Unsupervised identification of states from voltage recordings of neural networks

Davide Garolini, Andreas Vitalis, Amedeo Caflisch*

Department of Biochemistry, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

ARTICLE INFO

Keywords:
Multi-neuron simulation
Neural state identification
Network inference
GABAergic interneurons
NetSAP
SAPPHIRE plot

ABSTRACT

Background: Modern techniques for multi-neuronal recording produce large amounts of data. There is no automatic procedure for the identification of states in recurrent voltage patterns.

New method: We propose NetSAP (Network States And Pathways), a data-driven analysis method that is able to recognize multi-neuron voltage patterns (states). To capture the subtle differences between snapshots in voltage recordings, NetSAP infers the underlying functional neural network in a time-resolved manner with a sliding window approach. Then NetSAP identifies states from a reordering of the time series of inferred networks according to a user-defined metric. The procedure for unsupervised identification of states was developed originally for the analysis of molecular dynamics simulations of proteins.

Results: We tested NetSAP on neural network simulations of GABAergic inhibitory interneurons. Most simulation parameters are chosen to reproduce literature observations, and we keep noise terms as control parameters to regulate the coherence of the simulated signals. NetSAP is able to identify multiple states even in the case of high internal noise and low signal coherence. We provide evidence that NetSAP is robust for networks with up to about 50% of the neurons spiking randomly. NetSAP is scalable and its code is open source.

Comparison with existing methods: NetSAP outperforms common analysis techniques, such as PCA and k-means clustering, on a simulated recording of voltage traces of 50 neurons.

Conclusions: NetSAP analysis is an efficient tool to identify voltage patterns from neuronal recordings.

https://doi.org/10.1016/j.jneumeth.2019.01.019

Received 21 August 2018; Received in revised form 31 January 2019; Accepted 31 January 2019

* Corresponding author.
E-mail address: caflisch@bioc.uzh.ch (A. Caflisch).
1. Introduction

Biological neural networks exhibit intricate wiring architectures, exceptional variability at the level of single neurons and highly specific temporal dynamics. The most-used techniques for multi-neuronal recordings are two-photon calcium imaging (Denk et al., 1990; Wang et al., 2003; Stosiek et al., 2003; Helmchen and Denk, 2005), multi-electrode arrays (Du et al., 2011; Buzsáki, 2004; Najafi et al., 1990; Henze et al., 2000), and, more recently, light-sheet fluorescence microscopy (Ahrens et al., 2013; Dunn et al., 2016). These experiments produce a very large amount of data with high temporal resolution (more than 15 kHz multi-channel sampling (Li et al., 2016)) and spatial resolutions (up to 80,000 neurons recorded simultaneously (Ahrens et al., 2013; Dunn et al., 2016)). Despite these technical advancements, a robust understanding of neural encoding is still missing (Stevenson and Kording, 2011; Rolls and Treves, 2011; Romo and de Lafuente, 2013; Cohen and Kohn, 2011; Yuste, 2015), and an unequivocal matching of stimuli with neuronal responses is not routinely achievable (Rolls and Treves, 2011; Cohen and Kohn, 2011; Bialek et al., 1991; Engel et al., 2001; Shadlen and Newsome, 1998; Harris et al., 2003).

In practice, the most common approach is to derive firing rates from calcium or voltage recordings using spike sorting or inference (Denk et al., 1990; Wang et al., 2003; Stosiek et al., 2003; Helmchen and Denk, 2005; Du et al., 2011; Buzsáki, 2004; Najafi et al., 1990; Henze et al., 2000) and analyze them using various dimensionality reduction techniques (Cunningham and Yu, 2014) such as principal component analysis (Cohen and Kohn, 2011; Harris et al., 2003; Harvey et al., 2012; Freeman et al., 2014; Lopes-dos Santos et al., 2011; Peyrache et al., 2009), clustering algorithms (like k-means (Freeman et al., 2014), or other correlation-based techniques (Ahrens et al., 2013; Cohen and Kohn, 2011). Among the most relevant theoretical frameworks for the analysis of neural recordings we mention here point process analysis (Kass et al., 2014), generalized linear models (GLMs) (Truccolo et al., 2005), entropy-based methods (max entropy models (Schneidman et al., 2006), transfer entropy information routing (Kirst et al., 2016; Palmigiano et al., 2017)), statistical decoders (maximum likelihood estimation (Paninski, 2004; Cocco et al., 2017), maximum a posteriori estimation (Ota et al., 2009), Ising decoders (Schaub and Schultz, 2012)), manifold analysis (Gallego et al., 2017), trajectory-based decomposition (Harvey et al., 2012), and graph-based inference (Pitkow and Angelaki, 2017). Additionally, hidden Markov models (HMM) have been used to explain how neural networks encode “hidden” variables, which are indirectly related to external stimuli (observed variables) (Kollmorgen and Hahnloser, 2014; Vidaurre et al., 2016).

All these methods define the levels of information content that appear to be significant for specific stimuli encoded by a given neuronal population. To do so, the raw data are usually preprocessed and translated into spikes, i.e., they are transformed from an analog to a digital representation (Sengupta et al., 2014). For example, one of the most successful predictors of behavioral context are GLMs (Kass et al., 2014; Truccolo et al., 2005) which usually require a binary data representation. This approach subscribes to the fundamental idea that information is represented by spike-encoded signals (Cohen and Kohn, 2011), which is known to be maximally efficient only at the population level (Boerlin and Denève, 2011; Boerlin et al., 2013). However, it has been argued that the (analog) membrane potentials contain valuable information, e.g., in their sub-threshold variations (Sengupta et al., 2014; Debanne et al., 2012). Moreover, it has been found experimentally that the spiking rates follow a Poisson-like process with little regularity at the single-neuron level while the membrane potentials have been argued to exhibit significant autocorrelation (Gentet et al., 2010; Yu and Ferster, 2010). For these reasons, here we decided to focus on raw, analog signals.

We propose NetSAP (Network States and Pathways) for the analysis of voltage recordings. NetSAP is based on the automatic procedure called SAPPHIRE (States And Pathways Projected with High REsolution), developed originally for identifying states along molecular dynamics simulations of proteins (Blöchliger et al., 2013; Blöchliger et al., 2014; Blöchliger et al., 2015; Vitalis and Caffisch, 2012). Protein motion and neuronal recordings share a high degree of complexity which is inherent to large and flexible molecules and a large set of neurons in a network, respectively. More precisely, neural networks and proteins (e.g., enzymes) can populate multiple states with specific functional roles in the dynamical system. For example, neural networks alter their functional connectivity depending on the information content which is transmitted or transformed. Similarly, the activity of an enzyme is usually modulated by small-molecule modifiers (Baici, 2015), which influence the conformation of the enzyme upon binding. These states are often metastable, meaning that they have a life time shorter than the one of the enzyme itself and longer than fluctuations due to thermal motion. In stochastic dynamical systems this implies recurrence, as each state can be visited an arbitrarily large number of times.

We are able to routinely analyze long time series, e.g., long and/or many trajectories of protein motion from molecular dynamics simulations. This is due to two mitigating factors. First, the scalability of our method is \( N \log N \) where \( N \) is the number of snapshots and \( D \) is the dimensionality of an individual snapshot. Second, there are hardly any overly sensitive parameters to tune. Both advantages together allow a seamless application to large data sets. Indeed, one of the major difficulties in approaches found in the literature is parameter fitting, which often scales poorly with the amount of data available. Our work is focused on avoiding strict impositions of any particular external factor, such as the assumption of learning or fitting a specific model, in order to differentiate neural information. Moreover, we decided to apply NetSAP on raw voltage recordings as we believe these to be the most direct and in some ways challenging form of data. Nonetheless, NetSAP is similarly applicable to other experimental data such as from functional magnetic resonance imaging (fMRI) (Huettel et al., 2004) or calcium imaging (Denk et al., 1990; Wang et al., 2003; Stosiek et al., 2003; Helmchen and Denk, 2005) as the underlying data mining pipeline is general.

The NetSAP approach proposed in this manuscript is an unsupervised method intended to capture the differences between neural messages without delving into efforts to directly translate the neural code. In other words, we try to identify neural states automatically, that is, without consulting associated behavioural tasks and outcomes. We rely on the assumption that neural packages of information are exchanged under the form of dynamic functional connectivity (Preti et al., 2017) which is a common assumption in state-driven analyses of fMRI (Allen et al., 2014; Yasoubi et al., 2015a,b) and electroencephalography (EEG) data (Hirayama et al., 2015; Allen et al., 2018). This approach has been able to differentiate clinically relevant mental disorders (Buckley et al., 2015; Kirino et al., 2017), even with spatial and temporal limitations (Hutchison et al., 2013; Hindricks et al., 2016). As mentioned above, we use a density-based cluster estimator called SAPPHERE (Blöchliger et al., 2013; Blöchliger et al., 2014; Blöchliger et al., 2015; Vitalis and Caffisch, 2012), which shares some characteristics with DBSCAN (Ester et al., 1996) and HDBSCAN (Campello et al., 2013, 2015; McInnes et al., 2017). During the writing of our paper, a work was published that uses HDBSCAN to find neural states from digital spike representations (Grossberger et al., 2018).

To test our analysis method we use a controlled computer simulation. For the data generation, we adopt a network model of GABAergic inhibitory interneurons (i.e., expressing \( \gamma \)-aminobutyric acid neurotransmitter), which are considered reliable network regulators (Töth et al., 1997; Gulyás et al., 1990; Freund and Antal, 1988; Pelkey et al., 2017). Each neuron in the simulated network receives as input an external current that mimics excitation of pyramidal neurons. The simulated system has been tailored to emulate experimental voltage recordings from the hippocampus with important physiological features.
like afterhyperpolarization (Wang and Buzsáki, 1996; McKay et al., 2013), excitatory/inhibitory balance (Xue et al., 2014), and theta/gamma rhythms (Buzsáki, 2002; Belluscio et al., 2012; Fries, 2015; Headley and Paré, 2017).

2. Materials and methods

This section consists of two parts. First, we generate realistic voltage recordings using a simulated neural network. Second, we take the voltage recordings and, without preprocessing, analyze them in an unsupervised manner. The main goal is the automatic identification of states without knowing the external context and by using only raw voltages. A general pipeline is depicted in Fig. 1 and described in the following.

2.1. Simulation

The graph structure in Fig. 1 shows an example of the simulated neural network. We focused on inhibitory Wang-Buzsáki neurons (Wang and Buzsáki, 1996) because they have been shown to be fundamental for population coding (Stefanelli et al., 2016; Buzsáki, 2001) and have a clear voltage trace with typical afterhyperpolarization (Wang and Buzsáki, 1996; McKay et al., 2013) (Fig. 2a). Moreover, it has been shown that these neurons, which represent 10–15% of the neuronal population in the hippocampus, are a major determinant of the regulation of information flow by controlling the spiking in a time-windowed manner (Tóth et al., 1997; Gulyás et al., 1990; Freund and Antal, 1988; Pelkey et al., 2017). These neurons have been reported to drive the timing of systemic excitability by means of broad innervation (Pelkey et al., 2017). In the hippocampus they are, therefore, considered fundamental regulators of the oscillatory activity across different frequencies (Buzsáki, 2002). Significantly, networks of interneurons have been suggested as one of the major players in frequent neurological disorders such as schizophrenia (Marín, 2012; Lisman et al., 2008; Lewis et al., 2005) and epilepsy (Trevelyan et al., 2007; Marx et al., 2013). In both cases, a modification at the circuitry level and persistent defects in theta/gamma waves were found to be fundamental anomalies (Lisman et al., 2008; Trevelyan et al., 2007; Marx et al., 2013).

Here, we considered a system focusing on the subset of interneurons (Pelkey et al., 2017) that are disinhibitors (Freund and Buzsáki, 1996; Blasco-Ibáñez and Freund, 1995; Bezaire and Soltesz, 2013; Freund and...
Fig. 2. Characteristics of the simulated data set. (a) Voltage values for one single neuron during 200 ms of simulation (black). The simulated experimental error is added as pink and white noise (blue). (b) Univariate distribution of voltage levels from one interneuron during a 20-s simulation. (c) Scatter plot of spikes showing bursts of activity mimicking the theta oscillations in the input. (d) The total input current for all the 50 neurons during one second of simulation. (e) One neuronal output during one second of simulation is decomposed into its three main components: constant \( I_{\text{const}} \), oscillatory \( I_{\text{osc}} \), and random \( I_{\text{rand}} \) currents. (f) Excitatory/inhibitory balance. The orange line represents the average excitatory contribution of the network in terms of current (\( I_{\text{exc}} \), black), roughly six oscillation in one second shown. (g) Firing rates (FR) of the neural population with bin sizes of 1 and 15 ms (gray and black, respectively). Frequencies on the y-axis are indicative only for the gray curve. The gamma oscillations (in gray, 40–100 Hz) are nested in theta rhythms (in black, roughly six oscillation in one second shown).

\[
C_i \frac{dV}{dt} = -I_{\text{Na,i}} - I_{\text{K,i}} - I_{\text{leak,i}} - I_{\text{syn,i}} + I_{\text{ext,i}}
\]

\[I_{\text{ext,i}} = I_{\text{leak,i}} + I_{\text{Na,i}} + I_{\text{K,i}}\]

where \( C_i \) is the membrane capacitance while \( I_{\text{leak,i}} \) and \( I_{\text{ext,i}} \) are the leaky and applied electric currents (external to the network), respectively. Furthermore, \( I_{\text{Na,i}} \) and \( I_{\text{K,i}} \) follow the Hodgkin-Huxley formulation of the ion channels (Hodgkin and Huxley, 1952). In this implementation, kinetics and maximal conductances are modified according to Wang and Buzsáki (1996) to fit hippocampal fast-spiking neurons. These cells show brief voltage afterhyperpolarization of about 15 mV after each spike (McCormick et al., 1985) (Fig. 2a and b) and can occasionally reach high frequencies (200–400 Hz (McCormick et al., 1985)) although they usually operate in the 20–100 Hz range (gamma oscillations). In our model, variations in cell features were approximated by a small variability in the \( C_i \) (extracted from a normal distribution with mean \( \mu_{C_i} = 1 \mu F/cm^2 \) and standard deviation \( \sigma_{C_i} = 0.1 \mu F/cm^2 \)).

The synaptic current \( I_{\text{syn}} \) (Eq. (3)) depends on the unitless gating variable \( s_i \), which roughly describes the fraction of open ion channels. This variable is the only inhibitory connection between interneurons. The variable \( s_i \) is bound between 0 and 1, and its dynamics are described by a single differential equation (Eq. (4)).

\[
I_{\text{syn,i}} = g_{\text{syn}} V_i - V_{\text{syn}}
\]

\[
\frac{ds_i}{dt} = \frac{\alpha (1 - s_i)}{1 + e^{-\tau s_i / \beta i}} - \beta_i
\]

The conductance \( g_{\text{syn}} = 0.1 \text{mS/cm}^2 \), channel opening rate \( \alpha = 12 \)

\( \beta_i = 36 \) (extracted from a normal distribution with mean \( \mu_{\beta_i} = 36 \) mS/cm and standard deviation \( \sigma_{\beta_i} = 5 \text{mS/cm} \)).
ms^{-1}, channel closing rate \( \beta = 0.1 \text{ ms}^{-1} \), and \( V_{\text{pre}} = -75 \text{ mV} \) are kept constant (Wang and Buzsáki, 1996). \( V_j \) is the membrane potential of the post-synaptic neuron while \( V_{\text{pre}} \) is the pre-synaptic voltage.

The system of differential equations (1–4) was simulated using Brian2 (Goodman and Brette, 2008; Goodman et al., 2014). The input to the network is modeled as external currents (\( I_{\text{ext}} \)) that resemble the receptive field of each interneuron from different populations of excitatory pyramidal cells. This external contribution has an important role in the formation of theta oscillations in the hippocampus and is considered the major input to interneurons (Buzsáki et al., 1983). We modeled the oscillations using oscillatory currents with a frequency of 6.4 Hz. Importantly, the model naturally gives rise to gamma oscillations (25-100 Hz) nested in theta oscillations (Fig. 2g) in accordance with in vivo observations (Buzsáki, 2002; Belluscio et al., 2012; Fries, 2015; Headley and Paré, 2017).

The applied current \( I_{\text{ext}} \) (Eq. (2), Fig. 2e) consists of three main components: first, a constant baseline \( I_{\text{bo}} \) of 0.6 \( \mu \text{A} \) (green line); second, an input process with a periodic sinusoidal signal \( I_{\text{si}} \) (amplitude of 0.6 \( \mu \text{A} \), red line); last, an Ornstein-Uhlenbeck process (represented by the \( I_{\text{ou}} \) contribution) that adds noise to the differential equations (blue line). A fundamental aspect of the simulations is the balance between inhibitory and excitatory currents (Xue et al., 2014). This property is supported by experimental evidence (Atallah and Scanziani, 2009) and is believed to be crucial for the homeostatic coupling needed for efficient coding and computation (Lim and Goldman, 2014, 2013). In particular, we highlight here the concept that the excitatory/inhibitory balance can be positive or negative in order to effectively convey and transform information in time in a manner that goes beyond the tight homeostatic excitatory/inhibitory balance (Denève and Machens, 2016). In other words, packages of information may be propagated in the neural network using specific alterations in the excitatory/inhibitory balance (Lim and Goldman, 2014, 2013). In Fig. 2f, we show the momentary shifts in the equilibrium (black line) as the difference between the mean of all excitatory currents (orange) vs. that of all inhibitory currents (gray).

All the simulated parameters have been set following experimental findings (Pelkey et al., 2017; Wang and Buzsáki, 1996) or an ‘edge-of-chaos’ approach (van Vreeswijk and Sompolinsky, 1996). For the latter, we intended to avoid possible divergences in the information content of the neural signal. In other words, we allow neurons to spike irregularly while we still keep trace of the underlying signal which must be recurrent. We use coherence to measure voltage variability and derive adequate parameters for the simulation:

\[
C_{xy} = \frac{S_{xy}}{S_x S_y}
\]

where the cross spectral density estimate \( S_{xy} \) is divided by the single spectral densities of \( x \) and \( x' \), \( S_x \) and \( S_y \), respectively. The spectral densities are calculated using Welch’s method (Welch, 1967) with segments of 200 snapshots and 90% of overlap. In detail, we divided every simulation into two equal parts and computed the coherence for every neuron, i.e., \( x \) is the same neuron as \( x' \), but in the first and second halves of the data set, respectively. In this way, we can evaluate for individual frequencies how the signal of each neuron changes during the simulation. In the case of zero coherence, we expect that the first part of the simulation is completely different from the second part. For the analysis of the simulation parameters we always used simulations of 20 s (20,000 snapshots).

Finally, experimental error is simulated using white and pink noise that is added to each point of the voltage traces. The error terms were generated using the same dispersion (standard deviation) as employed in the simulated data set (around 25 mV). In Fig. 2a, this process is shown for one neuron.

2.2. NetSAP analysis

To analyze neural time series, we have extended here the SAPPHIRE algorithm developed originally for the analysis of molecular dynamics simulations (Blöchliger et al., 2013; Blöchliger et al., 2014; Blöchliger et al., 2015; Vitalis and Caffisch, 2012). This method is useful to sort and classify instantaneous snapshots from dynamical processes that change continuously in time. In practice, the method’s only essential hyperparameter is a distance metric between snapshots. Briefly, the algorithm starts from an arbitrary time point, subsequently adds the snapshot that is geometrically nearest to any of the snapshots already added and thus constructs a reordered time line (called progress index) without repetition. Those points in the input data that are visited extensively and come from a structurally homogeneous region will be grouped closely. A kinetic annotation (Krivov and Karplus, 2006) highlights specific states or basins based on their temporal persistence, at least when visited recurrently. In general, the density of snapshots, which is metric-dependent, should be high in the basins (states or attractors) and low in the connecting areas (‘barriers’ or transition states). This similarity clustering is highly efficient at revealing the stochastic dynamics of very complex systems, e.g., when there are many small attractors found across very long recording times. Notably, the algorithm is already scalable (i.e., applicable to very large data sets) and has been used to analyze millions of time points and hundreds of variables (Bacci et al., 2015).

We used mainly the minimum spanning tree (MST) construction to create the SAPPHIRE plot because it has no auxiliary parameters. When computing times were unfeasible, we used an approximation, the short spanning tree (Blöchliger et al., 2013; Vitalis and Caffisch, 2012), which smoothly converges to the MST with straightforward parameter choices. In particular, the number of search attempts defines how many snapshots are compared to connect the spanning tree (Fig. 1c). A larger number of attempts gives a better approximation of the MST but the dependency of the metric of interest (NMI, see below) is very weak here.

We adopted the SAPPHIRE algorithm because it can elucidate subtle temporal features and resolve rare events as it preserves kinetic information (Blöchliger et al., 2013; Blöchliger et al., 2014; Blöchliger et al., 2015; Vitalis and Caffisch, 2012). In particular, the SAPPHIRE analysis identifies states that contribute to the preservation of information for short periods of times, which we hypothesize to constitute the physical basis for neural encoding. We assume that neural networks exchange information in finite packages that persist during prolonged and repeated input presentations. Therefore, the approach is based on analyzing the data with the goal of highlighting the functional state of the network, i.e., the precise condition of the information exchange at a certain time point. How can this functional state be represented for an instantaneous time point? Here, we use a sliding window procedure as follows. For every snapshot, data from the time window centered on this point are used to calculate a correlation measure for every pair of neuron traces and we chose the inverse of the Minkowski distance for this. These data constitute a complete correlation (or similarity) matrix between neurons for each individual snapshot. The time series of these matrices provides a dynamical picture of the functioning of the network. Fed into the SAPPHIRE analysis, we employ the Euclidean distance between matrices as the (dis)similarity measure. However, to counterbalance the presence of weakly informative variables, a dimensionality reduction procedure is applied to the elements of these matrices, and only the first 15 principal components are retained. The work-flow is depicted in Fig. 1.

We termed the overall procedure NetSAP (Network States And Pathways). The method belongs to the group of time-based clustering descriptors (Liao, 2005). Our aim is to infer the landscape of accessible states from observations made at discrete and generally regular time points. Broadly speaking, NetSAP serves as an unsupervised clustering algorithm that relies on transient network states to identify the system’s...
local minima (basins or cluster centers) and their connections. Notably, NetSAP does not require a priori definition of the number of clusters or their sizes and the window length is its main impactful parameter.

To quantify the performance of NetSAP, we compare the original input annotation (true labels) and the states identified by SAPPHIRE analysis (predicted labels) using a compact descriptor of the clustering performance. As shown in Results, here the true labels derive from controlled variations in the input presentation to the network. In particular, we adopt the normalized mutual information score (NMI (Strehl and Ghosh, 2002)) to evaluate the identified states:

\[
\text{NMI}(tc, pc) = \frac{\text{MI}(tc, pc)}{\sqrt{H(tc)H(pc)}}
\]

where tc and pc denote true and predicted labels, respectively, while \(H(\cdot)\) is the Shannon entropy and \(\text{MI}(\cdot, \cdot)\) is the mutual information between two different partitions. In case the exploratory analysis of the SAPPHIRE plot highlights fewer or more states than the true annotation, we do not impose any merging or splitting. A fundamental property of clustering comparison metrics is the so-called constant baseline property (Vinh et al., 2010). If this property is not fulfilled, then the comparison metric will be positive, even in case of labels that were generated randomly. In practice, this happens when the number of data points is scarce and the number of clusters is high. Vinh et al. suggested that NMI complies with this property if \(N/n_	ext{cluster} > 100\), where \(N\) is the number of points and \(n_	ext{cluster}\) is the number of clusters (Vinh et al., 2010). Our simulated data is above this soft limit. Nevertheless, we adopted the adjusted Rand index (ARI) (Hubert and Arabie, 1985) in the case of more than four clusters (Figs. 9c–f, 10, 11).

The SAPPHIRE analysis is used to identify the most relevant barriers between states or, in other words, the best separations between basins. The identification can be done by manually selecting the barriers on the cut profile of the SAPPHIRE plot or by using an automatic procedure. For this latter purpose, we focus on evaluating the overlap between points on the temporal annotation of the progress index (gray dots in Fig. 3a and b). To do so we uniformly divide the x-axis in partitions and repeat this process with different numbers of divisions, as so to have a variable number of separators in different positions of the progress index (Fig. 3a). As a standard, we divide the progress index in uniform partitions 50 times by using a variable number of separators, usually between 10 and 200. In Fig. 3a, we highlight nine uniform separators of the progress index distribution with vertical red lines. The resulting 10 divisions include groups of points that are successively divided into horizontal histograms, i.e., we took the y-axis (experimental time) as the traditional binning axis (Fig. 3b, 20 bins). By measuring how well two consecutive groups of points overlap, we obtain an estimate that should be proportional to the likelihoods that they come from a homogeneous basin. We take the maximal information coefficient as the measure of overlap (Reshef et al., 2011).

In practice, we estimate distributions of points with horizontal histograms and we calculate the inverted maximal information coefficient (IMIC) between consecutive pairs of vectorized histogram counts. The metric shows low values where the points are similarly distributed, while it presents peaks where there is vertical separation, i.e., there is a jump between distributions of points. The IMIC between two count vectors a and b can be written as

\[
\text{IMIC}(a, b) = 1 - \max_{A < B} \frac{-\text{MI}^*(A, B)}{\log(\min(A, B))}
\]

The values of the horizontal histograms a and b are used as numerical vectors and the scatter plot between these values is used by MIC to infer their relationship. In particular, MIC assumes the following: if there is a functional relationship between two variables, it must be possible to define a specific 2D histogram on the scatter plot which maximizes their mutual information. In Eq. (7), (A, B) are bins of vectors a and b, respectively. To limit the search space of possible bins, we used \(n^6\) where n is the number of elements in a and b, and a is a constant (fixed at 0.4). \(\text{MI}^*(A, B)\) is the maximum mutual information over all possible bins (A, B). These calculations are made using the R package minerva (Albanese et al., 2013).

Next, we linearly interpolate the scores found for each group of barriers, and we average them, thus obtaining the black curves in Fig. 3a and b (and the red curves in Fig. S3a–e). We use a peak finding algorithm with a span of 2500 to identify all relevant barriers and sort these maxima by their associated IMIC values. Considered in succession, they give rise to the corresponding profile in Fig. 3c (red, segmented line). Because the IMIC is monotonously decreasing with growing number of states, the selection of the number of barriers to use can be ambiguous (e.g., Fig. S3e). For this reason, the PBM index (Pakhira et al., 2004) (black), a consensus choice of four states emerged. The NMI score (blue) is an a posteriori validation using the true labels, i.e., the ground truth.
different data sets and partitions (Vendramin et al., 2019). The final selection of barriers is shown by vertical blue lines in Fig. 3a (and by dotted vertical lines in Fig. S3a–e). Generally, the resulting choice of barriers is in good accordance with a simple visual inspection.

We require one parameter for the automatic identification of peaks, which is the aforementioned minimum number of points (span) between them (for example, in Fig. 3 the minimum number is 2500 snapshots). This is a soft requirement as it is mainly necessary to avoid the selection of multiple barriers from the same peak. Since the span requirement limits the number of peaks that can be found, we use the average peak height as a cutoff for inferring the number of states. The IMIC curve is not particularly sensitive to the other parameters, i.e., the number of bins for the histogram construction and the number of uniform splits, and we keep them fixed here. As mentioned above, we used from 10 to 200 splits 50 times, and 50 bins (only 20 bins are shown in Fig. 3b to avoid overcrowding).

To be certain of the NMI robustness against random inputs, we selected the simulation with low noise (Fig. S3a) and shuffled its progress index. The resulting NMI is close to zero (Fig. S4a). As a second test, we also shuffled the true label vector, i.e., the annotation, and obtained similarly low scores (Fig. S4b). Finally, we wondered how the choice of barriers could affect the results. To investigate this possibility, we took different simulations and picked the barriers 500 times at random. Each random pick was constrained to be distant from every other pick by at least 2500 snapshots (the same limit used for the automatic barrier identification). Afterwards, the resulting distribution of NMI values was compared with the score calculated using the automatic procedure (Z-score, Fig. S3f–j, obtaining p-values lower than 5%, see Fig. S3f–j). These statistical tests indicate that the automatic procedure for the choice of barriers is robust and performs substantially better than a random choice.

3. Results

3.1. Analysis of simulation parameters by coherence of voltage patterns

The coherence is influenced strongly by the noise current \( I_n \) (Fig. 4a). It deteriorates to values close to 0.01 for a maximal semi-amplitude of \( I_n = 0.4 \mu \text{A} \) at frequencies higher than 30 Hz. An oscillatory component larger than 0.5 \( \mu \text{A} \) is required for a high coherence (Fig. 4b). In particular, the oscillatory current \( I_w \) regulates directly the theta frequency of the wave (around 6 Hz, Fig. 2g), while the gamma ripples are defined by the finer details of the neural network, e.g., the inhibitory connections. The other components of \( I_{ext} \) decrease signal coherence (Fig. 4c and d). In the particular cases in which \( I_w > 2I_c \), we see a drop in the coherence of the theta waves (6 Hz, green in Fig. 4b–e). Nonetheless, the low frequency regime is generally more robust toward parameter variation than the higher frequency regime.
is the size of the sliding window. In which

\[ p = 80 \text{ s}. \]

Four distinct states were defined by input currents that differed slightly in timing and intensity for each neuron of the neural network. In detail, the phase and amplitude of the oscillatory input currents were taken randomly from a normal distribution having standard deviations of 0.1 radians and 0.01 μA, respectively. We subsequently simulated the four states along windows of variable lengths (between 1 s and 10 s) and connected the resulting data sets in a single time series. No other parameter was changed during the simulation, and the network

connections were kept identical, similar to the one shown in Fig. 1. We did not take into consideration a possible bias when the simulation is reinitialized because, in a simulation with low noise terms, 300 ms are the maximum time needed to stabilize the spiking patterns. The simulated input sequence is shown in Fig. 5a as a continuous line between 60 ms and 70 s, which corresponds to only 1s, i.e., 1000 consecutive snapshots (highlighted in green).

(b) Temporal order of true states (T) and the states predicted using NetSAP (P-N) and k-means++ (Arthur and Vassilvitskii, 2007) (P.K). The latter method, using \( k = 4 \), is only able to recognize the blue state partially (NMI of 0.15).

3.2. Identification of neural states by NetSAP analysis

Using the settings described in the coherence analysis, a first benchmark simulation of a 50-neuron network was performed, and voltage values were recorded for all neurons every millisecond during 80 s. Four distinct states were defined by input currents that differed slightly in timing and intensity for each neuron of the neural network. In detail, the phase and amplitude of the oscillatory input currents were taken randomly from a normal distribution having standard deviations of 0.1 radians and 0.01 μA, respectively. We subsequently simulated the four states along windows of variable lengths (between 1 s and 10 s) and connected the resulting data sets in a single time series. No other parameter was changed during the simulation, and the network

connections were kept identical, similar to the one shown in Fig. 1. We did not take into consideration a possible bias when the simulation is reinitialized because, in a simulation with low noise terms, 300 ms are the maximum time needed to stabilize the spiking patterns. The simulated input sequence is shown in Fig. 5a as a continuous line between 60 ms and 70 s, which corresponds to only 1s, i.e., 1000 consecutive snapshots (highlighted in green).

(b) Temporal order of true states (T) and the states predicted using NetSAP (P-N) and k-means++ (Arthur and Vassilvitskii, 2007) (P.K). The latter method, using \( k = 4 \), is only able to recognize the blue state partially (NMI of 0.15).
Fig. 6b), we can see a clear correspondence (NMI of 0.92). Conversely, a clustering procedure on the inferred functional networks managed to achieve only a partial recognition of states. Using k-means++ (Arthur and Vassilvitskii, 2007) we obtained an NMI of only 0.15 (P-K, Fig. 6b). As additional comparisons, we performed a principal component analysis (PCA) of the whole data set and constructed a SAPPHIRE plot without network inference (Fig. 7a and b, respectively). Both of these controls fail to separate the four states.

Importantly, the NetSAP algorithm is able to resolve very short excursions into different states as highlighted by the green rectangle in Fig. 6a. The overall analysis procedure required 15 min (Fig. S2) on a commodity computer running Linux (Core i7-4790 CPU with 3.60 GHz x8 and 16 GB RAM). Indeed, the approximate algorithm (the so-called short spanning tree procedure) can be highly efficient and scales comfortably to millions of data points. For the data set considered here, 5000 search attempts for the construction of the short spanning tree are enough to obtain satisfactory performance (Fig. S1c).

3.3. Robustness of NetSAP

First we investigated the robustness of NetSAP for increasing noise current. We generated 20 different simulations with identical settings but using different amplitudes for the noisy input current. For frequencies higher than 6 Hz, the coherence deteriorates significantly already at a noise current of 0.1 μA (Fig. 8a). Using NetSAP and automatically selecting the barriers as described previously, we were able to find the four relevant states and to calculate an NMI score for each simulation (Fig. 8b). For further analyses, we selected five indicative simulations with the following noise levels: 0.01, 0.04, 0.12, 0.22, and 0.30 μA (gray shaded circles). Using these data sets, we estimated the variability of NetSAP by running the analysis 35 times for each noise level changing only the random seed for the short spanning tree construction (5000 search attempts) and the automatic selection of barriers. The NMI score shows that the NetSAP analysis correctly identifies basins up to a noise level of 0.2 μA (Fig. S1a) while the standard deviation of the NMI score does not depend on the noise level. Moreover, the principal sources of variability, i.e., the number of search attempts and the automatic barrier selection, can be potentially limited by manual optimization. For example, one can improve the identification of the barriers by enhancing the number of bins on the y-axis or by using more splits of the progress index distributions.

All the predicted states have been calculated using the SAPPHIRE plots, and in particular using only the temporal annotations (distribution of points on the progress index). For this reason, we tested the procedure of automatic barrier selection with randomized null models (Fig. S4). In addition, we compared the NMI score generated by the automatic selection of the barriers and the distribution of NMI scores produced by random picks of divisors. The distance of the optimal score from the distribution has been estimated using Z-scores and p-values (Fig. S3f-j). As expected, the automatic procedure produces significantly better results than the random selection, e.g., Fig. S3a-d.

Instead, when the NMI score drops below a threshold of around 0.25, the selected barriers produce similar scores in comparison to a random
We also wanted to test if the automatic barrier selection and the resulting NMI score are dependent on the number of clusters in the analyzed system. To this end, we first artificially increased or reduced the number of states in the true label vector (annotation). In practice, we divided one state into two, thereby obtaining five states overall (Fig. S5c,f). Alternatively, we merged two states to arrive at three final states (Fig. S5b,e). Moreover, we tried to look for four states after merging two in the true annotation (Fig. S5a,d). In these cases, the better divisors in terms of NMI are generated from the random picks (red dots in Fig. S5a–c). This is expected because the temporal distribution is, in all cases, mismatched with the true label vector. These results provide...
evidence that the automatic identification makes specific use of the information about the kinetic partitioning of the states, which is desired. The best results using random splits are superior because, in essence, they are optimized using the real annotation, i.e., by calculating the final NMI score. The simple nature of the modification of the label vector means that maximum values based on random splits can stay close to the scores obtained with the original assignment of four states (NMI score of 0.84, Figs. S3a and S5a–c).

The NetSAP analysis of the 4-state system with an average coherence of 0.7 showed high sensitivity, even to short 1-s input presentations (Fig. 6a). As a more realistic test, we thus wanted to explore the limits of NetSAP performance for a voltage recording with lower coherence and more states. Therefore, we carried out a 200-s simulation of a 50-neuron network with 20 states, a noise current of $I_n = 0.02 \mu A$ (mean coherence of 0.5), and an average residence time between 2000 ms and 3000 ms (Fig. 9). This test is particularly challenging as we did not impose any threshold on the minimum difference between inputs, and the neural states can thus overlap. The identification of states (as measured by an ARI of 0.54) is above chance levels (Fig. 9c–f). We wanted also to analyze if a higher number of neurons is beneficial for the correct identification of states. To do so, we performed 400 simulations of 100 s with a number of states ranging from 2 to 40, and the number of neurons ranging from 10 to 105. The parameters were kept identical to those in Fig. 5, with the exception of $I_n = 0.02 \mu A$ instead of $I_n = 0.01 \mu A$, and the average residence time which was randomly picked from a flat distribution between 1000 ms and 3000 ms. The resulting ARI scores are shown in Fig. 10a. As expected, a higher number of neurons permits a better identification of states. Conversely, NetSAP is not reliable when the number of states is larger than about 30. This failure is possibly due to overall simulation length, which does not allow for a sufficient level of recurrence when there are many states.

Next, we considered the case of multiple voltage patterns transmitted through different subpopulations of the network at the same time. For this purpose, we simulated a neural network consisting of two groups of neurons where each group represents a different voltage pattern. In practice, we made 10 simulations of 90 s each in which a subpopulation is involved in a 6-state system while the other is related to a different group of states (3-state system). To show that raw voltages are essential while firing rates would not be an effective pre-processing method, we simulated a network in which the two populations convey the voltage patterns synchronously, meaning that the theta oscillations are aligned. The other parameters were kept identical to Fig. 10a. We varied from 10 to 60 the number of neurons involved in the population representing the 6-state system and simulated a total of 70 neurons. The remaining neurons encoded the 3-state system. For the SAPPHIRE analysis we imposed a true annotation with 3 or 6 states. The profile of the ARI value as a function of the subpopulation size increases for the networks with increasing amount of neurons that code for the system with the number of states used in the analysis (red dots in Fig. 10b). Qualitatively, an ARI score higher than 0.6 is obtained for the networks in which at least 40 of the 70 neurons are involved in the representation of the system with the same number of states used for SAPPHIRE analysis.

Finally, we considered the possibility of having a subpopulation of completely unrelated neurons that spike at random times (Fig. 11). We modeled these neurons by removing the external oscillatory current $I_e$ and increasing their noise current $I_n$ by a factor of four. The other parameters were kept identical to the simulations used for Fig. 10a with 70 total neurons and 4 patterns, while we varied the number of unrelated neurons. The firing rates (Fig. 11c) differ from those of the original system (Fig. 2g) because the spiking times of the decoupled neurons are randomly distributed. The different voltage traces of the two populations are depicted in Fig. 11d. We simulated 22 neural networks with different numbers of unrelated neurons (from 2 to 65), and we calculated the resulting ARI (Fig. 11e). We found that the ARI score drops significantly when the number of random spiking neurons grows above 50% of the total of 70 neurons.

4. Discussion

We propose NetSAP as a data-driven analysis tool to identify neural states from voltage recordings. NetSAP was successfully applied to realistic albeit simplified neural simulations. With four different external inputs to the neural network, NetSAP was able to identify their four major corresponding neural states using only the raw voltage levels (Fig. 6a and b). Conversely, a standard clustering approach recognized only partially one of the states (k-means++ (Arthur and Vassilvitskii, 2007), Fig. 6b). A naive principal component analysis failed to isolate any of the four neural states (Fig. 7a).

In a more challenging test with 20 different patterns in 200 s of simulation (mean coherence of 0.5) NetSAP successfully mapped more than 65% of the snapshots to their correct labels, resulting in a total ARI of 0.54 (Fig. 9). The studied system was particularly challenging due to the low levels of recurrence and the short life times of states. In addition, we explored how the identification of states can be affected by the number of neurons and states involved (Fig. 10a). In the cases in which the number of states is high, a fixed number of snapshots limits the
The ARI scores highlight those neurons that are most consistent with a given annotation. For this task, specific techniques like gradient optimization and simulated annealing are available as well (Satuvori et al., 2018). This procedure would go beyond the usual experimental methods, which aim at average activity as their major selection rule (Ahrens et al., 2013; Cohen and Kohn, 2011; Harvey et al., 2012; Freeman et al., 2014; Lopes-dos Santos et al., 2011; Peyrache et al., 2009). The number of neurons, C, is also of practical importance for NetSAP. The initial inference of networks leads to N snapshots of dimensionality $D = C^2$, and these are fed into the dimensionality reduction by PCA. In a sparse regime, where $N$ is not significantly larger than $C$, this may cause issues. However, as NetSAP relies on differences in sampling density for identifying states, this regime is poorly suited to begin with.

For real data sets, it is clearly of supreme interest to match a ground truth defined by a relevant behavioral annotation with neural states. As mentioned above, it is a frequent assumption that not all neurons in the network are relevant, and the identification of behaviorally critical ones is a common goal. For example, consider an experiment aimed at recognizing hallmarks of a specific neural disorder against a healthy control group. A direct analysis may rely on overall activity levels for individual neurons or on the correlation of this activity with the behavioral ground truth. Instead, the NetSAP pipeline offers an indirect route to prune the measured data sets: by bootstrapping in neuron space and a subsequent, unsupervised inference of neuronal states, it may be possible to find partitions that have significant NMI values with the behavioral annotation. This offers a natural route to not only prevent overfitting but also to independently estimate whether the behavioral state is clearly encoded in these neurons at all.

The NetSAP analysis pipeline can be improved further. For example, more sophisticated network inference methods could unveil complex relationships between neurons (e.g., using transfer entropy (Schreiber, 2000) or directed information (Cai et al., 2017)). As alluded to above, the metric used for measuring similarity between inferred networks (snapshots) could be optimized using network-specific metrics like the Ipsen-Mikhailov one (Ipsen and Mikhailov, 2002).

Fundamental questions do remain, however. For example, is it possible that the distributed and transient nature of the neural code is simply too elusive for a detailed recognition of behavioural context? Even then, we believe that a combination of experiments aimed at discovering the real synaptic topology (structural connectivity) and a
behavioral NetSAP analysis (functional connectivity) could provide hints for how neurons exchange information. For these reasons, the NetSAP procedure has the potential to become a valuable tool in the analysis of multi-neuronal voltage recordings. Nevertheless, applications on experimental data will be essential in demonstrating NetSAP's usefulness to a wider community.

One area of particular interest to us is the analysis of neural systems affected by neurological disorders. In future work, we intend to explore how neural networks can (or cannot) act in compromised states. Consequently, applications will focus on highlighting aspects that play a key role in neural deregulation, e.g., in the case of epileptic seizure onset.

**NetSAP is coded in CampRaR, an R package which is open source and already available on gital (https://github.com/CallischLab/CampRaR). This software is developed from the Fortran version of the progress index algorithm implemented in CAMPARI. (http://campari.sourceforge.net/index.html).**

**Conflict of interest statement**

The authors declare no conflicts of interest.

**Acknowledgements**

We thank Francesco Cocina for many interesting discussions. This work was supported by grants of the Swiss National Science Foundation (to A.C.) and the Forschungskredit of the University of Zürich (to D.G.).

**Appendix A. Supplementary data**

Supplementary data associated with this article can be found in the online version, at https://doi.org/10.1016/j.jneumeth.2019.01.019.

**References**


Acsády, L., Gönczi, T.J., Freund, T.F., 1996b. Different populations of vasoactive intestinal polypeptide-immunoreactive interneurons are specialized to control pyramidal cells or interneurons in the hippocampus. Neuroscience 73 (2), 317–334.


