The Computational Structure of Spike Trains

LETTER Communicated by Herbert Jaeger

The Computational Structure of Spike Trains

Robert Haslinger
robhh@nmr.mgh.harvard.edu
Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Charlestown, MA 02129, U.S.A., and Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Kristina Lisa Klinkner
klinkner@stat.cmu.edu
Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

Cosma Rohilla Shalizi
cshalizi@stat.cmu.edu
Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A., and Santa Fe Institute, Santa Fe, NM 87051, U.S.A.

Neurons perform computations, and convey the results of those computations through the statistical structure of their output spike trains. Here we present a practical method, grounded in the information-theoretic analysis of prediction, for inferring a minimal representation of that structure and for characterizing its complexity. Starting from spike trains, our approach finds their causal state models (CSMs), the minimal hidden Markov models or stochastic automata capable of generating statistically identical time series. We then use these CSMs to objectively quantify both the generalizable structure and the idiosyncratic randomness of the spike train. Specifically, we show that the expected algorithmic information content (the information needed to describe the spike train exactly) can be split into three parts describing (1) the time-invariant structure (complexity) of the minimal spike-generating process, which describes the spike train statistically; (2) the randomness (internal entropy rate) of the minimal spike-generating process; and (3) a residual pure noise term not described by the minimal spike-generating process. We use CSMs to approximate each of these quantities. The CSMs are inferred nonparametrically from the data, making only mild regularity assumptions, via the causal state splitting reconstruction algorithm. The methods presented here complement more traditional spike train analyses by describing not only spiking probability and spike train entropy, but also the complexity of a spike train's structure. We demonstrate our approach using both simulated spike trains and experimental data recorded in rat barrel cortex during vibrissa stimulation.

1 Introduction

The recognition that neurons are computational devices is one of the foundations of modern neuroscience (McCulloch & Pitts, 1943). However, determining the functional form of such computation is extremely difficult, if only because while one often knows the output (the spikes), the input (synaptic activity) is almost always unknown. Often, therefore, scientists must draw inferences about the computation from its results, namely the output spike trains and their statistics. In this vein, many researchers have used information theory to determine, via calculation of the entropy rate, a neuron's channel capacity: how much information the neuron could conceivably transmit, given the distribution of observed spikes (Rieke, Warland, de Ruyter van Steveninck, & Bialek, 1997). However, entropy quantifies randomness and says little about how much structure a spike train has, or the amount and type of computation that it must have, at a minimum, taken place to produce this structure. Here, and throughout this letter, we mean "computational structure" information theoretically: the most compact effective description of a process capable of statistically reproducing the observed spike trains. The complexity of this structure is the number of bits needed to describe it. This is different from the algorithmic information content of a spike train, which is the number of bits needed to reproduce the latter exactly, describing not only its regularities but also its accidental, noisy details.

Our goal is to develop rigorous yet practical methods for determining the minimal computational structure necessary and sufficient to generate neural spike trains. We are able to do this through nonparametric analysis of the directly observable spike trains, without resorting to a priori assumptions about what kind of structure they have. We do this by identifying the minimal hidden Markov model (HMM), which can statistically predict the future of the spike train without loss of information. This HMM also generates spike trains with the same statistics as the observed train. It thus defines a program that describes the spike train's computational structure, letting us quantify, in bits, the structure's complexity.

From multiple directions, several groups, including our own, have shown that minimal generative models of time series can be discovered by clustering histories into "states," based on their conditional distributions over future events (Crutchfield & Young, 1989; Grassberger, 1986; Jaeger, 2000; Knight, 1975; Littman, Sutton, & Singh, 2002; Shalizi & Crutchfield, 2001). The observed time series need not be Markovian (few spike trains are), but the construction always yields the minimal HMM capable of generating and predicting the original process. Following Shalizi (2001) and Shalizi and Crutchfield (2001), we will call such an HMM a causal state model (CSM). Within this framework, the model discovery algorithm, called causal state splitting reconstruction (CSSR; Shalizi & Klinkner, 2004), is an adaptive nonparametric method that consistently estimates a system's

CSM from time-series data. In this letter, we adapt CSSR for use in spike train analysis.

CSSR provides nonparametric estimates of the time- and history-dependent spiking probabilities found by more familiar parametric analyses. Unlike those analyses, it is also capable, in the limit of infinite data, of capturing all the information about the computational structure of the spike-generating process contained in the spikes themselves. In particular, the CSM quantifies the complexity of the spike-generating process by showing how much information about the history of the spikes is relevant to their future, that is, how much information is needed to reproduce the spike train statistically. This is equivalent to the log of the effective number of statistically distinct states of the process (Crutchfield & Young, 1989; Grassberger, 1986; Shalizi & Crutchfield, 2001). While this is not the same as the algorithmic information content, we show that CSMs can also approximate the average algorithmic information content, splitting it into three parts: (1) the generative process's complexity in our sense; (2) the internal entropy rate of the generative process, the extra information needed to describe the exact state transitions the undergone while generating the spike train; and (3) the residual randomness in the spikes, unconstrained by the generative process. The first of these quantifies the spike train's structure, the last two its randomness.

We give precise definitions of these quantities—both their ensemble averages (in section 2.3) and their functional dependence on time (in section 2.4). The time-dependent versions allow us to determine when the neuron is traversing states requiring complex descriptions. Our methods put hard numerical lower bounds on the amount of computational structure that must be present to generate the observed spikes. They also quantify, in bits, the extent to which the neuron is driven by external forces. We demonstrate our approach using both simulated and experimentally recorded single-neuron spike trains. We discuss the interpretation of our measures and how they add to our understanding of neuronal computation.

2 Theory and Methods

Throughout this letter, we treat spike trains as stochastic binary time series, with time divided into discrete, equal-duration bin steps (typically at 1 millisecond resolution); 1 corresponds to a spike and 0 to no spike. Our aim is to find a minimal description of the computational structure present in such a time series. Heuristically, the structure present in a spike train can be described by a program, which can reproduce the spikes statistically. The information needed to describe this program (loosely speaking, the program length) quantifies the structure's complexity. Our approach uses minimal, optimally predictive HMMs, or causal state models (CSMs), reconstructed from the data, to describe the program. (We clarify our use of

minimal below.) The CSMs are then used to calculate various measures of the computational structure, such as its complexity.

The states are chosen so that they are optimal predictors of the spike train's future, using only the information available from the train's history. (We discuss the limitations of this below.) Specifically the states S_t are defined by grouping the histories of past spiking activity $X_{-\infty}^{t}$, which occur in the spike train, into equivalence classes, where all members of a given equivalence class are statistically equivalent in terms of predicting the future spiking X_{t+1}^{∞}. ($X_{t'}^{t}$ denotes the sequence of random observables, i.e., spikes or their absence, between t' and $t > t'$, while X_t denotes the random observable at time t. The notation is similar for the states.) This construction ensures that the causal states are Markovian, even if the spike train is not (Shalizi & Crutchfield, 2001). Therefore, at all times t, the system and its possible future evolutions can be specified by the state S_t. Like all other HMMs, a CSM can be represented pictorially by a directed graph, with nodes standing for the process's hidden states and directed edges the possible transitions between these states. Each edge is labeled with the observable or symbol emitted during the corresponding transition (1 for a spike and 0 for no spike) and the probability of traversing that edge given that the system started in that state. The CSM also specifies the time-averaged probability of occupying any state (via the ergodic theorem for Markov chains).

The theory is described in more detail below, but at this point, examples may clarify the ideas. Figures 1A and 1B show two simple CSMs. Both are built from simulated ≈ 40 Hz spike trains 200 seconds in length (1 msec time bins, $p = 0.04$ independent and identically distributed (i.i.d.) at each time when spiking is possible). However, spike trains generated from the CSM in Figure 1B have a 5 msec refractory period after each spike (when $p = 0$), while the spiking rate in nonrefractory periods is still 40 Hz ($p = 0.04$). The refractory period is additional structure, represented by the extra states. State A represents the status of the neuron during 40 Hz spiking, outside of the refractory periods. While in this state, the neuron either emits no spike ($X_{t+1} = 0$), staying in state A, or emits a spike ($X_{t+1} = 1$) with probability $p = 0.04$ and moves to state B. The equivalence class of past spiking histories defining state A therefore includes all past spiking histories for which the most recent five symbols are 0, symbolically $\{*00000\}$. State B is the neuron's state during the first msec of the refractory period. It is defined by the set of spiking histories $\{*1\}$. No spike can be emitted during a refractory period, so the transition to state *C* is certain, and the symbol emitted is always 0. In this manner, the neuron proceeds through states C to F and back to state A, where it is possible to spike again.

The rest of this section is divided into four subsections. First, we briefly review the formal theory behind CSMs (for details, see Shalizi, 2001; Shalizi & Crutchfield, 2001) and discuss why they can be considered a good choice for understanding the structural content of spike trains. Second, we

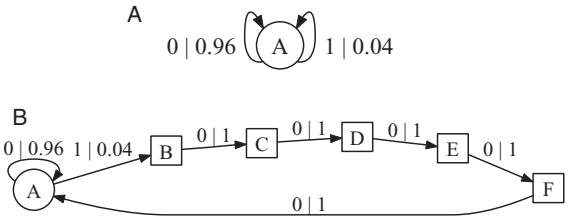

Figure 1: Two simple CSMs reconstructed from 200 sec of simulated spikes using CSSR. States are represented as the nodes of a directed graph. The transitions between states are labeled with the symbol emitted during the transition (1 = spike, 0 = no spike) and the probability of the transition given the origin state. (A) The CSM for a 40 Hz Bernoulli spiking process consists of a single state A, which always transitions back to itself, emitting a spike with probability $p = 0.04$ per msec. (B) CSM for a 40 Hz Bernoulli spiking process with a 5 msec refractory period imposed after each spike. State A again spikes with probability $p = 0.04$. Upon spiking, the CSM transitions through a deterministic chain of states B to F (squares), which represent the refractory period. The increased structure of the refractory period requires a more complex representation.

describe the causal state splitting reconstruction (CSSR) algorithm used to reconstruct CSMs from observed spike trains (Shalizi & Klinkner, 2004). We emphasize that CSSR requires no a priori knowledge of the structure of the CSM discovered from the spike train. Third, we discuss two different notions of spike train structure: statistical complexity and algorithmic information content. These two measures can be interpreted as different aspects of a spike train's computational structure, and each can be related to the reconstructed CSM. Finally, we show how the reconstructed CSM can be used to predict spiking, measure the neural response, and detect the influence of external stimuli.

2.1 Causal State Models. The foundation of the theory of causal states is the concept of a predictively sufficient statistics. A statistic, η, on one random variable, X, is sufficient for predicting another random variable, Y, when $\eta(X)$ and X have the same information[1] about Y, $I[X; Y] = I[\eta(X); Y]$. This holds if and only if X and Y are conditionally independent given $\eta(X)$: $\mathbb{P}(Y \mid X, \eta(X)) = \mathbb{P}(Y \mid \eta(X))$. This is a close relative of the familiar idea of parametric sufficiency; in Bayesian statistics, where parameters are random variables, parametric sufficiency is a special case of predictive sufficiency (Bernardo & Smith, 1994). Predictive sufficiency shares all of parametric sufficiency's optimality properties (Bernardo & Smith, 1994). However, a statistic's predictive sufficiency depends on only the actual joint distribution

[1]See Cover and Thomas (1991) for information-theoretic definitions and notation.

of X and Y, not on any parametric model of that distribution. Again as in the parametric case, a minimal predictively sufficient statistic ϵ is one that is a function of every other sufficient statistic η: $\epsilon(X) = h(\eta(X))$ for some h. Minimal sufficient statistics are the most compact summaries of the data, which retain all the predictively relevant information. A basic result is that a minimal sufficient statistic always exists and is (essentially) unique, up to isomorphism (Bernardo & Smith, 1994; Shalizi & Crutchfield, 2001).

In the context of stochastic processes such as spike trains, ϵ is the minimal sufficient statistic of the history $X_{-\infty}^{t}$ for predicting future of the process, X_{t+1}^{∞}. This statistic is the optimal predictor of the observations. The sequence of values of the minimal sufficient statistic, $S_t = \epsilon(X_{-\infty}^{t})$, is another stochastic process. This process is always a homogeneous Markov chain, whether or not the X_t process is (Knight, 1975; Shalizi & Crutchfield, 2001). Turned around, this means that the original X_t process is always a random function of a homogeneous Markov chain, whose latent states, named the *causal states* by Crutchfield and Young (1989), are optimal, minimal predictors of the future of the time series.

A causal state model or causal state machine is a stochastic automaton or HMM constructed so that its Markov states are minimal sufficient statistics for predicting the future of the spike train and consequently can generate spike trains statistically identical to those observed.[2] Causal state reconstruction means inferring the causal states from the observed spike train. Following Crutchfield and Young (1989) and Shalizi and Crutchfield (2001), the causal states can be seen as equivalence classes of spike train histories $X_{-\infty}^{t}$, which maximize the mutual information between the states and the future of the spike train X_{t+1}^{∞}. Because they are sufficient, they predict the future of the spike train as well as it can be predicted from its history alone. Because they are minimal, the number of states or equivalence classes is as small as it can be without discarding predictive power.[3]

Formally, two histories, x^- and y^-, are equivalent when $\mathbb{P}(X_{t+1}^{\infty} \mid X_{-\infty}^{t} = x^-) = \mathbb{P}(X_{t+1}^{\infty} \mid X_{-\infty}^{t} = y^-)$. The equivalence class of x^- is $[x^-]$. Define the function that maps histories to their equivalence classes:

$$\begin{aligned}\epsilon(x^-) &\equiv [x^-] \\ &= \left\{ y^- : \mathbb{P}\left(X_{t+1}^{\infty} \mid X_{-\infty}^{t} = y^-\right) = \mathbb{P}\left(X_{t+1}^{\infty} \mid X_{-\infty}^{t} = x^-\right) \right\}.\end{aligned}$$

[2]Some authors use *hidden Markov model* only for models where the current observation is independent of all other variables given the current state, and call the broader class, which includes CSMs, *partially observable Markov model*.

[3]There may exist more compact representations, but then the states, or their equivalents, can never be empirically identified (see Shalizi & Crutchfield, 2001, or Löhr & Ay, 2009).

The causal states are the possible values of ϵ (i.e., the equivalence classes). Each corresponds to a distinct distribution for the future. The state at time t is $S_t = \epsilon(X_{-\infty}^t)$. Clearly, $\epsilon(x^-)$ is a sufficient statistic. It is also minimal, since if η is sufficient, then $\eta(x^-) = \eta(y^-)$ implies $\epsilon(x^-) = \epsilon(y^-)$. One can further show (Shalizi & Crutchfield, 2001) that ϵ is the unique minimal sufficient statistic, meaning that any other must be isomorphic to it.

In addition to being minimal sufficient statistics, the causal states have some other important properties that make them ideal for quantifying structure (Shalizi & Crutchfield, 2001). As mentioned, $\{S_t\}$ is a Markov process, and one can write the observed process X as a random function of the causal state process—X has a natural hidden-Markov-model representation. The causal states are recursively calculable; there is a function T such that $S_{t+1} = T(S_t, X_{t+1})$ (see appendix A). And CSMs are closely related to the predictive state representations of controlled dynamical systems (Littman et al., 2002; Singh, Littman, Jong, Pardoe, & Stone, 2003; see appendix C).

2.2 Causal State Splitting Reconstruction. Our goal is to find a minimal sufficient statistic for the spike train, which will form a hidden Markov model. The states of this model are equivalence classes of spiking histories $X_{-\infty}^t$. In practice, we need an algorithm that can both cluster histories into groups that preserve their conditional distribution of futures and find the history length Λ at which the past may be truncated while preserving the computational structure of the spike train. The former is accomplished by the CSSR algorithm (Shalizi & Klinkner, 2004) for inferring causal states from data by building a recursive next-step-sufficient statistic.[4] We do the latter by minimizing Schwartz's Bayesian information criterion (BIC) over Λ.

To save space, we just sketch the CSSR algorithm here.[5] CSSR starts by treating the process as an independent and identically distributed sequence, with one causal state. It adds states when statistical tests show that the current set of states is not sufficient. Suppose we have a sequence $x_1^N = x_1, x_2, \ldots, x_N$ of length N from a finite alphabet $\mathcal{A}$ of size k. We wish to derive from this an estimate $\hat{\epsilon}$ of the minimal sufficient statistic ϵ. We do this by finding a set Σ of states, each of which will be a set of strings,

[4] A next-step-sufficient statistic contains all the information needed for optimal one-step-ahead prediction, $I[X_{t+1}; \eta(X_{-\infty}^t)] = I[X_{t+1}; X_{-\infty}^t]$, but not necessarily for longer predictions. CSSR relies on the theorem that if η is next step sufficient and it is recursively calculable, then η is sufficient for the whole of the future (Shalizi & Crutchfield, 2001). CSSR first finds a next-step sufficient statistic and then refines it to be recursive.

[5] In addition to Shalizi and Klinkner (2004), which gives pseudocode, some details of convergence, and applications to process classification, are treated in Klinkner and Shalizi (2009) and Klinkner, Rinaldo, and Shalizi (2009). An open source C++ implementation is available online at http://bactra.org/CSSR/. The CSMs generated by CSSR can be displayed graphically, as we do in this letter, with the open source program dot (http://www.graphviz.org/).

or finite-length histories. The function $\hat{\epsilon}$ will then map a history x^- to whichever state contains a suffix of x^- (taking "suffix" in the usual string-manipulation sense). Although each state can contain multiple suffixes, one can check (Shalizi & Klinkner, 2004) that the mapping $\hat{\epsilon}$ will never be ambiguous.

The null hypothesis is that the process is Markovian on the basis of the states in Σ,

$$\mathbb{P}\left(X_t \mid X_{t-L}^{t-1} = a\, x_{t-L+1}^{t-1}\right) = \mathbb{P}\left(X_t \mid \hat{S} = \hat{\epsilon}\left(x_{t-L+1}^{t-1}\right)\right), \tag{2.1}$$

for all $a \in \mathcal{A}$. In words, adding an extra piece of history does not change the conditional distribution for the next observation. We can check this with standard statistical tests, such as χ^2 or Kolmogorov-Smirnov (KS). In this letter, we used a KS test of size $\alpha = 0.01$.[6] If we reject this hypothesis, we fall back on a restricted alternative hypothesis: that we have the right set of conditional distributions but have matched them with the wrong histories, that is,

$$\mathbb{P}\left(X_t \mid X_{t-L}^{t-1} = a\, x_{t-L+1}^{t-1}\right) = \mathbb{P}\left(X_t \mid \hat{S} = s^*\right), \tag{2.2}$$

for some $s^* \in \Sigma$, but $s^* \neq \hat{\epsilon}(x_{t-L+1}^{t-1})$. If this hypothesis passes a test of size α, then s^* is the state to which we assign the history.[7] Only if equation 2.2 is itself rejected do we create a new state, with the suffix $a\, x_{t-L+1}^{t-1}$.[8]

The algorithm itself has three phases. Phase 1 initializes Σ to a single state, which contains only the null suffix $\emptyset$ (i.e., $\emptyset$ is a suffix of any string). The length of the longest suffix in Σ is L; this starts at 0. Phase 2 iteratively tests the successive versions of the null hypothesis, equation 2.1, and L increases by 1 each iteration, until we reach some maximum length Λ. At the end of II, $\hat{\epsilon}$ is (approximately) next step sufficient. Phase 3 makes $\hat{\epsilon}$ recursively calculable by splitting the states until they have deterministic transitions. Under mild technical conditions (a finite true number of states), CSSR converges in probability on the correct CSM as $N \to \infty$, provided only that Λ is long enough to discriminate all of the states. The error of the predicted distributions of futures $\mathbb{P}(X_{t+1}^{\infty} \mid X_{-\infty}^{t})$, measured by total variation

[6] For finite N, decreasing α tends to yield simpler CSMs with fewer states. In a sense, it is a sort of regularization coefficient. The influence of this regularization diminishes as N increases. For the data used in section 3, varying α in the range $0.001 < \alpha < 0.1$ made little difference.

[7] If more than one such state s^* exists, we chose the one for which $\widehat{\mathbb{P}}(X_t \mid \hat{S} = s^*)$ differs least, in total variation distance, from $\widehat{\mathbb{P}}(X_t \mid_{t-L}^{t-1} = a\, x_{t-L+1}^{t-1})$, which is plausible and convenient. However, which state we chose is irrelevant in the limit $N \to \infty$, so long as the difference between the distributions is not statistically significant.

[8] The conceptually similar algorithm of Kennel and Mees (2002) in effect always creates a new state, which leads to more complex models, sometimes infinitely more complex ones. See Shalizi and Klinkner (2004).

distance, decays as $N^{-1/2}$. Section 4 of Shalizi and Klinkner (2004) details CSSR's convergence properties. Comparisons of CSSR's performance with that of more traditional expectation-maximization-Based approaches can also be found in Shalizi and Klinkner (2004) as can time complexity bounds for the algorithm. Depending on the machine used, CSSR can process an $N = 10^6$ time series in under 1 minute.

2.2.1 Choosing Λ. CSSR requires no a priori knowledge of the CSM's structure, but it does need a choice of of Λ. Here, pick it by minimizing the BIC of the reconstructed models over Λ,

$$BIC \equiv -2 \log \mathcal{L} + d \log N, \tag{2.3}$$

where $\mathcal{L}$ is the likelihood, N is the data length, and d is the number of model parameters—in our case, the number of predictive states.[9] BIC's logarithmic-with-N penalty term helps keep the number of causal states from growing too quickly with increased data size, which is why we use it instead of the Akaike information criterion (AIC). Also, BIC is known to be consistent for selecting the order of Markov chains and variable-length Markov models (Csiszár & Talata, 2006), both of which are subclasses of CSMs.

Writing the observed spike train as x_1^N and the state sequence as s_0^N, the total likelihood of the spike train is

$$\mathcal{L} = \sum_{s_0^N \in \Sigma^{N+1}} \mathbb{P}\big(X_1^N = x_1^N \,\big|\, S_0^N = s_0^N\big)\mathbb{P}\big(S_0^N = s_0^N\big), \tag{2.4}$$

the sum over all possible causal state sequences of the joint probability of the spike train and the state sequence. Since the states update recursively, $s_{t+1} = T(s_t, x_{t+1})$, the starting state s_0 and the spike train x_1^N fix the entire state sequence s_0^N. Thus, the sum over state sequences can be replaced by a sum over initial states,

$$\mathcal{L} = \sum_{s_i \in \Sigma} \mathbb{P}\big(X_1^N = x_1^N \,\big|\, S_0 = s_i\big)\mathbb{P}(S_0 = s_i), \tag{2.5}$$

[9]The number of independent parameters d involved in describing the CSM will be (number of states) * (number of symbols $-$ 1) since the sum of the outgoing probabilities for each state is constrained to be 1. Thus, for a binary alphabet, d = number of states.

with the state probabilities $\mathbb{P}(S_0 = s_i)$ coming from the CSM. By the Markov property,

$$\mathbb{P}\big(X_1^N = x_1^N \,\big|\, S_0 = s_i\big) = \prod_{j=1}^{N} \mathbb{P}(X_j = x_j \mid S_{j-1} = s_{j-1}). \tag{2.6}$$

Selecting Λ is now straightfoward. For each value of Λ, we build the CSM from the spike train, calculate the likelihood using equations 2.5 and 2.6, and pick the value, and CSM, minimizing equation 2.3. We try all values of Λ up to a model-independent upper bound. For a wide range of stochastic processes, Marton and Shields (1994) showed that the length m of sub-sequences for which probabilities can be consistently and nonparametrically estimated can grow as fast as $\log N/h$, where h is the entropy rate, but no faster. CSSR estimates the distribution of the next symbol given the previous Λ symbols, which is equivalent to estimating joint probabilities of blocks of length $m = \Lambda + 1$. Thus, Marton and Shield's result limits the usable values of Λ:

$$\Lambda \leq \frac{\log N}{h} - 1. \tag{2.7}$$

Using equation 2.7 requires the entropy rate h. The latter can either be upper-bounded as the log of the alphabet size (here, $\log 2 = 1$) or by some other, less pessimistic, estimator of the entropy rate (such as the output of CSSR with $\Lambda = 1$). Use of an upper bound on h results in a conservative maximum value for Λ. For example, a 30 minute experiment with 1 msec time bins lets us use at least $\Lambda \approx 20$ by the most pessimistic estimate of $h = 1$; the actual maximum value of Λ may be much larger. We use $\Lambda \leq 25$ in this letter but see no indication that this cannot be extended further if need be.

2.2.2 Condensing the CSM. For real neural data, the number of causal states can be very large—hundreds or more. This creates an interpretation problem, if only because it is hard to fit such an CSM on a single page for inspection. We thus developed a way to reduce the full CSM while still accounting for most of the spike train's structure. Our "state culling" technique found the least-probable states and selectively removed them, appropriately redirecting state transitions and reassigning state occupation probabilities. By keeping the most probable states, we focus on the ones that contribute the most to the spike train's structure and complexity. Again, we used BIC as our model selection criterion.

First, we sorted the states by probability, finding the least probable state ("remove" state) with a single incoming edge from a state (its "ancestor") with outgoing transitions to two different states: the remove state and a

second "keep" state. We redirected both of the ancestor's outgoing edges to the keep state. Second, we reassigned the remove state's outgoing transitions to the keep state. If the outgoing transitions from the keep state were still deterministic (at most a single 0 emitting edge and a single 1 emitting edge), we stopped. If the transitions were nondeterministic, we merged states reached by emitting 0s with each other (likewise, those reached by 1s), repeating this until termination. Third, we checked that there existed a state sequence of the new model that could generate the observed spikes. If there was, we accepted the new CSM. If not, we rejected the new CSM and chose the next lowest probability state from the original CSM to remove.

This culling was iterated until removing any state made it impossible for the CSM to generate the spike train. At each iteration, we calculated BIC (as described in the previous section) and ultimately chose the culled CSM with the minimum BIC. This gave a culled CSM for each value of Λ; the final one we used was chosen after also minimizing BIC over Λ. The CSMs shown in section 3 result from this minimizing of BIC over Λ and state culling.

2.2.3 ISI Bootstrapping. While we do model selection with BIC, we also want to do model checking or adequacy testing. For the most part, we do this by using the CSM to bootstrap point-wise confidence bounds on the interspike interval (ISI) distribution and checking their coverage of the empirical ISI distribution. Because this distribution is not used by CSSR in reconstructing the CSM, it provides a check on the latter's ability to accurately describe the spike train's statistics.

Specifically, we generated confidence bounds as follows. To simulate one spike train, we picked a random starting state according to the CSM's inferred state occupation probabilities and then ran the CSM forward for N time steps, N being the length of the original spike train. This gives a binary time series, where a 1 stands for a spike and a 0 for no spike, and gave us a sample of interspike intervals from the CSM. This in turn gave an empirical ISI distribution. Repeated over 10^4 independent runs of the CSM, and taking the 0.005 and 0.995 quantiles of the distributions at each ISI length, gives 99% pointwise confidence bounds. (Pointwise bounds are necessary because the ISI distribution often modulates rapidly with ISI length.) If the CSM is correct, the empirical ISI will, by chance, lie outside the bounds at $\approx$1% of the ISI lengths.

If we split the data into training and validation sets, a CSM reconstructed from the training set can be used to bootstrap ISI confidence bounds, which can be compared to the ISI distribution of the test set. We discuss this sort of of cross-validation, as well as an additional test based on the time rescaling theorem, in appendix B.

2.3 Complexity and Algorithmic Information Content. The algorithmic information content $K(x_1^n)$ of a sequence x_1^n is the length of the shortest

complete (input-free) computer program that will output x_1^n exactly and then halt (Cover & Thomas, 1991).[10] In general, $K(x_1^n)$ is strictly uncomputable, but when x_1^n is the realization of a stochastic process X_1^n, the ensemble-averaged algorithmic information essentially coincides with the Shannon entropy (Brudno's theorem; see Badii & Politi, 1997), reflecting the fact that both are maximized for completely random sequences (Cover & Thomas, 1991). Both the algorithmic information and the Shannon entropy can be conveniently written in terms of a minimal sufficient statistic Q:

$$\begin{aligned}\mathbb{E}\left[K\left(X_1^n\right)\right] &= H\left[X_1^n\right] + o(n) \\ &= H[Q] + H\left[X_1^n \mid Q\right] + o(n). \end{aligned} \tag{2.8}$$

The equality $H[X_1^n] = H[Q] + H[X_1^n \mid Q]$ holds because Q is a function of X_1^n, so $H[Q \mid X_1^n] = 0$.

The key to determining a spike train's expected algorithmic information is thus to find a minimal sufficient statistic. By construction, causal state models provide exactly this: a minimal sufficient statistic for x_1^n is the state sequence $s_0^n = s_0, s_1, \ldots s_n$ (Shalizi & Crutchfield, 2001). Thus, the ensemble-averaged algorithmic information content, dropping terms $o(n)$ and smaller, is

$$\begin{aligned}\mathbb{E}\left[K\left(X_1^n\right)\right] &= H\left[S_0^n\right] + H\left[X_1^n \mid S_0^n\right] \\ &= H[S_0] + \sum_{i=1}^{n} H[S_i \mid S_{i-1}] + \sum_{i=1}^{n} H[X_i \mid S_i, S_{i-1}]. \end{aligned} \tag{2.9}$$

Going from the first to the second line uses the causal states' Markov property. Assuming stationarity, equation 2.9 becomes

$$\begin{aligned}\mathbb{E}\left[K\left(X_1^n\right)\right] &= H[S_t] + n\left(H[S_t \mid S_{t-1}] + H[X_t \mid S_t, S_{t-1}]\right) \\ &= C + n\left(J + R\right). \end{aligned} \tag{2.10}$$

This separates terms representing structure from those representing randomness:

The first term in equation 2.10 is the complexity, C, of the spike-generating process (Crutchfield & Young, 1989; Grassberger, 1986; Shalizi, Klinkner, & Haslinger, 2004).

$$C = H[S_t] = -\mathbb{E}[\log \mathbb{P}(S_t)]. \tag{2.11}$$

[10] The algorithmic information content is also called the Kolmogorov complexity. We do not use this term, to avoid confusion with our "complexity" C—the information needed to reproduce the spike train statistically rather than exactly (see equation 2.11). See Badii and Politi (1997) for a detailed comparison of complexity measures.

C is the entropy of the causal states, quantifying the structure present in the observed spikes. This is distinct from the entropy of the spikes themselves, which quantifies not their structure but their randomness (and is approximated by the other two terms). Intuitively, C is the (time-averaged) amount of information about the past of the system which is relevant to predicting its future. For example, consider again the i.i.d. 40 Hz Bernoulli process of Figure 1A. With $p = 0.04$, this has an entropy of 0.24 bits/msec, but because it can be described by a single state, the complexity is zero. (That state emits either a 0 or a 1, with respective probabilities 0.96 and 0.04, but either way, the state transitions back to itself.) In contrast, adding a 5 ms refractory period to the process means six states are needed to describe the spike trains (see Figure 1B). The new structure of the refractory period is quantified by the higher complexity, $C = 1.05$ bits.

The second and third terms in equation 2.10 describe randomness, but of distinct kinds. The second term, the internal entropy rate J, quantifies the randomness in the state transitions. It is the entropy of the next state given the current state:

$$J = H[S_{t+1} \mid S_t] = -\mathbb{E}[\log \mathbb{P}(S_{t+1} \mid S_t)]. \tag{2.12}$$

This is the average number of bits per time step needed to describe the sequence of states the process moved through (beyond those given by C). The last term in equation 2.10 accounts for any residual randomness in the spikes that is not captured by the state transitions:

$$R = H[X_{t+1} \mid S_t, S_{t+1}] = -\mathbb{E}[\log \mathbb{P}(X_{t+1} \mid S_t, S_{t+1})]. \tag{2.13}$$

For long trains, the entropy of the spikes, $H[X_1^n]$, is approximately the sum of these two terms, $H[X_1^n] \approx n(J + R)$. Computationally, C represents the fixed generating structure of the process, which needs to be described once, at the beginning of the time series, and $n(J + R)$ represents the growing list of details that pick out a particular time series from the ensemble that could be generated; this needs, on average, $J+R$ extra bits per time step. (Cf. the "sophistication" of Gács, Tromp, & Vitanyi, 2001.)

Consider again the 40 Hz Bernoulli process. As there is only one state, the process always stays in that state. Thus, the entropy of the next state, $J = 0$. However, the state sequence yields no information about the emitted symbols (the process is i.i.d.) so the residual randomness $R = 0.24$ bits/msec—as it must be, since the total entropy rate is 0.24 bits/msec. In contrast, the states of the 5 msec refractory process are informative about the process's future. The internal entropy rate $J = 0.20$ bits/msec and the residual randomness $R = 0$. All of the randomness is in the state transitions, because they uniquely define the output spike train. The randomness in the state transition is confined to state A, where the process "decides" whether it will stay in A, emitting no spike, or emit a spike and go to B. The decision

needs, or gives, 0.24 bits of information. The transitions from B through F and back to A are fixed and contribute 0 bits, reducing the expected J.

The important point is that the structure present in the refractory period makes the spike train less random, lowering its entropy. Averaged over time, the mean firing rate of the process is $p = 0.0333$. Were the spikes i.i.d., the entropy rate would be 0.21 bits/msec, but in fact $J + R = 0.20$ bits/msec. This is because a minimal description of a long sequence $X_{t_1}, \ldots, X_{t_N} = X_{t_1}^{t_N}$, the generating process needs to be only described once (C), while the internal entropy rate and randomness need to be updated at each time step ($n(J + R)$). Simply put, a complex, structured spike train can be exactly described in fewer bits than one that is entirely random. The CSM lets us calculate this reduction in algorithmic information and quantify the structure by means of the complexity.

2.4 Time-Varying Complexity and Entropies. The complexity and entropy are ensemble-averaged quantities. In the previous section, the ensemble was the entire time series, and the averaged complexity and entropies were analogous to a mean firing rate. The time-varying complexity and entropies are also of interest, for example, their variation after stimuli. A peristimulus time histogram (PSTH) shows how the firing probability varies with time; the same idea works for the complexity and entropy.

Since the states form a Markov chain, and any one spike train stays within a single ergodic component, we can invoke the ergodic theorem (Gray, 1988) and (almost surely) assert that

$$\sum_{S_t, S_{t+1}} \mathbb{P}(S_t, S_{t+1}, X_{t+1}) f(S_t, S_{t+1}, X_{t+1}) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} f(S_t, S_{t+1}, X_{t+1}) = \lim_{N\to\infty} \left\langle f(S_t, S_{t+1}, X_{t+1}) \right\rangle_N \quad (2.14)$$

for arbitrary integrable functions $f(S_t, S_{t+1}, X_{t+1})$.

In the case of the mean firing rate, the function to time average is $l(t) \equiv X_{t+1}$. For the time-averaged complexity, internal entropy, and residual randomness, the functions (respectively c, j, and r) are

$$\begin{aligned} c(t) &= -\log \mathbb{P}(S_t) \\ j(t) &= -\log \mathbb{P}(S_{t+1} \mid S_t) \\ r(t) &= -\log \mathbb{P}(X_{t+1} \mid S_t, S_{t+1}), \end{aligned} \quad (2.15)$$

and time-varying entropy $h(t) = j(t) + r(t)$.

The PSTH averages over an ensemble of stimulus presentations rather than time

$$\lambda_{PSTH}(t) = \frac{1}{M} \sum_{i=1}^{M} l_i(t) = \frac{1}{M} \sum_{i=1}^{M} X_{t+1,i}, \quad (2.16)$$

with M being the number of stimulus presentations and t reset to zero at each presentation. Analogously, the PSTH of the complexity is

$$C_{PSTH}(t) = \frac{1}{M}\sum_{i=1}^{M} c_i(t) = \frac{1}{M}\sum_{i=1}^{M} -\log \mathbb{P}(S_{t,i}). \tag{2.17}$$

For the entropies, replace c with j, r, or h as appropriate. Similar calculations can be made with any well-defined ensemble of reference times, not just stimulus presentations; we will also calculate c and the entropies as functions of the time since the latest spike.

We can estimate the error of these time-dependent quantities as the standard error of the mean as a function of time, $SE_t = s_t/\sqrt{M}$, where s_t is the sample standard deviation in each time bin t and M is the number of trials. The probabilities appearing in the definitions of $c(t)$, $j(t)$, $r(t)$ also have some estimation errors, either because of sampling noise or, more interesting, because the ensemble is being distorted by outside influences. The latter creates a gap between their averages (over time or stimuli) and what the CSM predicts for those averages. In the next section, we explain how to use this to measure the influence of external drivers.

2.5 The Influence of External Forces. If we know that $S_t = s$, the CSM predicts that the firing probability is $\lambda(t) = \mathbb{P}(X_{t+1} = 1 \mid S_t = s)$. By means of the CSM's recursive filtering property (see appendix A), once a transient regime has passed, the state is always known with certainty. Thereafter, the CSM predicts what the firing probability should be at all times, incorporating the effects of the spike train's history. As we show in the next section, these predictions give good matches to the actual response function in simulations where the spiking probability depends on only the spike history. But real neurons' spiking rates generally also depend on external processes (e.g., stimuli). As currently formulated, the CSM is (or, rather, converges on) the optimal predictor of the future of the process given its own past. Such an "output-only" model does not represent the (possible) effects of other processes, and so ignores external covariates and stimuli. Determining the precise form of spike trains' responses to external forces is best left to parametric models.

However, we can use output-only CSMs to learn something about the computation. The PSTH-calculated entropy rate $H_{PSTH}(t) = J_{PSTH}(t) + R_{PSTH}(t)$ quantifies the extent to which external processes drive the neuron. (The PSTH subscript is henceforce supressed.) Suppose we know the true firing probability $\lambda_{true}(t)$. At each time step, the CSM predicts the firing probability $\lambda_{CSM}(t)$. If $\lambda_{CSM}(t) = \lambda_{true}(t)$, then the CSM correctly describes the spiking, and the PSTH entropy rate is

$$H_{CSM}(t) = -\lambda_{CSM}(t)\log[\lambda_{CSM}(t)] - (1-\lambda_{CSM}(t))\log[1-\lambda_{CSM}(t)]. \tag{2.18}$$

However, if $\lambda_{CSM}(t) \neq \lambda_{true}(t)$, then the CSM misdescribes the spiking because it neglects the influence of external processes. Simply put, the CSM has no way of knowing when the stimuli happen. The PSTH entropy rate calculated using the CSM becomes

$$H_{CSM}(t) = -\lambda_{true}(t) \log [\lambda_{CSM}(t)] - (1 - \lambda_{true}(t)) \log [1 - \lambda_{CSM}(t)]. \tag{2.19}$$

Solving $\lambda_{true}(t)$,

$$\lambda_{true}(t) = \frac{H_{CSM}(t) + \log [1 - \lambda_{CSM}(t)]}{\log [1 - \lambda_{CSM}(t)] - \log [\lambda_{CSM}(t)]}. \tag{2.20}$$

The discrepancy between $\lambda_{CSM}(t)$ and $\lambda_{true}(t)$ indicates how much of the apparent randomness in the entropy rate is actually due to external driving. The true PSTH entropy rate $H_{true}(t)$ is

$$H_{true}(t) = -\lambda_{true}(t) \log [\lambda_{true}(t)] - (1 - \lambda_{true}(t)) \log [1 - \lambda_{true}(t)]. \tag{2.21}$$

The difference between $H_{CSM}(t)$ and $H_{true}(t)$ quantifies, in bits, the driving by external forces as a function of the time since stimulus presentation:

$$\begin{aligned} \Delta H &= H_{CSM}(t) - H_{true}(t) \\ &= \lambda_{true}(t) \log \left[\frac{\lambda_{true}(t)}{\lambda_{CSM}(t)} \right] + (1 - \lambda_{true}(t)) \log \left[\frac{1 - \lambda_{true}(t)}{1 - \lambda_{CSM}(t)} \right]. \end{aligned} \tag{2.22}$$

This stimulus-driven entropy ΔH is the relative entropy or Kullback-Leibler divergence $D(X_{true} \| X_{CSM})$ between the true distribution of symbol emissions and that predicted by the CSM. Information theoretically, this relative entropy is the error in our prediction of the next state due to assuming the neuron is running autonomously when it is actually externally driven. Since every state corresponds to a distinct distribution over future behavior, this is our error in predicting the future due to ignorance of the stimulus.[11]

3 Results

We now present a few examples. (All of them use a time step of 1 millisecond.) We begin with idealized model neurons to illustrate our technique. We recover CSMs for the model neurons using only the simulated spike trains

[11] Cf. the informational coherence introduced by Klinkner, Shalizi, and Camperi (2006) to measure information sharing between neurons by quantifying the error in predicting the distribution of the future of one neuron due to ignoring its coupling with another.

as input to our algorithms. From the CSM, we calculate the complexity, entropies, and, when appropriate, stimulus-driven entropy (Kullback-Leibler divergence between the true and CSM predicted firing probabilities) of each model neuron. We then analyze spikes recorded in vivo from a neuron in layer II/III of rat SI (barrel) cortex. We use spike trains recorded both with and without external stimulation of the rat's whiskers. (See Andermann & Moore, 2006, for experimental details.)

3.1 Model Neuron with a Soft Refractory Period and Bursting. We begin with a refractory, bursting model neuron whose spiking rate depends on only the time since the last spike. The baseline rate is 40 Hz. Every spike is followed by a 2 msec "hard" refractory period, during which spikes never occur. The spiking rate then rebounds to twice its baseline, to which it slowly decays. (See the dashed line in the first panel of Figure 3B.) This history dependence mimics that of a bursting neuron, and is intuitively more complex than the simple refractory period of the model in Figure 1.

Figure 2 shows the 17-state CSM reconstructed from a 200 second spike train (at 1 msec resolution) generated by this model. It has a complexity of $C = 3.16$ bits (higher than that of the model in Figure 1, as anticipated), an internal entropy rate of $J = 0.25$ bits/msec, and a residual randomness of $R = 0$ bits/msec. The CSM was obtained with $\Lambda = 17$ (selected by BIC). Figure 3A shows how the 99% ISI bounds bootstrapped from the CSM enclose the empirical ISI distribution, with the exception of one short segment.

The CSM is easily interpreted. State A is the baseline state. When it emits a spike, the CSM moves to state B. There are then two deterministic transitions, to C and then D, which never emit spikes; this is the hard 2 msec refractory period. Once in D, it is possible to spike again, and if that happens, the transition is back to state B. However, if no spike is emitted, the transition is to state E. This is repeated, with varying firing probabilities, as states E through Q are traversed. Eventually the process returns to A, and so to baseline.

Figure 3B plots the firing rate, complexity, and internal entropies as functions of the time since the last spike, conditional on no subsequent spike emission. This lets us compare the firing rate predicted by the CSM (solid line squares) to the specification of the model that generated the spike train (dashed line) and a PSTH calculated by triggering the last spike (solid line). Except at 16 and 17 msec postspike, the CSM-predicted firing rate agrees with both the generating model and the PSTH. The discrepancy arises because the CSM discerns only the structure in the data, and most of the ISIs are shorter than 16 msec. There is much closer agreement between the CSM and the PSTH if firing rates are plotted as a function of time since a spike without conditioning on no subsequent spike emission (not shown).

The middle and bottom panels of Figure 3 plot the time-dependent complexity and entropies. The complexity is much higher after the emission of a

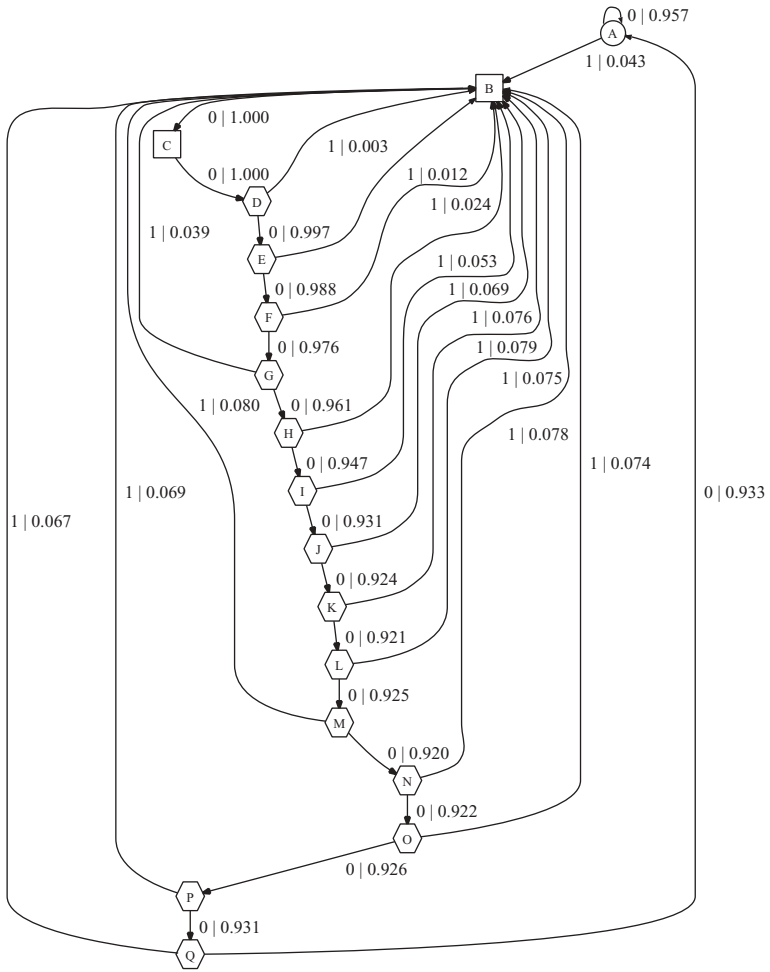

Figure 2: CSM reconstructed from a 200 sec simulated spike train with a soft refractory or bursting structure. $C = 3.16$, $J = 0.25$, $R = 0$. State A (circle) is the baseline 40 Hz spiking state. Upon emitting a spike, the transition is to state B. States B and C (squares) are "hard" refractory states from which no spike may be emitted. States D through Q (hexagons) compromise a refractory or bursting chain from which if a spike is emitted, the transition is back to state B. On exiting the chain, the CSM returns to the baseline state A.

spike than during baseline, because the states traversed (B–Q) are less probable and represent the additional structures of refractoriness and bursting. The time-dependent entropies (bottom panel) show that just after a spike, the refractory period imposes temporary determinism on the spike train,

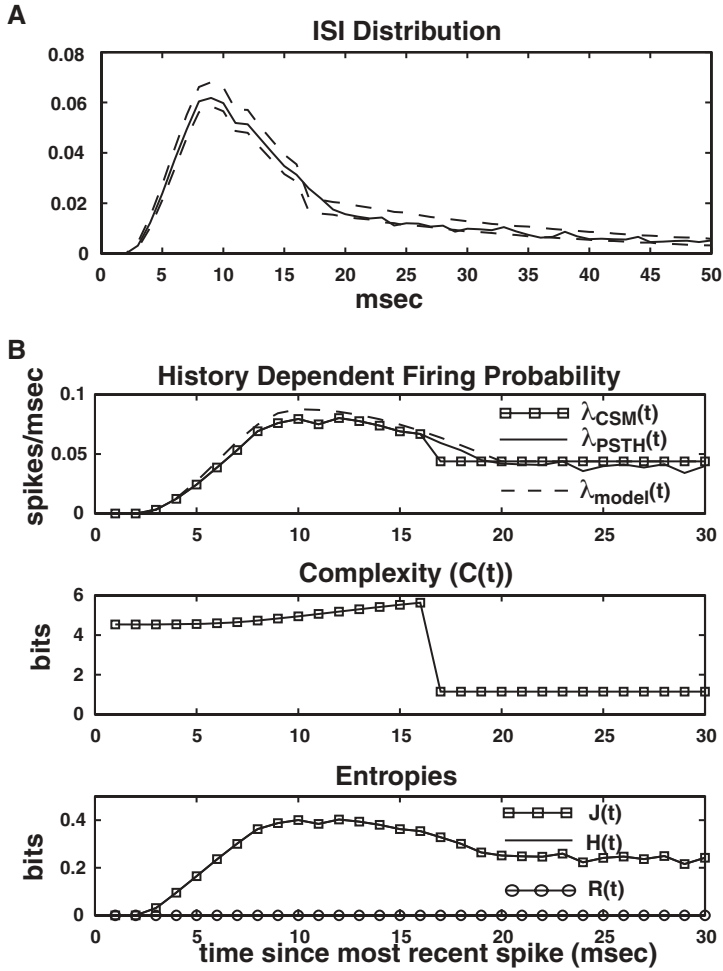

Figure 3: Soft refractory and bursting model ISI distribution and time-dependent firing probability, complexity, and entropies. (A) ISI distribution and 99% confidence bounds bootstrapped from the CSM. (B) Top panel: Firing probability as a function of time since the most recent spike. Line with squares = firing probability predicted by CSM. Solid line = firing probability deduced from PSTH. Dashed line = model firing rate used to generate spikes. Middle panel: Complexity as a function of time since the most recent spike. Bottom panel: Entropies as a function of time since the most recent spike. Squares = internal entropy rate circles = residual randomness, solid line = entropy rate (overlaps squares).

but burstiness increases the randomness before the dynamics return to the baseline state.

3.2 Model Neuron Under Periodic Stimulation. Figure 4 shows the CSM for a periodically stimulated model neuron. This CSM was reconstructed from 200 seconds of spikes with a baseline firing rate of 40 Hz ($p = 0.04$). Each second, the firing rate rose over the course of 5 msec to $p = 0.54$ spikes/msec, falling slowly back to baseline over the next 50 msec. This mimics the periodic presentation of a strong external stimulus. (The exact inhomogeneous firing rate used was $\lambda(t) = 0.93[e^{-t/10} - e^{-t/2}] + 0.04$ with t in msec. See Figure 5B, top panel, dashed line.) In this model, the firing rate does not directly depend on the spike train's history, but there is a sort of history dependence in the stimulus time course, and this is what CSSR discovers.

BIC selected $\Lambda = 7$, giving a 16-state CSM with $C = 0.89$ bits, $J = 0.27$ bits/msec, and $R = 0.0007$ bits/msec. The baseline is again state A, and if no spike is emitted, then the process stays in A. Spikes are either spontaneous and random or stimulus driven. Because the stimulus is external, it is not immediately clear which of these two causes produced a given spike. Thus, if a spike is emitted, the CSM traverses states B through F, deciding, so to speak, whether the spike is due to a stimulus. If two spikes happen within 3 msec of each other, the CSM decides that it is being stimulated and goes to one of states G, H, or M. States G through P represent the response to the stimulus. The CSM moves between these states until no spike is emitted for 3 msec, when it returns to the baseline, A.

The ISI distribution from the CSM matches that from the model (see Figure 5A). However, because the stimulus does not depend on the spike train's history, the CSM makes inaccurate predictions during stimulation. The top panel of Figure 5B plots the firing rate as a function of time since stimulus presentation, comparing the model (dashed line) and the PSTH (solid line) with the CSM's prediction (line with squares). The discrepancy between these is due to the CSM's having no way of knowing that an external stimulus has been applied until several spikes in a row have been emitted (represented by states B–F).[12] Despite this, $c(t)$ shows that something more complex than simple random firing is happening (see the middle panel of Figure 5B), as do $j(t)$ and $r(t)$ (see the bottom panel). Further, something is clearly wrong with the entropy rate, because it should be upper-bounded by $h = 1$ bit/msec (when $p = 0.5$). The fact that $h(t)$ exceeds this bound indicates that an external force, not fully captured by the CSM, is at work.

As discussed in section 2.5, drive from the stimulus can be quantified with a relative entropy (see Figure 5C). Stimuli are presented at $t = 1$ msec,

[12] In effect, this part of the CSM implements Bayes's rule, balancing the increased likelihood of a spike after a stimulus against the low a priori probability or base rate of stimulation.

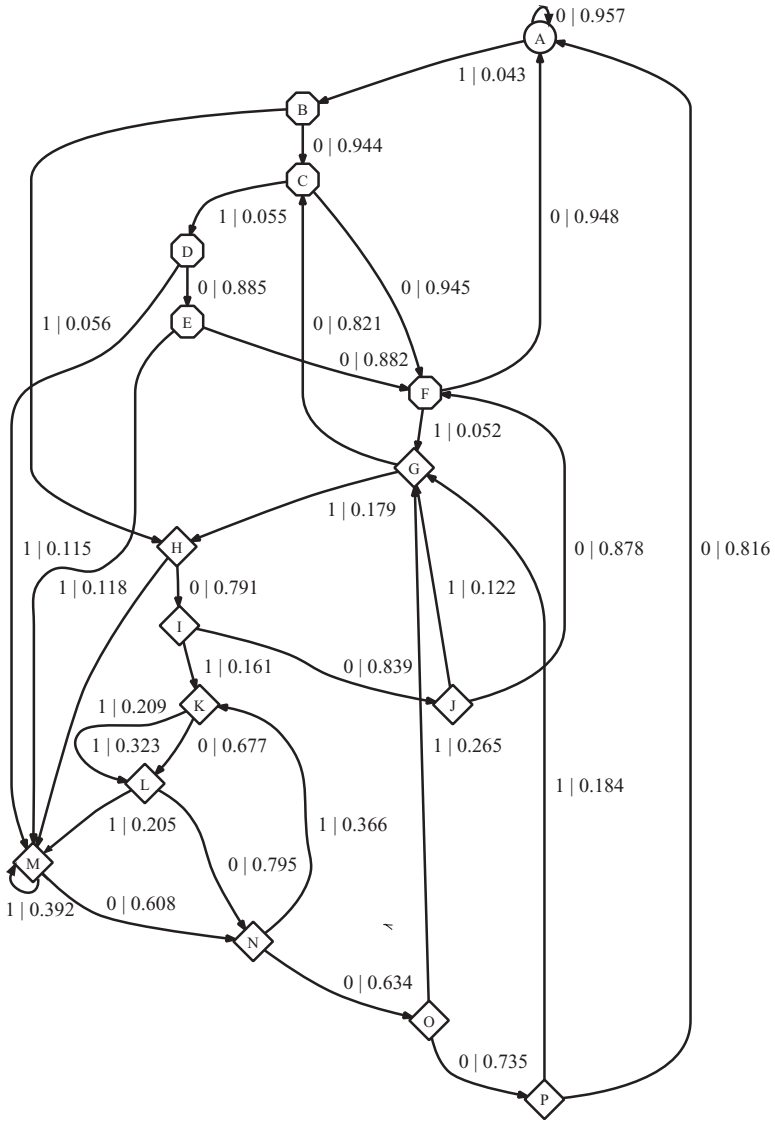

Figure 4: Sixteen-state CSM reconstructed from 200 sec of simulation of periodically stimulated spiking. $C = 0.89$, $J = 0.27$, $R = 0.0007$. State A is the baseline state. States B through F (octagons) are "decision" states in which the CSM evaluates whether a spike indicates a stimulus or was spontaneous. Two spikes within 3 msec cause the CSM to transition to states G through P, which represent the structure imposed by the stimulus. If no spikes are emitted within 5 (often fewer) sequential msec, the CSM goes back to the baseline state A.

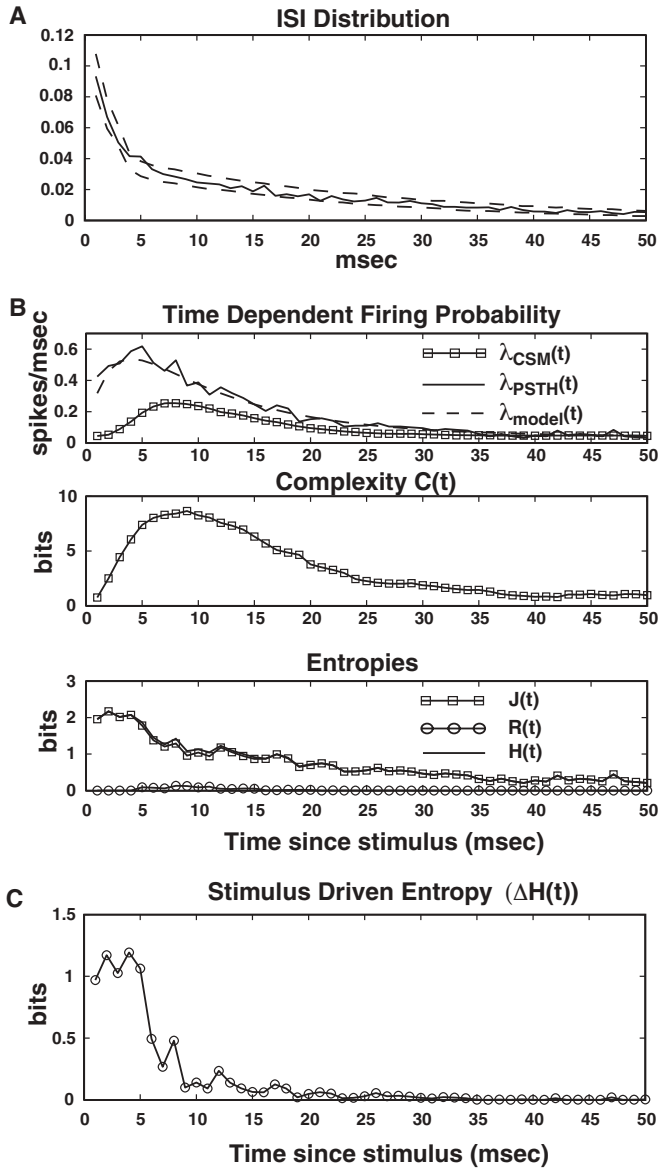

Figure 5: Stimulus model ISI distribution and time-dependent complexity and entropies. (A) ISI distribution and 99% confidence bounds. (B) Top panel: Firing probability as a function of time since the stimulus presentation. Middle panel: Time-dependent complexity. Bottom panel: Time-dependent entropies. (C) The stimulus-driven entropy is more than 1 bit, indicating a strong external drive. See text for discussion.

where $\Delta H(t) > 1$ bit. It is not until ≈25 msec poststimulus that $\Delta H(t) \approx 0$ and the CSM once again correctly describes the internal entropy rate. Thus, as expected, the stimulus strongly influences neuronal dynamics immediately after its presentation. The true internal entropy rate $H_{true}(t)$ is slightly less than 1 bit/msec shortly after stimulation, when the true spiking rate has a maximum of $p_{\max} = 0.54$. The fact that the CSM gives an inaccurate value for J actually lets us find the number of bits of information gain supplied by the stimulus, for example, $\Delta H > 1$ bit, immediately after the stimulus is presented.

3.3 Spontaneously Spiking Barrel Cortex Neuron. We reconstructed a CSM from 90 seconds of spontaneous (no vibrissa deflection) spiking recorded from a layer II/III FSU barrel cortex neuron. CSSR, using $\Lambda = 21$, discovered a CSM with 315 states, a complexity of $C = 1.78$ bits, and an internal entropy rate of $J = 0.013$ bits/msec. After state culling (see section 2.7.2), the reduced CSM, plotted in Figure 6, has 14 states, $C = 1.02$, $J = 0.10$ bits/msec, and residual randomness of $R = 0.005$ bits/msec. We focus on the reduced CSM from this point onward.

This CSM resembles that of the spontaneously firing model neuron of section 3.1 and Figure 2. The complexity and entropies are lower than those of our model neuron because the mean spike rate is much lower, and so simple descriptions suffice most of the time. (Barrel cortex neurons exhibit notoriously low spike rates, especially during anesthesia.) There is a baseline state A that emits a spike with probability $p = 0.01$ (i.e., 10 Hz). When a spike is emitted, the CSM moves to state B and then on through the chain of states C through N, returning to A if no spike is subsequently emitted. However, the CSM can emit a second or even third spike after the first, and indeed this neuron displays spike doublets and triplets. In general, emitting a spike moves the CSM to B, with some exceptions that show the structure to be more intricate than the model neuron's.

Figure 7A shows the CSM's 99% confidence bounds almost completely enclosing the empirical ISI distribution. The top panel of Figure 7B plots the history-dependent firing probability predicted by the CSM as a function of the time since the latest spike, according to both the PSTH and the CSM's prediction. They are highly similar in the first 13 msec postspike, indicating that the CSM gets the spiking statistics right in this epoch. The CSM and PSTH then diverge after this, for two reasons. First, as with the model neuron, there are few ISIs of this length. Most of the ISIs are either shorter, due to the nueron's burstiness, or much longer, due to the low baseline firing rate. Second, 90 seconds does not provide many data. We show in Figure 10 that a CSM reconstructed from a longer spike train does capture all of the structure. We present the results of this shorter spike train to emphasize that as a nonparametric method, CSSR uncovers the statistical structure only in the data—no more, no less.

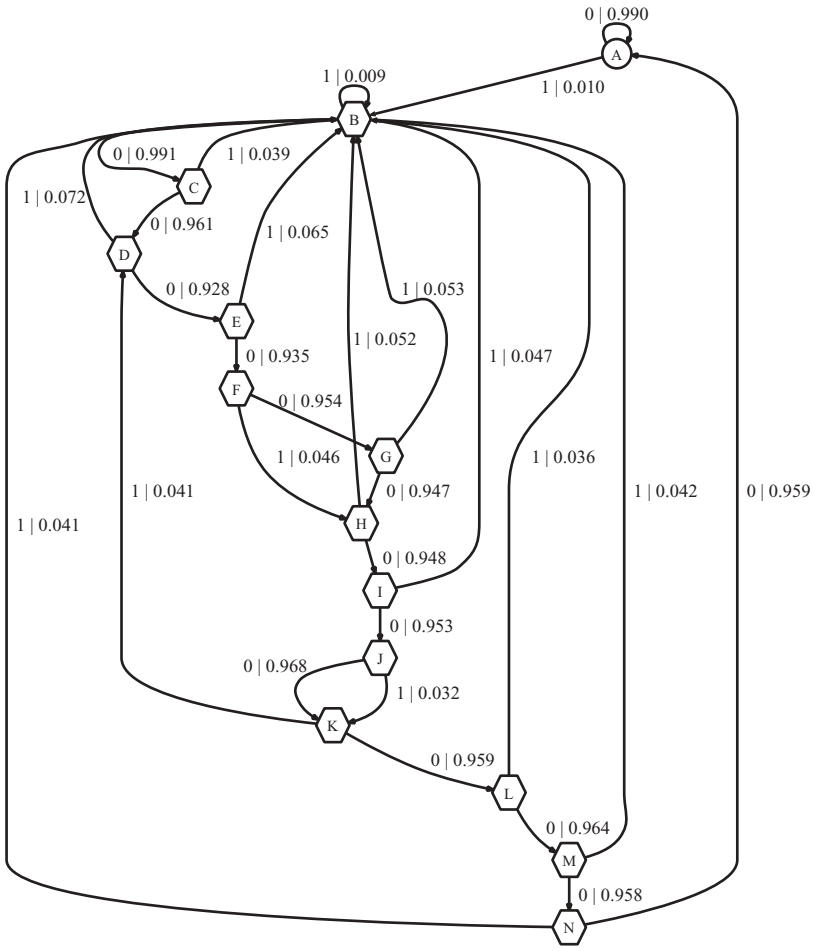

Figure 6: Fourteen-state CSM reconstructed from 90 sec of spiking recorded from a spontaneously spiking (no stimulus) neuron located in layer II/III of rat barrel cortex. $C = 1.02$, $J = 0.10$, $R = 0.005$. State A (circle) is baseline 10 Hz spiking. States B through N comprise a refractory or bursting chain similar to but with a somewhat more intricate structure than that of the model neuron in Figure 2.

Finally, the middle and bottom panels of Figure 6B show, respectively, the complexity and entropies as functions of the time since the latest spike. As with the model of section 3.1, the structure in the process occurs after spiking, during the refractory and bursting periods. This is when the complexity is largest and also when the entropies vary most.

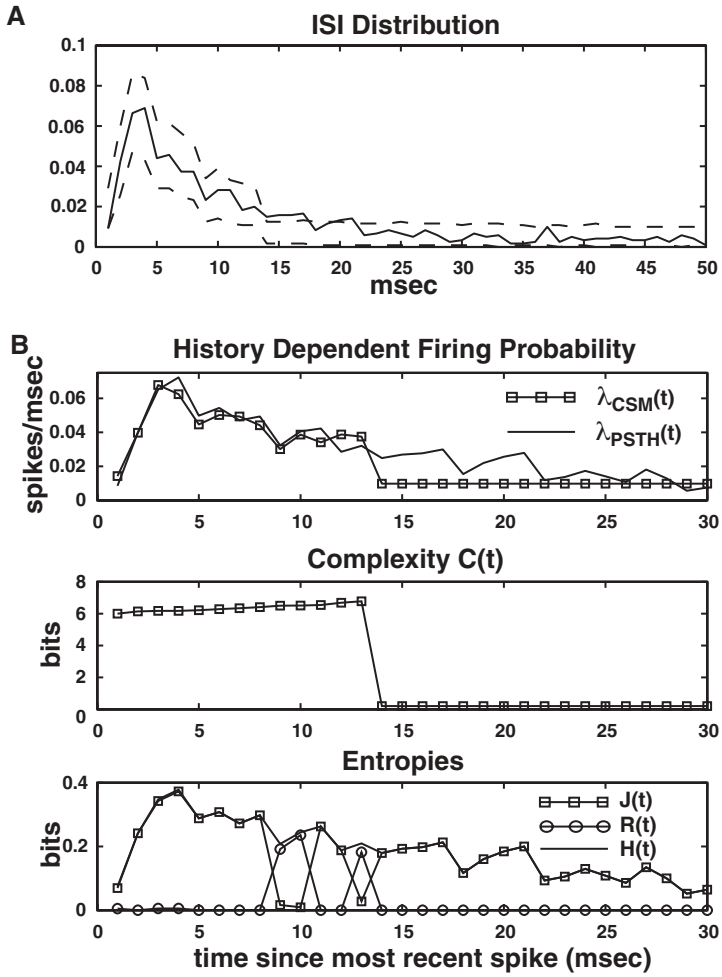

Figure 7: Spontaneously spiking barrel cortex neuron. (A) ISI distribution and 99% bootstrapped confidence bounds. (B) Top panel: Time-dependent firing probability as a function of time since the most recent spike. See the text for an explanation of the discrepancy between CSM and PSTH spike probabilities. Middle panel: Complexity as a function of time since the most recent spike. Bottom panel: Entropy rates as a function of time since the most recent spike.

3.4 Periodically Stimulated Barrel Cortex Neuron. We reconstructed CSMs from 335 seconds of spike trains taken from the same neuron used above, but recorded while it was being periodically stimulated by vibrissa deflection. BIC selected $\Lambda = 25$, giving the 29-state CSM shown in Figure 8. (Before state culling, the original CSM had 1916 states, $C = 2.55$

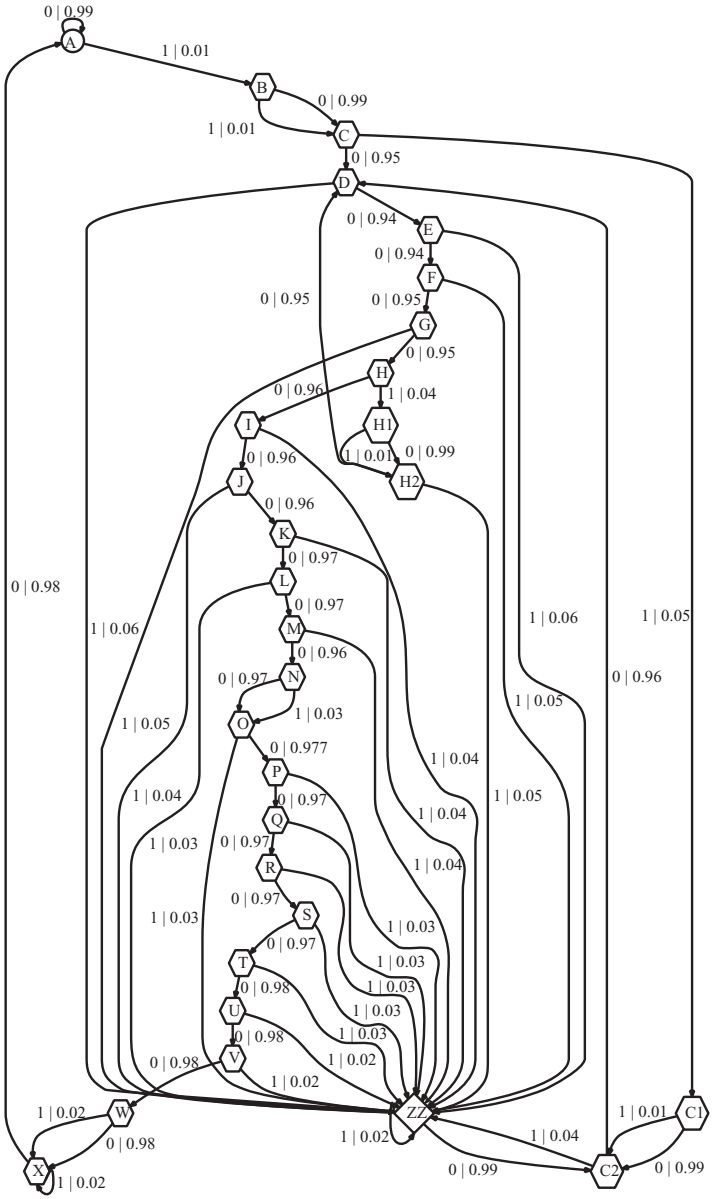

Figure 8: Twenty-nine-state CSM reconstructed from 335 seconds of spikes recorded from a layer II/III barrel cortex neuron undergoing periodic (125 msec interstimulus interval) stimulation via vibrissa deflection. $C = 1.97$, $J = 0.11$, $R = 0.004$. Most of the states are devoted to refractory or bursting behavior; however states C1, C2, and ZZ represent the structure imposed by the external stimulus. See the text for discussion.

and $J = 0.11$.) The reduced CSM has a complexity of $C = 1.97$ bits, an internal entropy rate of $J = 0.10$ bits/msec, and a residual randomness of $R = 0.005$ bits/msec. Note that C is higher when the neuron is being stimulated as opposed to when it is spontaneously firing, indicating more structure in the spike train.

While at first the CSM may seem to represent only history-dependent refractoriness and bursting, ignoring the external stimulus, this is not quite true. Once again, there is a baseline state A, and most of the other states (B–X) comprise a refractory/bursting chain, like this neuron has during spontaneous firing. However, the transition on A emitting a spike is not back to B and then down the chain again, but to either state C_1, and subsequently C_2, or, more often, to state ZZ. These three states represent the structure induced by the external stimulus, as we saw with the model-stimulated neuron of section 3.2 and Figure 4. (The state ZZ is comparable to the state M of the model-stimulated neuron: both loop back to themselves if they emit a spike.) Three states are enough because in this experiment, barrel cortex neurons spike extremely sparsely—0.1 to 0.2 spikes per stimulus presentation.

Figure 9A plots the ISI distribution, nicely enclosed by the bootstrapped confidence bounds. Figure 9B shows the firing rate, complexity, and entropies as functions of the time since stimulus presentation (averaged over all presentations). These plots look much like those in Figure 7B. However, there is a clear indication that something more complex takes place after stimulation: the CSM's firing rate predictions are wrong. The stimulus-driven entropy ΔH turns out to be as large as 0.02 bits within 5 to 15 msec poststimulus. This agrees with the known $\approx$5 to 10 msec stimulus propagation time between vibrissae and barrel cortex (Andermann & Moore, 2006). The reason that ΔH is so much smaller for the real neuron than the stimulated model neuron of section 3.2 is that the former's firing rate is much lower. Although the firing rate poststimulus can be almost twice as large as the CSM's prediction, the actual rate is still low: $\max \lambda(t) \approx 0.04$ spikes/msec. Most of the time the neuron does not spike, even when stimulated, so on average, the stimulus provides little information per presentation. For completeness, Figure 10 shows the spike probability, complexity, and entropies as functions of the time since the latest spike. Averaged over this ensemble, the CSM's predictions are highly accurate.

4 Discussion

The goal of this letter was to present methods for determining the structural content of spike trains while making minimal a priori assumptions as to the form that structure takes. We use the CSSR algorithm to build minimal, optimally predictive hidden Markov models (CSMs) from spike trains, Schwartz's Bayesian information criterion to find the optimal history length Λ of the CSSR algorithm, and bootstrapped confidence bounds

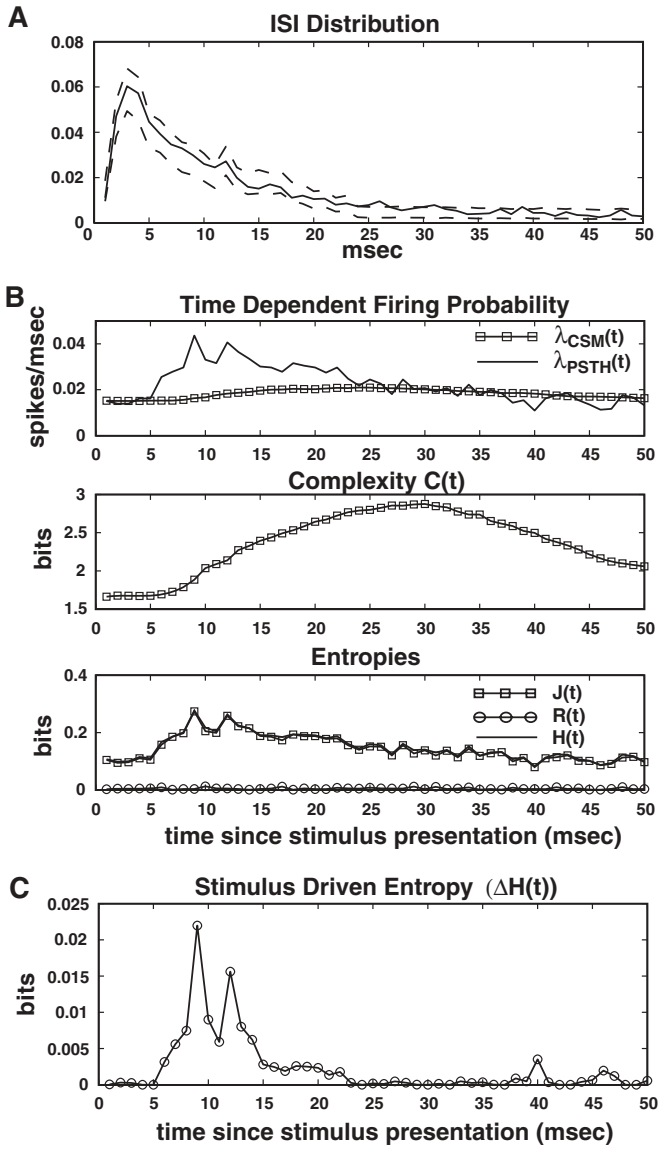

Figure 9: Stimulated barrel cortex neuron ISI distribution and time-dependent complexity and entropies. (A) ISI distribution and 99% confidence bounds. (B) Top panel: Firing probability as a function of time since stimulus presentation. Middle panel: Time-dependent complexity. Bottom panel: Time-dependent entropies. (C) The stimulus-driven entropy (maximum of 0.02 bits/msec) is low because the number of spikes per stimulus ($\approx$0.1–0.2) is very low, and hence the stimulus does not supply much information.

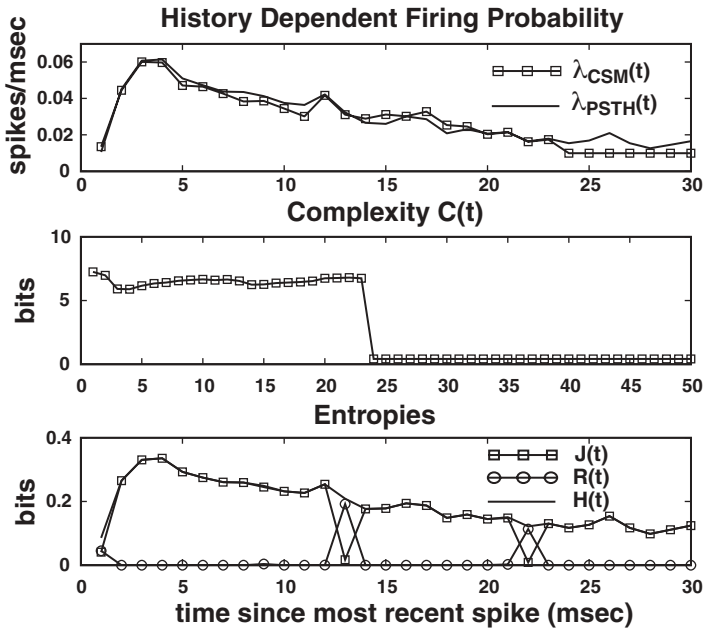

Figure 10: Firing probability complexity, and entropies of the stimulated barrel cortex neuron as a function of time since the most recent spike.

on the ISI distribution from the CSM to check goodness of fit. We demonstrated how CSMs can estimate a spike train's complexity, thus quantifying its structure, and its mean algorithmic information content, quantifying the minimal computation necessary to generate the spike train. Finally we showed how to quantify, in bits, the influence of external stimuli on the spike-generating process. We applied these methods to both simulated spike trains, for which the resulting CSMs agreed with intuition, and real spike trains recorded from a layer II/II rat barrel cortex neuron, demonstrating increased structure, as measured by the complexity, when the neuron was being stimulated.

We are unaware of any other practical techniques for quantifying the complexity and computational structure of a spike train as we define them. Intuitively, neither random (Poisson) nor highly ordered (e.g., strictly periodic, as in Olufsen, Whittington, Camperi, & Kopell, 2003). spike trains should be thought of as complex since they do not possess a structure requiring a sophisticated program to generate. Instead, complexity lies between order and disorder (Badii & Politi, 1997), in the nonrandom variation of the spikes. Higher complexity means a greater degree of organization in neural activity than would be implied by random spiking. It is the reconstruction of the CSM through CSSR that allows us to calculate the complexity.

Our definition of complexity stands in stark contrast to other complexity measures, which assign high values to highly disordered systems. Some of these, such as Lempel Ziv complexity (Amigo, Szczepanski, Wajnryb, & Sanchez-Vives, 2002, 2004; Jimenez-Montano, Ebeling, Pohl, & Rapp, 2002; Szczepanski, Amigo, Wajnryb, & Sanchez-Vives, 2004) and context-free grammar complexity (Rapp et al., 1994) have been applied to spike trains. However, both of these are measures of the amount of information required to reproduce the spike train exactly and take on very high values for completely random sequences. These complexity measures are therefore much more similar to total algorithmic information content and even to the entropy rate than to our sort of complexity.

Our measure of complexity is the entropy of the distribution of causal states. This has the desired property of being maximized for structured rather than ordered or disordered systems, because the causal states are defined statistically as equivalence classes of histories conditioned on future events. Other researchers have also calculated complexity measures that are entropies of state distributions but have defined their states differently. Amigo et al. (2002) uses the observables (symbol strings) present in the spike train to define a kth-order Markov process and calls each individual length k string that appears in the spike train a state. Gorse and Taylor (1990) similarly use single-suffix symbol strings to define the states of a Markov process. In both cases, i.i.d. Bernoulli sequences could exhibit up to 2^k states (in long enough sequences) and possess an extremely high "complexity." However, all of these states make the same prediction for the future of the process. The minimal representation is a single causal state—a CSM with a complexity of zero.

There are also many works that model spike trains using HMMs, but in which the hidden states represent macrostates of the system (e.g., awake/asleep, up/down), and spiking rates are modeled separately in each macrostate (Abeles et al., 1995; Achtman et al., 2007; Chen, Vijayan, Barbieri, Wilson, & Brown, 2008; Danoczy & Hahnloser, 2004; Jones, Fontanini, Sadacca, & Katz, 2007). Although the graphical representation of such HMMs may look like those of CSMs, the two kinds of states have very different meanings. Finally, there are also state-space methods that model the dynamical state of the system as a continuous hidden variable, the most well known of which is the linear gaussian model with Kalman filtering. These have been extensively applied to neural encoding and decoding problems (Eden, Frank, Barbieri, Solo, & Brown, 2004; Smith et al., 2004; Srinivasan, Eden, Mitter, & Brown, 2007). Interestingly, for a univariate gaussian ARMA model in state-space form, the Kalman filter's one-step-ahead prediction and mean-squared prediction error are, jointly, minimally sufficient for next-step prediction, and since they can be updated recursively, they in fact constitute the minimal sufficient statistic, and hence the causal state in this special case.

Neurons are driven by their afferent synapses. Although as discussed in appendix C, there is a parallel "transducer" formalism for generating CSMs that takes external influences into account, this is not yet computationally implemented, and our current approach reconstructs CSMs only from the spike train. Since the history of the neuron under study is typically connected with the history of the network in which it is located, this CSM will, in general, reflect more than a neuron's internal biophysical properties. Nonetheless, in both our model neurons and in the real barrel cortex neuron, states not interpretable as simple refractoriness or bursting appeared when a stimulus was present, proving we can detect stimulus-driven complexity. Further, we showed that the CSM can be used to determine the extent (in bits) to which a neuron is driven by external stimuli.

The methods presented here complement more established modes of spike train analysis, which have different goals. Parametric methods, such as PSTHs or maximum likelihood estimation (Brown, Kass, & Mitra, 2004; Truccolo, Eden, Fellow, Donoghue, & Brown, 2005), generally focus on determining a neuron's firing rate (mean, instantaneous, or history dependent) and on how known external covariates modulate that rate. They have the advantage of requiring fewer data than nonparametric methods such as CSSR but the disadvantage, for our purposes, of imposing the structure of the model at the outset. When the experimenter wants to know how a neuron encodes a particular aspect of a covariate (e.g., how neurons in the sensory periphery or primary sensory cortices encode stimuli), parametric methods have proved highly illuminating. However, in many cases, the identity or even existence of relevant external covariates is uncertain. For example, one could envision using CSMs to analyze recordings in prefrontal cortex during different cognitive tasks or perhaps compare spiking structures during different attentional states. In both cases, the relevant external covariates are not at all clear, but CSMs could still be used to quantify changes in computational structure for single neurons or groups of them. For neural populations, one can envision generating distributions (over the population) of complexities and examining how these distributions change in different cortical macrostates. This would be entirely analogous to analyzing distributions of firing rates or tuning curves.

In addition to calculations of the complexity, the whole array of mutual information analyses can be applied to CSMs, but instead of calculating mutual information between the spikes and the covariates (which could include other spike trains), one can calculate the mutual information between the covariates and the causal states. The advantage is that the causal states represent the behavioral patterns of the spike-generating process, and so are closer to the actual state of the system than the spikes (output observables) are themselves. Results on calculating the mutual information between the causal states of different neurons (informational coherence) in a large simulated network show that synchronous neuronal dynamics

are more effectively revealed than when calculated directly from the spikes (Klinkner et al., 2006).

Our methods provide a way to understand structure in spike trains and should be considered as complements to traditional analysis methods. We rigorously define structure and show how to discover it from the data themselves. Our methods go beyond those that seek to describe the observed variation in the spiking rates by also describing the underlying computational process (in the form of a CSM) needed to generate that variation. A CSM can not only that the spike rate has changed, but also exactly how it has changed.

Appendix A: Filtering with CSMs

A common difficulty with hidden Markov models is that predictions can be made only from a knowledge of the state, which must itself be guessed at from the time series, since it is, after all, hidden. This creates the state estimation or filtering problem. Under strong assumptions (linear gaussian stochastic dynamics, linearly observed through i.i.d. additive gaussian noise), the Kalman filter is an optimal yet tractable solution. For nonlinear processes, however, optimal filtering essentially amounts to maintaining a posterior distribution over the states and updating it by Bayes's rule (Ahmed, 1998). (This distribution is sometimes called the process's information state.)

One convenient and important feature of CSMs is that this whole machinery of filtering is unnecessary because of their recursive-updating property. Given the state at time t, S_t, and the observation at time $t+1$, X_{t+1}, the state at time $t+1$ is fixed, $S_{t+1} = T(S_t, X_{t+1})$, for some transition function T. Clearly if the state is known with certainty at any time, it will remain known. However, the same recursive updating property also allows us to show that the state does become certain; after some finite (but possibly random) time τ, $\mathbb{P}(S_\tau = s \mid X_1^\tau)$ is either 0 or 1 for all states s. For Markov chains of order k, clearly $\tau \leq k$; under more general circumstances, $\mathbb{P}(\tau \geq t)$ goes to zero exponentially or faster.

Thus, after a transient period, the state is completely unambiguous. This will be useful to us in multiple places, including understanding the computational structure of the process and predicting the firing rate of the neuron. It also leads to considerable numerical simplifications, compared to approaches that demand conventional filtering. Further, recursive filtering is easily applied to a new spike train—not merely the one from which the CSM was reconstructed. This helps in cross-validating CSMs, as discussed in appendix B.

Appendix B: Cross-Validation

It is often desirable to cross-validate a statistical model by spliting one's data set in two, using one part (generally the larger) as a training set for

the model and the other part to validate the model by some statistical test. In the case of CSMs, it is particularly important to check the validity of the BIC used to regularize the Λ control setting.

One possible test is the ISI bootstrapping of section 2.3. A second, somewhat stronger, goodness-of-fit test is based on the time rescaling theorem of Brown, Barbieri, Ventura, Kass, and Frank (2002). This test rescales the interspike intervals as a function of the integrated history-dependent spiking rate over the ISI,

$$\tau_k = 1 - e^{-\int_{t_k}^{t_{k+1}} \lambda(t)dt}, \tag{B.1}$$

where the $\{t_k\}$ are the spike times and $\lambda(t)$ is the history-dependent spiking rate from the CSM. If the CSM describes the data well, then rescaled ISI's $\{\tau_k\}$ should follow a uniform distribution. This can be tested using either a Kolmogorov-Smirnov test or by plotting the empirical CDF of the rescaled times against the CDF of the uniform distribution (Kolmogorov-Smirnov or KS plot) (Brown et al., 2002).

Figure 11 gives cross-validation results for the rat barrel cortex neuron during both spontaneous firing and periodic vibrissae deflection. Ninety seconds of spontaneously firing spikes were split into a 75 second training set and a 15 second validation set. The 335 seconds of stimulus-evoked firing were split into a 270 second training set and a 65 second validation set. Figures 11A and 11B show the ISI bootstrapping results for the spontaneous and stimulus-evoked firing, respectively. The dashed lines are 99% confidence bounds from a CSM reconstructed from the training set, and the solid line is the ISI distribution of the validation set. The ISI distribution largely falls within these bounds for both the spontaneous and stimulus-evoked data.

Figures 11C through 11F display the time-rescaling test. Figures 11C and 11D show the time-rescaling plots for the spontaneous and stimulus-evoked training data, respectively. The dashed lines are 95% confidence bounds. The spontaneous KS plot largely falls within the bounds. The stimulus evoked does not, but this is expected because the CSM does not completely capture the imposition of the external stimulus. (The jagged "steps" in both plots result from the 1 msec temporal discretization.) Figures 11E and 11F show the time-rescaling plots for, respectively, the spontaneous and stimulus-evoked validation data. The fits here are somewhat worse. In the stimulated case, this is not surprising. In the spontaneous case, the cause is likely nonstationarity in the data, a problem shared with other spike train analysis techniques, such as the generalized linear model approaches described in appendix C. It should be emphasized that the point of reconstructing CSMs is not to obtain perfect fits to the data, but instead to estimate the structure inherent in the spike train; the cross-validation results should be viewed in this light.

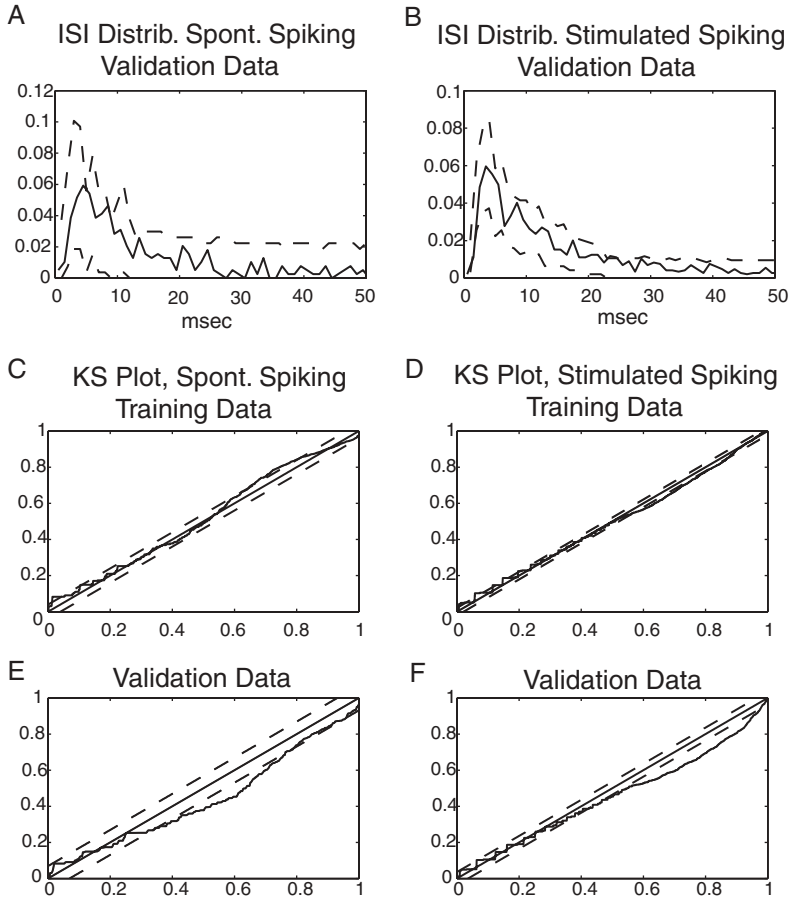

Figure 11: Cross-validation of CSMs reconstructed from spontaneously firing and stimulus evoked rat barrel cortex on an independent validation training set. (A,B) ISI distribution of spontaneously and stimulus-evoked firing validation sets and 99% confidence bounds bootstrapped from CSM. (C–D) Time-rescaling plots of training data sets for spontaneously firing and stimulus-evoked firing, respectively. Dashed lines are 95% confidence bounds, and the solid line is the rescaled ISIs. The solid line along the digagonal is for visual comparison to an ideal fit. (E–F) Similar time-rescaling plots for the validation data sets.

Appendix C: Causal State Transducers and Predictive State Representations

Mathematically, CSMs can be expanded to include the influence of external stimuli on the process, yielding causal state transducers, which are optimal

representations of the history-dependent mapping from inputs to outputs (Shalizi, 2001). Such causal state transducers are a type of partially observable Markov decision process, closely related to predictive state representations (PSRs) (Littman et al., 2002). In both formalisms, the right notion of "state" is a statistic, a measurable function of the observable past of the process. Causal states represent this through an equivalence relation on the space of observable histories. For PSRs, the representation is through "tests"—a distinguished set of input-output sequence pairs. The idea is that states can be uniquely characterized by their probabilities of producing the output sequences conditional on the input sequences.

An algorithm for reconstructing causal state transducers would begin by estimating probability distributions of future histories conditioned on both the history of the spikes and the history of an external covariate Y, for example, $\mathbb{P}(X_{t+1}^{\infty} \mid X_{-\infty}^{t}, Y_{-\infty}^{t})$, and otherwise be entirely parallel to CSSR. This has not yet been implemented.

Acknowledgments

We thank Mark Andermann and Christopher Moore for the use of their data. R.H. thanks Emery Brown, Anna Dreyer, and Christopher Moore for valuable discussions. C.R.S. thanks Anthony Brockwell, Dave Feldman, Chris Genovese, Rob Kass, and Alessandro Rinaldo for valuable discussions.

References

Abeles, M., Bergman, H., Gat, I., Meilijson, I., Seidemann, E., Tishby, N., et al. (1995). Cortical activity flips among quasi-stationary states. *Proc. Natl. Acad. Sci. USA.*, *92*, 8616–8620.

Achtman, N., Afshar, A., Santhanam, G., Yu, B. M., Ryu, S. I., & Shenoy, K. V. (2007). Free paced high-performance brain-computer interfaces. *Journal of Neural Engineering*, *4*, 336–347.

Ahmed, N. U. (1998). *Linear and nonlinear filtering for scientists and engineers.* Singapore: World Scientific.

Amigo, J. M., Szczepanski, J., Wajnryb, E., & Sanchez-Vives, M. V. (2002). On the number of states of the neuronal sources. *Biosystems*, *68*, 57–66.

Amigo, J. M., Szczepanski, J., Wajnryb, E., & Sanchez-Vives, M. V. (2004). Estimating the entropy rate of spike trains via Lempel-Ziv complexity. *Neural Computation*, *16*, 717–736.

Andermann, M. L., & Moore, C. I. (2006). A sub-columnar direction map in rat barrel cortex. *Nature Neuroscience*, *9*, 543–551.

Badii, R., & Politi, A. (1997). *Complexity: Hierarchical structures and scaling in physics*. Cambridge: Cambridge University Press.

Bernardo, J. M., & Smith, A. F. M. (1994). *Bayesian theory*. New York: Wiley.

Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., & Frank, L. M. (2002). The time rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, *14*, 325–346.

Brown, E. N., Kass, R. E., & Mitra, P. P. (2004). Multiple neural spike train data analysis: State-of-the-art and future challenges. *Nature Neuroscience*, *7*, 456–461.

Chen, Z., Vijayan, S., Barbieri, R., Wilson, M. A., & Brown, E. N. (2008). Discrete- and continuous-time probablistic models and inference algorithms for neuronal decoding of Up and Down states. *Neural Computation*, *21*, 1797–1862.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Crutchfield, J. P., & Young, K. (1989). Inferring statistical complexity. *Physical Review Letters*, *63*, 105–108.

Csiszár, I., & Talata, Z. (2006). Context tree estimation for not necessarily finite memory processes, via BIC and MDL. *IEEE Transactions on Information Theory*, *52*, 1007–1016.

Danoczy, M. G., & Hahnloser, R. H. R. (2004). Efficient estimation of hidden state dynamics. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems*, (pp. 227–234). Cambridge, MA: MIT Press.

Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., & Brown, E. N. (2004). Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Computation*, *16*, 971–998.

Gács, P., Tromp, J. T., & Vitanyi, P. M. B. (2001). Algorithmic statistics. *IEEE Transactions on Information Theory*, *47*, 2443–2463.

Gorse, D., & Taylor, J. G. (1990). A general model of stochastic neural processing. *Biological Cybernetics*, *63*, 299–306.

Grassberger, P. (1986). Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics*, *25*, 907–938.

Gray, R. M. (1988). *Probability, random processes, and ergodic properties*. New York: Springer-Verlag.

Jaeger, H. (2000). Observable operator models for discrete stochastic time series. *Neural Computation*, *12*, 1371–1398.

Jimenez-Montano, M. A., Ebeling, W., Pohl, T., & Rapp, P. E. (2002). Entropy and complexity of finite sequences and fluctuating quantities. *Biosystems*, *64*, 23–32.

Jones, L. M., Fontanini, A., Sadacca, B. F., & Katz, D. B. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proc. Natl. Acad. Sci. USA*, *104*, 18772–18777.

Kennel, M. B., & Mees, A. I. (2002). Context-tree modeling of observed symbolic dynamics. *Physical Review E*, *66*, 056209.

Klinkner, K. L., Rinaldo, A., & Shalizi, C. R. (2009). *Adaptive nonparametric prediction and bootstrapping of discrete time series.* Unpublished manuscript.

Klinkner, K. L., & Shalizi, C. R. (2009). *CSSR: A nonparametric algorithm for predicting and classifying time series*. Unpublished manuscript.

Klinkner, K. L., Shalizi, C. R., & Camperi, M. F. (2006). Measuring shared information and coordinated activity in neuronal networks. In Y. Weiss, B. Schölkopf, & J. C. Platt (Eds.), *Advances in neural information processing systems, 18*, (pp. 667–674). Cambridge, MA: MIT Press.

Knight, F. B. (1975). A predictive view of continuous time processes. *Annals of Probability*, *3*, 573–596.

Littman, M. L., Sutton, R. S., & Singh, S. (2002). Predictive representations of state. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14* (pp. 1555–1561). Cambridge, MA: MIT Press.

Löhr, W., & Ay, N. (2009). On the generative nature of prediction. *Advances in Complex Systems*, *12*, 169–194.

Marton, K., & Shields, P. C. (1994). Entropy and the consistent estimation of joint distributions. *Annals of Probability, 22*, 960–977. Correction, *Annals of Probability*, *24* (1996), 541–545.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

Olufsen, M. S., Whittington, M. A., Camperi, M., & Kopell, N. (2003). New roles for the gamma rhythm: Population tuning and processing for the beta rhythm. *Journal of Computation Neuroscience*, *14*, 33–54.

Rapp, P. E., Zimmerman, I. D., Vining, E. P., Cohen, N., Albano, A. M., & Jimenez-Montano, M. A. (1994). The algorithmic complexity of neural spike trains increases during focal seizures. *Journal of Neuroscience*, *14*, 4731–4739.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1997). *Spikes: Exploring the neural code.* Cambridge, MA: MIT Press.

Shalizi, C. R. (2001). *Causal architecture, complexity and self-organization in time series and cellular automata*. Unpublished doctoral dissertation, University of Wisconsin.

Shalizi, C. R., & Crutchfield, J. P. (2001). Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, *104*, 817–879.

Shalizi, C. R., & Klinkner, K. L. (2004). Blind construction of optimal nonlinear recursive predictors for discrete sequences. In M. Chickering & J. Y. Halpern (Eds.), *Uncertainty in artificial intelligence: Proceedings of the Twentieth Conference (UAI 2004)* (pp. 504–511). Arlington, VA: AUAI Press.

Shalizi, C. R., Klinkner, K. L., & Haslinger, R. (2004). Quantifying self-organization with optimal predictors. *Physical Review Letters*, *93*, 118701.

Singh, S., Littman, M. L., Jong, N. K., Pardoe, D., & Stone, P. (2003). Learning predictive state representations. In T. Fawcett & N. Mishra (Eds.), *Proceedings of the Twentieth International Conference on Machine Learning*, (pp. 712–719). New York: AAAI Press.

Smith, A. C., Frank, L. M., Wirth, S., Yanike, M., Hu, D., Kubota, Y., et al. (2004). Dynamic analysis of learning in behavioral experiments. *Journal of Neuroscience*, *24*, 447–461.

Srinivasan, L., Eden, U. T., Mitter, S. K., & Brown, E. N. (2007). General purpose filter design for neural prosthetic devices. *Journal of Neurophysiology*, *98*, 2456–2475.

Szczepanski, J., Amigo, J. M., Wajnryb, E., & Sanchez-Vives, M. V. (2004). Characterizing spike trains with Lempel-Ziv complexity. *Neurocomputing*, *58*, 79–84.

Truccolo, W., Eden, U. T., Fellow, M. R., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble and covariate effects. *Journal of Neurophysiology*, *93*, 1074–1089.

Received December 20, 2007; accepted April 28, 2009.