

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

Emily L. Mackevicius^{1†}, Andrew H. Bahle^{1†}, Alex H. Williams², Shijie Gu^{1,3}, Natalia I. Denissenko¹, Mark S. Goldman^{4*}, Michale S. Fee^{1*}

*For correspondence:

fee@mit.edu (MSF);
msgoldman@ucdavis.edu
(MSG)

†These authors contributed
equally to this work

¹McGovern Institute for Brain Research, Department of Brain and Cognitive Sciences, MIT; ²Stanford Neuroscience Institute; ³School of Life Sciences and Technology, ShanghaiTech University; ⁴Center for Neuroscience, Physiology and Behavior, UC Davis

1

Abstract

2 The ability to identify interpretable, low-dimensional features that capture the dynamics
3 of large-scale neural recordings is a major challenge in neuroscience. Dynamics that
4 include repeated temporal patterns (which we call sequences), are not succinctly
5 captured by traditional dimensionality reduction techniques such as principal
6 components analysis (PCA) and non-negative matrix factorization (NMF). The presence of
7 neural sequences is commonly demonstrated using visual display of trial-averaged firing
8 rates [15, 32, 19]. However, the field suffers from a lack of task-independent,
9 unsupervised tools for consistently identifying sequences directly from neural data, and
10 cross-validating these sequences on held-out data. We propose a tool that extends a
11 convolutional NMF technique to prevent its common failure modes. Our method, which
12 we call seqNMF, provides a framework for extracting sequences from a dataset, and is
13 easily cross-validated to assess the significance of each extracted factor. We apply
14 seqNMF to recover sequences in both a previously published dataset from rat
15 hippocampus, as well as a new dataset from the songbird pre-motor area, HVC. In the
16 hippocampal data, our algorithm automatically identifies neural sequences that match
17 those calculated manually by reference to behavioral events [15, 32]. The second data set
18 was recorded in birds that never heard a tutor, and therefore sang pathologically variable
19 songs. Despite this variable behavior, seqNMF is able to discover stereotyped neural
20 sequences. These sequences are deployed in an overlapping and disorganized manner,
21 strikingly different from what is seen in tutored birds. Thus, by identifying temporal
22 structure directly from neural data, seqNMF can enable dissection of complex neural
23 circuits with noisy or changing behavioral readouts.
24

25

26 Introduction

27 The ability to detect and analyze temporal sequences embedded in a complex sensory
28 stream is an essential cognitive function, and as such is a necessary capability of neuronal
29 circuits in the brain [10, 23, 3, 21], as well as artificial intelligence systems [11, 42]. The
30 detection and characterization of temporal structure in signals is also useful for the
31 analysis of many forms of physical and biological data. In neuroscience, recent advances
32 in technology for electrophysiological and optical measurements of neural activity have
33 enabled the recording of hundreds or thousands of neurons [6, 26, 38, 24], in which
34 neuronal dynamics are often structured in sparse sequences [18, 19, 31, 32].

35 While sequential patterns are simple to conceptualize, identifying these patterns in
36 high-dimensional datasets is surprisingly challenging. Traditional techniques for identi-
37 fying low dimensional structure in high dimensional datasets such as PCA and NMF do
38 not work for sequences, because those methods only model zero-time-lag correlations in
39 data. It is sometimes possible to identify neural sequences by heuristically aggregating
40 pairwise cross-correlations across neurons or across timebins [37, 17], but these correla-
41 tions are easily confounded [4], leading to mathematically complex and computationally
42 expensive procedures. In some cases, sequences can be identified by simply averaging
43 across multiple behavioral trials, but this approach requires stereotyped behavior.

44 Of increasing interest is the study of internal dynamics in the brain, without reference
45 to behavior, for example, neural dynamics during learning, sleep, or diseased states. A
46 promising approach for the unsupervised detection of temporal patterns is convolutive
47 matrix factorization (CNMF) [41, 40] (Figure 1), which has primarily been applied to audio
48 signals such as speech [30, 40, 45]. CNMF identifies exemplar patterns in conjunction
49 with the times at which each pattern occurs. This strategy eliminates the need to average
50 activity aligned to any external behavioral variables, and CNMF has recently been used to
51 extract repeated patterns in spontaneous neural activity [34]. While CNMF factorizations
52 produce an excellent reconstruction of the data, this algorithm will find a much larger
53 number of factors than minimally required. Because of this redundancy, there are many
54 different possible factorizations that explain the data equally well, and the algorithm
55 arbitrarily chooses among them each time it is run, producing inconsistent results [34].

56 When describing and interpreting data, the principle of ‘Occam’s razor’, a key scientific
57 doctrine, tells us to prefer minimal models. In this paper, we describe a modification of the
58 CNMF algorithm that penalizes redundant factors, biasing the results toward factorizations
59 with the smallest number of factors and providing a simple explanation of the data. We
60 do this by incorporating a regularization term into the CNMF cost function. Unlike other
61 common approaches [20] such as sparsity regularization [47, 30, 36] that constrain the
62 make-up of each factor, our regularization penalizes the correlations between factors that
63 result from redundant factorizations. We build on earlier applications of soft-orthogonality
64 constraints to NMF [7] to capture the types of temporally offset correlations that may
65 occur in the convolutional case.

66 Our algorithm, which we call seqNMF, produces minimal and consistent factorizations
67 in synthetic data under a variety of noise conditions, with high similarity to ground-truth
68 sequences. We further tested seqNMF on hippocampal spiking data in which neural
69 sequences have previously been described. Finally, we use seqNMF to extract sequences
70 in a functional calcium imaging dataset recorded in vocal/motor cortex of untutored
71 songbirds that sing pathologically variable songs. We found that repeatable neural

72 sequences are activated in an atypical and overlapping fashion, suggesting potential
 73 neural mechanisms for this pathological song variability.

74 Results

75 Matrix factorization framework for unsupervised discovery of fea- 76 tures in neural data

77 Matrix factorization underlies many well known unsupervised learning algorithms [44]
 78 with applications to neuroscience [12], including principal component analysis (PCA) [33],
 79 non-negative matrix factorization (NMF) [27], dictionary learning, and k-means clustering.
 80 We start with a data matrix, \mathbf{X} , containing the activity of N neurons at T times. If the
 81 neurons exhibit a single repeated pattern of synchronous activity, the entire data matrix
 82 can be reconstructed using a column vector \mathbf{w} representing the neural pattern, and a row
 83 vector \mathbf{h} representing the times at which that pattern occurs (temporal loadings). In this
 84 case, the data matrix \mathbf{X} is mathematically reconstructed as the outer product of these two
 85 vectors ($\tilde{\mathbf{X}}_{nt} = w_n h_t$). If multiple patterns are present in the data, then each pattern can be
 86 reconstructed by a separate outer product, where the reconstructions are summed to
 87 approximate the entire data matrix (Figure 1A) as follows:

$$\mathbf{X}_{nt} \approx \tilde{\mathbf{X}}_{nt} = \sum_{k=1}^K w_{nk} h_{kt} = (\mathbf{WH})_{nt} \quad (1)$$

88 Here, in order to store K different patterns, \mathbf{W} is a $N \times K$ matrix containing the K
 89 exemplar patterns, and \mathbf{H} is a $K \times T$ matrix containing the K timecourses:

$$\mathbf{W} = \begin{bmatrix} | & | & & \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \\ | & | & & \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} - & \mathbf{h}_1 & - \\ - & \mathbf{h}_2 & - \\ \vdots & & \end{bmatrix} \quad (2)$$

90 Given a data matrix with unknown patterns, the goal of these unsupervised learning
 91 algorithms is to discover a small set of patterns (\mathbf{W}) and a corresponding vector of
 92 temporal loadings (\mathbf{H}) that approximate the data. This corresponds to a dimensionality
 93 reduction, whereby the data is expressed in more compact form ($K < N, T$). NMF
 94 additionally requires that \mathbf{W} and \mathbf{H} must contain only positive numbers. The discovery
 95 of unknown factors is often accomplished by minimizing the following cost function,
 96 which measures (using the Frobenius norm) the sum of all squared errors between the
 97 reconstruction $\tilde{\mathbf{X}} = \mathbf{WH}$ and the original data matrix \mathbf{X} :

$$(\tilde{\mathbf{W}}, \tilde{\mathbf{H}}) = \arg \min_{\mathbf{W}, \mathbf{H}} \|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 \quad (3)$$

98 While this general strategy works well for extracting synchronous activity, it is un-
 99 suitable for discovering temporally extended patterns—first, because each element in
 100 a sequence must be represented by a different factor, and second, because NMF as-
 101 sumes that the columns of the data matrix are independent ‘samples’ of the data, so
 102 permutations in time have no effect on the factorization of a given dataset. It is therefore
 103 necessary to adopt a different strategy for temporally extended features.

104 *Convolutional non-negative matrix factorization (CNMF)*

105 Convolutional NMF (CNMF) [41, 40] extends NMF to provide a framework for extracting
 106 temporal patterns and sequences from data. While classical NMF represents each pattern
 107 as a single vector (Figure 1A), CNMF explicitly represents an exemplar pattern of neural
 108 activity over a brief period of time; the pattern is stored as an $N \times L$ matrix, where each
 109 column (indexed by $\ell = 1$ to L) indicates the activity of neurons at different timelags
 110 within the pattern (Figure 1B, where we call this matrix pattern \mathbf{w}_1 for analogy with NMF).
 111 The times at which this pattern/sequence occurs are stored using timeseries vector \mathbf{h}_1 ,
 112 as for NMF. The reconstruction is produced by convolving the $N \times L$ pattern with the
 113 timeseries \mathbf{h}_1 (Figure 1B).

114 If the dataset contains multiple patterns, each pattern is captured by a different $N \times L$
 115 matrix and a different associated timeseries vector \mathbf{h} . A collection of K different patterns
 116 can be compiled together into an $N \times K \times L$ tensor \mathbf{W} and a corresponding $K \times T$ timeseries
 117 matrix \mathbf{H} . Analogously to NMF, CNMF reconstructs the data as a sum of K convolutions
 118 between each neural activity pattern (\mathbf{W}), and its corresponding temporal loadings (\mathbf{H}):

$$\mathbf{X}_{nt} \approx \tilde{\mathbf{X}}_{nt} = \sum_k \sum_{\ell} w_{nk\ell} h_{k(t-\ell)} = (\mathbf{W} \circledast \mathbf{H})_{nt} \quad (4)$$

119 where the tensor/matrix convolution operator \circledast (notation summary, Table 1) reduces to
 120 matrix multiplication in the $L = 1$ case, which is equivalent to standard NMF. The quality
 121 of this reconstruction can be measured using the same cost function shown in Equation
 122 3, and \mathbf{W} and \mathbf{H} may be found iteratively using the same multiplicative gradient descent
 123 updates often used for standard NMF [27, 41, 40].

124 While CNMF can perform extremely well at reconstructing sequential structure, it
 125 suffers from a significant problem—namely, it reconstructs data using many more factors
 126 than are minimally required. This is because an individual temporal pattern may be
 127 approximated equally well by a single pattern or by a linear combination of multiple
 128 sub-patterns. A related problem is that running the CNMF algorithm from different
 129 random initial conditions produces inconsistent results, finding different combinations of
 130 sub-patterns on each run [34]. These inconsistency errors fall into three main categories
 131 (Figure 1C):

- 132 • *Type 1:* Two or more factors are used to reconstruct the same instances of a se-
 133 quence.
- 134 • *Type 2:* Two or more factors are used to reconstruct temporally different parts of
 135 the same sequence, for instance the first half and the second half.
- 136 • *Type 3:* Identical factors are used to reconstruct different instances of a sequence.

137 Together, these failure modes manifest as strong correlations between different redun-
 138 dant factors, as seen in the similarity of their temporal loadings (\mathbf{H}) and of their exemplar
 139 activity patterns (\mathbf{W}).

140 *SeqNMF: A regularized convolutional non-negative matrix factorization*

141 Regularization is a common technique in optimization that allows the incorporation
 142 of constraints or additional information with the goal of improving generalization or
 143 simplifying solutions [20]. To reduce the occurrence of redundant factors (and inconsistent
 144 factorizations) in CNMF, we sought a principled way of penalizing the correlations between

145 factors by introducing a regularization term into the CNMF cost function of the following
 146 form:

$$(\tilde{\mathbf{W}}, \tilde{\mathbf{H}}) = \arg \min_{\mathbf{W}, \mathbf{H}} \left(\|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 + \mathcal{R} \right) \quad (5)$$

147 In the next section, we will motivate a novel cost function that effectively minimizes the
 148 number of factors by penalizing temporal correlations between different factors. We
 149 will build up the full cost function by addressing, one at a time, the types of correlations
 150 generated by each failure mode.

151 Regularization has previously been used in NMF to address the problem of duplicated
 152 factors, which, similar to Type 1 errors above, present as correlations between the \mathbf{H} 's
 153 [7]. Such correlations are measured by computing the correlation matrix $\mathbf{H}\mathbf{H}^T$, which
 154 contains the correlations between the temporal loadings of every pair of factors. The
 155 regularization may be implemented using the cost term $\mathcal{R} = \lambda \|\mathbf{H}\mathbf{H}^T\|_{1, i \neq j}$. The norm
 156 $\|\cdot\|_{1, i \neq j}$ sums the absolute value of every matrix entry except the diagonal (notation
 157 summary, Table 1) so that correlations between different factors are penalized, while the
 158 obvious correlation of each factor with itself is not. Thus, during the minimization process,
 159 similar factors compete, and a larger factor drives down the \mathbf{H} of a correlated smaller
 160 factor. The parameter λ is controls the magnitude of the regularization term \mathcal{R} .

161 In CNMF, a regularization term based on $\mathbf{H}\mathbf{H}^T$ yields an effective method to prevent
 162 errors of Type 1, because it penalizes the associated zero lag correlations. However, it does
 163 not prevent errors of the other types, which exhibit different types of correlations. For
 164 example Type 2 errors result in correlated temporal loadings that have a small temporal
 165 offset and thus are not detected by $\mathbf{H}\mathbf{H}^T$. To address this problem, we smoothed the \mathbf{H} 's in
 166 the regularization term with a square window of length $2L-1$ using the smoothing matrix \mathbf{S}
 167 ($s_{ij} = 1$ when $|i-j| < L$ and otherwise $s_{ij} = 0$). The resulting regularization, $\mathcal{R} = \lambda \|\mathbf{H}\mathbf{S}\mathbf{H}^T\|$,
 168 allows factors with small temporal offsets to compete, effectively preventing errors of
 169 Type 1 and 2.

170 Unfortunately this regularization does not prevent errors of Type 3, in which redundant
 171 factors with highly similar patterns in \mathbf{W} are used to explain different instances of the
 172 same sequence. Such factors have temporal loadings that are segregated in time, and
 173 thus have low correlations, to which the cost term $\|\mathbf{H}\mathbf{S}\mathbf{H}^T\|$ is insensitive. One way to
 174 resolve errors of Type 3 might be to include an additional cost term that penalizes the
 175 similarity of the factor patterns in \mathbf{W} . A challenge with this approach is that, in the CNMF
 176 framework, there is no constraint on temporal translations of the sequence within \mathbf{W} . For
 177 example, if two redundant factors containing identical sequences that are simply offset by
 178 one timebin (in the L dimension), then these patterns would have zero correlation. Such
 179 offsets might be accounted for by smoothing the \mathbf{W} matrices in time before computing
 180 the correlation (Table 2), analogous to $\|\mathbf{H}\mathbf{S}\mathbf{H}^T\|$. The general approach of adding an
 181 additional cost term for \mathbf{W} correlations has the disadvantage that it requires setting an
 182 extra parameter, namely the λ associated with this cost.

183 Thus, we chose an alternative approach to resolve errors of Type 3 that simultaneously
 184 detects correlations in \mathbf{W} and \mathbf{H} using a single cost term. We note that redundant factors of
 185 this type have a high degree of overlap with the data at the same times, even though their
 186 temporal loadings are segregated at different times. To introduce competition between
 187 these factors, we compute the pairwise correlation between the temporal loading of each
 188 factor and the overlap of every other factor with the data, given by $\mathbf{W} \otimes \mathbf{X}_{i \neq j}^T$ (notation

189 summary, Table 1). The regularization then sums up these correlations across all pairs of
 190 factors, implemented as follows:

$$\mathcal{R} = \lambda \|\mathbf{W} \circledast \mathbf{XSH}^T\|_{1,i \neq j} \quad (6)$$

191 When incorporated into the update rules, this causes any factor that has a high overlap
 192 with the data to suppress the temporal loading (\mathbf{H}) of any other factors active at that time.
 193 Thus, factors compete to explain each feature of the data, favoring solutions that use a
 194 minimal set of factors to give a good reconstruction. We refer to this minimal set as an
 195 efficient factorization. The resulting global cost function is:

$$(\tilde{\mathbf{W}}, \tilde{\mathbf{H}}) = \arg \min_{\mathbf{W}, \mathbf{H}} \left(\|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 + \lambda \|\mathbf{W} \circledast \mathbf{XSH}^T\|_{1,i \neq j} \right) \quad (7)$$

196 The update rules for \mathbf{W} and \mathbf{H} are based on the derivatives of this global cost function,
 197 leading to a simple modification of the standard multiplicative update rules used for NMF
 198 and CNMF [27, 41, 40] (Table 2).

199 **Testing the performance of seqNMF on simulated sequences**

200 To compare the performance of seqNMF to unregularized CNMF, we simulated neural
 201 sequences of a sort commonly encountered in neuronal data (Figure 2A). The simulated
 202 data were used to test several aspects of the seqNMF algorithm: consistency of factoriza-
 203 tions, the ability of the algorithm to discover the correct number of sequences in the data,
 204 and robustness to noise.

205 *Consistency of seqNMF factorization*

206 We set out to determine if seqNMF exhibits the desirable property of consistency—namely
 207 whether it returns similar sequences each time it is run on the same dataset using different
 208 random initializations of \mathbf{W} and \mathbf{H} . Consistency was assessed as the extent to which there
 209 is a good one-to-one match between factors across different runs (Methods 10). Due
 210 to the inefficiencies outlined in Figure 1, CNMF yielded low consistency scores typically
 211 ranging from 0.2 to 0.4 on a scale from zero to one. In contrast, seqNMF factorizations
 212 were nearly identical across different fits of noiseless data, producing consistency scores
 213 that were always higher than any we measured for CNMF, and typically (>80% of the time)
 214 higher than 0.99 (Figure 2B). Both CNMF and seqNMF had near perfect reconstruction
 215 error for all combinations of K and L that exceed the number and duration of sequences
 216 in the data (not shown). However, CNMF exhibited low consistency scores, a problem
 217 that was further exacerbated for larger values of K . In contrast, seqNMF exhibited high
 218 consistency scores across a wide range of values of both K and L .

219 We also tested the consistency of seqNMF factorizations for the interesting case in
 220 which a population of neurons is active in multiple different sequences. In fact neurons
 221 that are shared across different sequences have been observed in several different neu-
 222 ronal datasets [31, 32, 19]. For one test, we constructed two sequences in which shared
 223 neurons were active at a common pattern of latencies in both sequences; in another test,
 224 shared neurons were active in a different pattern of latencies in each sequence. In both
 225 tests, seqNMF achieved near-perfect reconstruction error, and consistency was similar to
 226 the case with no shared neurons (Figure 2).

227 *Cross-validating to assess the statistical significance of sequences*

228 SeqNMF allows a simple procedure for assessing the statistical significance of each
229 extracted sequence. Candidate sequences are extracted by applying SeqNMF to a subset
230 of the data; the significance of each candidate sequence is then assessed on separate
231 held-out data. If an extracted sequence corresponds to a real sequence present in the
232 data, then the overlap of that factor with the held-out data ($\mathbf{W} \otimes \mathbf{X}$) will have large values at
233 the times at which the sequence occurs (relative to other times). The resulting abundance
234 of high overlap values will create a distribution of overlaps with high skewness compared
235 to a null distribution. In contrast, a candidate sequence that does not reliably occur in the
236 held-out data will have a smaller number of high overlaps, and a distribution of overlaps
237 with lower skewness. We compare the skewness of the actual distribution of overlaps
238 with that of distributions generated from null factors to determine the significance of
239 each candidate sequence (Figure S1, Methods 10). Null factors were created by random
240 circular shifts in time lag, along the L dimension, of the pattern matrices \mathbf{W} .

241 Runs of seqNMF on simulated and real data have revealed that the algorithm produces
242 two types of factors that can be immediately ruled out as candidate sequences: 1)
243 empty factors with zero amplitude in all neurons at all lags and 2) factors that have
244 amplitude in only one neuron. The latter case occurs often in datasets where one neuron
245 is substantially more active than other neurons, and thus accounts for a large amount
246 of variance in the data. SeqNMF also occasionally generates factors that appear to
247 capture one moment in the test data, especially in short datasets, where this can account
248 for a substantial fraction of the data variance. Such sequences are easily identified as
249 non-significant when tested on held-out data using the skewness test.

250 Note that if λ is set too small, seqNMF will produce multiple redundant factors to
251 explain one sequence in the data. In this case, each redundant candidate sequence will
252 pass the significance test outlined here. We will address below a procedure for choosing
253 λ and methods for determining the number of sequences.

254 *Estimating the number of sequences in a dataset*

255 A successful factorization should contain the same number of significant factors as exist
256 sequences in the data. To compare the ability of seqNMF and CNMF to recover the true
257 number of patterns in a dataset, we generated simulated data containing between 1
258 and 10 different sequences. We then ran many independent fits of these data, using
259 both seqNMF and CNMF, and measured the number of significant factors. We found that
260 CNMF overestimates the number of sequences in the data, returning K significant factors
261 on nearly every run. In contrast, seqNMF tends to return a number of significant factors
262 (N_{sig}) that closely matches the actual number of sequences (N_{seq}). The standard deviation
263 of the error ($N_{seq} - N_{sig}$) tended to grow linearly with the actual number of sequences
264 (Figure 2C).

265 *Robustness to noisy and challenging data*

266 Having established that seqNMF can produce both consistent and efficient factorizations
267 of noiseless synthetic data, we next probed the capacity of seqNMF to detect sequences
268 in the presence of common types of noise. These included: participation noise, in which
269 individual neurons participate probabilistically in instances of a sequence; additive noise,
270 in which neuronal events occur randomly outside of normal sequence patterns; temporal

271 jitter, in which the timing of individual neurons is shifted relative to their typical time in a
272 sequence; and finally, temporal warping, in which each instance of the sequence occurs
273 at a different randomly selected speed.

274 To test the robustness of seqNMF to each of these noise conditions, we factorized data
275 containing two neural sequences at variety of noise levels. The value of λ was chosen using
276 methods described in the next section. SeqNMF proved relatively robust to all four noise
277 types, as measured by the similarity of the factors to the ground-truth. We defined the
278 ground-truth sequences those used to generate the synthetic data prior to the addition
279 of noise. We then quantified the correlation between seqNMF factors and ground-truth
280 sequences (Methods section 10, Figure 3). For low noise conditions, seqNMF produced
281 factors that were highly similar to ground-truth; this similarity gracefully declined as
282 noise increased. Visualization of the extracted factors revealed that they tend to match
283 ground-truth sequences even in the presence of high noise (Figure 3). Together, these
284 findings suggest that seqNMF is suitable for extracting sequence patterns from neural
285 data with realistic forms of noise.

286 *Method for choosing an appropriate value of λ*

287 In general, the seqNMF algorithm performs differently using different values of λ , and
288 application to the noisy datasets revealed that the optimal choice of this parameter may
289 depend on the degree and type of noise contamination. Choosing λ involves a trade
290 off between reconstruction accuracy and the efficiency and consistency of the resulting
291 factorizations (Figure 4). Indeed, perfect reconstruction is no longer a goal in noisy data,
292 since it would imply fitting all of the noise as well as the signal. Rather, the goal is to
293 reconstruct only the repeating temporal patterns in the data and to do so with an efficient,
294 maximally uncorrelated set of factors. For any given factorization, the reconstruction
295 error may be estimated as $\|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2$, and the efficiency may be estimated using the
296 seqNMF regularization term ($\|\mathbf{W} \oplus \mathbf{XSH}^T\|_{1,i \neq j}$) which we refer to as correlation cost.

297 We have developed a quantitative strategy to guide the choice of λ , by analyzing the
298 dependence on λ of both reconstruction error and correlation cost in synthetic datasets
299 containing two sequences (Figure 4). SeqNMF was run with many random initializations
300 over a range of λ spanning six orders of magnitude. For small λ , the behavior of seqNMF
301 approaches that of CNMF, producing a large number of redundant factors with high
302 correlation cost. In the regime of small λ , correlation cost saturates at a large value and
303 reconstruction error saturates at a minimum value (Figure 4A). At the opposite extreme,
304 in the limit of large λ , seqNMF returns a single significant factor with zero correlation cost
305 because all other factors have been suppressed to zero amplitude. In this limit, the single
306 factor is unable to reconstruct multi-sequence data, resulting in large reconstruction error.
307 Between these extremes, there exists a region in which increasing λ produces a rapidly
308 increasing reconstruction error and a rapidly decreasing correlation cost. Following the
309 intuition that the optimal choice of λ for seqNMF would lie in this cross-over region
310 where the costs are balanced, we set out to quantitatively identify, for known synthetic
311 sequences, the optimal λ at which seqNMF has the highest probability of recovering the
312 correct number of significant factors, and at which these factors most closely match the
313 ground truth sequences.

314 The following procedure was implemented: For a given dataset, seqNMF is run several
315 times at a range of values of λ , and terminal reconstruction cost and correlation cost

316 are recorded. These costs are normalized to vary between 0 and 1, and the value of λ
317 at which the reconstruction and correlation cost curves intersect is determined (Figure
318 4). This intersection point, λ_0 , then serves as a precise reference by which to determine
319 the correct choice of λ . We then separately calibrated the reference λ_0 to the λ 's that
320 performed well in synthetic datasets, with and without noise, for which the ground-truth
321 is known. This analysis revealed that values of λ between λ_0 and $5\lambda_0$ performed well
322 across different noise types and levels (Figure 4B,C). For additive noise, performance was
323 better when λ was chosen to be near λ_0 , while with other noise types, performance was
324 better at higher ($\approx 5\lambda_0$). Note that this procedure does not need to be run on every
325 dataset analyzed, rather, only when seqNMF is applied to a new type of data for which a
326 reasonable range of λ is not already known.

327 Sometimes there is not a clear correct answer for how many sequences exist in a
328 dataset. In fact, different values of λ can lead to different sensible factorizations. It can
329 be useful to explore the factorization for different values of λ between λ_0 and $10\lambda_0$. We
330 observed a notable example of this in datasets that included sequences with a high
331 degree of temporal warping. In this case, high λ led seqNMF to extract a single factor for
332 each ground truth sequence. In contrast, at low λ seqNMF extracted multiple factors for
333 each ground truth sequence, corresponding to slow and fast variations of the sequence.
334 Thus, seqNMF clusters sequences with different granularity depending on the strength of
335 the regularization term λ .

336 *Adding additional sparsity regularization to seqNMF*

337 Sparsity regularization is a widely used strategy for achieving more interpretable results
338 across a variety of algorithms and datasets [47], including CNMF [30, 36]. In some of
339 our datasets, we found it useful to add $L1$ regularization for sparsity, in addition to
340 regularizing for factor competition. The multiplicative update rules for these variants are
341 included in Table 2, and as part of our code package. Sparsity on the matrices \mathbf{W} and
342 \mathbf{H} may particularly useful in cases when sequences are repeated rhythmically (Figure
343 S2). For example, the addition of a sparsity regularizer on the \mathbf{W} update will bias the \mathbf{W}
344 exemplars to include only a single repetition of the repeated sequence, while the addition
345 of a sparsity regularizer on \mathbf{H} will bias the \mathbf{W} exemplars to include multiple repetitions of
346 the repeated sequence. This gives one fine control over how much structure in the signal
347 to pack into \mathbf{W} versus \mathbf{H} . Of course, these are both equally valid interpretations of the
348 data, but each may be more useful in different contexts.

349 *Further considerations of shared neurons*

350 The existence of neurons that are shared between different sequences raises an inter-
351 esting ambiguity in the types of factorizations that seqNMF can produce, an example of
352 which is illustrated in Figure S3. In this case, there are two different, but equally valid, fac-
353 torizations: in one factorization, there are two types of events, one in which a population
354 of neurons generates a sequence by itself, and another in which a second population
355 of neurons is also simultaneously active. In another factorization, these same data are
356 interpreted by seqNMF as two different populations of neurons that are sometimes active
357 separately and sometimes active together. Note that these two factorizations produce
358 very different correlations between the factors. In the first, 'events-based' factorization,
359 the \mathbf{H} s are orthogonal (uncorrelated) while the \mathbf{W} s have high overlap. In the second,
360 'parts-based' factorization, the \mathbf{W} s are orthogonal while the \mathbf{H} s are strongly correlated.

361 We have found that seqNMF will produce both types of factorizations depending on
362 initial conditions and the structure of shared neurons in the data. We note that these
363 different factorizations may correspond to different intuitions about underlying mech-
364 anisms. Therefore, it may be useful to explicitly bias the probability of these different
365 factorizations by the addition of further regularization on either \mathbf{W} or \mathbf{H} correlations, as
366 demonstrated in Figure S3. Update rules to implement both of these regularizations are
367 derived in Appendix 1, and shown in Table 2, and included as options in our code.

368 **Application of seqNMF to hippocampal sequences**

369 To test the ability of seqNMF to discover patterns in electrophysiological data, we ana-
370 lyzed the activity of a set of simultaneously recorded hippocampal neurons in a publicly
371 available dataset in which sequences have previously been reported [32]. In these experi-
372 ments, rats were trained to alternate between left and right turns in a T-maze to earn a
373 water reward. Between alternations, the rats ran on a running wheel during an imposed
374 delay period lasting either 10 or 20 seconds. By averaging spiking activity during the delay
375 period, the authors reported long temporal sequences of neural activity spanning the
376 delay. In some rats, the same sequence occurred on left and right trials, while in other
377 rats, different sequences were active in the delay period during the different trial types.

378 Without reference to the behavioral landmarks, seqNMF was able to extract different
379 types of sequences in two different rats. The automated method described above was
380 used to choose λ (Figure 5). In Rat 1, a single significant factor was extracted, corre-
381 sponding to a sequence active throughout the running wheel delay period (Figure 5B).
382 In Rat 2, three significant factors were identified (Figure 5C). The first two corresponded
383 to distinct sequences active for the duration of the delay period on alternating trials.
384 The third sequence was active immediately following each of the alternating sequences,
385 corresponding to the time at which the animal exits the wheel and runs up the stem
386 of the maze. Taken together, these results suggest that seqNMF can detect multiple
387 neural sequences without the use of any behavioral landmarks. Having validated this
388 functionality in both simulated data and previously published neural sequences, we then
389 applied seqNMF to find structure in a novel dataset, in which the ground truth is unknown,
390 and difficult to ascertain using previous methods.

391 **Application of seqNMF to abnormal sequence development in avian 392 motor cortex**

393 We applied seqNMF to analyze new functional imaging data recorded in songbird HVC
394 during singing. Normal adult birds sing a highly stereotyped song, making it possible to
395 detect sequences by averaging neural activity aligned to the song. Using this approach, it
396 has been shown that HVC neurons generate precisely timed sequences that tile each song
397 syllable [18, 35, 29]. In contrast to adult birds, young birds sing highly variable babbling
398 vocalizations, known as subsong, for which HVC is not necessary [1]. The emergence of
399 sequences in HVC occurs gradually over development, as the song matures from subsong
400 to adult song [31].

401 Songbirds learn their song by imitation and must hear a tutor to develop normal adult
402 vocalizations. Birds isolated from a tutor sing highly variable and abnormal songs as
403 adults [14]. Such 'isolate' birds provide an opportunity to study how the absence of normal
404 auditory experience leads to pathological vocal/motor development. However, the high

405 variability of pathological ‘isolate’ song makes it difficult to identify neural sequences
406 using the standard approach of aligning neural activity to vocal output.

407 Using seqNMF, we were able to identify repeating neural sequences in isolate song-
408 birds (Figure 6A). We found that the HVC network generates several distinct premotor
409 sequences (Figure 6B-C), including sequences deployed during syllables of abnormally
410 long and variable durations (Figure 6D-F).

411 In addition, the extracted sequences exhibit properties not observed in normal adult
412 birds. We see an example of two distinct sequences that sometimes, but not always,
413 co-occur (Figure 6). We observe that a short sequence occurs alone on some syllable
414 renditions, while on other syllable renditions, a second longer sequences is generated
415 simultaneously. This probabilistic overlap of different sequences is highly atypical in nor-
416 mal adult birds [18, 28, 35, 29]. Furthermore, this pattern of neural activity is associated
417 with abnormal variations in syllable structure—in this case resulting in a longer variant
418 of the syllable when both sequences co-occur. This acoustic variation is a characteristic
419 pathology of isolate song [14]. Thus, even though we observe HVC generating some
420 sequences in the absence of a tutor, it appears that these sequences are deployed in a
421 highly abnormal fashion.

422 **Application of seqNMF to a behavioral dataset: song spectrograms**

423 Although we have focused on the application of seqNMF to neural activity data, this
424 method naturally extends to other types of high-dimensional datasets, including behav-
425 ioral data with applications to neuroscience. The neural mechanisms underlying song
426 production and learning in songbirds is an area of active research. However, the identifi-
427 cation and labeling of song syllables in acoustic recordings is challenging, particularly in
428 young birds where song syllables are highly variable. Because automatic segmentation
429 and clustering often fail, song syllables are still routinely labelled by hand [31]. We tested
430 whether seqNMF, applied to a spectrographic representation of zebra finch vocalizations,
431 is able to extract meaningful features in behavioral data. SeqNMF correctly identified
432 repeated acoustic patterns in juvenile songs, placing each distinct syllable type into a
433 different factor (Figure 7). The resulting classifications agree with previously published
434 hand-labeled syllable types [31]. A similar approach could be applied to other behavioral
435 data, for example movement data or human speech, and could facilitate the study of
436 neural mechanisms underlying even earlier and more variable stages of learning.

437 **Discussion**

438 As neuroscientists strive to record larger datasets, there is a need for rigorous new
439 tools to reveal underlying structure in high-dimensional data [16, 39, 8, 5]. In particular,
440 sequential structure is increasingly regarded as a fundamental property of neuronal
441 circuits [18, 19, 31, 32], but tools for extracting such structure in neuronal data have
442 been lacking. While convolutional NMF provides a promising framework for extracting
443 sequential structure in high-dimensional datasets, it suffers from a number of weaknesses:
444 It is highly unconstrained, producing many redundant factors that provide a large number
445 of factorizations with equally low reconstruction error. Others have approached the
446 problem of achieving a minimal set of factors by running unregularized CNMF many times
447 from different initial conditions and identifying a subset of the resultant factors that are
448 most reliably produced [34]. Our approach has been to construct a regularizer that, when

449 incorporated into the multiplicative update rules, drives competition between factors and
450 produces highly consistent factorizations.

451 While seqNMF regularization is particularly useful when the number of sequences in
452 the data is not known *a priori*, seqNMF does more than simply minimize the number of
453 factors. Even in the context of a minimal set of factors, there are often several different
454 reasonable factorizations. SeqNMF provides a framework for biasing factorizations in a
455 principled way between alternative interpretations of the data. For example, the choice of
456 λ can control the granularity of the clustering of sequences into different factors. At high λ ,
457 seqNMF tends to combine similar sequences into a single factor, while at lower λ it tends
458 to place different variants of a sequence into different factors, as shown for the case of
459 temporally warped sequences. As another example, addition of a sparseness regularizer
460 can be used to control the trade off placing features in the pattern exemplars or in the
461 temporal loadings. Similarly, we have found that by including additional orthogonality
462 constraints on \mathbf{W} and \mathbf{H} , one can bias factorizations toward parts-based or events-based
463 factorizations, respectively.

464 While seqNMF is generally quite robust, proper preprocessing of the data can be
465 important to obtaining reasonable factorizations. A key principle is that, in minimizing the
466 reconstruction error, seqNMF is most strongly influenced by parts of the data that exhibit
467 high variance. This can be problematic if the regions of interest in the data have relatively
468 low amplitude. For example, high firing rate neurons may be prioritized over those
469 with lower firing rate. Additionally, variations in behavioral state may lead to seqNMF
470 factorizations that prioritize regions of the data with high variance and neglect other
471 regions. It may be possible to mitigate these effects by normalizing data, or by restricting
472 analysis to particular subsets of the data, either by time or by neuron.

473 SeqNMF addresses a key challenge in extracting neural sequences in complex animal
474 behaviors. Prior analysis methods required aligning neural activity to behavioral events,
475 such as animal position for the case of hippocampal and cortical sequences [19, 32], or
476 vocal output for the case of songbird vocalizations [31]. But this method is not ideally
477 suited for the case highly variable behaviors, such as in early learning and development
478 [31], either normal or abnormal. For example, by applying seqNMF, we were able to
479 identify neural sequences underlying a pathologically variable vocal behavior in the
480 songbird. This technique should enable similar approaches in other cases, expanding
481 the repertoire of behaviors available to neuroscience from those that are repeated and
482 stereotyped to include those that may be variable and rapidly changing.

483 Acknowledgements

484 This work was supported by the National Institutes of Health [grant number R01 DC009183],
485 The G. Harold & Leila Y. Mathers Charitable Foundation, and the Simons Collaboration for
486 the Global Brain. ELM received support through the NDSEG Fellowship program. AHW
487 received support from the U.S. Department of Energy Computational Science Graduate
488 Fellowship (CSGF) program. Thanks to Pengcheng Zhou for advice on his CNMF_E calcium
489 data cell extraction algorithm. Thanks to Wiktor Młynarski for helpful CNMF discussions.
490 Thanks to Michael Stetner, Galen Lynch, Nhat Le, Dezhe Jin and Jane Van Velden for
491 comments on the manuscript and on our code package. Special thanks to the 2017
492 Methods in Computational Neuroscience course at the Woods Hole Marine Biology Lab,
493 where this collaboration was started.

494 **Author contributions**

495 ELM, AHB, AHW, MSG and MSF conceived the project. ELM, AHB and MSF designed and
496 tested the seqNMF regularizers, the method for cross-validation, and the method for
497 choosing λ . ELM and AHB wrote the algorithm and demo code. ELM and NID collected
498 the imaging data in singing birds. ELM and SG analyzed imaging data. ELM, AHB and MSF
499 wrote the manuscript with input from AHW and MSG.

500 **Methods and Materials**

501 **Table of key resources**

502 Key resources, and references for how to access them, are listed in Table 3.

503 **Contact for resource sharing**

504 Further requests should be directed to Michale Fee (fee@mit.edu).

505 **Software and data availability**

506 Our seqNMF MATLAB code is publicly available as a github repository, along with some of
507 our data for demonstration:

508 <https://github.com/FeeLab/seqNMF>

509 The repository includes the seqNMF function, as well as helper functions for selecting
510 λ and testing the significance of factors, plotting, and other functions. It also includes a
511 demo script that goes through an example of how to select λ for a new dataset, test for
512 significance of factors, and plot the seqNMF factorization.

513 We plan to post more of our data publicly on the CRCNS data-sharing platform.

514 **Generating simulated data**

515 We simulated neural sequences containing between 1 and 10 distinct neural sequences in
516 the presence of various noise conditions. Each neural sequence was made up of 10 con-
517 secutively active neurons. The binary activity matrix was convolved with an exponential
518 kernel to resemble neural calcium imaging activity.

519 **SeqNMF algorithm details**

520 Our algorithm for seqNMF (CNMF with additional regularization to promote efficient
521 factorizations) is a direct extension of the multiplicative update CNMF algorithm [41], and
522 draws on previous work regularizing NMF to encourage factor orthogonality [7].

523 The uniqueness and consistency of traditional NMF has been better studied than
524 CNMF, but in special cases, NMF has a unique solution comprised of sparse, ‘parts-
525 based’ features that can be consistently identified by known algorithms [13, 2]. However,
526 this ideal scenario does not hold in many practical settings. In these cases, NMF is
527 sensitive to initialization, resulting in potentially inconsistent features. This problem can
528 be addressed by introducing additional constraints or regularization terms, and instead
529 encourage the model to extract sparse or approximately orthogonal features [22, 25].
530 Both theoretical work and empirical observations suggest that these modifications result
531 in more consistently identified features [43, 25].

532 For seqNMF, we added to the CNMF cost function a term that promotes competition
533 between overlapping factors, resulting in the following cost function:

$$(\tilde{\mathbf{W}}, \tilde{\mathbf{H}}) = \arg \min_{\mathbf{W}, \mathbf{H}} \left(\|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 + \lambda \|\mathbf{W} \otimes \mathbf{X} \mathbf{S} \mathbf{H}^T\|_{1, i \neq j} \right) \quad (8)$$

534 We derived the following multiplicative update rules for \mathbf{W} and \mathbf{H} (Appendix 1):

$$\mathbf{W}_{.. \ell} \leftarrow \mathbf{W}_{.. \ell} \times \frac{\mathbf{X} \left(\overset{\ell \rightarrow}{\mathbf{H}} \right)^T}{\tilde{\mathbf{X}} \left(\overset{\ell \rightarrow}{\mathbf{H}} \right)^T + \overset{\leftarrow \ell}{\lambda \mathbf{X} \mathbf{S} \mathbf{H}^T (\mathbf{1} - \mathbf{I})}} \quad (9)$$

$$\mathbf{H} \leftarrow \mathbf{H} \times \frac{\mathbf{W} \otimes \mathbf{X}}{\mathbf{W} \otimes \tilde{\mathbf{X}} + \lambda (\mathbf{1} - \mathbf{I}) (\mathbf{W} \otimes \mathbf{X} \mathbf{S})} \quad (10)$$

535 Where the division and \times are element-wise. The operator $(\cdot)^{\overset{\ell \rightarrow}{}}$ shifts a matrix in the \rightarrow
536 direction by ℓ timebins, i.e. a delay by ℓ timebins, and $(\cdot)^{\overset{\leftarrow \ell}{}}$ shifts a matrix in the \leftarrow direction
537 by ℓ timebins (notation summary, Table 1). Note that multiplication with the $K \times K$
538 matrix $(\mathbf{1} - \mathbf{I})$ effectively implements factor competition because it places in the k th row a
539 sum across all other factors. These update rules are derived in Section 1 by taking the
540 derivative of the cost function in Equation 8.

541 In addition to the multiplicative updates outlined in Table 2, we also shift factors to be
542 centered in time, renormalize so rows of \mathbf{H} have unit norm, and in the final iteration run
543 one additional step of unregularized CNMF to prioritize the cost of reconstruction error
544 over the regularization (Algorithm 1).

Algorithm 1: SeqNMF

Input: Data matrix \mathbf{X} , factor number K , factor duration L , regularization strength λ

Output: Factor exemplars \mathbf{W} , and factor timecourses \mathbf{H}

1 Initialize \mathbf{W} and \mathbf{H} randomly

2 lter = 1

3 **while** (lter < Nlter) & (Δ cost > tolerance) **do**

545 4 Update \mathbf{H} using multiplicative update from Table 2

5 Shift \mathbf{W} and \mathbf{H} to center \mathbf{W} 's in time

6 Renormalize \mathbf{W} and \mathbf{H} so rows of \mathbf{H} have unit norm

7 Update \mathbf{W} using multiplicative update from Table 2

8 lter = lter+1

9 Do one final unregularized CNMF update of \mathbf{W} and \mathbf{H}

10 **return**

Calculating consistency

546 The consistency between two factorizations measures the extent to which it is possible to
547 create a one-to-one match between factors in factorization A and factors in factorization
548 B . Specifically, given two factorizations $(\mathbf{W}^A, \mathbf{H}^A)$ and $(\mathbf{W}^B, \mathbf{H}^B)$ respectively, consistency
549 is measured with the following procedure:
550

551 1. For each factor number k , compute the part of the reconstruction explained by this
552 factor in each reconstruction, $\tilde{\mathbf{X}}_k^A = \mathbf{W}_{\cdot k}^A \otimes \mathbf{H}_{k \cdot}^A$ and $\tilde{\mathbf{X}}_k^B = \mathbf{W}_{\cdot k}^B \otimes \mathbf{H}_{k \cdot}^B$.

- 553 2. Reshape $\tilde{\mathbf{X}}_k^A$ and $\tilde{\mathbf{X}}_k^B$ into vectors containing all the elements of each matrix re-
554 spectively, then compute \mathbf{C} , a $K \times K$ correlation matrix where \mathbf{C}_{ij} is the correlation
555 between the vectorized $\tilde{\mathbf{X}}_i^A$ and $\tilde{\mathbf{X}}_j^B$.
- 556 3. Permute the factors greedily so factor 1 is the best matched pair of factors, factor 2
557 is the best match pair of the remaining factors, etc.
- 558 4. Measure consistency as the ratio of the power (sum of squared matrix elements)
559 contained on the diagonal of the permuted \mathbf{C} matrix to the total power in \mathbf{C} .

560 Thus, two factorizations are perfectly consistent when there exists a permutation of factor
561 numbers for which there is a one-to-one match between what parts of the reconstruction
562 are explained by each factor.

563 *Testing the significance of each factor on held-out data*

564 In order to test whether a factor is significantly present in held-out data, we measure the
565 overlap of the factor with the held-out data, and compare this to the null case (Figure S1).
566 Overlap with the data is measured as $\mathbf{W} \otimes \mathbf{X}$, so this quantity will be high at moments
567 when the sequence occurs, producing a distribution of $\mathbf{W} \otimes \mathbf{X}$ with high skew. In contrast,
568 a distribution of overlaps exhibiting low skew indicates a sequence is not present in the
569 data, since there are few moments of particularly high overlap. We estimate what skew
570 levels would appear by chance by constructing null factors where temporal relationships
571 between neurons have been eliminated; within the null factors, the timecourse of each
572 neuron is circularly shifted by a random amount between 0 and L . We measure the skew
573 of the overlap distributions for each null factor, and ask whether the skew we measured
574 for the real factor is significant at p-value α , that is, if it exceeds the $((1 - \frac{\alpha}{K}) \times 100)^{th}$
575 percentile of the null skews. Note the required Bonferroni correction for K comparisons
576 when testing K factors.

577 *Choosing appropriate parameters for a new dataset*

578 Choice of appropriate parameters (λ , K and L) will depend on the data type (sequence
579 length, number, and density; amount of noise; etc.).

580 In practice, we find that results are relatively robust to choice of parameters. When K
581 or L is set larger than necessary, seqNMF tends to simply leave the unnecessary factors
582 or time bins empty. For λ , the goal is to find the 'sweet spot' (Figure 4) to explain as
583 much data as possible while still producing sensible factorizations, that is, uncorrelated
584 factors, with low values of $\|\mathbf{W} \otimes \mathbf{XSH}^T\|_{1,i \neq j}$. Our software package includes demo code
585 for determining the best parameters for a new type of data, using the following strategy:

- 586 1. Start with K slightly larger than the number of sequences anticipated in the data
- 587 2. Start with L slightly longer than the maximum expected factor length
- 588 3. Run seqNMF for a range of λ 's, and for each λ measure the reconstruction error
589 $(\|\mathbf{X} - \mathbf{W} \otimes \mathbf{H}\|_F^2)$ and the factor competition regularization term $(\|\mathbf{W} \otimes \mathbf{XSH}^T\|_{1,i \neq j})$
- 590 4. Choose a λ slightly above the crossover point λ_0
- 591 5. Decrease K if desired, as otherwise some factors will be consistently empty
- 592 6. Decrease L if desired, as otherwise some time bins will consistently be empty

593 In some applications, achieving the desired accuracy may depend on choosing a λ
594 that allows some inconsistency. It is possible to deal with this remaining inconsistency

595 by comparing factors produced by different random initializations, and only considering
596 factors that arise from several different initializations, a strategy that has been previously
597 applied to standard CNMF on neural data [34].

598 During validation of our lambda choosing strategy we compared factorizations to
599 ground truth sequences as shown in figure 4. To find the optimal lambda we used the
600 product of two curves. The first curve was obtained by calculating the fraction of fits in
601 which the true number of sequences was recovered as a function of λ . The second curve
602 was obtained by calculating similarity to ground truth as a function of λ . The product
603 of these two curves was smoothed using a three sample boxcar sliding window and the
604 width was found as the lambda on either side of the peak value which was nearest the
605 half-maximum.

606 *Measuring performance on noisy data by comparing seqNMF sequences to* 607 *ground-truth sequences*

608 We wanted to measure the ability of seqNMF to recover ground-truth sequences even
609 when the sequences are obstructed by noise. Our noisy data consisted of two ground-
610 truth sequences, obstructed by a variety of noise types. We first took the top seqNMF
611 factor, and made a reconstruction with only this factor. We then measured the correlation
612 between this reconstruction and reconstructions generated from each of the ground-
613 truth factors, and chose the best match. Next, we measured the correlation between the
614 remaining ground-truth reconstruction and the second seqNMF factor. The mean of these
615 two correlations was used as a measure of similarity between the seqNMF factorization
616 and the ground-truth (noiseless) sequences.

617 *Algorithm speed*

618 In practice, our algorithm converges rapidly: fewer than 100 iterations on a typical 150
619 neuron by 10,000 time point data matrix, typically less than 30 seconds on a standard
620 PC. However, applications to much larger datasets may require faster performance. In
621 these cases, we recommend running seqNMF on smaller subsets of the dataset, perhaps
622 by incorporating seqNMF regularization into an online version of CNMF [46], and/or
623 parallelizing the algorithm by running it on shorter datasets and merging/recombining
624 factors that are common across these shorter runs (finding common factors by e.g. [34]).

625 **Hippocampus data**

626 The hippocampal data we used was collected in the Buzsaki lab [32], and is publicly
627 available on the Collaborative Research in Computational Neuroscience (CRCNS) Data
628 sharing website. The dataset we refer to as 'Rat 1' is in the [hc-5](#) dataset, and the dataset
629 we refer to as 'Rat 2' is in the [hc-3](#) and dataset. Before running seqNMF, we processed
630 the data by convolving the raw spike trains with a gaussian kernel of standard deviation
631 100ms.

632 **Animal care and use**

633 We used male zebra finches (*Taeniopygia guttata*) from the MIT zebra finch breeding facility
634 (Cambridge, MA). Animal care and experiments were reviewed and approved by the
635 Massachusetts Institute of Technology Committee on Animal Care.

636 In order to prevent exposure to a tutor song, birds were foster-raised by female birds,
637 which do not sing, starting on or before post-hatch day 15. For experiments, birds were
638 housed singly in custom-made sound isolation chambers.

639 **Calcium imaging**

640 The calcium indicator GCaMP6f was expressed in HVC by intercranial injection of the viral
641 vector AAV9.CAG.GCaMP6f.WPRE.SV40 [6] into HVC. In the same surgery, a cranial window
642 was made using a GRIN (gradient index) lens (1 mm diameter, 4mm length, Inscopix).
643 After at least one week, in order to allow for sufficient viral expression, recordings were
644 made using the Inscopix nVista miniature fluorescent microscope.

645 Neuronal activity traces were extracted from raw fluorescence movies using the
646 CNMF_E algorithm, a constrained non-negative matrix factorization algorithm specialized
647 for microendoscope data by including a local background model to remove activity from
648 out-of-focus cells [48].

649 **References**

- 650 [1] **Aronov D**, Andalman AS, Fee MS. A specialized forebrain circuit for vocal babbling in the
651 juvenile songbird. *Science* (New York, NY). 2008 may; 320(5876):630–4. <http://www.ncbi.nlm.nih.gov/pubmed/18451295>, doi: 10.1126/science.1155140.
- 652
- 653 [2] **Arora S**, Ge R, Kannan R, Moitra A. Computing a Nonnegative Matrix Factorization – Provably.
654 *ArXiv e-prints*. 2011 Nov; .
- 655 [3] **Bapi RS**, Pammi VSC, Miyapuram KP, Ahmed. Investigation of sequence processing: A cognitive
656 and computational neuroscience perspective. *Current Science*. 2005; 89(10):1690–1698. <http://www.jstor.org/stable/24111208>.
- 657
- 658 [4] **Brody CD**. Correlations Without Synchrony. *Neural Computation*. 1999; 11(7):1537–1551.
659 <https://doi.org/10.1162/089976699300016133>, doi: 10.1162/089976699300016133.
- 660 [5] **Bzdok D**, Yeo BTT. Inference in the age of big data: Future perspectives on neuroscience.
661 *NeuroImage*. 2017; 155(Supplement C):549 – 564. <http://www.sciencedirect.com/science/article/pii/S1053811917303816>, doi: <https://doi.org/10.1016/j.neuroimage.2017.04.061>.
- 662
- 663 [6] **Chen TW**, Wardill TJ, Sun Y, Pulver SR, Renninger SL, Baohan A, Schreiter ER, Kerr
664 RA, Orger MB, Jayaraman V, Looger LL, Svoboda K, Kim DS. Ultrasensitive fluo-
665 rescent proteins for imaging neuronal activity. *Nature*. 2013 jul; 499(7458):295–
666 300. <http://www.ncbi.nlm.nih.gov/pubmed/23868258><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3777791>, doi: 10.1038/nature12354.
- 667
- 668 [7] **Chen Z**, Cichocki A. Nonnegative matrix factorization with temporal smoothness and/or spatial
669 decorrelation constraints. In: *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech.*
670 *Rep*; 2005. .
- 671 [8] **Churchland AK**, Abbott LF. Conceptual and technical advances define a key moment for
672 theoretical neuroscience. *Nature Neuroscience*. 2016 feb; 19(3):348–349. <http://www.nature.com/doi/10.1038/nn.4255>, doi: 10.1038/nn.4255.
- 673
- 674 [9] **Cichocki A**. Nonnegative Matrix and Tensor Factorizations : Applications to Ex-
675 ploratory Multi-way Data Analysis and Blind Source Separation. Wiley; 2009.
676 [http://libproxy.mit.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&](http://libproxy.mit.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=287301&site=ehost-live&scope=site)
677 [db=nlebk&AN=287301&site=ehost-live&scope=site](http://libproxy.mit.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=287301&site=ehost-live&scope=site).

- 678 [10] **Clegg BA**, Digirolamo GJ, Keele SW. Sequence learning. Trends in cognitive sciences. 1998
679 aug; 2(8):275–81. <http://www.ncbi.nlm.nih.gov/pubmed/21227209>, doi: 10.1016/S1364-
680 6613(98)01202-9.
- 681 [11] **Cui Y**, Ahmad S, Hawkins J. Continuous Online Sequence Learning with an Unsupervised
682 Neural Network Model. Neural Computation. 2016; 28(11):2474–2504. [https://doi.org/10.1162/](https://doi.org/10.1162/NECO_a_00893)
683 [NECO_a_00893](https://doi.org/10.1162/NECO_a_00893), doi: 10.1162/NECO_a_00893, pMID: 27626963.
- 684 [12] **Cunningham JP**, Yu BM. Dimensionality reduction for large-scale neural recordings. Nature
685 Neuroscience. 2014 nov; 17(11):1500–1509. <http://www.nature.com/articles/nn.3776>, doi:
686 [10.1038/nn.3776](https://doi.org/10.1038/nn.3776).
- 687 [13] **Donoho D**, Stodden V. When Does Non-Negative Matrix Factorization Give a Correct Decom-
688 position into Parts? In: Thrun S, Saul LK, Schölkopf B, editors. Advances in Neural Information
689 Processing Systems 16 MIT Press; 2004.p. 1141–1148. [http://papers.nips.cc/paper/](http://papers.nips.cc/paper/2463-when-does-non-negative-matrix-factorization-give-a-correct-decomposition-into-parts.pdf)
690 [2463-when-does-non-negative-matrix-factorization-give-a-correct-decomposition-into-parts.](http://papers.nips.cc/paper/2463-when-does-non-negative-matrix-factorization-give-a-correct-decomposition-into-parts.pdf)
691 [pdf](http://papers.nips.cc/paper/2463-when-does-non-negative-matrix-factorization-give-a-correct-decomposition-into-parts.pdf).
- 692 [14] **Fehér O**, Wang H, Saar S, Mitra PP, Tchernichovski O. De novo establishment of wild-type song
693 culture in the zebra finch. Nature. 2009 may; 459(7246):564–8. [http://www.pubmedcentral.](http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2693086&tool=pmcentrez&rendertype=abstract)
694 [nih.gov/articlerender.fcgi?artid=2693086&tool=pmcentrez&rendertype=abstract](http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2693086&tool=pmcentrez&rendertype=abstract), doi:
695 [10.1038/nature07994](https://doi.org/10.1038/nature07994).
- 696 [15] **Fujisawa S**, Amarasingham A, Harrison M, Buzsáki G. Behavior-dependent short-term as-
697 sembly dynamics in the medial prefrontal cortex. Nature Neuroscience. 2008; 11(7):823–833.
698 <https://www.nature.com/articles/nn.2134>, doi: [10.1038/nn.2134](https://doi.org/10.1038/nn.2134).
- 699 [16] **Gao P**, Ganguli S. On Simplicity and Complexity in the Brave New World of Large-Scale
700 Neuroscience. ArXiv e-prints. 2015 Mar; .
- 701 [17] **Gerstein GL**, Williams ER, Diesmann M, Gründ S, Trengove C. Detecting synfire
702 chains in parallel spike data. Journal of Neuroscience Methods. 2012; 206:54–64. doi:
703 [10.1016/j.jneumeth.2012.02.003](https://doi.org/10.1016/j.jneumeth.2012.02.003).
- 704 [18] **Hahnloser RHR**, Kozhevnikov AA, Fee MS. An ultra-sparse code underlies the generation
705 of neural sequences in a songbird. Nature. 2002 09; 419:65–70. [http://dx.doi.org/10.1038/](http://dx.doi.org/10.1038/nature00974)
706 [nature00974](http://dx.doi.org/10.1038/nature00974).
- 707 [19] **Harvey CD**, Coen P, Tank DW. Choice-specific sequences in parietal cortex during a virtual-
708 navigation decision task. Nature. 2012 03; 484:62–68. <http://dx.doi.org/10.1038/nature10918>.
- 709 [20] **Hastie T**, Tibshirani R, Friedman JH. The elements of statistical learning : data mining,
710 inference, and prediction. Springer; 2009.
- 711 [21] **Hawkins J**, Ahmad S. Why Neurons Have Thousands of Synapses, a Theory of Sequence
712 Memory in Neocortex. Frontiers in Neural Circuits. 2016; 10:23. [https://www.frontiersin.org/](https://www.frontiersin.org/article/10.3389/fncir.2016.00023)
713 [article/10.3389/fncir.2016.00023](https://www.frontiersin.org/article/10.3389/fncir.2016.00023), doi: [10.3389/fncir.2016.00023](https://doi.org/10.3389/fncir.2016.00023).
- 714 [22] **Huang K**, Sidiropoulos ND, Swami A. Non-Negative Matrix Factorization Revisited: Uniqueness
715 and Algorithm for Symmetric Decomposition. IEEE Transactions on Signal Processing. 2014
716 Jan; 62(1):211–224. doi: [10.1109/TSP.2013.2285514](https://doi.org/10.1109/TSP.2013.2285514).
- 717 [23] **Janata P**, Grafton ST. Swinging in the brain: shared neural substrates for behaviors related to
718 sequencing and music. Nature Neuroscience. 2003 jul; 6(7):682–687. [http://www.nature.com/](http://www.nature.com/articles/nn1081)
719 [articles/nn1081](http://www.nature.com/articles/nn1081), doi: [10.1038/nn1081](https://doi.org/10.1038/nn1081).

- 720 [24] **Jun JJ**, Steinmetz NA, Siegle JH, Denman DJ, Bauza M, Barbarits B, Lee AK, Anastassiou CA,
721 Andrei A, Aydin Ç, Barbic M, Blanche TJ, Bonin V, Couto J, Dutta B, Gratiy SL, Gutnisky DA,
722 Häusser M, Karsh B, Ledochowitsch P, et al. Fully integrated silicon probes for high-density
723 recording of neural activity. *Nature*. 2017 nov; 551(7679):232–236. [http://www.nature.com/](http://www.nature.com/doi/10.1038/nature24636)
724 [doi/10.1038/nature24636](http://www.nature.com/doi/10.1038/nature24636), doi: 10.1038/nature24636.
- 725 [25] **Kim J**, Park H. Sparse Nonnegative Matrix Factorization for Clustering. In: ; 2008. .
- 726 [26] **Kim TH**, Zhang Y, Lecoq J, Jung JC, Li J, Zeng H, Niell CM, Schnitzer MJ. Long-Term
727 Optical Access to an Estimated One Million Neurons in the Live Mouse Cortex.
728 *Cell reports*. 2016 dec; 17(12):3385–3394. [http://www.ncbi.nlm.nih.gov/pubmed/](http://www.ncbi.nlm.nih.gov/pubmed/28009304)
729 [28009304](http://www.ncbi.nlm.nih.gov/pubmed/28009304)<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5459490>, doi:
730 [10.1016/j.celrep.2016.12.004](https://doi.org/10.1016/j.celrep.2016.12.004).
- 731 [27] **Lee DD**, Seung HS. Learning the parts of objects by non-negative matrix factorization. *Nature*.
732 1999; 401(6755):788–791.
- 733 [28] **Long MA**, Jin DZ, Fee MS. Support for a synaptic chain model of neuronal sequence generation.
734 *Nature*. 2010 nov; 468(7322):394–9. [http://www.pubmedcentral.nih.gov/articlerender.fcgi?](http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2998755&tool=pmcentrez&rendertype=abstract)
735 [artid=2998755&tool=pmcentrez&rendertype=abstract](http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2998755&tool=pmcentrez&rendertype=abstract), doi: 10.1038/nature09514.
- 736 [29] **Lynch G**, Okubo T, Hanuschkin A, Hahnloser RR, Fee M. Rhythmic Continuous-
737 Time Coding in the Songbird Analog of Vocal Motor Cortex. *Neuron*. 2016;
738 90(4):877 – 892. <http://www.sciencedirect.com/science/article/pii/S0896627316301088>, doi:
739 <https://doi.org/10.1016/j.neuron.2016.04.021>.
- 740 [30] **O’Grady PD**, Pearlmutter BA. Convolutional Non-Negative Matrix Factorisation with a Sparseness
741 Constraint. In: 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for
742 Signal Processing; 2006. p. 427–432. doi: [10.1109/MLSP.2006.275588](https://doi.org/10.1109/MLSP.2006.275588).
- 743 [31] **Okubo TS**, Mackevicius EL, Payne HL, Lynch GF, Fee MS. Growth and splitting of neural
744 sequences in songbird vocal development. *Nature*. 2015 nov; 528(7582):352–357. [http://www.](http://www.ncbi.nlm.nih.gov/pubmed/26618871)
745 [ncbi.nlm.nih.gov/pubmed/26618871](http://www.ncbi.nlm.nih.gov/pubmed/26618871)[http://www.pubmedcentral.nih.gov/articlerender.fcgi?](http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4957523)
746 [artid=PMC4957523](http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4957523)<http://www.nature.com/doi/10.1038/nature15741>, doi: 10.1038/na-
747 [ture15741](http://www.nature.com/doi/10.1038/nature15741).
- 748 [32] **Pastalkova E**, Itskov V, Amarasingham A, Buzsáki G. Internally Generated Cell Assembly
749 Sequences in the Rat Hippocampus. *Science*. 2008; 321(5894):1322–1327. [http://science.](http://science.sciencemag.org/content/321/5894/1322)
750 [sciencemag.org/content/321/5894/1322](http://science.sciencemag.org/content/321/5894/1322), doi: [10.1126/science.1159775](https://doi.org/10.1126/science.1159775).
- 751 [33] **Pearson K**. LIII. <i>On lines and planes of closest fit to systems of points in space</i>.
752 *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 1901
753 nov; 2(11):559–572. <https://www.tandfonline.com/doi/full/10.1080/14786440109462720>, doi:
754 [10.1080/14786440109462720](https://www.tandfonline.com/doi/full/10.1080/14786440109462720).
- 755 [34] **Peter S**, Durstewitz D, Diego F, Hamprecht FA. Sparse convolutional coding for neuronal
756 ensemble identification. *ArXiv e-prints*. 2016 Jun; .
- 757 [35] **Picardo M**, Merel J, Katlowitz K, Vallentin D, Okobi D, Benezra S, Clary R, Pnevmatikakis E,
758 Paninski L, Long M. Population-Level Representation of a Temporal Sequence Underlying Song
759 Production in the Zebra Finch. *Neuron*. 2016; 90(4):866 – 876. [http://www.sciencedirect.com/](http://www.sciencedirect.com/science/article/pii/S0896627316001094)
760 [science/article/pii/S0896627316001094](http://www.sciencedirect.com/science/article/pii/S0896627316001094), doi: <https://doi.org/10.1016/j.neuron.2016.02.016>.
- 761 [36] **Ramanarayanan V**, Goldstein L, Narayanan SS. Spatio-temporal articulatory movement
762 primitives during speech production: Extraction, interpretation, and validation. *The Journal of*

- 763 the Acoustical Society of America. 2013; 134(2):1378–1394. <https://doi.org/10.1121/1.4812765>,
764 [doi: 10.1121/1.4812765](https://doi.org/10.1121/1.4812765).
- 765 [37] **Russo E**, Durstewitz D. Cell assemblies at multiple time scales with arbitrary lag constellations.
766 *eLife*. 2017; 6:e19428. <https://doi.org/10.7554/eLife.19428>, [doi: 10.7554/eLife.19428](https://doi.org/10.7554/eLife.19428).
- 767 [38] **Scholvin J**, Kinney JP, Bernstein JG, Moore-Kochlacs C, Kopell N, Fonstad CG, Boyden
768 ES. Close-Packed Silicon Microelectrodes for Scalable Spatially Oversampled Neural
769 Recording. *IEEE Transactions on Biomedical Engineering*. 2016 Jan; 63(1):120–130. [doi:](https://doi.org/10.1109/TBME.2015.2406113)
770 [10.1109/TBME.2015.2406113](https://doi.org/10.1109/TBME.2015.2406113).
- 771 [39] **Sejnowski TJ**, Churchland PS, Movshon JA. Putting big data to good use in neuroscience.
772 *Nature neuroscience*. 2014 nov; 17(11):1440–1. [http://www.ncbi.nlm.nih.gov/pubmed/](http://www.ncbi.nlm.nih.gov/pubmed/25349909)
773 [25349909](http://www.ncbi.nlm.nih.gov/pubmed/25349909)<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4224030>, [doi:](https://doi.org/10.1038/nn.3839)
774 [10.1038/nn.3839](https://doi.org/10.1038/nn.3839).
- 775 [40] **Smaragdis P**. Convolutional Speech Bases and Their Application to Supervised Speech Separation.
776 *IEEE Transactions on Audio, Speech, and Language Processing*. 2007 Jan; 15(1):1–12. [doi:](https://doi.org/10.1109/TASL.2006.876726)
777 [10.1109/TASL.2006.876726](https://doi.org/10.1109/TASL.2006.876726).
- 778 [41] **Smaragdis P**. In: Puntonet CG, Prieto A, editors. *Non-negative Matrix Factor Deconvolution;*
779 *Extraction of Multiple Sound Sources from Monophonic Inputs* Berlin, Heidelberg: Springer
780 Berlin Heidelberg; 2004. p. 494–499. https://doi.org/10.1007/978-3-540-30110-3_63, [doi:](https://doi.org/10.1007/978-3-540-30110-3_63)
781 [10.1007/978-3-540-30110-3_63](https://doi.org/10.1007/978-3-540-30110-3_63).
- 782 [42] **Sutskever I**, Vinyals O, Le QV. Sequence to Sequence Learning with Neural Networks. In:
783 Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. *Advances in Neural*
784 *Information Processing Systems 27* Curran Associates, Inc.; 2014.p. 3104–3112. [http://papers.](http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf)
785 [nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf](http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf).
- 786 [43] **Theis FJ**, Stadlthanner K, Tanaka T. First results on uniqueness of sparse non-negative matrix
787 factorization. In: *2005 13th European Signal Processing Conference*; 2005. p. 1–4.
- 788 [44] **Udell M**, Horn C, Zadeh R, Boyd S. Generalized Low Rank Models. *Foundations*
789 *and Trends in Machine Learning*. 2016; 9(1). <http://dx.doi.org/10.1561/22000000055>, [doi:](https://doi.org/10.1561/22000000055)
790 [10.1561/22000000055](https://doi.org/10.1561/22000000055).
- 791 [45] **Vaz C**, Toutios A, Narayanan S. Convex Hull Convolutional Non-negative Matrix Factorization for
792 Uncovering Temporal Patterns in Multivariate Time-Series Data. In: *Interspeech* San Francisco,
793 CA; 2016. p. 963–967.
- 794 [46] **Wang D**, Vipperla R, Evans N, Zheng TF. Online Non-Negative Convolutional Pattern Learning
795 for Speech Signals. *IEEE Transactions on Signal Processing*. 2013 Jan; 61(1):44–56. [doi:](https://doi.org/10.1109/TSP.2012.2222381)
796 [10.1109/TSP.2012.2222381](https://doi.org/10.1109/TSP.2012.2222381).
- 797 [47] **Zhang Z**, Xu Y, Yang J, Li X, Zhang D. A survey of sparse representation: algorithms and
798 applications. *ArXiv e-prints*. 2016 Feb; .
- 799 [48] **Zhou P**, Resendez SL, Rodriguez-Romaguera J, Jimenez JC, Neufeld SQ, Stuber GD, Hen R,
800 Kheirbek MA, Sabatini BL, Kass RE, Paninski L. Efficient and accurate extraction of in vivo
801 calcium signals from microendoscopic video data. *ArXiv e-prints*. 2016 May; .

802 **Figures**

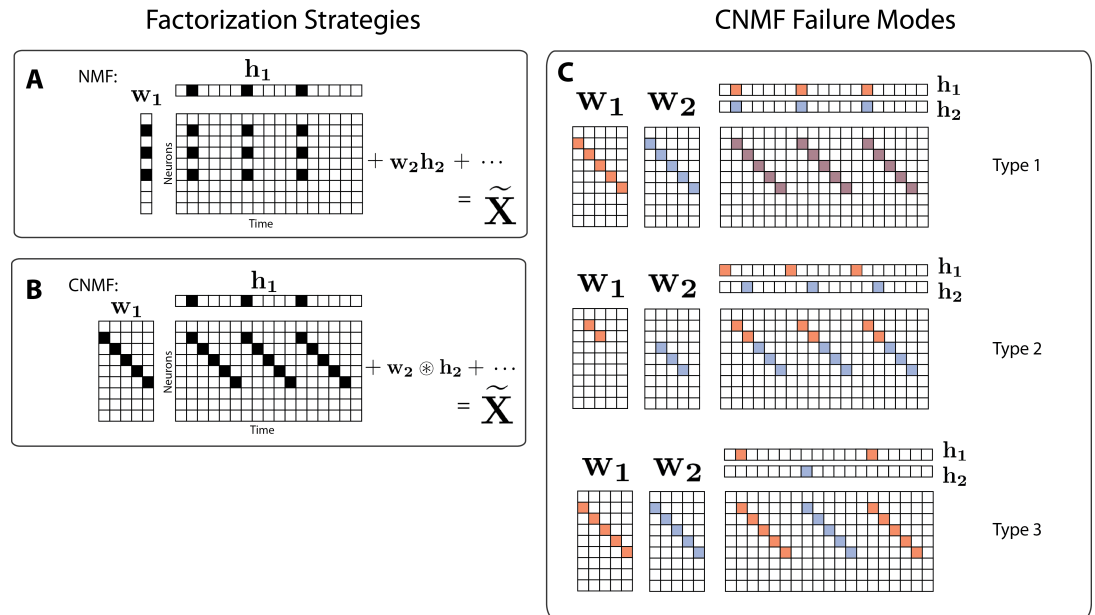


Figure 1. Introduction to CNMF factorization failure modes motivating seqNMF regularization
(A) NMF (non-negative matrix factorization) approximates a dataset containing N neurons at T timepoints as a sum of K rank-one matrices. Each matrix is generated as the outer product of two nonnegative vectors: \mathbf{w}_k of length N , which stores a neural ensemble, and \mathbf{h}_k of length T , which holds the times at which the neural ensemble is active. **(B)** Convolutional NMF also approximates an $N \times T$ dataset as a sum of K matrices. Each matrix is generated as the convolution of two components: a non-negative matrix \mathbf{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, \mathbf{h}_k , which holds the times at which each factor pattern is active in the data. **(C)** Three types of inefficiencies are present in unregularized CNMF: 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 piecewise manner, and Type 3 in which two factors are used to reconstruct different instances of the same sequence.

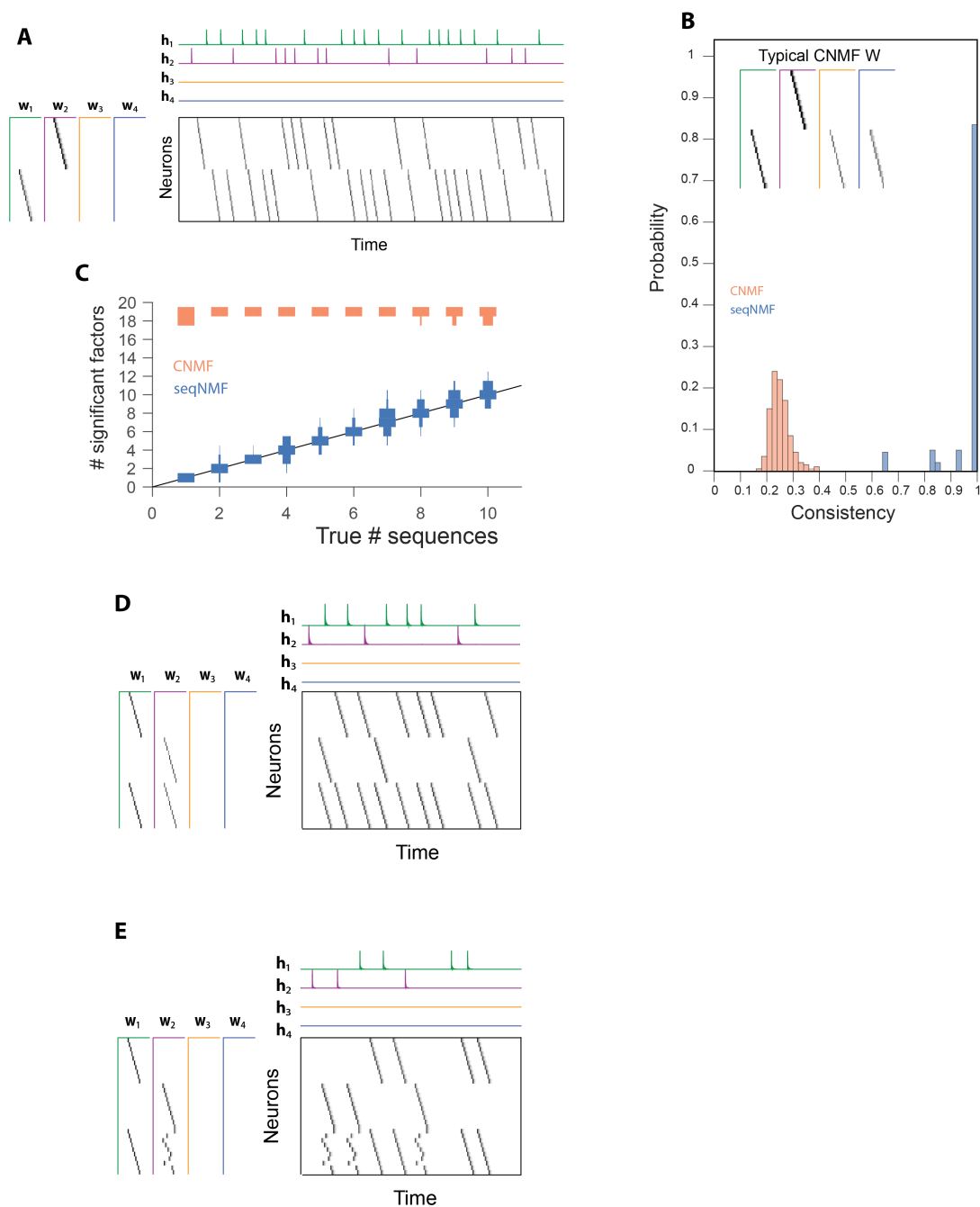


Figure 2. Testing seqNMF on simulated data

(A) A simulated dataset with two simulated neural sequences and a seqNMF factorization ($K = 4$, $L = 250$, $\lambda = 0.0005$) **(B)** SeqNMF is far more consistent than unregularized CNMF across 100 independent fits ($K = 20$, $L = 250$, $\lambda = 0.0005$). Inset: neural patterns for a typical CNMF factorization showing redundant copies of the lower sequence. **(C)** Discrete violin plots showing the number of statistically significant factors vs. true number of simulated sequences for seqNMF and CNMF for 100 fits of simulated data containing between 1 and 10 sequences ($K = 20$, $L = 250$, $\lambda = 0.0005$). **(D)** A seqNMF factorization of two simulated neural sequences with shared neurons that participate at the same latency in both sequences **(E)** A seqNMF factorization of two simulated neural sequences with shared neurons that participate at different latencies in each sequence.

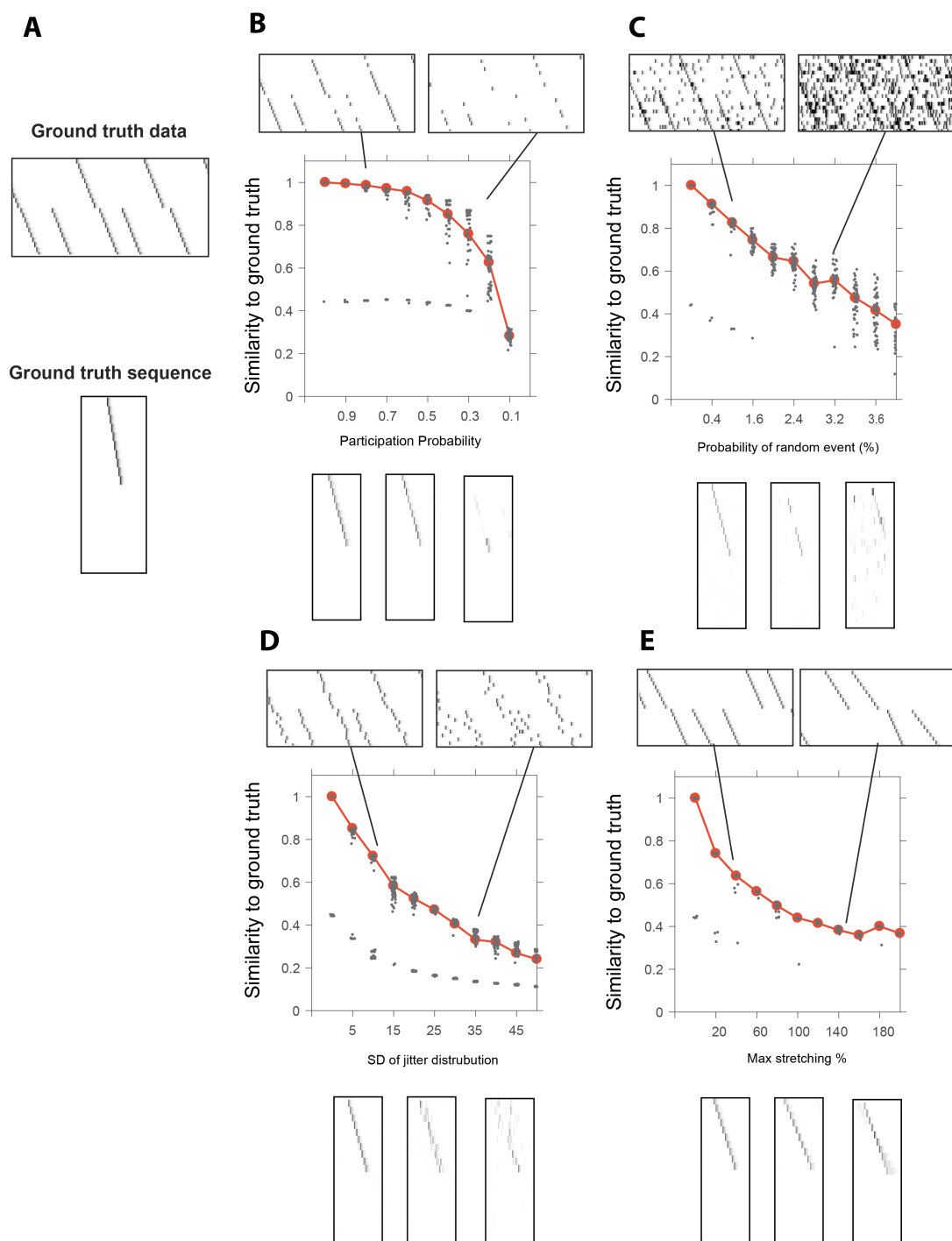


Figure 3. Testing seqNMF performance on sequences contaminated with noise

(A) Ground-truth (noiseless) data, as well as an example of one ground-truth sequence used to generate the data. Performance of seqNMF was tested under 4 different noise conditions: (B) probabilistic participation, (C) additive noise, (D) timing jitter, and (E) sequence warping. For each noise type, we show: (top) examples of synthetic data at 2 different noise levels, (middle) similarity of seqNMF factors to ground-truth factors across a range of noise levels, showing 50 fits for each noise level, with red lines indicating the median, and (bottom) example W 's extracted at 3 different noise levels. SeqNMF was run with $K = 20$, $L = 250$, and λ chosen using the automated procedure outlined in Figure 4.

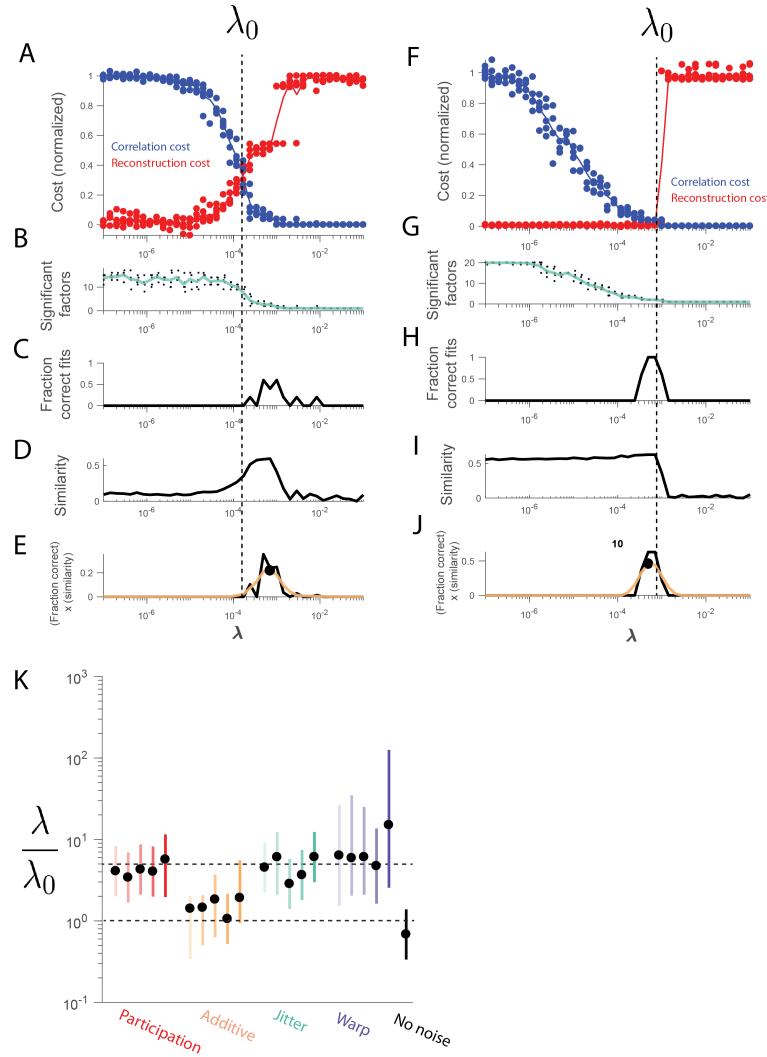


Figure 4. Procedure for choosing λ for a new dataset based on finding a balance between reconstruction cost and correlation cost in noisy and noiseless data

(A) Normalized reconstruction cost ($\|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2$) and correlation cost ($\|\mathbf{W} \otimes \mathbf{XSH}^T\|_{1,i \neq j}$) as a function of λ for simulated data containing two sequences in the presence of participation noise (70% participation probability). The cross-over point λ_0 is marked. **(B)** The number of significant factors obtained from 20 fits of these data as a function of λ (mean number plotted in green). **(C)** The fraction of fits returning the correct number of significant factors (two) as a function of λ . **(D)** Similarity of the top two factors to ground-truth (noiseless) factors as a function of λ . **(E)** The product of the curves shown in (C) and (D), (smoothed curve plotted in orange) with a circle marking the peak. **(F)** Normalized reconstruction cost and correlation cost as a function of λ for simulated data containing two noiseless sequences. **(G-J)** Same as (B-E) but for the noiseless data. **(K)** Summary plot showing the range of values of λ (vertical bars), relative to the cross-over point λ_0 , that work well for each noise condition (\pm half height points of the curve shown in panel E; note that this curve is a product of two other curves, and thus narrower, giving a conservative estimate of the range of effective λ s). Circles indicate the value of λ at the peak of the curves in (E). For each noise type, results for the first five non-zero noise levels from Figure 3 are shown (increasing color saturation at high noise levels; Red, participation: 90,80,70,60 and 50%; Orange, additive noise 0.4, 0.8, 1.2, 1.6 and 2%; Green, jitter: 5,10,15,20, and 25 timesteps; Purple, timewarp: 10,20,30,40, and 50%)

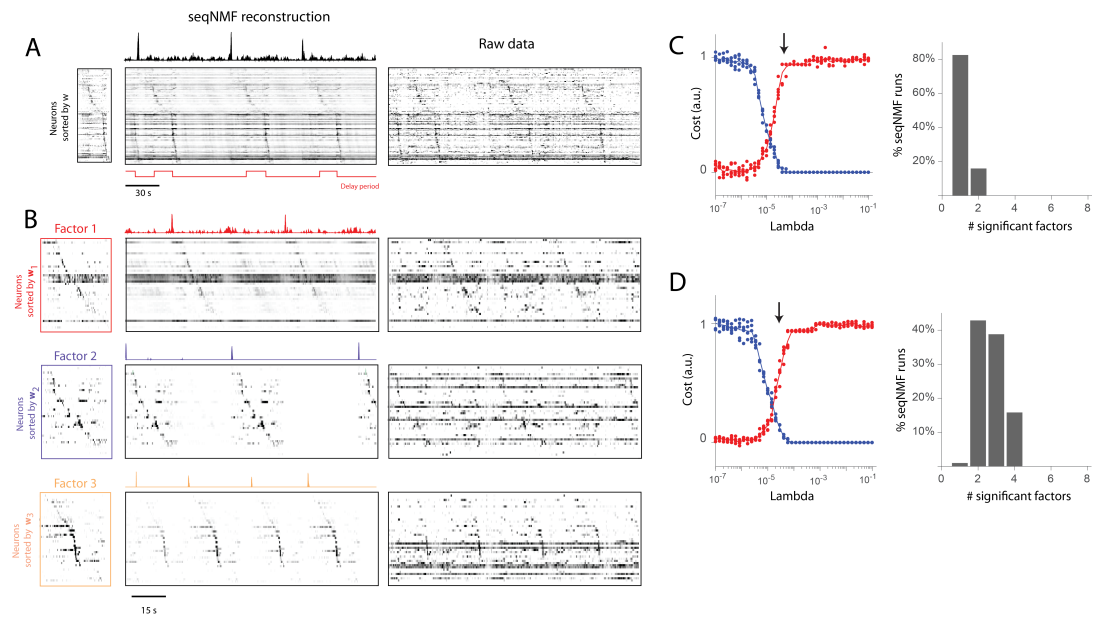


Figure 5. Application of seqNMF to extract hippocampal sequences from two rats

(A) Firing rates of 110 neurons recorded in the hippocampus of Rat 1 during an alternating left-right task with a delay period [32], as well as the seqNMF factor. Neurons are sorted according to their latency 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 [32]. Neurons are sorted according to their latency within each of the three significant extracted sequences. Both seqNMF reconstruction of each factor (left) and raw data (right) are shown. The first two factors correspond to left and right trials, and the third corresponds to running along the stem of the maze.

(C) (Left) Reconstruction (red) and correlation (blue) costs as a function of λ for Rat 1. Arrow indicates $\lambda = 6 \times 10^{-5}$, used for seqNMF factorization shown in (A) (Right) Histogram of the number of significant factors across 30 runs of seqNMF.

(D) Same as in (C) but for Rat 2. Arrow indicates $\lambda = 3 \times 10^{-5}$ used for factorization shown in (B).

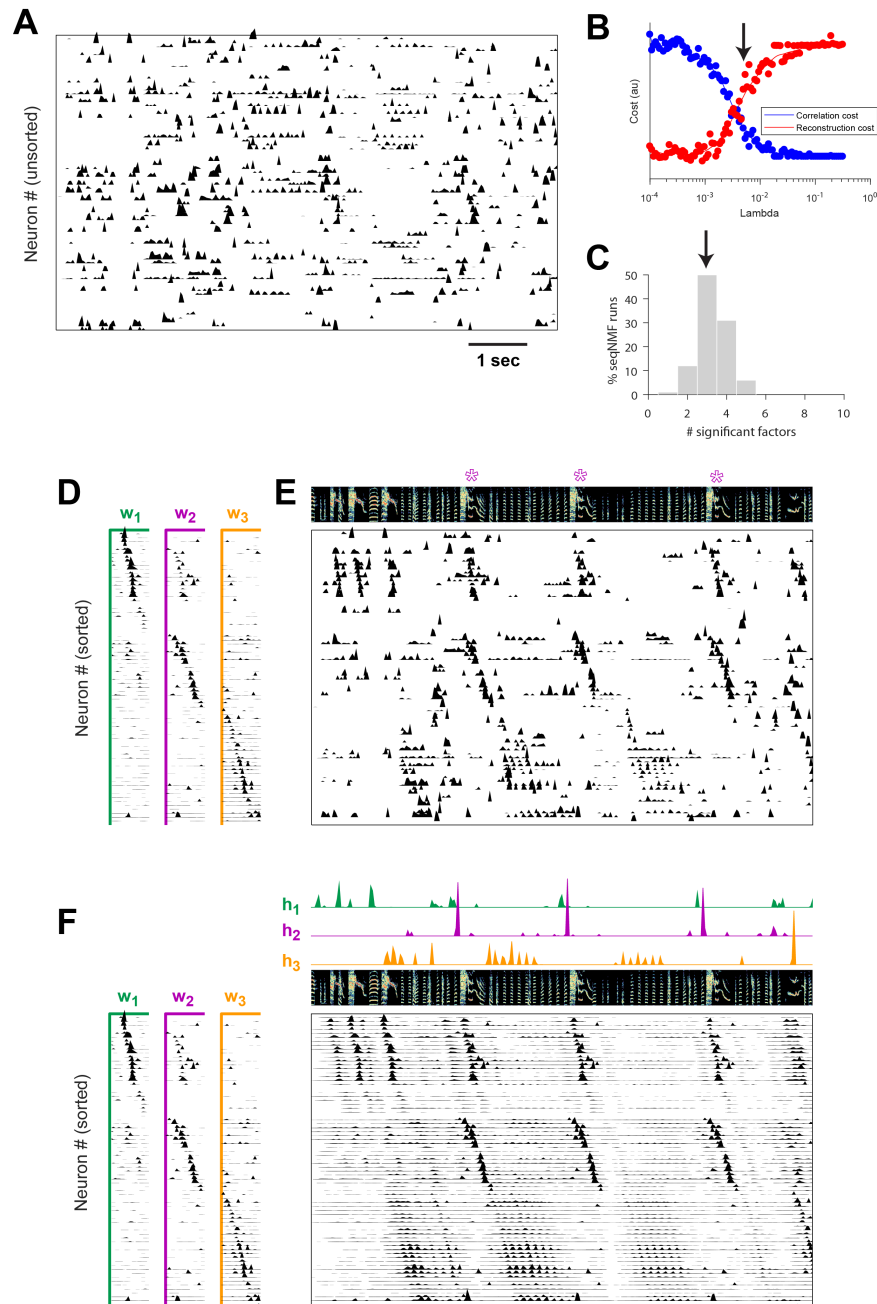


Figure 6. SeqNMF applied to calcium imaging data from a singing isolate bird reveals abnormal sequence deployment

(A) Functional calcium signals recorded from 75 neurons, unsorted, in a singing isolate bird. (B) Reconstruction and correlation cost as a function of lambda. The arrow at $\lambda = 0.005$ indicates the value selected for the rest of the analysis. (C) Number of significant factors for 100 runs of seqNMF with $K = 10$, $\lambda = 0.005$. Arrow indicates 3 is the most common number of significant factors. (D) SeqNMF factor exemplars (W 's), sorting neurons by their latency within each 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.

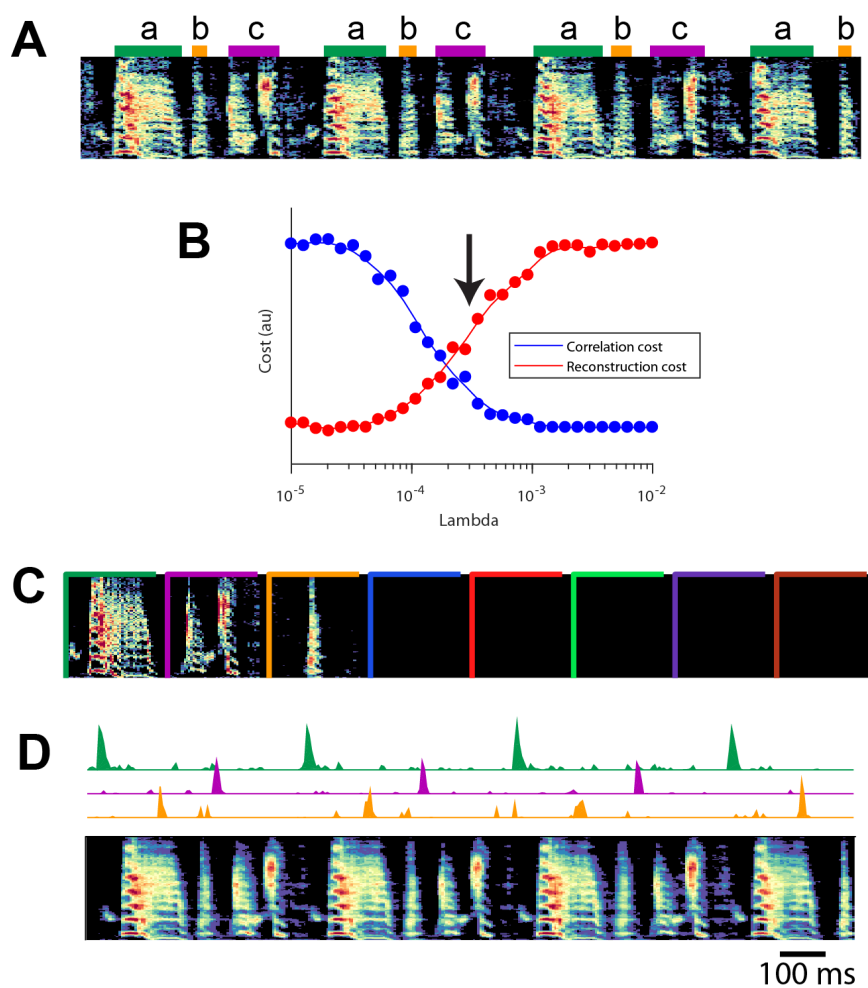


Figure 7. SeqNMF applied to song spectrograms

(A) Spectrogram of juvenile song, with hand-labeled syllable types [31]. (B) Reconstruction cost and correlation cost for these data as a function of λ . Arrow denotes $\lambda = 0.0003$, which was used to run seqNMF (C) SeqNMF \mathbf{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) SeqNMF \mathbf{H} 's (for the three non-empty factors) and seqNMF reconstruction of the song shown in (A) using these factors.

Table 1. Notation for convolutional matrix factorization

Shift operator

The operator $\overset{\ell \rightarrow}{(\cdot)}$ shifts a matrix in the \rightarrow direction by ℓ timebins:

$$\overset{\ell \rightarrow}{(\mathbf{A})}_t = \mathbf{A}_{\cdot, (t-\ell)} \text{ and likewise } \overset{\ell \leftarrow}{(\mathbf{A})}_t = \mathbf{A}_{\cdot, (t+\ell)}$$

The shift operator inserts zeros when $(t - \ell) < 0$ or $(t + \ell) > T$

Tensor convolution operator

Convolutional matrix factorization reconstructs a data matrix \mathbf{X}

using a $N \times K \times L$ tensor \mathbf{W} and a $K \times T$ matrix \mathbf{H} :

$$\tilde{\mathbf{X}} = \mathbf{W} \circledast \mathbf{H} = \sum_{\ell} \mathbf{W}_{\cdot \cdot \ell} \overset{\ell \rightarrow}{\mathbf{H}}$$

Note that each neuron n is reconstructed as the sum of k convolutions:

$$\tilde{\mathbf{X}}_{nt} = \sum_k \sum_{\ell} \mathbf{W}_{nk\ell} \mathbf{H}_{k(t-\ell)} \equiv (\mathbf{W} \circledast \mathbf{H})_{nt}$$

Transpose tensor convolution operator

The following quantity is useful in several contexts:

$$\mathbf{W} \circledast^{\top} \mathbf{X} = \sum_{\ell} \mathbf{W}_{\cdot \cdot \ell}^{\top} \overset{\ell \leftarrow}{\mathbf{X}}$$

Note that each element $(\mathbf{W} \circledast^{\top} \mathbf{X})_{kt} = \sum_{\ell} \mathbf{W}_{\cdot k \ell}^{\top} \mathbf{X}_{\cdot, (t+\ell)}$ measures

the overlap (correlation) of factor k with the data at time t

CNMF reconstruction

$$\mathbf{X} \approx \tilde{\mathbf{X}} = \sum_k \mathbf{W}_{\cdot k} \circledast \mathbf{H}_k = \mathbf{W} \circledast \mathbf{H}$$

Note that NMF is special case of CNMF, where $L = 1$

$L1$ norm excluding diagonal

For any $K \times K$ matrix \mathbf{C} ,

$$\|\mathbf{C}\|_{1, i \neq j} \equiv \sum_k \sum_{j \neq k} \mathbf{C}_{jk}$$

Special matrices

$\mathbf{1}$ is a $K \times K$ matrix of ones

\mathbf{I} is the $K \times K$ identity matrix

\mathbf{S} is a smoothing matrix: $s_{ij} = 1$ when $|i - j| < L$ and otherwise $s_{ij} = 0$

Table 2. Regularized NMF and CNMF: cost functions and algorithms

NMF

$$\mathcal{L} = \frac{1}{2} \|\tilde{\mathbf{X}} - \mathbf{X}\|_2^2 + \mathcal{R}$$

$$\tilde{\mathbf{X}} = \mathbf{W}\mathbf{H}$$

$$\mathbf{W} \leftarrow \mathbf{W} \times \frac{\mathbf{X}\mathbf{H}^\top}{\tilde{\mathbf{X}}\mathbf{H}^\top + \frac{d\mathcal{R}}{d\mathbf{W}}}$$

$$\mathbf{H} \leftarrow \mathbf{H} \times \frac{\mathbf{W}^\top \mathbf{X}}{\mathbf{W}^\top \tilde{\mathbf{X}} + \frac{d\mathcal{R}}{d\mathbf{H}}}$$

CNMF

$$\mathcal{L} = \frac{1}{2} \|\tilde{\mathbf{X}} - \mathbf{X}\|_2^2 + \mathcal{R}$$

$$\tilde{\mathbf{X}} = \mathbf{W} \circledast \mathbf{H}$$

$$\mathbf{W}_{..l} \leftarrow \mathbf{W}_{..l} \times \frac{\overset{l \rightarrow \top}{\mathbf{X}} \mathbf{H}}{\tilde{\mathbf{X}} \overset{l \rightarrow \top}{\mathbf{H}} + \frac{d\mathcal{R}}{d\mathbf{W}_{..l}}}$$

$$\mathbf{H} \leftarrow \mathbf{H} \times \frac{\mathbf{W} \circledast \mathbf{X}}{\mathbf{W} \circledast \tilde{\mathbf{X}} + \frac{d\mathcal{R}}{d\mathbf{H}}}$$

L1 regularization for \mathbf{H} (L1 for \mathbf{W} is analogous)

$$\mathcal{R} = \lambda \|\mathbf{H}\|_1$$

$$\frac{d\mathcal{R}}{d\mathbf{W}_{..l}} = 0$$

$$\frac{d\mathcal{R}}{d\mathbf{H}} = \lambda$$

Soft orthogonality for \mathbf{H}

$$\mathcal{R} = \frac{\lambda}{2} \|\mathbf{H}\mathbf{H}^\top\|_{1, i \neq j}$$

$$\frac{d\mathcal{R}}{d\mathbf{W}_{..l}} = 0$$

$$\frac{d\mathcal{R}}{d\mathbf{H}} = \lambda(\mathbf{1} - \mathbf{D})\mathbf{H}$$

Smoothed soft orthogonality for \mathbf{H} (favors 'events-based')

$$\mathcal{R} = \frac{\lambda}{2} \|\mathbf{H}\mathbf{S}\mathbf{H}^\top\|_{1, i \neq j}$$

$$\frac{d\mathcal{R}}{d\mathbf{W}_{..l}} = 0$$

$$\frac{d\mathcal{R}}{d\mathbf{H}} = \lambda(\mathbf{1} - \mathbf{D})\mathbf{H}\mathbf{S}$$

Smoothed soft orthogonality for \mathbf{W} (favors 'parts-based')

$$\mathcal{R} = \frac{\lambda}{2} \|\mathbf{W}_{flar}^\top \mathbf{W}_{flar}\|_{1, i \neq j}$$

$$\frac{d\mathcal{R}}{d\mathbf{W}_{..l}} = \lambda \mathbf{W}_{flar} (\mathbf{1} - \mathbf{I})$$

$$\text{where } (\mathbf{W}_{flar})_{nt} = \sum_{\ell} \mathbf{W}_{nk\ell}$$

$$\frac{d\mathcal{R}}{d\mathbf{H}} = 0$$

Smoothed cross-factor orthogonality (main seqNMF \mathcal{R})

$$\mathcal{R} = \lambda \|\mathbf{W} \circledast \mathbf{X}\mathbf{S}\mathbf{H}^\top\|_{1, i \neq j}$$

$$\frac{d\mathcal{R}}{d\mathbf{W}_{..l}} = \lambda \overset{\leftarrow \ell}{\mathbf{X}} \mathbf{S} \mathbf{H}^\top (\mathbf{1} - \mathbf{I})$$

$$\frac{d\mathcal{R}}{d\mathbf{H}} = \lambda (\mathbf{1} - \mathbf{D}) \mathbf{W} \circledast \mathbf{X}\mathbf{S}$$

Table 3. Key resources

Software/algorithm	Source	Link to code
seqNMF	This paper	https://github.com/FeeLab/seqNMF
CNMF	[41, 40]	https://github.com/colinvaz/nmf-toolbox
Sparse CNMF	[30, 36]	https://github.com/colinvaz/nmf-toolbox
Soft orthogonal NMF	[7]	
Other NMF extensions	[9]	
NMF	[27]	
CNMF_E (cell extraction)	[48]	https://github.com/zhoupuc/CNMF_E
MATLAB	MathWorks	www.mathworks.com
Dataset	Source	Link to data
HVC, Isolate songbird	This paper	will upload to CRCNS after publication
Hippocampus, running wheel task	[32]	https://crcns.org/data-sets/hc/hc-3 and /hc-5
Other	Source	Link
Zebra finches (<i>Taeniopygia guttata</i>)	MIT animal facility	
AAV9.CAG.GCaMP6f.WPRE.SV40	[6]	https://pennvectorcore.med.upenn.edu
Miniature microscope	Inscopix	https://www.inscopix.com/nvista

803 Supplemental Figures

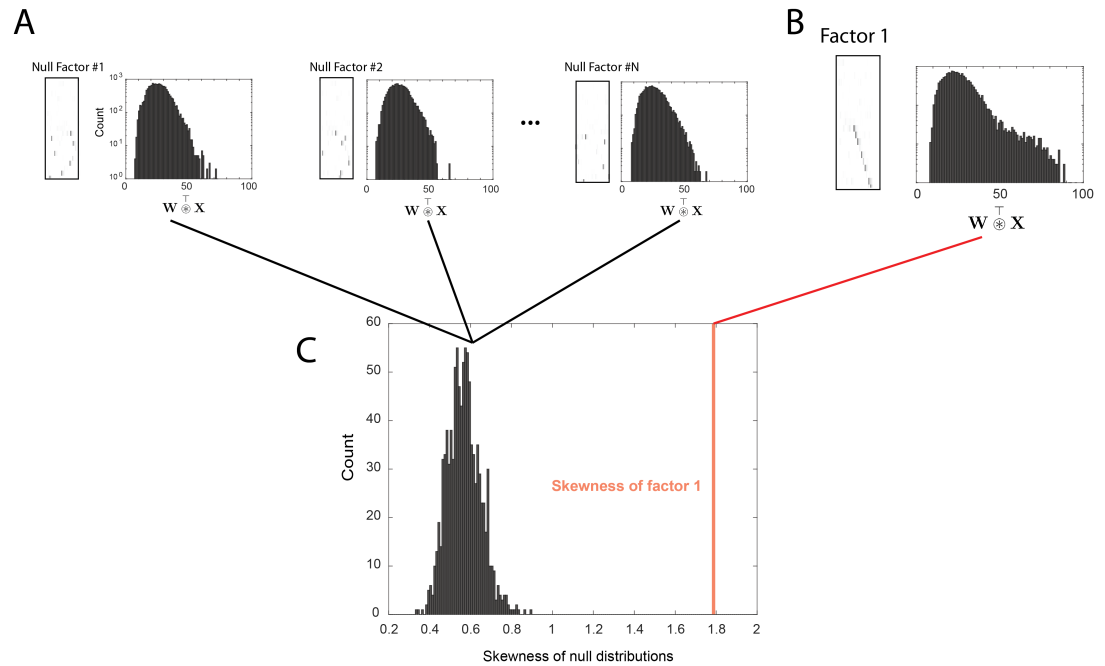


Figure S1. Outline of the procedure used to assess factor significance.

(A) In order to test the significance of a factor on held-out data, we constructed null (shifted) versions of the factor, and measured the distribution of overlap values ($\mathbf{W}^{\top} \mathbf{X}$) between each null factor and the held-out data. **(B)** We also measured the distribution of overlap values between the real factor and the held-out data. **(C)** We then compared the skewness of the actual distribution to the skewness of null distributions, and asked whether it was significantly higher than the null case.



Figure S2. Biasing factorizations between sparsity in \mathbf{W} or \mathbf{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 \mathbf{W} versus \mathbf{H} . Both use $K = 3$ and $\lambda = 0.001$. **(A)** Factorization achieved using additional smoothed soft orthogonality for \mathbf{H} , with $\lambda_{L1H} = 1$. **(B)** Factorization achieved using additional smoothed soft orthogonality for \mathbf{W} , with $\lambda_{L1W} = 1$.

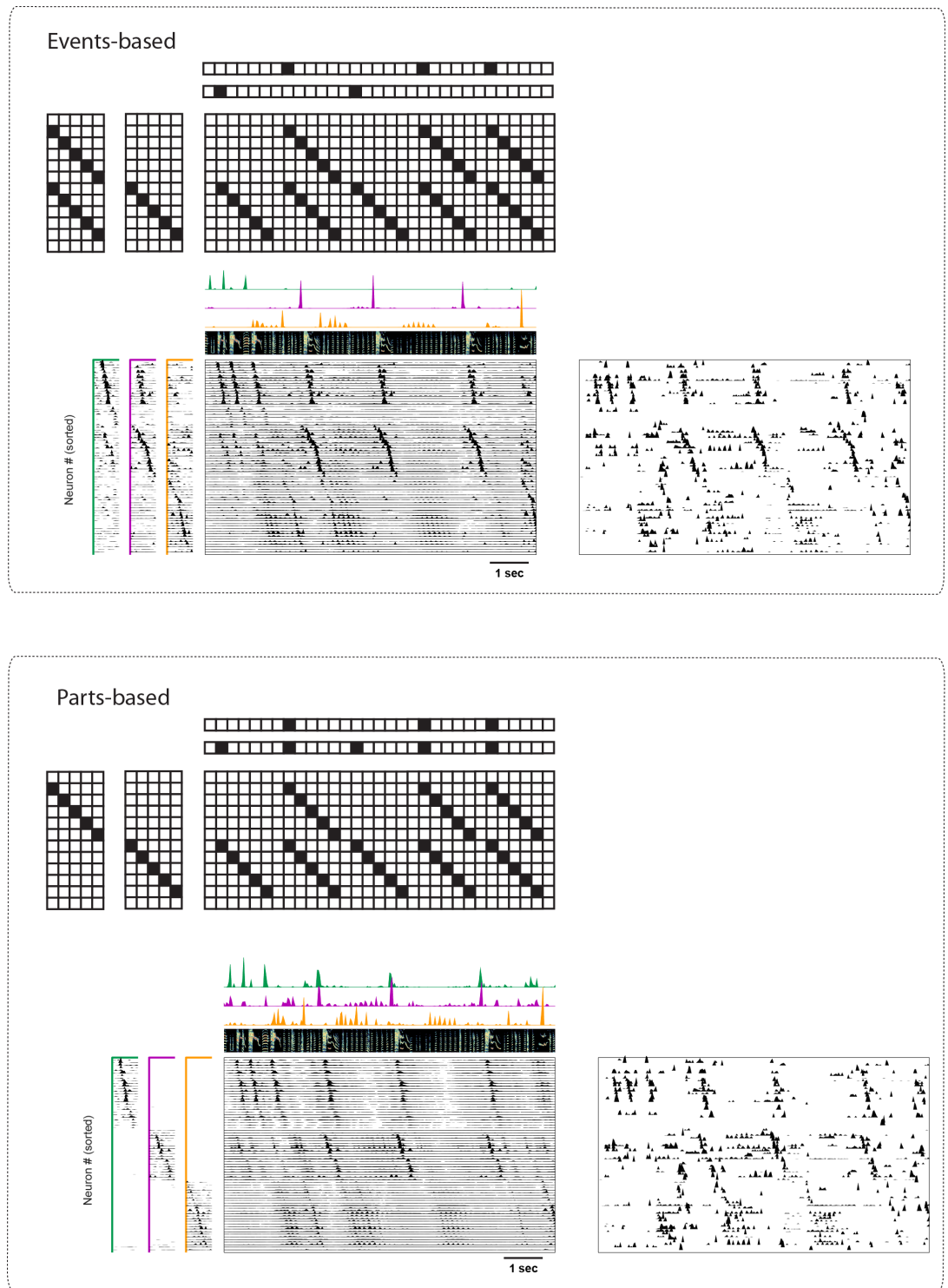


Figure S3. Biasing towards parts-based and events-based factorizations

Illustration of a trade-off between parts-based (\mathbf{W} is more strictly orthogonal) and events-based (\mathbf{H} is more strictly orthogonal) factorizations in a dataset where some neurons are shared between different sequences. The same data as in Figure 6 is factorized using smoothed soft orthogonality on \mathbf{H} (top, events-based), or on \mathbf{W} (bottom, parts-based). Below each motivating cartoon factorization, we show seqNMF fits (\mathbf{W} and \mathbf{H} together with the reconstruction) of the data in Figure 6. 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.

Appendix 1

Deriving multiplicative update rules

Standard gradient descent methods for minimizing a cost function must be adapted when solutions are constrained to be non-negative, since gradient descent steps may result in negative values. Lee and Seung invented an elegant and widely-used algorithm for non-negative gradient descent that avoids negative values by performing multiplicative updates [27]. They derive these multiplicative updates by choosing an adaptive learning rate that makes additive terms cancel from standard gradient descent on the cost function. We will reproduce their derivation here, and detail how to extend it to the convolutional case [41] apply several forms of regularization [30, 36, 7]. See Table 2 for a compilation of cost functions, derivatives and multiplicative updates for NMF and CNMF under several different regularization conditions.

Standard NMF

NMF factorizes data $\mathbf{X} \approx \tilde{\mathbf{X}} = \mathbf{WH}$. NMF factorizations seek to solve the following problem:

$$(\tilde{\mathbf{W}}, \tilde{\mathbf{H}}) = \arg \min_{\mathbf{W}, \mathbf{H}} \mathcal{L}(\mathbf{W}, \mathbf{H}) \quad (11)$$

$$\mathcal{L}(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 \quad (12)$$

$$\tilde{\mathbf{W}}, \tilde{\mathbf{H}} \geq 0 \quad (13)$$

This problem is convex in \mathbf{W} and \mathbf{H} separately, not together, so a local minimum is found by alternating \mathbf{W} and \mathbf{H} updates. Note that:

$$\frac{d}{d\mathbf{W}} \mathcal{L}(\mathbf{W}, \mathbf{H}) = \tilde{\mathbf{X}}\mathbf{H}^\top - \mathbf{X}\mathbf{H}^\top \quad (14)$$

$$\frac{d}{d\mathbf{H}} \mathcal{L}(\mathbf{W}, \mathbf{H}) = \mathbf{W}^\top \tilde{\mathbf{X}} - \mathbf{W}^\top \mathbf{X} \quad (15)$$

Thus, gradient descent steps for \mathbf{W} and \mathbf{H} are:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta_{\mathbf{W}} (\tilde{\mathbf{X}}\mathbf{H}^\top - \mathbf{X}\mathbf{H}^\top) \quad (16)$$

$$\mathbf{H} \leftarrow \mathbf{H} - \eta_{\mathbf{H}} (\mathbf{W}^\top \tilde{\mathbf{X}} - \mathbf{W}^\top \mathbf{X}) \quad (17)$$

To arrive at multiplicative updates, Lee and Seung [27] set:

$$\eta_{\mathbf{W}} = \frac{\mathbf{W}}{\mathbf{W}\mathbf{H}\mathbf{H}^\top} \quad (18)$$

$$\eta_{\mathbf{H}} = \frac{\mathbf{H}}{\mathbf{W}^\top \mathbf{W}\mathbf{H}} \quad (19)$$

Thus, the gradient descent updates become multiplicative:

$$\mathbf{W} \leftarrow \mathbf{W} \times \frac{\mathbf{X}\mathbf{H}^\top}{\mathbf{W}\mathbf{H}\mathbf{H}^\top} = \mathbf{W} \times \frac{\mathbf{X}\mathbf{H}^\top}{\tilde{\mathbf{X}}\mathbf{H}^\top} \quad (20)$$

$$\mathbf{H} \leftarrow \mathbf{H} \times \frac{\mathbf{W}^\top\mathbf{X}}{\mathbf{W}^\top\mathbf{W}\mathbf{H}} = \mathbf{H} \times \frac{\mathbf{W}^\top\mathbf{X}}{\mathbf{W}^\top\tilde{\mathbf{X}}} \quad (21)$$

where the division and \times are element-wise.

Standard CNMF

Convolutional NMF factorizes data $\mathbf{X} \approx \tilde{\mathbf{X}} = \sum_{\ell} \mathbf{W}_{\cdot\cdot\ell} \overset{\ell \rightarrow}{\mathbf{H}} = \mathbf{W} \circledast \mathbf{H}$. CNMF factorizations seek to solve the following problem:

$$(\tilde{\mathbf{W}}, \tilde{\mathbf{H}}) = \arg \min_{\mathbf{W}, \mathbf{H}} \mathcal{L}(\mathbf{W}, \mathbf{H}) \quad (22)$$

$$\mathcal{L}(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 \quad (23)$$

$$\tilde{\mathbf{W}}, \tilde{\mathbf{H}} \geq 0 \quad (24)$$

The derivation above for standard NMF can be applied for each ℓ , yielding the following update rules for CNMF [41]:

$$\mathbf{W}_{\cdot\cdot\ell} \leftarrow \mathbf{W}_{\cdot\cdot\ell} \times \frac{\mathbf{X}\mathbf{H}^\top}{\tilde{\mathbf{X}}\mathbf{H}^\top} \quad (25)$$

$$\mathbf{H} \leftarrow \mathbf{H} \times \frac{\sum_{\ell} \mathbf{W}_{\cdot\cdot\ell}^\top \overset{\leftarrow}{\tilde{\mathbf{X}}}}{\sum_{\ell} \mathbf{W}_{\cdot\cdot\ell}^\top \overset{\leftarrow}{\tilde{\mathbf{X}}}} = \mathbf{H} \times \frac{\mathbf{W}^\top \circledast \mathbf{X}}{\mathbf{W}^\top \circledast \tilde{\mathbf{X}}} \quad (26)$$

Note that NMF is a special case of CNMF where $L = 0$.

Incorporating regularization terms

Suppose we want to regularize by adding a new term, \mathcal{R} to the cost function:

$$(\tilde{\mathbf{W}}, \tilde{\mathbf{H}}) = \arg \min_{\mathbf{W}, \mathbf{H}} \mathcal{L}(\mathbf{W}, \mathbf{H}) \quad (27)$$

$$\mathcal{L}(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \|\tilde{\mathbf{X}} - \mathbf{X}\|_F^2 + \mathcal{R} \quad (28)$$

$$\tilde{\mathbf{W}}, \tilde{\mathbf{H}} \geq 0 \quad (29)$$

Using a similar trick to Lee and Seung, we choose a $\eta_{\mathbf{W}}, \eta_{\mathbf{H}}$ to arrive at a simple multiplicative update. Below is the standard NMF case, which generalizes trivially to the CNMF case.

Note that:

$$\frac{d\mathcal{L}}{d\mathbf{W}} = \tilde{\mathbf{X}}\mathbf{H}^\top - \mathbf{X}\mathbf{H}^\top + \frac{d\mathcal{R}}{d\mathbf{W}} \quad (30)$$

$$\frac{d\mathcal{L}}{d\mathbf{H}} = \mathbf{W}^\top\tilde{\mathbf{X}} - \mathbf{W}^\top\mathbf{X} + \frac{d\mathcal{R}}{d\mathbf{H}} \quad (31)$$

We set:

$$\eta_{\mathbf{W}} = \frac{\mathbf{W}}{\tilde{\mathbf{X}}\mathbf{H}^\top + \frac{d\mathcal{R}}{d\mathbf{W}}} \quad (32)$$

$$\eta_{\mathbf{H}} = \frac{\mathbf{H}}{\mathbf{W}^\top\tilde{\mathbf{X}} + \frac{d\mathcal{R}}{d\mathbf{H}}} \quad (33)$$

Thus, the gradient descent updates become multiplicative:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta_{\mathbf{W}} \frac{d\mathcal{L}}{d\mathbf{W}} = \mathbf{W} \times \frac{\mathbf{X}\mathbf{H}^\top}{\tilde{\mathbf{X}}\mathbf{H}^\top + \frac{d\mathcal{R}}{d\mathbf{W}}} \quad (34)$$

$$\mathbf{H} \leftarrow \mathbf{H} - \eta_{\mathbf{H}} \frac{d\mathcal{L}}{d\mathbf{H}} = \mathbf{H} \times \frac{\mathbf{W}^\top\mathbf{X}}{\mathbf{W}^\top\tilde{\mathbf{X}} + \frac{d\mathcal{R}}{d\mathbf{H}}} \quad (35)$$

where the division and \times are element-wise.

This framework enables flexible incorporation of different types of regularization into the multiplicative NMF update algorithm. This framework also extends naturally to the convolutional case. See Table 2 for examples of several regularization terms, including $L1$ sparsity [30, 36] and soft orthogonality [7], as well as the terms we introduce here to combat the types of inefficiencies and cross correlations we identified in convolutional NMF, namely, smoothed orthogonality for \mathbf{H} and \mathbf{W} , and smoothed cross-factor orthogonality, the primary seqNMF regularization term. For the seqNMF regularization term, $\lambda \|\mathbf{W} \circledast \mathbf{X}\mathbf{S}\mathbf{H}^\top\|_{1, i \neq j}$, the multiplicative update rules are:

$$\mathbf{W}_{.. \ell} \leftarrow \mathbf{W}_{.. \ell} \times \frac{\mathbf{X} \left(\overset{\ell \rightarrow}{\mathbf{H}} \right)^\top}{\tilde{\mathbf{X}} \left(\overset{\ell \rightarrow}{\mathbf{H}} \right)^\top + \lambda \overset{\leftarrow \ell}{\mathbf{X}} \mathbf{S} \mathbf{H}^\top (\mathbf{1} - \mathbf{I})} \quad (36)$$

$$\mathbf{H} \leftarrow \mathbf{H} \times \frac{\mathbf{W} \overset{\top}{\circledast} \mathbf{X}}{\mathbf{W} \overset{\top}{\circledast} \tilde{\mathbf{X}} + \lambda (\mathbf{1} - \mathbf{I}) (\mathbf{W} \overset{\top}{\circledast} \mathbf{X} \mathbf{S})} \quad (37)$$

Where the division and \times are element-wise. The operator $\overset{\ell \rightarrow}{(\cdot)}$ shifts a matrix in the \rightarrow direction by ℓ timebins, i.e. a delay by ℓ timebins, and $\overset{\leftarrow \ell}{(\cdot)}$ shifts a matrix in the \leftarrow direction by ℓ timebins (Table 1). Note that multiplication with the $K \times K$ matrix $(\mathbf{1} - \mathbf{I})$ effectively implements factor competition because it places in the k th row a sum across all other factors.