t)$, which dynamically defines the value of the threshold. Both $V(t)$ and $\Theta_S(t)$ after the spike linearly depend on their values before the spike; between spikes their evolution equations are independent. 3D model: $GLIF_3$ include $V(t)$ and two variables corresponding to two after spike currents with different time constants. 4D model: $GLIF_4$ combines the spike induced threshold changes and after spike currents. 5D model: $GLIF_5$ adds $\Theta_V(t)$ as a membrane potential dependent component of the threshold. (b) Illustration of the five distinct GLIF models using exemplary recorded neurons from two different transgenic lines (left and right column). The injected current, $I_e(t)$, is plotted at the top in black. In all panels, the model trans-membrane potential, $V(t)$, is pictured in blue (spike rasters are marked below), and the threshold, Θ , in dashed green. For models that contain after-spike currents, the total current is in red. The scale bar applies to all panels except to the injected current. $GLIF_1$: Equivalent to the standard LIF model, when $V(t)$ reaches the constant threshold, a spike is produced. After a short refractory period, V is reset to a constant value. $GLIF_2$: The threshold Θ_s modestly increase with each spike, and decays to a constant. When $V(t)$ reaches Θ_s , the neuron spikes. After the refractory period, both V and Θ are reset to a value which is dependent on the state of the neuron before the spike. $GLIF_3$: Every action potential induces after-spike currents which decay back to zero. The level of induction depends on the state of the neuron before the spike. $GLIF_4$: Here, spike-induced changes in threshold and after-spike currents are combined and reset for all variables which are dependent on the state of the neuron before the spike. $GLIF_5$: The addition of a voltage-dependence to the threshold Θ .

2 Results

2.1 Model Definitions

We developed five different GLIF models, each of which successively incorporates the time evolution of one to five state variables. The set of five state variables $X = \{V(t), \Theta_s(t), I_{(j=1,2)}(t), \Theta_v(t)\}$ represent the neuronal membrane potential, the spike-dependent component of the threshold, two after-spike currents and the membrane potential-dependent component of the threshold respectively. Figure 1 illustrates the iteration between the state variables in the model equations. It is assumed that these state variables evolve in a linear manner between spikes:

$$V'(t) = \frac{1}{C} \left(I_e(t) + \sum_j I_j(t) - \frac{1}{R} (V(t) - E_L) \right) \quad (1)$$

$$\Theta'_s(t) = -b_s \Theta_s(t) \quad (2)$$

$$I'_j(t) = -k_j I_j(t); j = 1, 2 \quad (3)$$

$$\Theta'_v(t) = a_v (V(t) - E_L) - b_v \Theta_v(t) \quad (4)$$

where C represents the neurons's capacitance, R , the membrane resistance, E_L , the resting membrane potential, I_e , the external current, $1/k_j$, $1/b_s$, $1/b_v$ are the time constant of the afterspike currents, spike and voltage dependence of the threshold and a_v couples the membrane potential to the threshold.

If $V(t) > \Theta_v(t) + \Theta_s(t) + \Theta_\infty$, a spike is generated and the state variables are updated with a linear dependence on the state before the spike:

$$V(t_+) \leftarrow E_L + f_v \times (V(t_-) - E_L) - \delta V \quad (5)$$

$$\Theta_s(t_+) \leftarrow \Theta_s(t_-) + \delta \Theta_s \quad (6)$$

$$I_j(t_+) \leftarrow f_j \times I_j(t_-) + \delta I_j \quad (7)$$

$$\Theta_v(t_+) \leftarrow \Theta_v(t_-) \quad (8)$$

The 1D model, $GLIF_1$ is represented by equation 1, with a simple reset. $GLIF_2$ adds Θ_s and the more complex reset rules (equations 5 and 6). $GLIF_3$ uses membrane potential and two spike induced currents, $GLIF_4$ uses membrane potential, spike induced threshold dependence and afterspike currents, while $GLIF_5$ uses all the variables described. A detailed description of the models along with their mathematical equations are available in the "4.1 Model Definitions" section of the Supplementary Material.

2.2 Data

Intracellular electrophysiological recordings were carried out via a highly standardized process (Allen Institute for Brain Science, 2016b). The majority of the data (approx. 600 cells) can be accessed on the Allen Cell Types Database. Data that pass standardized electrophysiology quality control criteria but fail downstream quality control criteria such as imaging or reconstruction are included in this manuscript but are not released in the Allen Cell Types Database. The available data consists of *in vitro* electrophysiology data collected from 19 different transgenic lines, recording from both labeled (Cre positive) and non labeled (Cre negative) neurons (Figure 2).

Transgenic lines selective for different types of neurons were used for targeting. Excitatory neurons from each layer are identified using layer-selective lines (Figure 2). For inhibitory neurons, cells are targeted across lamina using lines selective for known neuronal subtypes Tasic et al. (2016). Cells from each of the three major inhibitory subtypes (Pvalb (primarily basket cells), Sst (primarily Martinotti cells), and Htr3a (diverse morphologies)) are represented. The Vip, Ndnf, and Chat Cre lines label a subtype of Htr3a-positive interneurons.

Only transgenic lines which contained more than 7 neurons that had the necessary data to create a basic $GLIF_1$ model described in Figure 2 were included in this study. In addition, models which had biologically unrealistic parameter values are eliminated from the data set (Please see section "4.7 Exclusion Criteria" in the Supplementary Material for a detailed explanation of excluded data). Figure 2 summarizes the properties of the different transgenic lines considered here. Colors and short hand names represent all transgenic line data throughout this manuscript. In general, shades of red represent inhibitory transgenic lines and shades of blue represent excitatory lines. Overall a total of 771 neurons from 14 transgenic lines meet the criteria for fitting the $GLIF_1$ and $GLIF_3$ models. Of these, 289 neurons meet the additional requirements for models with reset rules ($GLIF_2$ and $GLIF_4$) and 288 neurons meet all the criteria for a GIF_5 model.

Throughout the manuscript data and models from two exemplary canonical neurons are consistently shown as examples: an Htr3 inhibitory neuron (specimen ID 474637203) and a Rorb pyramidal neuron (specimen ID 479766169).

We designed a set of stimuli optimized for testing parameters of our GLIF models (Figure 2). The noise stimuli were picked to have similar statistics to *in vivo* patch clamp recordings. Using stimuli with highly varied structure is important, as it allows the model to explore the parameter space. Using stereotyped stimuli, e.g. square pulses, would present only a small subset of possible histories (both spiking and sub-threshold), and bias the dataset to a non-physiological regime. A short (1s) stimulus chunk is repeated three times to allow the analysis of short term adaptation. These three second bouts are repeated at different amplitudes, one sub-threshold, one peri-threshold and one supra-threshold to explore different parameter regimes. This stimulus sequence is repeated at least twice to check for reliability of the neurons under the same stimulus history. A second instantiation of a noise with the same characteristics is used to check the performance of the model under naturalistic stimulus conditions. A very short (3 ms) square stimulus of multiple amplitudes is used to compute the neuron's instantaneous threshold. A series of such short squares at different frequencies are used to characterize the spike-induced changes in the absence of additional membrane potential-induced changes. These are part of the electrophysiological stimulation protocol which is the subject of a different study.

2.3 Post-hoc optimization of instantaneous threshold, Θ_∞

GLIF model fitting is achieved in two steps. In the first step, parameters are fit directly from the electrophysiological data using linear methods described in detail in section "4.2 Parameter fitting and distributions" of the Supplementary Material: E_L from the mean membrane potential at rest, R , and C from fitting the membrane potential during a subthreshold noise stimulus, $\delta I_j, k_j$ from fitting the membrane potential between spikes during a suprathreshold stimulus, $f_v, \delta t, \delta V$ relating the potential before and after a spike during during the noise stimulus, $b_s, \delta \Theta_S$ from the potential of subsequent spikes during a series of brief square pulses, a_v and b_v from the potential of subsequent spikes during the suprathreshold noise stimulus. Slices from this parameter space are shown in Figure 3.

In the second step, the instantaneous threshold, Θ_∞ , was optimized to fit the probability of a neuron model reproducing observed spike times. This is realized using a non-linear Nelder Mead optimization strategy as described in detail in section "4.4 Post-Hoc Optimization" of the Supplementary Material. Θ_∞ was chosen for further optimization because it is an essential parameter in the overall excitability of neurons and is likely the best parameter to counteract any error in the fitting or the inherent error introduced by simplifying any complex system to a simple model. Although Θ_∞ did change between the fitting and post-hoc optimization step, on average the value remains consistent (See Figure 7 c and d in the Supplementary Material).

(Transgenic Lines)

name	n	E/I	L1	L2/3	L4	L5	L6
all neurons	771						
inhibitory	315	I	X	X	X	X	X
excitatory	456	E		X	X	X	X
Scnn1a-Tg2	34	E			X		
Scnn1a-Tg3	47	E			X	X	
Nr5a1	64	E			X		
Rorb	108	E			X	X	
Cux2	74	E		X	X		
Ntsr1	44	E					X
Ctgf	21	E					X
Rbp4	64	E				X	
Sst	92	I		X	X	X	X
Pvalb	82	I		X	X	X	X
Htr3a	96	I	X	X	X	X	
Ndnf	13	I	X				
Chat	7	I		X	X	X	X
Vip	25	I	X	X	X	X	

(Stimuli)

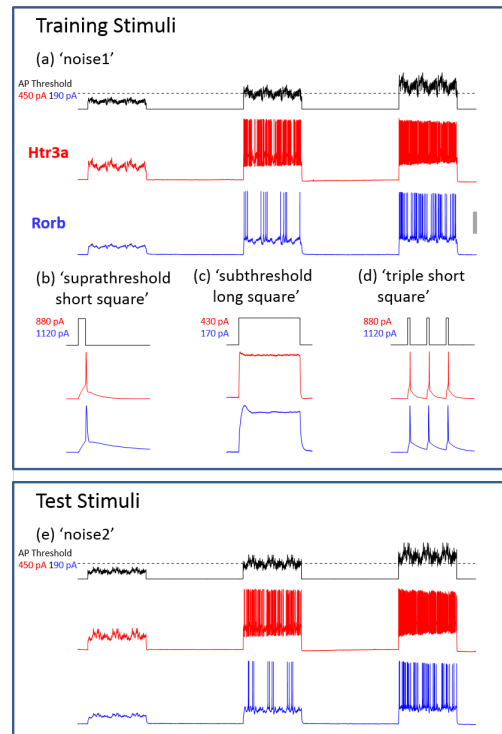
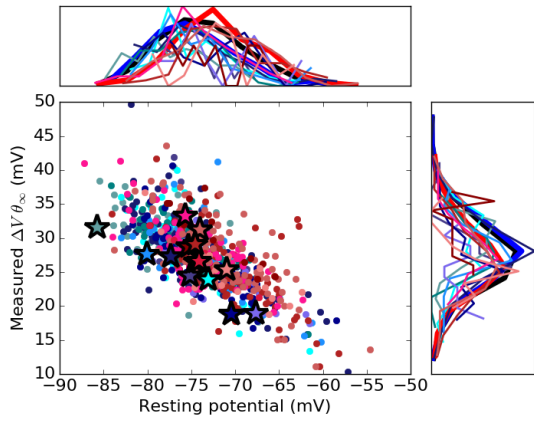
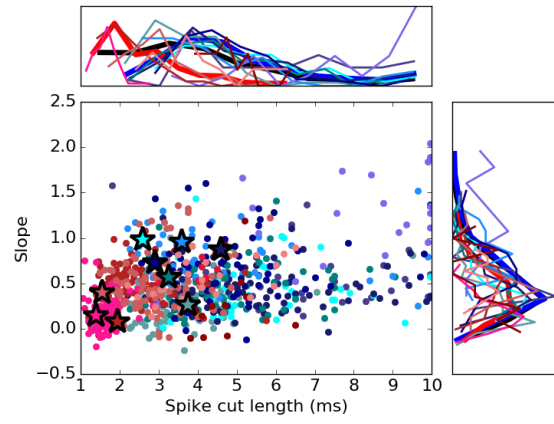


Figure 2: **771 different neurons from 14 transgenic lines containing all the required stimuli on the Allen Cell Types Database are considered in this study.** (left) Illustrated colors correspond to the different transgenic lines in all figures. 'n' describes the number of neurons for which the lowest level model ($GLIF_1$) could be generated. Transgenic lines can identify either inhibitory (I) or excitatory (E) cells which reside in layer 1 (L1) through layer 6 (L6). (right) A minimal set of stimuli were required for training and testing different GLIF models. GLIF models were trained using (a) at least two repeats of pink noise stimuli (3 s each, 1/f distribution of power, 1 - 100 Hz) with amplitudes centered at 75, 100, and 125 percent of action potential threshold, (b) a short (3 ms) just suprathreshold pulse in order to fit the instantaneous threshold, Θ_∞ , (c) a long square (1 s) pulse just below threshold to estimate the intrinsic noise present in the voltage traces (used in the *post hoc* optimization step), and (d) a series of three peri-threshold short pulse sets for any model with reset rules ($GLIF_2$, $GLIF_4$, and $GLIF_5$). GLIF models were then tested using a holdout stimulus set (e) of at least two sweeps of a second pink noise stimuli generated in an identical manner to the training but initialized with a different random seed. Representative data shown from an Htr3a-Cre positive neuron (resting membrane potential (RMP) -74 mV) and a Rorb-Cre positive neuron (RMP = -77 mV). Scale bar in (a) corresponds to 40 mV (panels a, b, d, e) or 15 mV (panel c).

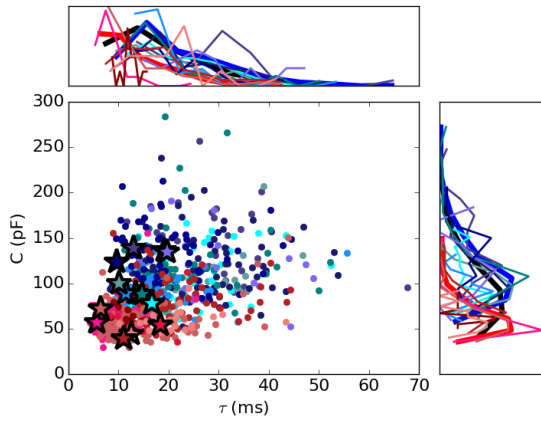
(a)



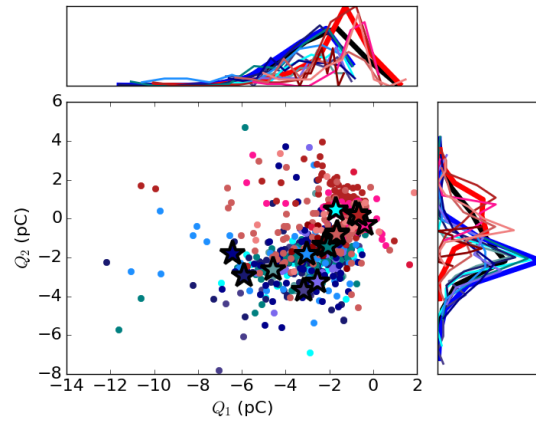
(b)



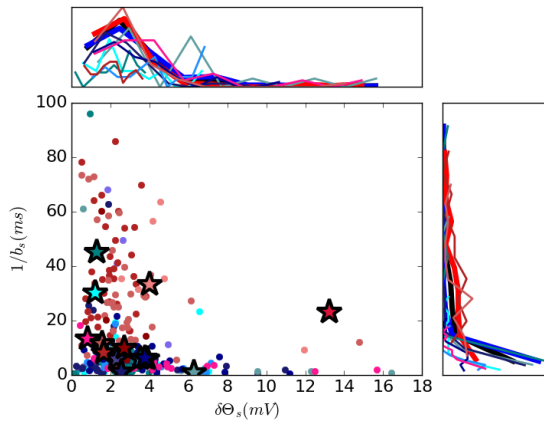
(c)



(d)



(e)



(f)

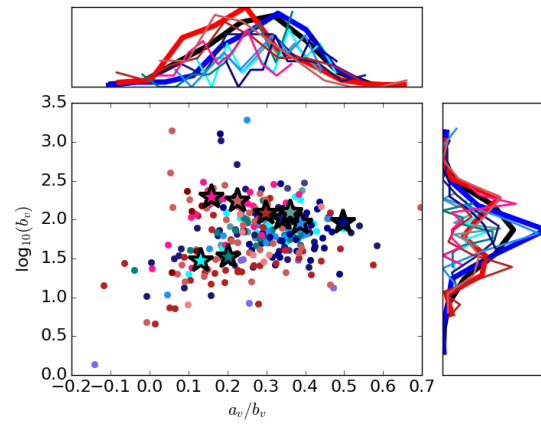


Figure 3: Slices from the complex parameter space fit from electrophysiological data. Five parameters are fit to the basic $GLIF_1$ (or LIF) model. Four additional parameters are fit to models with after-spike currents (ASC) ($GLIF_3$, $GLIF_4$, $GLIF_5$) and another two parameters are fit to models with reset rules (R) ($GLIF_2$, $GLIF_4$, $GLIF_5$). Finally two additional parameters are fit for the $GLIF_5$ model containing adapting threshold (AT), resulting in a total of 15 parameters fit for the most sophisticated model considered here (Table 1). Detailed parameter fitting methodology (Figure 2) is available in the Supplementary Material. (a) Resting potential is measured as the average voltage during rest before training noise (noise1) current is injected. The threshold relative to rest, ΔV , is measured by subtracting resting voltage from the threshold obtained from the suprathreshold short square pulse. (b) The spike waveform is removed from the voltage trace by aligning all spikes and fitting a line to the voltage before and after a spike. The best fit line within a window of 10 ms after spike initiation was chosen (Figure 8). This spike width is used in all models, and voltage measurements before and after a spike are used to reset voltage in ($GLIF_2$, $GLIF_4$, and $GLIF_5$). (c) Capacitance and resistance are fit via linear regression to subthreshold voltage (Figure 9). The membrane time constant, $\tau = RC$ is plotted. (d) Total charges of the fast Q_1 and slow Q_2 after-spike currents deposited each time there is a spike. (e) The amplitude, a_s , and decay b_s of the spiking component of the threshold, Δ_s , (used in $GLIF_2$, $GLIF_4$, and $GLIF_5$) is fit to the triple short square data set (Figure 10). (f) In the $GLIF_5$, the threshold is influenced by the voltage of the neuron according to (Equation 4). The two parameters of this equation are plotted here.

2.4 Model performance

After optimization, we quantified the model performance by how much of the temporal variance in the spike times of the data in response to a *in vivo* like stimulus can be explained by the model at a 10 ms temporal scale. Please see section 4.5 in the Supplementary Material. *GLIF*₁ (equivalent to a leaky integrate and fire model) has a surprisingly high explained variance of 70%. It is better for inhibitory than for excitatory neurons: 76% versus 67% (Figure 5, Table 4). The addition of reset rules and after spike currents independently do not significantly improve the explained variance. However, when both are jointly added, they improve the explained variance from 70% to 74% (86% for inhibitory and 71% for excitatory neurons). On top of that, the addition of the adaptive threshold provides a significant but smaller improvement from 74% to 76% (87% for inhibitory and 73% for excitatory neurons). Performance of individual transgenic lines can be viewed in the Supplementary Material (Figures 13 and 14)

2.5 Example neurons

In order to provide examples of models that do well reproducing the spike times of the different transgenic cell lines, for each line we select the model which has the highest explained variance from neurons that have all five model levels. Example neurons are labeled with stars in all figures (Figures 7, 9, and 10 and can be found on the website via the following specimen IDs: Pvalb: 481018305 *GLIF*₄, Rorb: 314822529 *GLIF*₅, Ctgf: 510106222 *GLIF*₄, Vip: 509604672 *GLIF*₅, Ndnf: 527464013 *GLIF*₅, Cux2: 490376252 *GLIF*₅, Chat: 519220630 *GLIF*₄, Ntsr1: 485938494, *GLIF*₃, Scnn1a-Tg2: 490205998 *GLIF*₅, Scnn1a-Tg3: 323834998 *GLIF*₃, Nr5a1: 469704261 *GLIF*₅, Htr3a: 477975366 *GLIF*₅, Sst: 313862134 *GLIF*₅, Rbp4: 490944352 *GLIF*₅. Although these neurons are the best performers they may not be the best representational cells. Each of these models have at least one parameter which lies outside the 5 to 95 percentiles for the cre line. To facilitate the choice of other good representative models, neurons that contain all GLIF model levels and all model parameters lie within 5 to 95 percentiles are available in Table 5 of the Supplementary Material.

2.6 Clustering

In order to assess how well clustering on GLIF parameters can potentially classify cell types, we perform a set of clustering experiments using the parameters obtained from the GLIF models. We compare the number of clusters from model results with those obtained from clustering the associated features extracted directly from the electrophysiological traces. We assess how well the putative clusters agree with known cell type information based on the transgenic lines from which the cells were derived. To perform clustering a large data set is required. Therefore we use only parameters that are available for all 771 cells. Table 1 describes the parameters used in each clustering paradigm.

We use an iterative binary splitting method using standard hierarchical clustering methods (Please see section "4.6 Supplementary Material"), run over 100 iterations to assess cluster robustness. Using only the parameters which characterize the *GLIF*₁ model results in eight clusters, with two clusters comprising predominantly inhibitory neurons. Using only parameters from the *GLIF*₃ model, eight distinct clusters emerge. The addition of reset rules, which are included in the *GLIF*₂ and *GLIF*₄ models, results in nine distinguishable putative cell types using the same clustering method (Figures available in Supplementary Material). However, including the full set of all optimized parameters results in 18 clusters (Figure 6a).

GLIF	Num. cells	Variables	Model parameters	Parameters in clustering	Explained Variance $\Delta t=10$ ms	Num. clusters
1	771	$V(t)$	$R, C, E_l, \Theta_\infty, \delta t$	$R, C, E_l, \Theta_\infty, \delta t$	70.0	8
2	289	$V(t), \Theta_s(t)$	$R, C, E_l, \Theta_\infty, \delta t, f_v, \delta V, b_s, \delta \Theta_s$	$R, C, E_l, f_v, \delta V, \Theta_\infty, \delta t$	69.7	9
3	771	$V(t), I_1(t), I_2(t)$	$R_{ASC}, C, E_l, \Theta_\infty, \delta t, k_1, \delta I_1, k_2, \delta I_2$	$R_{ASC}, C, E_l, \Theta_\infty, \delta t, \delta I_1/k_1, \delta I_2/k_2$	71.3	8
4	289	$V(t), \Theta_s(t), I_1(t), I_2(t)$	$R_{ASC}, C, E_l, \Theta_\infty, \delta t, f_v, \delta V, b_s, \delta \Theta_s, k_1, \delta I_1, k_2, \delta I_2$	$R_{ASC}, C, E_l, \Theta_\infty, \delta t, f_v, \delta V, \delta I_1/k_1, \delta I_2/k_2$	74.3	9
5	288	$V(t), \Theta_s(t), I_1(t), I_2(t), \Theta_v(t)$	$R_{ASC}, C, E_l, \Theta_\infty, \delta t, f_v, \delta V, b_s, \delta \Theta_s, k_1, \delta I_1, k_2, \delta I_2, a_v, b_v$	N/A	76.2	N/A
all param	N/A	N/A	N/A	$R, R_{ASC}, C, E_l, \Theta_\infty$ (from GLIF 1 & 3), $\delta t, f_v, \delta V, \delta I_1/k_1, \delta I_2/k_2$	N/A	18

Table 1: **Summary of GLIF models and results.** "GLIF" column reports the model level. The "Num. cells" column reports the number of cells for which a GLIF model is constructed. The variables for each model level are listed in the "Variables" column. The parameters fit for each model level are listed in "Model parameters". Note that resistance was fit along with after-spike currents in models where after-spike currents were implemented. R , denotes the resistance fit without ASC and R_{ASC} denotes the resistance fit along with ASC. Clustering is only performed on parameters that are available for all 771 neurons. Therefore, the 'Parameters in clustering' list is a subset of the total model parameters available for any level. $GLIF_5$ does not report clusters because there are no additional parameters available for clustering than in $GLIF_4$. We are unable to cluster on the time scales of every ASC alone as there were five discrete possible values but only two were chosen for each neuron. Therefore, we cluster on the total charge deposited over short $\delta I_1/k_1$ and long $\delta I_2/k_2$ time scales (continuous numbers) for the model levels that contain after-spike currents. The average explained variance at a time resolution of 10 ms for all neurons at each level is reported as well as the number of clusters that were found using the aforementioned clustering parameters. In the last row, optimized parameters available for all neurons were used in clustering.

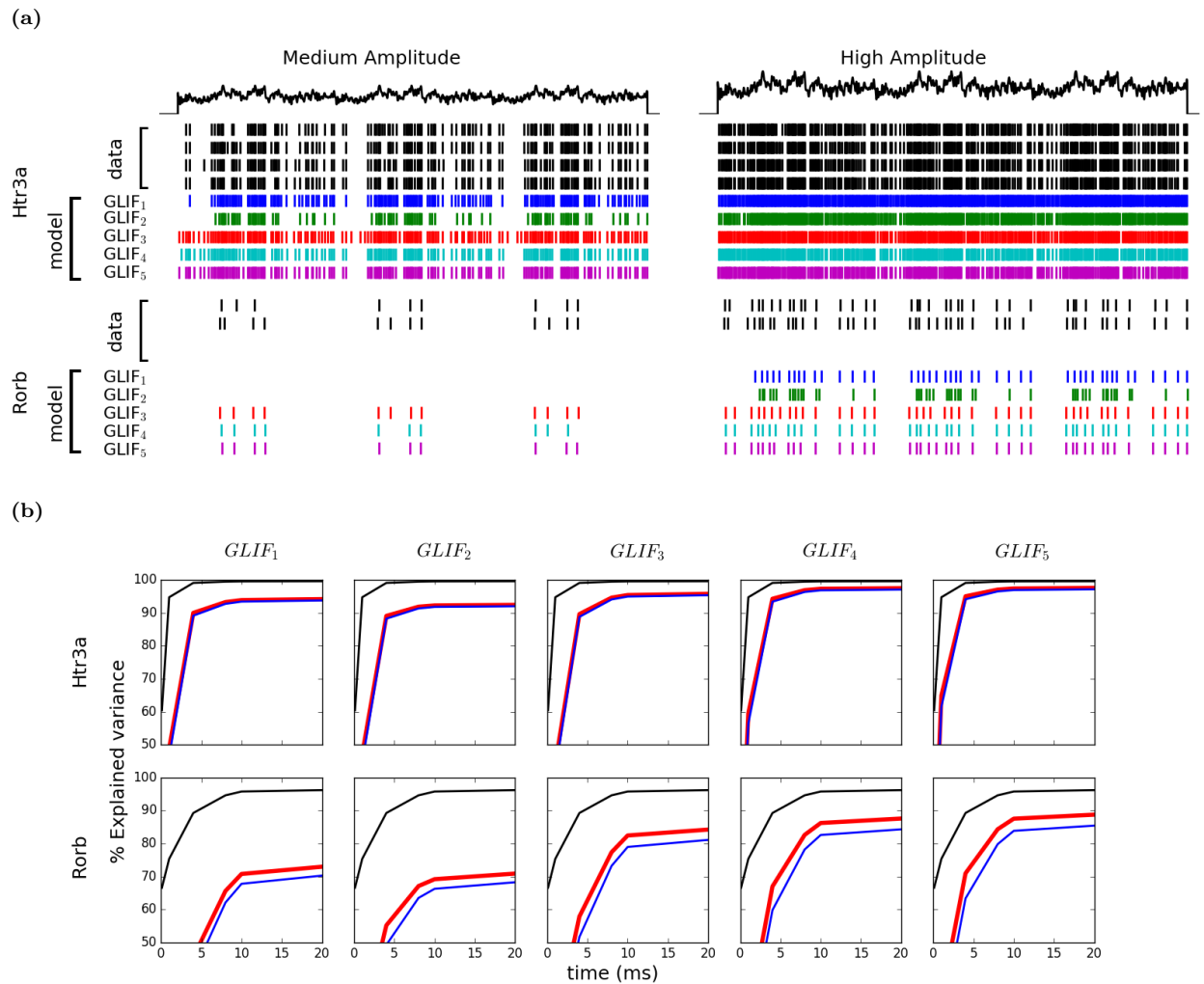


Figure 4: Rastergrams and explained variance of biological data and all optimized model levels for 'hold out' test data. (a) Data for the two example cells: Htr3a 477975366 and Rorb 314822529. Injected current shown in black. Black rasters are spikes from recorded neurons to repeated current injections. Colored rasters correspond to the five different, deterministic models. The current injection is 3 s long. It is observed that as the $GLIF_1$ and $GLIF_2$ do not have a spike frequency adaptation mechanism, they have trouble reproducing simultaneously the firing patterns at multiple input amplitudes. (b) Explained variance for the different model levels at different levels of time window resolution Δt . The black lines represent the explained variance of the data (how well the neuron repeats its own spiking behavior). This trace would reach 100% if the spike times of the data were all exactly at the same time in each repeated stimulus: the fact that they are not 100% reflects the intrinsic variation in the spike times within the experimental data. The blue line illustrates the pairwise explained variance of the model with the data. Because the model can not be expected to explain the data better than the data can explain itself, the red line is the ratio of the pairwise explained variance of the model (blue) and the data divided by the explained variance of the data (black). This ratio value at a $\Delta t=10$ ms time bin is used for the explained variance performance metric in the main text.

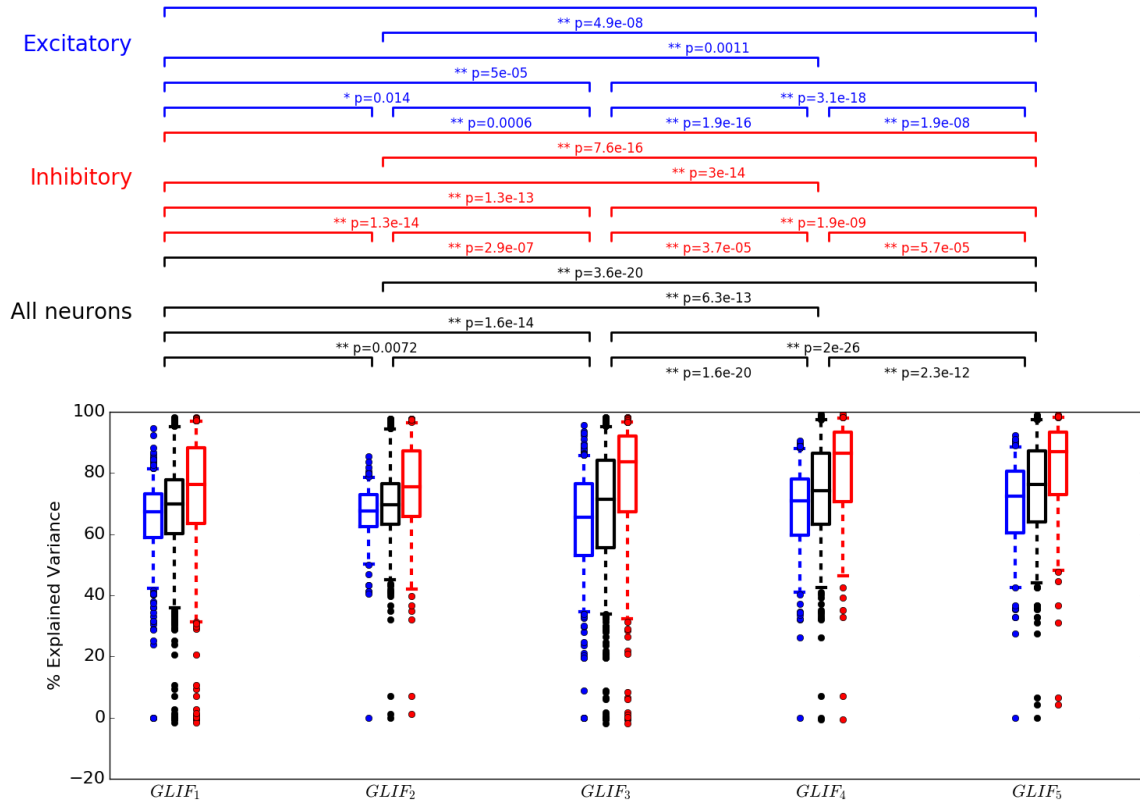


Figure 5: **Model performance improves most significantly with $GLIF_4$ model.** Explained variance summary for GLIF models for all 771 neurons (black), all 315 inhibitory neurons (red) and all 456 excitatory neurons (blue). Box plots display median, and quartiles. Whiskers reach to 5% and 95% quartiles. Individual data points lie outside whiskers. Brackets above plot denote significant difference between distributions assessed via a Wilcoxon sign ranked test: a single asterisk (*) represents a p-value < 0.05, a double asterisk (**) represents a p-value < 0.01. Distribution values are available in Table 4

Figure 6: **When clustering all 771 GLIF models we identify 18 discrete putative clusters. 11 electrophysiological cell types are obtained clustering based on the features extracted from traces.** (a) Summary of clusters obtained by iterative binary hierarchical clustering using all optimized model parameters 8. Each row represents a cluster, and each column a transgenic line. The size of the circle indicates the fraction of cells from a given transgenic line falling into a specific cluster (such that the sum of fractions in a column add up to 1). The dendrogram on the y-axis shows the iterative binary splitting into clusters using the algorithm explained in the text, and the numbers at the leaves and intermediate nodes indicate how often out of 100 clustering runs a given split or terminal cluster was identified. Although the segregation is not perfect, there is separation among lines labeling inhibitory and excitatory cells. In addition, transgenic Cre lines marking Pvalb+, Ntsr1+, and Ctgf+ cells tend to segregate into distinct clusters. (b) A summary of 11 electrophysiological cell types is obtained using the features extracted from traces (Table 10). Each row represents a cluster, and each column a transgenic line. The size of the circle indicates the fraction of cells from a given transgenic line falling into a specific cluster (such that the sum of fractions in a column add up to 1). The dendrogram on the y-axis shows the iterative binary splitting into clusters, as explained in the text, with the numbers at the leaves and intermediate nodes indicating how often out of 100 clustering runs a given split or terminal cluster was identified. Although there is separation among lines labeling inhibitory and excitatory cells, the cells from the Ctgf and Ntsr lines are not strongly separated from other cells among the clusters. (c) Schematic of numbers of clusters found with the parameters from each model level. (d) Overlap between GLIF and feature-defined clusters. There is a clear segregation between excitatory and inhibitory neurons. Among excitatory neurons the two classifications are somewhat distinct, with the GLIF clusters coming closer to the laminar distribution of neurons.

The transgenic line composition of each terminal cluster in Figure 6a suggests that clustering based on model fit parameters broadly segregates neurons into previously identified classes of cells. Neurons from transgenic lines labeling predominantly excitatory neurons cluster separately from those labeling mainly interneurons. In addition, sub-categories of neurons appear: these include parvalbumin-positive interneurons, Layer 6a corticothalamic neurons (derived from the *Ntsr1* transgenic line), and Layer 6b neurons (derived from the *Ctgf* transgenic line).

Examining the electrophysiological features of the recorded cells found in the clusters based on the full set of GLIF parameters reveals distinctive characteristics between some of the clusters (Table 10). Cluster 15 contains most of the parvalbumin-positive neurons and accordingly demonstrates many features associated with fast-spiking interneurons: a low action potential upstroke/downstroke ratio, a steep slope of the f-I curve, and a low membrane time constant. Clusters 16, 17, and 18 contains many of the other inhibitory interneurons from the *Sst* and *Vip* transgenic lines; these clusters exhibit higher median input resistances, the low upstroke/downstroke ratios (although not as low as the aforementioned cluster 15), and a relatively deep trough following the action potential. Interestingly, many of the cells from the *Ndnf* transgenic line (which labels inhibitory neurogliaform cells) are found in cluster 6 (and neighboring clusters 5 and 7) on the predominantly excitatory side of the tree. Cluster 6 has the lowest median membrane time constant on that side of the tree, and it exhibits a higher upstroke/downstroke ratio and longer average interspike interval than the other large inhibitory clusters described here.

Among the excitatory-neuron-dominated clusters, there is more similarity among the electrophysiological features, but some relevant differences can be observed. Clusters 9 and 10 (containing many neurons derived from the *Ctgf* transgenic line) exhibit some of the highest median upstroke/downstroke ratios across all clusters. Cluster 4 (containing many neurons derived from the *Ntsr1* transgenic line) has neurons with a shorter membrane time constant, lower upstroke/downstroke ratio, and higher rheobase ($I_{\text{threshold}}$ on long steps) than most other excitatory-dominant clusters. Clusters 5, 6, and 7 contain many of the excitatory cells that exhibit bursting behavior. Cluster 3 is notable for its low median input resistance, the shallow median slope of the f-I curve, and relatively high spike-frequency adaptation.

Interestingly, when the same clustering technique is applied directly to the electrophysiological feature data set (Figure 6b), the correspondence between clusters and the excitatory transgenic lines is somewhat weaker. In particular, the layer 6 neurons that emerged in separate clusters with the model parameters are not as well isolated into distinct groups here. Clusters 9, 10, and 11 appear to separate interneurons from the *Sst*, *Htr3a/Vip*, and *Pvalb* transgenic lines relatively well, while cells from the *Ndnf* transgenic line again cluster more closely with the excitatory cells. Overall, this suggests that dimensionality reduction onto the space of model fit parameters allows for discriminability similar to that obtained by the extraction of a subset of electrophysiological features.

3 Discussion

In the adult cortex, the majority of communication between neurons is via chemical synapses from axons onto dendritic or somatic membrane (with a fraction of inhibitory neurons coupled by gap junctions as notable exceptions). The response of these synapses is generally dependent only on the action potentials generated by the presynaptic cell. Thus, we focus on reproducing the temporal properties of spike trains using computationally compact simple GLIF models. We show that a) additional mechanisms included in GLIF models provide an increased explanatory power for test stimuli and b) that the low dimensional description generated by the models capture enough of the complexity of the neuron behavior such that the number of cell types obtained from clustering model parameters is modestly larger than the number of cell types resulting from clustering electrophysiological features.

3.1 Model performance of spike time reproducibility

The performance of different models is dependent on the properties of the input currents i.e. if there are large variations in the input current, the capacity to predict the spiking output is better. We chose to compute the performance metrics on stimuli which have the same coefficient of variation 0.2 as the currents measured with slice patching, and a 1/f (pink noise) power distribution of frequencies between 1 and 100 Hz, using two supra-threshold amplitudes. It is important to note that exact comparison of explained variance for models across different publications is difficult, as having stimuli with large temporal variations can lead to more precise firing (Mensi et al., 2012; Pozzorini et al., 2015).

After optimizing the neuronal parameters we were surprised on how well the leaky integrate and fire neuron model performs under such naturalistic conditions, explaining 70.0% of the variance on average. While the reset rules and the afterspike currents separately do not directly improve the model performance, together they lead to a small but significant increase to 74.3%. There is a further small but significant improvement, to 76.2%, with the introduction of the voltage-dependent threshold. While the model complexity increases, the fraction of explained variance is computed on a hold-out data-set, not the training data, such that over-fitting on the training data would result in a decreased test explained variance.

Of course, there are limitations to our models: the traces which are reproduced are coming from a current injection into the soma of the neuron in a stereotyped recording condition. We try to keep the statistics of the test stimulus similar to in vivo patch clamp currents, but in vivo additional large sources of variability will occur from stochastic transmission at synapses, complex dendritic integration, neuromodulation etc. These mechanisms, which are necessary for exact integration of our models into systems models are beyond the scope of the current study.

3.2 Neuronal Types

In the cerebral cortex there is a wide diversity of neuron types observed at transcriptomic levels Tasic et al. (2016). The existence of different types may be driven by a need to develop particular cell type specific connectivity, neuromodulation, or perform different cell specific computations. When constructing a system level model, it is important to know how many types of neurons are needed to describe the relation between an input current and the resulting output spiking pattern measured in electrophysiological experiments.

In an attempt to characterize cell types associated with the input/output transform measured by electrophysiological experiments, a clustering can be run on a set of electrophysiology features. However, it is not entirely clear which are the most important features to consider. An alternative method is use the entire spike train by synthesizing the input/output relationship into a model and then performing the clustering on the model parameters. A seemingly intuitive model to use for such a clustering would be biophysically detailed model, as the parameters map well to biological mechanisms. However, a problem with such models is that they do not provide unique solutions: repeated optimization with the same input

can lead to solutions which are far away in parameter space. Thus, another approach is to use simpler linear models which have both a) unique parameter solutions, and b) have been shown to reproduce spike times of biological data (Pozzorini et al., 2015). Indeed, previous studies have touched upon the potential for clustering using simplified models (Dong et al. (2013); Mensi et al. (2012)).

Clustering upon 11 fit and optimized GLIF parameters available for all 771 neurons (Table 1) we obtained 18 clusters using a simple iterative hierarchical clustering method. We were surprised at this large number considering that we started clustering with cells from 14 transgenic lines (all transgenic lines for which we had less than 7 neuron models were excluded) and only 11 clusters were obtained using 16 electrophysiological parameters (Supplementary Material Table 10). This suggests that the GLIF model parameters have at least a similar capacity of separating distinct clusters of cell as the features especially designed to do that. The unsupervised clusters do not perfectly match the transgenic lines, which is not surprising given that the transgenic lines are known to comprise multiple molecularly defined cell types. However, the segregation of neurons into putative types based on the model fit parameters does recapitulate some expected differences among classes of neurons based on other data modalities.

It seems that the mechanism which is most responsible for the increase in the capacity to classify cells is combination of the resistances fit both with and without afterspike currents and the optimized th_∞ for $GLIF_1$ and $GLIF_3$ models (optimized th_∞ values for $GLIF_2$, $GLIF_4$, and $GLIF_5$ are not available for all neurons).

3.3 Conclusion

Here we conclude that GLIFs are an exceptionally useful tool for single neuron models. They can simultaneously reproduce the spike times of biological neurons and reduce the complex mapping between the input current to the spike train output to a small number of parameters. These parameters synthesize the richness of the observed neuronal behaviors to allow the same number of classes to be obtained from models as there are from features characterizing the data.

We provide a database of GLIF models that can facilitate the development of system models with cell types available on the Allen Cell Types Database at <http://celltypes.brain-map.org/>.

4 Supplementary Material

4.1 Model Definitions

The five different GLIF models are described below, in order of increasing level of complexity, with their evolution equations and reset rules. For all the models, C represents the membrane capacitance, R , is the membrane resistance, E_L is the resting potential and $I_e(t)$ is an external current injected into the cell. t_+ and t_- represent the time just after and before a spike respectively.

4.1.1 Leaky Integrate-And-Fire ($GLIF_1$)

The traditional Leaky Integrate-And-Fire (LIF) neuron is a hybrid system characterized by an evolution equation for the membrane potential $V(t)$. We refer to this model as $GLIF_1$ as it is the starting point for further GLIF models.

$$V'(t) = \frac{1}{C} \left(I_e(t) - \frac{1}{R} (V(t) - E_L) \right) \quad (9)$$

When the membrane potential becomes larger than a threshold $V(t) > \Theta$ (which in this case is just the instantaneous threshold, Θ_∞), a reset rule is evoked which sets the membrane voltage to a reset voltage V_r .

$$V(t) \leftarrow V_r \quad (10)$$

4.1.2 Leaky Integrate-And-Fire with biologically defined reset rules ($GLIF_2$)

$GLIF_2$ has an evolution equation for two state variables, the membrane potential $V(t)$ and a spike-dependent threshold component $\Theta_s(t)$ which is updated by spikes and decays back to zero with a time constant $(1/b_s)$.

$$\begin{aligned} V'(t) &= \frac{1}{C} \left(I_e(t) - \frac{1}{R} (V(t) - E_L) \right) \\ \Theta'_s(t) &= -b_s \Theta_s(t) \end{aligned} \quad (11)$$

along with a reset rule for both voltage and the spike-dependent threshold if the membrane potential becomes larger than a threshold $V(t) > \Theta_\infty + \Theta_s(t)$

$$\begin{aligned} V(t_+) &\leftarrow E_L + f_v \times (V(t_-) - E_L) - \delta V \\ \Theta_s(t_+) &\leftarrow \Theta_s(t_-) + \delta \Theta_s \end{aligned} \quad (12)$$

The update to the spike-dependent component of the threshold is additive, with $\delta \Theta_s$ added after every spike. The update to the membrane potential has a multiplicative coefficient f_v and an additive constant δV

4.1.3 Leaky Integrate-And-Fire with after-spike currents ($GLIF_3$)

In GLIF models, the rapid membrane potential fluctuation due to the fast voltage-activated (i.e. sodium and potassium) ion currents during a spike, is considered separately from the slow, subthreshold region of the membrane potential and is incorporated into the reset rules. However, ion currents activated by a spike could have effects over longer time scales. After the stereotypical, sharp membrane potential transitions during an action potential are eliminated, the longer term effects are modeled as additional currents I_j with pre-defined time constants $(1/k_j)$,

$$\begin{aligned} I'_j(t) &= -k_j I_j(t); \quad j = 1, \dots, N \\ V'(t) &= \frac{1}{C} \left(I_e(t) + \sum_j I_j(t) - \frac{1}{R_{ASC}} (V(t) - E_L) \right) \end{aligned} \quad (13)$$

Here R_{ASC} is used to explicitly point out that the resistance is fit along with the after spike currents. The update rule, which applies if $V(t) > \Theta_\infty$, is given by:

$$\begin{aligned} I_j(t_+) &\leftarrow f_j \times I_j(t_-) + \delta I_j \\ V(t_+) &\leftarrow V_r \end{aligned} \quad (14)$$

with the currents updated using a multiplicative constant $f_j = \exp(-k_j \delta t)$ (because after spike currents decay through the spike cut length) and an additive constant, δI_j .

4.1.4 Leaky Integrate-And-Fire with biologically defined reset rules and after-spike currents ($GLIF_4$)

Combining the biological reset rules model, $GLIF_2$ with the after-spike current model $GLIF_3$ described above gives a model defined by:

$$\begin{aligned} I'_j(t) &= -k_j I_j(t); \quad j = 1, \dots, N \\ V'(t) &= \frac{1}{C} \left(I_e(t) + \sum_j I_j(t) - \frac{1}{R_{ASC}} (V(t) - E_L) \right) \end{aligned} \quad (15)$$

$$\Theta'_s(t) = -b_s \Theta_s(t) \quad (16)$$

The update rule, which applies if $V(t) > \Theta_s(t) + \Theta_\infty$,

$$\begin{aligned} I_j(t_+) &\leftarrow f_j \times I_j(t_-) + \delta I_j \\ V(t_+) &\leftarrow E_L + f_v \times (V(t_-) - E_L) - \delta V \end{aligned} \quad (17)$$

$$\Theta_s(t_+) \leftarrow \Theta_s(t_-) + \delta \Theta_s \quad (18)$$

4.1.5 Leaky Integrate-And-Fire with biologically defined reset rules, after-spike currents and a voltage dependent threshold ($GLIF_5$)

For the $GLIF_5$ model, the state variables are the membrane potential $V(t)$ (as in all GLIF models), a threshold component which is evoked when there is a spike (Θ_s) (as introduced in $GLIF_2$), a set of after-spike currents $I_j(t)$ (as introduced in $GLIF_3$), and an new additional threshold parameter that is dependent of the membrane potential (Θ_v). It is assumed that these state variables evolve in a linear manner between spikes:

$$\begin{aligned} I'_j(t) &= -k_j I_j(t); \quad j = 1, \dots, N \\ V'(t) &= \frac{1}{C} \left(I_e(t) + \sum_j I_j(t) - \frac{1}{R_{ASC}} (V(t) - E_L) \right) \end{aligned} \quad (19)$$

$$\begin{aligned} \Theta'_s(t) &= -b_s \Theta_s(t) \\ \Theta'_v(t) &= a_v (V(t) - E_L) - b_v \Theta_v(t) \end{aligned}$$

Where $1/b_v$ is the time constant of the voltage-dependent component of the threshold and, a can be interpreted as a 'leak-conductance' for the voltage-dependent component of the threshold.

If $V(t) > \Theta_v(t) + \Theta_s(t) + \Theta_\infty$, a spike is generated and the state variables are updated:

Mechanism	Symbol	Variable
	$V(t)$	Membrane Potential
ASC	$I_j(t)$	After-spike currents
reset	$\Theta_s(t)$	Spike-dependent threshold component
adapt. th.	$\Theta_v(t)$	Voltage-dependent threshold component

Table 2: Variable description

Mechanism	Symbol	Parameter	Fit From	Post-hoc Optimized
	C	capacitance	V during low amplitude noise	No
	G	membrane conductance	subthreshold V during noise	No
	E_L	resting potential	resting V before noise	No
	Θ_∞	instantaneous threshold	short square input	Yes
reset	f_v	voltage fraction following reset	all noise spikes	No
reset	δV	voltage addition following reset	all noise spikes	No
reset	b_s	spike-induced threshold time constant	triple short square	No
reset	$\delta\Theta$	threshold addition following reset	triple short square	No
ASC	δI_j	afterspike current amplitudes	subthreshold V during noise	No
ASC	k_j	afterspike current time constants	subthreshold V during noise	No
adapt. th.	a_v	adaptation index of threshold	prespike V during noise	No
adapt. th.	b_v	voltage-induced threshold time constant	prespike V during noise	No

Table 3: Parameter description

$$\begin{aligned}
 I_j(t_+) &\leftarrow f_j \times I_j(t_-) + \delta I_j \\
 V(t_+) &\leftarrow E_L + f_v \times (V(t_-) - E_L) - \delta V \\
 \Theta_s(t_+) &\leftarrow \Theta_s(t_-) + \delta\Theta_s \\
 \Theta_v(t_+) &\leftarrow \Theta_v(t_-)
 \end{aligned} \tag{20}$$

For all state variables $X_k(t)$, the value immediately after the spike, $X_k(t_+)$ was related to the value immediately before the spike, $X_k(t_-)$ via a set of update parameters: f_k represented the fraction of the prespike value of X_k which is maintained after the spike and δX_k represents the values updated by a spike.

4.2 Parameter fitting and distributions

Model parameters are extracted from the electrophysiology data as described below. Python code for all algorithms can be found in the Allen Software Development Kit (SDK) on the Allen Cell Types Database website. All code was executed using Python 2.7. In Figures 7,8, 9, and 10 we show slices of the parameter space obtained via fitting.

Spike initiation time and threshold [t_s, Θ_{bio}]: Spikes are typically detected by the following procedure. The upstroke of a spike is defined as the portion of the action potential wave form between the instant at which the rise in potential for a given time exceeds 20 mV/ms and the peak of the action potential. For each neuron the average value of the maximum dV/dt during the upstrokes of all action potentials is calculated. The time of spike initiation and threshold is then defined by the time and voltage at which dV/dt reaches

5% of the average maximum dV/dt during an upstroke.

For stimuli that involve a short, intense current pulse stimulation, the passive rise in voltage preceding a spike can exceed the standard 20 mV/ms initial detection value, leading to inaccurate spike time identification. In those cases, the maximum dV/dt is measured during the current pulse, and the spike detection value is adjusted to be 10% higher than that. In addition, the threshold identification value (typically 5% of the average maximum dV/dt) is also adjusted to ensure that it is at least 20% higher than the dV/dt at the end of the stimulus pulse.

Spike cut length and voltage reset $[\delta t, V_{t+}]$: GLIF models aim to reproduce the timing of the spikes via sub-threshold data (not the voltage waveform dictated by the highly non-linear ion fluctuations during the action potential). Therefore, the action potential waveforms are removed from the voltage trace. Here we use a principled way to estimate the duration of the spike that is removed and the reset voltage in models where biological reset rules are implemented (*GLIF₂*, *GLIF₄*, and *GLIF₅*). Action potentials in a neuron have stereotyped shapes with a sharp rise, followed by a hyperpolarized refractory period and a return to a voltage that is most often lower than the voltage before the spike (voltage reset). As shown in Figure 8, we align all of the action potentials of the training noise data (noise 1) to the spike initiation and ask at what time within a window of 1 to 10 ms after spike initiation does the post spike voltage best predict the voltage at spike initiation given a linear dependence. i.e. a line is fit between the voltages at spike initiation (prespike voltages) and the voltages at each time point after spike initiation (postspike voltages). The fit which minimizes the residuals is chosen to define the time at which the spike ends. In all models, the duration of the spike is removed from the voltage traces. For the *GLIF₂*, *GLIF₄*, and *GLIF₅* models, the postspike voltage is reset via the best fit linear model $V_{t+} = E_L + f_v * (V_{t-} - E_L) + \delta V$, where V_{t-} is the voltage of the model before the spike, δV is the voltage intercept, f_v is the slope of the fit, and V_{t+} is the voltage of the model after the spike.

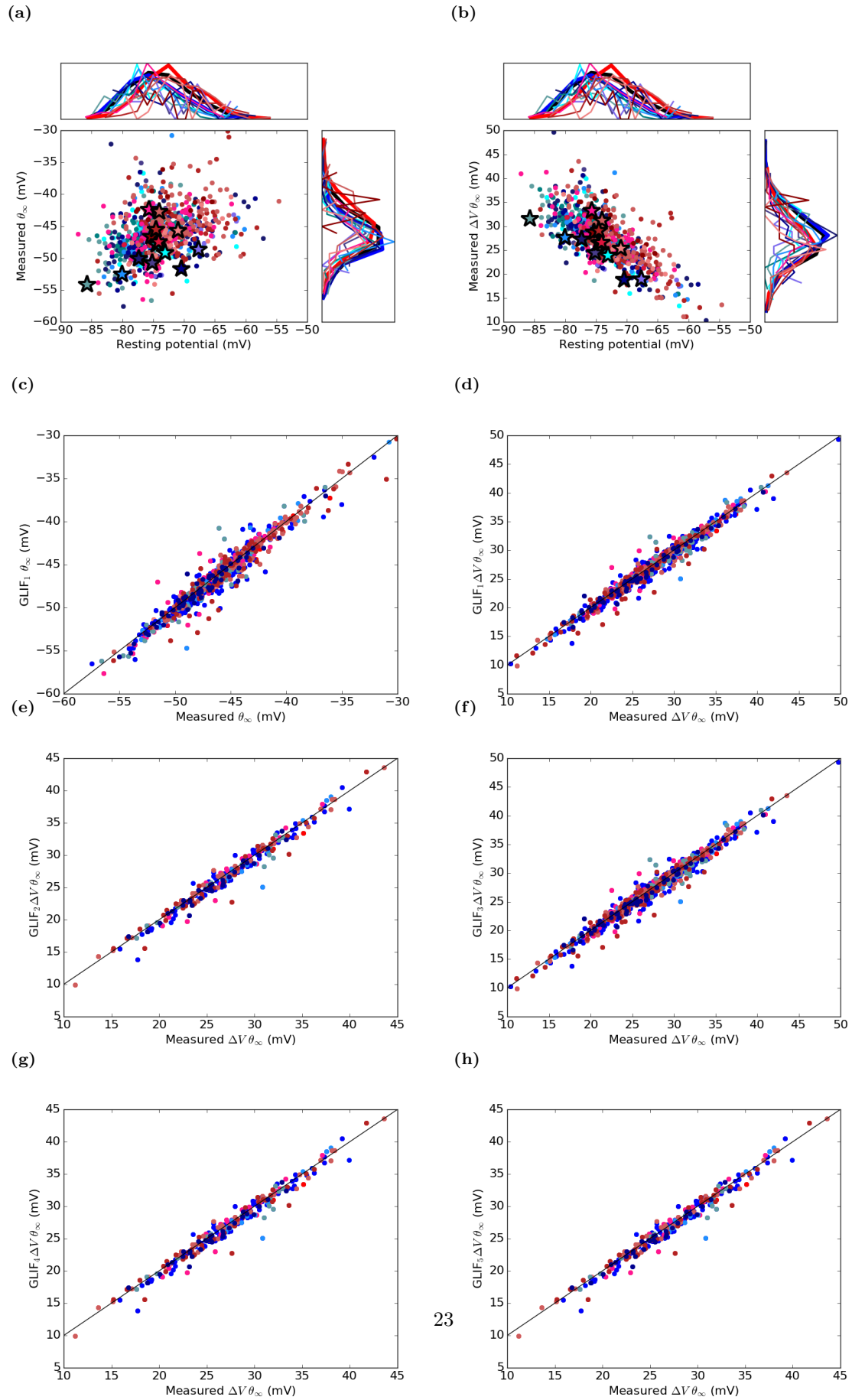


Figure 7: Resting potential and instantaneous threshold for biologically measured values and post-hoc optimized models. In (a) absolute threshold is shown, in (b) threshold is in reference to the resting membrane potential for the biologically measured threshold from the short square stimulus (Figure 2Stimuli). This figure is also shown in the main text (Figure 3) (c) and (d) show the similarity between threshold values before and after post-hoc optimization for absolute threshold and (c) and relative threshold (d). (e), (f), (g), and (h) compare the biologically measured relative threshold versus the relative threshold obtained by optimizing θ_∞ of (e) *GLIF*₂, (f) *GLIF*₃, (g), *GLIF*₄, and (h) *GLIF*₅. Black lines represents unity. Shades of blue represent excitatory, transgenic lines, shades of red represent inhibitory lines. The full list of colors corresponding to specific transgenic lines can be found in Table 2Transgenic Lines. Normalized histograms of the data are shown in the side panels. The thick black histogram represents data from all transgenic lines, the thick blue line represents all data from excitatory and the thick red represents inhibitory lines. Stars denote example neurons shown throughout the manuscript. All neurons with a measured instantaneous threshold below -60 mV were globally excluded from the data set.

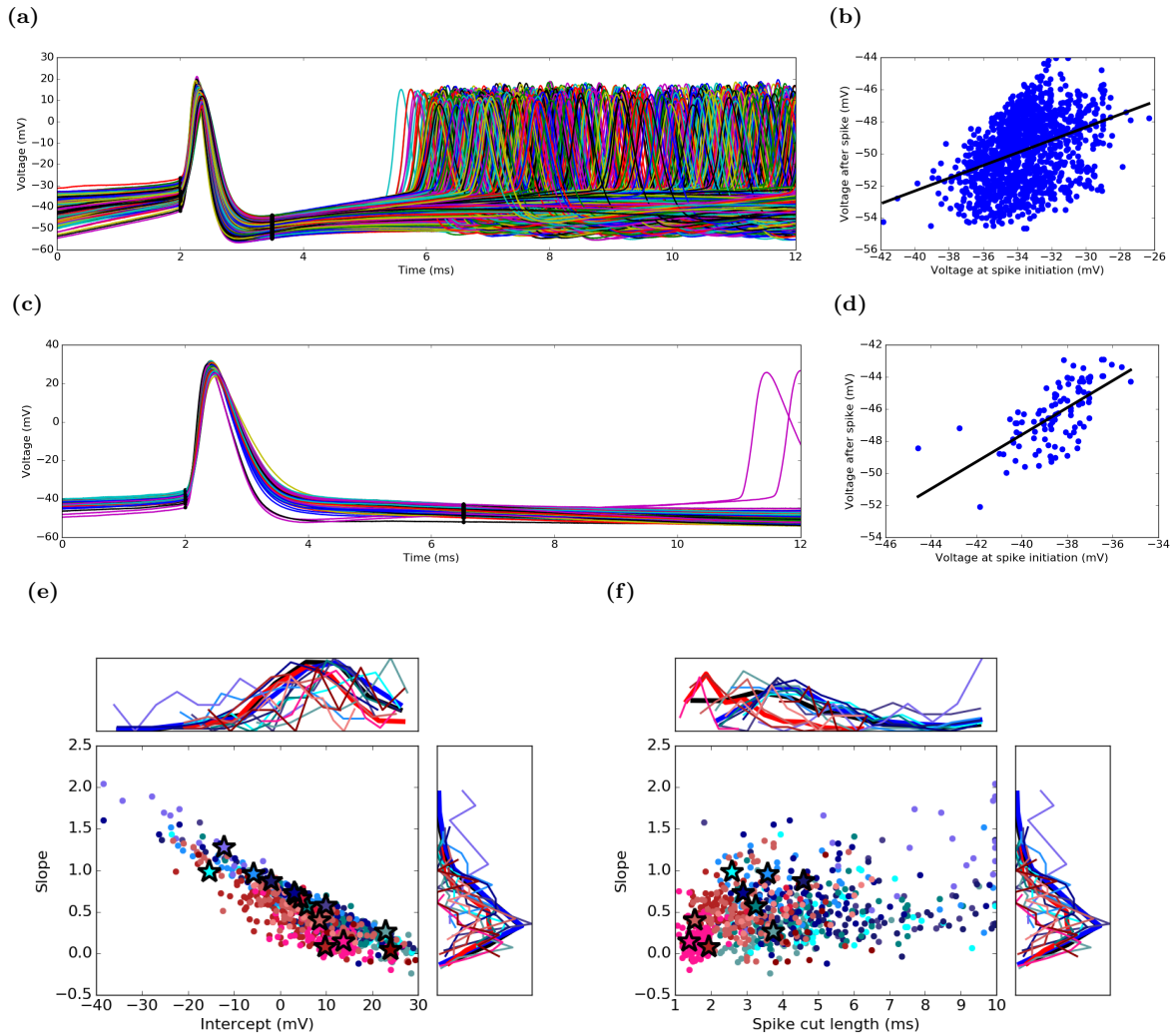


Figure 8: The action potentials waveforms are excluded from model fitting but are used to derive voltage reset rules. (a-d) illustrate examples of the spike cutting and reset rule calculation from two example neurons (a, b are from the Htr3 477975366 neuron; c,d are from the Rorb 314822529 neuron) from two different lines. (a, c) All spikes from the noise 1 stimuli (Figure 2Stimuli) are aligned to spike initiation. Here different colors represent different spikes of one neuron (not different transgenic lines). Dots represent spike initiation and termination found by minimizing the residuals from a linear regression between pre- and post-spike voltage in a window 1 to 10 ms after spike initiation (b, d). In the (*GLIF*₂, *GLIF*₄, and *GLIF*₅) models, voltage reset is calculated by inserting the model voltage when it reaches threshold into the equation defined by the line (Equations 5, 12, 17, 20). (e) and (f) show distributions for slope, intercepts, and the spike cut length for all neurons. Normalized histograms of the data are shown in the side panels. All neurons with an intercept < 30 mV were globally excluded from the data set. (f) is a replicate of the figure shown in the main article (Figure 3).

Instantaneous threshold [Θ_∞]: All models (neurons) spikes when the voltage crosses threshold. An estimate of the instantaneous threshold, Θ_∞ , is the voltage at spike initiation of the lowest amplitude supra-threshold short square (Figure 2Stimuli). Threshold is a critical parameter affecting the general excitability of the cell. The relationship between voltage and stimulus likely becomes non-linear near threshold. Because small errors in Θ_∞ measurement will greatly influence spiking behavior Θ_∞ is tuned during the post-hoc optimiza-

tion of every model such that the likelihood of the observed spike train in the training stimulus is maximized.

Resting potential $[E_L]$: The resting potential was defined as the mean resting potential of the noise 1 sweeps calculated by averaging the pre-stimulus membrane potential.

Resistance $[R]$, Capacitance $[C]$, and afterspike currents $[I_j(t)]$: Because the GLIF model is linear, many of the parameters can be fit via linear regression on subthreshold data similar to the methods in Pozzorini et al. (2015). Initially, R and C are calculated via a linear regression of the subthreshold noise in the first epoch of noise stimulation within the three-epoch noise sweeps (this stimulus does not elicit any spikes) as observed in Figure 9d. The neurons which elicited spikes during the subthreshold stimulation were eliminated from subsequent fits. Please see Supplementary Material for example calculations. These values of resistance are used for models that do not include after-spike currents ($GLIF_1$, $GLIF_2$).

For the GLIF model levels where after-spike currents are included ($GLIF_3$, $GLIF_4$, $GLIF_5$), after spike currents and resistance are fit via a linear regression on the periods between the spikes in the supra-threshold noise data. During this calculation, the capacitance and resting potential are forced to the previously calculated values. We choose to force capacitance while continuing to fit resistance because capacitance is a consistent property of the membrane which does not change based on the state of a neuron. However, the resistance of membrane will depend on the state of a neuron and will most likely differ with the activation of ion currents whose subthreshold affects are models here with after-spike currents.

After spike currents are modeled using exponential decaying basis functions with an amplitude and a time scale (the dynamics of the exponential functions are described in equation 15). The time scales and amplitudes are obtained by providing two from a set of five basis currents with varied time scales ($1/k_j$) ([3.33, 10, 33.3, 100, 333.33] ms) to a generalized linear model (GLM). The GLM (implemented via Python's statsmodels.api.GLM) is used to calculate the amplitudes corresponding to each basis current and the resistance by regressing the sum of the derivative of the voltage and the external current divided by the capacitance ($\frac{dV}{dt} + \frac{I_L}{C}$) against the basis currents and the leak term. C and E_L are calculated as mentioned in the above text. The GLM is run with all combinations of two of the five possible time scales leading to a choice among (${}^5C_2 = 10$) pairs of basis currents. The optimal pair of basis currents that yield the maximum log-likelihood value are chosen. The total charge, Q, deposited by one after-spike current in the neuron per spike is $\delta I_j/k_j$. Since total charge is a biologically relevant parameter and it is easier to perform clustering on continuous values, we often use the total charge instead of the individual parameters, δI_j , and, k_j (which is discrete), represented in the equations (Figures 3, 9, and Tables 6, 8)

Spike initiated component of the threshold $[\Theta_s]$: The evolution of the threshold of ($GLIF_2$, $GLIF_4$, and $GLIF_5$) contains a spiking component of the threshold, Θ_s . This contribution to the threshold represents the effect spiking has on the threshold of the neuron due to the inactivation of a voltage-dependent sodium current. This inactivation can be interpreted as a rise in the threshold of the neuron, and the movement from the inactive to a closed state can be modeled as a linear dynamical process. The change in threshold is fit with an exponential, and the values of its amplitude and time constant are calculated from the triple short square sweep set data as seen in Figure 10. For each pulse in the triple short square data that produces a spike, the voltage at spike initiation (the threshold) is calculated. The mean of the threshold of the first spike of each individual triple short square stimulus was taken to be the reference threshold. For all spikes that are not first spikes in each individual triple stimulus, the time since the last spike is calculated (referred to as an ISI) along with the threshold relative to the reference threshold. An exponential is then fit to the ISI versus threshold data. The exponential is forced to decay to the reference threshold.

Subthreshold voltage dependent component of the threshold $[\Theta_v]$: $GLIF_5$ contains both a spiking component of the threshold, Θ_s , (see above) and a subthreshold voltage component of the threshold, Θ_v . The subthreshold voltage component represents the effect of subthreshold membrane potential on the threshold (e.g., voltage dependence of sodium current reactivation). The voltage component of the threshold evolves according to $\Theta'_v(t)$ in equation 19. In order to fit a_v , and b_v the sum of the squared error between the voltage

component of the threshold of the biological neuron Θ_{vbio} and the voltage component of the threshold of the model Θ_v is minimized. Θ_{vbio} is found by subtracting the spike component of the threshold, Θ_s from the actual value of the voltage at the time of spike initiation Θ_{bio} . $\Theta_{vbio} = \Theta_{bio} - \Theta_s$. Although one could simulate the voltage component of the threshold using Euler's method, the analytical solution (for derivation please see Supplementary Material) was used to speed up the minimization process. Minimization was performed using a Nelder-Mead simplex algorithm (fmin in the scipy optimization tool box: `scipy.optimize.fmin`).

4.3 Evaluation of Model Spike Times

After all parameters of the model were optimized, the time-evolution of the models was determined using an Euler exact method. The spike times of the model were evaluated against the spike times of the neuron using the 'explained temporal variance' metric described below (Figure 4). This metric describes how well the temporal dynamic of the spiking response is captured by the model for a particular level of temporal granularity Δt .

A spike train is represented as a time series, at resolution Δt , of binary numbers. All numbers in a spike train are zero unless a spike occurs: a spike was denoted with a one. Any spike train can be converted into a single peristimulus time histogram, stPSTH, by convolving a spike train with a Gaussian.

In cases with many repeats of the same stimulus, a peristimulus time histogram (PSTH) could be calculated by taking the mean of the stPSTH at each instant in time. If there were n stPSTH_{*i*} denoted with index *i*, where *i* goes from 1 to n , $i=1,2,\dots,n$, the PSTH was equal to the column mean of the stPSTH_{*i*}:

$$PSTH = \sum_i^n \frac{stPSTH_i}{n}$$

The variance in spiking output of neurons is described by the variance of the PSTH. The explained variance (EV) between any two PSTHs (multiple or single train) is:

$$EV(PSTH_1, PSTH_2) = \frac{var(PSTH_1) + var(PSTH_2) - var(PSTH_1 - PSTH_2)}{var(PSTH_1) + var(PSTH_2)}$$

The explained temporal variance by the mean across-trials PSTH of the neuron, EV_D is the upper limit on how well the model can perform:

$$EV_D = mean_i (EV(stPSTH_{D_i}, PSTH_D)) \quad (21)$$

where *i* denotes the *i*th data stPSTH and n denotes the total number of data sweeps.

Since the models here were deterministic, the pairwise explained variance of the data with the model is calculated by taking the mean of the explained variance between every data stPSTH_{*D*} and the model PSTH_{*M*}.

$$pwEV_{DM} = mean_i (EV(stPSTH_{D_i}, PSTH_M)) \quad (22)$$

where *i* denotes the *i*th data stPSTH and n denotes the total number of data sweeps.

The ratio of the pwEV of the model to the data versus the pwEV within the data is bounded from below, by 0, where nothing is explained, and from above by 1, explaining the data as best as possible. Explained variance illustrated in Figure 4 for the two exemplary canonical neurons and reported. The ratio of the pwEV of the model to the data versus the pwEV within data at a resolution of $\Delta_t = 10$ ms converted to a percentage is used as the explained variance metric used throughout the manuscript (Figures 5, 13, 14 and Table 5)

4.4 Post-Hoc Optimization

An additional non-linear optimization step was performed to optimize the instantaneous threshold using a Nelder-Mead simplex algorithm (Python scipy module `scipy.optimize.fmin`). The instantaneous threshold

is an essential parameter governing the excitability of the neuron. As the current/voltage dependence for the neuron becomes highly nonlinear near the threshold, and we rely on linear models, to better fit the excitability of the models we search for the value of the instantaneous threshold which best predicts the spike train during the test stimulus. Comparing Figure 5 of the main text with Figure 11 illustrates the importance of the post-hoc optimization on explained variance.

4.4.1 Forced-spike Model Paradigm

During optimization, each spike should be fit without erroneous after-spike current history affecting the fitting of the subsequent spikes. The model voltage, threshold, and after-spike amplitudes are forced to reset at the time the biological neuron spikes. The reset rules are denoted in the GLIF model section above. When a model is run in its normal forward-running manner, the prespike values of the voltage, $V(t_-)$, threshold, $\Theta(t_-)$, and AS currents, $I_j(t_-)$, are the values of the model at the point where the model voltage crosses the model threshold. In the forced spike paradigm, to ensure that the voltage model is reset below the threshold of the model, the prespike voltage was set equal to the prespike threshold, $V(t_-) = \Theta(t_-)$ at the time of the neuron spike. Note that this reset only affects $GLIF_2$, $GLIF_4$, $GLIF_5$, where the prespike voltage affects the postspike voltage.

4.4.2 Objective Function: Maximum Likelihood Based on Internal Noise (MLIN)

Similar to a series of previous studies Paninski et al. (2004); Dong et al. (2011); Mensi et al. (2012); Pozzorini et al. (2015), the likelihood that the observed spike train was obtained by the model was maximized. However, the exact method of constructing the likelihood was different in that the noise is not tuned but rather uses a direct estimate of the biological neurons internal noise (Maximum Likelihood based on Internal Noise) was used.

Because these GLIF models are deterministic, estimating the likelihood required adding a source of noise external to the model. Rather than searching for the parameters of this noise, a parametric description of the internal noise of each neuron is used (Figure 12). As this internal noise could depend on the membrane potential, and the most relevant potential is near threshold, the variation in membrane potential during the steady state period of the largest subthreshold square pulse response is characterized.

The probability density of the neuron being at a potential v away from its mean is fit with an exponentially decaying function:

$$p(v) = \frac{1}{2\delta v} \exp\left(-\frac{|v|}{\delta v}\right) \quad (23)$$

This variation was considered to be additive to the membrane potential of the deterministic neuron. This allowed the computation of the probability that a neuron with the given noise would produce a spike if the deterministic model had a difference between threshold and membrane potential:

$$\Delta V(t) = \Theta(t) - V(t) = \Theta_s(t) + \Theta_v(t) + \Theta_\infty - V(t) \quad (24)$$

$$p_{spike}(t) = \int_{\Delta V(t)}^{\infty} p(v)dv = \int_{\Delta V(t)}^{\infty} \frac{1}{2\delta v} \exp\left(-\frac{|v|}{\delta v}\right)dv = 1 - c(\Delta V(t)) \quad (25)$$

where c represents the cumulative distribution of the intrinsic noise.

The likelihood of the model producing a set of spikes at the times the biological neuron produces them is:

$$p_{spikes} = \prod_{t \in t_s} p_{spike}(t) \quad (26)$$

However, the model neuron must both produce spikes when the biological neuron does, and not produce spikes when the biological neuron does not.

The likelihood of a model neuron not producing spikes is not independent for two nearby time points, as the intrinsic noise has a nontrivial autocorrelation. To estimate the likelihood of the model neuron not producing spikes at the times the biological neuron does not produce spikes, one sample for the internal noise is drawn for each time period of the autocorrelation, and following this time scale a new independent sample is drawn. As such, the inter-spike intervals are binned with bin sizes equal to the autocorrelation (typical autocorrelations time scales are much smaller than the inter-spike intervals). The grid times start from a spike time and advance by the autocorrelation time scale, ending at a predefined short time (5 ms) before the next spike. To estimate the likelihood of a neuron not producing a spike within a bin, the minimal difference between threshold and membrane potential within the bin is chosen which generates the grid differences:

$$p_{nospike} = \prod_{t \in t_{grid}} c(\Delta V_{grid}) \quad (27)$$

The log likelihood of the entire spike train being exactly reproduced was

$$LLIN = \text{Log}(p_{spikes} p_{nospike}) = \sum_{t \in t_s} (1 - c(\Delta V(t))) + \sum_{t \in t_{grid}} c(\Delta V_{grid}) \quad (28)$$

4.4.3 Simplex Method

The instantaneous threshold, Θ_∞ is optimized to minimize MLIN using a simplex algorithm (Nelder-Mead). During this procedure Θ_∞ is itself not optimized, instead, a multiplicative factor (coefficient) is optimized. To ensure that the optimization routine did not return a sub-optimal local minimum, the overall optimization is rerun three times. Each time the overall optimization is rerun, the coefficients are randomly perturbed between an interval of ± 0.3 of their last found optimized value. Within each of the overall reruns, the stability of the convergence is confirmed by reinitializing the algorithm (i.e. re-inflating the simplex) three times at the optimal position in parameter space with a small random perturbation within an interval of ± 0.01 and then re-running the simplex.

Because the voltage dynamics are governed by a first-order linear differential equation, the voltage dynamics are forward simulated over a single time step using an exact Euler time stepping method. The time step is chosen to be 0.2 ms. The current that forces the dynamics is averaged across the time step.

4.5 Performance of individual transgenic lines

The explained variance of individual transgenic lines are shown in Figures 14 and 13 and 14.

4.6 Clustering

In order to cluster cells by model parameters or extracted electrophysiological features, we used an iterative binary splitting method using standard hierarchical clustering methods, as follows:

1. Select parameters to cluster on (See Table 1 in the main text).

	n	$GLIF_1$	$GLIF_2$	$GLIF_3$	$GLIF_4$	$GLIF_5$
		70	69.7	71.3	74.3	76.2
all neurons	771	[35.7, 95.3] 76.4	[44.4, 94.6] 75.5	[33.8, 95.3] 83.7	[41.6, 97.5] 86.4	[43.3, 97.4] 87
inhibitory	315	[31.4, 97] 67.3	[41.6, 96.4] 67.7	[31.9, 96.7] 65.5	[45.5, 98.1] 70.8	[48, 98.2] 72.6
excitatory	456	[42.1, 81.5] 60.4	[50.1, 78.7]	[34.4, 85.8] 53	[40.7, 88.1]	[42.6, 88.8]
Scnn1a-Tg2	34	[45.3, 77.7] 67.7	NaN 69.5	[33.7, 74.8] 68.7	NaN 71.6	NaN 73.2
Nr5a1	64	[48.8, 81.4] 65.6	[62.5, 77.3] 66.3	[47.3, 82.9] 58.3	[55.5, 88.1] 64.6	[60.9, 88.7] 64.6
Scnn1a-Tg3	47	[45.1, 76.5] 65	[53.9, 80.6] 69.2	[36.5, 84.7] 63.4	[42.6, 88.3] 68.1	[36.5, 87.6] 71
Rorb	108	[40.8, 81] 69.9	[51.1, 78.6] 67.8	[21.6, 84] 70.2	[38, 84.3] 76.1	[43.1, 86.4] 76.9
Cux2	74	[46.9, 81] 71.3	[58.7, 74.6] 65.1	[37.1, 86.6] 79.2	[50.8, 87.6] 79.5	[54, 88] 81.6
Ntsr1	44	[50.3, 82.3] 61.2	[42.8, 81.2]	[42.6, 88.7] 48.2	[53.6, 90]	[53.3, 90.7]
Ctgf	21	[28.9, 73.6] 69.6	NaN 68	[23.5, 75.2] 66	NaN 72.7	NaN 73.8
Rbp4	64	[46.4, 83.9] 75	[55.6, 78.5] 75.5	[34.9, 84] 75.3	[47.5, 85.8] 79.4	[49.6, 89.2] 80.1
Sst	92	[19.1, 95.1] 95	[39.2, 93] 94.9	[15.2, 95] 94.4	[45.2, 95.9] 97.9	[48, 96] 97.9
Pvalb	82	[29.9, 97.8] 73.2	[33.9, 97.8] 69	[29.5, 97.6] 84.8	[38.5, 98.8] 86.8	[40.6, 98.8] 87.1
Htr3a	96	[35.4, 86.1] 80.5	[47.6, 87.3]	[41.9, 93.4] 85.3	[47.5, 95.1]	[53.6, 95.3]
Ndnf	13	[67.3, 89.2]	NaN	[69.2, 90.3]	NaN	NaN
Chat	7	NaN	NaN	NaN	NaN	NaN
		67.5		72		
Vip	25	[34.6, 86.8]	NaN	[36.2, 89.7]	NaN	NaN

Table 4: **Explained variance of all model levels after post-hoc optimization.** The single number in each cell is the median. The low and high quartiles are in brackets below the median. All Cre lines with ten or more neurons (n) present in the LIF level are included. Some neurons have the required stimuli for LIF models but do not have the stimuli required for higher level models. When there are not more than 7 neurons in the level, the values are denoted with a NaN. Data illustrated in Figures(5, 13, and 14)

Specimen ID	1	2	3	4	5
Pvalb					
475545311	0.943	0.928	0.935	0.961	0.965
515305346	0.945	0.978	0.944	0.982	0.984
481127173	0.974	0.963	0.960	0.979	0.980
484742372	0.971	0.946	0.967	0.983	0.982
478949560	0.922	0.940	0.927	0.942	0.943
Rorb					
473020156	0.604	0.717	0.725	0.742	0.721
478590454	0.437	0.469	0.339	0.324	0.275
480353286	0.554	0.604	0.537	0.613	0.583
502887600	0.609	0.550	0.434	0.450	0.442
476595373	0.673	0.691	0.720	0.747	0.745
480171386	0.600	0.743	0.634	0.582	0.522
485837504	0.661	0.628	0.562	0.630	0.568
329549957	0.627	0.644	0.470	0.531	0.561
500857324	0.548	0.570	0.356	0.555	0.584
324065524	0.750	0.798	0.684	0.781	0.836
478498617	0.540	0.655	0.598	0.654	0.655
502574847	0.726	0.717	0.831	0.828	0.828
313861677	0.640	0.649	0.443	0.461	0.491
501951290	0.605	0.673	0.433	0.548	0.559
480798873	0.700	0.663	0.702	0.725	0.730
Cux2					
487358945	0.703	0.644	0.592	0.604	0.661
486239338	0.536	0.731	0.462	0.663	0.613
486198953	0.692	0.645	0.619	0.764	0.765
490126119	0.807	0.775	0.833	0.864	0.859
486108147	0.669	0.718	0.604	0.691	0.697
Ntsr1					
488426904	0.751	0.679	0.847	0.906	0.902
485339983	0.728	0.681	0.685	0.742	0.795
485905411	0.608	0.567	0.795	0.783	0.777
471688285	0.723	0.413	0.627	0.806	0.804
503814817	0.689	0.590	0.846	0.869	0.885
490263438	0.840	0.723	0.842	0.887	0.910
Scnn1a-Tg3					
476557750	0.669	0.663	0.639	0.732	0.735
486893033	0.610	0.618	0.480	0.583	0.690
476263004	0.727	0.643	0.885	0.854	0.835
476127518	0.614	0.689	0.592	0.743	0.746
476269122	0.602	0.626	0.531	0.650	0.646
476271271	0.547	0.613	0.583	0.579	0.566
Nr5a1					
489731121	0.658	0.695	0.597	0.698	0.705
502267531	0.587	0.621	0.529	0.666	0.654
502245101	0.732	0.740	0.706	0.737	0.699
489750097	0.683	0.718	0.705	0.727	0.734
476305247	0.691	0.693	0.653	0.716	0.732
502193385	0.680	0.687	0.630	0.644	0.632
321893798	0.509	0.626	0.400	0.453	0.496
318733871	0.570	0.704	0.472	0.711	0.732
Htr3a					
482654108	0.759	0.664	0.916	0.916	0.917
488686369	0.712	0.691	0.864	0.864	0.871
475622793	0.694	0.656	0.917	0.905	0.919
482438913	0.758	0.724	0.885	0.907	0.870
464188580	0.667	0.600	0.863	0.849	0.869
487176969	0.851	0.871	0.885	0.887	0.890
519517188	0.731	0.653	0.852	0.827	0.832
482516216	0.717	0.739	0.837	0.872	0.844
475586130	0.678	0.663	0.790	0.783	0.769
Sst					
322197295	0.420	0.397	0.406	0.483	0.491
485248699	0.803	0.822	0.848	0.851	0.879
324495712	0.433	0.538	0.455	0.517	0.551
476686112	0.694	0.747	0.709	0.684	0.760
328015342	0.747	0.706	0.676	0.748	0.776
502382506	0.789	0.794	0.752	0.805	0.815
322229594	0.671	0.753	0.640	0.682	0.662
504608602	0.409	0.421	0.401	0.393	0.534
328031983	0.891	0.889	0.884	0.869	0.872
478409545	0.823	0.837	0.845	0.858	0.852
475542907	0.753	0.755	0.770	0.786	0.788
475057898	0.863	0.889	0.842	0.885	0.883
Rbp4					
485577686	0.692	0.684	0.618	0.728	0.741
471131311	0.775	0.737	0.688	0.754	0.736
490916919	0.703	0.659	0.720	0.725	0.792
508911693	0.507	0.635	0.708	0.636	0.640
485573974	0.720	0.717	0.725	0.785	0.776
490612844	0.668	0.665	0.710	0.704	0.715
485835016	0.674	0.632	0.624	0.721	0.715
485880739	0.618	0.614	0.663	0.665	0.606
490982660	0.742	0.639	0.832	0.865	0.884

Table 5: **Neurons which have all 5 GLIF models and all model parameters are within 5 and 95 percentiles within each transgenic line.** Columns 1 through 5 denote the explained variance of models 1 through 5. The best explained variance within the set is emboldened.

2. Z-score all parameters and hierarchically cluster cells using Euclidean distance and Ward's method.
3. Split cells into two clusters, as defined by the top branch split of the dendrogram from the hierarchical clustering.
4. Calculate the likelihood associated with the data being derived from two multivariate Gaussians (defined by the two clusters) versus a single multivariate Gaussian. This was calculated using the sigClust package in R.
5. If the assessment of the separation into two clusters (from step 4) is significant (sigClust p-value ≤ 0.001), repeat steps 2-4 on each of the two subpopulations.
6. If the assessment of the separation into two clusters (from step 4) is not significant, assign the given population to a terminal cluster.
7. End clustering once all cells are assigned to a terminal cluster that cannot be split further.

Because the estimation of the separation into two clusters (step 4) has some stochasticity (using the current implementation of the sigClust package), clustering was run 100 times for each set of parameters, and only final clusters that were identified in at least 50 runs were included as part of the final run. Given that the identity of cells segregated into two branches is not stochastic (as opposed to the evaluation of the branch separation), all clustering runs yield final trees that only differ in their depth, not in their segregation across branches. As a result, the final set of clusters chosen for a given set of parameters can be coherently assembled into a single tree, with each leaf and intermediate node being assigned a percentage based on the number of runs in which it appeared, as illustrated in the corresponding figure.

4.7 Exclusion Criteria

Here, since we are interested in the models of transgenic lines, we include only cre positive cells and neurons from transgenic lines that have more than 7 modelable neurons. After parameters were fit as described in section "3 Parameter fitting and distributions", models containing outlier parameters or parameters that are biologically unrealistic as listed below are excluded from the data set.

1. Neurons were eliminated because the training subthreshold noise epochs (used for fitting resistance via linear regression) contained a spike.
2. Neurons with an intercept greater than 30 mV resulting from the linear regression performed during the spike cut width calculation (Figure 8).
3. Neurons with an instantaneous threshold, Θ_{∞} , less than -60 mV before post-hoc optimization (Figure 7) are eliminated.
4. Neurons are eliminated because they contain a resistance greater than 1000 $M\Omega$ (Figure 9).
5. Neurons that fail to complete the post-hoc optimization without error.

Additional neurons were eliminated from models containing reset rules based on the following criteria:

1. Models had an amplitude of the spike component of threshold, δ_s that was larger than 20 mV (Figure 10).
2. Models had an amplitude of the spike component of threshold, δ_s that was less than zero (Figure 10)
3. Models had a $1/b_s$ that was greater than 100 mS^{-1} (Figure 10).

Finally models with the following voltage dependent threshold are excluded:

1. Models that have a_v less than -50 s^{-1} (Figure 10).

2. Models that have a b_v is less than 0.1 s^{-1} (Figure 10).

This leaves a total of 771 neurons that have a $GLIF_1$, and $GLIF_3$ model, 289 neurons that have a $GLIF_2$ and a $GLIF_4$ model, and 288 neurons that have a $GLIF_5$ model.

4.8 Using linear regression to solve for variables using a noisy stimulus

One way to fit the resistance, R , capacitance, C , and resting potential, El is to use linear regression on the voltage, V , response to a noisy stimulus, I . The standard leaky integrate and fire model for the dynamics of a neuron is:

$$C \frac{dV}{dt} = -\frac{(V - El)}{R} + I \quad (29)$$

This equation can be rearranged so that it fits into the form $y = X\beta + \epsilon$ where y is the dependent variable, X is a the independent variable and β are the parameters being fit. For implementation this means that the parameters being fit should be associated with the matrix on the (rhs) of the equation, *lhs* and the known parameters should be on the left hand side (lhs) of the equation. After the equation has been arranged in the proper form, a standard least squares algorithm such as `np.linalg.lstsq` in python can be used.

For example let us say we want to fit El and R in Equation 29. Equation 29 can be rearranges as follows:

$$\begin{aligned} \frac{dV}{dt} &= \frac{-V + El}{RC} + \frac{I}{C} \\ \frac{\Delta V}{\Delta t} &= \frac{-V}{RC} + \frac{El}{RC} + \frac{I}{C} \\ V_{t+1} - V_t &= \frac{-V_t \Delta t}{RC} + \frac{El \Delta t}{RC} + \frac{I_t \Delta t}{C} \\ V_{t+1} &= \frac{-V_t \Delta t}{RC} + V_t + \frac{El \Delta t}{RC} + \frac{I_t \Delta t}{C} \\ &= V_t \left(1 - \frac{\Delta t}{RC}\right) + \frac{El \Delta t}{RC} + \frac{I_t \Delta t}{C} \end{aligned} \quad (30)$$

Here, Δt is the amount of time between V and V_{t+1} (time step) where t is the timestep index which runs from $t = 1$ to $t = T$ where T is the total number of time steps in the data.

For implementation, each term must be associated with a vector. The vector may be output data or input stimulus (for example, here V , V_{t+1} , or I) or a vector of ones when a term is not associated with a time series. In vector form, Equation 30 is:

$$\begin{bmatrix} V_2 \\ V_3 \\ \cdot \\ \cdot \\ V_T \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ V_{T-1} \end{bmatrix} \frac{-\Delta t + 1}{RC} + \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \frac{El \Delta t}{RC} + \begin{bmatrix} I_1 \\ I_2 \\ \cdot \\ \cdot \\ I_{T-1} \end{bmatrix} \frac{\Delta t}{C}$$

Fitting R , C and El results in the following equation:

$$\begin{bmatrix} V_2 \\ V_3 \\ \cdot \\ \cdot \\ V_N \end{bmatrix} = \begin{bmatrix} V_1 & 1 & I_1 \\ V_2 & 1 & I_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ V_{N-1} & 1 & I_{N-1} \end{bmatrix} \begin{bmatrix} \frac{-\Delta t + 1}{RC} \\ \frac{El \Delta t}{RC} \\ \frac{\Delta t}{C} \end{bmatrix}$$

The output will be an vector where

$$\begin{aligned} out[1] &= \frac{-\Delta t + 1}{RC} \\ out[2] &= \frac{El \Delta t}{RC} \\ out[3] &= \frac{\Delta t}{C} \end{aligned} \quad (31)$$

Solving this system of equations yields:

$$C = \frac{\Delta t}{out[3]} \quad (32)$$

$$R = \frac{-\Delta t}{C(out[1] - 1)} = \frac{-\Delta t}{\frac{\Delta t}{out[3]}(out[1] - 1)} = \frac{-out[3]}{(out[1] - 1)} \quad (33)$$

$$El = out[2] \left(\frac{RC}{\Delta t} \right) = out[2] \left(\frac{R \frac{\Delta t}{out[3]}}{\Delta t} \right) = out[2] \left(\frac{R}{out[3]} \right) \quad (34)$$

Now let's say that we do not want to fit the capacitance during the regression. In this case the term containing the I vector will be shifted to the left hand side of the equation:

$$\begin{bmatrix} V_2 \\ V_3 \\ \vdots \\ V_T \end{bmatrix} - \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_{T-1} \end{bmatrix} \frac{\Delta t}{C} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_{T-1} \end{bmatrix} \frac{-\Delta t + 1}{RC} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \frac{El \Delta t}{RC}$$

In matrix form:

$$\begin{bmatrix} V_2 - I_1 \\ V_3 - I_2 \\ \vdots \\ V_T - I_{T-1} \end{bmatrix} \frac{\Delta t}{C} = \begin{bmatrix} V_1 & 1 \\ V_2 & 1 \\ \vdots & \vdots \\ V_{T-1} & 1 \end{bmatrix} \begin{bmatrix} \frac{-\Delta t + 1}{RC} \\ \frac{El \Delta t}{RC} \end{bmatrix}$$

Solving this system of equations yields:

$$R = \frac{\Delta t}{C(out[1] - 1)} \quad (35)$$

$$El = \left(\frac{RC}{\Delta t} \right) out[2] \quad (36)$$

Above, we solved this problem by separating ΔV into the voltages before and after the time step Δt , i.e. $V_{n+1} - V$. This problem can also be solved by calculating $\frac{\Delta V}{\Delta t}$ directly from the data.

4.9 Analytical solution for the dynamics of the voltage component of the threshold

The voltage component of the threshold evolves according to the following dynamics:

$$\frac{d\theta}{dt} = a(V(t) - El) - b(\theta - \theta_\infty) \quad (37)$$

Where θ is the voltage component of the threshold, V is the voltage of the neuron which changes in time t , El is the resting potential, θ_∞ , and a and b are constants which will be fit to the data. Below we will find the analytical solution to equation (37).

The solution to a differential equation with the form:

$$\dot{x} + p(t)x = q(t) \quad (38)$$

is

$$x(t) = \frac{1}{u(t)} \int u(t)q(t)dt + K \quad (39)$$

where K is the standard integration constant and

$$u(t) = e^{\int p(t)dt} \quad (40)$$

Equation (37) can be rearranged into this form:

$$\frac{d\theta}{dt} + b\theta = a(V(t) - El) + b\theta_\infty \quad (41)$$

Where

$$\begin{aligned} u(t) &= e^{\int bdt} = e^{bt} \\ q(t) &= a(V(t) - El) + b\theta_\infty \end{aligned} \quad (42)$$

Therefore:

$$\begin{aligned} \theta(t) &= \frac{1}{e^{bt}} \int e^{bt}[a(V(t) - El) + b\theta_\infty]dt + K \\ &= \frac{1}{e^{bt}} \int [ae^{bt}V(t) - ae^{bt}El + b\theta_\infty e^{bt}]dt + K \end{aligned} \quad (43)$$

The analytical solution for $V(t)$

$$V(t) = V_0 e^{-\frac{tg}{c}} + \left(\frac{I + gEl}{g} \right) (1 - e^{-tg/c}) \quad (44)$$

Plugging in $V(t)$ for $\theta(t)$ and defining $\beta = \frac{I + gEl}{g}$ gives:

$$\begin{aligned} \theta(t) &= \frac{1}{e^{bt}} \int [ae^{bt}(V_0 e^{-\frac{tg}{c}} + \beta(1 - e^{-tg/c})) - ae^{bt}El + b\theta_\infty e^{bt}]dt + K \\ &= \frac{1}{e^{bt}} \int [ae^{bt}V_0 e^{-\frac{tg}{c}} + ae^{bt}\beta(1 - e^{-tg/c}) - ae^{bt}El + b\theta_\infty e^{bt}]dt + K \\ &= \frac{1}{e^{bt}} \int [aV_0 e^{(b-\frac{g}{c})t} + a\beta e^{bt} - ae^{(b-\frac{g}{c})t} - ae^{bt}El + b\theta_\infty e^{bt}]dt + K \end{aligned} \quad (45)$$

Setting $\alpha = (b - \frac{g}{c})$

$$\begin{aligned} \theta(t) &= \frac{1}{e^{bt}} \left[\frac{aV_0}{\alpha} e^{\alpha t} + \frac{a\beta}{b} e^{bt} - \frac{a\beta}{\alpha} e^{\alpha t} - \frac{aEl}{b} e^{bt} + \frac{b\theta_\infty}{b} e^{bt} + K \right] \\ &= \frac{aV_0}{\alpha} e^{-\frac{g}{c}t} + \frac{a\beta}{b} - \frac{a\beta}{\alpha} e^{-\frac{g}{c}t} - \frac{aEl}{b} + \theta_\infty + \frac{K}{e^{bt}} \\ &= \left[\frac{aV_0}{\alpha} - \frac{a\beta}{\alpha} \right] e^{-\frac{g}{c}t} + \frac{a\beta}{b} - \frac{aEl}{b} + \theta_\infty + \frac{K}{e^{bt}} \\ &= \frac{a}{\alpha} (V_0 - \beta) e^{-\frac{g}{c}t} + \frac{K}{e^{bt}} + \frac{a}{b} (\beta - El) + \theta_\infty \end{aligned} \quad (46)$$

at $t = 0$, $\theta(t) = \theta_0$. Therefore:

$$\theta_0 = \frac{a}{\alpha}(V_0 - \beta) + K + \frac{a}{b}(\beta - El) + \theta_\infty \quad (47)$$

$$K = \theta_0 - \frac{a}{\alpha}(V_0 - \beta) - \frac{a}{b}(\beta - El) - \theta_\infty \quad (48)$$

Plugging in K :

$$\begin{aligned} \theta(t) &= \frac{a}{\alpha}(V_0 - \beta)e^{-\frac{g}{c}t} + \frac{1}{e^{bt}} \left[\theta_0 - \frac{a}{\alpha}(V_0 - \beta) - \frac{a}{b}(\beta - El) - \theta_\infty \right] + \frac{a}{b}(\beta - El) + \theta_\infty \quad (\text{or}) \\ &= \frac{a}{b - \frac{g}{c}} \left(V_0 - \frac{I + gEl}{g} \right) e^{-\frac{g}{c}t} \\ &\quad + \frac{1}{e^{bt}} \left[\theta_0 - \frac{a}{b - \frac{g}{c}} \left(V_0 - \frac{I + gEl}{g} \right) - \frac{a}{b} \left(\frac{I + gEl}{g} - El \right) - \theta_\infty \right] \\ &\quad + \frac{a}{b} \left(\frac{I + gEl}{g} - El \right) + \theta_\infty \quad (49) \end{aligned}$$

	R (M Ω)	τ (ms)	C (pF)	EI (mV)	th $_{\infty}$ (mV)	Δ th $_{\infty}$ (mV)	R $_{ASC}$ (M Ω)	Q $_1$ (pC)
inhibitory	193	11.4	59.7	-72.4	-45.4	26.6	201	-1.28
	[135, 278]	[7.98, 17.9]	[50.6, 71.9]	[-75.3, -69.2]	[-47.6, -42.8]	[23.8, 30.2]	[140, 309]	[-2.32, -0.742]
excitatory	180	18.7	106	-75.3	-47	27.6	191	-2.84
	[136, 244]	[14.7, 27]	[88.6, 129]	[-78.1, -71.8]	[-48.9, -45.1]	[24.9, 31]	[138, 280]	[-3.96, -2.04]
Senn1a-Tg2	179	27.7	129	-74.4	-46	26.8	215	-3.36
	[126, 232]	[16.9, 31.7]	[112, 166]	[-76.4, -71.9]	[-49.5, -44.7]	[25.5, 29.8]	[126, 321]	[-4.18, -2.13]
Nr5a1	188	17.7	94.9	-77	-47.2	30.8	184	-2.65
	[146, 265]	[13.8, 24.5]	[83.3, 107]	[-80.2, -73.3]	[-49.3, -45.4]	[27.4, 32.1]	[129, 265]	[-3.54, -2.21]
Senn1a-Tg3	189	20.9	104	-75.7	-46.4	28.6	212	-2.54
	[154, 246]	[14.4, 28.2]	[86.6, 131]	[-78, -72]	[-48.3, -44.6]	[25, 32.8]	[146, 294]	[-3.58, -1.68]
Rorb	192	19.1	98.3	-75.2	-46.8	28	197	-2.78
	[143, 262]	[15.4, 29.1]	[86.9, 123]	[-78.3, -71.3]	[-49.1, -44.9]	[25.2, 30.8]	[150, 299]	[-3.76, -1.99]
Cux2	166	16.8	105	-75.3	-47.2	27.5	175	-2.69
	[136, 221]	[14, 21.5]	[88.5, 118]	[-78.1, -72.7]	[-48.4, -45.5]	[26, 30.9]	[137, 223]	[-3.94, -1.91]
Ntsr1	186	16.7	91.2	-75.4	-47.3	28.1	192	-2.9
	[153, 227]	[15, 19.3]	[80.5, 104]	[-77.7, -72.3]	[-48.3, -45.4]	[25.6, 30.5]	[146, 232]	[-4.75, -2.13]
Ctgf	245	25.5	119	-73.7	-46.3	24.5	275	-2.36
	[160, 345]	[20.5, 37]	[97, 135]	[-75.4, -67.6]	[-48.5, -44.7]	[19.2, 28.2]	[183, 405]	[-3.13, -1.83]
Rbp4	137	18.3	135	-73.8	-47.8	24.9	162	-3.18
	[104, 195]	[14, 26]	[111, 150]	[-77.8, -68.8]	[-49.4, -45.9]	[22.2, 28.9]	[113, 253]	[-4.54, -2.22]
Sst	234	16.7	70	-71.3	-45.8	25.3	242	-1.36
	[171, 319]	[11.2, 22.1]	[57.7, 84.2]	[-73.9, -68.4]	[-47.9, -43.4]	[22.3, 28.8]	[168, 340]	[-1.78, -0.858]
Pvalb	137	7.55	54.4	-74.6	-46.2	27.6	143	-0.734
	[110, 179]	[6.52, 9.42]	[49.5, 65.3]	[-76.4, -71.5]	[-48, -43.6]	[24.9, 30.5]	[109, 180]	[-1.24, -0.507]
Htr3a	195	11.1	55.4	-72.1	-44	26.9	207	-2.07
	[136, 335]	[7.92, 16.4]	[45.2, 66.2]	[-75, -67.7]	[-45.9, -41]	[24, 30.8]	[148, 345]	[-3.84, -1.02]
Ndnf	189	11.9	62.3	-72.7	-42.4	29.7	203	-2.32
	[140, 215]	[11.5, 13]	[59.4, 74.5]	[-75.3, -68.3]	[-42.8, -40.6]	[26.4, 35.1]	[154, 234]	[-2.56, -1.93]
Chat	223	14.9	55.3	-74.1	-47.3	26.6	226	-1.25
	[181, 325]	[11, 20]	[50.9, 64.3]	[-76.8, -73]	[-47.7, -46.7]	[25.6, 28.8]	[189, 338]	[-1.85, -0.895]
Vip	324	20.3	57.9	-72.2	-46.3	25.1	342	-1.04
	[260, 422]	[14.2, 23.2]	[51.5, 66.2]	[-74.6, -69.8]	[-48, -44.7]	[22.7, 27.4]	[277, 492]	[-2.28, -0.807]

Table 6: **Summary characterization of GLIF parameters.** All Cre lines with 7 or more LIF neurons are included. The single number in each cell is the median. The low and high quartiles are specified in brackets below the median. Some neurons have the required stimuli for LIF models but do not have the stimuli required for higher level models. When there are not more than 7 neurons that have the parameter, these values are denoted with a NaN.

	Q_2 (pC)	δt (ms)	slope	intercept (mV)	$a_{s,pike}$ (V)	$1/(b_{spike})$ (s)	$(a_w)/(b_w)$	$log_{10}(b_w)$
inhibitory	0.18 [-0.569, 0.82] -2.04	2.2 [1.7, 3.12] 4.53	0.4 [0.187, 0.591] 0.464	5.89 [-0.822, 11.3] 9.19	0.00238 [0.00173, 0.00341] 0.00266	0.022 [0.00836, 0.0363] 0.00414	0.226 [0.141, 0.293] 0.307	1.85 [1.42, 2.17] 1.84
excitatory	[-2.68, -1.55] -2.27	[3.6, 5.85] 6.23	[0.264, 0.709] 0.467	[1.45, 16.2] 10	[0.00163, 0.00393] 0.000861	[0.00203, 0.00738] 0.00226	[0.233, 0.381] 0.357	[1.64, 2.02] 2.03
Senn1a-Tg2	[-3.62, -1.6] -1.99	[3.98, 7.78] 3.68	[0.354, 0.623] 0.238	[3.03, 13.8] 20.4	[0.000432, 0.00151] 0.00346	[0.00121, 0.00297] 0.00159	[0.353, 0.398] 0.295	[1.88, 2.16] 1.9
Nr5a1	[-2.36, -1.69] -2.05	[2.93, 4.45] 4.8	[0.0784, 0.408] 0.375	[12, 24.3] 12.4	[0.00267, 0.00678] 0.00222	[0.00124, 0.00416] 0.00335	[0.23, 0.37] 0.352	[1.78, 2.05] 1.88
Senn1a-Tg3	[-2.45, -1.75] -1.97	[3.95, 6.22] 4.6	[0.216, 0.587] 0.39	[3.08, 19.3] 10.3	[0.00156, 0.00367] 0.00256	[0.0019, 0.00942] 0.00355	[0.215, 0.385] 0.27	[1.74, 2.07] 1.82
Rorb	[-2.37, -1.57] -1.98	[3.74, 6.08] 4.53	[0.252, 0.57] 0.41	[4.7, 16.2] 10.9	[0.00132, 0.00333] 0.00267	[0.00238, 0.00762] 0.00445	[0.196, 0.347] 0.287	[1.62, 2.02] 1.81
Cux2	[-2.6, -1.46] -2.03	[3.56, 5.65] 3.85	[0.238, 0.596] 0.781	[6.08, 15.6] -1.67	[0.00166, 0.00311] 0.00434	[0.00178, 0.00597] 0.0036	[0.226, 0.343] 0.301	[1.75, 1.98] 1.84
Ntsr1	[-2.73, -1.16] -2.08	[3.04, 5.48] 8.75	[0.618, 0.971] 1.28	[-8.82, 5.75] -12.2	[0.00281, 0.00632] 0.00231	[0.00263, 0.00499] 0.0114	[0.24, 0.362] 0.24	[1.76, 1.94] 1.41
Ctgf	[-2.71, -0.895] -2.44	[6.5, 9.9] 4.55	[0.957, 1.66] 0.681	[-22.8, -3.79] 4.21	[0.00192, 0.00326] 0.00282	[0.00855, 0.0454] 0.00659	[0.226, 0.318] 0.367	[1.03, 1.48] 1.77
Rbp4	[-3.1, -1.37] 0.933	[3.75, 5.44] 2.08	[0.456, 0.832] 0.488	[-1.6, 10.9] 0.106	[0.00209, 0.0042] 0.00195	[0.00396, 0.00837] 0.0354	[0.313, 0.426] 0.231	[1.61, 1.97] 1.43
Sst	[0.422, 1.58] 0.277	[1.74, 2.65] 1.65	[0.312, 0.655] 0.153	[-5.03, 4.63] 10.7	[0.00133, 0.00263] 0.00342	[0.0228, 0.0482] 0.00378	[0.158, 0.306] 0.21	[1.31, 1.84] 2.19
Pvalb	[-0.341, 0.638] -0.597	[1.45, 1.8] 2.9	[0.0449, 0.263] 0.445	[6.54, 14.4] 6.28	[0.00208, 0.00488] 0.00239	[0.00288, 0.00642] 0.0236	[0.147, 0.265] 0.226	[1.99, 2.28] 1.99
Htr3a	[-1.67, 0.226] -1.38	[2.39, 3.33] 4.7	[0.31, 0.632] 0.253	[-0.745, 11.6] 14	[0.00188, 0.00303] 0.00271	[0.0122, 0.0303] 0.0145	[0.15, 0.294] 0.108	[1.52, 2.22] 2.11
Ndnf	[-1.54, -0.965] -0.42	[4.15, 5.1] 4.1	[0.0405, 0.512] 0.568	[7.2, 19.6] 4.92	[0.00177, 0.00284] NaN	[0.00979, 0.0203] NaN	[0.0966, 0.12] NaN	[1.6, 2.18] NaN
Chat	[-0.594, 0.537] -0.223	[3.5, 4.15] 3.75	[0.518, 0.671] 0.485	[4.28, 5.62] 3.91	NaN 0.00414	NaN 0.0354	NaN 0.265	NaN 1.55
Vip	[-0.551, 0.632]	[3.1, 4.55]	[0.379, 0.731]	[0.867, 7.77]	[0.00372, 0.00464]	[0.0324, 0.0461]	[0.234, 0.3]	[1.51, 1.96]

Table 7: **SECOND HALF OF TABLE 6**

	R (M Ω)	τ (ms)	C (pF)	El (mV)	th $_{\infty}$ (mV)	Δ th $_{\infty}$ (mV)	R $_{ASC}$ (M Ω)	Q $_1$ (pC)
1	291 [233, 332] 140	31.3 [27.4, 37.2] 20.6	111 [93.7, 134] 158	-75.5 [-77.3, -72.5] -74.1	-46 [-48, -44.5] -48.1	29 [26.4, 32.2] 25	359 [288, 413] 164	-2.28 [-3.21, -1.66] -3.81
2	[107, 183] 91.5	[18.2, 29] 14	[147, 181] 150	[-75.6, -70.8] -76.1	[-49.6, -46.8] -47.9	[22.9, 28.1] 27	[130, 239] 85.9	[-4.74, -2.92] -5.08
3	[76.9, 115] 150	[11.1, 17.8] 12.7	[134, 169] 87.2	[-78.1, -72] -75.8	[-49.7, -46.2] -47.4	[24.5, 30.8] 28.9	[74.8, 110] 136	[-5.9, -4.42] -5.54
4	[123, 174] 253	[9.54, 15.4] 19.4	[59.1, 101] 76	[-77.7, -73.4] -76.5	[-48.7, -44.5] -45	[25.9, 30.6] 31.8	[115, 159] 263	[-7.95, -3.42] -2.92
5	[227, 282] 172	[16.1, 22.6] 12.4	[64, 90.5] 75.9	[-80.6, -73.5] -71.7	[-47, -43.5] -45.9	[30.2, 34.5] 26	[208, 284] 173	[-4.13, -2.5] -2.04
6	[145, 205] 146	[9.77, 16.8] 13	[60.6, 89.4] 90.4	[-75.1, -69.8] -79.5	[-47.7, -43.9] -48.2	[24.1, 29.8] 31.3	[142, 215] 139	[-2.58, -1.51] -2.82
7	[122, 169] 161	[10.2, 16.5] 17.8	[82.1, 99.5] 111	[-80.5, -77.5] -75.7	[-50.9, -45] -47.5	[28.4, 34.3] 28.6	[116, 154] 157	[-3.62, -2.37] -3.23
8	[142, 205] 360	[15.2, 21.5] 33.5	[105, 121] 98.8	[-78.4, -74.2] -66.1	[-49.1, -46.3] -46.1	[26.5, 30.4] 19	[136, 200] 505	[-3.91, -2.57] -2.02
9	[294, 392] 167	[26.3, 45] 23.4	[86.2, 115] 141	[-67.6, -61.7] -73.9	[-47.2, -44.5] -46.6	[17.6, 21.1] 26.2	[366, 546] 207	[-3.22, -1.74] -2.56
10	[115, 246] 186	[15.1, 29.4] 24.8	[128, 153] 113	[-78.1, -71.2] -68.7	[-48.6, -43] -45.9	[24.7, 30.4] 23.1	[140, 320] 261	[-2.98, -1.69] -2.22
11	[165, 250] 177	[18.5, 26.3] 14.4	[101, 129] 76.4	[-70.3, -66.3] -71.8	[-48.2, -44.2] -46.1	[17.7, 26] 25.6	[212, 301] 181	[-2.78, -1.82] -2.88
12	[153, 203] 245	[10.5, 17.1] 14.3	[61.2, 101] 61.1	[-74.1, -68] -67	[-48, -43.6] -41	[23.4, 27.9] 27.4	[162, 205] 260	[-4.05, -2.61] -1.97
13	[201, 337] 166	[10.4, 22.9] 14.5	[43.2, 85.1] 82.6	[-74.4, -65.6] -75.8	[-42.5, -37.7] -45.1	[24.7, 34.5] 30	[219, 313] 169	[-2.84, -1.44] -3.02
14	[129, 228] 148	[10.2, 18.1] 8.8	[68.3, 91.3] 62.1	[-77.7, -73.3] -73.7	[-48, -41.6] -46.7	[26.9, 34.7] 27	[138, 230] 161	[-3.74, -2.29] -0.76
15	[113, 219] 230	[6.63, 11.4] 12.1	[51.5, 71] 53.7	[-76.1, -71.3] -73.9	[-48.8, -44.6] -46.1	[24.2, 29.5] 27.6	[119, 219] 217	[-1.26, -0.508] -1.23
16	[169, 334] 434	[8.09, 18.1] 23.4	[47.9, 60.6] 53.5	[-76.8, -72] -68.7	[-47.9, -43.3] -45.5	[25.3, 31] 23.4	[147, 348] 534	[-1.7, -0.757] -0.89
17	[339, 498] 216	[17.4, 33.3] 18.4	[46.3, 72.7] 81.2	[-71.7, -65.7] -70.5	[-46.6, -44.2] -45.5	[21.1, 26.3] 25.4	[386, 649] 236	[-1.42, -0.676] -1.54
18	[172, 313]	[14.4, 24.1]	[66.8, 96]	[-73, -67.9]	[-47, -43.3]	[21.8, 28]	[183, 330]	[-2.03, -1.09]

Table 8: **Summary characterization of GLIF parameters within clusters.** The single number in each cell is the median. The low and high quartiles are demarked in brackets below the median. Some neurons have the required stimuli for LIF models but do not have the stimuli required for higher level models. When there are less than 7 neurons that have the parameter, these values are denoted with a NaN.

	Q_2 (pC)	δt (ms)	slope	intercept (mV)	a_{spike} (V)	$1/(b_{spike})$ (s)	$(a_v)/(b_v)$	$\log_{10}(b_v)$
1	-1.84 [-2.32, -1.55]	6.5 [4.72, 8.23]	0.403 [0.299, 0.575]	9.72 [5.45, 14.6]	0.00153 [0.000959, 0.00219]	0.00387 [0.00205, 0.00557]	0.284 [0.184, 0.364]	1.87 [1.77, 2.03]
2	-2.79 [-3.4, -1.67]	5.85 [4.6, 7.4]	0.723 [0.594, 0.853]	2.35 [-2.01, 5.3]	0.00151 [0.0012, 0.00338]	0.00464 [0.0016, 0.00873]	0.357 [0.335, 0.413]	1.67 [1.52, 2.02]
3	-3.01 [-4.04, -2.45]	4.72 [4.25, 5.59]	0.454 [0.372, 0.571]	12.2 [9.17, 15.1]	0.00286 [0.00224, 0.00364]	0.00359 [0.0019, 0.00608]	0.374 [0.316, 0.412]	1.97 [1.87, 2.02]
4	-0.874 [-3.02, 0.756]	2.98 [1.95, 3.61]	0.576 [0.433, 0.703]	5.74 [0.209, 8.46]	NaN	NaN	NaN	NaN
5	-1.28 [-1.85, -1.04]	3.5 [2.85, 4.14]	0.209 [0.123, 0.321]	20.5 [17.9, 25.1]	NaN	NaN	NaN	NaN
6	-1.57 [-1.92, -1.23]	3.5 [2.8, 3.95]	0.349 [0.225, 0.436]	10.8 [8.78, 15.3]	0.00304 [0.00189, 0.00465]	0.00538 [0.00235, 0.019]	0.227 [0.128, 0.29]	1.93 [1.61, 2.22]
7	-1.99 [-2.28, -1.73]	3.65 [3.18, 4.35]	0.173 [0.0784, 0.257]	23.1 [19.5, 25.3]	0.00295 [0.00222, 0.00637]	0.00318 [0.0013, 0.00683]	0.267 [0.163, 0.356]	2.09 [1.79, 2.18]
8	-2.3 [-2.75, -2.01]	5.15 [4.65, 6.24]	0.393 [0.274, 0.51]	11.9 [8.57, 16.7]	0.00266 [0.00203, 0.00331]	0.00475 [0.00262, 0.00604]	0.31 [0.273, 0.346]	1.82 [1.77, 2.02]
9	-1.35 [-1.67, -0.788]	9.93 [9.71, 9.95]	1.32 [0.996, 1.66]	-16.3 [-24.9, -7.4]	NaN	NaN	NaN	NaN
10	-3.25 [-3.95, -2.65]	6.4 [5.35, 7.78]	1.29 [1.08, 1.47]	-14 [-21.9, -10.7]	0.00271 [0.00187, 0.00341]	0.00635 [0.0056, 0.0072]	0.325 [0.252, 0.401]	1.48 [1.43, 1.61]
11	-1.81 [-2.66, -0.596]	4.2 [3.73, 4.53]	0.503 [0.284, 0.678]	3.07 [1.58, 11]	0.00274 [0.00189, 0.00346]	0.0097 [0.00553, 0.0112]	0.295 [0.235, 0.43]	1.97 [1.83, 2.09]
12	-0.727 [-2.37, -0.311]	3 [2.55, 3.61]	0.572 [0.347, 0.687]	4.22 [-1.44, 10.1]	0.00347 [0.00238, 0.00414]	0.0296 [0.00849, 0.0566]	0.356 [0.317, 0.425]	1.72 [1.55, 2.02]
13	-1 [-1.54, -0.497]	3.58 [3.08, 4.01]	1.19 [1.11, 1.31]	-17.7 [-20, -12.6]	0.00211 [0.00185, 0.00283]	0.0252 [0.0146, 0.0283]	0.158 [0.121, 0.384]	1.36 [1.19, 2.06]
14	-1.91 [-2.37, -1.26]	3.85 [3.28, 5]	0.853 [0.73, 1.06]	-2.88 [-9.83, 1.14]	0.00338 [0.00196, 0.00468]	0.00486 [0.00294, 0.00587]	0.294 [0.2, 0.378]	1.86 [1.67, 1.94]
15	0.304 [-0.334, 0.685]	1.7 [1.48, 1.9]	0.131 [0.0415, 0.224]	11.3 [8.38, 14.5]	0.00214 [0.00133, 0.00422]	0.007 [0.00353, 0.0128]	0.225 [0.15, 0.257]	2.15 [1.81, 2.26]
16	0.0671 [-0.368, 0.482]	2.15 [1.8, 3.03]	0.483 [0.314, 0.649]	4.31 [0.384, 7.76]	0.00224 [0.00134, 0.00405]	0.0289 [0.00878, 0.037]	0.255 [0.137, 0.298]	1.85 [1.47, 2.09]
17	-0.118 [-0.662, 0.79]	3.35 [2.39, 4.11]	0.399 [0.244, 0.52]	2.74 [-1.74, 8.07]	0.00223 [0.00192, 0.00319]	0.0279 [0.0186, 0.0481]	0.162 [0.0868, 0.268]	1.41 [1.12, 1.59]
18	1.53 [1.05, 2.01]	2.1 [1.77, 2.48]	0.603 [0.497, 0.688]	-2.82 [-6.58, 0.496]	0.00271 [0.00197, 0.00305]	0.0348 [0.0234, 0.0443]	0.261 [0.208, 0.394]	1.41 [1.32, 1.62]

Table 9: SECOND HALF OF TABLE 8

	τ_m (ms)	R_i (M Ω)	V_{rest} (mV)	$I_{threshold}^\dagger$ (pA)	$V_{threshold}^\dagger$ (mV)	V_{peak}^\dagger (mV)	$V_{fasttrough}^\dagger$ (mV)	V_{trough}^\dagger
1	24.3 [20.3, 28.8]	188 [148, 218]	-75 [-77.2, -72.2]	70 [50, 90]	-36.8 [-39.1, -35.2]	38.5 [33.7, 44.4]	-48 [-50.4, -45.9]	-55.7 [-58.4, -54.1]
2	15.5, 23.6 18.2	103, 144 103	-75.4, -70.9 -76.1	[90, 150] 210	[-41.4, -36] -36.5	32.8, 41.2 36.7	[-49.9, -44.4] -47	[-58.3, -54.1] -53.3
3	15.4, 20.2 12.3	83.6, 127 137	-78, -71.9 -75.8	[160, 270] 245	[-38.4, -34.3] -34.6	33.2, 40.4 26.6	[-48.2, -45.4] -50.2	[-55.3, -50.7] -52.5
4	8.88, 15.3 19.7	[110, 161] 189	[-77.7, -73.1] -76.1	[210, 325] 110	[-39.5, -32] -35.2	[14.6, 32.7] 33.8	[-59.2, -46.9] -48.6	[-59.3, -48.1] -52.9
5	14.4, 22.3 16.9	[170, 212] 175	[-80.5, -73.2] -71.7	[87.5, 122] 90	[-39.9, -31.6] -38.3	[29.6, 41.1] 37.3	[-52.2, -45.9] -50.7	[-55.7, -49.8] -54.4
6	10, 21.1 17.8	[146, 215] 149	[-75.3, -70.2] -79	[70, 140] 150	[-41.2, -36.2] -37	[26.5, 43] 38.9	[-52.6, -48.7] -50.6	[-56.5, -52.8] -54.8
7	13.9, 20.5 20.7	[125, 171] 164	[-80.7, -77.2] -75.6	[110, 190] 110	[-39.8, -35.8] -36.3	[32.2, 40.9] 37.6	[-53.3, -48.5] -47.6	[-56.6, -52.5] -53.9
8	18.4, 23.1 23.2	[151, 190] 213	[-78.3, -74] -65.8	[90, 130] 45	[-38.6, -33.4] -41.2	[35, 39.7] 38.1	[-49.1, -46] -50.6	[-55.6, -51.9] -55.9
9	18.8, 26.7 18.7	[207, 256] 150	[-67.1, -61.4] -74.2	[32.5, 50] 110	[-43.1, -40.3] -39.8	[35, 39.5] 38.7	[-51, -47.9] -46	[-57.7, -55] -54.5
10	14.2, 24.8 18.7	[91.1, 163] 171	[-77.6, -71] -68.4	[90, 190] 70	[-42.8, -35.6] -39.8	[35.8, 41.9] 36.8	[-49, -44.5] -48.8	[-56, -53] -56.2
11	16.4, 25.1 14.2	[132, 213] 148	[-70.1, -66.5] -71.3	[55, 90] 130	[-41.7, -38.2] -38	[33.9, 42.7] 31.2	[-51.9, -46.7] -51.3	[-57.3, -54.1] -54.3
12	10.2, 18.3 20.1	[130, 178] 219	[-74.1, -68.3] -66.9	[110, 150] 85	[-41.4, -34.8] -34.6	[23, 37.1] 26.9	[-53.8, -49.3] -49.1	[-56.1, -52.2] -51.6
13	10.1, 23.6 14.2	[187, 254] 152	[-73.3, -65.7] -75.3	[70, 120] 150	[-38.4, -32.3] -35.8	[17.6, 30.7] 33.3	[-51.5, -46.9] -47.9	[-55.2, -48.8] -51.7
14	11.8, 19.3 7.01	[119, 196] 114	[-77.4, -73] -73.3	[100, 220] 190	[-38.1, -33.2] -39	[26.6, 37.9] 16.5	[-50.6, -46] -60.4	[-54.2, -48.8] -60.6
15	6.1, 10.3 12	[82.7, 149] 180	[-75.9, -70.8] -74	[130, 290] 130	[-42.7, -35.8] -39.1	[12.5, 25] 20.3	[-64.2, -56.6] -56.8	[-64.4, -56.9] -57
16	7.11, 15.1 19.7	[125, 243] 279	[-76.5, -71.9] -67.9	[72.5, 228] 30	[-43.8, -36.1] -43	[13.5, 29.9] 28.8	[-62.1, -52.5] -55.1	[-62.2, -53.7] -57.5
17	14.5, 26.1 19.7	[240, 339] 205	[-72, -65.3] -70.6	[30, 50] 90	[-45.7, -39.1] -42.2	[22.3, 35.3] 28.7	[-60.9, -50.8] -58.4	[-61.5, -54.1] -58.5
18	14.9, 26.7	[170, 249]	[-72.6, -67.1]	[52.5, 110]	[-45.2, -40.1]	[22.5, 34.2]	[-60.5, -55.8]	[-60.6, -56]

Table 10: Data features of clusters found by clustering GLIF parameters: * short square, † long square.

	up:downstroke†	up:downstroke*	sag (mV)	Avg. ISI (ms)	$f-I$ curve slope (spikes $s^{-1} pA^{-1}$)	latency (s)	Adaptation	Max. burst index
1	3.62 [3.13, 3.97] 3.66	3.83 [3.21, 4.17] 3.77	0.0559 [0.0323, 0.099] 0.0892	69.3 [55.9, 83.2] 100	0.198 [0.173, 0.234] 0.146	0.0555 [0.0428, 0.0637] 0.0585	0.022 [0.0162, 0.0342] 0.0349	0.156 [0, 0.215] 0
2	3.8 [3.22, 4.15] 2.28	3.99 [3.19, 4.06] 3.99	0.047 [0.0596, 0.127] 0.047	116 [79.6, 126] 116	0.0696 [0.107, 0.171] 0.0696	0.0578 [0.0485, 0.0736] 0.0578	0.097 [0.0138, 0.0665] 0.097	0 [0, 0.193] 0
3	3.11 [3.56, 4.22] 2.28	3.43 [3.56, 4.22] 2.48	0.0829 [0.0223, 0.0677] 0.0829	75.1 [88.5, 136] 75.1	0.116 [0.0391, 0.109] 0.116	0.0201 [0.0444, 0.0818] 0.0201	0.0255 [0.0373, 0.187] 0.0255	0 [0, 0.0963] 0
4	3.23 [1.37, 2.61] 3.11	3.43 [1.61, 2.81] 3.43	0.0303 [0.0353, 0.118] 0.0303	33.8, 95.4 [33.8, 95.4] 63.3	0.168 [0.0161, 0.168] 0.205	0.0368 [0.01, 0.0368] 0.0408	0.188 [-0.00848, 0.188] 0.0386	0.23 [0, 0.23] 0.229
5	3.53 [2.89, 3.53] 3.23	3.64 [2.77, 3.64] 3.34	0.0522 [0.0199, 0.0522] 0.0398	60, 73.8 [60, 73.8] 63.4	0.232 [0.163, 0.232] 0.23	0.0451 [0.0342, 0.0451] 0.0378	0.0606 [0.0171, 0.0606] 0.0312	0.169 [0.17, 0.291] 0.169
6	3.61 [2.89, 3.56] 3.61	3.71 [2.56, 3.66] 3.71	0.0296 [0.0253, 0.0605] 0.0296	73.9 [51.2, 78.7] 73.9	0.143 [0.182, 0.267] 0.143	0.0436 [0.0285, 0.0444] 0.0436	0.0557 [0.0129, 0.0575] 0.0557	0.196 [0, 0.217] 0.196
7	3.62 [3.29, 3.94] 3.62	3.95 [3.31, 4.09] 3.95	0.0474 [0.0178, 0.0442] 0.0474	86.1 [59.8, 94.8] 86.1	0.152 [0.107, 0.192] 0.152	0.0576 [0.0371, 0.0576] 0.0521	0.122 [0.0218, 0.122] 0.0773	0.247 [0.126, 0.247] 0.12
8	3.91 [3.42, 4.08] 3.91	4.09 [3.71, 4.3] 4.09	0.085 [0.0247, 0.0678] 0.085	73, 103 [73, 103] 60.8	0.196 [0.0997, 0.196] 0.234	0.0611 [0.0418, 0.0611] 0.0319	0.123 [0.0337, 0.123] 0.0191	0.193 [0, 0.193] 0
9	4 [3.51, 4.37] 4	4.23 [3.44, 4.23] 3.88	0.116 [0.0702, 0.116] 0.063	68.4 [56.2, 68.4] 84.8	0.254 [0.225, 0.254] 0.156	0.0411 [0.0267, 0.0411] 0.0598	0.0397 [0.0105, 0.0397] 0.0258	0.0578 [0, 0.0578] 0
10	3.09 [3.47, 4.4] 3.09	3.12 [3.43, 4.09] 3.12	0.116 [0.042, 0.103] 0.116	75.9 [73.5, 111] 75.9	0.183 [0.148, 0.184] 0.183	0.0439 [0.0439, 0.0723] 0.0439	0.0296 [0.0171, 0.0367] 0.0296	0.301 [0, 0.199] 0.301
11	2.73 [2.81, 3.35] 2.73	2.62 [2.97, 3.39] 2.62	0.0667 [0.0416, 0.172] 0.0667	67 [62, 86.5] 67	0.158 [0.147, 0.234] 0.158	0.0288 [0.0302, 0.0501] 0.0288	0.0259 [0.0165, 0.043] 0.0259	0.105 [0.185, 0.457] 0.105
12	2.86 [2.36, 3.19] 2.86	2.55 [2.43, 3.3] 2.55	0.0594 [0.0325, 0.119] 0.0594	50.6 [45.9, 92.2] 50.6	0.251 [0.0686, 0.186] 0.251	0.0262 [0.0197, 0.0473] 0.0262	0.0107 [0.0104, 0.0767] 0.0107	0 [0, 0.246] 0
13	3.12 [2.19, 2.95] 3.12	3.19 [2.28, 2.7] 3.19	0.0436 [0.014, 0.15] 0.0436	78 [35.3, 70.2] 78	0.195 [0.19, 0.344] 0.195	0.038 [0.0136, 0.0445] 0.038	0.0341 [0.00843, 0.0245] 0.0341	0.145 [0, 0] 0.145
14	1.32 [2.63, 3.73] 1.32	1.41 [2.61, 3.82] 1.41	0.0518 [0.023, 0.102] 0.0518	18.2 [58.5, 94.2] 18.2	0.74 [0.173, 0.226] 0.74	0.0127 [0.0309, 0.0542] 0.0127	0.00343 [0.0215, 0.0595] 0.00343	0 [0, 0.268] 0
15	1.8 [1.21, 1.48] 1.8	1.78 [1.34, 1.61] 1.78	0.0549 [0.0333, 0.0796] 0.0549	26.1 [13.6, 32.2] 26.1	0.317 [0.554, 1.01] 0.317	0.0114 [0.00881, 0.0242] 0.0114	0.0299 [0.00142, 0.0254] 0.0299	0 [0, 0] 0
16	2.34 [1.42, 2.33] 2.34	2.38 [1.53, 2.39] 2.38	0.0699 [0.0322, 0.0998] 0.0699	32.8 [10.4, 55] 32.8	0.317 [0.0995, 0.805] 0.317	0.017 [0.0084, 0.0191] 0.017	0.0144 [0.00718, 0.075] 0.0144	0 [0, 0] 0
17	1.76 [1.73, 2.94] 1.76	1.83 [1.71, 2.99] 1.83	0.0758 [0.029, 0.122] 0.0758	44.8 [25, 43.9] 44.8	0.274 [0.206, 0.47] 0.274	0.0279 [0.0122, 0.0278] 0.0279	0.0297 [0.0102, 0.036] 0.0297	0 [0, 0.102] 0
18	2.08 [1.63, 2.08] 2.08	1.97 [1.68, 1.97] 1.97	0.128 [0.0328, 0.128] 0.128	58 [30.9, 58] 58	0.397 [0.0869, 0.397] 0.397	0.0345 [0.0206, 0.0345] 0.0345	0.167 [0.0124, 0.167] 0.167	0 [0, 0] 0

Table 11: Second half of Table 10

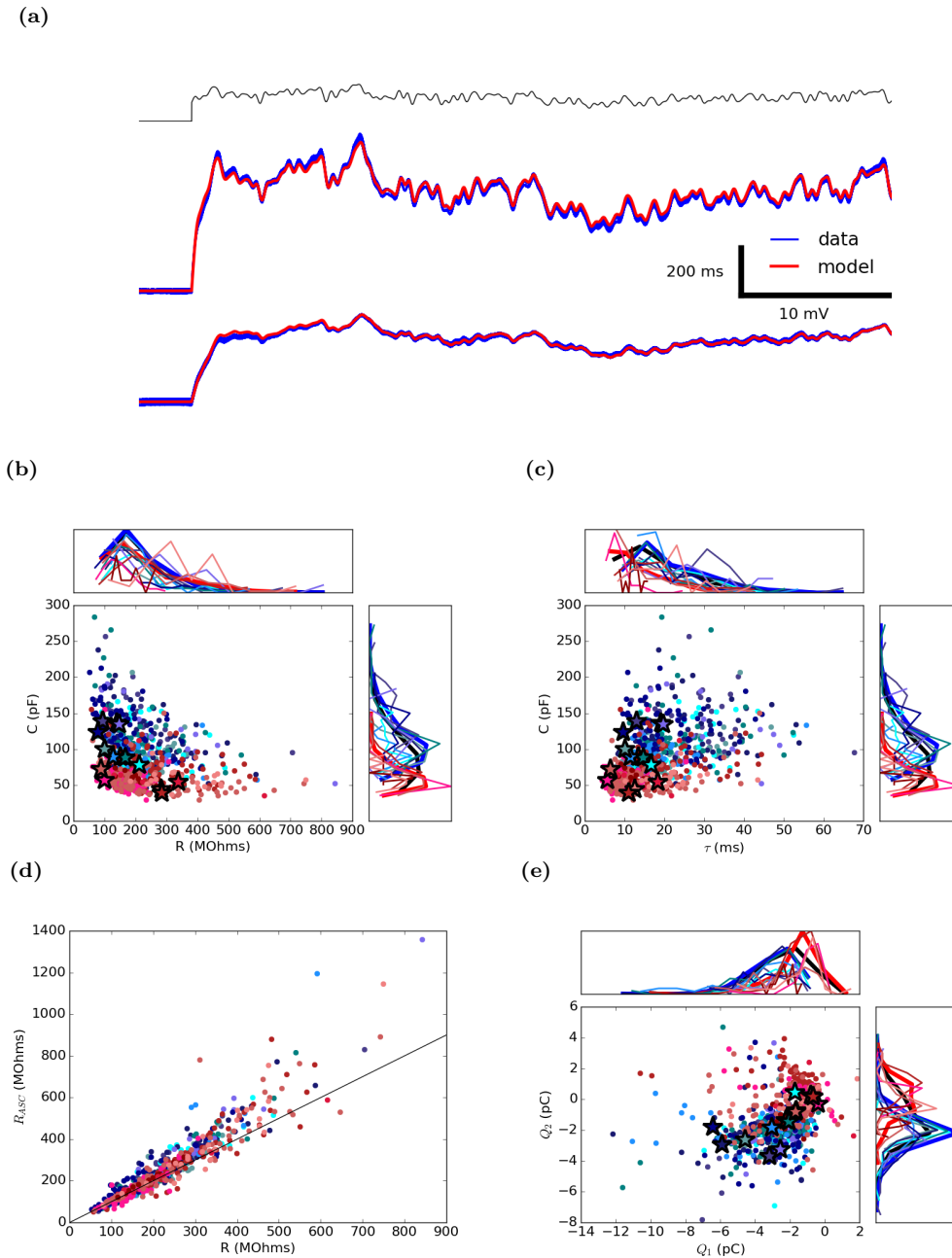


Figure 9: Resistance, Capacitance and amplitudes of after-spike currents are fit via linear regression. (a) Two examples of subthreshold voltage epochs fit via linear regression. Top traces are from the example cells shown in all figure examples (Top: Htr3a 477975366, Bottom: Rorb 314822529). Injected current trace shown in black, voltage of biological neuron recorded from repeated current injections are in blue, voltage of model neuron red. Values of (b) resistance and capacitance and (c) the membrane time constant $\tau=R*C$ and capacitance of all neurons fit to subthreshold data without the simultaneous fitting of after-spike currents. (d) Comparison of resistance values obtained by fitting subthreshold data without after-spike currents and supra-threshold data with the simultaneous fitting of after-spike currents. (e) Total charge deposited by after-spike currents. In scatter plots and corresponding distributions, shades of blue represent excitatory transgenic lines and red denote inhibitory transgenic lines. The thick black histograms represent data from all transgenic lines together, the thick red histograms represent data from inhibitory transgenic lines and the thick blue histogram represents data from excitatory transgenic lines. Stars denote canonical neurons. The full list of colors corresponding to specific transgenic lines can be found in Table 2 Transgenic Lines. Any neuron containing a spike in the subthreshold noise epoch of the training set was globally eliminated from the data set. (c) and (e) are shown in main article (Figure 3)

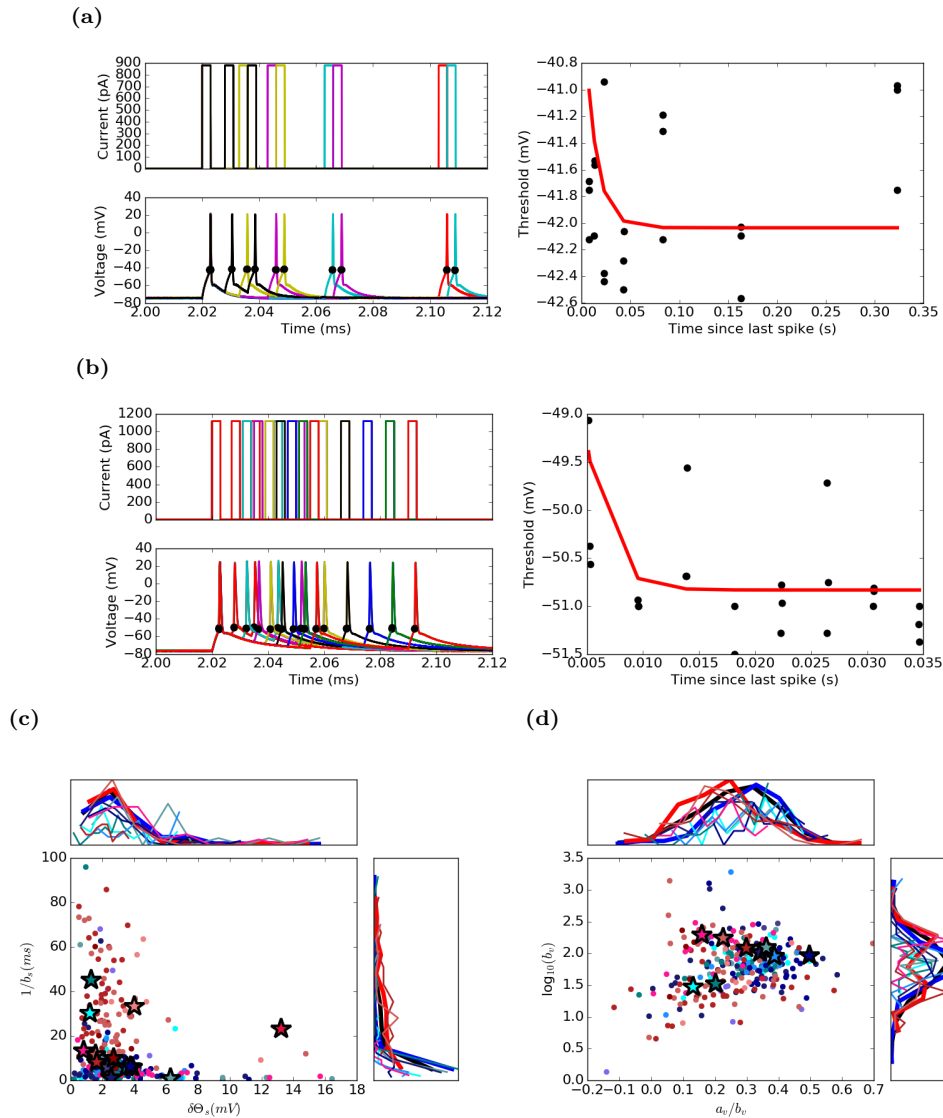


Figure 10: **The spiking component and voltage component of the threshold are incorporated into higher level models.** The spiking component of the threshold is defined by an exponential function fit to the time between spikes and the voltage at spike initiation of the triple short square data set. Exponential fitting of (a) Htr3a 477975366 and (b) Rorb 314822529. Spiking in response to current pulses (left, with colors representing different sweeps) and exponential fit (right). Black dots denote spike initiation. (c) shows the fit amplitude, a , incorporated by the reset rule δ_s and the decay constant, b of all neurons with triple square pulse data. All neurons with an $0 < a < 20\text{mV}$ or a $1/b > 0.1\text{s}$ are excluded. (d) In the $GLIF_5$ model the threshold is influenced by the voltage of the neuron. Parameters from equation 20 are plotted. $GLIF_5$ models where a_v is < -50 or b_v is < 0.1 are excluded. (c) and (d) are shown in main article (Figure 3)

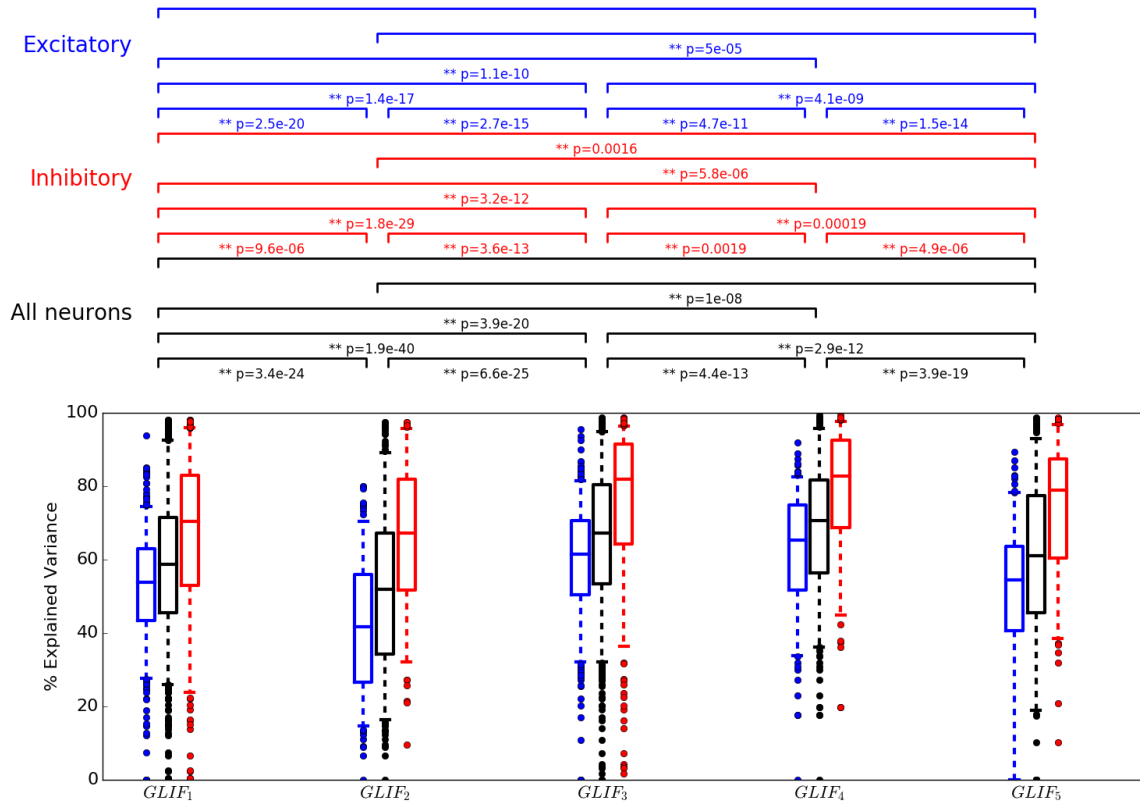


Figure 11: **Post-hoc Optimization of θ_∞ improves explained variance.** Summary for GLIF models for all 771 neurons (black), all 315 inhibitory neurons (red) and all 456 excitatory neurons (blue) for models with measured as opposed to post-hoc optimized θ_∞ . Although on average θ_∞ values change very little (Figure 7) before and after optimization, average explained variance values rise (Can an be seen by comparing this figure with Figure 5 of the main text). Box plots display median, and quartiles. Whiskers reach to 5% and 95% quartiles. Individual data points lie outside whiskers. Brackets above plot denote significant difference between distributions assessed via a Wilcoxon sign ranked test: a single asterisk (*) represents a p-value < 0.05, a double asterisk (**) represents a p-value < 0.01.

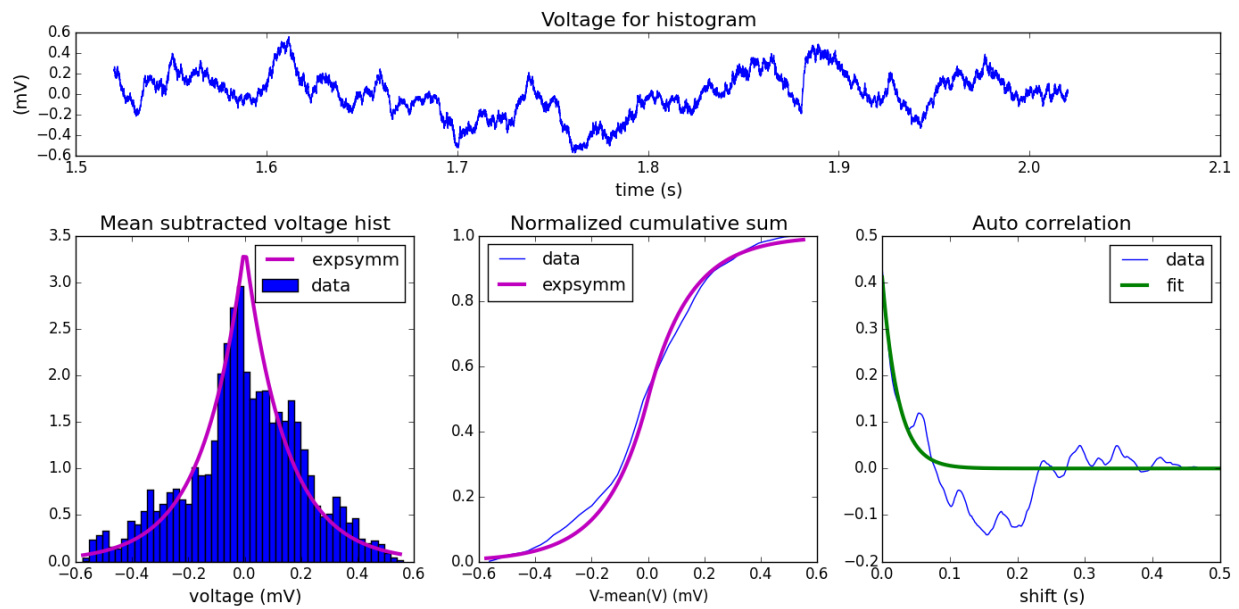


Figure 12: **Parameters of the MLIN objective function were extracted from the data.** A distribution of voltages is created from the tailing end of the largest amplitude subthreshold square pulse available (top panel). This distribution is then fit by a symmetrically decaying exponential function denoted as `expsymm` in the plot legends. The width of the non-spiking bins is chosen via a fit of the autocorrelation.

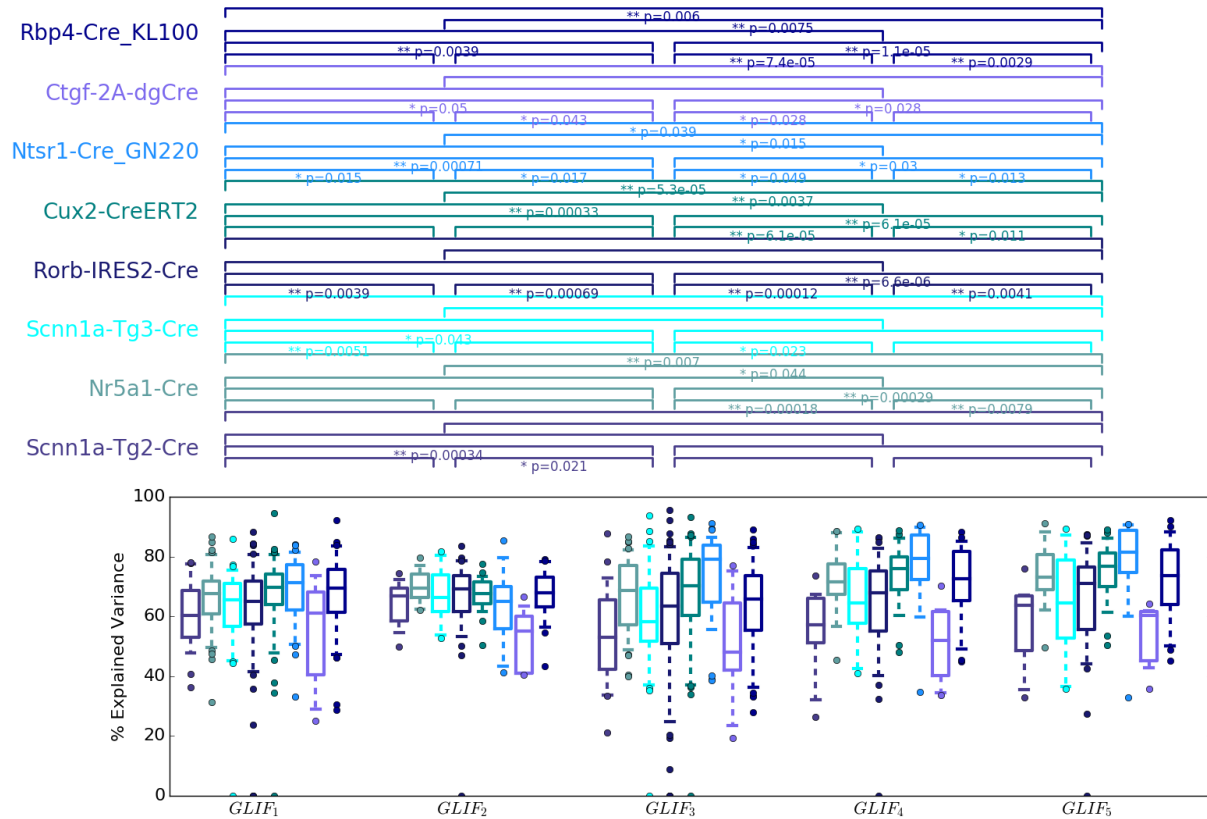


Figure 13: **Explained variance summary for excitatory GLIF models of different levels using threshold infinity obtained via MLIN optimization.** Box plots display median, and quartiles. Whiskers reach to 5% and 95% quartiles. Individual data points lie outside whiskers. Brackets above plot denote significant difference between distributions assessed via a Wilcoxon sign ranked test: a single asterisk (*) represents a p-value < 0.05, a double asterisk (**) represents a p-value < 0.01. Distribution values are available in Table 4

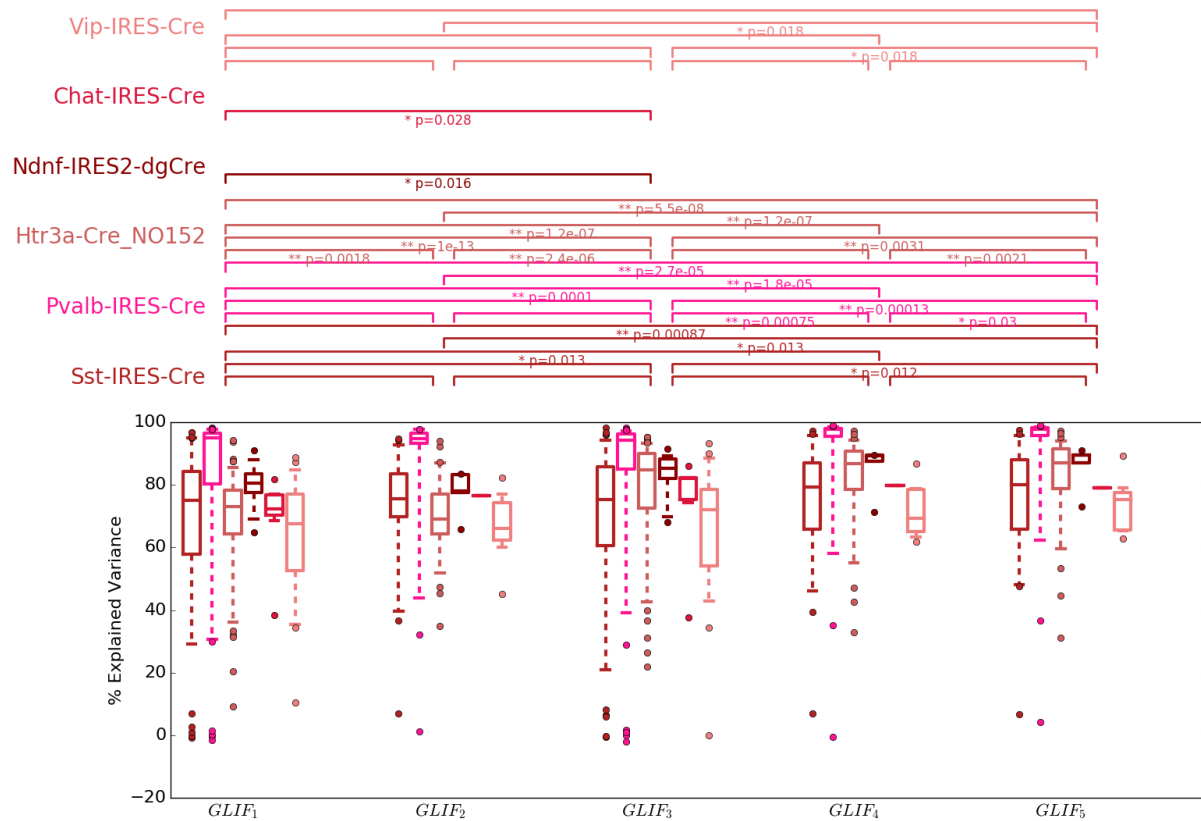
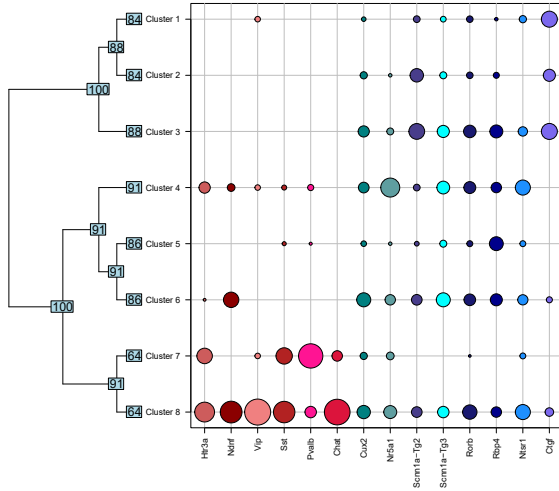
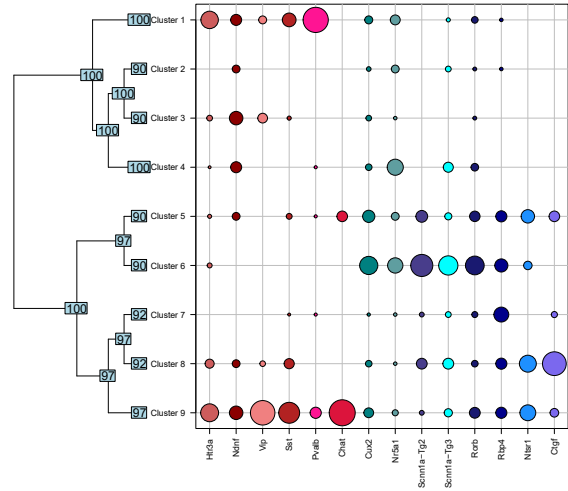


Figure 14: **Explained variance summary for inhibitory GLIF models of different levels using threshold infinity obtained via MLIN optimization.** Box plots display median, and quartiles. Whiskers reach to 5% and 95% quartiles. Individual data points lie outside whiskers. Brackets above plot denote significant difference between distributions assessed via a Wilcoxon sign ranked test: a single asterisk (*) represents a p-value < 0.05, a double asterisk (**) represents a p-value < 0.01. Distribution values are available in Table 4

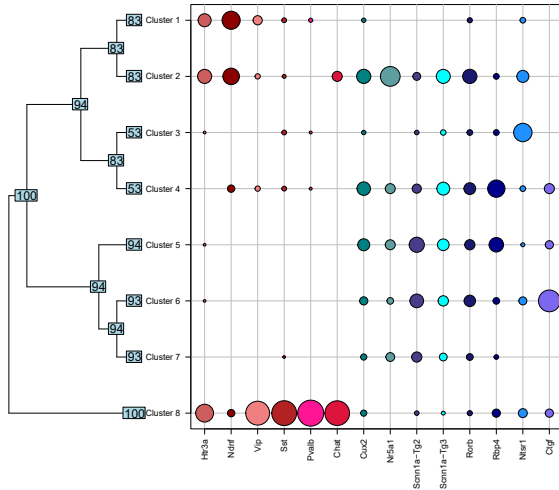
(a) GLIF 1



(b) GLIF 2



(c) GLIF 3



(d) GLIF 4

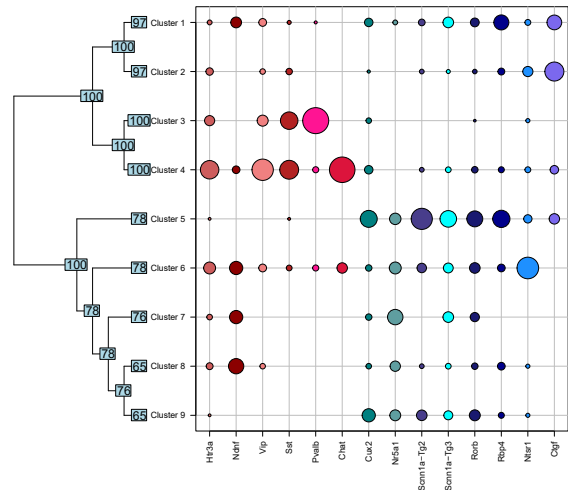


Figure 15: Clustering of cells using parameters obtained from four of the GLIF models described in the text (Table 1).

References

- Allen Institute for Brain Science, 2016a. Allen Cell Types Database. URL: <http://celltypes.brain-map.org/>.
- Allen Institute for Brain Science, 2016b. Allen Cell Types Database, Technical White Paper: Electrophysiology.
- Brette, R., Gerstner, W., 2005. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology* 94, 3637–3642.
- Dong, Y., Mihalas, S., Kim, S.S., Yoshioka, T., Bensmaia, S., Niebur, E., 2013. A simple model of mechanotransduction in primate glabrous skin. *Journal of Neurophysiology* 109, 1350–1359.
- Dong, Y., Mihalas, S., Russell, A., Etienne-Cummings, R., Niebur, E., 2011. Estimating parameters of generalized integrate-and-fire neurons from the maximum likelihood of spike trains. *Neural computation* 23, 2833–2867.
- FitzHugh, R., 1961. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal* 1, 445.
- Gerstner, W., Naud, R., 2009. How good are neuron models? *Science* 326, 379–380.
- Gollisch, T., Meister, M., 2010. Eye smarter than scientists believed: neural computations in circuits of the retina. *Neuron* 65, 150–164.
- Hay, E., Hill, S., Schürmann, F., Markram, H., Segev, I., 2011. Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS Comput Biol* 7, e1002107.
- Herz, A.V., Gollisch, T., Machens, C.K., Jaeger, D., 2006. Modeling single-neuron dynamics and computations: a balance of detail and abstraction. *science* 314, 80–85.
- Hodgkin, A.L., Huxley, A.F., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology* 117, 500.
- Izhikevich, E.M., 2004. Which model to use for cortical spiking neurons. *IEEE Transactions on Neural Networks* 15, 1063–1070.
- Izhikevich, E.M., 2007. *Dynamical systems in neuroscience*. MIT press.
- Izhikevich, E.M., et al., 2003. Simple model of spiking neurons. *IEEE Transactions on neural networks* 14, 1569–1572.
- Koch, C., 2004. *Biophysics of computation: information processing in single neurons*. Oxford university press.
- Lapicque, L., 1907. Recherches quantitatives sur l’excitation électrique des nerfs traitée comme une polarisation. *Journal de Physiologie et Pathologie General* 9, 620–635.
- Markram, H., Müller, E., Ramaswamy, S., Reimann, M.W., Abdellah, M., Sanchez, C.A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al., 2015. Reconstruction and simulation of neocortical microcircuitry. *Cell* 163, 456–492.
- Mensi, S., Naud, R., Pozzorini, C., Avermann, M., Petersen, C.C., Gerstner, W., 2012. Parameter extraction and classification of three cortical neuron types reveals two distinct adaptation mechanisms. *Journal of neurophysiology* 107, 1756–1775.
- Mihalas, S., Niebur, E., 2009. A generalized linear integrate-and-fire neural model produces diverse spiking behaviors. *Neural computation* 21, 704–718.

- Morris, C., Lecar, H., 1981. Voltage oscillations in the barnacle giant muscle fiber. *Biophysical journal* 35, 193.
- Paninski, L., Pillow, J.W., Simoncelli, E.P., 2004. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural computation* 16, 2533–2561.
- Pozzorini, C., Mensi, S., Hagens, O., Naud, R., Koch, C., Gerstner, W., 2015. Automated high-throughput characterization of single neurons by means of simplified spiking models. *PLoS Comput Biol* 11, e1004275.
- Prinz, A.A., Billimoria, C.P., Marder, E., 2003. Alternative to hand-tuning conductance-based models: construction and analysis of databases of model neurons. *Journal of Neurophysiology* 90, 3998–4015.
- Rossant, C., Goodman, D.F., Platkiewicz, J., Brette, R., 2010. Automatic fitting of spiking neuron models to electrophysiological recordings. *Frontiers in neuroinformatics* 4, 2.
- Tasic, B., Menon, V., Nguyen, T.N., Kim, T.K., Jarsky, T., Yao, Z., Levi, B., Gray, L.T., Sorensen, S.A., Dolbeare, T., et al., 2016. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature neuroscience* 19, 335–346.