Non-Negative Matrix Factorization for Analyzing State Dependent Neuronal Network Dynamics in Calcium Recordings

Daniel Carbonero[1,3,4], Jad Noueihed[1,3,4], Mark A. Kramer[2,3], John A. White[1,3,4*]

[1] Department of Biomedical Engineering, Boston University, Boston, Massachusetts, United States of America

[2] Department of Mathematics and Statistics, Boston University, Boston, Massachusetts, United States of America

[3] Center for Systems Neuroscience, Boston University, Boston, Massachusetts, United States of America

[4] Neurophotonics Center, Boston University, Boston, Massachusetts, United States of America

* Corresponding Author

E-mail: jwhite@bu.edu

Abstract

Calcium imaging allows recording from hundreds of neurons in vivo with the ability to resolve single cell activity. Evaluating and analyzing neuronal responses, while also considering all dimensions of the data set to make specific conclusions, is extremely difficult. Often, descriptive statistics are used to analyze these forms of data. These analyses, however, remove variance by averaging the responses of single neurons across recording sessions, or across combinations of neurons to create single quantitative metrics, losing the temporal dynamics of neuronal activity, and their responses relative to each other. Dimensionally Reduction (DR) methods serve as a good foundation for these analyses because they reduce the dimensions of the data into components, while still maintaining the variance. Non-negative Matrix Factorization (NMF) is an especially promising DR analysis method for calcium imaging because of its mathematical constraints, which include positivity and linearity. We adapt NMF for our analyses and compare its performance to alternative dimensionality reduction methods on both artificial and in vivo data. We find that NMF is well-suited for analyzing calcium imaging recordings, accurately capturing the underlying dynamics of the data, and outperforming alternative methods in common use.

Introduction

Recent advances in neural recording methods have made it possible to collect large, complex data sets that can be used to study context-dependent neural dynamics in great detail(1,2). For example, calcium imaging allows simultaneous recording from hundreds of neurons in a single brain region in vivo, while maintaining single cell-resolution(3,4). Other tools allow cell "tagging" based on factors, including activity during a specified time window and molecular genotype (5–8). The resultant data set consists of several hundred time series thousands of points long, multiplicatively increasing per experimental condition added, in addition to any tagging data. The process of characterizing such single-cells responses, relating them to those of other cells across

time, and accounting for the tagging information, all while considering changes in animal behavior, is a difficult but essential task.

Basic descriptive statistics (mean, median, variance) are a common approach to analyze large neural datasets. However, these approaches are limited because they average over the temporal dynamics of neuronal activity and can obscure interactions among neurons at fine time scales(9,10). Dimensionality reduction (DR) is a more sophisticated approach that can help to reveal low-dimensional dynamics by transforming high-dimensional data into interpretable, low-dimensional components(11). Compressing the data in this way can preserve temporal patterns and correlations and focus subsequent analysis on the dynamical patterns of interest.

DR approaches have many applications in the analysis of neural data, including analysis of electrophysiological recordings(9,12,13), automation of trace extraction of neuronal activity in fluorescent recordings(1,14–18), and as a pre-processing step to more complex behavioral decoding(19). To analyze network dynamics in calcium recordings, non-linear DR methods have emerged to map dynamics to low dimensional manifolds(20–23). While excellent for very low dimensional visualization(24,25), the dynamics are difficult to interpret beyond the manifold being modeled, with manifolds being difficult to interpret beyond three dimensions. Linear DR methods have likewise been employed to either statically cluster cells(18,26–28), or to abstractly analyze activity in low-dimensional subspaces(29–32). As a result, for our data, we argue that linear methods will provide a more interpretable low dimensional representation of the activity, using a few more dimensions to describe the majority of the dynamics, at the cost of a difficult to interpret, low dimensional manifold.

Non-negative Matrix Factorization (NMF) is an especially promising linear decomposition method for analysis of activity recorded in calcium imaging. NMF operates under the constraint that every element in the data, and the decomposition, be non-negative(33), as is naturally true for neuronal calcium traces(18). Further, the positivity constraint, coupled with the linearity of the reduced

dimensions, ensures that each of the dimensions sum to form a low dimensional representation of the data, simplifying the interpretation of the data(34). In neuroscience, NMF has been applied to quantify single cell pairwise relationships (35) and statically cluster cells under a series of strict assumptions(27). Here, we develop an application of NMF to analyze holistic network activity, capturing how precise sub-networks of neuronal dynamics evolve in different experimental contexts. To assess performance of the NMF approach, we generate artificial calcium imaging data from ground-truth networks and assess how well our approach captures the underlying network structures responsible for the emergent dynamics of the network compared to alternative widely used approaches. We then apply the same NMF framework to characterize the dynamics of a representative *in vivo* calcium imaging data set. We conclude that, compared to alternative methods, NMF best captures the ground truth networks that underlie the observed calcium dynamics and provides a useful low-dimensional description of complex physiological data.

Dimensionality Reduction (DR) Methods:

Non-Negative Matrix Factorization (NMF):

NMF is a linear, matrix-decomposition method requiring all elements be non-negative(33). Each reduced dimension, or component, identified in NMF can be interpreted as a specific combination of input features representing distinct sources of variance in the data(34). The components then sum to represent the original data, rather than canceling variables (using negativity) to achieve a geometrically strict representation(36). Mathematically, NMF decomposes an original matrix X into two constituent, lower rank, matrices W and H, given by:

$$X = WH$$

(Eq. 1)

with the aim of iteratively updating W and H to minimize an objective function so that the product of the two deconstructed matrices optimally reconstructs the original(33). Here, X is a column-wise representation of the data (e.g., each column is the time-series fluorescent activity of a neuron), that is decomposed into two lower rank matrices, W and H. For NMF, W can be interpreted as a feature matrix, each vector representing a distinct pattern in the input data, and H interpreted as a coefficient matrix, representing the contribution of each feature vector to each point in the original matrix. When using DR for analyses, especially in the context of neural recordings, the underlying assumption is that the original matrix can be decomposed into a lower rank series of matrices. The rank, k, should be significantly smaller than n (k << n), the number of neurons recorded, while still recapturing a majority of the variance present in the data(37). Model performance can be assessed by the variance explained as represented by the coefficient of determination (R^2)(38), given by:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}},$$

(Eq.2)

where,

$$SS_{res} = \sum_{i,j} (X_{i,j} - (WH)_{i,j})^2 ,$$

$$SS_{tot} = \sum_{i,j} (X_{i,j} - \overline{X})^2 .$$

(Eq. 3,4)

Thus, the model captures the original data perfectly in the limit as R^2 approaches one. R^2 increases as components are added, because the rank of the two constituent matrices are increasing. Akaike Information Criterion (AIC) has been adapted for NMF to inform the rank to

which the dimensionality should be reduced(34,39), with the optimal model minimizing it(40); for our implementation of NMF:

$$AIC = 2(\frac{R^2}{\sigma^2} + k(n + t))$$

(Eq. 6)

Where R^2 is the reconstruction error defined in Eq. 2, σ^2 is the estimated variance of the data set, k is the number of components the model is decomposing into, n is the number of neurons in the recording, and t is the number of time points in each of the time series being decomposed. For our analysis, we interpret each row of H as a sub-network, with each entry of that row as a neuron's contribution of to that sub-network. Each column of W is then interpreted as the time-series for that sub-network. We provide the pseudocode from an implementation of the NMF analysis pipeline:

```
Input: X [time points x neurons]
Output: NMF_model
normalize data between (0,1)*
initialize list of AICs
for i in range(1, num_neurons)
    W, H = init_WH(n_components = i)
    NMF_model = optimize(X, W, H)
    AIC_list.append(NMF_model.aic)
    if AIC[i-1] < AIC[i]:
        n_components = i-1
        break
W,H = init_WH(n_components = n_components)
NMF_model = optimize(X,W,H)
```

* we normalize using the global maximum and minimum of the trace set to preserve the relative difference between them

The above pipeline describes the sequence of using AIC to find an optimal number of components, and then fit a model to that number of components. Our implementation of NMF is a modification of the implementation found in Python's Scikit-Learn package(41). We manually initialize W and H, as opposed to the automated method found in the package, due to the wide variety of initialization methods possible for NMF(42,43), and to maintain consistent, precise, and easily reproduceable control over the initial conditions of our models. For the analyses found in this paper, we apply non-negative dual singular value decomposition (nndsvd), a consistent and efficient initialization method(44).

Principal Component Analysis (PCA):

We implement PCA because of its widespread use in neural data analysis(12,18,29–31,45–47). PCA, efficiently calculated by a Singular Value Decomposition (SVD)(36):

$$X = U\Sigma V^T$$

(Eq. 7)

where X is a column-wise representation of our data (i.e., each column is the time-series fluorescent activity of a neuron, assumed to have zero mean) decomposed into U, Σ, and V^T. V^T is the set of basis or singular vectors, also referred to as the principal components (PCs) of the original matrix, where each row of this matrix is a component of the data. Further, each of the PCs are necessarily orthogonal to each other by construction. U is the projection of the original matrix X onto the principal components in V^T, and Σ is a matrix containing the singular values of X describing the magnitude of the transformation, ordered by decreasing singular values(36). The variance explained by a PC is the proportion of its singular value compared to the sum of all

singular values. For our analysis, we interpret each PC as a sub-network or prevalent pattern of activity, with each entry of the PC being a neuron's weight or contribution to that PC. We then interpret the projection of the data onto each PC as the activity or time series of that sub-network, attributing the dominance of that sub-network to the variance explained. Our implementation of PCA is a modification of the implementation found in Python's Scikit-Learn package(41). We provide the pseudocode from an implementation of the PCA pipeline below:

```
Input: X [time points x neurons]
Output: PCA_model
1:  normalize data between (0,1)*
2:  subtract mean of each time series
3:  PCA_model = PCA(X) #data must be in column-format at this step.
* we normalize using the global maximum and minimum of the trace set to preserve the relative difference between them
```

This is a modification of the implementation of PCA found in sklearn(41). PCA calls for every variable, or trace in our case, to be centered about zero(36). However, we remove the uniform normalization from sklearn's implementation to allow for finer tuning of data normalization before decomposition with the model.

Independent Component Analysis (ICA):

We implement ICA because of its ability to extract independent sources from complex signals, which could help separate overlapping patterns of neural activity in a single calcium recording. While ICA has been used for automated trace extraction in calcium recordings(48), this approach has seen limited application in the further analyses of neural signals estimated calcium recordings. ICA is a statistical technique that aims to decompose a multivariate signal into statistically independent components(49), rather than a set of orthogonal components, as

found in PCA. ICA operates under the assumption that k independent sources produce the data, and that the data are a linear mixture of these underlying sources. ICA assumes the data to be a linear mixture of sources:

$$x = As$$

(Eq. 8)

where x is the original data, s are the underlying independent sources, and A is a mixing matrix that mixes the components of the sources(49).

In practice, the independent sources found in s, are decomposed using an unmixing matrix, W, such that the linear transformation of the data by W, estimates the underlying independent sources(49):

$$\hat{s} = Wx$$

(Eq. 9)

Given that W transforms x to estimate the independent sources, we interpret the components of W as the contribution of each neuron to the sub-network activities, which are the independent sources in s. Here, we interpret the s independent sources as the underlying patterns of activity of each sub-network that drive the observed dynamics. Like PCA, ICA assumes that inputs are zero mean. While PCA is designed to maximize the variance of the data along the principal components(36), ICA attempts to separate the data into k independent sources. We use the fastICA implementation found in sklearn(41), and provide the pseudocode from an implementation of the ICA analysis pipeline below:

Input: X [time points x neurons]

Output: ICA_model (contains A, s, amongst other information)

1: normalize data between (0,1)*

2: subtract mean of each time series

3: ICA_model = ICA(X)

* we normalize using the global maximum and minimum of the trace set to preserve the relative difference between them

While ICA calls for unit variance for each of the features, which in this case would be the calcium time series, we maintain the relative difference in variance between each neuronal time series to maintain the relative difference in activity recorded as a function of the fluorescence.

Uniform Manifold-Approximation Projection (UMAP):

We implement UMAP because of its increasing use to analyze life science data(50,51) and its ability to capture non-linear relationships. UMAP has been shown to capture complex patterns to visualize and cluster data in very low dimensions(52), though the technique remains untested on calcium recordings. Briefly, UMAP aims to optimize between preserving local and global structures, where structure is local between neighboring points and global between data points extending beyond the neighborhood. In neuronal calcium imaging, "neighborhood" refers to local relationships among neurons with similar activity patterns at a time point, while global relationships extend beyond those neighbors at the same point. UMAP subsequently directly maps this structure to a lower dimension(52), resulting in the output being a single matrix (the equivalent of H, V^T, and S), and not a product of matrices. Therefore, UMAP does not provide temporal information about the activity of the lower dimensional patterns or sub-networks. Further, there is no well-established way to quantify variance explained in UMAP as in the other linear methods. We use the python package UMAP-learn to implement UMAP, and provide the pseudocode from an implementation of the UMAP analysis pipeline below:

Input: X [neurons x time points] (NOTE TRANSPOSITION FOR THIS CASE, COMPARATIVELY)

Output: UMAP_model (contains low_d_projection, amongst other information)

1: normalize data between (0,1)*

2: UMAP_model = UMAP(X)

* we normalize using the global maximum and minimum of the trace set to preserve the relative difference between them

In addition to the number of components, k, being a manually set hyperparameter, the number of points in the neighborhood is also manually set, in addition to the other 37 tunable parameters in the package(52). Because our main objective is to identify how neurons group to form sub-networks and shift as a function of experimental contexts, we apply UMAP to decompose the data X into a *k x n* matrix. k is the number of components or subnetworks we determine to decompose into, while n is the number of neurons, aiming to give a similar representation to H, V^T, and S in our other methods.

Model	**Variance Explained**	**Information Criterion**	**Neuronal Weights**	**Decomposed Activity**
NMF	✓	✓	✓	✓
UMAP	X	X	✓	X
PCA	✓	X	✓	✓
ICA	X	X	✓	✓

Table 1: Model Attributes

Network Simulation Methods:

To assess performance of the DR methods, we implement two simulation paradigms. In each paradigm, we simulate the spiking activity in a network of interconnected neurons and estimate calcium imaging traces for each neuron. We then apply each DR approach to the simulated calcium fluorescence data. We describe each simulation paradigm below.

Perfectly Intraconnected, Independent Nodes:

We first consider a network of 100 neurons organized into 5 independent nodes, each consisting of 20 neurons. Neurons within the same node are perfectly connected, such that a spike by any neuron in one node produces a spike in all other neurons in that node. Neurons in different nodes are disconnected, such that activity in two neurons of different nodes are independent (Fig. 2A). The activity of a single neuron is governed by a basic Poisson spike generator(53), given by:

$$spike[t] = x_{rand} < r * dt$$

(Eq. 10)

where a neuron spikes at time t if x_{rand} (uniform [0, 1]) is less than the product of the set firing rate, r, and the time step, dt. We update this generator to include additional network effects, and a refractory period. We represent the connectivity between neurons as an adjacency matrix A. Within any node of 20 neurons, A is 1, while between nodes A is 0; we exclude self-connections by setting the diagonal of A to 0. We also include a refractory period of duration q time steps, such that after a spike a neuron is temporarily unable to generate a subsequent spike.

We simulate the spiking activity of a single neuron in this network n = {1,2, … N} as:

$$spike[t]_n = x_{rand} < \left(\frac{rk}{N}dt + A_{n\,x\,N} * spikes[t-1]_{N\,x\,1}{}^{T}\right) * RS[t]_n$$

(Eq. 11)

where r = 3 is the set nominal rate, N = 100 is the total number of neurons, k = 5 is the number of nodes, and dt is the time step. A is the adjacency matrix of dimensions N x N, where each row represents the incoming connections to a neuron, and each column represents the outgoing connections of a neuron, and $RS[t]_n = \{0,1\}$ is the refractory state of neuron n at time t. A

neuron spikes at time t if the randomly generated number x_{rand} (uniform [0,1]) is less than the product in Eq. 11. We note that any multiplication of an adjacency of 1 to neuron n, in conjunction with a spike at time t-1 will result in a spike at time t in neuron n, given the neuron n is not in a refractory state. After simulating the network we convolve each spike train with a calcium kernel(27,54) as described below. The result is a series of 100 fluorescent neuronal activities (Fig. 2B), of which there are five groups of twenty that are perfectly correlated (Fig. 2C).

Random Process Propagation:

We generate a second simulation paradigm to more closely represent a physiologically relevant dynamic in calcium fluorescence data: a small number of independent random spiking patterns simultaneously driving calcium events in all neurons. To do so, we first simulate k = 5 independent spiking patterns as,

$$process\ spike[t]_n = x_{rand} < r\ dt\ .$$

(Eq. 12)

We then consider a population of N = 150 neurons that receive inputs from each of the underlying spiking patters (Fig. 3A) weighted as follow: 20% of the neurons have weights uniformly distributed between 0.2 and 1.0, and 80% of the neurons have weights exponentially distributed (rate parameter $\lambda = 8.0472$) with maximum value of 0.2 (Figure 3B). In this way, most neurons are connected with weights following an exponential distribution(55), and the few neurons with strong weights drive a majority of the activity. The resulting adjacency matrix A between the k spiking patterns and N neurons has dimensions k x N, where each element describes the weight of the kth underlying process input to neuron n = {1, 2, … N}. We simulate each neuron's spike train as:

$$neuron\ spike[t]_n = x_{rand} < (A_{n\,x\,k} * process\ spike[t-1]_{k\,x\,1}) * RS_n$$

(Eq. 13)

where n indicates the nth neuron, k indicates the kth underlying spike processes, and we include a refractory period for each neuron. Doing so results in N = 150 neuronal spike trains. We convolve the spike train of each neuron with the calcium kernel(27,54), resulting in simulated calcium data similar to our in vivo recordings, mirroring: activity rate, activity distribution, underlying macroscopic drivers, and calcium neuronal time series characteristics.

Calcium Kernel:

We convolve the simulated spike trains with the following calcium kernel to represent the fluorescence recorded from the genetically encoded calcium (GECI) kernel jGCaMP7f:

$$[Ca^{+2}]_t - [Ca^{+2}]_{t-1} = -\frac{\mathrm{dt}}{\tau}([Ca^{+2}]_{t-1} - [Ca^{+2}]_b) + A * spike[t] + \sigma_c\sqrt{dt}\,\epsilon_t$$

(Eq. 14)

where $Ca^{+2}_{t,t-1,b}$ are the calcium concentrations at t, t-1, and the baseline, respectively. Here, dt is the time step, τ is the time constant of the GECI, σ_c is the variance of the calcium noise, and ϵ_t is a random variable following the standard gaussian distribution. We then calculate the fluorescence as:

$$F_t = \alpha([Ca^{+2}]_t) + \beta + \sigma_F\,\epsilon_t\ ,$$

(Eq. 15)

where F_t is the fluorescence at time t, α is the intensity, β is the bias, σ_F is the variance of the calcium noise, and ϵ_t is a random variable following the standard gaussian distribution. We set τ

to 0.265 and σ_c to 0.5 to match the kinetics of jGCaMP7f(56). We set the calcium baseline to 0.1, A and α to 5, β to 10, and σ_F to 1(27).

Component Assignment Success

Perfectly Connected, Intraconnected Nodes:

To assess whether nodes were successfully assigned to components, we develop a framework for each of our generated network architectures. For the nodal model, we begin by summing the weights for each node, in each of the decomposed components, described by the pseudocode below:

```
Input: H (matrix of neuronal weights), nodes (array detailing which neurons belong to each node)
Output: aggregated_weight_matrix
aggregated_weight_matrix = zeros(num_nodes, num_components)
for i in range(num_nodes)
        for j in range(num_compnents)
                aggregated_weight_matrix [ i, j ] = sum(H [nodes[ i ] , j )
```

The result is a five by five matrix, where the element [i , j] is the sum of the weights for neurons in the ith node for the jth component. To then determine success of prediction, we establish two conditions that must be met. First, within a particular component, the sum of weights for the node with the highest value should exceed the sums of weights for all other nodes within the same component. Second, this highest sum should also be the greatest for the node, across all components. By applying these conditions, we can assess the assignment for each node individually. In essence, once we obtain the aggregated weight matrix, an ideal model would exhibit the same element as the maximum value for each row and column.

Random Process Propagation:

We calculate a 5 x 5 correlation matrix between the weights in each of the 5 components and the connection probabilities from each of the 5 generated random processes sampled from the defined exponential distribution. To assess assignment, we again consider two conditions. First, for a chosen component, the highest correlation with the connection probabilities from one random process should surpass the correlation to the other random processes in the same component. Second, this highest correlation within each component should identify a different random process; i.e., each component should identify a unique random process. We consider a component successfully assigned to one of the five random processes if its highest correlation is more correlated with that process than the other four processes.

Awake-State Data:

To investigate the application of each DR technique to *in vivo* recordings, we also consider calcium imaging data recorded from 36 awake mice. Briefly, recordings in layer 2/3 of murine S1(57) were collected during the awake, resting state(8). Traces representing calcium transients were extracted from the recordings and processed using standard techniques (58).

Results

We test NMF against three existing DR approaches in common use: UMAP, PCA, and ICA. We do so using two different simulated neuronal network architectures to assess the relative performance of each DR approach, when the network structure is known. We first build a simple network with five groups of independent neurons, each with 20 neurons, in which the neurons in each independent node are perfectly connected (Fig. 2). In this architecture, a spike in one neuron in a node will drive all other neurons in that node to spike. We use such an architecture to test if, and how well, models can identify the five nodes given traces of simulated activity.

To further evaluate the performance between DR methods, we examine a more intricate network architecture aligned with our spontaneous calcium recordings. In doing so, we create five distinct

random processes driving 150 neurons at varying degrees of strength (Fig. 3). This approach better reflects the dynamics we aim to capture, encompassing underlying activity patterns in diverse contexts driven by weighted sub-networks of neurons. Utilizing this surrogate architecture, our objective is to assess how effectively the models can extract both the underlying activity patterns, and the corresponding contributions by individual neurons, given a clear end target for performance assessment. We present two different analyses for both simulated network architectures. We both perform a qualitative analysis for a single generated network, and a more scaled quantitative approach for a series of 256 random instantiations and simulations of the networks. We find that NMF best captures the dynamics of the simulated networks, with components summing to provide an accurate macroscopic low dimensional representation of the data. We argue the parts-based representation provided inherently by NMF supports the improved performance for these data and the goal of interpretably extracting low dimensional neuronal network dynamics.

Perfectly Intraconnected, Independent Nodes:

We generate the nodal networks and fit each of the DR methods to the simulated calcium traces. Initially, we evaluate the average variance explained and the AIC across 256 networks to gauge how well the models capture the original data at different lower dimensionality levels. Both NMF and PCA exhibit similar variance explanation patterns across all components, with the average variance explained stabilizing at five components (Fig. 4A, left axis), as required to capture the dynamics of simulated activity in five nodes. This stabilization is influenced by the introduction of unique noise increments in each trace (Eq. 14 & Eq. 15). Additionally, the AIC for the NMF method consistently minimizes at five components across all 256 model instantiations, effectively optimizing for the correct number of nodes (Fig. 4A, right axis).

To evaluate how the different approaches capture the independent nodes of activity as components, we devised a discrete metric to determine success or failure of individual node

assignment to components, detailed in the methods, for each architecture. We show the assignment success for a single model for each of the detailed DR methods (Fig. 4B). We find the best performance with the NMF method, which successfully maps each node to a component, followed by PCA, ICA, and UMAP. We repeat this process for 256 replicate models (Fig. 4E) and find that NMF always maps each node to a separate component, followed by PCA which performs with a mean accuracy of 0.8578, and then ICA and UMAP which perform with mean accuracies 0.6445 and 0.6242, respectively, and typically fail to successfully map the entire model.

We then visualize how each of the DR methods map the nodal patterns of activity to each of the components. We do so by showing the weight for each neuron, for each component, for each method, delimited by node (Fig. 4C), for a single instantiation of the simulated neural activity. In this example, we find a very clear separation of weights from each node for the NMF method; each component assigns weights to the neurons within a node. We argue this separation is a result of the parts-based representation that NMF inherently embodies(34). In a parts-based representation, each component contributes an additive combination of components to faithfully represent the original data. This means that NMF decomposes the overall activity into identifiable parts, enabling a clear separation of weights for each node. Conversely, the PCA and ICA methods produce distributions of node weights with less clear separations within a component. While the first 5 components of the PCA method each identify neurons in each node, the constraints of the method result in counterbalancing variables that obfuscate the underlying input resulting in the data, hindering performance(36). UMAP poorly separates nodes into five components because a strength of UMAP, and manifold learning in general (e.g., TSNE), is visualization in very low dimensions (restricted to two or three dimensions)(24,25).

To provide a quantitative assessment of model performance, we calculate the average displacement between neuronal weights assigned to a component and all other weights within the same component, for all components of each of the 256 instantiations of the model (Fig. 4D).

We do so to provide a quantitative and continuous measurement of model performance (separation between weights in node and not in node, at assigned component) beyond binary success/failure, with the magnitude indicating the degree of separation between the neurons in the assigned node and all other neurons within a given component. Normalized per model, a positive value describes an assigned node whose average weight was higher than the weights of the other node in that component. A negative value would then describe the opposite, with the magnitude of the displacement describing the degree of difference between the weights. Our findings reveal NMF best separates nodes into components, followed by PCA, UMAP, and ICA. While this supports our finding that NMF specifically assigns patterns of nodal activity to components, the other methods do tend to separate a majority of the nodes of neurons individually in each component (the normalized distance tends to exceed 0), with the differing representations being attributable to the different mathematical constraining during fitting (UMAP – very low dimensional manifold, PCA – orthogonality, ICA – independence). However, as we show in our next case, this structure breaks down on the introduction of more complicated data.

Random Process Propagation:

We simulate a network with a more complicated architecture to further evaluate model performance. This architecture involves generating five underlying random processes to drive the network activity. Each underlying random process provides input to each of 150 neurons, with a weight randomly sampled from a generated exponential distribution (Fig. 3). By analyzing these artificial networks, we aim to probe how well NMF maps the random process by observations from the received neuronal inputs to components. We similarly aim to compare the performance and low dimensional representation of NMF and the other methods we implement.

Analyzing the average variance explained for all 256 uniquely seeded networks of this architecture, we find that NMF explains significantly more at all components (Fig. 5A, left axis). We argue this is a function of the difference between optimizations. Rather than having a strict

orthogonality constraint between dimensions as in PCA, NMF aims to reconstruct the data with as little error as possible into the specified number of dimensions, allowing for explaining more variance with fewer components. The AIC shows a minimum at five components, the number of random processes used to generate the network (Fig. 5A, right axis).

We quantify assignment success of this model differently from the nodal architecture, due to the more continuous nature of the data, detailed in the methods. For or an example model instantiation (Fig. 5B), only the NMF method successfully assigns each process to one component. Assessing the performance over 256 realizations of the network (Fig 5F), we find that NMF successfully assigns each random process to a component. Consistent with the results for the nodal architecture, the other methods (PCA, ICA, and UMAP) have reduced performance. We conclude that NMF reconstructs the latent network structure from the observed data, capturing the five underlying random processes in a parts-based representation.

Beyond discrete prediction, we consider the correlation between the assigned connection probabilities from the latent processes to the observed neurons, and the neuronal weights of the predicted component. To start, we visualize the results for one instance of the NMF model, relating the neuronal weights of each component to the assigned outgoing connection probability from the assigned underlying random process to each neuron (Fig. 5C). We find that large neuronal weights in each component tend to occur for large connection probabilities for the predicted random process with a high correlation of 0.84 (Fig. 5D). Repeating this analysis for 256 realizations of the model, and regardless of prediction success, we find that NMF consistently infers components with high correlations between the connection probabilities and weights (Fig. 5E). The other methods have reduced performance, consistent with the results for the nodal architecture with PCA, ICA, and UMAP following in performance, respectively.

We finally analyze how the activity of each component correlates with the activity of the underlying random process. Qualitatively, for a single realization of the model, the component activity

captures the underlying processes very well (Fig. 5G) with correlations between the activities exceeding 0.85. Repeating this analysis across 256 realizations of the network, we once again find that NMF best captures the activity of the latent input processes, followed by PCA and ICA (Fig 5H); we note that UMAP does not estimate the times series of the latent processes.

We propose that the positivity and parts-based representation of the NMF method enable accurate reconstruct of the high-dimensional activity through low dimensional inputs. We propose that the constraints implemented by the PCA and ICA methods may obfuscate inputs to the observed neural network in its low dimensional representations. Finally, we interpret the poor performance of the UMAP method as indicating this method is better adapted for low dimensional visualization of less variant data. We conclude that NMF accurately reconstructs the latent inputs to a biophysically-motivated neuronal network that simulates calcium fluorescence recordings, despite multiple barriers to accurate identification (e.g., the underlying process is unobserved, with random connectivity to the observed neurons whose calcium signal is obfuscated with two types of noise). These results demonstrate NMF outperforms existing methods in common use to extract the underlying dynamics present in a series of neuronal activities recorded in calcium imaging.

Awake-State Data:

As a final proof of concept, we analyze application of the DR methods to awake state calcium imaging recordings from 36 mice. In doing so we apply the same analytical framework as for the simulated data. Analysis of the variance explained for each model (Fig. 6A) shows similar performance between the NMF and PCA methods, with a weak trend towards increased values for the NMF method. This small advantage in increased variance explained may be due to the NMF method acting to capture the original data as quickly as possible into a set number of components (Eq. 2), rather than using a complex geometrical constraint to analyze the data(49). We further find that NMF, using AIC, identifies an optimal number of components to describe the

data in each of the recordings (Fig. 6B), with number of neurons helping inform the number of components at a correlation of 0.662. We qualitatively characterize an NMF model fit to a single recording, using 18 components as determined by AIC. When characterizing the neuronal weights estimated by the NMF method, we find behavior consistent with the simulated random process network (Fig. 6C); the NMF method results in in the neuronal weights with an approximate exponential distribution (Fig. 6D), with most neurons contributing little to the overall network activity, and few neurons very significantly contributing. We further briefly characterize the correlations between activity decomposed for each component (Fig. 6F). We find that correlations are consistently low between the activity of the components. Given that the correlations are low, and we capture a significant portion of the variance, we interpret these results to indicate that the NMF, and the parts-based nature of the model, successfully extracts unique predominant patterns of activity in the data. We conclude that, applied to these *in vivo* calcium imaging recordings, that the NMF method identifies components with distinct (uncoupled) dynamics. While promising, future work is required to relate these inferred components and dynamics to biological mechanisms and behavior. We conclude that the NMF method is a promising tool to analyze neuronal network dynamics and identify meaningly sub-network activity via a relatively simple and interpretable approach.

Discussion

We have developed an analytical pipeline to more thoroughly model neuronal network dynamics with NMF, considering all information provided by the model in its low dimensional representation. We generate a series of calcium traces recorded from ground-truth artificial neuronal networks at two degrees of complexity to assess how well our analyses extract the ground-truth responsible for the simulated dynamics, and compare the performance of NMF to UMAP, PCA, and ICA. We find the NMF significantly performs the best, followed by PCA, ICA, and UMAP, respectively. We

then apply our NMF pipeline to a series of in vivo baseline recordings and find that NMF confers a similar, and easily interpretable representation to the underlying random process network.

We argue this performance is a result of the “parts-based representation” conferred by NMF(34). Because every single element of the model is necessarily positive, and the decomposition is aiming to recapture the original data, every component sums to give a low dimensional macroscopic representation of the original data. Interpreting the decomposition as the sum of its parts, rather than a complex cancelling of variables to achieve an algebraic constraint, as in PCA(36), results in NMF being able to provide a better representation of the dynamics. Further, the only fundamental assumption made in NMF is that the data can be represented exclusively as positive values. Our other methods require more stringent assumptions. UMAP makes the assumption that the data can be embedded on a low-dimensional manifold(52). While very advantageous in less variant data (large scale trial averaged electrophysiological recordings(51)), we argue our data is far too variant with limited scalability (single trial, single recording) for this method to be as effective. ICA assumes that components are both non-normal, and independent(49). While the assumption of non-normality of traces, variables, is a positive feature given that calcium traces are clearly non-normal data, we argue it is unfair to assume sub-network dynamics are completely independent of each other. Finally, PCA assumes the data is best described by the eigenvectors of the covariance matrix(36). We note that this algebraic constraint does provide PCA the unique attribute of the model being the same for the first i components, regardless of the number of components fit. NMF, UMAP, AND ICA will differ in their representation of the data, dependent on the number of components fit too. As a result, PCA is guaranteed to perfectly lossless in an inverse transformation of the data, given all components. However, PCA is clearly less efficient than NMF in generating compact descriptions of data like ours.

This pipeline was developed with the intent of analyzing state dependent neuronal network dynamics, and future work will analyze differential network dynamics that arise from unique experimental contexts. However, NMF, and other DR methods, require an underlying structure in the data and can be especially sensitive to potential extremes in the number of components and rates of activity in the network. An extreme sparsity or extreme saturation of events will lead to a much noisier representation in the model. Further, a fundamental assumption of DR is that the data can be represented with significantly fewer components than there are neurons(9). Data that would carry hundreds of underlying drivers of activity (or any case where the number of patterns approach the number of neurons) would be obfuscated in a model such as this. This analysis is able to go beyond summary statistics, providing a low dimensional representation of dynamics while still considering the activities of each neuron. Therefore, this approach will enable more sophisticated and nuanced analysis of different states of neuronal networks, and how they shift as a function of context.

In sum, NMF provides a superior method for the holistic analysis of network dynamics recorded in calcium imaging. The mathematical constraints required by NMF, linearity and positivity, complement the nature of fluorescent recording and neuronal activities well. Further, the parts-based nature of NMF provides a simple and interpretable representation of sub-networks of activity summing to drive macroscopic dynamics. As a result, we have developed an NMF pipeline to be an exceptionally valuable tool for elegantly demystifying shifting neuronal network dynamics.

Acknowledgements

We thank Professor Chandramouli Chandrasekaran for providing a significant amount of the motivation for this research. We further thank Dr. Jacob F. Norman and Dr. Fernando R. Fernandez for their feedback and guidance in the development of this research. Research reported in this publication was supported by the National Institute Of Mental Health of the National Institutes of Health (F31MH133306 to DC, R01MH085074 to JW). The content is solely

the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. Further, this research was partially supported by NIH Translational Research in Biomaterials Training Grant: T32 EB006359.

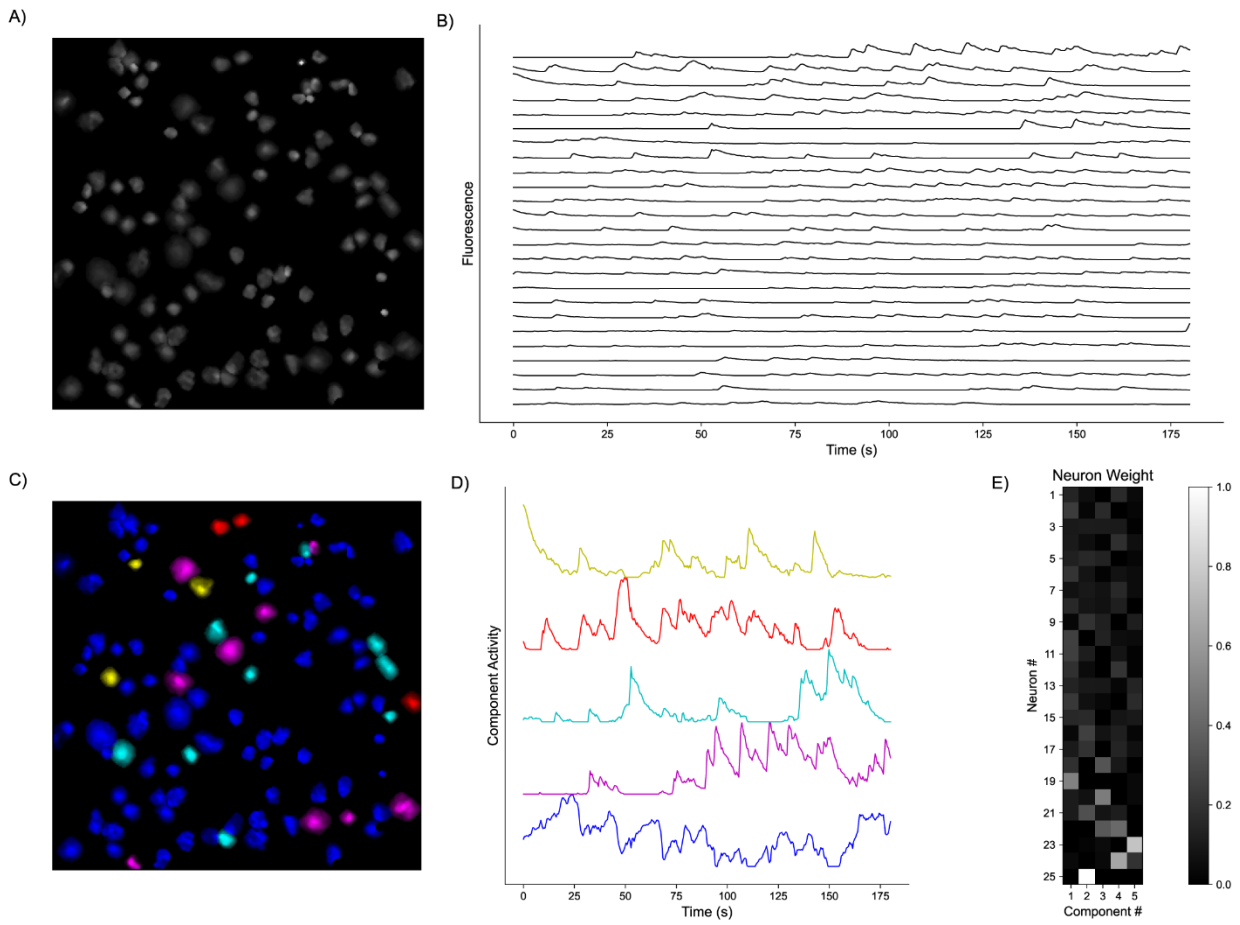

Fig. 1. Representative Model Pipeline. A) Recording Field of View (FOV) with detected Regions of Interest (ROIs). **B)** Extracted fluorescent activity traces from ROIs. **C)** Same recording FOV shown in A) with ROIs colored according to largest component contribution. **D)** Decomposed activities of the components shown in C). **E)** Sample truncated heatmap showing neuronal contribution or weight for each component.

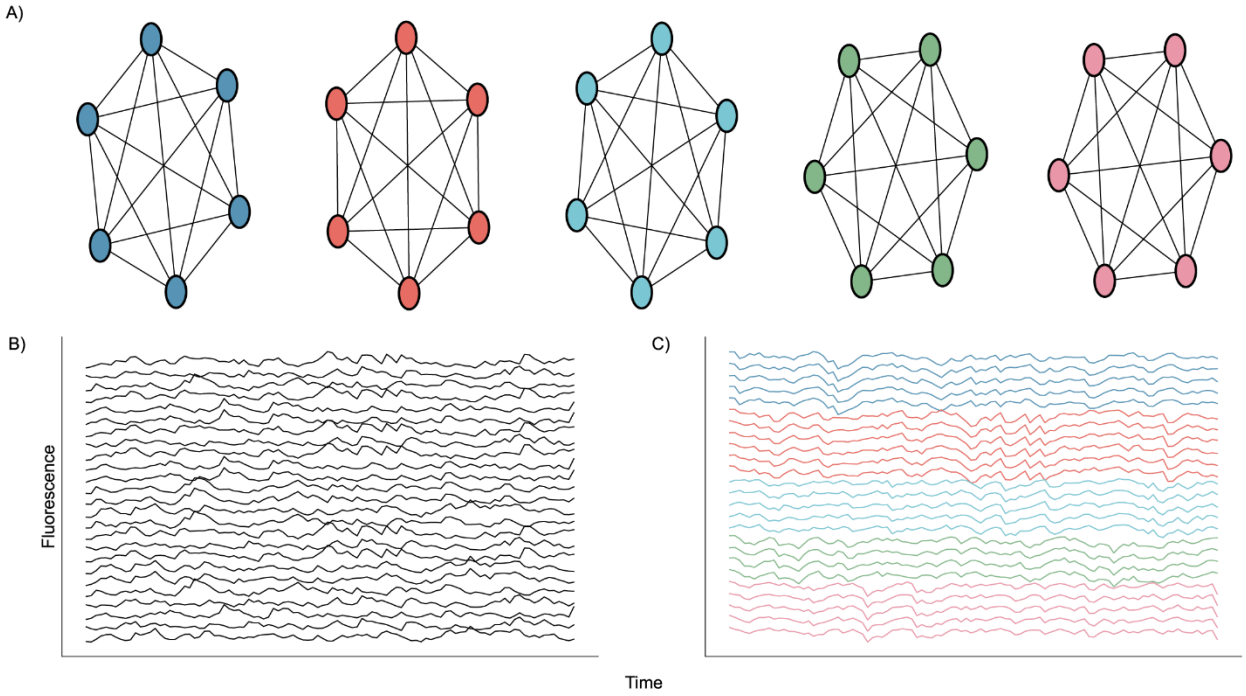

Fig. 2. Nodal Network Architecture. A) Diagram showing perfect intraconnection between neurons and independence between nodes. **B)** Randomly selected calcium traces generated by the network. **C)** Same randomly selected calcium traces shown in B), but sorted according to node.

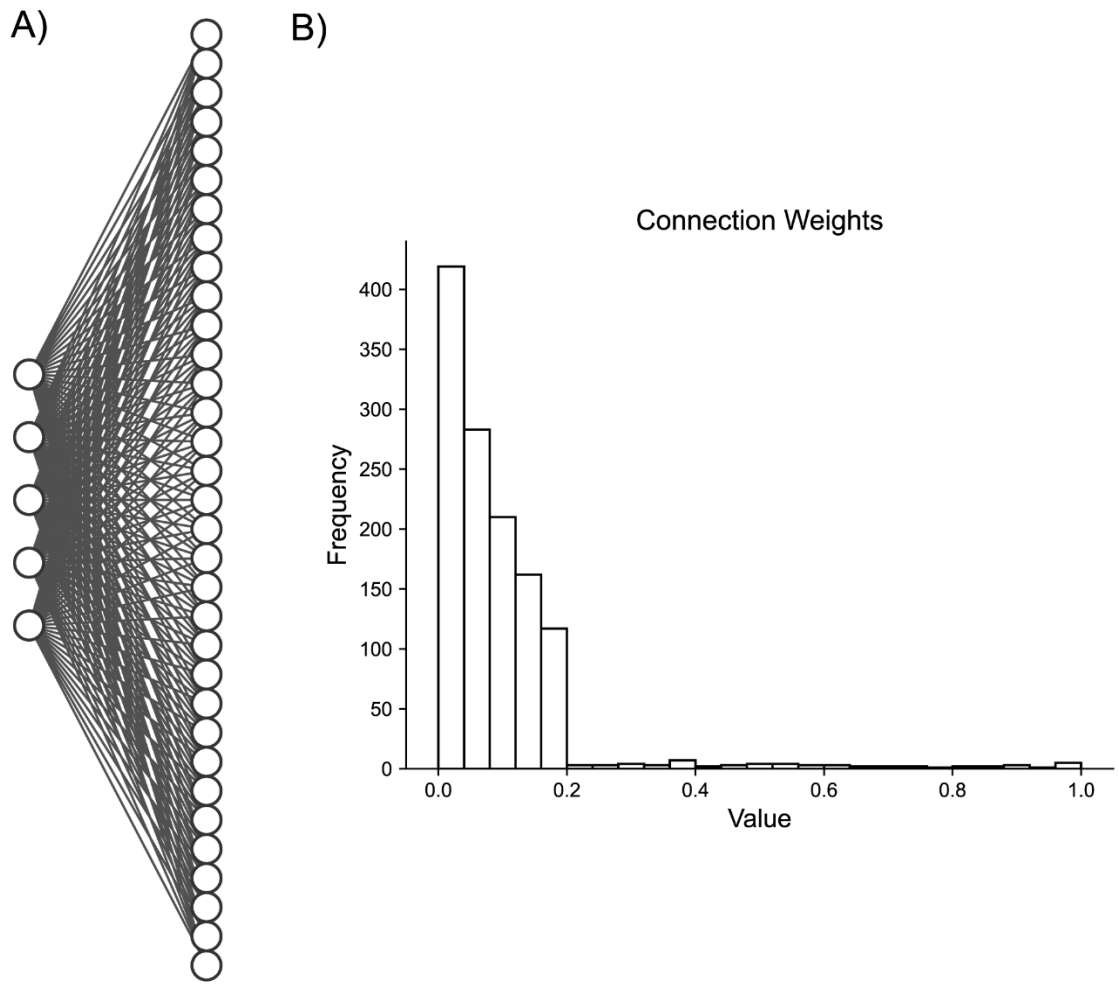

Fig. 3. Artificial Random Propagating Process Network Architecture. A) Sample connection diagram showing all random processes projecting to all neurons. **B)** Single histogram of generated connections between processes and neurons from sampled distribution.

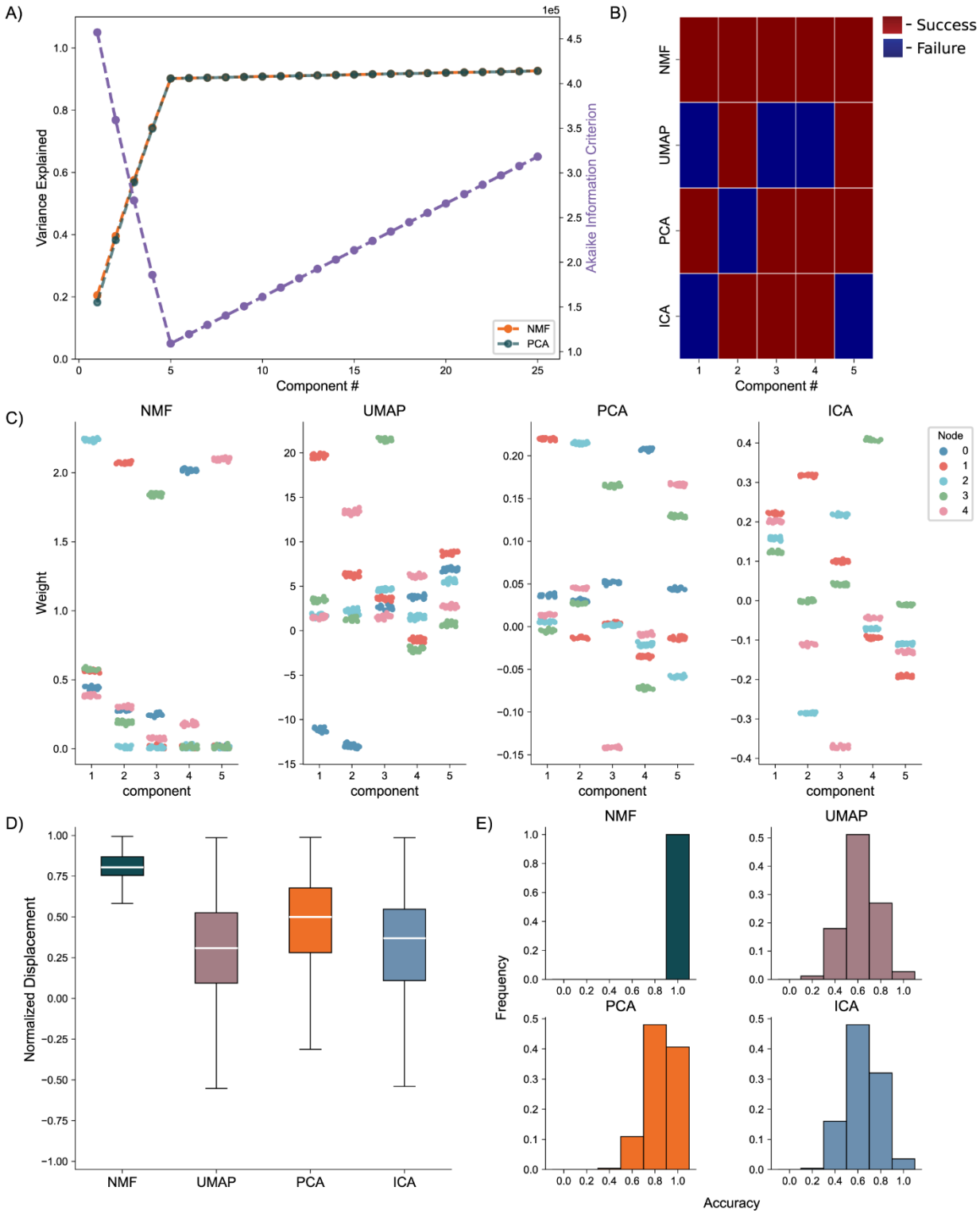
A)
Variance Explained
Akaike Information Criterion
Component #
1e5
NMF
PCA
B)
- Success
- Failure
NMF
UMAP
PCA
ICA
Component #
C)
NMF
UMAP
PCA
ICA
Weight
component
Node
0
1
2
3
4
D)
Normalized Displacement
NMF
UMAP
PCA
ICA
E)
NMF
UMAP
PCA
ICA
Frequency
Accuracy

Fig. 4. Artificial Nodal Network Analysis. A) Variance Explained for NMF and PCA, left axis (n = 256 networks) per component added to the model, and Akaike Information Criterion (AIC) for NMF, right axis (n = 256 networks). **B)** Prediction visualization between nodes and components for single network. **C)** Neuronal weights (y axis) for each model (panel) for each node (color) for each component (x axis). **D)** Average between neurons inside and outside predicted nodes for each component (n = 256 networks). **E)** Model assignment accuracy (n = 256 networks).

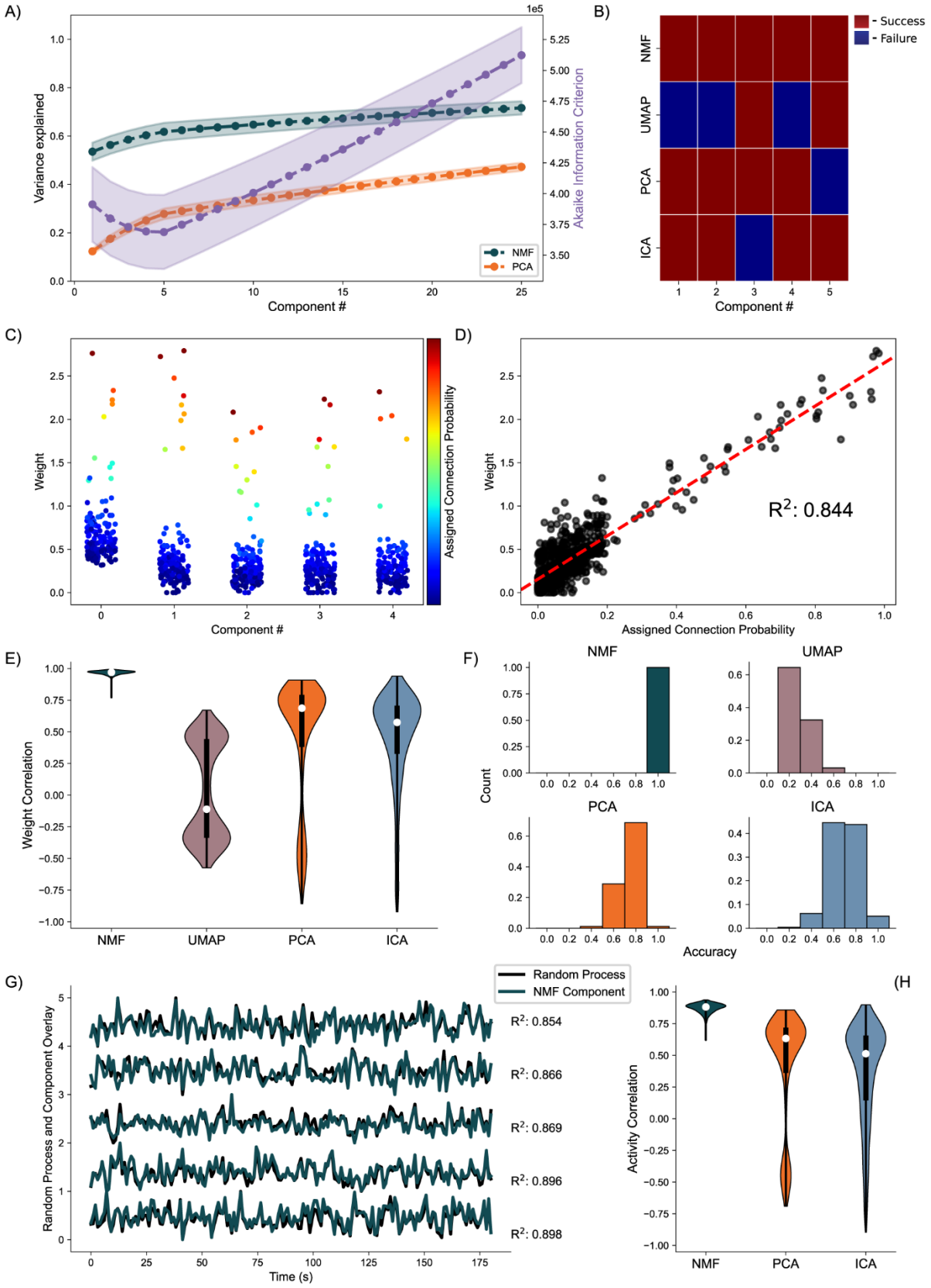
A)
Variance explained
Akaike Information Criterion
Component #
NMF
PCA
B)
Success
Failure
NMF
UMAP
PCA
ICA
Component #
C)
Weight
Component #
Assigned Connection Probability
D)
Weight
Assigned Connection Probability
R^2: 0.844
E)
Weight Correlation
NMF
UMAP
PCA
ICA
F)
NMF
UMAP
PCA
ICA
Count
Accuracy
G)
Random Process
NMF Component
Random Process and Component Overlay
Time (s)
R^2: 0.854
R^2: 0.866
R^2: 0.869
R^2: 0.896
R^2: 0.898
(H
Activity Correlation
NMF
PCA
ICA

Fig. 5. Artificial Random Propagating Process Network Analysis. A) Variance Explained for NMF and PCA, left axis (n = 256 networks) per component added to the model, and Akaike Information Criterion (AIC) for NMF, right axis (n = 256 networks). **B)** Assignment visualization between nodes and components for single network. **C)** Neuronal weights for NMF model (y axis) for each component (x axis) compared to assigned connection probabilities (color scale) for the predicted random process. **D)** Correlation between neuronal weights for NMF model (y axis) and the respective assigned connection probability (x axis). **E)** Average Correlation between neuronal weights for model and the respective assigned connection probabilities (n = 256 networks). **F)** Model assignment accuracy (n = 256 networks). **G)** Correlation between component activity from NMF model and assigned random process activity. **H)** Correlations between component activity from model and assigned random process activity (n = 256 networks).

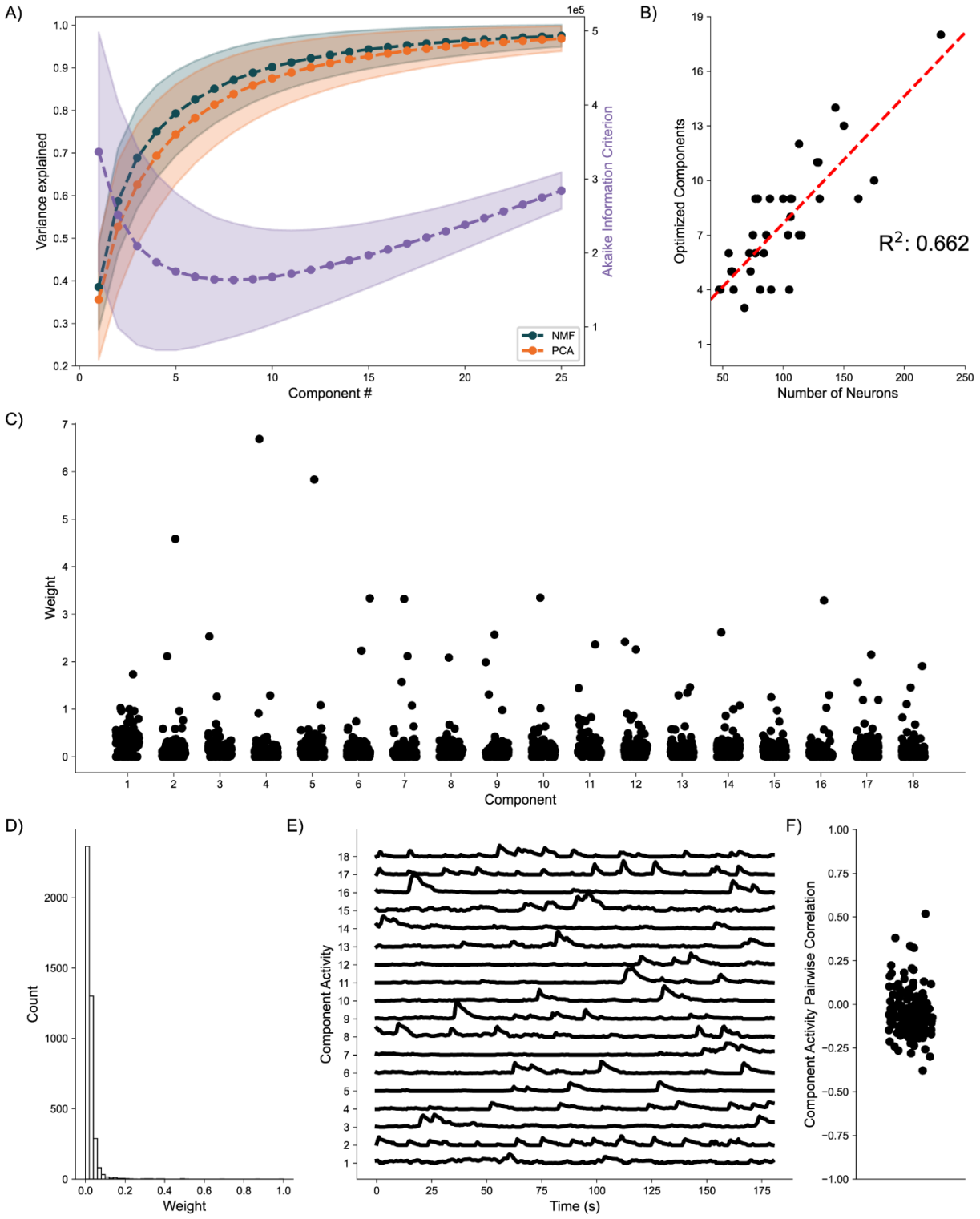
A)
Variance explained
Akaike Information Criterion
Component #
NMF
PCA
1e5
B)
Optimized Components
Number of Neurons
R^2: 0.662
C)
Weight
Component
D)
Count
Weight
E)
Component Activity
Time (s)
F)
Component Activity Pairwise Correlation

Fig. 6. Baseline Awake State Network Analysis A) Variance Explained for NMF and PCA, left axis (n = 36 mice) per component added to the model, and Akaike Information Criterion (AIC) for NMF, right axis (n = 36 mice). **B)** Correlation between number of neurons (x axis) and number of optimized components (y axis) (n = 36 mice). **C)** Neuronal weights for NMF model (y axis) for each component (x axis) for a single mouse. **D)** Neuronal weight histogram from NMF model for the same mouse as C). **E)** Component activity for NMF model of the same mouse as in C) and D). **F)** Pairwise correlations between component activities shown in E).

References:

1. Pnevmatikakis EA. Analysis pipelines for calcium imaging data. Curr Opin Neurobiol. 2019 Apr;55:15–21.

2. Stevenson IH, Kording KP. How advances in neural recording affect data analysis. Nat Neurosci. 2011 Feb;14(2):139–42.

3. Stringer C, Pachitariu M. Computational processing of neural recordings from calcium imaging data. Curr Opin Neurobiol. 2019 Apr;55:22–31.

4. Birkner A, Tischbirek CH, Konnerth A. Improved deep two-photon calcium imaging in vivo. Cell Calcium. 2017 Jun;64:29–35.

5. Tonegawa S, Liu X, Ramirez S, Redondo R. Memory Engram Cells Have Come of Age. Neuron. 2015;87(5):918–31.

6. Josselyn SA, Tonegawa S. Memory engrams: Recalling the past and imagining the future. Science. 2020;367(6473).

7. Kitamura T, Ogawa SK, Roy DS, Okuyama T, Morrissey MD, Smith LM, et al. Engrams and circuits crucial for Systems Consolidation of a Memory. Science. 2017;78(April):73–8.

8. Norman JF, Rahsepar B, Noueihed J, White JA. Determining the optimal expression method for dual-color imaging. J Neurosci Methods. 2021 Mar;351:109064.

9. Cunningham JP, Yu BM. Dimensionality reduction for large-scale neural recordings. Nat Neurosci. 2014 Nov;17(11):1500–9.

10. Sanger TD, Kalaska JF. Crouching tiger, hidden dimensions. Nat Neurosci. 2014 Mar;17(3):338–40.

11. Géron A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems. Second edition. Beijing [China] ; Sebastopol, CA: O'Reilly Media, Inc; 2019. 819 p.

12. Peyrache A, Benchenane K, Khamassi M, Wiener SI, Battaglia FP. Principal component analysis of ensemble recordings reveals cell assemblies at high temporal resolution. J Comput Neurosci. 2010 Aug;29(1–2):309–25.

13. Low RJ, Lewallen S, Aronov D, Nevers R, Tank DW. Probing variability in a cognitive map using manifold inference from neural dynamics [Internet]. Neuroscience; 2018 Sep [cited 2022 Oct 4]. Available from: http://biorxiv.org/lookup/doi/10.1101/418939

14. Giovannucci A, Friedrich J, Gunn P, Kalfon J, Brown BL, Koay SA, et al. CaImAn an open source tool for scalable calcium imaging data analysis. eLife. 2019 Jan 17;8:e38173.

15. Pnevmatikakis EA, Soudry D, Gao Y, Machado TA, Merel J, Pfau D, et al. Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. Neuron. 2016 Jan;89(2):285–99.

16. Pachitariu M, Stringer C, Dipoppa M, Schröder S, Rossi LF, Dalgleish H, et al. Suite2p: beyond 10,000 neurons with standard two-photon microscopy [Internet]. Neuroscience; 2016 Jun [cited 2022 Oct 4]. Available from: http://biorxiv.org/lookup/doi/10.1101/061507

17. Pnevmatikakis EA, Giovannucci A. NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. J Neurosci Methods. 2017 Nov;291:83–94.

18. Romano SA, Pérez-Schuster V, Jouary A, Boulanger-Weill J, Candeo A, Pietri T, et al. An integrated calcium imaging processing toolbox for the analysis of neuronal population dynamics. Graham LJ, editor. PLOS Comput Biol. 2017 Jun 7;13(6):e1005526.

19. Batty E, Whiteway MR, Saxena S, Biderman D, Abe T, Musall S, et al. BehaveNet: nonlinear embedding and Bayesian neural decoding of behavioral videos. :12.

20. Sotskov VP, Pospelov NA, Plusnin VV, Anokhin KV. Calcium Imaging Reveals Fast Tuning Dynamics of Hippocampal Place Cells and CA1 Population Activity during Free Exploration Task in Mice. Int J Mol Sci. 2022 Jan 7;23(2):638.

21. Rubin A, Sheintuch L, Brande-Eilat N, Pinchasof O, Rechavi Y, Geva N, et al. Revealing neural correlates of behavior without behavioral measurements. Nat Commun. 2019 Dec;10(1):4745.

22. Driscoll LN, Pettit NL, Minderer M, Chettih SN, Harvey CD. Dynamic Reorganization of Neuronal Activity Patterns in Parietal Cortex. Cell. 2017 Aug;170(5):986-999.e16.

23. Wenzel M, Han S, Smith EH, Hoel E, Greger B, House PA, et al. Reduced Repertoire of Cortical Microstates and Neuronal Ensembles in Medically Induced Loss of Consciousness. Cell Syst. 2019 May;8(5):467-474.e4.

24. Anowar F, Sadaoui S, Selim B. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). Comput Sci Rev. 2021 May;40:100378.

25. Izenman AJ. Introduction to manifold learning: Introduction to manifold learning. Wiley Interdiscip Rev Comput Stat. 2012 Sep;4(5):439–46.

26. Ghandour K, Ohkawa N, Fung CCA, Asai H, Saitoh Y, Takekawa T, et al. Orchestrated ensemble activities constitute a hippocampal memory engram. Nat Commun. 2019 Dec;10(1):2637.

27. Nagayama M, Aritake T, Hino H, Kanda T, Miyazaki T, Yanagisawa M, et al. Detecting cell assemblies by NMF-based clustering from calcium imaging data. Neural Netw. 2022 May;149:29–39.

28. Nagayama M, Aritake T, Hino H, Kanda T, Miyazaki T, Yanagisawa M, et al. Sleep State Analysis Using Calcium Imaging Data by Non-negative Matrix Factorization. In: Tetko IV, Kůrková V, Karpov P, Theis F, editors. Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation [Internet]. Cham: Springer International Publishing; 2019 [cited 2022 Oct 4]. p. 102–13. (Lecture Notes in Computer Science; vol. 11727). Available from: http://link.springer.com/10.1007/978-3-030-30487-4_8

29. Briggman KL, Abarbanel HDI, Kristan WB. Optical Imaging of Neuronal Populations During Decision-Making. Science. 2005 Feb 11;307(5711):896–901.

30. Ahrens MB, Li JM, Orger MB, Robson DN, Schier AF, Engert F, et al. Brain-wide neuronal dynamics during motor adaptation in zebrafish. Nature. 2012 May;485(7399):471–7.

31. Morcos AS, Harvey CD. History-dependent variability in population dynamics during evidence accumulation in cortex. Nat Neurosci. 2016 Dec;19(12):1672–81.

32. Makino H, Ren C, Liu H, Kim AN, Kondapaneni N, Liu X, et al. Transformation of Cortex-wide Emergent Properties during Motor Learning. Neuron. 2017 May;94(4):880-890.e8.

33. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999 Oct;401(6755):788–91.

34. Devarajan K. Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. PLoS Comput Biol. 2008;4(7):12.

35. Miyazaki T, Kanda T, Tsujino N, Ishii R, Nakatsuka D, Kizuka M, et al. Dynamics of Cortical Local Connectivity during Sleep–Wake States and the Homeostatic Process. Cereb Cortex. 2020 Jun 1;30(7):3977–90.

36. Shlens J. A Tutorial on Principal Component Analysis [Internet]. arXiv; 2014 [cited 2022 Oct 4]. Available from: http://arxiv.org/abs/1404.1100

37. Cunningham JP, Ghahramani Z. Linear Dimensionality Reduction: Survey, Insights, and Generalizations. :42.

38. Roh J, Cheung VCK, Bizzi E. Modules in the brain stem and spinal cord underlying motor behaviors. J Neurophysiol. 2011 Sep;106(3):1363–78.

39. Cheung VCK, Devarajan K, Severini G, Turolla A, Bonato P. Decomposing time series data by a non-negative matrix factorization algorithm with temporally constrained coefficients. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) [Internet]. Milan: IEEE; 2015 [cited 2022 Oct 4]. p. 3496–9. Available from: http://ieeexplore.ieee.org/document/7319146/

40. Akaike H. Information Theory and an Extension of the Maximum Likelihood Principle. In: Parzen E, Tanabe K, Kitagawa G, editors. Selected Papers of Hirotugu Akaike [Internet]. New York, NY: Springer New York; 1998 [cited 2023 Apr 20]. p. 199–213. (Springer Series in Statistics). Available from: http://link.springer.com/10.1007/978-1-4612-1694-0_15

41. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. :6.

42. Esposito F. A Review on Initialization Methods for Nonnegative Matrix Factorization: Towards Omics Data Experiments. Mathematics. 2021 Apr 29;9(9):1006.

43. Hafshejani SF, Moaberfard Z. Initialization for Nonnegative Matrix Factorization: a Comprehensive Review [Internet]. arXiv; 2021 [cited 2022 Oct 4]. Available from: http://arxiv.org/abs/2109.03874

44. Boutsidis C, Gallopoulos E. SVD based initialization: A head start for nonnegative matrix factorization. Pattern Recognit. 2008 Apr;41(4):1350–62.

45. Harvey CD, Coen P, Tank DW. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. Nature. 2012 Apr;484(7392):62–8.

46. Kato HK, Chu MW, Isaacson JS, Komiyama T. Dynamic Sensory Representations in the Olfactory Bulb: Modulation by Wakefulness and Experience. Neuron. 2012 Dec;76(5):962–75.

47. Awal MR, Wirak GS, Gabel CV, Connor CW. Collapse of Global Neuronal States in *Caenorhabditis elegans* under Isoflurane Anesthesia. Anesthesiology. 2020 Jul 1;133(1):133–44.

48. Mukamel EA, Nimmerjahn A, Schnitzer MJ. Automated Analysis of Cellular Signals from Large-Scale Calcium Imaging Data. Neuron. 2009 Sep;63(6):747–60.

49. Shlens J. A Tutorial on Independent Component Analysis [Internet]. arXiv; 2014 [cited 2023 Feb 25]. Available from: http://arxiv.org/abs/1404.2986

50. Cid E, Marquez-Galera A, Valero M, Gal B, Medeiros DC, Navarron CM, et al. Sublayer- and cell-type-specific neurodegenerative transcriptional trajectories in hippocampal sclerosis. Cell Rep. 2021 Jun;35(10):109229.

51. Lee EK, Balasubramanian H, Tsolias A, Anakwe SU, Medalla M, Shenoy KV, et al. Non-linear dimensionality reduction on extracellular waveforms reveals cell type diversity in premotor cortex. eLife. 2021 Aug 6;10:e67490.

52. McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction [Internet]. arXiv; 2020 [cited 2022 Nov 2]. Available from: http://arxiv.org/abs/1802.03426

53. Dayan P, Abbott LF, Abbott LF. Theoretical neuroscience: computational and mathematical modeling of neural systems. First paperback ed. Cambridge, Mass.: MIT Press; 2005. 460 p. (Computational neuroscience).

54. Vogelstein JT, Watson BO, Packer AM, Yuste R, Jedynak B, Paninski L. Spike Inference from Calcium Imaging Using Sequential Monte Carlo Methods. Biophys J. 2009 Jul;97(2):636–55.

55. Song S, Sjöström PJ, Reigl M, Nelson S, Chklovskii DB. Highly Nonrandom Features of Synaptic Connectivity in Local Cortical Circuits. Friston KJ, editor. PLoS Biol. 2005 Mar 1;3(3):e68.

56. Dana H, Sun Y, Mohar B, Hulse BK, Kerlin AM, Hasseman JP, et al. High-performance calcium sensors for imaging activity in neuronal populations and microcompartments. Nat Methods. 2019 Jul;16(7):649–57.

57. Noueihed J. Balance of Excitation and Inhibition in the Primary Somatosensory Cortex Layer 2/3 Under Isoflurane Anesthesia (Submitted).

58. Noueihed J. Multisession Processing of Multiphoton Calcium Imaging (Submitted).