Figures and figure supplements

Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience

Emily L Mackevicius et al
Figure 1. Convolutional NMF factorization. (A) NMF (non-negative matrix factorization) approximates a data matrix describing the activity of \( N \) neurons at \( T \) timepoints as a sum of \( K \) rank-one matrices. Each matrix is generated as the outer product of two nonnegative vectors: \( w_k \) of length \( N \), which stores a neural ensemble, and \( h_k \) of length \( T \), which holds the times at which the neural ensemble is active, and the relative amplitudes of this activity. (B) Convolutional NMF also approximates an \( N/C^2 \times T \) data matrix as a sum of \( K \) matrices. Each matrix is generated as the convolution of two components: a non-negative matrix \( w_k \) of dimension \( N \times L \) that stores a sequential pattern of the \( N \) neurons at \( L \) lags, and a vector of temporal loadings, \( h_k \), which holds the times at which each factor pattern is active in the data, and the relative amplitudes of this activity. (C) Three types of inefficiencies present in unregularized convNMF: Type 1, in which two factors are used to reconstruct the same instance of a sequence; Type 2, in which two factors reconstruct a sequence in a piece-wise manner; and Type 3, in which two factors are used to reconstruct different instances of the same sequence. For each case, the factors (\( W \) and \( H \)) are shown, as well as the reconstruction (\( \hat{X} = W \odot H = w_1 \odot h_1 + w_2 \odot h_2 + \cdot \cdot \cdot \)).

DOI: https://doi.org/10.7554/eLife.38471.002
Figure 1—figure supplement 1. Quantifying the effect of different penalties on convNMF. (A) Example factorizations of a synthetic dataset for convolutional NMF, and for convolutional NMF with three different penalties designed to eliminate correlations in H or in both H and W. Notice that different penalties lead to different types of redundancies in the corresponding factorizations. (B) Quantification of correlations in H and W for each different penalty. H correlations are measured using \( \| H S H^T \|_{1, ij} \) and W correlations are measured using \( \| W_{\text{flat}} W_{\text{flat}}^T \|_{1, ij} \), where \( W_{\text{flat}} = \sum_w (W \cdot w) \).

DOI: https://doi.org/10.7554/eLife.38471.003
Figure 2. Effect of the x-ortho penalty on the factorization of sequences. (A) A simulated dataset with three sequences. Also shown is a factorization with x-ortho penalty ($K = 20$, $L = 50$, $\lambda = 0.003$). Each significant factor is shown in a different color. At left are the exemplar patterns ($W$) and on top are the timecourses ($H$). (B) Reconstruction error as a function of iteration number. Factorizations were run on a simulated dataset with three sequences and 15,000 timebins ($\approx 60$ instances of each sequence). Twenty independent runs are shown. Here, the algorithm converges to within 10% of the asymptotic error value within $\approx 100$ iterations. (C) The x-ortho penalty produces more consistent factorizations than unregularized convNMF across 400 independent fits ($K = 20$, $L = 50$, $\lambda = 0.003$). (D) The number of statistically significant factors (Figure 2—figure supplement 1) vs. the number of ground-truth sequences for factorizations with and without the x-ortho penalty. Shown for each condition is a vertical histogram representing the number of significant factors over 20 runs ($K = 20$, $L = 50$, $\lambda = 0.003$). (E) Factorization with x-ortho penalty of two simulated neural sequences with shared neurons that participate at the same latency. (F) Same as E but for two simulated neural sequences with shared neurons that participate at different latencies.

DOI: https://doi.org/10.7554/eLife.38471.006
**Figure 2—figure supplement 1.** Outline of the procedure used to assess factor significance. (A) Distribution of overlap values between an extracted factor and the held-out data. (B) A null factor was constructed by randomly circularly shifting each row of a factor independently. Many null factors were constructed and the distribution of overlap values \( W^T X \) was measured between each null factor and the held-out data. (C) A comparison of the skewness values for each null factor and the skewness of the overlaps of the original extracted factor. A factor is deemed significant if its skewness is significantly greater than the distribution of skewness values for the null factor overlaps.

DOI: https://doi.org/10.7554/eLife.38471.007
Figure 2—figure supplement 2. Number of significant factors as a function of λ for datasets containing between 1 and 10 sequences. The number of significant factors obtained by fitting data containing between 1 and 10 ground truth sequences using the x-ortho penalty (K = 20, L = 50) for a large range of values of λ. For each value of λ, 20 fits are shown and the mean is shown as a solid line. Each color corresponds to a ground-truth dataset containing a different number of sequences and no added noise. Values of λ ranging between 0.001 and 0.1 tended to return the correct number of significant sequences at least 90% of the time.

DOI: https://doi.org/10.7554/eLife.38471.008
Figure 3. Testing factorization performance on sequences contaminated with noise. Performance of the x-ortho penalty was tested under four different noise conditions: (A) probabilistic participation, (B) additive noise, (C) temporal jitter, and (D) sequence warping. For each noise type, we show: (top) examples of synthetic data at three different noise levels; (middle) similarity of extracted factors to ground-truth patterns across a range of noise levels (20 fits for each level); and (bottom) examples of extracted factors W’s for one of the ground-truth patterns. Examples are shown at the same three noise levels illustrated in the top row. In these examples, the algorithm was run with $K = 20$, $L = 50$ and $\lambda = 2\lambda_0$ (via the procedure described in Figure 4). For C, jitter displacements were draw from a discrete Gaussian distribution with the standard deviation in timesteps shown above. For D, time warp conditions 1–10 indicate: 0, 66, 133, 200, 266, 333, 400, 466, 533 and 600 max % stretching respectively. For results at different values of $\lambda$, see Figure 3—figure supplement 1. DOI: https://doi.org/10.7554/eLife.38471.009
Figure 3—figure supplement 1. Robustness to noise at different values of $\lambda$. Performance of the x-ortho penalty was tested under four different noise conditions, at different values of $\lambda$ than in Figure 3 (where $\lambda = 2\lambda_0$): (A) probabilistic participation, $\lambda = 5\lambda_0$, (B) additive noise, $\lambda = \lambda_0$ (C) timing jitter, $\lambda = 5\lambda_0$ and (D) sequence warping, $\lambda = 5\lambda_0$. For each noise type, we show: (top) examples of synthetic data at three different noise levels; (middle) similarity of x-ortho factors to ground-truth factors across a range of noise levels (20 fits for each level); and (bottom) example of one of the W’s extracted at three different noise levels (same conditions as data shown above). The algorithm was run with $K = 20$, $L = 50$.

DOI: https://doi.org/10.7554/eLife.38471.010
Figure 3—figure supplement 2. Robustness to small dataset size when using the $x$-ortho penalty. (A) A short (400 timestep) dataset containing one example each of three ground-truth sequences, as well as additive noise. (B) As a function of dataset size, similarity of extracted factors to noiseless, ground-truth factors. At each dataset size, 20 independent fits of penalized convNMF are shown. Median shown in red. Three examples of each sequence were sufficient to achieve similarity scores within 10% of asymptotic performance. (C) Example factors fit on data containing 2, 3, 4 or 20 examples of each sequence. Extracted factors were significant on held-out data compared to null (shuffled) factors even when training and test datasets each contained only 2 examples of each sequence.

DOI: https://doi.org/10.7554/eLife.38471.011
Figure 3—figure supplement 3. Robustness to different types of sequences. Characterization of the x-ortho penalty for additional types of noise. (A) An example of a factorization for a sequence with large gaps between members of the sequence. (B–D) Example factorizations of sequences with

Figure 3—figure supplement 3 continued on next page
neuronal activations that are highly overlapped in time. (B) An example of an x-ortho penalized factorization that reconstructs the data using complex patterns in $W$ and $H$. (C) An example of an x-ortho penalized factorization with the addition of an L1 penalty on $H$ models the data as an overlapping pattern with sparse activations. (D) An example of an x-ortho penalized factorization with the addition of an L1 penalty on $W$ models the data as a non-overlapping pattern with dense activations. (E) An example of an x-ortho penalized factorization for data in which neurons have varying durations of activation which form two patterns. (F) An example of an x-ortho penalized factorization for data in which neurons have varying durations of activation which are random. (G–I) Examples factorizations of sequences with statistics that vary systematically. (G) An example of an x-ortho penalized factorization for data in which neurons have systematically varying changes in duration of activity. (H) An example of an x-ortho penalized factorization for data in which neurons have systematically varying changes in the gaps between members of the sequence. (I) An example of an x-ortho penalized factorization for data in which neurons have systematically varying changes in the amount of jitter.
Figure 4. Procedure for choosing $\lambda$ for a new dataset based on finding a balance between reconstruction cost and $x$-ortho cost. (A) Simulated data containing three sequences in the presence of participation noise (50% participation probability). This noise condition is used for the tests in (B–F). (B) Normalized reconstruction cost $\|\tilde{X} - X\|_F^2$ and cross-orthogonality cost $\|\|(W @ X)S^{H}\|_F\|_1$ as a function of $\lambda$ for 20 fits of these data. The cross-over point $\lambda_0$ is marked with a black circle. Note that in this plot the reconstruction cost and cross-orthogonality...
Figure 4 continued

cost are normalized to vary between 0 and 1. (C) The number of significant factors obtained as a function of $\lambda$; 20 fits, mean plotted in orange. Red arrow at left indicates the correct number of sequences (three). (D) Fraction of fits returning the correct number of significant factors as a function of $\lambda$. (E) Similarity of extracted factors to ground-truth sequences as a function of $\lambda$. (F) Composite performance, as the product of the curves in (D) and (E) (smoothed using a three sample boxcar, plotted in orange with a circle marking the peak). Shaded region indicates the range of $\lambda$ that works well (± half height of composite performance). (G–L) same as (A–F) but for simulated data containing three noiseless sequences. (M) Summary plot showing the range of values of $\lambda$ (vertical bars), relative to the cross-over point $\lambda_0$, that work well for each noise condition (± half height points of composite performance). Circles indicate the value of $\lambda$ at the peak of the smoothed composite performance. For each noise type, results for all noise levels from Figure 3 are shown (increasing color saturation at high noise levels; Green, participation: 90, 80, 70, 60, 50, 40, 30, and 20%; Orange, additive noise 0.5, 1, 2, 2.5, 3, 3.5, and 4%; Purple, jitter: SD of the distribution of random jitter: 5, 10, 15, 20, 25, 30, 35, 40, and 45 timesteps; Grey, timewarp: 66, 133, 200, 266, 333, 400, 466, 533, 600, and 666 max % stretching. Asterisk (*) indicates the noise type and level used in panels (A–F). Gray band indicates a range between 2$\lambda_0$ and 5$\lambda_0$, a range that tended to perform well across the different noise conditions. In real data, it may be useful to explore a wider range of $\lambda$.

DOI: https://doi.org/10.7554/eLife.38471.013
Figure 4—figure supplement 1. Analysis of the best range of $\lambda$. Here, we quantify the full width at the half maximum for the composite performance scores in different noise conditions. For each condition, a box and whisker plot quantifies the number of orders of magnitude over which a good factorization is returned (median denoted by a white circle). Next to each box plot individual points are shown, corresponding to different noise level. Color saturation reflects noise level as in Figure 4.
DOI: https://doi.org/10.7554/eLife.38471.014
Figure 4—figure supplement 2. Procedure for choosing $\lambda$ applied to data with shared neurons. (A) Simulated data containing two patterns which share 50% of their neurons, in the presence of participation noise (70% similarity).
Figure 4—figure supplement 2 continued

(B) Normalized reconstruction cost ($\|\tilde{X} - X\|_F^2$) and cross-orthogonality cost ($\|\|W_{\odot}X\|SH\|_1,\infty\|$) as a function of $\lambda$ for these data. The cross-over point $\lambda_0$ is marked with a black circle. (C) The number of significant factors obtained from 20 fits of these data as a function of $\lambda$ (mean number plotted in orange). The correct number of factors (two) is marked by a red triangle. (D) The fraction of fits returning the correct number of significant factors as a function of $\lambda$. (E) Similarity of the top two factors to ground-truth (noiseless) factors as a function of $\lambda$. (F) Composite performance measured as the product of the curves shown in (D) and (E), (smoothed curve plotted in orange with a circle marking the peak). Shaded region indicates the range of $\lambda$ that works well ($\pm$ half height of composite performance). For this dataset, the best performance occurs at $\lambda = 5\lambda_0$, while a range of $\lambda$ between $2\lambda_0$ and $10\lambda_0$ performs well.

DOI: https://doi.org/10.7554/eLife.38471.015
Figure 4—figure supplement 3. Using cross-validation on held-out (masked) data to choose $\lambda$. A method for choosing a reasonable value of $\lambda$ based on cross validation is shown for five different noise types (each column shows a different noise type, from left to right: (I) participation noise, (II) additive noise, (III) jitter, (IV) temporal warping), and (V) a lower level of participation noise. The cross-validated test error is calculated by fitting x-ortho penalized factorizations while randomly holding out 10% of the elements in the data matrix as a test set (Wold, 1978; Bro et al., 2008). In many of our test datasets, there was a minimum or a divergence point in the difference between the test and training error, that agreed with the procedure described in Figure 4, based on $\lambda_0$. (A) Examples of each dataset. (B) Test error (blue) and training error (red) as a function of $\lambda$ for each of the different noise conditions. (C) The difference between the test error and training error values shown above. (D) Normalized reconstruction cost ($\|\|W_{\otimes}X\|\|_F$) and cross-orthogonality cost ($\|\|X - X_0\|\|_F$) as a function of $\lambda$ for each of the different noise conditions. (E) Composite performance as a function of $\lambda$. Panels D and E are identical to those in Figure 4, and are included here for comparison. (V) These data have a lower amount of participation noise than (I). Note that in low-noise conditions, test error may not exhibit a minima within the range of $\lambda$ that produces the ground truth number of factors.

DOI: https://doi.org/10.7554/eLife.38471.016
Figure 4—figure supplement 4. Quantifying the effect of L1 sparsity penalties on $W$ and $H$. (A) An example window of simulated data with three sequences and 40% dropout. (B) The fraction of fits to data with the noise level in (A) that yielded three significant factors, as a function of two L1 sparsity regularization parameters on $W$ and $H$. Each bin represents 20 fits of sparsity penalized convNMF with $K = 20$ and $L = 50$. (C) The mean similarity to ground-truth for the same 20 factorizations as in (B). (D–F) same as panels A–C but with additional noise events in 2.5% of the bins. (H–J) same as panels A–C but with a jitter standard deviation of 20 bins. (K–M) same as panels A–C but for warping noise with a maximum of 260% warping.

DOI: https://doi.org/10.7554/eLife.38471.017
Figure 4—figure supplement 5. Comparing the performance of convNMF with an x-ortho or a sparsity penalty. (A) The fraction of 20 x-ortho penalized fits which had the same number of significant factors as the ground-truth for all noise conditions shown in Figure 3 at the λ with the best performance. (B) The similarity to ground-truth for 20 x-ortho penalized fits to all noise conditions shown in Figure 3 at the λ with the best performance. (C) The number of significant factors for 100 fits with an x-ortho penalty (black) and with sparsity penalties on W and H (red) of four different noise conditions at the level indicated in (A) and (B). Penalty parameters used in (C–E) were selected by performing a parameter sweep and selecting the parameters which gave the maximum composite score as described above. (D) The fraction of 20 x-ortho or sparsity penalized fits with the ground truth number of significant sequences. Noise conditions are the same as in (C). Values for λ were selected as those that give the highest composite performance (see Figure 4F). (E) Similarity to ground-truth for the fits shown in (C–D). Median is shown with black dot and bottom and top edges of boxes indicate the 25th and 75th percentiles.

DOI: https://doi.org/10.7554/eLife.38471.018
Figure 5. Direct selection of K using the diss metric, a measure of the dissimilarity between different factorizations. Panels show the distribution of diss as a function of K for several different noise conditions. Lower values of diss indicate greater consistency or stability of the factorizations, an indication of low factor redundancy. (A) probabilistic participation (60%), (B) additive noise (2.5% bins), (C) timing jitter (SD = 20 bins), and (D) sequence warping (max warping = 266%). For each noise type, we show: (top) examples of synthetic data; (bottom) the diss metric for 20 fits of convNMF for K from 1 to 10; the black line shows the median of the diss metric and the dotted red line shows the true number of factors.

DOI: https://doi.org/10.7554/eLife.38471.019
Figure 5—figure supplement 1. Direct selection of K using the diss metric for all noise conditions. (A) Participation noise, (B) additive noise, (C) jitter and (D) warping. For each panel, the top shows an example of data with three sequences and each noise type. The bottom panel shows the dissimilarity of factorizations in different levels of noise, as a function of K. A condition with no noise is shown in blue and dark red represents the highest noise condition with the color gradient spanning the levels between. Noise levels are the same as in Figure 3 and Figure 4. Notice that there is often either a minimum or a distinct feature at K = 3, corresponding to the ground-truth number of sequences in the data.

DOI: https://doi.org/10.7554/eLife.38471.020
Figure 5—figure supplement 2. Estimating the number of sequences in a dataset using cross-validation on randomly masked held-out datapoints. (A) Reconstruction error (RMSE) for test data (red) and training data (blue) plotted as a function of the number of components (K) used in convNMF. Twenty independent convNMF fits are shown for each value of K. This panel shows results for 10% participation noise. For synthetic data fits, 10% of the data was held out as the test set. For neural data 5% of the data was held out. Other noise conditions are shown as follows: (B) jitter noise (10 timestep SD); (C) warping (13%); (D) higher additive noise (2.5%); (E) higher jitter noise (25 timestep SD); (F) higher warping (33%). (G) Reconstruction error for test data (red) and training data (blue) plotted as a function of the number of components (K) used in convNMF. This panel shows results for 5% participation noise. For synthetic data fits, 5% of the data was held out as the test set. For neural data 5% of the data was held out. Other noise conditions are shown as follows: (B) jitter noise (10 timestep SD); (C) warping (13%); (D) higher additive noise (2.5%); (E) higher jitter noise (25 timestep SD); (F) higher warping (33%). (G) Reconstruction error for test data (red) and training data (blue) plotted as a function of the number of components (K) used in convNMF. This panel shows results for 5% participation noise. For synthetic data fits, 5% of the data was held out as the test set. For neural data 5% of the data was held out. Other noise conditions are shown as follows: (B) jitter noise (10 timestep SD); (C) warping (13%); (D) higher additive noise (2.5%); (E) higher jitter noise (25 timestep SD); (F) higher warping (33%).
Figure 5—figure supplement 2 continued

error vs. K for neuronal data collected from premotor cortex (area HVC) of a singing bird (Figure 9) and (H) hippocampus of rat 2 performing a left-right alternation task (Figure 8).

DOI: https://doi.org/10.7554/eLife.38471.021
Figure 6. Using penalties to bias toward events-based and parts-based factorizations. Datasets that have neurons shared between multiple sequences can be factorized in different ways, emphasizing discrete temporal events (events-based) or component neuronal ensembles (parts-based), by using orthogonality penalties on $H$ or $W$ to penalize factor correlations (see Table 2). (Left) A dataset with two different ensembles of neurons that participate in two different types of events, with (A) events-based factorization obtained using an orthogonality penalty on $H$ and (B) parts-based factorizations obtained using an orthogonality penalty on $W$. (Right) A dataset with six different ensembles of neurons that participate in three different types of events, with (C) events-based and (D) parts-based factorizations obtained as in (A) and (B).

DOI: https://doi.org/10.7554/eLife.38471.022
Figure 6—figure supplement 1. Biasing factorizations between sparsity in $W$ or $H$. Two different factorizations of the same simulated data, where a sequence is always repeated precisely three times. Both yield perfect reconstructions, and no cross-factor correlations. The factorizations differ in the amount of features placed in $W$ versus $H$. Both use $K = 3$ and $\lambda = 0.001$. (A) Factorization achieved using a sparsity penalty on $H$, with $\lambda_{L1H} = 1$. (B) Factorization achieved using a sparsity penalty on $W$, with $\lambda_{L1W} = 1$.

DOI: https://doi.org/10.7554/eLife.38471.023
Figure 7. Using seqNMF to assess the prevalence of sequences in noisy data. (A) Example simulated datasets. Each dataset contains 10 neurons, with varying amounts of additive noise, and varying proportions of synchronous events versus asynchronous sequences. For the purposes of this figure, ‘sequence’ refers to a sequential pattern with no synchrony between different neurons in the pattern. The duration of each dataset used below is 3000 times, and here 300 timebins are shown. (B) Median percent power explained by convNMF (L = 12; K = 2; λ=0) for each type of dataset (100 examples of each dataset type). Different colors indicate the three different levels of additive noise shown in A. Solid lines and filled circles indicate results on unshuffled datasets. Note that performance is flat for each noise level, regardless of the probability of sequences vs synchronous events. Dotted lines and open circles indicate results on column-shuffled datasets. When no sequences are present, convNMF performs the same on column-shuffled data. However, when sequences are present, convNMF performs worse on column-shuffled data. (C) For datasets with patterns ranging from exclusively synchronous events to exclusively asynchronous sequences, convNMF was used to generate a ‘Sequenciness’ score. Colors correspond to different noise levels shown in A. Asterisks denote cases where the power explained exceeds the Bonferroni-corrected significance threshold generated from column-shuffled datasets. Open circles denote cases that do not achieve significance. Note that this significance test is fairly sensitive, detecting even relatively low presence of sequences, and that the ‘Sequenciness’ score distinguishes between cases where more or less of the dataset consists of sequences.

DOI: https://doi.org/10.7554/eLife.38471.024
Figure 8. Application of seqNMF to extract hippocampal sequences from two rats. (A) Firing rates of 110 neurons recorded in the hippocampus of Rat 1 during an alternating left-right task with a delay period (Pastalkova et al., 2015). The single significant extracted x-ortho penalized factor. Both an x-ortho penalized reconstruction of each factor (left) and raw data (right) are shown. Neurons are sorted according to the latency of their peak activation within the factor. The red line shows the onset and offset of the forced delay periods, during which the animal ran on a treadmill. (B) Firing rates of 43 hippocampal neurons recorded in Rat 2 during the same task (Mizuseki et al., 2013). Neurons are sorted according to the latency of their peak activation within each of the three significant extracted sequences. The first two factors correspond to left and right trials, and the third corresponds to running along the stem of the maze. (C) The diss metric as a function of K for Rat 1. Black line represents the median of the black points. Notice the minimum at K = 1. (D) (Left) Reconstruction (red) and correlation (blue) costs as a function of λ for Rat 1. Arrow indicates λ = 8 × 10^{-3}, used for the x-ortho penalized factorization shown in (A). (E) Histogram of the number of significant factors across 30 runs of x-ortho penalized convNMF. (D) The diss metric as a function of K for Rat 2. Notice the minimum at K = 3. (G–H) Same as in (D–E) but for Rat 2. Arrow indicates λ = 8 × 10^{-3}, used for the factorization shown in (B).

DOI: https://doi.org/10.7554/eLife.38471.025
Figure 9. SeqNMF applied to calcium imaging data from a singing isolate bird reveals abnormal sequence deployment. (A) Functional calcium signals recorded from 75 neurons, unsorted, in a singing isolate bird. (B) Reconstruction and cross-orthogonality cost as a function of $\lambda$. The arrow at $\lambda = 0.005$. 

Figure 9 continued on next page.
indicates the value selected for the rest of the analysis. (C) Number of significant factors for 100 runs with the x-ortho penalty with $K = 10, \lambda = 0.005$. Arrow indicates three is the most common number of significant factors. (D) X-ortho factor exemplars (W’s). Neurons are grouped according to the factor in which they have peak activation, and within each group neurons are sorted by the latency of their peak activation within the factor. (E) The same data shown in (A), after sorting neurons by their latency within each factor as in (D). A spectrogram of the bird’s song is shown at top, with a purple ‘*’ denoting syllable variants correlated with $w_2$. (F) Same as (E), but showing reconstructed data rather than calcium signals. Shown at top are the temporal loadings (H) of each factor.

DOI: https://doi.org/10.7554/eLife.38471.026
Figure 9—figure supplement 1. Further analysis of sequences. (A) For each of the three extracted sequences, examples of song spectrograms triggered at moments where there is a peak in H. Different examples are separated by a red line. Note that each sequence factor corresponds to a particular syllable type. (B) As a function of K, diss measure across all combinations of 10 fits of convNMF. Note the local minima at K = 3. (C) Percent power explained (for convNMF with K = 3 and λ = 0) as a function of L. Note the bend that truncates at approximately 0.25 s, corresponding to a typical syllable duration.

DOI: https://doi.org/10.7554/eLife.38471.027
Figure 9—figure supplement 2. Events-based and parts-based factorizations of songbird data. Illustration of a trade-off between parts-based (W is more strictly orthogonal) and events-based (H is more strictly orthogonal) factorizations in a dataset where some neurons are shared between different conditions.
sequences. The same data as in Figure 9 is factorized with an orthogonality cost just on $\mathbf{H}$ (A, events-based), or just on $\mathbf{W}$ (B, parts-based). Below each motivating cartoon factorization, we show x-ortho penalized convNMF fits ($\mathbf{W}$ and $\mathbf{H}$ together with the reconstruction) of the data in Figure 9. The right panels contain the raw data sorted according to these factorizations. Favoring events-based or parts-based factorizations is a matter of preference. Parts-based factorizations are particularly useful for separating neurons into ensembles. Events-based factorizations are particularly useful for identifying what neural events occur when.

DOI: https://doi.org/10.7554/eLife.38471.028
Figure 10. SeqNMF applied to song spectrograms. (A) Spectrogram of juvenile song, with hand-labeled syllable types (Okubo et al., 2015). (B) Reconstruction cost and x-ortho cost for these data as a function of $\lambda$. Arrow denotes $\lambda = 0.0003$, which was used to run convNMF with the x-ortho penalty (C) W’s for this song, fit with $K = 8$, $L = 200ms$, $\lambda = 0.0003$. Note that there are three non-empty factors, corresponding to the three hand-labeled syllables a, b, and c. (D) X-ortho penalized H’s (for the three non-empty factors) and reconstruction of the song shown in (A) using these factors.

DOI: https://doi.org/10.7554/eLife.38471.029
Appendix 2—figure 1. Example of redundancy with three factors. In addition to the direct overlap of \( h_2 \) and \( h_3 \), and of \( w_1 \) and \( w_3 \), there is a ‘triplet’ penalty \( R_{12} \) on factors 1 and 2 that occurs because each has an overlap (in either \( W \) or \( H \)) with the 3rd factor (\( w_3 \), \( h_3 \)). This occurs even though neither \( w_1 \) and \( w_2 \), nor \( h_1 \) and \( h_2 \), are themselves overlapping.

DOI: https://doi.org/10.7554/eLife.38471.034