SIMNETS: a computationally efficient and scalable framework for identifying networks of functionally similar neurons

Jacqueline B. Hynes [1, 2], **David M. Brandman**,[2, 3], **Jonas B. Zimmerman**[4], **John P. Donoghue**[1, 2, 4, 5], **Carlos E. Vargas-Irwin** [*, 1, 2]

1 Department of Neuroscience, Brown University, Providence, RI, USA.
2 Robert J. and Nancy D. Carney Institute for Brain Science, Brown University, Providence, RI, USA
3 Department of Surgery (Neurosurgery), Dalhousie University, Halifax, NS, Canada.
4 WYSS Institute, Chemin des Mines 9, CH-1202, Geneve, Switzerland.
5. Department of Engineering, Brown University, Providence, RI, USA.

∗ Correspondence to: carlos_vargas_irwin@brown.edu

Keywords: neural ensembles, dimensionality reduction, clustering algorithm, neuronal networks

Abstract

We present the Similarity Networks (SIMNETS) algorithm, a computationally efficient and scalable method for identifying groups of functionally related neurons within larger, simultaneously recorded ensembles. Our approach begins by independently measuring the intrinsic relationship between the activity patterns of each neuron across experimental conditions before making comparisons across neurons (instead of directly comparing firing patterns using measures such as correlations in firing rate or synchrony). This strategy estimates the intrinsic geometry of each neuron's output space and allows us to capture the information processing properties of each neuron in a format that is easily compared between neurons. Dimensionality reduction tools are then used to map the pairwise neuron comparisons into a low-dimensional space where groupings of functionally related neurons are identified using clustering techniques. SIMNETS' computational complexity scales almost linearly with the number of neurons analyzed and makes minimal assumptions about single-unit encoding properties, making this approach especially well-suited for examining large networks of neurons engaged in complex behaviors. We validate the ability of our approach to detect functional groupings using simulated data with known ground-truth as well as three datasets including ensemble activity from primate primary visual and motor cortex as well as rat hippocampal CA1 region.

Introduction

The neural computations underlying complex sensory, cognitive, and motor information processing are thought to emerge from interactions across large networks of neurons organized across multiple topographical scales. Within large networks, smaller ensembles of neurons ('sub-nets') engaged in similar information processing tasks have been proposed to embody computational units that support specific functions including perceptual integration, memory storage/retrieval, and dexterous motor control[1–3]. Identifying functional sub-nets would greatly simplify the process of tracking information flow in cortical circuits, modeling neural activity, and ultimately understanding and potentially replicating the general principles of neural computation [4–7]. While it is now possible to record ever larger neural populations, detecting functional groupings of neurons and characterizing their computational operations has proven notoriously difficult because of the scale of data processing involved and the lack of accepted mathematical tools to partition large networks into smaller functional components [8].

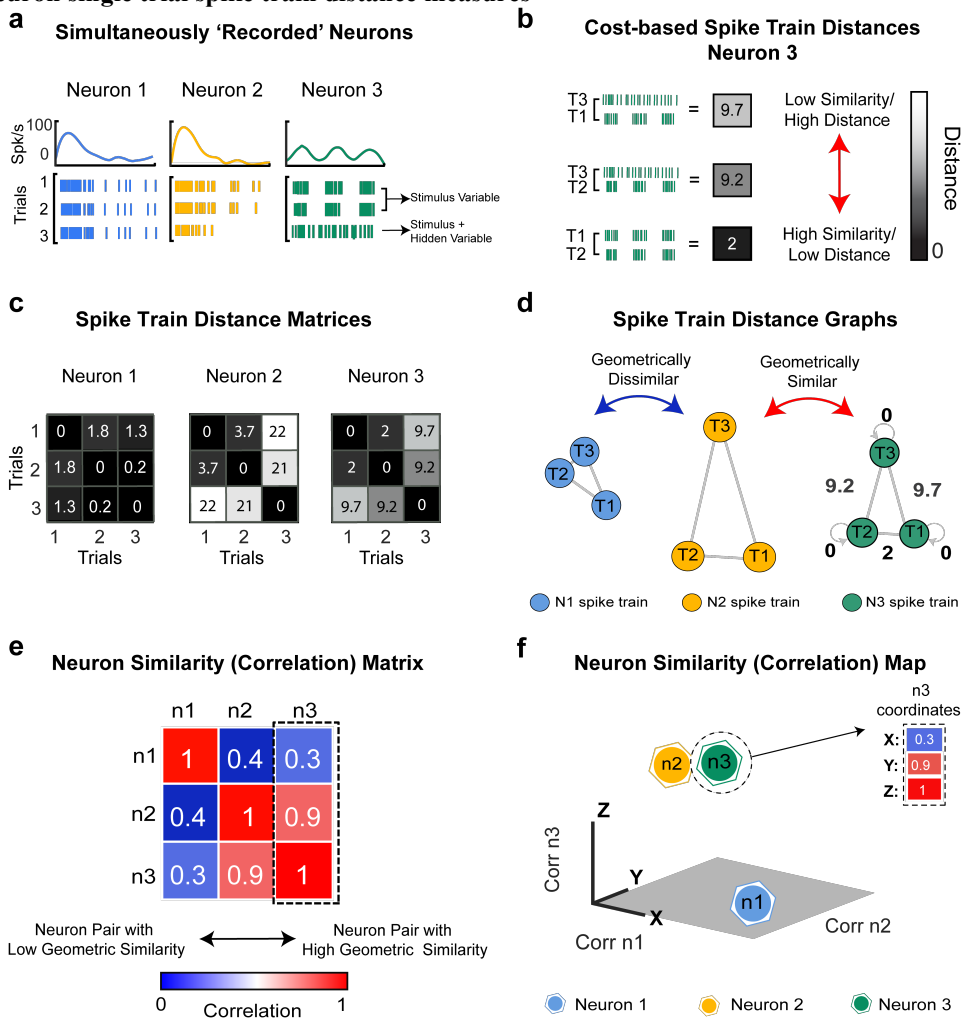
One of the critical challenges lies with the selection of an appropriate quantitative definition of 'functional similarity' across neurons. Motivated by theories of Hebbian cell assemblies, several neuron clustering approaches have focused on using synchrony or firing rate covaritaions to define functional associations between neurons [9–14]. One widely discussed hypothesis proposes that synchronously-active neurons could act as an independent coding dimension to facilitate perceptual or cognitive integration of the information encoded in the firing rates of individual neurons [15–17]. Although a number of studies have observed synchronous and correlated activity between neurons in multiple brain regions, discrepant reports regarding the functional and statistical significance of the detected correlations have led to some doubts about usefulness of this approach for detecting functionally meaningful ensembles[18,19]. While focusing on spike rate and spike time covariations is intuitive, cross correlation methods do not scale well for large datasets and could limit the complexity of the functional relationships that can potentially be detected between neurons. Specifically, these methods prioritize grouping neurons that exhibit similar firing patterns[9,11,20], as opposed to grouping neurons that exhibit similar information processing properties. The underlying premise with this approach is that similar firing patterns imply similar encoding properties, however, a growing body of work suggests that this may be an oversimplified view [21]. Individual neurons in higher-level brain areas[22,23], motor areas[2425], and even primary sensory areas [26], can exhibit highly heterogeneous and temporally complex responses [21]. Previously, trial-to-trial fluctuations of single neuron spiking activity were interpreted as biological noise; however, these studies suggest that heterogeneity and temporal complexity are relevant for information processing operations taking place across the network. It follows that measures of trial-averaged spike rate or spike time covariations may neglect important aspects of a neuron's activity that can reveal it's functional associations to other neurons. Here, we propose a more general strategy to assess and compare the information processing properties of individual neurons.

SIMNETS: a novel mathematical framework to identify functional neuronal sub-ensembles

The SIMNETS algorithm is designed to transform spike trains from simultaneously recorded neurons into a common representation that captures their information processing characteristics in an abstract way. Instead of comparing the spiking output between neurons directly, we seek to compare the intrinsic structures of their individual sets of spike train outputs, i.e., how 'self-similar' a neuron's spike trains are across time. We envision each neuron as performing some unknown operation on a set of high dimensional inputs (potentially thousands of synaptic connections). To be useful as a computational element, each neuron should have a relatively consistent internal mapping between inputs and outputs -- allowing for noise and the potential to change the mapping over time, i.e. learning. A given neuron may be insensitive to certain changes taking place across the network where it is embedded, and so the neuron's spike train outputs will appear similar across many different global network states. On the other hand, other global network states may elicit dramatic changes in the output of the neuron, depending on the strength of the relevant synaptic connections. Thus, the spike train outputs generated by a single neuron across different perceptual, cognitive, or behavioral states highlight the differences between some global network states and generalize over others (note that we define a 'network state' holistically to encompass all neural activity including both incoming sensory information as well as internal drives evolving through time). The key insight is that we can characterize the information processing occurring at the level of individual neurons by examining the differences or similarities between the set of spike train outputs generated during different conditions across time. Our basic assumption is that the outputs of neurons that are 'computationally equivalent' will generalize and differentiate across the same sets of network states. Note that this approach makes it possible to compare the operations performed by simultaneously recorded neurons on a trial-by-trial basis, without requiring explicit knowledge of the type of function they may be computing. Mathematically, we can represent the relationship between spike trains originating from a given neuron across a set of events of interest, e.g., stimulus presentations, behavioral responses, etc., using a pairwise distance matrix, where each entry represents the similarity between a pair of spike trains (note that several metrics to quantify spike train similarity have been proposed, see Online Methods for a detailed discussion). We shall refer this type of matrix as a Spike-Train Similarity (SSIM) matrix (Fig. 1*a-c*). A SSIM matrix can be thought of as a high-dimensional representation of the relationship between the neuron's outputs across experimental conditions, agnostic to any experimenter-selected parametric encoding model[27] (see Supplementary Fig 1-2 for example single neuron SSIM matrices). Stated another way, a SSIM matrix represents the intrinsic geometry of the output space of a neuron across a set of sampled conditions (Fig 1*d*). Geometric models of similarity data have a long history of application in the field of psychology, where they have been used to model the perceptual relationships between sensory stimuli, i.e. perceptual metric-space[29]. However, it is only more recently that this approach has found application in the field of Neuroscience, where it has been successfully used to model the relational structure of neuronal ensemble activity patterns[30–32] and fMRI activity patterns[33]. Crucially, when neurons are recorded simultaneously, the intrinsic geometry of their output spaces can be directly compared. This can be accomplished by comparing their SSIM matrices using standard correlation statistics, such as Pearson's correlation (Fig 1*e*). Thus, our proposed strategy involves calculating the pairwise spike train distances between all spike trains generated by a given neuron, and then comparing each neuron's distance matrix to that of

every other neuron in the population. The resulting *N*x*N* pairwise correlation measures is represented as a single Neuron Correlation (NC) matrix (Fig. 1f), where each column of the matrix can be viewed as a vector that represents the functional relationship of a given neuron to all others *N*-1 neurons in the population (Fig. 1g). Standard dimensionality reduction techniques (e.g. multidimensional scaling, t-distributed stochastic neighborhood embedding) can be used to project these vectors into a low-dimensional Neuron Similarity (NS) map such that neurons are positioned according to their information processing properties. Applying dimensionality reduction makes the data easier to visualize and facilitates statistical analysis. Overall, this representation reduces the problem of identifying functional sub-ensembles to one of detecting clusters of neurons within the NS map. This step can be accomplished using standard clustering algorithms (e.g. k-means), and validated using a Monte-Carlo based approach relying on shuffled distance matrices to avoid false cluster discovery (see Methods; supplementary Fig 4). We call this new strategy for identifying sub-nets of neurons with similar informational properties SIMNETS. The main steps of algorithm are outlined in Fig. 2. Note that there is a wide choice of (1) similarity metrics for spike trains, (2) dimensionality reduction algorithms, and (3) clustering algorithms that can be employed within the proposed analytical framework.

Fig 1| Assessing functional relationships between neurons by comparing state spaces generated from within-neuron single trial spike train distance measures

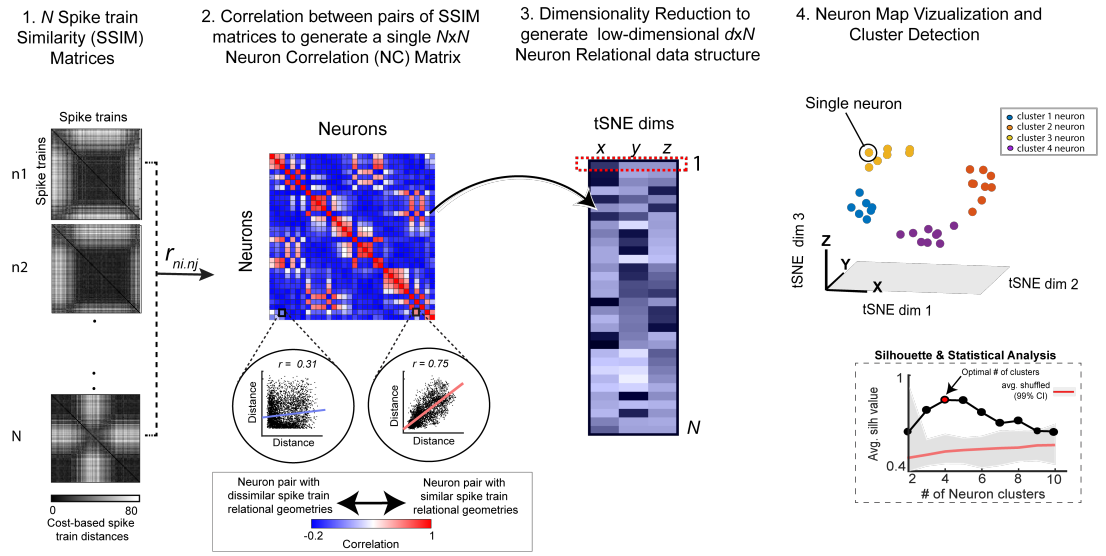


a. Simulated spike train data for three 'simultaneously recorded' synthetic neurons (n) during repeated

stimulus presentations: n2 was designed to exhibit similar time-varying firing rate functions to n1 and shared, trial-to-trial fluctuations in spike patterns with n3. *Top row*: trial-averaged firing rate functions for each neuron. *Bottom row*: spike train outputs of each neuron in response to repeated stimulus presentations, however, note that n2 and n3 encode a 'mixed signal' variable on trial 3 as a reduction in spike rate (n2) and a change in the temporal pattern of the output (n3). **b.** Victor-Purpura (VP) spike train metric applied to n3 spike trains: the metric assigns an 'edit cost' to spike train pairs in accordance to the similarity of their spike patterns. **c.** The relationship between a set of spike trains originating from a common neuron is summarized as a pairwise distance matrix; each entry corresponds to the edit-distance between two spike trains. We refer to these data structures as spike train similarity (SSIM) matrices **d.** Visualization of SSIM matrix using a graph representation (**c**); a separate graph is plotted for each matrix. Each point represents a spike train from a given neuron and the line connecting pair of points corresponds to the distance values assigned to the spike trains. Note that the sizes of the spike train graphs are different across neurons, but that shape of n2 and n3 graphs are similar, i.e., covariation in their side lengths. (**e**) Pairs of single neuron SSIM matrices are compared using Pearson's correlation and represented as a Neuron Correlation (NC) matrix. Note that the diagonal represents the comparison of each neuron to itself, and so always equal to one. Each NC matrix column (broken line) represents the correlations between one neuron versus all others. Each of these column vectors can be used to map a neuron into an *N*-dimensional neuron space. **f**. Scatter plot showing the data contained in the NC matrix (**e**): each point corresponds to a single neuron with coordinates defined by a *N* dimensional vector of correlation values, i.e., a NC matrix column. Note that neurons 2 and 3 are located next to each other in the map because of the high correlation between their SSIM matrices, which is ultimately dependent upon the geometric similarity of their spike train output spaces.

Unlike methods based on measures of synchronous spiking activity[9,20,34], the SIMNETS algorithm identifies neurons with similar functional properties, even if they exhibit diverse firing statistics, i.e., even if their firing rates or spike times are different. Critically, unlike other pairwise methods [9,11,20], SIMNETS is well suited for studying datasets with large numbers of neurons and relatively small numbers of experimental trials. Largely because the computational cost of the generating the high-dimensional neuron embedding (Fig 2) grows linearly with the number of numbers (each additional neuron only requires the generation of one additional computationally expensive SSIM matrix). Further, SIMNETS can be implemented without a priori knowledge of neural tuning functions or trial labels, making it particularly useful for the analysis of complex, naturalistic behaviors.

Fig. 2|The SIMNETS algorithm can be used to identify functionally related groups of neurons.



The basic steps of the algorithm (demonstrated using synthetic data) are as follows: Step 1. Generate *N* Single-neuron Spike Train Similarity (SSIM) matrices**:** extract *S* spike train events of equal duration from *N* simultaneously recorded neurons and calculate the pairwise spike train distances between all spike trains using a predetermined metric, e.g., Victor-Purpura. Step 2. Neuron Correlation (NC) Matrix: pairwise correlations between single neuron SSIM matrices are used to generate an *N*x*N* NC matrix. *Left Circular Inset*: example neuron pair with a low SSIM correlation value indicating dissimilar spike train geometries. *Right Circular Inset*: neuron pair with a high SSIM correlation value indicating similar spike train geometries. Step 3: NC Matrix Dimensionality reduction: project the *N*x*N* correlation matrix down into a smaller number of *d*x*N* dimensions using standard techniques (e.g. t-SNE). Each row represents the coordinates of a single neuron (red broken line) in a low-dimensional space. Step 4. Visualization of Neuron Similarity (NS) map and Cluster Detection: each colored point represents a single neuron with *x, y, z* coordinates that correspond to a single row of the low-dimensional data structure shown in step 3. Clustering: neuron clusters (denoted using color) are detected in a 10-*d* space using agglomerative clustering methods (e.g., *k*-means clustering). *Inset*: Optimal cluster number is estimated as the maximum average silhouette value for a range of test cluster number, where a high silhouette value indicates good cluster separation. The statistical significance of the estimated optimal cluster number is determined from a null-distribution of silhouette values (gray band), which was obtained by repeating steps 2 - 3 on shuffled SSIM matrices (step 1) over multiple iterations (see Methods for more details). Neurons assigned to same cluster will have similar spike train geometries.

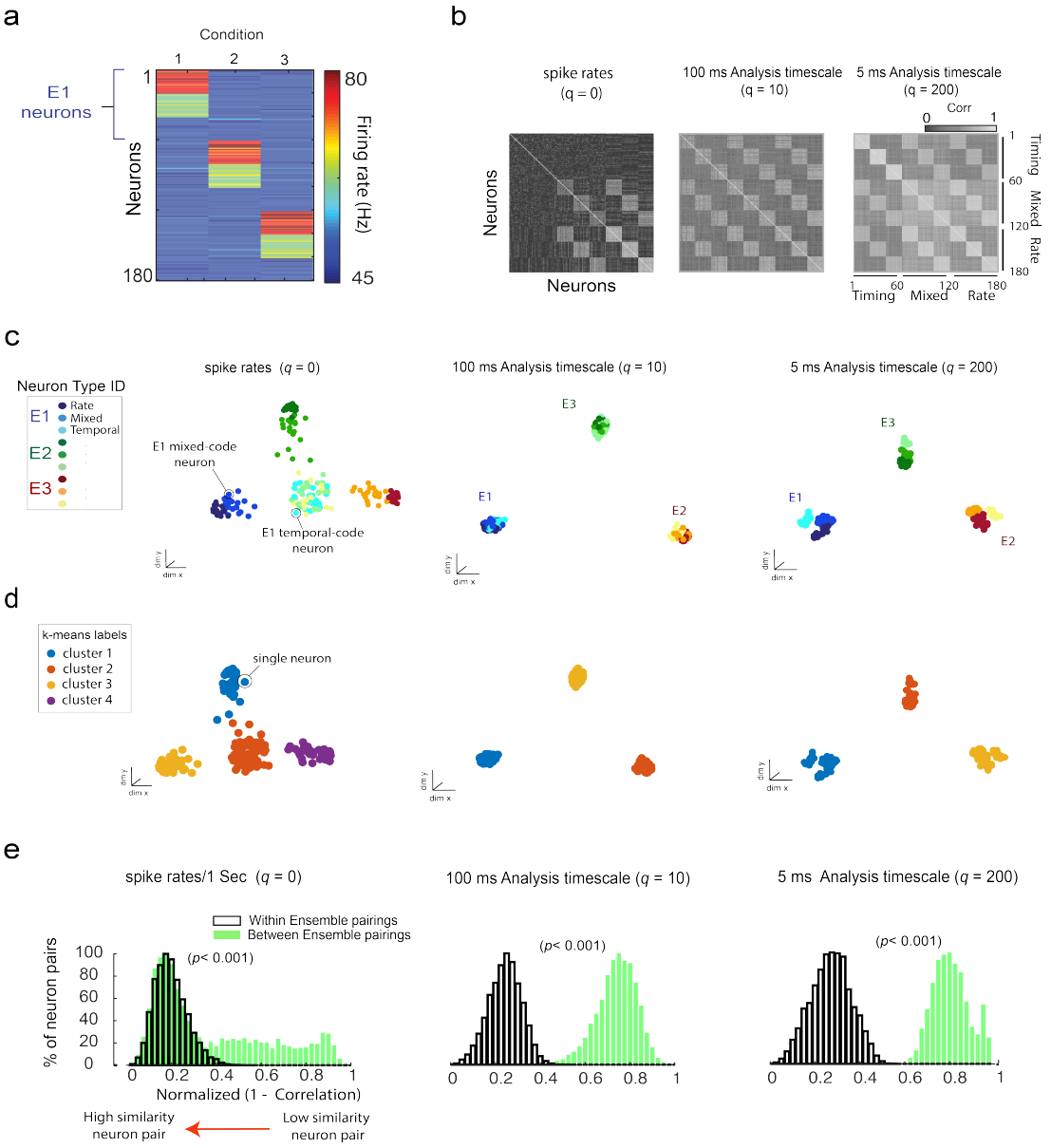
Results

Here, we apply SIMNETS to four different datasets to validate its ability to detect functional associations between neurons. We first apply the algorithm to a synthetic dataset with known functional ensembles. Performance is compared against an alternative approach representative of traditional methods[14,35] that use spike train metrics to compare the spike patterns between neurons directly. Next, to demonstrate the generalizability of SIMNETS, we apply the algorithm to three datasets of real neurons (i.e., single units) recorded from non-human primate V1[36], primary motor cortex (M1) [37], and the rat CA1 hippocampal region[38] (see Methods, table 2). For the V1 and M1 datasets, the SIMNETS neuron functional maps are validated against the estimated computational properties of the neurons calculated using parametric tuning models. We use the rat CA1 dataset to demonstrate how SIMNETS can be used for exploring the functional properties of neurons when the tuning functions of the neurons are not easily quantifiable or are unknown.

Synthetic Neuron Population — clustering functionally similar neurons exhibiting distinct firing patterns

Here, we apply the SIMNETS algorithm to simulated spike train data from a population of 180 synthetic neurons comprised of 3 functionally distinct 'ensembles' (E_1, E_2, E_3). Each E_i ensemble (n = 60, neurons) was designed to represent a sub-group of computationally equivalent neurons that exhibited heterogenous firing patterns. Specifically, sub-groups of neurons within ensemble E_i responded to a common 'preferred' test condition through either a change in spike rates (n = 20, neurons), a change in the precise timing of their spikes (n = 20, neurons), or a change in both spike rates and spike timing (n = 20, neurons) (Fig 3*a*) (see Methods for more details). We simulated 30, one-second spike trains for each of the 180 neurons, which included 10 repetitions of each stimulus (S = 30, spike trains per neuron).

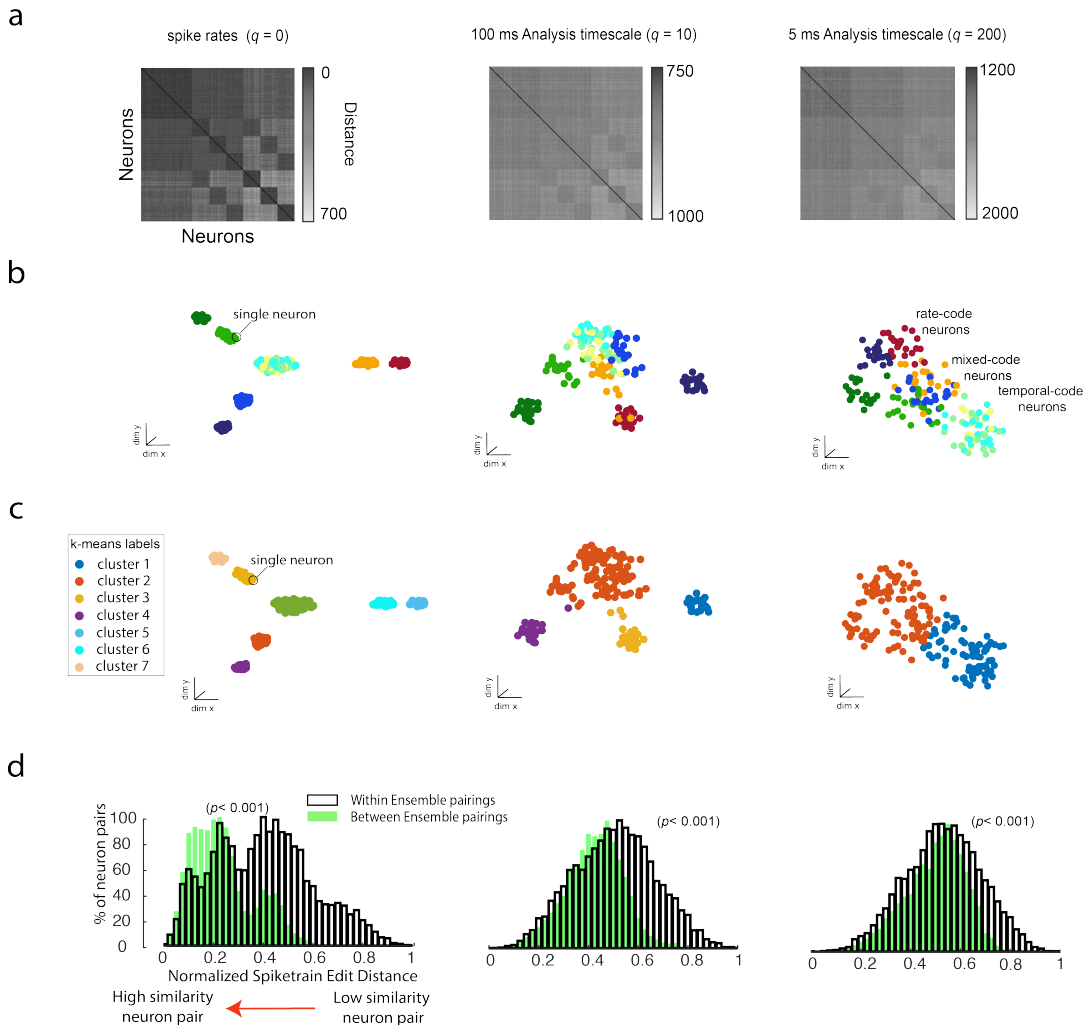
Figure 3| SIMNETS Detects Ensembles of Functionally Similar Neurons in Population of Synthetic Neurons with Mixed Coding Schemes



a. Trial-averaged firing rates for a population of 180 synthetic neurons as a function of three different stimulus condition. Neurons are ordered along the y-axis according to preferred condition and by encoding strategies. The red-yellow, green, and blue color-hue range of filled circles indicate ensemble membership (e.g., $n_{1\text{-}60}$, ensemble 1). Filled circles with dark, mid, and light color-hues indicates the rate neurons, hybrid neurons, and temporal coding neurons, respectively. **b**. SIMNETS NC matrix for three different analysis temporal sensitivity values (*left-to-right*): rate, 100 ms, 5 ms. SIMNETS color-bar represent a range of normalized correlation values, where higher correlation values (light pixels) correspond to greater similarities. **c**. Low-dimensional NS map. Each dot represents a single neuron and different color dots indicate each neurons' ensemble membership and coding scheme (color notation similar to **a**). **d**. Low-dimensional SIMNETS NS map (same as **c**) with neurons labeled according to *k*-means cluster assignment. **e**. Histogram of normalized 'Within-' and 'Between-ensemble' correlations from SIMNETS NC matrix (**b**). Non–overlapping Within/Between distributions (e.g., middle and right SN plot) corresponds to good separation between each of the three ensembles in the Neuron Map (*ranksum*, *p<.001)*.

SIMNETS was applied to the resulting *N*x*S* spike trains using three different temporal accuracy settings for the Victor-Purpura spike train metric: 5ms (q = 200), 100ms (q = 10), and temporal sensitivity (q = 0). As expected, with a setting of q = 0, the neurons operating with a rate-based encoding scheme ('rate-code' neurons) and rate/temporal-based coding scheme ('mixed-code' neurons) are grouped into three functionally dinstinct clusters in the NS map, while the functionally dissimilar 'temporal-code' neurons form a single cluster at the center of the map (Fig 3*c, left column)*. When the temporal sensitivity of the analysis is

Figure 4| Direct Comparison Method (DCM) fails to Detects Functional Ensembles in Population of Synthetic Neurons.



a. DCM *NxN* distance matrices shown for three different analysis temporal sensitivity values (*left-to-right*): rate, 100 ms, 5 ms. Color-bars represent a range of distance values where smaller distance values (dark pixels) correspond to greater similarities. **b.** Low-dimensional DCM NS maps were generated from the high-dimensional neuron distance matrices (**a**). Each point represents a single neuron and different color points indicate each neuron's ensemble membership (E1, E2, or E3) and coding scheme (color notation same as Fig 3). **c.** Low-dimensional DCM NS maps (*bottom row*) with neurons labeled according to k-means cluster assignment. **d.** Histogram of normalized 'Within-' and 'Between- ensemble' distances from DCM distance matrices (**a**).set to values of q > 0, the algorithm sensitive to differences in spike timing in addition to the total number of spikes. Using these settings, SIMNETS correctly groups all neurons into three distinct clusters that reflect the ground-truth functional ensemble assignments (Fig 3*c*, *middle and right column*). At very high q values, the NS map shows sub-groupings within each of the main clusters that reflect the coding properties of

the neurons; however, the optimal number of clusters remains in agreement with the ground-truth functional ensemble assignments. By specifying a higher partition value for the k-means clustering step of the algorithm (e.g., k = 9), it is possible to confirm that the sub-groupings within the detect clusters are defined by the coding properties of the neurons (data not shown). For a demonstration of the interaction between the cluster number and the SIMNETS hyper parameters, *perplexity* and q, see Supplementary Fig. 5.

Ideally, the SIMNETS algorithm should rank neurons in the same functional group as being more similar to each other than to neurons in a different functional group. In order to quantify this trend, we compared the distribution of similarity estimates (entries in the NC matrix) within and between the artificially generated ensembles (Fig. 3*e*). Within-ensemble similarity was significantly higher than between-ensemble values in all cases (Mann-Whitney p<.001). For q values > 0, there was no overlap between the two distributions, indicating complete separation of the functional classes. Our results demonstrate that the SIMNETS algorithm can accurately separate neurons according to their computational properties, even if they employ different coding schemes to represent information.

In order to demonstrate the potential pitfalls of traditional approaches that directly compare the spike trains *between* neurons on a trial-by-trial basis[14] we applied a 'Direct Comparison Method (DCM)', to the synthetic dataset (see online Methods and Fig. 4). This algorithm also uses Victor-Purpura spike train metrics, but compares spike trains from different neurons directly, without generating SSIM matrices as an intermediate step. In this case, each entry of the resulting NxN matrix is the sum of the spike train distances between matching trials across a neuron. For example, neurons that generate similar spike train outputs on matching trials will have a lower the sum of their spike train distances A neuron pair with a similar trial-to-trial spike train outputs will have a low distances Overall, the DCM failed to cluster functionally similar neurons into the three ground-truth functional ensembles for any of the tested q values (Fig 4c). The distributions of similarity estimates for neurons within and between ensembles (Fig. 4*d*) displayed broad overlaps, reflecting the poor separation between functional sub-ensembles. Our results demonstrate that grouping neurons based on the similarity of their spike train outputs does not necessarily reflect their informatinal content (and presumed computational properties).

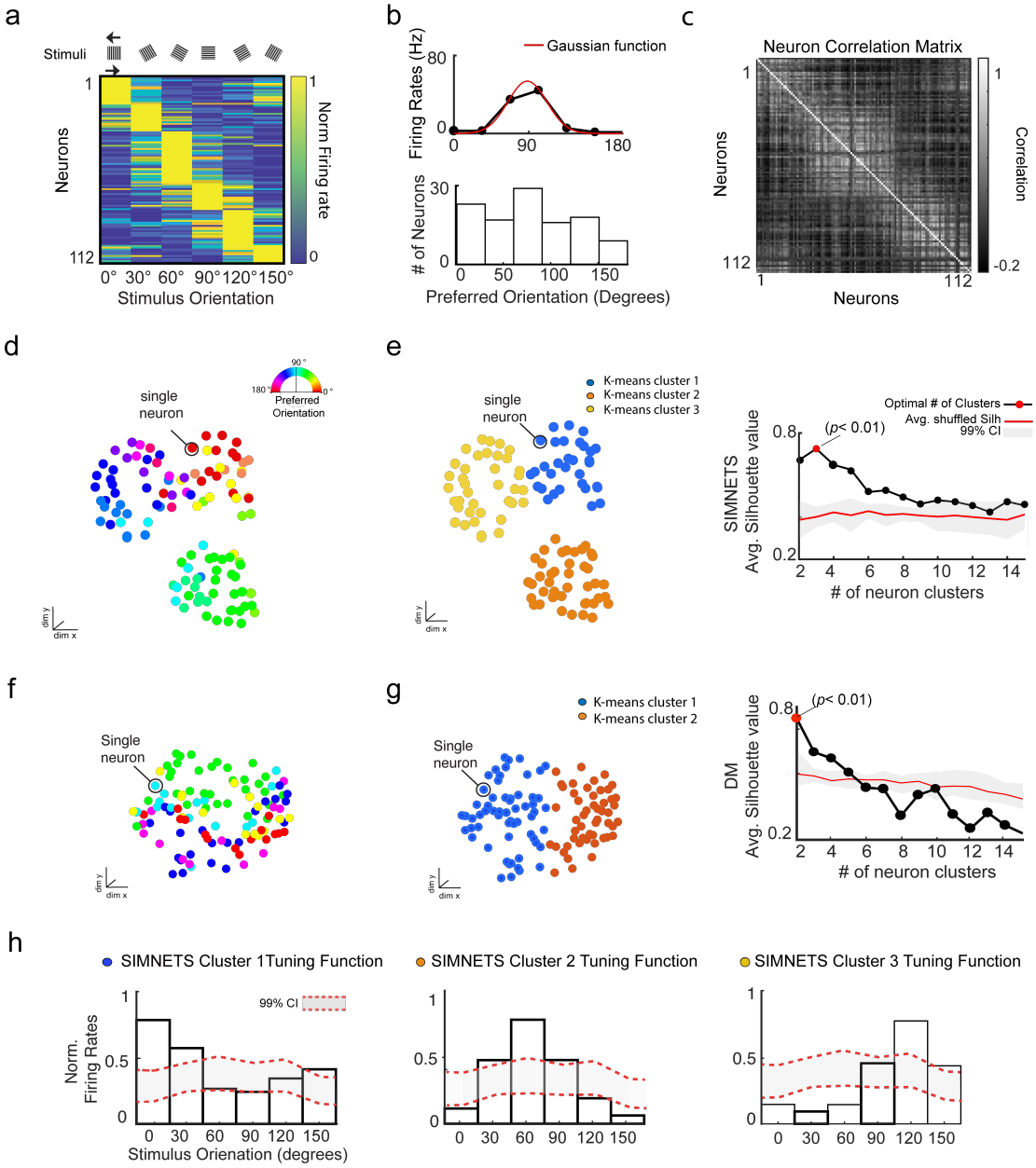
V1 Neuron Population – clustering real neurons with known tuning functions

We next analyzed a previously described dataset of 112 Macaque V1 neurons simultaneously recorded using a 96-channel electrode array during the presentation of drifting sinusoidal gratings [36,39] (Fig 5). We extracted 1 second of spiking data from the first 30 repetitions of each stimulus (S = 360), starting 0.28 seconds after stimulus onset (Fig 5*a*). Each neuron's receptive field orientation ('preferred' orientation) was estimated by finding the orientation that maximizes a Gaussian function fitted to the stimulus-dependent firing rates (Fig. 5*b*) (see Methods for more details).

We examined the NS map produced using SIMNETS in order to determine if it accurately captured the functional relationships between neurons (Fig 5*d*-*e*). A Circular-linear correlation ($r$cl) analysis shows a significant positive relationship between preferred orientation and neuron location in the map (Pearson, $r$cl = 0.88; p = .001), confirming that

neurons with similar computational properties are located close to each other in the NS map. Applying the k-means algorithm to the NS map revealed an optimal number of $\hat{k} = 3$ neuron clusters ($\hat{h} = 0.74$, max average silhouette value), indicating that the neurons are organized into three separate sub-ensembles (Fig. 5*e*). The statistical significance of the number of estimated optimal clusters was determined using the shuffle-based statistical test

Figure 5| V1 neuron population during the presentation of drifting sinusoidal grating at multiple orientation



a. Normalized trial-averaged firing rates of a population of V1 neurons (N = 112, neurons) during the presentation of 12 different drifting grating stimuli (T = 360, trials). Stimuli were presented for 1.28 s at 6 different orientations (0, 60, 90, 120, 150 degrees) and 2 drift directions (rightward and leftward drift, orthogonal to orientation). **b**. Distribution of calculated preferred grating orientations for all neurons. **c**. SN NC matrix with neurons ordered according to preferred grating orientation. Color-bar represent a range of correlation values where higher correlation values (light pixels) correspond to greater similarities between neurons. **d.** SN NS map with neurons labeled according to preferred orientations. **e.** SN NS map with neurons labeled according to k-means cluster assignments. *Right* (g): Average silhouette value for SIMNETS RN map

as a function of the number of clusters (red filled circle denotes optimal cluster separation), compared to expected chance distribution obtained from shuffled data. (bootstrap, $p < 0.01$). **f-g.** DCM NS map with neurons labeled according to preferred directions (**f**) and k-means cluster assignments (**g**). Right (g): Average silhouette value for DCM RN map as a function of the number of clusters (red filled circle denotes optimal cluster separation), compared to expected chance distribution obtained from shuffled data. **h.** Ensemble orientation tuning functions for each of the SN neuron clusters (**e**) with significant peaks around 0, 60, 120 degrees for clusters 1,2, and 3, respectively. Red line and gray band correspond to the mean and 99% confidence intervals of the null-distribution (bootstrapped).

The shuffle-test involves generating a null-distribution of silhouette values by shuffling each of the *N* SSIM matrices, calculating a new NC matrix, and the associated silhouette value. This procedure is repeated over multiple iterations until a distribution of silhouette values is generated. The estimated number of neuron clusters is considered statistically significant if the original silhouette value falls outside the 99% confidence interval of the null-distribution of silhouette values (see Methods and Supplementary Fig 4 for more details. We examined the computational properties of each of the detected clusters by calculating ensemble tuning functions that take into account the average activity of all neurons within each identified cluster. Our analysis revealed that sub-ensembles displayed significant tuning with peaks evenly distributed at Δ60° intervals. (Fig.5*e*). Tuning strength and direction-of-motion tuning preferences did not appear to contribute to the cluster organization (data not shown).

The SIMNETS results were again compared against DCM results to demonstrate how a more traditional approach fails to organize the neurons according to their functional properties (Fig. 5). Although neuron clusters were detected using DCM, we observed a weak and non-significant r_{cl} correlation between neuron location and preferred orientation (Pearson, $r_{cl} = 0.01$; $p = .56$), indicating that the two detected DCM clusters ($\hat{k} = 2$; $\hat{h} = 0.82$) were unlikely to exhibit a tuning preference for any particular orientation. We also compared SIMNETS performance to a modified cross-correlation analysis (See Supplementary Fig. 7*a-b*) and found that measures of cross-correlation between the spike trains of different neurons failed to capture the neuron's estimated functional properties.

M1 Neuron Population – clustering real neurons with known tuning functions

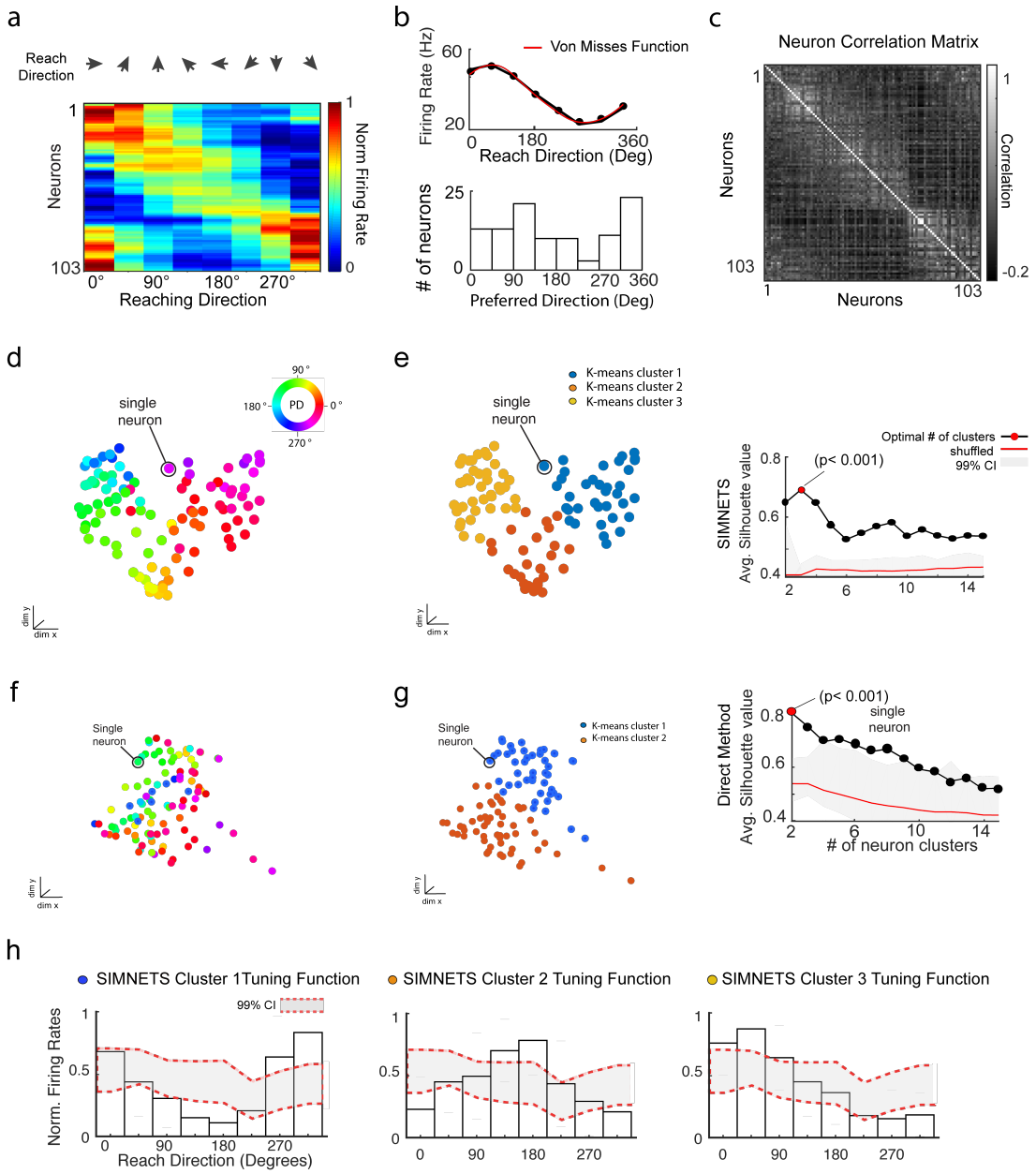
We next applied the SIMNETS algorithm to a dataset of 103 M1 neurons recorded using a 96-channel electrode array in a macaque performing a planar 8-direction instructed-delay reaching task (see Methods, Section 3.1). Using a standard metric, each neuron's preferred reach direction was estimated by fitting a von Misses distribution [40] to the firing rates as a function of direction (Fig. 6*b*). This dataset and task has previously been described elsewhere [41,42] (see Methods for more details).

We extracted 1-second spike train events ($S = 114$) from each neuron during all trials where the monkey sucesfully reached the cued target, starting 0.1 seconds before movement onset. As with the V1 data, the layout of the neurons in the SIMNETS NS map accurately reflected the estimated tuning properties (Fig 6*d* - *e*). A circular-linear correlation analysis found a significant positive relationship between preferred direction and mapped location (Pearson, $rcl = 0.91$; $p = .001$). SIMNETS revealed a statistically significant optimal number of $\hat{k} = 3$ clusters ($\hat{h} = 0.71$), indicative of three functional sub-ensembles. Each cluster displayed ensemble-level tuning with significant peaks at 45°, 180°, and 315°.

Additionally, our results show that neurons are not distributed along a uniform continuum within the NS map, but instead from statistically separable clusters in space. These results are in agreement with previous findings, supporting the hypothesis that the biomechanical constraints of the limb are reflected in an uneven distribution of preferred directions among motor cortical neurons [44,45].

As before, the DCM failed to organize the neurons according to their preferred directions (Fig 6*d*), resulting in weak, non-significant relationship between the neurons' preferred direction and location in the NS map (Pearson, $rcl = 0.18$; $p = 0.56$). Again, we also applied compared SIMNETS performance to a modified cross-correlation analysis (See Fig. 7*c*-d) and found that measures of cross-correlation between the spike trains of different M1 neurons failed to capture the functional relationships between the majority of the neurons.

Figure 6| M1 neuron population during center-out reaching task



a. Trial-averaged, normalized firing rates for each neuron (N = 103) as a function of reach direction for a planar 8-directional reaching task. **b**. Histogram of estimated preferred reach direction for the population. **c**. SIMNETS NC matrix. Color-bar represent a range of correlation values, where higher correlation values (light pixels) correspond to greater similarities between neurons. **d - e**. SN NS map with neurons labeled according to the estimated preferred reach directions (**d**) or k-means cluster assignments (**e**). *Right* (**e**): Average silhouette value for SIMNETS RN map as a function of the number of clusters (red filled circle denotes optimal cluster separation), compared to expected chance distribution obtained from shuffled data (bootstrap, $p < 0.01$). **f - g**. DCM NS map with neurons labeled according to preferred reach directions (**f**) or k-means cluster assignments (**g**). Right (**g**): Average silhouette value for DCM RN map as a function of the number of (bootstrap, $p < 0.01$). **h.** Ensemble tuning functions for each of the SN neuron clusters (**e**) with significant peaks around 30, 180, 315 degrees for clusters 1, 2, and 3, respectively. Red line and gray band correspond to the mean and 99% confidence intervals of the bootstrapped null-distribution (see Methods).

Hippocampal Dataset – clustering neurons with complex or unknown tuning properties

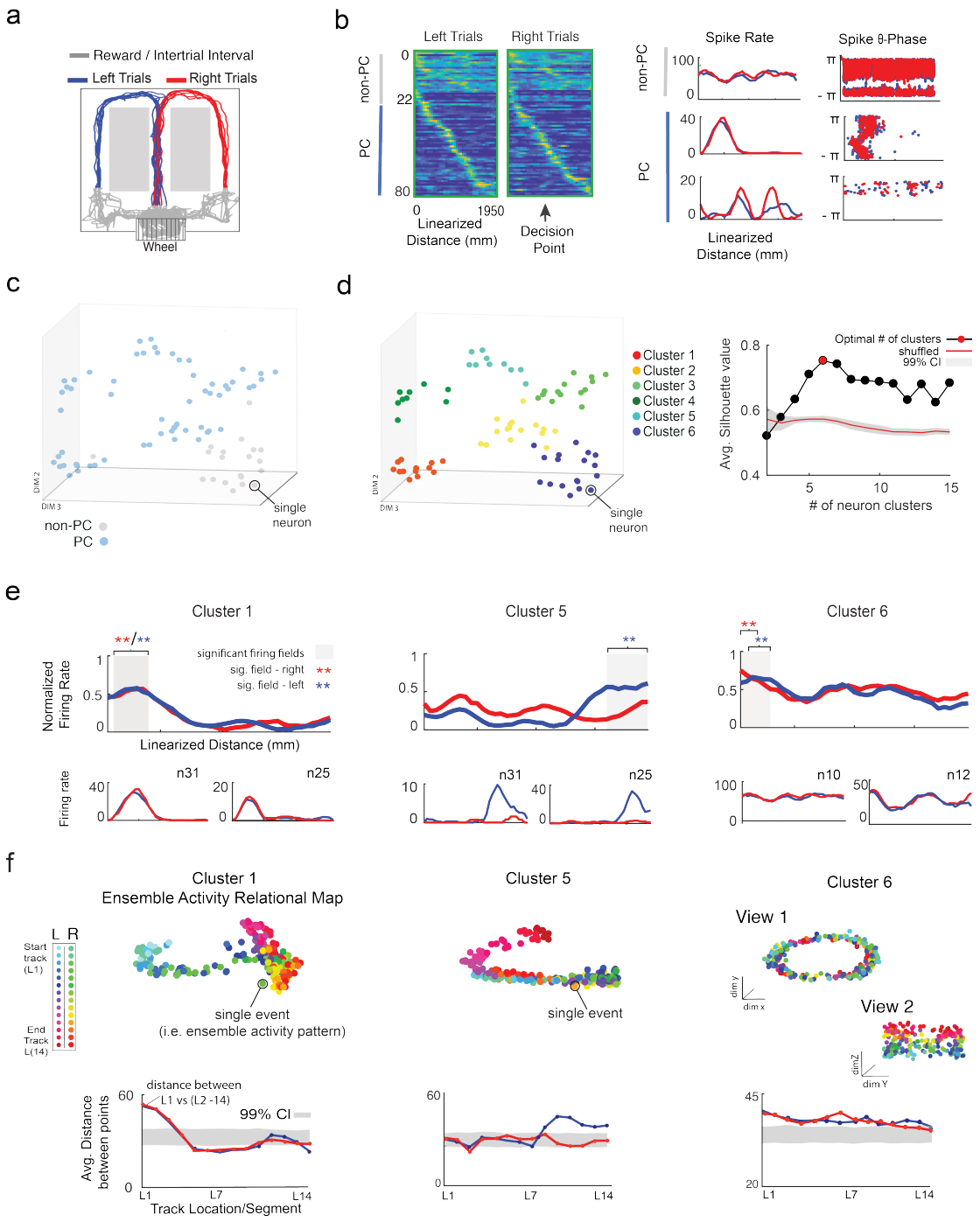
We applied SIMNETS to a publically available [38] dataset of N = 80 rat CA1 hippocampal neurons recorded using Multi-site silicon probes while the rat performed left/right-alternation navigation task in a 'figure-8' maze[46,47] (Fig 7*a*). The rat performed 17 correct trials (T = 17, trials) taking on average 4.3 seconds to reach the reward location at either end of the arms. The input to the SIMNETS algorithm was obtained by dividing the linearized trajectories [38] of the rat's path along the track into six equal segments and extracting 0.75 s spike train events, starting from the time that the rat entered each segment (*see* Methods for more details). This resulted in S = 102 spike train events from each of the N = 80 neurons. In order to validate SIMNETS performance, the CA1 neurons were characterized as having non-place cell (n = 22, non-PCs) or place cells (n = 58, PCs) based on their spatial firing properties and took note of their receptive field locations (Fig 6*b*; Supplementary Figure 6*c*) (see Methods for more details on neuron characterization and exclusion criteria).

SIMNETS analysis revealed an uneven distribution of neurons within the NS map (Fig 7*c),* with PCs and non-PCs occupying different regions: distances were significantly smaller between non-PCs (M = 33, STD = 18.55; *rank-sum* test, $p < 0.001$) than between non-PCs and PCs (M = 77, STD = 2 1). K-means analysis suggests that the neurons formed six separate clusters (Fig 7*d*). One of the putative sub-ensembles was composed almost entirely (96%) of non-PCs, while the other five clusters were either entirely or almost entirely (> 92%) composed of PCs. An inspection of the ensemble firing rate maps indicates that each of the 'PC' clusters were comprised of neurons with over-lapping or partially overlapping place-fields (Fig 7*e;* Supplementary Fig. 6*b*). PCs with single and multiple peaks in their spatial firing maps were found within the same ensemble if they shared a common firing field (for example, see Supplementary Fig. 6*c*, cluster-2 and 4). Interestingly, despite being made up of neurons that lack place-dependent signals, the cluster-6 spatial firing map exhibited a single significant firing field (Fig 7*e*, *last column).* In order to get a better understanding of the computational properties of this cluster in relation to the other detected clusters, we examined the ensemble activity patterns of each cluster using a spike train relational analysis framework [30] (*see* Methods).

The ensemble spike train similarity analysis (described in detail in Vargas-Irwin et al., 2015) generates Ensemble Activity Similarity maps similar to those presented in Fig. 1*c*, but encompassing the activity of multiple neurons rather than just one. Each point in these

ensemble activity Similarity maps corresponds to activity patterns across all neurons in a particular ensemble (Fig. 7*f*). As expected from the previous analysis, the topology of the Ensemble-Activity Relational map for the 'place-field cell-assemblies' (clusters 1 and 3) (Fig 7*f, columns 1 and 2*) captured the modulation of the ensembles firing patterns as the rat traversed the neurons' place-fields (Fig 7*f, columns 1 and 2*).

Figure 7|Hippocampal neurons during left-right alternation task.



a. Maze showing rats position in a 'figure-8' maze during left-right alternation task. Red and blue lines show the rat's location during correct right and left trials (T = 18, trials), respectively (similar color convention

used throughout b, e-f). Gray lines show the rats location during reward and inter-trial interval period. **b.** *Left*: Normalized firing rates are shown for each neuron (N = 80, neurons) as a function of linearized distance on track (50 mm bins, 150mm Gaussian smoothing) during left and right trials. Neurons were ordered according to the latency of their peak response along the track and according to their characterization as a non-place cell (n = 22, non-PC) or a place cell (n = 58, PC). Arrow indicates the left/right decision point in maze. *Right*: Spike rate maps and spike theta-phase spatial maps for one example non-PC (*top row*) and two example PCs (bottom *rows*). **c-d.** SN NS map with neurons labeled as non-PCs (gray points) and PCs labels (blue points) (**c**) and k-means clustering labels (**d,** *left*) and Silhouette plot (**d**, *right*). An optimal number of k = 6 neuron clusters were detected in the NS map. **e.** Ensemble spatial firing rate maps shown for three example SIMNETS clusters (top row), and the spatial firing map for two representative neurons from each example cluster (*middle, bottoms rows*). Gray bands indicate the significant firing fields (rank-sum, $p < 0.01$, in-field vs. out-of-field comparison) for the left and right trials (blue and red asterisks)**.** **f**. *Top row*: Ensemble Activity Relational map showing the similarity between ensemble activity patterns as a function of the rat's location on the linearized track during left and right trials (L1 - L14). Each point represents an ensemble activity pattern on a single trial and the distance between points represents the similarity of the ensemble spiking patterns. Different colors correspond to the different locations on the left and right track. *Bottom row*: distance plots showing the mean distance between points corresponding to each event type and all other points (e.g., distance between L1 points and all other points). Gray bands represent the 99% confidence interval of a null-distribution (bootstrap analysis).

Interestingly, the neuron cluster that was primarily non-PCs (cluster 6), had a 'torus-shaped' topology that exhibited variance along the z-dimension according to rat's position along the track (Fig 7*f,* view 2), and variance in the *x*- and y-dimension (Fig 7*f,* view 2) according to an unknown variable, or variables. This result suggests that the activity of the non-PC sub-ensemble displays dynamics that may reflect a non-spatial task-variable or potentially the intrinsic dynamics of the circuit. The toroid structure of the ensemble activity relational map suggests that the unknown variable is likely to be periodic in nature. Although this phenomenon warrants further investigation, it is outside of the scope of the present work. In general, the results from this dataset highlight the advantages of applying SIMNETS to neural recordings where the tuning properties of the neurons are not readily apparent or known *a priori*.

Computational efficiency and analysis run-time

The SIMNETS algorithm processed each of the synthetic, M1, and Hippocampal datasets in under 5 seconds. Because of a larger trial number, the V1 dataset was processed in a relatively slower time of around 20 seconds. In general, SIMNETS' run-time for a dataset of 100 neurons, with 100 spike trains per neuron, takes approximately ~4 s (see Supplementary Fig. 8a). Importantly, the computational complexity of the algorithm scales almost linearly with neuron number and quadratically with the number of spike trains, meaning that datasets of up to 1000 neurons can be analyzed in a reasonable amount of time (~ 4 minutes). By comparison, calculating the pairwise cross-correlation [39,48] for 100 and 1000 neurons would take approximately 6 minutes and 6 hours, respectively, using the same hardware (supplementary Fig. 8*b*).

Discussion

Summary of Findings

Advances in multi-electrode recording technology have now made it possible to record or image from 100s and even 1000s of single-units simultaneously [7,49–51]. By contrast, the development of analytical tools capable of parsing out the complexity of large-scale neural activity patterns have lagged behind advances in recording technology. Here, we introduce a computationally efficient and scalable method to quantify the functional similarity between individual simultaneously recorded neurons, allowing us to identify and visualize computationally related sub-networks within large neuronal ensembles.

Our analysis of simulated data with known ground truth demonstrates that SIMNETS is capable of organizing the neurons into functionally related sub-nets, even when computationally equivalent neurons utilize very different encoding schemes (e.g., rate, temporal, or mixed encoding schemes). Our analysis of V1 and M1 recordings shows that SIMNETS can generate neuron maps that capture the computational properties of cortical neurons without imposing stimulus or movement driven tuning models *a priori*. The tuning functions for single units in these two areas have been extensively studied, allowing us to validate the performance of the SIMNETS algorithm. Our results with the hippocampal CA1 data suggest that it will be possible to use this approach to simplify the functional characterization of groups of neurons where the underlying tuning functions are unknown or very complex. Our results also suggest that SIMNETS may be able to detect functional sub-ensembles hypothesized to support ensemble place-coding[52], memory-recall[53], or complex feature conjunction[54]. Although it was beyond the scope of this report to demonstrate functional significance of the detected putative functional sub-ensembles, our results strongly suggest that sub-nets detected using SIMNETS are statistically and physiologically meaningful. Our particular choice of datasets allowed us to demonstrate that this method generalizes well to neural recordings from a variety of brain regions (including sensory, motor, and hippocampal areas) and across multiple species (including rat and non-human primate).

Comparison to existing methods

The concept of a low dimensional embedding that captures the functional relationship between neurons was introduced in the seminal papers by Gerstein & Aertsen [48,55]. They used a technique called 'Gravitational Clustering' (GC) to identify groups of neurons with synchronous spiking patterns. GC is based on an analogy of the physics of the gravitational forces governing the dynamics and interactions of macroscopic particles. It treats the N neurons as N particles moving within an N-dimensional space, where charges that influence the attractive and repulsive interactions between particles are dictated by the temporal dynamics of pairwise synchronous spiking activity between neurons. The end result is a visualization of particle clusters (and their trajectories) that represent dynamically evolving assemblies of synchronously-active neurons. Recent formulations of the GC algorithm have improved visualization and sensitivity, but retain the same basic strategy [11,34].

Several previous studies have used spike train metrics to identify putative functional sub-ensembles [56,57]. As with GC, these studies have operated under the general assumption that the detection of similar sequences of spike patterns across neurons is indicative of a potential functional link [57,58]With the simulated neuron population, we demonstrated that traditional methods relying on such an assumption could result in either a spurious fractioning or a collapse of functionally similar neurons into clusters that are primarily

defined by the neurons' spike statistics or encoding timescales (as opposed to their computational properties). In contrast, SIMNETS was capable of clustering the simulated neurons according to their ground-truth functional ensembles and, by varying the temporal sensitivity of the analysis, organizing the neurons within each sub-net according to their encoding timescales. This feature of SIMNETS could be particularly useful for determining if neurons with different encoding timescales – such as, for example, fast-spiking PV inhibitory neurons and slow-spiking pyramidal neurons – are involved different information processing operations [59,60].

The critical component that differentiates the SIMNETS framework is our novel application of spike train metrics. We emphasize that, unlike other related algorithms, SIMNETS does not directly compare spiking responses between neurons; instead, our approach compares intrinsic geometry of the output spaces of each neuron (represented by SSIM matrices), which reflects information-processing properties in a more general way (i.e. regardless of how the information is encoded). Our results demonstrate that estimates of correlation between SSIM matrices provide a simple, yet, powerful approach for quantifying the functional similarities between neurons. Our analysis strategy shifts the emphasis from detecting coincident or correlated activation to comparing the intrinsic structure of single-trial firing patterns. This critical difference allows our method to detect neurons with similar computational properties even if they do not display coincident or correlated spiking. Although other approaches techniques to visualize pairwise and high-order neuron interactions have been proposed[11,12,61–64] these methods can become mathematically intractable or computationally expensive when extended beyond a small numbers of neurons. The combination of a short processing time (< 5 second per 100 neurons) and a computational complexity that scales linearly with the size of the neuron population makes SIMNETS an extremely efficient and, thus, appealing tool for exploring very large-scale neuron populations.

Limitations of SIMNETS

Several important limitations of SIMNETS are worth noting. First, estimates of similarity using spike train metrics require that the time windows of interest be of equal length, making it difficult to compare neural responses with different time courses. This particular weakness is common to all trial-averaging models commonly used in the literature. Second, although the SIMNETS framework does not require *a priori* assumptions about the variables potentially encoded by neural activity, experimental design and data selection will still have a direct effect on the results obtained. For example, a set of neurons identified as a functional sub-net could separate into smaller groups with different computational properties when additional task conditions are added to the analysis. Thus, the functional properties identified using SIMNETS are only valid within the context of the data examined, and may not necessarily extrapolate to different experimental conditions. Third, it is likely that neuronal sub-nets are constantly re-arranged depending on ethological demands. A neuron could potentially be functionally interacting with one group of neurons for one computation (or moment in time) and then another group of neurons for another computation (or another moment in time). The current version of the SIMNETS algorithm was not designed to distinguish between such rapidly changing network membership. However, it is possible to apply the SIMNETS algorithm multiple times over different epochs in order to determine if different sub-nets are present across different

conditions. Our future work will focus on examining the temporal evolution of sub-net clustering using this approach. Fourth, the computations required to generate a SSIM matrix for a neuron scale quadratically with the number of trials. Despite this computational cost, SIMNETS scales much better than methods that require computing power-sets (and thereby grow exponentially with the number of neurons).

The role for SIMNETS to mitigate experimenter bias

A considerable amount of research in systems neuroscience has focused on identifying new classes of neurons based on their information processing properties. The standard approach for many of these experiments involves recording single unit activity while a certain experimental variable of interest is manipulated (for example, providing different stimuli, or eliciting different movements, etc.). Standard statistical tests (ANOVA, etc.) are then used to determine if each neuron displays significant changes in firing rate across the experimental conditions. The percentage of significant neurons is usually reported, and highlighted as a 'class' of neurons sensitive to the variable of interest. It is common to exclude neurons that do not reach statistical significance or cannot be fit using a predetermined model from further analysis. This approach is prone to both selection and confirmation bias, and ultimately produces 'classes' of neurons identified based on arbitrary statistical thresholds imposed on what are likely continuous distributions of properties [65,66].

The SIMNETS framework provides provide an efficient way to represent the computational structure of neuronal networks and quantitatively assess if neurons represent discrete functionally separate classes or alternatively span a continuous gradient of properties. In addition to providing a principled way to determine if a consistent organization of information processing modules can be found across sessions and subjects, we believe that the ability to intuitively visualize relationships within networks of neurons will provide a unique perspective leading to new data-driven hypotheses and experimental refinement.

Bibliography

1. Carandini, M. & Heeger, D. J. Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* **13,** 51–62 (2011).
2. Zagha, E., Ge, X. & McCormick, D. A. Competing Neural Ensembles in Motor Cortex Gate Goal-Directed Motor Output. *Neuron* **88,** 565–577 (2015).
3. Pastalkova, E., Itskov, V., Amarasingham, A. & Buzsáki, G. Internally generated cell assembly sequences in the rat hippocampus. *Science* **321,** 1322–7 (2008).
4. Harris, K. D. Opinion: Neural signatures of cell assembly organization. *Nat. Rev. Neurosci.* (2005). doi:10.1038/nrn1669
5. Brown, E. N., Kass, R. E. & Mitra, P. P. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nat. Neurosci.* (2004). doi:10.1038/nn1228
6. Briggman, K. L., Abarbanel, H. DI & Kristan, W. B. From crawling to cognition: analyzing the dynamical interactions among populations of neurons. *Current Opinion in Neurobiology* (2006). doi:10.1016/j.conb.2006.03.014
7. Buzsáki, G. Large-scale recording of neuronal ensembles. *Nat. Neurosci.* **7,** 446–451 (2004).
8. Wallace, D. J. & Kerr, J. N. D. Chasing the cell assembly. *Current Opinion in Neurobiology* (2010). doi:10.1016/j.conb.2010.05.003
9. Aertsen, A. M. H. J., Gerstein, G. L., Habib, M. K. & Palm, G. Dynamics of Neuronal Firing Correlation: Modulation of " Effective Connectivity ". *JOURNALOFNEUROPHYSIOLOGY Print. i n U.S.A* **61,** (1989).
10. Abeles, M. & Gat, I. Detecting precise firing sequences in experimental data. *J. Neurosci. Methods* (2001). doi:10.1016/S0165-0270(01)00364-8
11. Baker, S. N. & Gerstein, G. L. Improvements to the Sensitivity of Gravitational Clustering for Multiple Neuron Recordings. *Neural Comput.* **12,** 2597–2620 (2000).
12. Grün, S., Diesmann, M. & Aertsen, A. Unitary Events in Multiple Single-Neuron Spiking Activity: II. Nonstationary Data. *Neural Comput.* (2002). doi:10.1162/089976602753284464
13. Rgy, G., Iki, B. & Chrobak, J. I. Temporal structure in spatially organized neuronal ensembles: a role for interneuronal networks. *Curr. Opin. Neurobiol.* **5,** 504–510 (1995).
14. Humphries, M. D. Spike-Train Communities: Finding Groups of Similar Spike Trains. *J. Neurosci.* (2011). doi:10.1523/JNEUROSCI.2853-10.2011
15. Singer, W. & Gray, C. M. Visual Feature Integration and the Temporal Correlation Hypothesis. *Annu. Rev. Neurosci.* **18,** 555–586 (1995).
16. Singer, W. Neuronal Synchrony: A Versatile Code for the Definition of Relations? *Neuron* **24,** 49–65 (1999).
17. Von Der Malsburg, C. *The Correlation Theory of Brain Function*.
18. Brody, C. D. Correlations Without Synchrony. *Neural Comput.* **11,** 1537–1551 (1999).
19. Shadlen, M. N. & Movshon, J. A. Synchrony Unbound. *Neuron* **24,** 67–77 (1999).
20. Grün, S., Diesmann, M. & Aertsen, A. Unitary Events in Multiple Single-Neuron Spiking Activity: II. Nonstationary Data. *Neural Comput.* **14,** 81–119 (2002).
21. Cunningham, J. P. & Yu, B. M. Dimensionality reduction for large-scale neural recordings. *Nat. Neurosci.* **17,** 1500–9 (2014).
22. Machens, C. K., Romo, R. & Brody, C. D. Functional, but not anatomical, separation of &quot in prefrontal cortex. *J. Neurosci.* **30,** 350–60 (2010).
23. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent

computation by recurrent dynamics in prefrontal cortex. *Nature* **503,** 78–84 (2013).
24. Truccolo, W., Hochberg, L. R. & Donoghue, J. P. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. *Nat. Neurosci.* **13,** 105–111 (2010).
25. Churchland, M. M. & Shenoy, K. V. Temporal Complexity and Heterogeneity of Single-Neuron Activity in Premotor and Motor Cortex. *J. Neurophysiol.* **97,** 4235–4257 (2007).
26. Chelaru, M. I. & Dragoi, V. Efficient coding in heterogeneous neuronal populations. *Proc. Natl. Acad. Sci. U. S. A.* **105,** 16344–9 (2008).
27. Vargas-Irwin, C. E., Brandman, D. M., Zimmermann, J. B., Donoghue, J. P. & Black, M. J. Spike Train SIMilarity Space (SSIMS): A Framework for Single Neuron and Ensemble Data Analysis. *Neural Comput.* **27,** 1–31 (2015).
28. Victor, J. D. & Purpura, K. P. Metric-space analysis of spike trains: theory, algorithms and application. *Netw. Comput. Neural Syst.* (1997). doi:10.1088/0954-898X_8_2_003
29. Zaidi, Q. *et al.* Perceptual spaces: mathematical structures to neural mechanisms. *J. Neurosci.* **33,** 17597–602 (2013).
30. Vargas-Irwin, C. E., Brandman, D. M., Zimmermann, J. B., Donoghue, J. P. & Black, M. J. Spike Train SIMilarity Space (SSIMS): A Framework for Single Neuron and Ensemble Data Analysis. *Neural Comput.* **27,** 1–31 (2015).
31. Lehky, S. R., Sereno, M. E. & Sereno, A. B. Population Coding and the Labeling Problem: Extrinsic Versus Intrinsic Representations. *Neural Comput.* (2013). doi:10.1162/NECO_a_00486
32. Victor, J. D. & Purpura, K. P. Nature and Precision of Temporal Coding in Visual Cortex: A Metric-Space Analysis. *J. Neurophysiol. Pt-intd U.S.A* **76,** (1996).
33. Kriegeskorte, N. & Kievit, R. A. Representational geometry: integrating cognition, computation, and the brain. *Trends Cogn. Sci.* **17,** 401–12 (2013).
34. Stuart, L., Walter, M. & Borisyuk, R. Visualisation of synchronous firing in multi-dimensional spike trains. in *BioSystems* (2002). doi:10.1016/S0303-2647(02)00084-9
35. Lopes-dos-Santos, V., Ribeiro, S. & Tort, A. B. L. Detecting cell assemblies in large neuronal populations. *J. Neurosci. Methods* **220,** 149–166 (2013).
36. Kohn, A., Smith, M. A. (2016). Utah array extracellular recordings of spontaneous and visually evoked activity from anesthetized macaque primary visual cortex (V1). *CRCNS.org* (2016). Available at: http://dx.doi.org/10.6080/K0NC5Z4X.
37. Rao, N. G. & Donoghue, J. P. Cue to action processing in motor cortex populations. *J. Neurophysiol.* **111,** 441–53 (2014).
38. Pastalkova E, Wang Y, Mizuseki K, B. G. Simultaneous extracellular recordings from left and right hippocampal areas CA1 and right entorhinal cortex from a rat performing a left / right alternation task and other behaviors. (2015). Available at: http://dx.doi.org/10.6080/K0KS6PHF.
39. Smith, M. A. & Kohn, A. Spatial and temporal scales of neuronal correlation in primary visual cortex. *J. Neurosci.* **28,** 12591–603 (2008).
40. Amirikian, B., Georgopoulos, A. P. & Georgopulos, A. P. Directional tuning profiles of motor cortical cells. *Neurosci. Res.* **36,** 73–9 (2000).
41. Rao, N. G. & Donoghue, J. P. Cue to action processing in motor cortex populations. *J. Neurophysiol.* **111,** 441–453 (2014).
42. Vargas-Irwin, C. E., Franquemont, L., Black, M. J. & Donoghue, J. P. Linking Objects to Actions: Encoding of Target Object and Grasping Strategy in Primate

Ventral Premotor Cortex. *J. Neurosci.* **35,** 10888–97 (2015).
43. Vargas-Irwin, C. E., Brandman, D. M., Zimmermann, J. B., Donoghue, J. P. & Black, M. J. Spike Train SIMilarity Space (SSIMS): A Framework for Single Neuron and Ensemble Data Analysis. doi:10.1162/NECO_a_00684
44. Lillicrap, T. P. & Scott, S. H. Preference Distributions of Primary Motor Cortex Neurons Reflect Control Solutions Optimized for Limb Biomechanics. *Neuron* **77,** 168–179 (2013).
45. Scott, S. H., Gribble, P. L., Graham, K. M. & Cabel, D. W. Dissociation between hand motion and population vectors from neural activity in motor cortex. *Nature* **413,** 161–165 (2001).
46. Pastalkova, E., Itskov, V., Amarasingham, A. & Buzsáki, G. Internally generated cell assembly sequences in the rat hippocampus. *Science* **321,** 1322–7 (2008).
47. Mizuseki, K., Sirota, A., Pastalkova, E. & Buzsáki, G. Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop. *Neuron* **64,** 267–280 (2009).
48. Gerstein, G. L. & Aertsen, A. M. Representation of cooperative firing activity among simultaneously recorded neurons. *J. Neurophysiol.* **54,** (1985).
49. Jun, J. J. *et al.* Fully integrated silicon probes for high-density recording of neural activity. *Nature* **551,** 232–236 (2017).
50. Csicsvari, J. *et al.* Massively Parallel Recording of Unit and Local Field Potentials With Silicon-Based Electrodes. (2003). doi:10.1152/jn.00116.2003
51. Kim, D. H. *et al.* Pan-neuronal calcium imaging with cellular resolution in freely swimming zebrafish. *Nat. Methods* **14,** 1107–1114 (2017).
52. Park, E., Dvorak, D. & Fenton, A. A. Ensemble Place Codes in Hippocampus: CA1, CA3, and Dentate Gyrus Place Cells Have Multiple Place Fields in Large Environments. *PLoS One* **6,** e22349 (2011).
53. Pastalkova, E., Itskov, V., Amarasingham, A. & Buzsaki, G. Internally Generated Cell Assembly Sequences in the Rat Hippocampus. *Science (80-.).* **321,** 1322–1327 (2008).
54. Geiller, T., Fattahi, M., Choi, J.-S. & Royer, S. Place cells are more strongly tied to landmarks in deep than in superficial CA1. *Nat. Commun.* **8,** 14531 (2017).
55. Gerstein, G., Perkel, D. & Dayhoff, J. Cooperative firing activity in simultaneously recorded populations of neurons: detection and measurement. *J. Neurosci.* **5,** (1985).
56. Lopes-dos-Santos, V., Ribeiro, S. & Tort, A. B. L. Detecting cell assemblies in large neuronal populations. *J. Neurosci. Methods* **220,** 149–166 (2013).
57. Humphries, M. D. Spike-Train Communities: Finding Groups of Similar Spike Trains. *J. Neurosci.* **31,** 2321–2336 (2011).
58. Fujisawa, S., Amarasingham, A., Harrison, M. T. & Buzsáki, G. Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nat. Neurosci.* **11,** 823–833 (2008).
59. Wilson, N. R., Runyan, C. A., Wang, F. L. & Sur, M. Division and subtraction by distinct cortical inhibitory networks in vivo. *Nature* **488,** 343–8 (2012).
60. Neske, G. T., Patrick, S. L. & Connors, B. W. Contributions of diverse excitatory and inhibitory neurons to recurrent network activity in cerebral cortex. *J. Neurosci.* **35,** 1089–105 (2015).
61. Gerstein, G. L. & Kirkland, K. L. Neural assemblies: Technical issues, analysis, and modeling. *Neural Networks* (2001). doi:10.1016/S0893-6080(01)00042-9
62. Martignon, L. *et al.* Neural coding: higher-order temporal patterns in the neurostatistics of cell assemblies. *Neural Comput.* **12,** 2621–53 (2000).

63. Lopes-Dos-Santos, V., Ribeiro, S. & Tort, A. B. L. ARTICLE IN PRESS Detecting cell assemblies in large neuronal populations. *J. Neurosci. Methods* (2013). doi:10.1016/j.jneumeth.2013.04.010
64. Shimazaki, H., Amari, S. ichi, Brown, E. N. & Grün, S. State-space analysis of time-varying higher-order spike correlation for multiple neural spike train data. *PLoS Comput. Biol.* (2012). doi:10.1371/journal.pcbi.1002385
65. Blanchard, T. C., Piantadosi, S. T. & Hayden, B. Y. Robust mixture modeling reveals category-free selectivity in reward region neuronal ensembles. *J. Neurophysiol.* **119,** 1305–1318 (2018).
66. Chandrasekaran, C., Peixoto, D., Newsome, W. T. & Shenoy, K. V. Laminar differences in decision-related neural activity in dorsal premotor cortex. *Nat. Commun.* **8,** 614 (2017).

SIMNETS: a computationally efficient and scalable framework for identifying networks of functionally similar neurons

Online Methods

Jacqueline B. Hynes [1, 2], **David M. Brandman**,[2, 3], **Jonas B. Zimmerman**[4], **John P. Donoghue**[1, 2, 4, 5], **Carlos Vargas-Irwin** [*, 1, 2]

1 Department of Neuroscience, Brown University, Providence, RI, USA.
2 Robert J. and Nancy D. Carney Institute for Brain Science, Brown University, Providence, RI, USA
3 Department of Surgery (Neurosurgery), Dalhousie University, Halifax, NS, Canada.
4 WYSS Institute, Chemin des Mines 9, CH-1202, Geneve, Switzerland.
5. Department of Engineering, Brown University, Providence, RI, USA.

∗ Correspondence to: carlos_vargas_irwin@brown.edu

Keywords: neural ensembles, dimensionality reduction, clustering algorithm, neuronal networks

SIMNETS Algorithm Implementation

Here, we provide a short description of the steps in the algorithm and a detailed description of the methods used to implement each step of the SIMNETS framework:

Step 1: Calculate Distances Between Within-neuron Spike Train Pairs

Pairwise spike train distances are calculated between within-neuron spike train pairs (S) using a spike train metric. This results in separate SxS Spike train Similarity (SSIM) matrices for each neuron. Here, we use the Victor-Purpura Spike train metric [1,2].

Step 2: Spike Train Similarity Matrix Correlation

Pairwise measures of correlation are calculated between all pairs of single-neuron SSIM matrices, resulting in a single NxN Neuron Correlation (NC) Matrix. Here, we use Pearson's Correlation.

Step 3: Dimensionality Reduction

The high-dimensional, NxN Neuron Correlation Matrix is projected down into a desired number of d dimensions and visualized in a scatter plot, resulting in what we refer to as the Neuron Similarity (NS) Map. The dimensionality reduction step is carried out using t-distributed Stochastic Neighbor Embedding (t-SNE)[3].

Step 4: Cluster Detection and Statistical test

Putative functional ensembles are detected in the $N \times d$ Neuron Similarity Map using the unsupervised k-means clustering algorithm [4]. The number of clusters in the data is determined using a silhoutte analyis[5] and the significance of the number of detected clusters is determined using a shuffle-based procedure.

The user selected parameters of the SIMNETS algorithm include: a) the VP spike train metric, q, which operationally defines the temporal resolution over which the similarity of two spikes trains are tested, and b) the t-SNE parameter, perplexity, which influences the number of *effective* nearest neighbors (i.e., neurons) included in calculations that results in the low-dimensional neuron map.

Step 1: Victor-Purpura Spike Train Metric

The Victor-Purpura (VP) metric is a cost-based spike train distance function (D) that describes the similarity between pairs of spike trains (common neuron) in terms of their 'edit-distances'. A single distance value (d) is assigned to each pair of spike trains through a process that involves calculating the minimum total 'cost' (c) of the edit-steps needed to transform spike train A into spike train B:

$$\boldsymbol{d}(\mathbf{A},\mathbf{B}) = \boldsymbol{min}\{\textstyle\sum_{j=0}^{m-1} \boldsymbol{c}\,(\boldsymbol{s}_{j,}\boldsymbol{s}_{j+1})\}, \qquad \textbf{(1)}$$

where $\{S_0, S_1, ..., S_m\}$ is the series of intermediate spike trains created after performing a single edit step. The list of possible edit-steps used in the VP transformation include: (1) inserting a spike, (2) deleting a spike, and (3) shifting a spike in time. Inserting or deleting a spike has a cost of $c = 1$, and shifting a single spike in time has a cost proportional to the amount of time that it is moved ($c = q\Delta t$). The set of edits-steps associated with the minimum total edit-cost defines the shortest path between two points (spike trains) in the neuron's spike train metric-space.

The q parameter influences the relative importance of spike count and spike time differences when assessing spike train similarities. When $q = 0$, the cost of shifting a spike to a desired location will always be cheaper than deleting and re-inserting a spike in a spike train. Thus, for $D_{[q=0]}$, the total minimum cost is a function of the difference in the number of spikes between the spike trains. As the q value is increased beyond zero, spike time jitter begins to impact the cost of matching the spike trains. For example, if $q = 10$, shifting a spike by 0.15 s will have a cost of $c = 1.5$, which is still just under the cost of deleting and re-inserting spike, i.e., $c = 2$, making it the more cost effective option. However, if $q = 15$, the deleting and reinserting a spike will become the cheaper option. This means that if we were matching two temporally jittered spike trains with a similar number of spikes, the assigned spike train distance would jump from a small to a high value as q increases (due to the increasing cost associated with shifting a spike). On the other hand, if we were matching two spike trains that differed only by spike number, i.e., no temporal jitter, the cost of shifting a spike would not impact the total cost of matching the spike trains, and so we would not expect a jump in the assigned spike train distances with an increase in q value. In this way, q controls the temporal resolution of the spike train comparison. In the context of the SIMNETS algorithm, a low q parameter will bias the algorithm towards groupings neurons based on the 'information' encoded over coarse timescales, whereas a high q parameter will bias the algorithm towards groupings neurons based on the information encoded over coarse and fine timescales.

Step 2: Spike Train Similarity Matrix Correlation

We characterize the functional similarities between neurons by calculating pairwise measures of correlation between all pairs of SSIM matrices. We used Pearson's Correlation (r) to compare the SSIM matrices in this report (Spearman's correlation or another correlation statistic can be used if an assumption of linearity between the SSIM matrices

cannot be made). The formula for calculating Pearson's r between a pair of SSIM matrices, $A = (a_{ij})$ and $B = (b_{ij})$ is given as:

$$r(\varphi) = \beta + h.exp\big(h.cos(\varphi - \mu)\big), \tag{2}$$

where *cov* is the covariance and σ is the standard deviation. This results in an *N*x*N* Correlation matrix, where each matrix entry corresponds to the correlation between a given pair of SSIM matrices, and each column (or row) of the matrix could be interpreted as the intrinsic coordinates of a single neuron in an *N*-dimensional space.

Step 3: t-SNE Dimensionality Reduction algorithm

In broad terms, the goal of the dimensionality reduction step is to reduce the number of variables required to represent each neuron's *N* dimensional correlation vector (*step 2*), i.e., its coordinates in the high-dimensional neuron space. This step improves clustering performance and allows us to simultaneously visualize the relationships between all neurons in a low-dimensional map. We used the t-distributed Stochastic Neighbor Embedding (t-SNE) dimensionality reduction algorithm, since it is capable of preserving local densities of the high-dimensional data, while also revealing global structure such as the presence of clusters at several scales [3]. It is also particularly well suited for visualizing high-dimensional data with varying cluster densities.

A t-SNE transform is calculated through a process that involves 1) converting the sets of high- and low-dimensional correlation/distance measures into sets of joint probability distributions that describe the 'similarity' between the data points in the respective high and low dimensional spaces, and 2) minimizing the Kullback-Leibler divergence[6] between the sets of joint probabilities in the high-dimensional space and the low-dimensional map via gradient descent.

In the first step, the similarity of the data point w_j to w_i in the high dimensional space is modeled as the conditional probability, $p_{j|i}$, that w_i would pick w_j as its neighbor if neighbors were (stochastically) picked in proportion to their probability density, P_i, under a Gaussian kernel centered at w_i. Mathematically, the condition probability $p_{j|i}$ is given by:

$$p_{i|j} = = exp\frac{\left(-||wi-wj||^2/2\sigma_i^2\right)}{\sum_{k\neq i} exp\left(-||wi-wk||^2/2\sigma_i^2\right)} \tag{3}$$

where σ_i is the variance of the Gaussian centered on w_i. Importantly, the variance of the Gaussian kernel adapts to the local density of the data around each point to produce a probability density (P_i) with a fixed perplexity (*perp*). The *perp* hyper-parameter specifies the number of effective nearest neighbors included in the probability calculations, where smaller perplexity values result in maps that are biased towards representing local relationships and larger values result in maps that represent local relationships with increasing consideration to any global structure that might exist. More formally, perplexity is a measure of information describing how well a probability distribution predicts a sample and is defined as 2 $H^{(Pi)}$, where $H^{(Pi)}$ is the Shannon entropy of *Pi* measured in bits.

Next, to simplify the optimization of the low-dimensional representation during the gradient descent, the conditional probabilities p_{ji} are converted to joint probabilities using a symmetrized version of the conditional probabilities, $p_{ji} = p_{j|i} + p_{i|j} / 2n$.

The low-dimensional probability (z_{ij}) distributions takes a similar form to p_{ji} except that a long-tailed Students t-distribution replaces the Gaussian distribution:

$$Z_{j,i} = \frac{\left(1+\left||ui-uj|\right|^2\right)^{-1}}{\sum_{v\neq i}\left(1+\left||uv-ul|\right|^2\right)^{-1}}. \tag{4}$$

The long-tail of the Students t-distribution ensures that moderately close points in the high-dimensional space are modeled by larger distance in the low-dimensional space, and as a result, eliminates any unwanted attractive forces between moderately dissimilar points that would have otherwise resulted in 'crowding' [3] in the low-dimensional representation between neighboring clusters with very different densities. Additionally, this particular form of the Students t-distribution (single degree of freedom) ensures that the low-dimensional representation is (mostly) scale invariant, meaning that clusters of points will interact in the same way as individual points. The effect is that the functional relationships between neurons are preserved across multiple scales of organization[3].

The overall aim of t-SNE is to find a low-dimensional data representation that minimizes any mismatch between the high-dimensional joint probability density, P, and the Students-t based joint probability distribution, Z. The minimization of the cost function is performed via gradient decent, with a gradient given by the equation:

$$\frac{\delta C}{\delta y_i} = 4\sum_j \left(p_{ij} - z_{ij}\right)\left(u_i - u_j\right)\left(1 + \left\|u_i - u_j\right\|^2\right)^{-1}. \tag{5}$$

In order to reduce computational complexity of this step, we perform a preliminary round of dimensionality reduction using principal component analysis (PCA) to project the $N\times N$ Neuron Correlation Matrix into smaller dimensional space (e.g., 50-d). The t-SNE algorithm then refines the resulting linear transform by minimizing the single Kullback-Leibler divergence between P and Q over multiple iterations. Seeding with a low-dimensional PCA projection also ensures that the algorithm converges to the same solution across repeated runs of the algorithm. This step results in the dxN Neuron Similarity map.

Step 4: k-means Clustering Algorithm, Silhouette Analysis, and Significance Test

k-means algorithm

The k-means algorithm is an unsupervised clustering method that partitions data into k clusters. We elected to use the k-means algorithm to cluster neurons in the NS map into putative functional ensembles because of its efficiency and its empirically evaluated performance in detecting functional groupings of neurons.

k-means clustering aims to partition the t-SNE outputs into *k* number of clusters, such that each data point belongs to a cluster with the nearest mean (see next section for selection of *k* value). The algorithm works iteratively to assign each data point (u_i) to one of the *C* centroids based on proximity, where the centroids have been initialized at the random locations C_1, C_2..., C_m. After all points are assigned, new centroids are calculated from the assigned data points. This procedure is repeated for a specific number of iterations, e.g., 100, or until the centroids no longer move between iterations. The algorithm aims to minimize the sum of the squared error (SSE) between each data point:

$$SSE = \sum_{j=1}^{m} \sum_{i=1}^{l} \left\| u_i^{(j)} - c_j \right\|^2 \tag{6}$$

Silhouette Analysis

We used a silhouette analysis to assess the quality of the k-means clustering partitions across a range of values of *k* [5], with the goal of finding an optimal partition number for the data. A silhouette value (h_i) is a measure of how similar point y_i is to other data points in its assigned cluster c_j as compared to other clusters:

$$h_i = \frac{(b_i - a_i)}{max(a_i, b_i)}, \tag{7}$$

where a_i is the average distance from u_i to other points in its assigned cluster c_j, and b_i is the average distance from u_i to points in the other clusters, minimized over all possible cluster configurations. An optimal number of clusters $\hat{k}$ is the value of *k* that maximizes the average silhouette $(\hat{h})$ value for $k = 2..., k_f$. Silhouette values ranges from −1 to +1, where a high value indicates that u_i is well matched to its own cluster and poorly matched to neighboring clusters. In general, a maximized average silhouette below 0.25 indicates data that are not structured while a value below 0.5 would indicate poor or potentially spurious clusters. In the next section, we outline a procedure for testing the statistical significance of the cluster number to determine if the data can be partitioned into statistically meaningful clusters.

SIMNETS Significance Test

We developed a significance test for the purpose of determining the likelihood of detecting a given number of clusters by chance under the null hypothesis that there is no genuine relationship between the inherent structures of the SSIM matrices.

The significance test involves generating a null-distribution of silhouette values based on shuffled data across a range of *k* values. In SIMNETS, functional similarities are captured by the pairwise measures of correlation between the single neuron SSIM matrices. Our test relies on a shuffling procedure that destroys the pairwise dependencies between the SSIM matrices, and subsequently, any significant measures of correlation in the Neuron Correlation Matrix.

Our approach is inspired by the Mantel test [7], a permutation based procedure that tests the significance of the observed correlation between two symmetrical matrices. The intuition

of a Mantel test is that if a significant relationship exists between the values of matrix A and matrix B, then randomizing the rows and columns of one matrix will destroy any existing dependencies. As a result, the correlation between the shuffled matrix pair will tend to be lower than the original correlation value observed between the un-shuffled matrix pair. The probability of observing $r_{A.B}$ is then calculated as the proportion of permutations for which the shuffled correlation measures are smaller than or equal to $r_{A.B.}$ Here, we carry out a similar permutation operation on the SSIM matrices, in that we destroy any dependencies that exist between the matrices; however, we use the average silhouette value as the test statistic, rather than the correlation values, as is the case with the Mantel test.

The procedure involves a symmetrically shuffling of the rows/column of each N SSIM matrix separately, and re-calculating the pairwise correlations between the SSIM matrices to generate a new NxN Correlation Matrix. This NxN correlation matrix is then transformed into a new NR map using t-SNE (i.e., step 3), and a new set of silhouette values is calculated (i.e., step 4) for the range of tested k values. This procedure – SSIM matrix shuffling, steps 3 and step 4 from SIMNETS – are repeated to generate a null-distribution of average silhouette values that is approximately normally distributed (e.g., 1000+ iterations). If the observed maximized average silhouette value $\hat{h}$ falls above the empirically calculated $(1-\alpha)100\%$ confidence interval, then the detected number of $\hat{k}$ clusters is considered statistically meaningful.

Explanation of relevant symbols associated with SIMNETS algorithm

Table 1: explanation of relevant symbols

Symbol	Description	Calculation/Selection
q	temporal sensitivity value for Victor-Purpura spike train metric	hyper-parameter (user specified)
$perp$	perplexity: a measure of information that controls the number of effective nearest neighbors in t-SNE dimensionality reduction algorithm	hyper-parameter (user specified)
d	desired dimensions for low-dimensional Neuron Similarity Map	hyper-parameter (user specified)
k	k-means partition number/ range of cluster numbers to test for during k-means clustering	range of k-values (user specified)
$\hat{k}$	Optimal number of detected clusters Estimated from Silhouette analysis over a range of k values	output (eqt. 6)
$\hat{h}$	maximized average silhouette value	output (eqt. 7)

Hardware, Software, and Processing Time

All analyses were run on a Dell PC with an Intel Xeon® Processor and 24 GB of RAM. All analyses were run using MATLAB® software from MathWorks, version 9.4, R2018. Armadillo, a C ++ linear algebra library (called from within MATLAB) performed some of the main matrix operations.

On this hardware, analysis run-time for a dataset of 100 neurons (100 one-second spike trains per neuron) takes approximately 3.0 seconds, while 1000 neurons takes approximately 4 minutes (Supplementary Fig. 8). By comparison, calculating the pairwise cross-correlation values (without a jitter/shuffle correction procedure) for 100 or 1000 neurons takes approximately 6 minutes and 6 hours, respectively, on the same hardware. The computational complexity of the SIMNETS algorithm scales almost linearly with neuron number and exponentially with the number of spike trains Introducing a new neuron only requires generating a single new SSIM matrix, however, adding a new trial requires generating a new SSIM matrix for each neuron. The low computational cost of adding new neurons means that datasets with large numbers of neurons could be functionally categorized and clustered in a reasonable amount of time (< 1 hr for 5,000 neurons).

A live code tutorial is available for download at: .. (see peer-reviewed publication for link).

Direct Comparison Method

We emphasize that the SIMNETS algorithm does not directly compare the firing patterns *between* different neurons. Instead, pairwise comparisons are performed between common spike trains of a single neuron, on a neuron-by-neuron basis. The between neuron comparisons are then made between all pairs of the single neuron SSIM matrices. This allows the algorithm to find neurons that generate a set of spike trains with common signature spike train geometries (i.e., set of distances), rather than grouping neurons based on the degree of coordination between their moment-to-moment firing patterns. In order to evaluate the effectiveness of this strategy, we compared the performance of SIMNETS to a representation of traditional approaches that directly compare the spike trains of different neurons.

The 'Direct Comparison Method' (DCM) computes pairwise spike train similarities between matching 'trials' between neuron pairs. In contrast to the SIMNETS method, the t-SNE dimensionality reduction step is applied to an *N*x*N* matrix of distance values, rather than SSIM correlation values. That is, for a set of neurons $N = \{n_1, n_2, \ldots n_k\}$, the DCM methods builds a *N*x*N* matrix, M, where each $M_{x,y}$ entry is the sum of the spike train distances between the spike trains of neuron n_x, $S = \{S_1, S_2, \ldots, S_j\}$, and the spike trains of neuron n_y , $U = \{U_1, U_2, \ldots, S_j\}$:

$$I_{spike} = \sum_{i=1}^{l} P_i \frac{v_i}{V} log_2 \frac{v_i}{V}, \tag{8}$$

where D is a vector of Victor-Purpura spike train distance of length *j* (equation 1).

Synthetic Dataset – Data Simulation and Analysis

Spike train Simulation We simulated the spiking activity of a population of N = 180 synthetic neurons that consisted of 3 functionally distinct 'ensembles' (E_1, E_2, E_3) of 60 neurons. Each functional ensemble was designed to produce similar spike-trains for two non-modulating conditions, referred to as the 'baseline' conditions, and a different pattern for a third condition, referred to as the 'modulating' condition. For example, ensemble E_1 was modulated during condition A and exhibited the same baseline activity spike pattern during both conditions B and C, whereas ensemble E_2 was modulated during condition B and exhibited the same baseline spike pattern during conditions A and C, etc. Each E_i ensemble was further divided into three sub-groups of n = 20 neurons, where each sub-group altered their spike-train patterns between the active and baseline states according to one of three different encoding strategies:

(1) Rate coding: firing rate increased by 50% for the modulating condition (all spike times were randomly chosen)
(2) Temporal coding: the two baseline conditions and the modulating condition were associated with specific (randomly generated) temporal sequences of spikes. The number of spikes was kept constant across baseline and modulating conditions. Spike times were jittered by +/- 50 ms for each trial.
(3) Mixed temporal/rate coding: Similar to the temporal coding, but the spikes were jittered in a temporal window of 5 ms. Additionally, the modulating condition included 25% more spikes.

In order to simulate stochastic variation in spiking patterns, 50% of the spikes were randomly removed for each condition. A total of 30 seconds of simulated recording time was generated, with the trial condition changing every second between A, B, and C patterns.

SIMNETS Cluster Characterization We demonstrate SIMNETS' ability to cluster functionally similar neurons in the synthetic dataset by comparing the pairwise similarity measures between and within ground-truth functional ensembles. We compare the distributions of the pairwise correlation values from the Neuron Correlation/Distance Matrices for neurons from the same functional ensemble ('Within' ensemble pairs) and different functional ensembles ('Between' ensemble pairs). A rank-sum statistical test was carried out on the Within and Between distributions of similarity values, using an alpha value of $\alpha = .001$.

Neural Datasets – Task Description and Data Analysis

Primate Primary Visual Cortex

Task Description We analyzed a previously described dataset of 112 primary visual (V1) single-units (which we refer to as neurons) recorded in an anesthetized *Macaca fascicularis* using a 96-channel microelectrode array [8,9]Briefly, sinusoidal gratings were presented at 6 different orientations θ = {0°, 30°, 60°, 90°, 120°, 150°} and 2 drift directions (rightward and leftward drift, orthogonal to orientation). Each stimulus was presented 112 times for 1.28 seconds. The position and size of the stimuli was sufficient to cover the receptive

fields of all recorded neurons. For more details on the data processing and task design, see Smith and Kohn (2008) and Kohn and Smith, (2016).

Single Neuron Functional Characterization We characterized the preferred orientation of each V1 neuron by fitting a Gaussian distribution to the firing rate function R:

$$R(\theta) = \hat{A}.exp^{\frac{-(\theta-\mu)^2}{2\sigma^2}}, \tag{9}$$

where θ is the stimulus orientation, $\hat{A}$ is the peak response, μ the mean, and σ^2 is variance of the Gaussian. The function takes on a maximum value at $\theta = \mu$, for $\theta = [0, 180)$, which corresponds to the neuron's preferred orientation. We choose not to use drift-direction preferences when characterizing the functional properties of the neurons[10,11] as only a very small percentage exhibited significant differences in the magnitude of their peak responses for drift-direction.

SIMNETS Analysis We extracted 1 second of spiking data from the first 30 repetitions of each stimulus (S = 360, spike trains), starting 0.28 seconds after stimulus onset. Only a small fraction of the total number of recorded trials was used in the analysis (25%) as we wanted to demonstrate SIMNETS ability to clusters neurons in datasets where only a small number of trials are available.

SIMNETS Cluster Characterization We used a circular-linear correlation (r_{cl}) analysis to assess SIMNETS' ability to group neurons according to their functional similarities. The correlation between each neuron's preferred orientation and its location in the low dimensional map y_i was calculated using:

$$r_{\theta,y} = \frac{cov(\theta_i, y_i)}{\sigma_A \sigma_B}, \tag{10}$$

where σ_A and σ_B are the standard deviation of the neurons' preferred orientations and y represents the neurons' locations in the map. A high correlation value indicates a strong relationship between a neuron's preferred orientation/direction and map location and demonstrates that functionally similar neurons were mapped to nearby regions of the map. We then characterized the functional properties of the ensembles identified using SIMNETS by calculating ensemble tuning functions (ETFs). ETFs were calculated by normalizing and averaging the joint firing rates across all neurons for each ensemble. A bootstrap resampling method was used to test for significant peaks in the ensemble tuning function (i.e., preferred ensemble tuning). A null-distribution was iteratively generated by randomly sampling a subset of neurons, equal in size to the number of neurons in the i-th ensemble, for which a new ensemble tuning function was calculated over each iteration (10000 iterations). The null distribution describes the probability of getting the observed peak response if the detected ensembles were chosen at random. A response that falls above (or below) the 99% confidence interval is considered significant.

Primate Primary Motor Cortex

Task Description

SIMNETS was applied to previously described dataset of 103 *Macaca mulatta* primary motor (M1) cortex neurons (i.e., single-units) recorded during a planar 8-direction reaching task [12,13]The single-unit activity was simultaneously recorded from the upper limb area of primary motor cortex using a chronically implanted microelectrode array. The monkey was operantly trained to move a cursor that matched its hand location to targets projected onto a horizontal reflective surface. A visual cue was used to signal movement direction during a variable duration instructed delay period (1 – 1.6 s) to one of eight radially distributed targets on the screen with the associated reach angles of φ = {0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°}. At the end of the instructed delay period, the central target was extinguished, instructing the monkey to reach towards the previously cued target.

SIMNETS Analysis

We analyzed 1 second of neural data from correct trials (S = 114, trials), starting 0.1 second before movement onset. Characterization of the detected SIMNETS clusters is similar to that described in section 5.1.

We characterized the preferred direction of each M1 neuron by fitting a von Misses distribution [14] to the firing rate function R:

$$r(\varphi) = \beta + h.exp\big(h.cos(\varphi - \mu)\big) \tag{11}$$

where β is the offset of the function, h is the depth of the tuning, is the reach angle and μ is preferred reach direction of the cell. The function takes on a maximum value at $\varphi = \mu$, which corresponds to the neuron's preferred reach angle.

Rat Hippocampal CA1

Task Description

We applied SIMNETS to a previously described dataset of rat hippocampal neurons[15,16] made publicly available by the Collaborative Research in Computational Neuroscience (CRCNS) data-sharing repository [17]. The neurons were simultaneously recorded from the CA1 hippocampal region using multi-site silicon probes while the rat performed a spatial navigation task in a maze. Briefly, the rat was trained to run through the arms of a 'figure-8' maze in a left/right alternating manner in order to receive a reward. The left/right track runs were interleaved with a wheel-run period that functionally served as a memory delay-period. The rat performed T = 17 correct trials (T_r = 8, left trials; T_l = 9, right trials), taking on average 4.3 seconds to reach the rewards located at either end of the arms. The rat's path along each arm of the track was linearized and divided into small (50 cm) or large (325 cm) spatial bins for the spatial firing field analysis or SIMNETS analysis, respectively (see next section for more details).

Single Neuron Functional Characterization

The rat's path along each arm of the track was linearized and divided into 50 mm spatial bins when generating the spatial firing field maps. Bins corresponding to reward locations and the inter-trial activity were excluded from the analysis, leaving a total of 39 bins for each of the left and right trajectories, where the first 19 spatial bins were common to both

trajectories. We generated a separate spatial firing map for the left and right trajectories of each neuron by dividing the number of spikes in the *i*-th bin by the rat's occupancy time t_i, and used a Gaussian kernel (width = 3 bins/150 mm) to smooth across the firing rates in each bin. Neurons that did not exhibit a 5 Hz firing rate in at least 1 spatial bin were not included in the analysis, leaving a total of $N = 80$ neurons. Neurons were characterized as non-place cells ($n = 22$, nPC) and place cells ($n = 58$, PC) based on their firing field properties and an information-theoretic measure of the spatial information in their spikes [15,18,19]. Neurons were classified as having place cell-like activity if the mean firing rate in three contiguous bins exceeded the mean of all other firing fields by 20% [18,20] (using 3.5 STD of the out-of-field firing rate produced similar results [15]) and if their information content exceeded 0.5 bits/spike [19] on either the left or right trajectories. The spatial information metric, I_{spike}, is a measure of the extent to which a neuron's spiking activity can be used to predict the rat's position along the track. The spatial information content of the neuron (measured in bits/spike) is defined as:

$$I_{spike} = \sum_{i=1}^{l} P_i \frac{v_i}{V} log_2 \frac{v_i}{V}, \tag{12}$$

where P_i is the occupancy probability, v_i is the firing rate in the *i*-th bin, and V is the overall mean firing rate of the cell across all bins in trajectory.

SIMNETS Analysis

We divided the T linearized trajectories into six 325 cm spatial bins and extracted 0.75 s spike trains beginning at the time that rat entered a given bin. The time window duration was selected to capture the smaller receptive fields ~ 0.6 s but still include a large portion of the average place field width (1 s – see Pastalkova et al. 2008). The spatial bin size corresponds to the approximate distance travelled in this time window. This resulted in $S = 108$ spike train events. The SIMNETS algorithm was applied to the resulting N x S spike train

SIMNETS Cluster Characterization

We compared the distances between neurons characterized as non-PCs and PCs in order to demonstrate the ability of SIMNETS to cluster the non-PCs to a specific region of the NS map. The Euclidean distances were calculated between all pairs of non-PCs (referred to as 'Within' pairs) and between pairs of non-PCs and PCs ('Between' pairs) in the low dimensional NS map. A rank-sum statistical test was carried out on the Within and Between distance distributions using a significance threshold of $p = .001$. *Ensemble firing rate maps* were generated for each of the three example SIMNETS clusters by averaging across the normalized single neuron firing rate spatial maps for all neurons in the *i*-th cluster. A bootstrap resampling method was used to test for significant peaks ($p = .01$) in the ensemble spatial firing map (i.e., place-field cell-assemblies). The procedure involved randomly sampling a subset of neurons from the neuron population, where each subset was equal in size to the number of neurons in the *i*-th cluster. A new ensemble tuning function was calculated over repeated iterations of this procedure (10000 iterations). The resulting null distribution describes the probability of getting the observed peak response if the detected neuron clusters were selected at random. A response that falls above (or below) the 99% confidence interval is considered significant.

Ensemble Activity Similarity Maps are low-dimensional neural activity state-space maps that capture the relationships between neural ensemble activity patterns on individual trials. This method of visualizing low-dimensional projections of ensemble activity has previously been described by this group [13,21] Generating the low-dimensional Ensemble Activity Similarity maps consists of three steps. 1) Calculate the pairwise spike train distances between the *i*-th spike train event of neuron *j* and all other spike trains belonging to that neuron $S_j = \{S_{1,i}, S_{2,j}, \ldots, S_{m,j}\}$. This results in a vector of pairwise spike-train distances $d(S_{i,j})$ of length m. 2) the pair-wise similarity vectors for the *i*-th trial of all *n* neurons of a given ensemble are concatenated and combined into a matrix, resulting in an m x mn matrix ($D_{ensemble}$). The resulting m x mn pairwise distance matrix constitutes the relational embedding of the entire data set. 3) The final step consists of projecting the high-dimensional m x mn distance matrix down into a m x d matrix, where d is the desired dimension of the projection.

Table 2: Summary of datasets

Symbol	Synthetic Dataset	V1 Dataset	M1 Dataset	CA1 Dataset
N	180	112	103	80
T	30	360	114	18
S	30	360	114	119
*S*pike train duration (s)	1	1	1	0.75

Table 3: Summary of SIMNETS Inputs/Outputs for each dataset

Symbol	Synthetic Dataset	V1 Dataset	M1 Dataset	CA1 Dataset
q	[0, 10, 200]	20	10	50
perp	50	30	20	15
d	3	3	3	3
$\hat{k}$	4,3,3	3	3	6
k	2-15+	2-15+	2-15+	2-15+
$\hat{h}$	0.5, 0.99, 0.95	0.71	0.68	0.78

Bibliography

1. Victor, J. D. & Purpura, K. P. Metric-space analysis of spike trains: theory, algorithms and application. *Netw. Comput. Neural Syst.* (1997). doi:10.1088/0954-898X_8_2_003
2. Victor, J. D. & Purpura, K. P. Nature and Precision of Temporal Coding in Visual Cortex: A Metric-Space Analysis. *J. Neurophysiol. Pt-intd U.S.A* **76,** (1996).
3. Van Der Maaten, L. & Hinton, G. *Visualizing Data using t-SNE*. *Journal of Machine Learning Research* **9,** (2008).
4. Lloyd, S. P. *Least Squares Quantization in PCM*. *IEEE TRANSACTIONS ON INFORMATION THEORY* **28,** (1982).
5. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20,** 53–65 (1987).
6. Author, S., Kullback, S. & Leibler, R. A. *On Information*. *Source: The Annals of Mathematical Statistics* **22,** (1951).
7. Mantel, N. The detection of disease clustering and a generalized regression approach. *Cancer Res.* **27,** 209–20 (1967).
8. Smith, M. A. & Kohn, A. Spatial and temporal scales of neuronal correlation in primary visual cortex. *J. Neurosci.* **28,** 12591–603 (2008).
9. Kohn, A., Smith, M. A. (2016). Utah array extracellular recordings of spontaneous and visually evoked activity from anesthetized macaque primary visual cortex (V1). *CRCNS.org* (2016). Available at: http://dx.doi.org/10.6080/K0NC5Z4X.
10. Gur, M. & Snodderly, D. M. Direction selectivity in V1 of alert monkeys: evidence for parallel pathways for motion processing. *J. Physiol.* **585,** 383–400 (2007).
11. Hawken, M. J., Parker, A. J. & Lund, J. S. Laminar organization and contrast sensitivity of direction-selective cells in the striate cortex of the Old World monkey. *J. Neurosci.* **8,** 3541–8 (1988).
12. Rao, N. G. & Donoghue, J. P. Cue to action processing in motor cortex populations. *J. Neurophysiol.* **111,** 441–53 (2014).
13. Vargas-Irwin, C. E., Brandman, D. M., Zimmermann, J. B., Donoghue, J. P. & Black, M. J. Spike Train SIMilarity Space (SSIMS): A Framework for Single Neuron and Ensemble Data Analysis. *Neural Comput.* **27,** 1–31 (2015).
14. Amirikian, B., Georgopoulos, A. P. & Georgopulos, A. P. Directional tuning profiles of motor cortical cells. *Neurosci. Res.* **36,** 73–9 (2000).
15. Pastalkova, E., Itskov, V., Amarasingham, A. & Buzsáki, G. Internally generated cell assembly sequences in the rat hippocampus. *Science* **321,** 1322–7 (2008).
16. Mizuseki, K., Sirota, A., Pastalkova, E. & Buzsáki, G. Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop. *Neuron* **64,** 267–280 (2009).
17. Pastalkova E, Wang Y, Mizuseki K, B. G. Simultaneous extracellular recordings from left and right hippocampal areas CA1 and right entorhinal cortex from a rat performing a left / right alternation task and other behaviors. (2015). Available at: http://dx.doi.org/10.6080/K0KS6PHF.
18. Bjerknes, T. L., Dagslott, N. C., Moser, E. I. & Moser, M.-B. Path integration in place cells of developing rats. *Proc. Natl. Acad. Sci. U. S. A.* **115,** E1637–E1646

(2018).
19. Skaggs, W. E., Skaggs, W. E., McNaughton, B. L., Gothard, K. M. & Markus, E. J. An Information-Theoretic Approach to Deciphering the Hippocampal Code. *IN* **5,** 1030--1037 (1993).
20. Jeffery, K. J., Donnett, J. G., Burgess, N. & O'Keefe, J. M. Directional control of hippocampal place fields. *Exp. brain Res.* **117,** 131–42 (1997).
21. Vargas-Irwin, C. E., Franquemont, L., Black, M. J. & Donoghue, J. P. Linking Objects to Actions: Encoding of Target Object and Grasping Strategy in Primate Ventral Premotor Cortex. *J. Neurosci.* **35,** 10888–97 (2015).