Supplementary note

Branching sequences

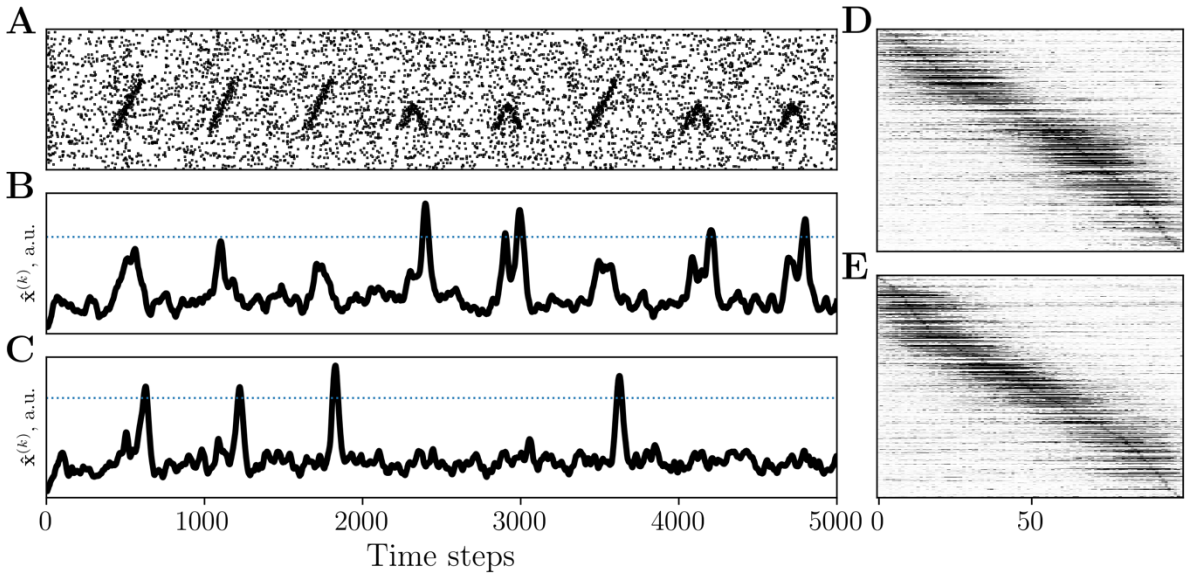


Fig. S1. An example of two sequences that completely overlap in space by half of the participating neurons. The sequences could be interpreted as one branching sequence in which half of the neurons always fire in one order, while its second half might proceed in the reverse direction.

Detecting sequences of place cells tiling a 2D enclosure

We consider a highly simplified hypothetical scenario in which we record the activity of 144 mouse place cells (with 2D gaussian place fields (std=10cm) that tile a 1.2x1.2m enclosure. The mouse runs in one direction on one of two trajectories (one from the top left to the bottom right corner, and the other from the top right corner straight to the opposing corner on the right).

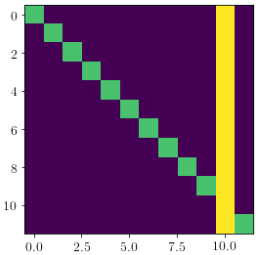


Fig. S2. *A 1.2 m by 1.2 m enclosure. The two trajectories are shown in green and yellow.*

As the mouse runs, the place cells spike with a probability proportional to how far the mouse is from the center of its place field. We build a synthetic spike dataset (of size 144 by 5000 timesteps in the following way. For each timestep, we simulate the movement of the mouse and based on where it is at this timestep, we calculate each neuron's firing probability and sample one spike according to that probability. Specifically, if the spike occurs on the neuron whose place field is centered at 40 cm down from the top and 40 cm from the right wall of the enclosure, we make a matrix of zeros set its element [3, 3] to 1, vectorize it and add it to a list. Repeating this process for T timesteps we obtain a matrix of size N x T, which after adding background noise and jittering the spikes in the time dimension looks like this:

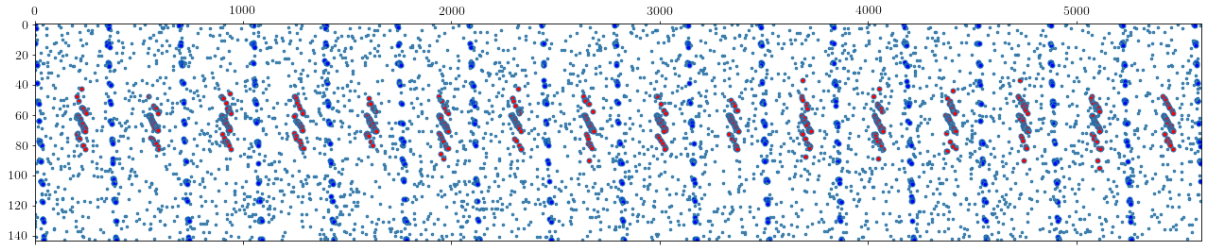


Fig. S3. *Synthetic sequences resulting from a mouse traversing place field tiling a 2D enclosure*

Notice that the sequences appear broken because the place fields adjacent in 2D are no longer adjacent after vectorization. Our method can detect the sequences successfully:

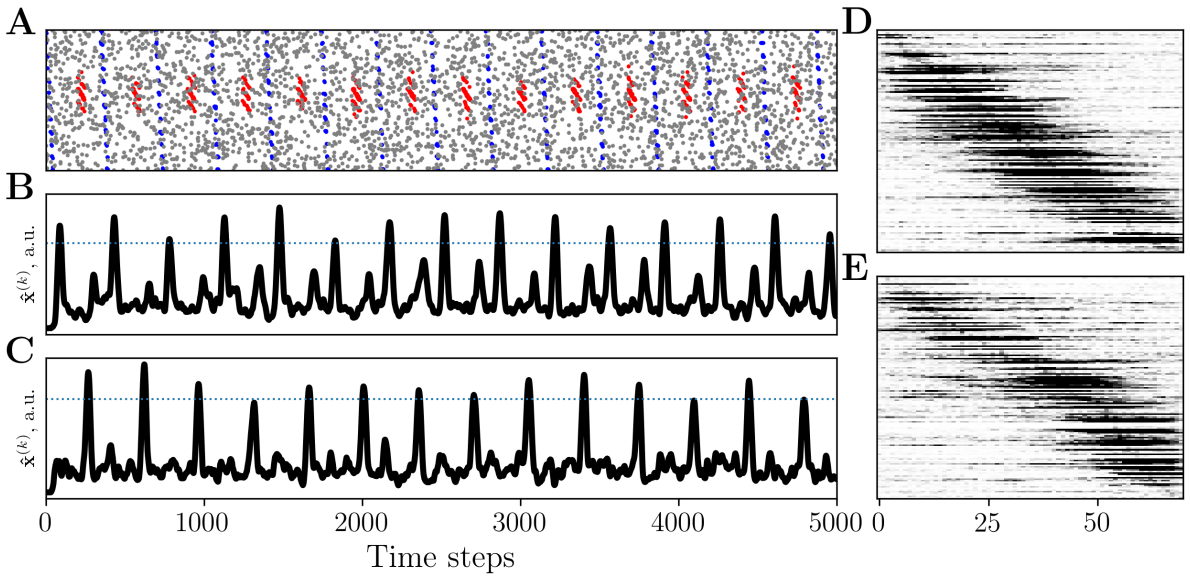
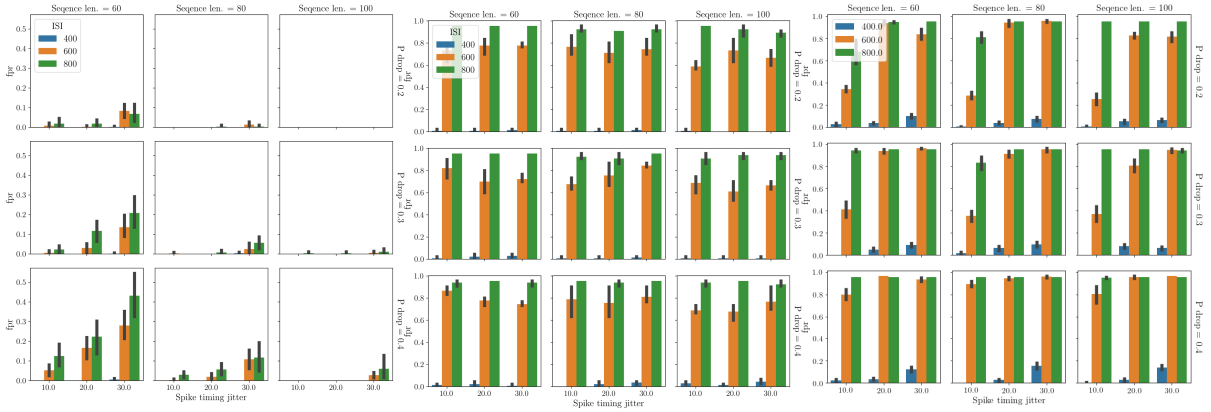


Fig. S4.

Comparing the TPR of PP-Seq and our method after matching for FPR

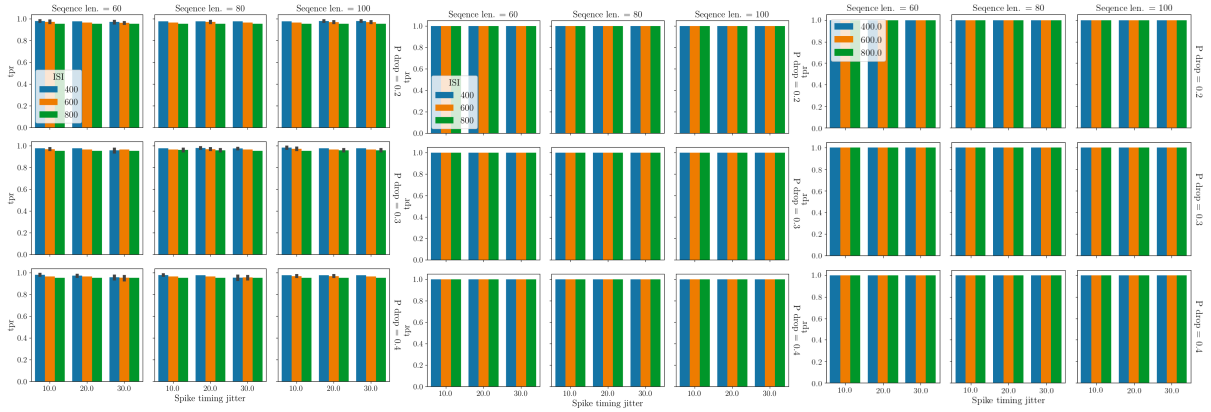
In addition to comparisons shown in Appendix B, we lowered the significance threshold (from 4 to 0.5 standard deviations) to approximately match the FPRs of our model and those of PP-Seq. Fig. S5 reproduces Figs. B.3 and B.4 (left and right panels, respectively) and additionally shows the FPR for the threshold of 0.5 sigma.



Ours (4σ) Ours (0.5σ) PP-Seq

Fig. S5. False positive rate.

Fig. S6 reproduces Figs. B.2 and B.4 (left and right panels, respectively) and additionally shows the FPR for the threshold of 0.5 sigma.



Ours (4σ) Ours (0.5σ) PP-Seq

Fig. S6. True positive rate.

Visualization of filter width over the data

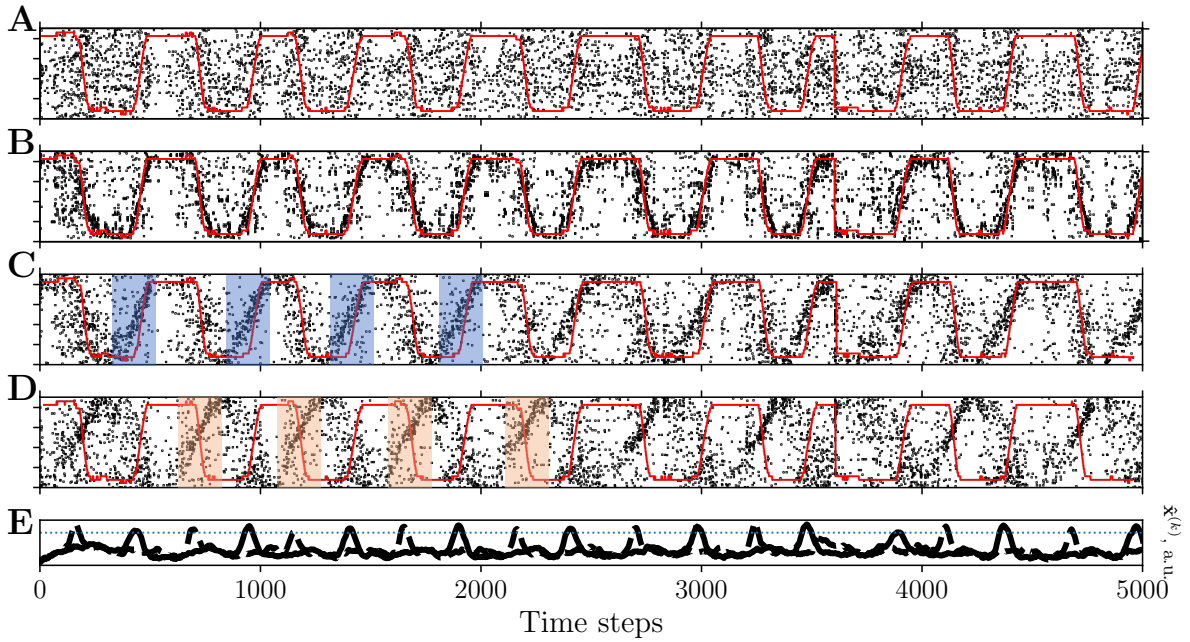


Fig. S7. *Sequences are contained within the temporal width of the filter (200 timesteps, shown as transparent rectangles).*

Detecting sequences unfolding at different time scales

It is possible for the same sequence (that is, a sequence involving the same neurons) to be expressed over more than one time scale. If one expects such sequences to be present in the data, we recommend running filter optimization with $K = 1$ with several progressively larger values of M. Smaller filters should become tuned to more "compressed" versions of the sequence, while larger ones will tend to capture slower versions of the sequence. We illustrate this on a synthetic dataset (with $N = 452$, dropout of 0.2, spike timing jitter = 15) with two sequences of 160 neurons, involving the same neurons but unfolding at different speeds (the second is 3 times slower than the first). We begin by setting $M = 200$ and optimize the first filter. As expected, the optimized filter only "responds" to the short sequence (Fig. S8), because the longer sequence is not fully contained within the filter's temporal length. Next, we set M = 600 and optimize the second filter. After optimization, this wider filter produces higher peaks in response to the longer sequence (Fig. S9).

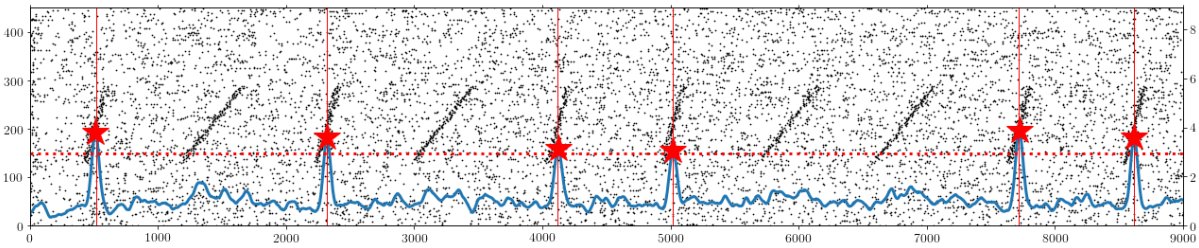


Fig. S8. *Convolution (blue curve) of the first filter (M = 200) with the data. Stars indicate significant detections.*

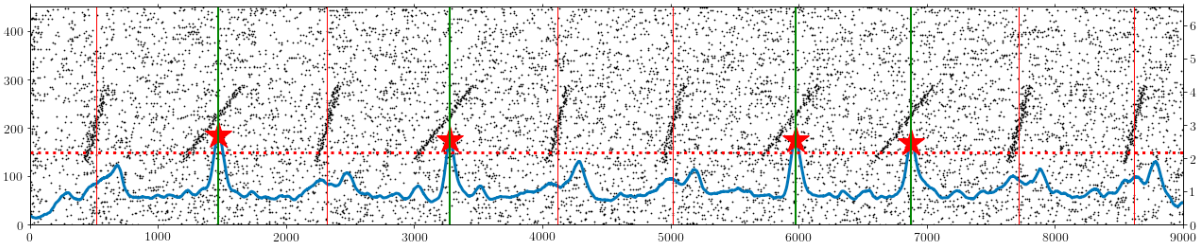


Fig. S9. *Convolution (blue curve) of the first filter (M = 600) with the data. Stars indicate significant detections.*