



Contents lists available at SciVerse ScienceDirect

Journal of Neuroscience Methods

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

Computational Neuroscience

Detecting cell assemblies in large neuronal populations

Vítor Lopes-dos-Santos*, Sidarta Ribeiro, Adriano B.L. Tort

Brain Institute, Federal University of Rio Grande do Norte, Brazil

ARTICLE INFO

Article history:

Received 6 January 2013

Received in revised form 11 April 2013

Accepted 17 April 2013

Keywords:

Cell assemblies

Principal component analysis

Independent component analysis

Assembly vectors

ABSTRACT

Recent progress in the technology for single unit recordings has given the neuroscientific community the opportunity to record the spiking activity of large neuronal populations. At the same pace, statistical and mathematical tools were developed to deal with high-dimensional datasets typical of such recordings. A major line of research investigates the functional role of subsets of neurons with significant co-firing behavior: the Hebbian cell assemblies. Here we review three linear methods for the detection of cell assemblies in large neuronal populations that rely on principal and independent component analysis. Based on their performance in spike train simulations, we propose a modified framework that incorporates multiple features of these previous methods. We apply the new framework to actual single unit recordings and show the existence of cell assemblies in the rat hippocampus, which typically oscillate at theta frequencies and couple to different phases of the underlying field rhythm.

© 2013 Elsevier B.V. All rights reserved.

Contents

1. Introduction	00
2. Review of the methods.....	00
2.1. Construction of the spike matrix	00
2.2. Determination of the number of cell assemblies	00
2.3. Extraction of cell assembly patterns and estimation of cell assembly activity.....	00
2.3.1. Principal component analysis	00
2.3.2. Assembly vector estimation.....	00
2.3.3. Independent component analysis	00
2.3.4. Other examples	00
3. Real data applications.....	00
4. Discussion	00
4.1. Limitations.....	00
5. Conclusion.....	00
Acknowledgements.....	00
Appendix A. Supplementary data.....	00
References	00

1. Introduction

A main concern in systems neuroscience is to understand how single neurons form functional neuronal circuits ultimately giving rise to complex information processing and behavior. Much of the current knowledge has been derived from the analysis of

the firing rate of single units (Adrian and Zotterman, 1926; Hubel and Wiesel, 1959; Okeefe and Dostrovsky, 1971; Perrett et al., 1982), pairwise correlations (Wilson and McNaughton, 1994), and synchrony between single cells and local field potentials (Siapas et al., 2005). Recent technological advances have opened the possibility of recording large populations of neurons simultaneously (Buzsaki, 2004). These recordings created a new demand for mathematical and statistical tools to analyze the activity of neuronal ensembles as a whole, instead of each unit at a time. New methods with different strategies have been employed, such as template matching (Lee and Wilson, 2002; Louie and Wilson, 2001; Ribeiro et al., 2004), principal component analysis (Chapin and Nicolelis, 1999;

* Corresponding author at: Brain Institute, Federal University of Rio Grande do Norte, Rua Nascimento de Castro, 2155 – Lagoa Nova, Natal, RN 59056-450, Brazil. Tel.: +55 84 32152709.

E-mail address: vitor@neuro.ufrn.br (V. Lopes-dos-Santos).

Lopes-dos-Santos et al., 2011; Peyrache et al., 2010), independent component analysis (Laubach et al., 1999), shuffling methods for detecting repeated firing sequences (Abeles and Gat, 2001; Abeles and Gerstein, 1988; Berger et al., 2010; Gansel and Singer, 2012), and methods based on information theory (Arabzadeh et al., 2004; Quiroga and Panzeri, 2009), among others.

In this review we focus on linear methods for detecting and tracking the activity of cell assemblies embedded in large neuronal populations. Here we define cell assemblies as subsets of neurons with significant co-activation behavior, as proposed by Hebb (1949). We compare three methods that rely on principal component analysis (PCA). Based on their efficiency in simulated neuronal networks, we propose a modified framework that incorporates multiple features of these methods. We aimed to provide an intuitive view of the methods and their possible applications. Rigorous mathematical derivations can be found in the references cited along the text. MATLAB codes and a tutorial for running the methods can be obtained from the corresponding author upon request.

2. Review of the methods

The general procedure can be structured in three main steps: (1) Construction of the spike matrix, where spike trains are binned and normalized; (2) Determination of the number of cell assemblies, where a null hypothesis distribution for cell assembly activity is generated; and (3) Extraction of cell assembly patterns and estimation of cell assembly activity, where co-activation patterns are found and used to track the activity of cell assemblies with single-bin resolution. Step 1 is standard for all methods reviewed here, while steps 2 and 3 may differ. Regarding step 2, here we compare surrogate methods with an analytical threshold recently introduced in Peyrache et al. (2009), and, regarding step 3, we compare previously published methods (Chapin and Nicolelis, 1999; Laubach et al., 1999; Lopes-dos-Santos et al., 2011; Peyrache et al., 2010) and suggest improvements.

2.1. Construction of the spike matrix

All methods studied here use matrix representations of spike trains. In this section we show how to construct the spike matrix.

Top panel of Fig. 1 shows the activity of 20 simulated neurons by means of a spike raster plot, in which each black mark denotes an action potential of a given neuron (vertical axis) in a given time (horizontal axis). The spike matrix is constructed by binning the raster plot and counting the number of spikes elicited by each neuron within each bin (Fig. 1 middle); At this point each matrix entry denotes the number of spikes of a given neuron (rows) in a given time bin (columns). In this example we employ a bin size of 100 milliseconds. Next, the spike count of each neuron (i.e., each row of the matrix) is normalized by z-score transformation (Fig. 1 bottom):

$$z_{ib} = \frac{s_{ib} - \langle s_i \rangle}{\sigma_{s_i}}$$

where z_{ib} is the z-scored spike count of neuron i in time bin b , s_{ib} is the number of spikes of neuron i in bin b , $\langle s_i \rangle$ is the mean spike count of neuron i over all time bins, and σ_{s_i} is the standard deviation of the spike counts of neuron i over bins. Thus, in the z-scored spike matrix each neuron is set to have null mean and unitary variance.

2.2. Determination of the number of cell assemblies

Before extracting assembly patterns it is important to know how many cell assemblies are there to be found. Peyrache et al. (2010, 2009) introduced the use of eigenvalue analysis for determining the statistical significance of assembly patterns. We provide some toy examples below to illustrate this procedure.

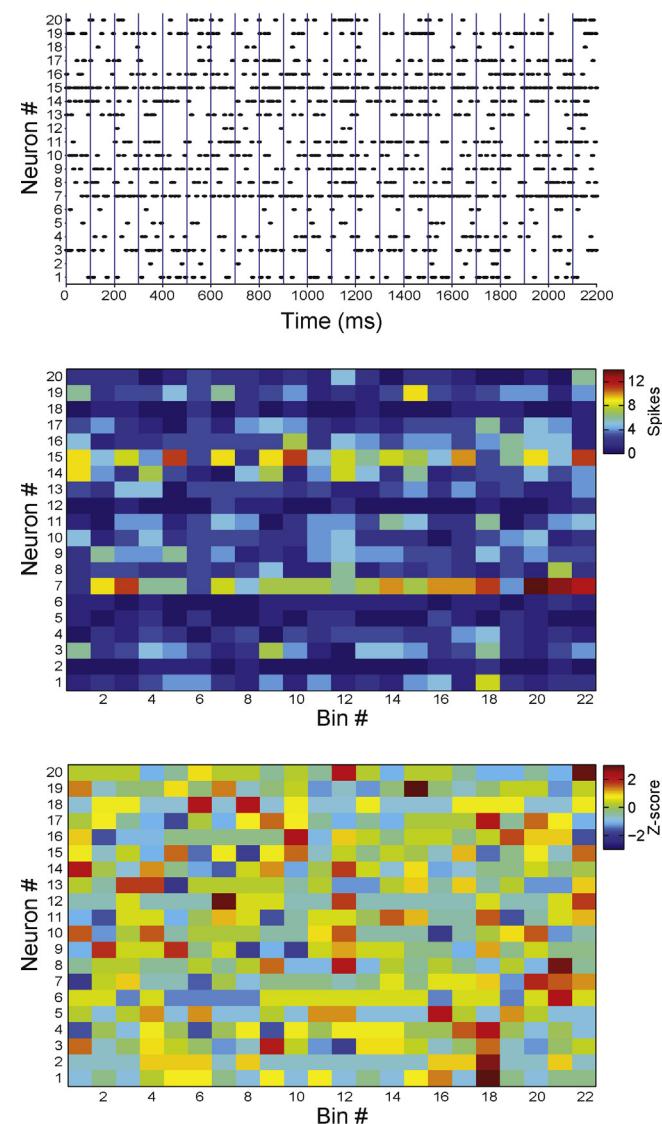


Fig. 1. Pre-processing of spiking activity data. Top panel: spike raster plot of 20 simulated neurons. Dots represent spike times and blue lines denote the boundaries of 100-ms time bins. Middle panel: Non-normalized spike matrix. Each element of the matrix is the number of spikes of a given neuron (row) within a given time bin (columns). Bottom panel: Normalized spike matrix. The spiking activity of each neuron is z-scored. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Fig. 2 shows two scatterplots in which each point represents the normalized spike count of a pair of neurons at the same time bin. The spiking activity of the neuron pair displayed in Fig. 2A is correlated, while the activity of the pair in Fig. 2B is not correlated. Notice that the variance of the data concentrates in a given direction when variables are correlated (Fig. 2A), while the variance is homogeneously distributed across all directions when variables are not correlated (Fig. 2B).

The directions of largest variances of multidimensional data can be found by PCA. By definition, the first principal component (PC) gives the direction of largest variance (that is, when data points are projected onto the direction defined by the first PC, the variance is larger than when projected onto any other direction); the second PC points the direction of largest variance orthogonal to the first PC, and so on. PCs are the eigenvectors of the covariance matrix of the spike matrix. Since spike counts are normalized to have null mean

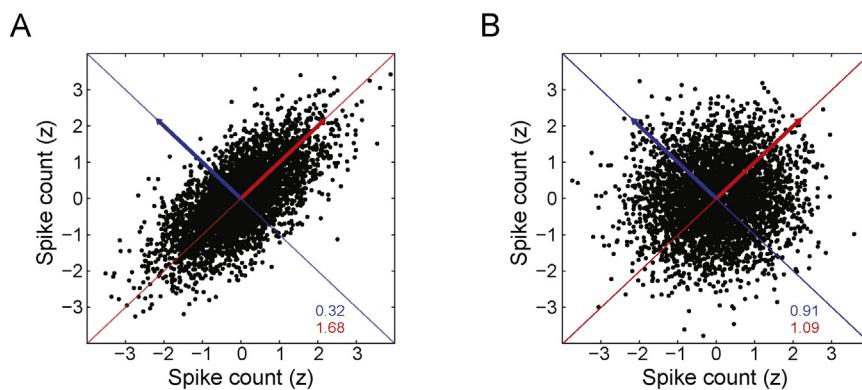


Fig. 2. Principal components and spike count correlations. (A,B) Panels show scatterplots of the spiking activity of two neurons. Each point denotes the z-scored spiking activity of a pair of simulated neurons (100-ms time bins were used; each neuron is represented by an axis). Red and blue arrows denote first and second principal components, respectively. Colored numbers inform the variance of the data in the axes spanned by the principal components (colored lines; same color convention). Panel A shows a correlated pair of neurons and panel B shows an uncorrelated pair. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

and unitary variance, in our case the covariance matrix is equal to the correlation matrix, and can be calculated as:

$$C = \frac{ZZ^T}{N_{\text{columns}}}$$

where Z is the (z-scored) spike matrix, T the transpose operator, and N_{columns} is the number of time bins of Z . Thus, the element at the i -th column and j -th row of C is the linear correlation between neurons i and j . Since C is necessarily real and symmetric, it follows from the spectral theorem that it can be decomposed as:

$$C = \sum \lambda_i x_i x_i^T$$

where x_i is the i -th eigenvector of C (that is, the i -th PC of Z) and λ_i its corresponding eigenvalue. Moreover, the outer product $x_i x_i^T$ is the projection matrix onto the direction of x_i , and λ_i is the variance of the data along the same axis.

In Fig. 2 the red and blue lines represent the first and second PCs, respectively, and their eigenvalues (i.e., variances) are shown with same color labels. As mentioned above, when spike trains are uncorrelated, as in Fig. 2B, the variance of the data projected onto any direction is roughly the same. In fact, they would be exactly equal if infinite samples were analyzed. Random fluctuations due to undersampling produce a direction with slightly larger variance that is detected as the first PC. In the case of correlated activity (Fig. 2A), the variance is significantly more concentrated in the direction of the first PC. These observations can be generalized for higher dimensional signals: data variance will be significantly larger in a given direction if there is a linearly correlated group of cells. Following this reasoning, a possible strategy to estimate the number of cell assemblies (subsets of neurons with correlated activity) is to find the number of PCs of the spike matrix with significantly large eigenvalues.

Peyrache et al. (2009, 2010) proposed the use of the Marčenko–Pastur distribution as a null hypothesis for the existence of cell assemblies. Marčenko and Pastur (1967) demonstrated that the eigenvalues of the correlation matrix of a normal random matrix M with statistically independent rows follow a probability function described by:

$$p(\lambda) = \frac{q}{2\pi\sigma^2} \frac{\sqrt{(\lambda_{\max} - \lambda)(\lambda - \lambda_{\min})}}{\lambda},$$

with $q = N_{\text{columns}}/N_{\text{rows}} \geq 1$, where σ^2 is the variance of the elements of M (in our case $\sigma^2 = 1$ due to z-score normalization), N_{columns} is the number of columns and N_{rows} the number of rows. λ_{\max} and

λ_{\min} are the maximum and minimum bounds, respectively, and are calculated as:

$$\lambda_{\min}^{\max} = \sigma^2(1 \pm \sqrt{1/q})^2$$

This probability function has finite support given by the interval $\lambda_{\min} \leq \lambda \leq \lambda_{\max}$. Thus, if the rows of M are statistically independent, the probability of finding an eigenvalue outside these bounds is zero. In other words, the variance of the data in any axis cannot be larger than λ_{\max} when neurons are uncorrelated. Therefore, λ_{\max} can be used as a statistical threshold for detecting cell assembly activity (Peyrache et al., 2010). That is, the number of eigenvalues above λ_{\max} can be used to estimate the number of cell assemblies in the spike matrix.

Although this theoretical bound was derived for large matrices, it is also a robust threshold for smaller matrices (Lopes-dos-Santos et al., 2011; Plerou et al., 2002). Nevertheless, Peyrache et al. (2010) proposed the use of a finite sample bias correction based on the Tracy–Widom distribution (Tracy and Widom, 1994). In addition, it should be noted that although the Marčenko–Pastur distribution was proved for random matrices whose entries are derived from Gaussian distributions, empirical simulations show that this distribution also provides a good bound for eigenvalues of matrices composed by independent rows (in our case, uncorrelated neurons) originated from other random processes (Biroli et al., 2007; Lopes-dos-Santos et al., 2011; Seba, 2003). Moreover, we never observed “false positive” eigenvalues in simulations employing Poisson neurons, i.e., the number of eigenvalues significantly larger than chance was always equal to or less than the real number of assemblies.

Fig. 3A displays examples of correlation matrices computed from three simulated spike matrices (the main diagonal was set to zero for clearer visualization); the empirical eigenvalue distributions are plotted along with the Marčenko–Pastur distribution (black line). Neurons were modeled as Poisson processes with mean rate of 1 spike/bin. In the example shown on top, all neurons fired independently of each other. Note that all eigenvalues lie within the bounds of the analytical distribution. In the second example, a subset of four neurons is correlated (Neurons # 1–4), as can be seen from the correlation matrix. In this case, one eigenvalue lies above the analytical distribution. The last example has two subsets of correlated neurons (Neurons # 1–5 and 6–8), and two eigenvalues are larger than the theoretical threshold. Thus, the upper bound of the Marčenko–Pastur distribution represents a statistical threshold for assembly activity, i.e., the number of cell assemblies in the network is estimated by the number of eigenvalues above this threshold (Lopes-dos-Santos et al., 2011).

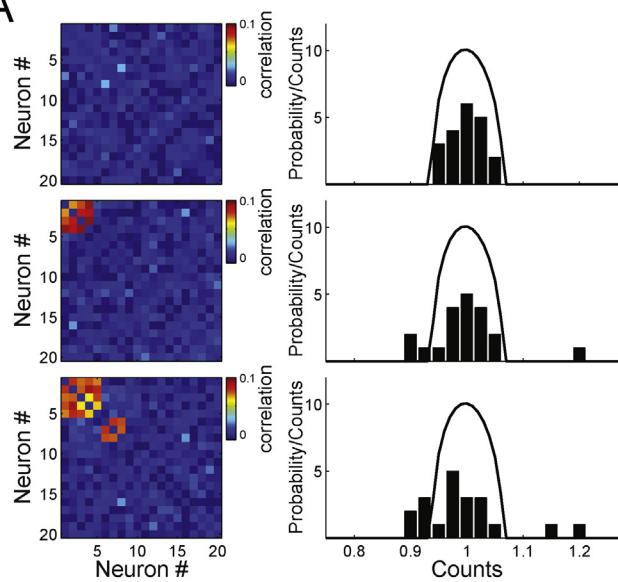
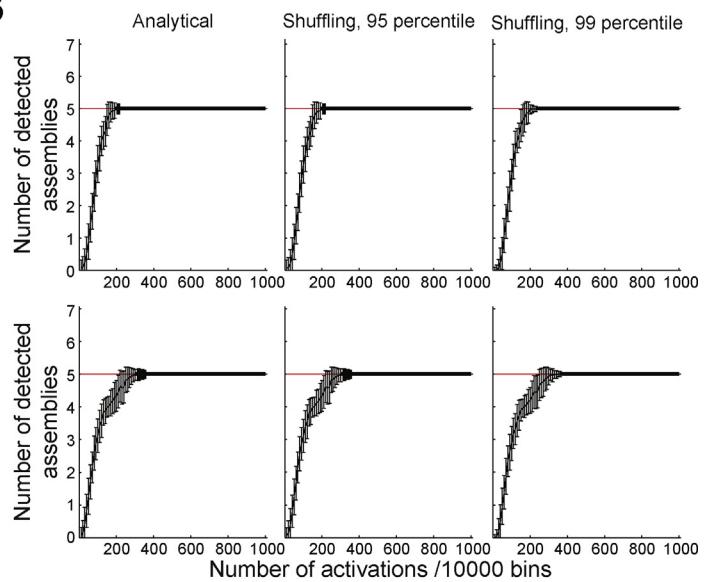
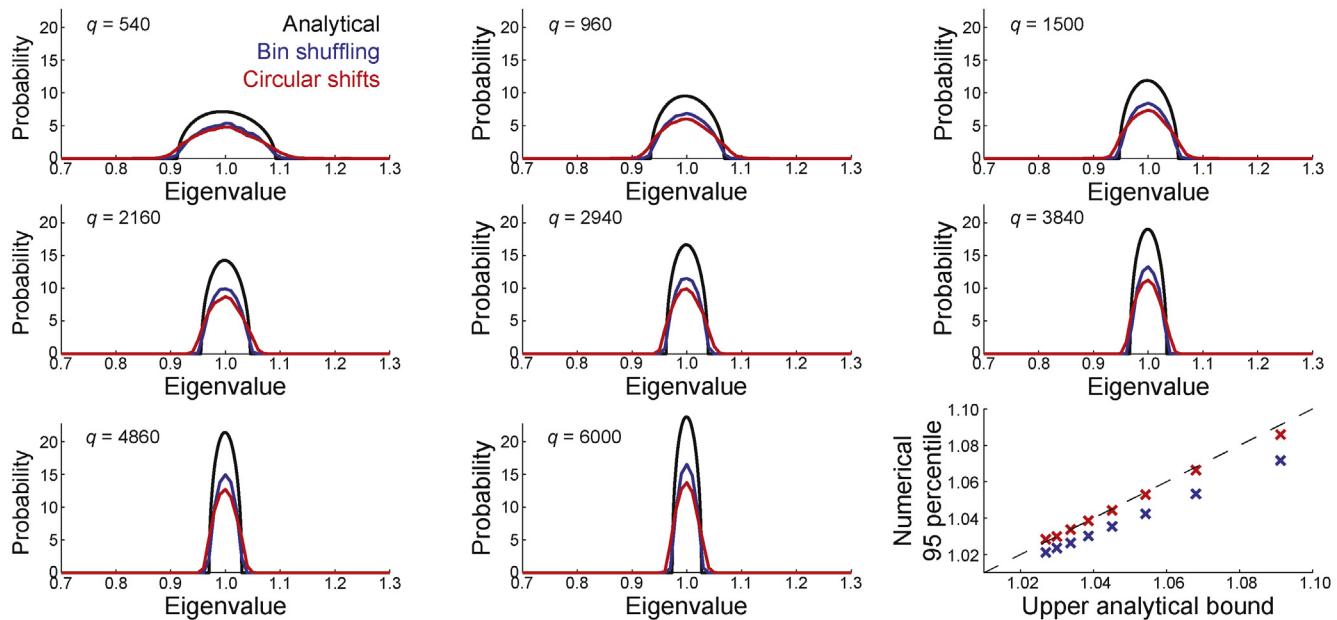
A**B****C**

Fig. 3. Estimating the number of cell assemblies. (A) An analytical threshold for assembly detection. Left panels show correlation matrices of three simulated neuronal populations (20 neurons were modeled as Poisson processes with mean rate of 1 spike/bin for 8000 time bins). Main diagonals set to zero for clearer visualization. Right panels show the theoretical Marčenko-Pastur distribution (black line) along with the empirical distribution of eigenvalues of the correlation matrices. The upper bound of the analytical distribution can be used as statistical threshold for assembly detection. In the first simulation all neurons fired independently (top left panel), and all eigenvalues lie within the analytical distribution (top right panel). In the second simulation four neurons (neurons #1, #2, #3 and #4) were correlated (middle left panel). Note that one eigenvalue lies above the upper bound of the analytical distribution (middle right panel). In the third simulation two subgroups presented significant correlation (bottom left panel), and two eigenvalues lie above the upper bound of the analytical distribution (bottom right panel). (B) Top panels show the mean number of assemblies detected in simulated spike matrices as a function of assembly activation rate (150 simulations were run for each activation rate; error bars denote standard deviation). Each spike matrix comprised 32 neurons modeled as Poisson processes; five cell assemblies composed by five non-overlapping neurons were programmed. Left panel displays results when the upper bound of the analytical distribution is used as threshold for detecting assembly activity. Middle and right panels display results when the threshold was determined by a surrogate method, in which the bins of each neuron are shuffled independently and the maximum eigenvalue of the correlation matrix is calculated. This procedure is repeated 3200 times in order to generate a null hypothesis distribution for maximal eigenvalues. A statistical threshold is defined as a percentile of this distribution (95% for middle panel and 99% for right panel). Bottom panels show the same as above but for overlapping assembly membership. See text for further details. (C) Eigenvalue distribution for independent activity. The activity of an actual hippocampal neuron was binned into 10-ms windows and spike matrices were constructed by two methods: in the first method, neurons were created by randomly shuffling bins; in the second method, neurons were created by random circular time shifts of the original spike train, thus preserving autocorrelation and ISI counts. No cell assembly was programmed (but see Fig. 10). For each method and values of q (number of time bins/number of neurons, see text), eigenvalue distribution was estimated by analyzing 1000 random spike matrices composed by 20 neurons. The analytical Marčenko-Pastur distribution is also displayed. Rightmost panel in the bottom row shows correspondence between the 95 percentile of the numerical distributions (y-axis, color coded) and the upper bound of the analytical distribution (x-axis). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

An alternative to the use of the Marčenko–Pastur distribution is to define a statistical threshold based on surrogate data. This procedure entails the shuffling of time bins for each neuron independently in order to destroy their temporal relations while maintaining the distribution of spike counts unchanged. The eigenvalues of correlation matrices obtained from shuffled spike matrices can be used to construct a null distribution; for example, the eigenvalues of the original spike matrix that are larger than a certain percentile (e.g., 95% or 99%) of the distribution of the maximal eigenvalues computed from shuffled data are regarded as significant (that is, in this framework each surrogate matrix contributes its maximal eigenvalue to the chance distribution, which is a more conservative approach than considering all surrogate eigenvalues).

In order to compare the analytical and shuffling methods, we simulated spike matrices with 32 neurons and 10,000 bins. Neurons were modeled as Poisson processes with unitary mean. Additionally, η bins of each neuron were randomly chosen as *activation bins*, in which the spike count was derived from a uniform random distribution from 0 to 6 spikes, thus likely above the overall mean. In order to simulate assemblies, we set five subsets of neurons (i.e., five cell assemblies) to have coincident activation bins (*assembly activations*). Note that although assembly members increase firing together, each assembly activation is unique since the firing of assembly members is stochastic. The top panels of Fig. 3B display results for spike matrices with the following assembly composition: assembly 1 neurons: #1, #2, #3, #4; assembly 2: #5, #6, #7, #8; assembly 3: #9, #10, #11, #12; assembly 4: #13, #14, #15, #16; assembly 5: #17, #18, #19, #20. Lines represent the mean over 150 simulations and bars indicate standard deviation. This example shows that the analytical and the shuffling methods converge to the correct number of assemblies after a certain number of activations have occurred. The bottom panels of Fig. 3B show similar results but for a set of assemblies in which each assembly shares neurons with at least two other assemblies (assembly 1 neurons: #1, #2, #3, #4, #5; assembly 2: #5, #6, #7, #8, #9; assembly 3: #9, #10, #11, #12, #13; assembly 4: #12, #13, #14, #15; assembly 5: #15, #16, #17, #1). The analytical and the shuffling methods also converge to the correct number of assemblies in this case, although more assembly activations are necessary than in the case of non-overlapping assemblies.

Next, we compare analytical and numerical distributions of random eigenvalues for data with a more realistic structure. Instead of using Poisson neurons, we created 20 neurons based on the spike trains of a neuron recorded from the rat hippocampus. Spikes were binned into 10-ms windows (120,000 bins, or 20 min, in total) and new spike trains were created by two surrogate methods. The first of them is the same bin shuffling method used for Fig. 3B. The second is achieved by means of a circular random shift. That is, for each neuron a time point is randomly chosen and used to divide the spike train in two non-overlapping segments which switch position. This method creates surrogate neurons with same rate and variance of the original neuron, and virtually the same autocorrelation and inter-spike-interval (ISI) distribution.

We estimated the distribution of eigenvalues for independent activity by computing the average eigenvalue histogram across 1000 simulations for each surrogate method. Since the distribution is affected by the ratio $q = N_{\text{columns}}/N_{\text{rows}}$, we repeated the procedure for matrices with different number of bins (N_{columns}) while keeping the number of neurons fixed. Fig. 3C displays the eigenvalue distribution estimated by each surrogate method along with the Marčenko–Pastur distribution for each q . Finally, we compare the 95 percentile drawn from the numerical distributions with the upper bound of the analytical distribution in the bottom right panel of Fig. 3C. Surprisingly, the 95 percentile of the circular shift method matches very closely the upper bound of the analytical distribution.

The 95 percentile of the bin shuffling procedure was lower than the upper analytical bound for all values of q studied. Altogether, the results indicate that the theoretical bound derived from the Marčenko–Pastur distribution is well suited for tracking the number of cell assemblies, since surrogate methods give similar results but are computationally demanding. More detailed parametrical studies on the Marčenko–Pastur distribution can be found in Lopes-dos-Santos et al. (2011). It is nevertheless advisable to compare the analytical and shuffling methods when working with actual spike recordings, since the specific statistical properties of a given dataset may be different from the simulated examples employed here. In this regard, a valuable feature of the surrogate method is that surrogate spike matrices can be generated in such a way that neurons preserve their ISI distribution and autocorrelation, thus preserving more statistical properties of the original data.

2.3. Extraction of cell assembly patterns and estimation of cell assembly activity

In linear models, the activity of a cell assembly is assumed to be a linear combination (a weighted sum) of the spike activity of all neurons. Thus, each co-activation pattern (also called assembly pattern) is a vector that attributes weights to each neuron. For each pattern, cell assembly neurons can be identified as those with the largest weights.

In mathematical terms, the activity of a cell assembly in a given time bin b can be defined as:

$$R_b = \sum_{i=N_{\text{neurons}}} w_i z_{ib} = w^T Z_b,$$

where N_{neurons} is the number of neurons, z_{ib} is the z-scored activity of neuron i at the time bin b , w_i is the weight of the corresponding neuron in the assembly, w is a column vector containing the weights of all neurons in the assembly and Z_b is the b -th column of the spike matrix Z .

This section deals with the problem of finding co-activation patterns w for each cell assembly. We study three different methods for obtaining w , which are based on PCA (Chapin and Nicolelis, 1999; Peyrache et al., 2010), assembly vectors (AV) (Lopes-dos-Santos et al., 2011), and independent component analysis (ICA) (Laubach et al., 1999).

2.3.1. Principal component analysis

Nicolelis et al. (1995) employed PCA to study the activity of thalamo-cortical assemblies during the occurrence of *mu* rhythm in rats, using the PCs themselves as co-activation patterns to track ensemble activity (Chapin and Nicolelis, 1999). Later, Peyrache et al. (2010) introduced a template-match protocol by which assembly patterns identified during learning epochs are used to assess assembly activity during pre- and post-learning sleep episodes. Employing this framework, they showed that subsets of medial prefrontal cortex neurons that were co-active during rewarded runs in a T-mask task also co-activated during subsequent slow-wave sleep, mainly during hippocampal ripple events (Peyrache et al., 2009). More recently, Benchenane et al. (2010) used the PCA approach to show that the co-firing of assembly neurons occurs at a preferred phase of the theta rhythm.

Roughly speaking, PCA extracts uncorrelated orthogonal components from the spike matrix, which are assumed to represent co-activation patterns. The framework is illustrated in Fig. 4. A time interval of a spike matrix and its correlation matrix are shown in Fig. 4A, with 20 neurons simulated as Poisson processes for 8000 bins with 1 spike/bin mean rate. Additionally, non-stationarities were introduced in the activity of each neuron by randomly choosing 250 bins as activation bins. In these bins, the firing rate

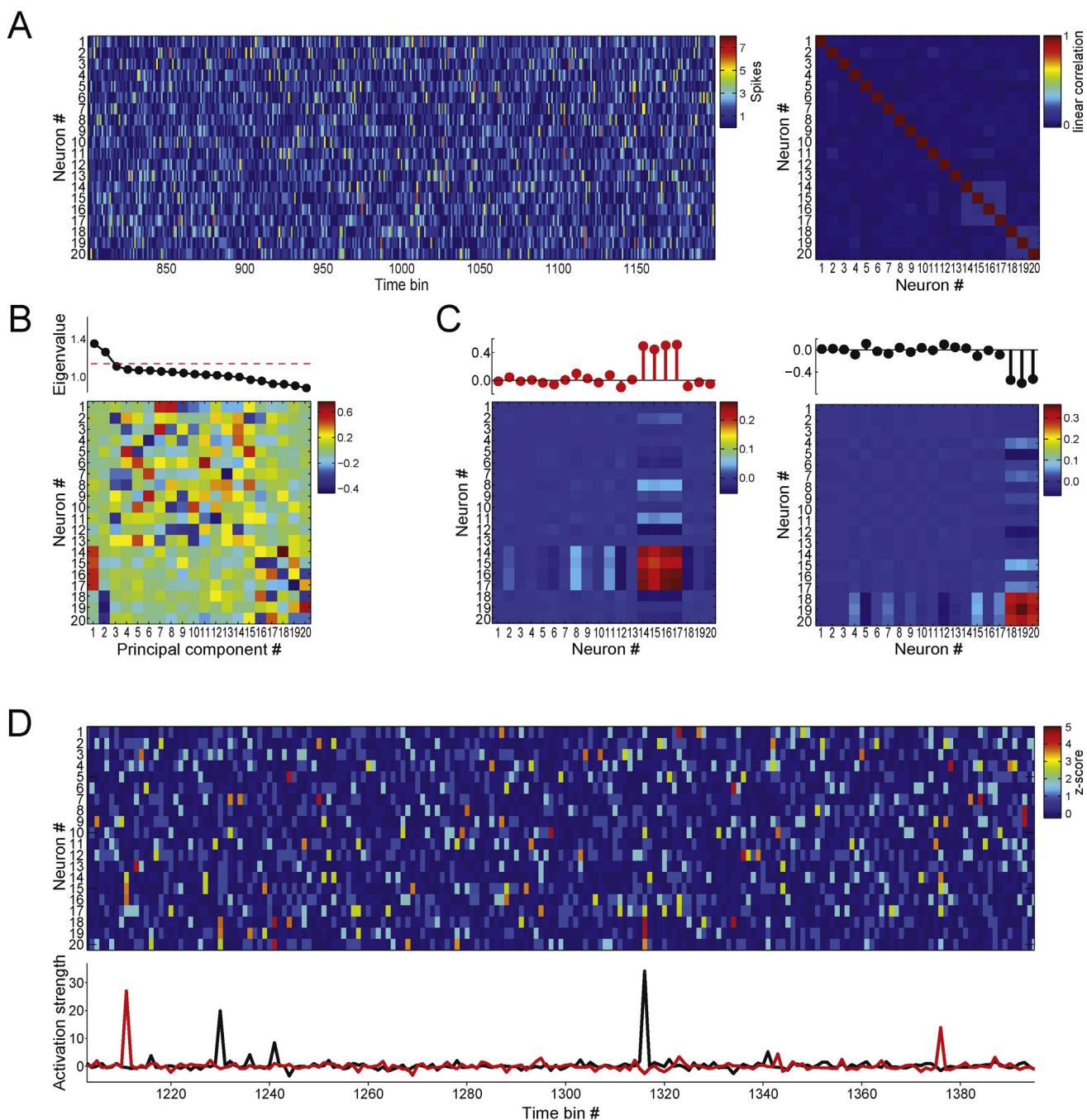


Fig. 4. Detecting cell assemblies and tracking their activity using principal component analysis. (A) Left panel: Non-normalized spike matrix composed by 20 simulated neurons (total simulation time: 8000 bins). Each neuron was modeled as a Poisson process with mean rate of 1 spike/bin. Additionally, 250 activation bins (around 3% of the total number of bins) were randomly chosen for each neuron: in these bins, the spike count was derived from a uniform distribution between 0 and 6 spikes. In order to simulate assemblies, subsets of neurons were set to have coincident activation bins. In this example, two assemblies are present: assembly 1 (neurons #14, #15, #16, #17) and assembly 2 (neurons #18, #19 and #20). Right panel: Correlation matrix. (B) Principal components and eigenvalues. Top panel shows the eigenvalues of the correlation matrix shown in A. Red dashed line denotes the analytical significance threshold. Two eigenvalues are significant in this case. Bottom panel shows principal components associated with the eigenvalues shown above. (C) Top panels display the principal components associated with the eigenvalues significantly above chance. Lower panels show the corresponding outer products of each principal component, which are projectors of the spike matrix used for computing cell assembly activity with single-bin resolution. (D) Top panel shows a representative time interval of the same spike matrix as in A after z-score normalization. Lower panel shows the activity of the detected assemblies estimated by the principal components shown in C. Note that the peaks of assembly activity correspond to co-activations of the corresponding assembly members. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

was uniformly distributed between 0 and 6 spikes. Two assemblies were programmed: assembly 1, composed by neurons #14, #15, #16, #17; and assembly 2, composed by neurons #18, #19 and #20. Fig. 4B displays the PCs along with their associated eigenvalues. The red dashed line represents the significance threshold provided by the Marčenko–Pastur distribution. In this example,

two eigenvalues are significant. Fig. 4C shows the significant PCs (top panels) along with their outer product (bottom panels). Note that these patterns isolate the activity of the cell assemblies, i.e., they attribute large weights to specific assembly members; in fact, in this example the first and second PCs represent the co-activation patterns of assemblies 1 and 2, respectively.

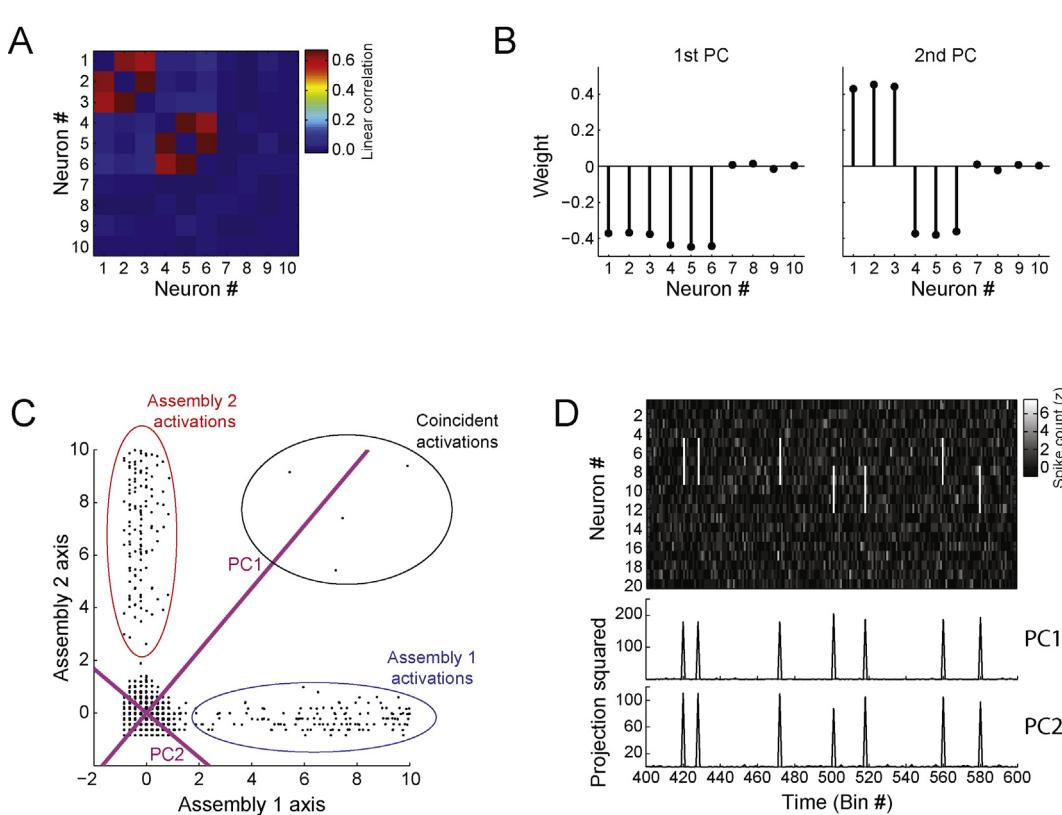


Fig. 5. Examples of limitations of assembly pattern estimation by principal components. (A) Correlation matrix of a simulated spike matrix with 8000 bins and 10 neurons. Two assemblies were programmed with the following composition: assembly 1, neurons #1, #2, #3; assembly 2, neurons #4, #5, #6. Neurons and assemblies were simulated as in Fig. 4. Each assembly was active for 200 bins, where the spike count of assembly members assumed a uniform distribution from 0 to 30. (B) Significant principal components of the correlation matrix shown in A. (C) Projection of the columns of the spike matrix onto the subspace spanned by the ideal assembly patterns. Ideal assembly patterns have same weights for assembly members and null weights for other neurons. Points representing activations of assembly 1 and 2 are circumscribed by blue and red ellipses, respectively. Bins in which both assemblies were active are circumscribed by a black ellipse. Axes spanned by the principal components are represented by magenta lines. (D) Top panel: A time interval of a simulated spike matrix with 20 neurons showing activations of two assemblies with overlapping membership (assembly 1, neurons #6, #7, #8, #9; assembly 2, neurons #8, #9, #10, #11). Bottom panels: assembly activities estimated when principal components are used to define assembly patterns. Adapted from Lopes-dos-Santos et al. (2011). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

The next step is to use the assembly patterns to compute the time course of each cell assembly activity with single-bin resolution. The activity of each assembly can be estimated by projecting the columns of the spike matrix onto the axis spanned by the corresponding assembly pattern (in this case, assumed to be the PCs). The projection of a column Z_b onto an axis can be calculated as:

$$\text{proj} = PZ_b,$$

where the projection matrix P is defined as:

$$P = w \otimes w = ww^T,$$

where \otimes is the outer product operator and w is a unitary vector that spans the axis. The vector w is the assembly pattern. At each time bin, the length of the projection is a measure of the similarity between the activity of the whole population and the assembly pattern. This length is calculated by taking the inner product between the assembly pattern w and Z_b (i.e., a weighted sum of the normalized spike counts). Alternatively, the strength of the projection was defined by Peyrache et al. (2010) as the square of the projection length, which can be calculated as:

$$R_b = Z_b^T P Z_b,$$

where P is the outer product of the assembly pattern. Hereafter we will adopt this definition of assembly activity. The projection matrices for the assembly patterns in the last example are shown in Fig. 4C bottom panels. Of note, since we are concerned with

co-activation patterns, the main diagonal of P is set to zero, which assures that isolated activations of assembly members do not contribute to R (Lopes-dos-Santos et al., 2011; Peyrache et al., 2010). Fig. 4D shows a time interval of the spike matrix Z along with the estimated time course of the activity of the two cell assemblies. Note that the peaks of the black time-course are coincident with the co-activation of neurons composing assembly 2, while the red time-course peaks with the activations of assembly 1.

Although the original PCA method can be efficient for extracting assembly patterns (Benchenane et al., 2010; Chapin and Nicolelis, 1999; Nicolelis et al., 1995; Peyrache et al., 2009, 2010), this framework has important limitations. As mentioned above, linear correlations among a subset of neurons lead to larger variance in a given axis that can be found by PCA. However, if two assemblies concentrate a similar amount of variance in their corresponding axes, the first PC will represent the average of the two assemblies and not an individual assembly. This issue is illustrated in Fig. 5. A spike matrix was simulated as before but with 10 neurons; each assembly was active in 200 bins, where the spike count of the assembly neurons assumed a uniform distribution from 0 to 30 spikes. Assembly 1 was composed by neurons #1, #2 and #3, and assembly 2 by neurons #4, #5 and #6. Since the two assemblies have the same number of neurons and activation bins, they have similar mean strength; consequently, the axes that describe these assemblies have similar variance. The correlation matrix for this simulation and its significant eigenvectors are shown in Fig. 5A,B. Notice that the first PC attributes large weights to neurons of assemblies 1 and 2, whereas

ideal assembly patterns should have large weights (with the same sign) only for the members of the corresponding assembly. Fig. 5C shows the projection of the data onto the space spanned by the ideal assembly patterns (a large value in a given axis means that the corresponding assembly was active at that time bin). The axes of the significant PCs are displayed as magenta lines. Note that length of the projections onto the axis defined by the first PC are similar for both assemblies; therefore, in this example the activity time course estimated by the outer product of the first PC peaks when either assembly 1 or 2 is active (not shown).

This problem gets more critical when assemblies share neurons (Lopes-dos-Santos et al., 2011). A shared neuron should ideally have large weight in both assembly patterns; however, this cannot be represented by PCs since PCs are orthogonal by definition. In order to illustrate this issue, a spike matrix was simulated with 20 neurons modeled as Poisson processes with unitary mean for 8000 time bins. For each assembly, 40 bins were randomly chosen as assembly activations, in which assembly members elicited 6 spikes. Two assemblies were included: assembly 1 composed by neurons #6, #7, #8 and #9, and assembly 2 composed by neurons #8, #9, #10, #11 and #12. Fig. 5D shows a time interval of the spike matrix along with assembly activities computed from the outer product of the significant PCs. Notice that both time courses peak whenever assembly 1 or 2 is active.

In an attempt to overcome these limitations, Lopes-dos-Santos et al. (2011) have proposed a method that searches for clusters of neurons in the subspace spanned by the PCs. We review this framework in the following section.

2.3.2. Assembly vector estimation

The PC weights (loadings) carry information about cell assembly membership, in the sense that large weighted neurons are part of an assembly while neurons with near zero weights are not. However, how large should the weight of a neuron be in order to make it part of an assembly? The assembly vector (AV) estimation method was developed to identify the membership of cell assemblies in addition to extracting assembly patterns (Lopes-dos-Santos et al., 2011). The framework is summarized in Fig. 6. A spike matrix composed of 32 neurons was simulated as in the example of Fig. 4. Two assemblies were programmed: assembly 1 with neurons #27, #28, #29, #30; and assembly 2 with neurons #29, #30, #31, #32. Fig. 6A shows the correlation matrix (left panel) along with its eigenvalues distribution (right panel). As in the previous method, the number of cell assemblies is determined by means of the Marčenko–Pastur distribution.

Note that even when the PCs mix assembly patterns, the threshold derived from the Marčenko–Pastur distribution still holds (Lopes-dos-Santos et al., 2011). This happens because activations of different assemblies cannot lie in the same axis (otherwise they would be a single assembly), and therefore variance increases in the same number of axes as the number of assemblies. Moreover, note that when two neurons are correlated, the variance concentrates in a given axis (the assembly axis) and decreases in the orthogonal axis (Fig. 2A), which leads to one eigenvalue significantly above chance distribution and another significantly below. This can be generalized for groups of neurons: for example, if four neurons participate in one assembly, the variance will increase in one axis and decrease in other three. Following this reasoning, Lopes-dos-Santos et al. (2011) have shown numerically that the number of eigenvalues outside the theoretical distribution can be used to estimate the number of assembly neurons. Thus, in the AV framework the number of neurons composing at least one assembly is estimated by counting the total number of eigenvalues outside the Marčenko–Pastur distribution, i.e., above or below the theoretical bounds (Lopes-dos-Santos et al., 2011). These bounds are indicated in the right panel of Fig. 6A by dashed red lines. In this example,

six eigenvalues lie outside the distribution (2 above and 4 below), which matches the number of neurons participating in cell assemblies.

In order to identify cell assembly neurons, the columns of the correlation matrix are projected onto the subspace spanned by the PCs associated with significantly large eigenvalues (i.e., above the upper limit of the Marčenko–Pastur distribution):

$$N_i = P_{AS} C_i,$$

where C_i is the i -th column of the correlation matrix and P_{AS} is the projection matrix of the assembly space, defined as:

$$P_{AS} = \sum_i P_{ci} P_{ci}^T = P_c P_c^T,$$

where P_{ci} is the i -th significant PC and P_c is a matrix containing all significant PCs (columns).

In this framework, this subspace is called *Assembly Space* (Fig. 6B), and the columns of the correlation matrix projected onto this subspace (N_i) are called *neuron vectors*.

Since six eigenvalues lie outside the theoretical distribution, the six neurons with largest neuron vectors are regarded as *significant neurons*. Note that neurons that have similar correlation patterns will have neuron vectors clustered together in the Assembly Space. On the other hand, if neurons have orthogonal correlation patterns their neuron vectors will tend to be separated in this space. Following this reasoning, the inner products between all neuron vectors and those of the significant neurons is a measure of similarity between their co-activation patterns. The matrix containing these inner products is called *Interaction Matrix* (Fig. 6C), which can be formally defined as:

$$M_{i,j} = N_i^T N_j,$$

where $M_{i,j}$ is its element at row i and column j , N_i is the neuron vector of neuron i and N_j is the neuron vector of significant neuron j .

In principle the same could be done by the direct use of the columns of the correlation matrix. However, it is reasonable to assume that assembly patterns can be described by a linear combination of the significant PCs (as in Fig 5B,C). Thus, PCs are used to filter the data and the inner products are taken inside the Assembly Space.

Ideally, the distribution of inner products is bimodal, and there is a threshold that separates large and low values (Fig 6D, left) and can be used to digitalize the interaction matrix ('1' is assigned to values above the threshold and '0' to others). Automated algorithms, such as standard k-means, can find such a threshold. A clustering algorithm is then applied to the digitalized interaction matrix to identify assembly membership (Fig. 6D, right), under the constraint that there are two assemblies (which correspond to the number of eigenvalues above the Marčenko–Pastur distribution in this example). Finally, the AVs are defined as the mean of all neuron vectors exclusive to an assembly (Fig. 6E):

$$AV_a = \frac{1}{n_a} \sum_i N_i,$$

where N_i is the neuron vector of an exclusive neuron of the corresponding assembly and n_a is the number of exclusive neurons.

For example, the vector of assembly 1 is the mean between the vectors of neurons #27 and #28. Fig. 6F compares assembly patterns defined by PCA and by the AV method. While the PCA method fails to sort apart the assembly patterns, the AV method properly identifies the two patterns. Thus, the projectors of a spike matrix constructed from the outer product of AVs will correctly track the

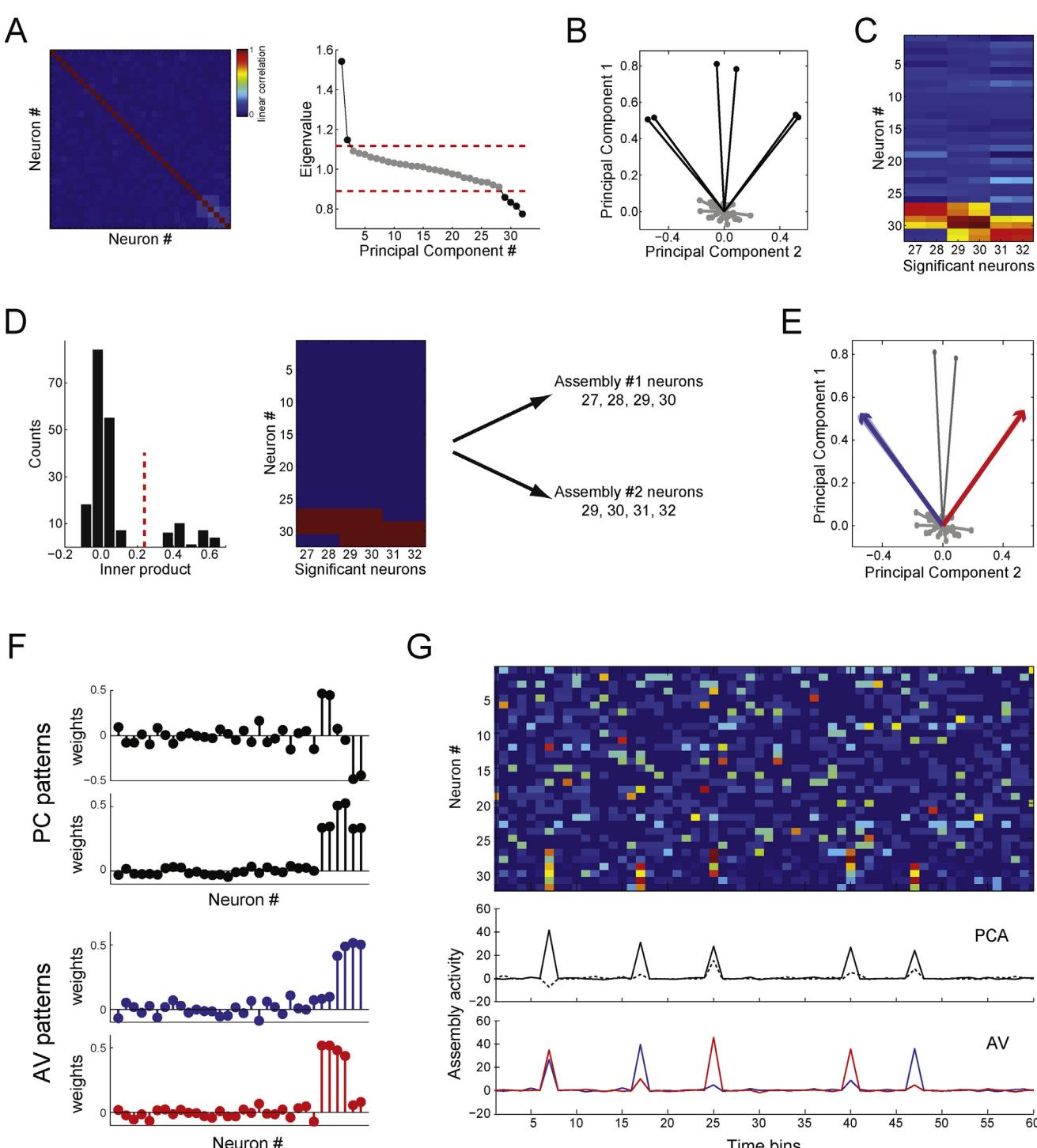


Fig. 6. Detecting and tracking cell assembly activity using *assembly vectors*. (A) Left: Correlation matrix. For this example, 32 neurons were simulated (mean 1 spike/bin; total simulation time: 10,000 bins). Each neuron had 500 activation bins (0 to 6 spikes uniformly distributed). Two assemblies were included (assembly 1: neurons #27, #28, #29, #30; assembly 2: neurons #29, #30, #31, #32). Right: Distribution of eigenvalues of the correlation matrix. The two significance bounds (upper and lower) of the analytical distribution are shown (red dashed lines). The number of eigenvalues above the upper bound denotes the number of cell assemblies, as in the previous method. Additionally, the number of eigenvalues outside the chance distribution (above and below) is used as a lower bound for the number of neurons participating in cell assemblies. (B) Projection of all correlation matrix columns in the subspace spanned by the significant principal components (Assembly Space). In this framework, these projections are referred to as neuron vectors. Neurons with largest projections in the Assembly Space are regarded as *significant neurons*. (C) Interaction matrix: The inner product of all neuron vectors with those of the significant neurons is computed. (D) Membership estimation. Left: histogram of the entries of the interaction matrix. The value that best separates the bimodal distribution is obtained by a one-dimensional k-means algorithm and used to threshold the interaction matrix (values above the threshold are set to '1', values below to '0'). Right: Interaction matrix shown in C after thresholding (*binary interaction matrix*). Assembly members are estimated by means of an iterative clustering algorithm applied to the binary interaction matrix (for details, see Lopes-dos-Santos et al., 2011). (E) Estimation of assembly vectors. Each assembly vector (blue and red arrows) is defined as the mean over all vectors in Assembly Space associated with neurons exclusively pertaining to an assembly. Similar to the previous method, the assembly vectors then form assembly patterns from which activity projectors (i.e., outer products) are constructed. (F) Assembly patterns estimated by principal components (PC) and assembly vectors (AV). Note in this example that the PC method mixes assembly members, while the AV method estimates assembly members more reliably. (G) Estimation of assembly activity. A time interval of the spike matrix is displayed (top) along with assembly activities estimated by PC (middle) and AV (bottom) methods. Note that the activities estimated by AVs better isolate activations of different assemblies. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

activity of individual cell assemblies, which cannot be achieved by the original PCA method in this case (Fig. 6G).

Despite being an improvement over the original PCA framework, the AV method has limitations. First, the AV method by definition requires each assembly to have at least one exclusive member. Moreover, the separation between small and large inner products in the Assembly Space (Fig. 6C) is not always clear. This is critical because k-means splits the data into two groups irrespective of whether they are clearly separated or not, which may lead to spurious separation. Below we describe another method for extracting assembly patterns that also relies on first analyzing the spike matrix using PCA, and then proceeds with ICA to identify assembly patterns even when cell assemblies have no exclusive neuron.

2.3.3. Independent component analysis

Laubach et al. (1999, 2000) introduced ICA to quantify interactions among neurons. In this section we review basic ICA theory and show how this tool can be used to estimate assembly patterns. Roughly speaking, ICA is a method to extract statistically independent non-Gaussian components of a multivariate signal. Rigorous mathematical derivations can be found elsewhere (Comon, 1994; Hyvärinen and Oja, 1997; Jutten and Herault, 1991). Assuming that the time course of the activity of an assembly is described as:

$$R = w^T Z,$$

note that we only know Z (the spike matrix) and must estimate both w and R .

ICA relies on the Central Limit Theorem, which states that (under certain conditions) the mix of two independent random variables is more Gaussian than the original variables. In other words, any linear combination of independent assembly activities is expected to be more Gaussian than the individual assembly activities themselves. From this reasoning, ICA iteratively rotates the axes of an initial random vector w in order to minimize some Gaussianity measure of $y = w^T Z$. As noted by Hyvärinen and Oja (2000), there are several measures of Gaussianity (or non-Gaussianity). Here we employ the fastICA algorithm (Hyvärinen, 1999; Hyvärinen and Oja, 1997) which uses approximating negentropy, defined as:

$$J_{\text{appr}}(y) = \sum_i [E\{G_i(y)\} - E\{G_i(v)\}]^2,$$

where v is a Gaussian variable with zero mean and unitary variance, and G_i are the non-quadratic functions:

$$G_1(u) = \frac{1}{a} \log \cosh au, \quad G_2(u) = -\exp(-\frac{u^2}{2}),$$

where $1 \leq a \leq 2$. For further details on approximating negentropy and other Gaussianity measures see Hyvärinen and Oja (2000).

Using ICA, Laubach et al. (1999, 2000) studied how neuronal ensembles in the motor cortex of rats evolved as the animals learned to perform a reaction-time task. In their work, however, the authors assumed that the number of eigenvalues larger than 1 represented the number of significant independent components (and hence the number of assembly patterns considered for further analyses). This procedure considerably overestimates the number of cell assemblies in the network, which, as reviewed above, should be taken as the number of eigenvalues above the upper limit of the Marčenko–Pastur distribution. We observed through simulations that the incorrect determination of the number of assemblies may lead to spurious results (not shown). This becomes more critical if ICA is performed in the original data space, i.e., with no prior dimensionality reduction. In this case, assembly patterns are broken into two or more independent components. In addition, since ICA by design does not extract co-activation patterns, some independent components may represent the activity of single neurons

that do not fire following a Gaussian distribution. To avoid these issues, here we suggest a modification of the original ICA approach which is achieved by incorporating the use of the Marčenko–Pastur distribution in order to estimate the number of cell assemblies. This modified approach involves first reducing the dimensionality of the spike matrix Z by projecting Z onto the subspace spanned by the significant PCs, and then computing the independent components through the fastICA algorithm. In simple terms, the new framework first finds the significant PCs and then rotates them to match the *ideal* assembly patterns.

Fig. 7 illustrates the performance of PCA and modified ICA frameworks in extracting non-orthogonal assembly patterns. A spike matrix was simulated as before, but with 20 neurons. Assembly membership was: assembly 1, neurons #11, #12, #13, #14; assembly 2, neurons #14, #15, #16, #17; and assembly 3, neurons #18, #19, #20. Note that neuron #14 participates in assemblies 1 and 2. The correlation matrix is shown in Fig. 7A, along with its eigenvectors and eigenvalues. Since there are three cell assemblies, three eigenvalues lie above the theoretical threshold. The significant PCs and the independent components are shown in Fig. 7B. Note that while the independent components correctly attribute larger weights to the proper assembly members, the PCs fail to correctly segregate assembly patterns. Fig. 7C shows a time interval of the spike matrix along with the time course of assembly activity estimated by both methods. Note that the *blue* assembly computed by the PCA method peaks for the activations of both assemblies 1 and 2, while assembly activities computed from independent components correctly track individual assembly activations.

2.3.4. Other examples

For further comparisons, we applied the three methods described above to two other examples. In both cases, assemblies were highly distributed and thus difficult to segregate.

The first example is shown in Fig. 8. A spike matrix with 40 neurons (1 spike/bin mean rate) and 20,000 time bins was simulated; each neuron had 800 activation bins. As shown in Fig. 8A, the assembly identities were: assembly 1, neurons #1, #2, #3, #4, #5, #13, #24; assembly 2, neurons #4, #5, #6, #7, #8, #9, #18, #29; assembly 3, neurons #9, #10, #11, #12, #13, #14, #30; assembly 4, neurons #3, #14, #15, #16, #17, #18, #19, #20; assembly 5, neurons #18, #20, #21, #22, #23, #24, #25; assembly 6, neurons #2, #6, #10, #15, #21, #31. Fig. 8B shows the correlation matrix; the eigenvalue distribution is shown in Fig. 8C. Six eigenvalues lie above the upper bound and twenty-two below the lower threshold, which matches the number of simulated assemblies (6) and the number of assembly neurons (28).

Fig. 8D shows how each method estimated the assembly patterns. While PCA mixed the assembly patterns, the AV and ICA methods correctly attributed larger weights with same signs for assembly members within a pattern. For instance, independent component #1 and assembly vector #1 represented assembly 6 (green assembly in Fig. 8A). Since activation bins are independent among assemblies, assembly activities should ideally not correlate. Fig. 8E shows the histogram of linear correlation coefficients obtained for all possible pairs of assembly activities. Note that the PCA method estimates assembly activities that are more correlated than when estimated by the AV and ICA methods. The left panels in Fig. 8F show the time course of estimated assembly activities. Each black circle denotes actual assembly activations (some are stronger than others due to increased but random spike counts in activation bins). Note that the activities estimated by the AV and ICA methods peak at the actual assembly activations, while the time course of assembly activity estimated by PCA does not provide robust results for all assemblies. The right panels in Fig. 8F show matrices whose entries are the correlation coefficients for pairs of assembly activities estimated by the different methods (these values were used

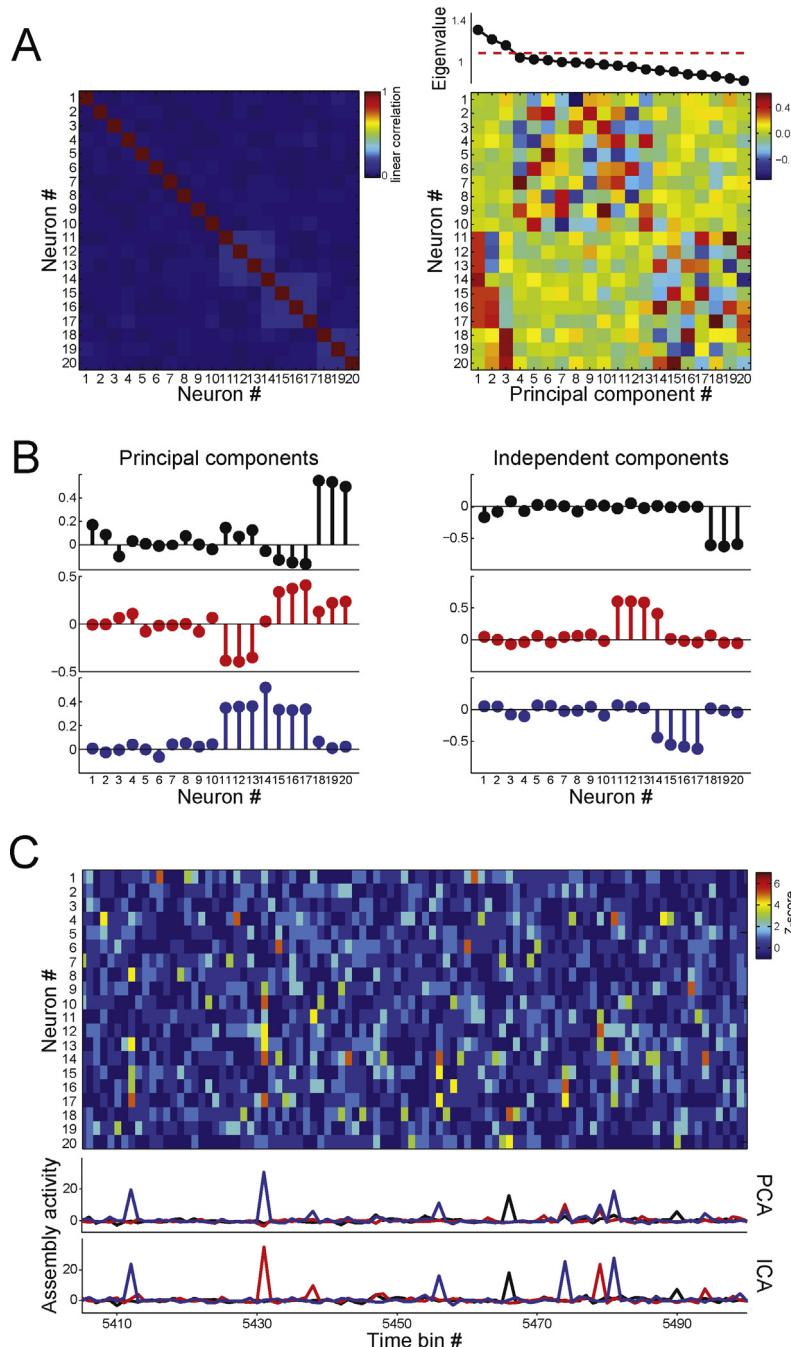


Fig. 7. Detecting and tracking cell assembly activity using independent component analysis. (A) Left: Correlation matrix of a spike matrix (not shown) with 20 neurons. In this example, 3 assemblies were present in the network (assembly 1: neurons #11, #12, #13, #14; assembly 2: neurons #14, #15, #16, #17; assembly 3: neurons #18, #19, #20). Right: Associated eigenvalues and principal components. Note that three eigenvalues are significant. (B) Assembly patterns estimated by principal components (left) and by independent components (right). Note that assembly patterns are better segregated in the latter case, i.e., each assembly pattern has large weights for its corresponding members. (C) Representative time interval of a spike matrix (top) and assembly activity time course estimated by principal component analysis (PCA) and independent component analysis (ICA) (bottom). Different colors represent projections obtained using different assembly patterns (as before, projectors are defined by the outer product of assembly patterns). Note that the blue trace in the PCA framework peaks for activations of both assemblies 1 and 2. On the other hand, assembly activity estimated by ICA properly segregates the three assemblies. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

in Fig. 8E); again, notice lower correlations for the AV and ICA methods.

Fig. 9 shows a case in which the AV method fails. The spike matrix was simulated as in the last example but with a slight difference in assembly composition: neuron #31 was removed from assembly 6, and thus this assembly had no exclusive neuron (Fig. 9A). The correct number of assemblies is still detected by the analysis of the eigenvalues of the correlation matrix (Fig. 9B,C).

The assembly patterns estimated by all three methods are shown in Fig. 9D. The AV method fails to identify the pattern of assembly 6; in fact, the assembly vector #6 is a mixture of other assembly vectors. This is reflected in the histogram of the correlation coefficients between pairs of estimated assembly activities (Fig. 9E); in contrast to the previous example, now both the PCA and AV methods exhibit cases of high correlations. Fig. 9F shows the misleading estimation of the activity of assembly 6 by the AV method more clearly:

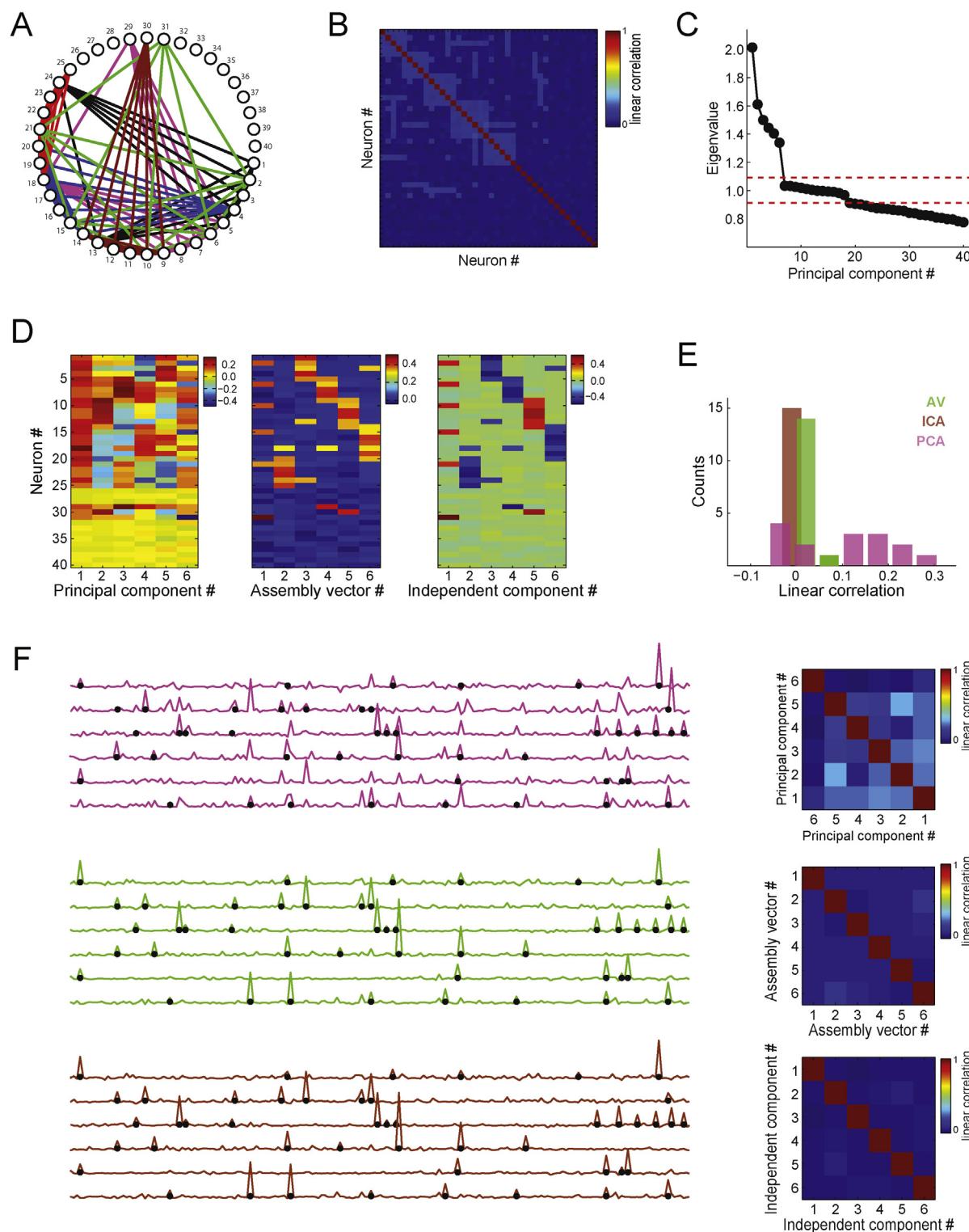


Fig. 8. PCA, ICA, and AV method performance for highly distributed assemblies. 40 neurons were simulated (mean 1 spike/bin) for 20,000 time bins. Each neuron had 800 activation bins (0 to 6 spikes uniformly distributed). (A) Simulated cell assemblies. Each circle represents a neuron. Neurons in the same assembly are connected by lines with same color (assembly 1: neurons #1, #2, #3, #4, #5, #13, #24; assembly 2: neurons #4, #5, #6, #7, #8, #9, #18, #29; assembly 3: neurons #9, #10, #11, #12, #13, #14, #30; assembly 4: neurons #3, #14, #15, #16, #17, #18, #19, #20; assembly 5: neurons #18, #20, #21, #22, #23, #24, #25; assembly 6: neurons #2, #6, #10, #15, #21, #31). (B) Correlation matrix. (C) Eigenvalue distribution. Six eigenvalues lie above the upper threshold and twenty-two lie below the lower threshold. (D) Assembly patterns estimated by the three methods. (E) Distribution of linear correlation coefficients among the time course of assembly activities when estimated by each of the three methods. Notice large correlations among cell assembly activities when estimated by the PCA framework. (F) Top panels: representative interval of the time course of assembly activities estimated from principal components (left). Black circles denote true (simulated) assembly activations. Note that assembly activations estimated by PCA do not perfectly match simulated activations. The right panel shows pairwise linear correlations between the time courses of different assemblies when estimated by PCA. Middle and bottom panels: Same as above but for ICA and AV methods, respectively. In these cases the peaks of the estimated cell assembly activities correspond well to the simulated activations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

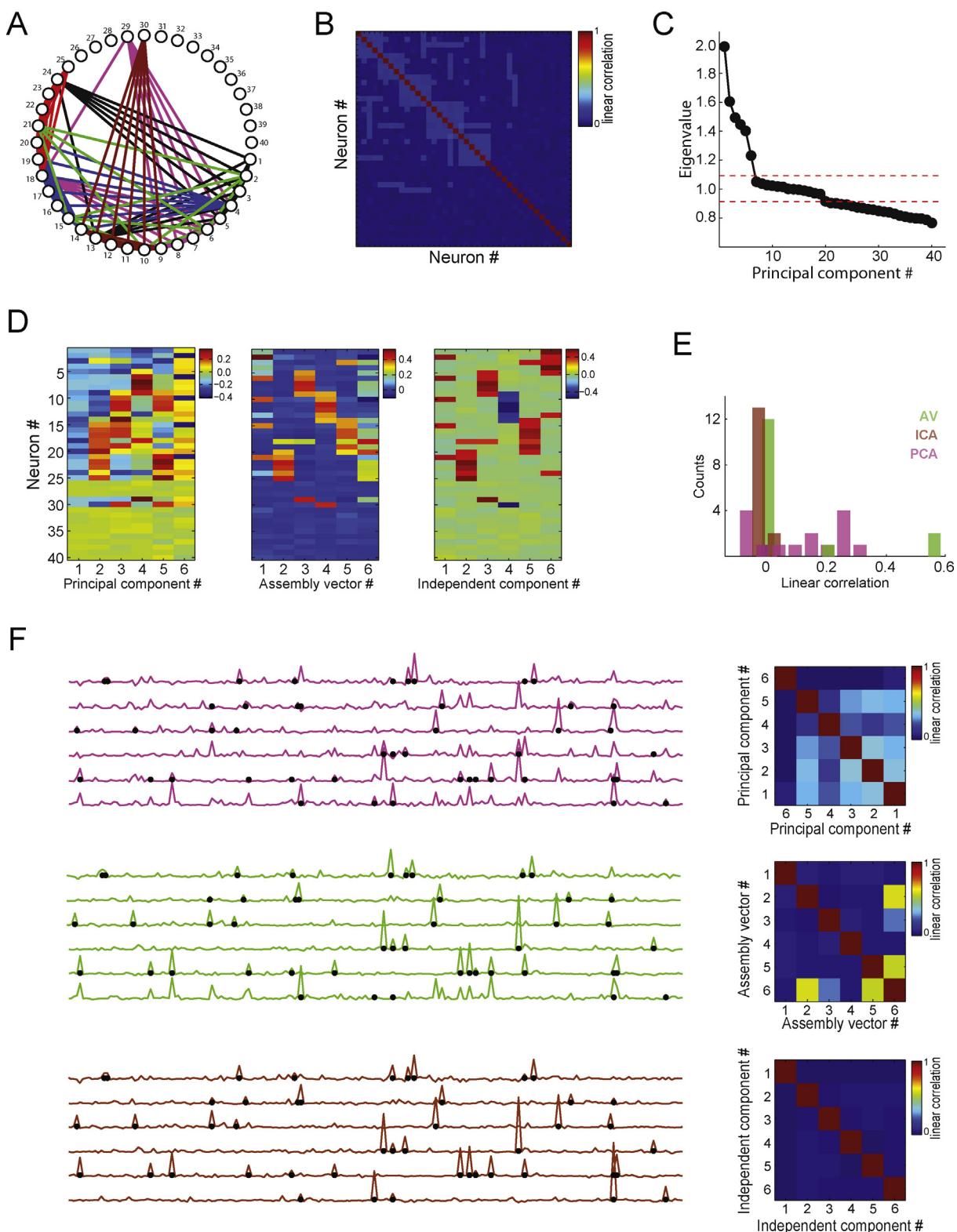


Fig. 9. PCA, ICA and AV method performance for cell assemblies with no exclusive neurons. 40 neurons were simulated as in Fig. 8. (A) Simulated cell assemblies. Same as Fig. 8, except that neuron #31 does not pertain to assembly 6, which consequently has no exclusive neuron (that is, all neurons in assembly 6 participate in at least one other assembly). (B) Correlation matrix. (C) Eigenvalue distribution. (D) Assembly patterns estimated by each method. Note that for this case the AV framework does not find assembly 6. (E) Same as Fig. 8E. (F) Same as Fig. 8F. Note that the AV method does not properly track the activity of assembly 6, which is estimated as a mixture of multiple assemblies. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

note that while the activities of the first five assemblies are well estimated, the activity of the sixth assembly is mostly a mixture of the third and the fifth cell assemblies (left panel), as can also be seen from the correlation coefficients shown in the right panel.

Finally, notice that, as in Fig. 8, the ICA method still provides a good estimation for all cell assemblies.

We next tested the ICA method in a more realist condition. A spike matrix with 20 neurons was generated by circular shifting

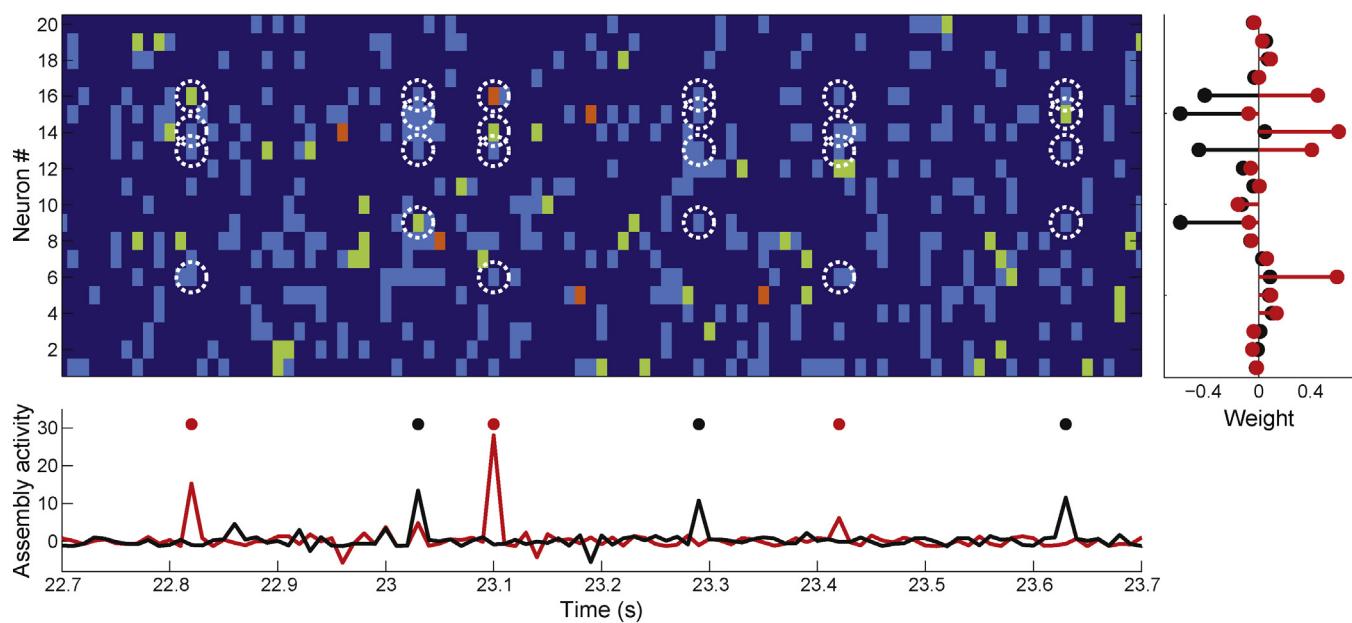


Fig. 10. Application of the ICA-based framework to realistic simulations. Each neuron of the spike matrix was created from a circular shift of actual spike times of a hippocampal neuron (same procedure as in Fig. 3C). Two cell assemblies were simulated by including common spikes to assembly members, which were drawn from independent Poisson processes for each assembly (mean rate: 2 Hz). Assembly 1 was composed by neurons #9, #13, #15 and #16; assembly 2 by neurons #6, #13, #14 and #16 (notice 50% of overlap between assemblies). Top left panel shows a time interval of the spike matrix; circles mark co-firing of assembly members. Right panel shows assembly patterns extracted by the ICA-based framework. Bottom panel shows estimated time course of assembly activity computed from the assembly patterns. Events of the Poisson processes used to generate assembly activations are indicated by colored dots. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

actual spike trains as in Fig. 3C. In order to simulate a cell assembly, common spike times generated by a Poisson process with mean rate of 0.02 spikes/bin (2 Hz) were added to the spike trains of a group of neurons. We simulated two overlapping cell assemblies: assembly 1, composed by neurons #6, #13, #14 and #16; and assembly 2, composed by neurons #9, #13, #15 and #16. Accordingly, the ICA-based method extracted two assembly patterns, which corresponded to the programmed cell assemblies (Fig. 10 top right panel). We show a representative time interval of the spike matrix in the top left panel of Fig. 10. Dashed circles show co-activations of assembly members. The bottom panel shows estimated assembly activities along with colored dots that indicate events of the Poisson processes used to generate assembly activations. Note that the time courses of assembly activities estimated by the ICA method peak accordingly.

3. Real data applications

In this section we analyze spike trains of 19 single-units recorded from the CA1 region of the hippocampus of a freely moving rat exploring an open field. The data was downloaded from the Collaborative Research in Computational Neuroscience (CRCN) webpage (<http://crcns.org/>). Detailed information on recordings, behavior and surgery can be found in Mizuseki et al. (2009). The spike matrix was constructed using 10-ms bins. To estimate the number of cell assemblies, the eigenvalues of the correlation matrix were compared to the theoretical threshold derived from the Marčenko–Pastur distribution. Assembly patterns were extracted from the spike matrix using the modified ICA-based framework described above. We here further adapted this framework by increasing its time resolution. Since neuronal firing can occur in different bins even when spikes are separated by a few milliseconds, we estimated the time course of assembly activity by directly applying the projector operators to spike trains after smoothing them with a Gaussian kernel (maximum value = 1, standard deviation = 12 ms). In this framework, therefore, assembly activities can

be computed with the same time resolution as the local field potential (LFP), and coupling between these signals can be evaluated.

A time interval of the spike raster plot of the neuronal population is shown in the top left panel of Fig. 11A. The right panels show three assembly patterns extracted from the associated spike matrix. Neurons with large weights in a same pattern are displayed with same color. Estimated assembly activities are shown in the plot underneath the raster plot. Notice that only the blue assembly is active in the depicted period, and that assembly activity peaks when most of its neurons are co-active; in contrast, isolated firing of the same neurons does not affect assembly activity. The ongoing LFP is shown in the bottom panel. Examples from the same recording session in which the magenta and red assemblies are active are shown in Supplementary Fig. 1.

We used standard analysis techniques to study the activity of CA1 cell assemblies identified by this framework. Although it is usually assumed that neurons with higher firing probability at the same phase of an ongoing oscillation are part of the same cell assembly, non-overlapping assemblies with same phase preference has been previously shown by Harris et al. (2003). In accordance with this result, here we also observed that independent assemblies can have similar phase preferences. Fig. 11B shows the autocorrelation of the three assemblies highlighted in Fig. 11A along with their corresponding theta-phase distributions of assembly activations. Theta-phase distributions were obtained by (1) filtering the LFP into the theta range (6–10 Hz), (2) computing the instantaneous theta phases using the Hilbert transform, (3) localizing the phases associated with the peaks of assembly activity, and (4) expressing the phases by means of a circular histogram. Notice that the blue and magenta assemblies have similar preferred phase, while the red assembly was most active in a different theta phase. All assemblies were significantly coupled to ongoing theta oscillations (Rayleigh test, $p < 10^{-6}$).

We next computed two theta-phase distributions for each cell assembly neuron. The first distribution was obtained by only taking into account spikes occurring during assembly activations (defined

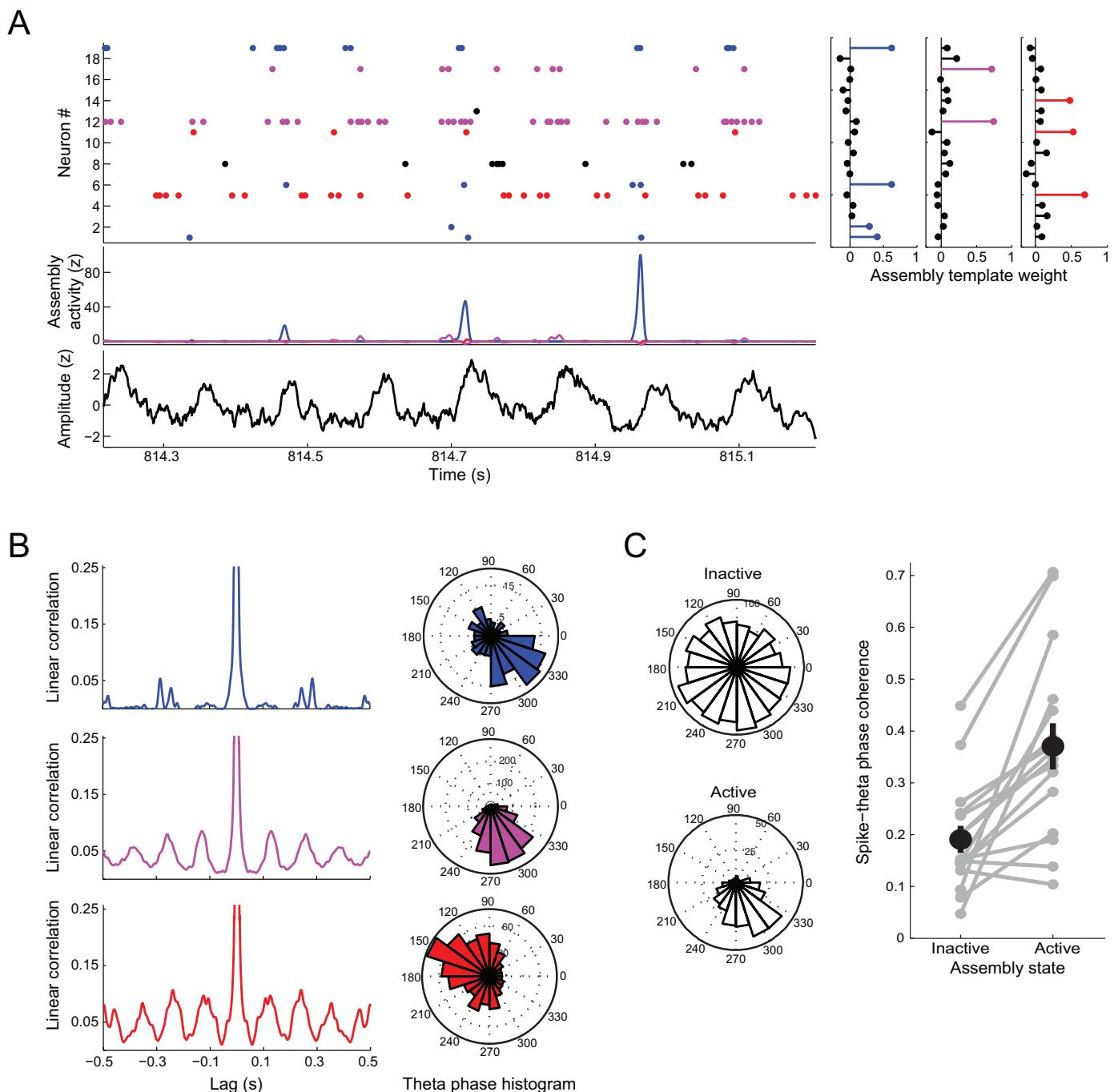


Fig. 11. Real data applications. (A) Top-left panel shows a time interval of the spike raster plot of 19 single-units recorded from the hippocampal CA1 pyramidal layer. Top-right panels show three assembly patterns detected by the ICA-based method. The neurons with largest weights in each assembly were color-coded. Estimated assembly activities are shown below the raster plot. The bottom panel shows the ongoing local field potential (LFP). Assembly activities and LFP are z-scored. Note that the co-activation of the blue neurons is indicated by the peaks of the corresponding assembly activity. (B) Left panels show autocorrelations of the same cell assemblies as in A. Right panels show the theta-phase histogram of assembly activations. Note clear coupling between assembly activity and hippocampal theta oscillations. (C) Left panels show the theta phase distribution of spikes for an assembly neuron inside (bottom) and outside (top) assembly activations. Right panel shows mean spike-theta phase coherence of assembly neurons when the cell assembly was inactive or active (black circle; error bars denote standard deviation). Individual cases are shown in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

as assembly activity above its mean + 2 standard deviations). The remaining spikes were used to construct the second distribution. A representative example is shown in Fig. 11C: the bottom left panel shows the distribution of spike phases during assembly activations, whereas the top left panel shows the phase distribution for spikes occurring when the cell assembly was inactive. Note that the neuron in this example is strongly coupled to the theta rhythm

when its assembly is active. To compute a group result, for each assembly neuron we calculated the spike-phase coherence inside and outside assembly activations. Spike-phase coherence was defined as the length of the mean phasor $e^{i\varphi}$, where φ is a theta-phase in which a spike occurred (Siapas et al., 2005). The right panel of Fig. 11C shows the spike-phase coherence for each assembly neuron in gray; the black circle denotes the means for

each case and bars indicate standard deviations. These results show that neurons couple more strongly to the theta rhythm when firing in synchrony with other assembly members ($p=0.0039$, Wilcoxon rank sum test).

4. Discussion

Recent advances have opened the possibility of testing influential theories on network functioning and its relation to behavior and memory. Here we reviewed three linear methods for computing cell assembly activity. We began by introducing the original PCA approach, which was the first employed to track cell assemblies (Nicolelis et al., 1995), and is still useful nowadays in generating insights about organized spiking activity (Benchenane et al., 2010; Peyrache et al., 2009). Next, we showed that assembly patterns estimated by PCs have important limitations that can be overcome the AV framework (Lopes-dos-Santos et al., 2011), which identifies cell assemblies based on correlation patterns within the subspace spanned by the significant PCs. Finally, we presented a modified version of a previously described ICA approach (Laubach et al., 1999), which incorporates the statistical threshold provided by the Marčenko–Pastur distribution. We showed that this method provides even more robust estimates of assembly patterns than the AV method.

The large interest in this field led to the development of analytical tools able to identify high-order correlations that can potentially replace pair-wise analyses (Stevenson and Kording, 2011). However, several methods can only be applied to small networks due to combinatorial explosion (Berger et al., 2010; Grun et al., 2002; Tetko

and Villa, 2001). In order to overcome this problem, some authors have proposed methodologies that detect high-order correlations ignoring how cells are involved in the coalitions (Louis et al., 2010; Staude et al., 2010a,b). The methods reviewed here can be applied to large neuronal populations, and neuronal activity is assessed as a whole through the use of eigenvalue analysis. Additionally, the Marčenko–Pastur distribution opens the possibility of using an analytical and reliable statistics instead of surrogate methods employed in previous frameworks (Abeles, 2009; Abeles and Gat, 2001; Abeles and Gerstein, 1988; Humphries, 2011; Maldonado et al., 2008; Shmuel et al., 2006). In addition to being computationally demanding, a problem inherent to the use of surrogates is the fact that there is no consensus about which statistical properties should be preserved in these control data (Berger et al., 2010; Grun, 2009). For example, we have shown here that surrogate methods that preserve ISI counts and auto-correlations provide different thresholds for statistical significance than less conservative shufflings (Fig. 3C).

We note that although ICA assumes a linear model for cell assemblies, it employs nonlinear equations in order to quantify Gaussianity, which can be greatly optimized by the fastICA algorithm (Hyvärinen and Oja, 1997). In fact, while the ICA-based framework is more computationally demanding than the PCA and AV methods, it is still faster than most nonlinear algorithms. In general, frameworks employing more complex mathematics require extensive data crunching that often yield results difficult to interpret by non-specialized researchers. The methods studied here are intuitive and generate results easy visualize in raw data, which should favor their comprehension and broad use by the scientific community.

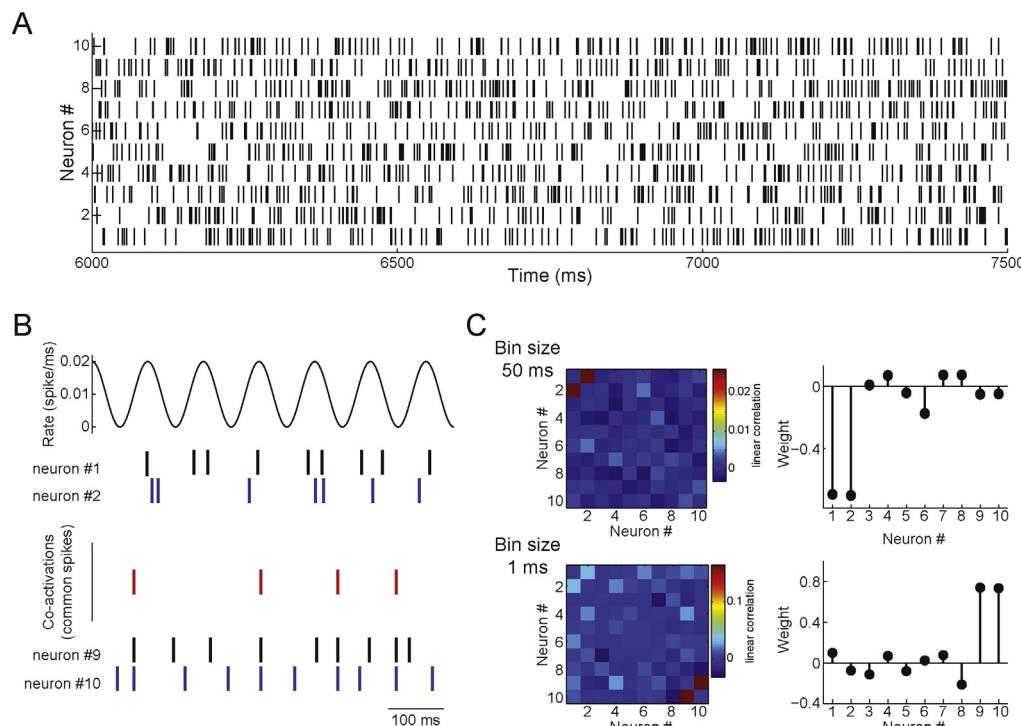


Fig. 12. The influence of bin size on assembly detection. (A) Spike raster plot of ten simulated neurons. Neurons #1 and #2 are Poisson processes with a common varying mean rate following a 10-Hz sinusoid (minimum and maximum rate: 0 and 0.02 spike/ms). Neurons #3 to #10 are independent Poisson processes with 0.01 spike/ms mean rate. In addition, spikes drawn from a Poisson process with 0.3-Hz mean rate were added to neurons #9 and #10 in order to simulate a cell assembly. (B) Top panels show the varying mean rate of neurons #1 and #2 along with their spike times. Note that neurons are more active around the peak of the sinusoid. Bottom panels show the Poisson events used to create common spikes for neurons #9 and #10, along with their spike times. (C) Top panels show the correlation matrix when 50-ms bins are used to construct the spike matrix (left) and the assembly pattern extracted by the ICA-based method (right). Note that the pattern corresponds to neurons #1 and #2 that co-activate at a time scale defined by the sinusoid. Bottom panels show the same results but for when 1-ms bins are used to construct the spike matrix. Note that at this time resolution only co-activations of neurons #9 and #10 are detected. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

4.1. Limitations

None of revised methods is optimal, and there is room for further improvements. For example, ICA cannot find assemblies if they follow a Gaussian distribution. Moreover, non-linear spike correlations may produce spurious results in assembly detection methods that depend on PCA. Roughly speaking, when the scatterplot of two random variables has a curved shape, the eigenvalue analysis may reveal two dimensions with large variance. By consequence, PCA-based methods would falsely identify two assemblies in the network, while only one pair of neurons is correlated. It should be noted that other methods, such as the one described in Humphries (2011), can be adapted to other measures of spike train similarity besides linear correlations.

It is also relevant to note that the bin size used to create the spike matrix (Fig. 1) defines the time scale of the detected assemblies, and thus different bin sizes can yield different results. We illustrate this possibility in Fig. 12. In this example, neurons in assembly 1 fire at a Poisson rate defined by a 10-Hz sinusoid (Fig. 12B top), while neurons in assembly 2 fire synchronous spikes based on the realizations of a Poisson process, despite also emitting independent random spikes outside assembly activations (Fig. 12B bottom). Using a 1-ms bin size to construct the spike matrix leads to the detection of assembly 2 (Fig. 12C bottom), which neurons spike in precise synchrony; assembly 1 is not detected because of the jitter in the spike times of its neurons. On the other hand, employing a 50-ms bin size leads to the exclusive detection of assembly 1 (Fig. 12C top), since at this time scale assembly 1 neurons are co-active and the random spikes of assembly 2 neurons occurring when the assembly is not active mask assembly 2 activations. Thus, the extracted co-activation patterns have a time scale defined by the bin size, which should be considered when interpreting results.

Finally, one should note that the reviewed methods do not disambiguate between stimulus-driven and internally generated spike correlations. For instance, spike correlations can be detected when neurons have similar tuning curves, irrespective of whether they are wired together or not. Thus, a pair of CA1 pyramidal cells with overlapping place fields can be potentially identified as composing an assembly depending on the time scale (bin size) employed in the analysis. However, we note that this feature is inherent to the definition of a cell assembly, which is usually taken as a group of cells that fire together (and collectively represent an object, a sensation, an action, etc.), independently of what causes the correlated firing. In our view, whether neurons presenting correlated activity due to common sensory inputs should or not be considered a cell assembly is a matter of definition.

5. Conclusion

Linear methods are computationally low demanding, and yet quite efficient for tracking cell assembly activity. Hebb's seminal work comprises one of the most influential theories in modern neuroscience. Yet, to date only few studies have addressed Hebb's ideas at the systems level. We hope that the methods reviewed here can lead to a proper estimation of co-activation patterns and help answering whether cell assemblies have any functional role, as originally envisioned by Hebb.

Acknowledgements

Supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and Fundação de Apoio à Pesquisa do Estado do Rio Grande do Norte (FAPERN). We thank the Buzsáki lab for making *in vivo* CA1 recordings publicly available through

the Collaborative Research in Computational Neuroscience website (<http://crcns.org/>), a data sharing website funded by the National Science Foundation. We also thank Sergio Conde-Ocázquez, Enio Aguiar and Adrien Peyrache for helpful discussions. The authors declare no competing financial interests.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.jneumeth.2013.04.010>.

References

- Abeles M. Synfire Chains 2009, <http://www.scholarpedia.org/article/Synfire.chains>
- Abeles M, Gat I. Detecting precise firing sequences in experimental data. *J Neurosci Methods* 2001;107:141–54.
- Abeles M, Gerstein GL. Detecting spatiotemporal firing patterns among simultaneously recorded single neurons. *J Neurophysiol* 1988;60:909–24.
- Adrian ED, Zotterman Y. The impulses produced by sensory nerve-endings Part II. The response of a single end-organ. *J Physiol (Lond)* 1926;61:151–71.
- Arabzadeh E, Panzeri S, Diamond M. Whisker vibration information carried by rat barrel cortex neurons. *J Neurosci* 2004;24:6011–20.
- Benchenane K, Peyrache A, Khamassi M, Tierney PL, Giovanni Y, Battaglia FP, et al. Coherent Theta oscillations and reorganization of spike timing in the hippocampal-prefrontal network upon learning. *Neuron* 2010;66:921–36.
- Berger D, Borgelt C, Louis S, Morrison A, Grün S. Efficient identification of assembly neurons within massively parallel spike trains. *Comput Intel Neurosci* 2010;18.
- Biroli G, Bouchaud JP, Potters M. On the top eigenvalue of heavy-tailed random matrices. *Europhys Lett* 2007;78:5.
- Buzsaki G. Large-scale recording of neuronal ensembles. *Nat Neurosci* 2004;7:446–51.
- Chapin JK, Nicolelis MAL. Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations. *J Neurosci Methods* 1999;94:121–40.
- Comon P. Independent component analysis, a new concept? *Signal Process* 1994;36:287–314.
- Gansel KS, Singer W. Detecting multineuronal temporal patterns in parallel spike trains. *Front Neuroinform* 2012;6.
- Grun S. Data-driven significance estimation for precise spike correlation. *J Neurophysiol* 2009;101:1126–40.
- Grun S, Diesmann M, Aertsen A. Unitary events in multiple single-neuron spiking activity: 1 detection and significance. *Neural Comput* 2002;14:43–80.
- Harris KD, Csicsvari J, Hirase H, Dragoi G, Buzsaki G. Organization of cell assemblies in the hippocampus. *Nature* 2003;424:552–6.
- Hebb DO. The organization of behaviour: a neuropsychological theory. New York: Psychology Press; 1949.
- Hubel DH, Wiesel TN. Receptive fields of single neurones in the cat's striate cortex. *J Physiol (Lond)* 1959;148:574–91.
- Humphries MD. Spike-train communities finding groups of similar spike trains. *J Neurosci* 2011.
- Hyvärinen A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans Neural Networks* 1999;10:626–34.
- Hyvärinen A, Oja E. A fast fixed-point algorithm for independent component analysis. *Neural Comput* 1997;9:1483–92.
- Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. *Neural Networks* 2000;13:411–30.
- Jutten C, Herault J. Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Process* 1991;24:1–10.
- Laubach M, Shuler M, Nicolelis M. Independent component analyses for quantifying neuronal ensemble interactions. *J Neurosci Methods* 1999;94:141–54.
- Laubach M, Wessberg J, Nicolelis MAL. Cortical ensemble activity increasingly predicts behaviour outcomes during learning of a motor task. *Nature* 2000;405.
- Lee AK, Wilson MA. Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron* 2002;36:1183–94.
- Lopes-dos-Santos V, Conde-Ocázquez S, Nicolelis M, Ribeiro S, Tort A. Neuronal assembly detection and cell membership specification by principal component analysis. *PLoS One* 2011;6:e20996.
- Louie K, Wilson M. Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron* 2001;29:145–56.
- Louis S, Borgelt C, Grun S. Complexity distribution as a measure for assembly size and temporal precision. *Neural Networks* 2010;23:705–12.
- Maldonado P, Babul C, Singer W, Rodriguez E, Berger D, Grun S. Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images. *J Neurophysiol* 2008;100:1523–32.
- Marčenko VA, Pastur LA. Distribution of eigenvalues for some sets of random matrices. *Math USSR-Sbornik* 1967;1:457–83.
- Mizuseki K, Sirota A, Pastalkova E, Buzsaki G. Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop. *Neuron* 2009;64.

- 18
- Nicolelis MAL, Baccala LA, Lin RCS, Chapin JK. Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system. *Science* 1995;268:1353–8.
- Okeefe J, Dostrovsky J. Hippocampus as a spatial map – preliminary evidence from unit activity in freely-moving rat. *Brain Res* 1971;34:171–5.
- Perrett D, Rolls E, Caan W. Visual neurones responsive to faces in the monkey temporal cortex. *Exp Brain Res* 1982;47:329–42.
- Peyrache A, Khamassi M, Benchenane K, Wiener SI, Battaglia FP. Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. *Nat Neurosci* 2009;12(919), U143.
- Peyrache A, Benchenane K, Khamassi M, Wiener SI, Battaglia FP. Principal component analysis of ensemble recordings reveals cell assemblies at high temporal resolution. *J Comput Neurosci* 2010;29:309–25.
- Plerou V, Gopikrishnan P, Rosenow B, Amaral LAN, Guhr T, Stanley HE. Random matrix approach to cross correlations in financial data. *Phys Rev E* 2002;65.
- Quiroga RQ, Panzeri S. Extracting information from neuronal populations: information theory and decoding approaches. *Nat Rev Neurosci* 2009;10:173–85.
- Ribeiro S, Gervasoni D, Soares ES, Zhou Y, Lin SC, Pantoja J, et al. Long-lasting novelty-induced neuronal reverberation during slow-wave sleep in multiple forebrain areas. *PLoS Biol* 2004;2, 126–37.
- Seba P. Random matrix analysis of human EEG data. *Phys Rev Lett* 2003;91:4.
- Shmiel T, Drori R, Shmiel O, Ben-Shaul Y, Nadasdy Z, Shemesh M, et al. Temporally precise cortical firing patterns are associated with distinct action segments. *J Neurophysiol* 2006;96, 2645–52.
- Siapas AG, Lubenov EV, Wilson MA. Prefrontal phase locking to hippocampal theta oscillations. *Neuron* 2005;46, 141–51.
- Staude B, Grun S, Rotter S. Higher-order correlations in non-stationary parallel spike trains: statistical modeling and inference. *Front Comput Neurosci* 2010a;4.
- Staude B, Rotter S, Grun S. CuBIC: cumulant based inference of higher-order correlations in massively parallel spike trains. *J Comput Neurosci* 2010b;29, 327–50.
- Stevenson IH, Kording KP. How advances in neural recording affect data analysis. *Nat Neurosci* 2011;14:139–42.
- Tetko IV, Villa AEP. A pattern grouping algorithm for analysis of spatiotemporal patterns in neuronal spike trains, detection of repeated patterns. *J Neurosci Methods* 2001;105:1–14.
- Tracy CA, Widom H. Level-spacing distributions and the airy kernel. *Commun Math Phys* 1994;159:151–74.
- Wilson MA, McNaughton BL. Reactivation of hippocampal ensemble memories during sleep. *Science* 1994;265:676–9.