

Spike2Vec: Converting Spike Trains to Vectors to Analyse Network States and State Transitions

Russell Jarvis

Pablo de Abreu Urbizagastegui

Yeshwanth Bethi

Alexandre Marcireau

Abstract— A scalable algorithm that can detect fine grained repetitions quickly across large spiking datasets is desirable, as such a frame work would provide a means to test for the tendency of neuronal activity to revisit states.

Quickly identifying repeated states in large scale neuronal data and simulation is important, as the degree of repetition should influence the mindset of the scientists undertaking an analysis of spike trains. For instance several established cortical network models have assumed that realistic cortical neuronal activity should be Asynchronous and Irregular Activity (AI) in character such as Brunel’s balanced model of cortex [1].

By ascertaining repetitions in large spiking datasets we can quantify the frequency of repetition, and achieve a better understanding of a networks ability to revisit states. To this end we represented time bound neural activity as simple geometric coordinates in a highdimensional space. Working with geometric representations of chaotic spike train recordings may enable researchers to find a common set of RTSPs. In this work we show the beginning contributions of compiled Vectorized library of neuronal spike train recordings, that contains recordings from different individuals and also individuals belonging to different mammal species. By compiling a Vectorized library of neuronal spike train recordings, we will be able to transform vectors by swapping axis order of the constituent vectors to find the vector axis order that maximises overlap between replayed events from different individuals.

The most basic aspect of approach of the spike2vec framework is very old, under the established and conventional technique a population of spike trains is converted to a series of spike firing rate encoded vectors. This idea has been around since the late 80s [2]. However this work departs from the conventional approach as we decided to construct vectors from the neurons instantaneous pattern of spiking variability in contrast to instantaneous firing rate, somewhat analogously

to sampling each neurons local variation as it evolves through time. When the neurons instantaneous spike time variability is measured it is entered into a population vector. As network behavior evolves through time, we are able to re-examine collections of these vectors to find approximately repeating temporal spatial patterns. When we have located a repeating temporal spatial patterns we are then able to consider each RTSP as a bag of Inter Spike Intervals, where the spatial structure of spike sources is discarded, and RTSPs overall distributions of RTSPs are analysed between different individuals of the same species to find out if there is universal temporal profiles of rodent RTSPs, in approach somewhat inspired by the work [3]

When it comes to electrical neuronal recordings of the mammal cortex, there is a risk of underestimating the frequency of neuronal replay events and overestimating the frequency of unrepeating random looking events. Repeating Temporal Spatial Patterns (RTSPs) aka sequences, and motifs are rarely observed in important electrical neuronal cortical recordings of sensory neurons [4], and older recordings of neocortex in rodents and primates. If the prevalence of repetition and replay was thoroughly characterized, cortical models could then appropriately increase the amount of structured coherent repeating patterns they generate and decrease the amount of Poisson process random noise activity represented in simulation outputs.

The reasons why there is a risk that RTSP is going unrecognized in classic neuronal recordings of spike trains will be discussed below, as most of these risks are avoidable. The first reason is that many electrical neuronal data recordings are only 3 seconds or less. Three seconds is a tiny window of opportunity, and it is not a reasonable duration for spike patterns to be repeated in a recording.

Reason two: Data analysis which was applied to the data, didn’t necessitate replay detection, as the replay wasn’t relevant to the analysis framework; also, if no significant pattern is perceptible to humans, the lack of curiosity seems justified.

Reason three, the final reason: Replay detection might fail because a dedicated detector fails to detect the RTSP. Some types of temporal and spatial patterns can defeat replay detectors. Also, some neuronal recordings may be so big that they render applications with results inaccessible because replay detection could not happen in a human-relevant time scale.

There is a caveat that 3 seconds might be plenty of time to see a repeat of the experimental paradigm is construed to elicit replay such as in [5]. If replay corresponds to memory recall, as some authors have suggested, it seems reasonable that RTSPs might take days to years to reoccur.

There may be latent evidence for the Network Attractor theory of neuronal learning in older spiking data sets, and it is imperative to uncover any evidence for the attractor network theory in the vast amount of public spiking data on the internet. Methods are needed to transform spike raster plots into attractor trajectories directly into state transition networks, and it is unclear if the existing implementations of SPADE can achieve this in a performant and large scale manner.

Furthermore there is a desire to perform neuronal data analysis in a manner that lends itself to extensibility with Julia, a modern language with high performance features, when the Julia ecosystem has a new framework, the combinations of performance orientated tools becomes possible.

An pre-existing theory [6], the attractor network theory of the mammal cortical dynamics is consistent with phenomenological observations about the mind, such that people commonly refer to “circular thinking”, in obsessive compulsive disorder. Furthermore action and perception are theorized to occur in alternating cycles, during “action-perception” loops. Neuronal synaptic weight changes, that happen as a result of STDP, simply bias the brain in a manner which will make salient brain states more likely to occur.

Both temporal and rate codes fail to fully explain how the cortical neurons of mammal brains are able to use synapses to learn about the world. The attractor network approach to understanding the brain is code agnostic. Deterministic chaotic principles are used to explain spike time irregularity. Spike time network attractors can explain why some of the brains activity patterns are able to repeat, and why the not all cortical activity is highly Asynchronous Irregular Activity (AI). The network attractor theory is compatible with recent spike train datasets that demonstrate “replay” in the hippocampus and prefrontal cortex of rats. Replay refers to a phenomena were in two time points of a spike train, there is macroscale similarity between spike patterns, and each pattern is generally reconizable, and regarded as approximately the same by human observers.

There is demand for a scalable algorithm that can detect repeating temporal spatial features in biological and synthetic data sets of cortical neuronal networks. An existing tool Spike

Pattern Detection and Evaluation (SPADE) [7, 8, 9], can detect repeating spatial temporal sequences, however it is only available in the Python language. Application of SPADE so far does not fully reconcile the attractor network theory of neuronal learning with spike train analysis. Reconciling the attractor network theory with in-silico and in-vivo spike trains is a prominent feature of this work.

An un-answered question is their commonality in the size and shape of RPS between similar but different mammal brains. When electrodes are inserted into different brains, the spatial location of recording electrodes, never reaches the same target twice. Variability in the structure of brains is a rule, and when comparing two neurons that occupy roughly the same position, but in different individual brains. It is rarely justified to consider the identity of cortical neurons would not be considered to be the same entity. Nonetheless it remains viable and desirable to transform vectors derived from spike trains into a common set of coordinates, by finding the exact sequence of vector axis that maximises overlap between repeated patterns. Analogous to word2vec word embedding models that were famously used in recommendation systems. RTSPs can be compared against spike2vec embedding models, however, before this could happen, spike2vec vectors would need to be transformed to a common coordinate system, where spike train vectors from different individuals would be reconcilable with each other. The word2vec approach was found to generalize to other realms such as product recommendations using meta data [10], and human EEG recordings. In this work we show the beginning contributions of compiled Vectorized library of neuronal spike train recordings, that contains recordings from different individuals and also individuals belonging to different mammal species. By compiling a Vectorized library of neuronal spike train recordings, we will be able to transform vectors by swapping axis order of the constituent vectors to find the vector axis order that maximises overlap between replayed events from different organisms.

Quickly identifying repeated states in large-scale neuronal data and simulation is essential, as the degree of repetition should influence the mindset of scientists analyzing spike trains. For instance, several established cortical network models have assumed that realistic cortical neuronal activity should be Asynchronous and Irregular (AI) in character. Two common models of cortical spiking networks are the, Potjan’s and Diesmon [11] model and the Brunel model [1], both of these models exist within a fluctuation driven regime. When each of these respective network models are simulated, observed spike times are typically appear to be poisson distributed psuedo random spike times. By design these models make it unlikely that fine grained recognizable repeating patterns also occur. The Potjan’s model can be used to make data points seperable. However, new data sets prominently capture

replayed states, and previously collected spike trains may, too, have latent and unpublished states of replay. The limited recordings from limited species may have biased previous recordings in a way that underrepresented the prevalence of replay.

After surveying the literature, we found evidence of two-three major replay detection algorithms, SPADE is written in Python, and FAST, which is advertised as “a spike sorting algorithm” is written in MATLAB [12]. We also found neuronal spike train recordings which are either long term, or bordering on chronic [13], [14]

Multivariate approaches to spike train network analysis often involve inferring a causal network diagram from spiking activity. Where the idea is that the spiking activity contains many hidden network states that emerge in large scale simulations, and that are not visible in the networks static connectome. To analyse effective connectivity in *in silico* and *in vitro* networks, spike train recordings are divided into time windows, and analysis compares previous (lagged time), with current time. Exhaustive pairwise iteration of pair-wise and or multivariate statistics is not computationally tractable at the scale of billions of neurons, and adding time lagged analysis of network cross-correlation, or transfer entropy makes the prospect of scaled temporal analysis even worse.

Under the dynamic systems view of the brain neuronal memories are analogous to attractor basins [15], [6]. Consider replay as a mechanism that is related both memory encoding and memory recall. If the view of memories as basins is correct then it should be possible to demonstrate synaptic learning as the mechanism that encodes memories as basins. Network attractor basins may be derived from the interleaved application of Spike Timing Dependent Plasticity (STPD) and sleep when synapses are able to change in a way that strongly biases some future spiking activities towards repeating observed patterns.

Application of STDP learning within to cortical models of Leaky Integrate and Fire (LIF) neurons a simple method to optimise network parameters a way that maximises the networks capacity to encode and revisit attractor states. Although spike trains are often analysed for information entropy at the spike to spike level. A spike2vec algorithm will enable researchers to investigate the spike trains states for redundancy at the sequential pattern level, in a Julia compliant way, and this will parallel efforts to develop Julia cortical simulators which intend to model prefrontal cortex in a data driven way.

I. THEORETICAL FRAMEWORK

Representational similarity [16] has been applied to decoding visual representations from ECG channels, by analysing the differences between channels, and how these differences evolve over time. Rather than applying representational similarity between pairs of neurons in the network, we instead

compare spike train distance across one neurons evolving spiking behavior, by assessing how much the neuron changes its representation.

To compute vectors from spike trains we apply representational dissimilarity between different windows of the neurons history, ie we evaluate spike train distance across one neurons evolving spiking behavior, by progressively evaluating how much the neuron deviates from uniform spiking in incremental windows.

A problem with converting spike train raster plots to attractor trajectories, is the that the most established method deriving attractor trajectories (and energy landscapes) requires the system under investigation to be encoded as a continuous differentiable function. A dominant approach which satisfies the continuous function requirement is to fit a differential equation that models a networks firing rate(s) in response to current injection the assumption underlying this approach, is that the rate coded information and network states are more important than or even exclude temporal codes.

Another approach to estimating attractor trajectories involves applying Delay Coordinate Embeddings framework. The advantage of this approach is that a model equation is not required, and a timeseries of system observations satisfies the algorithms requirements. Spikes time raster plots are sparsely encoded collections of events that are naturally encoded by ragged arrays, and delay coordinate embeddings requires a state space map. Vector matrices that are output from spike2vec are sufficient to satisfy Delay Coordinate Embeddings, however, the frame work is slow to evaluate, and the quality of the output of the algorithm dependent on many parameters (both in parameters of spike2vec and DCE).

Yet another approach is to use Recurrence Analysis. Recurrence Analysis is orders of magnitude faster than DCE, and the results of DCE usefully describe the network properties of state transition matrices. In order to find an optimal time window we could use consistently between data sets, we swept through a range of window lengths (ms), and found the window length which would maximise the correlation between peristimulus time histograms on exemplar spike raster patterns.

In order to test that the “auto spike train distance”, metric lead to more well defined network descriptors than other similar but more common metrics, We compared state vectors that were constructed by applying auto-covariance and local variation to the same spike windows, and we compared the spike2vec algorithms performance across all three metrics. [17] Julia simulation of learning. We simulated MNIST learning. [18] Dynamics systems view of the brain.

As an experiment we used a Julia package Emeddings.jl to convert spike train sequences to English words, by iterating

over word embedding vectors in large word2vec models, and finding closely matching vectors, such that we could apply a statistical analysis of written english to spike train recordings.

II. METHODOLOGICAL FRAMEWORK

Data sources analysed.

Zebra Finch song bird	chronic multi-day recording	NMIST	Prefrontal Cortex Re-play
[19]	[13]	[20]	[5]

Spike Train state descriptors tried:

Kreuz Distance against uniform ISI reference	Autocovariance	Coefficient of Variation	Local Variation
385.1	263.8	271.1	459.7

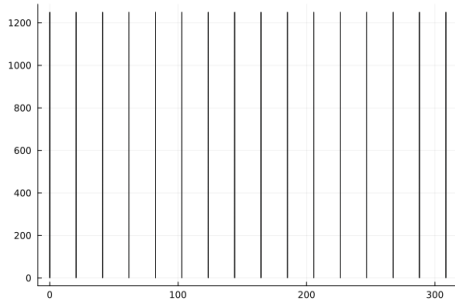
In order to verify that our particular application of the Kreuz spike distance metric lead to the most unique network state descriptions, we also constructed population vectors by applying other measurements of spike train variability. Matrices were populated by time varying vectors derived by applying the above metrics to some of the data sources. Complexity of applied spike2vec matrices, was computed by first applying columnwise normalization to the matrices, and then summing the total covariance contribution to get a scalar metric representing the complexity of the heatmap surface. Although the Local Variation metric had the highest metric for the sum of covariances, the LV matrix generally lacked repeating temporal structure, so it was not useful.

1) *Spike2Vec Algorithm Details*: The spike2vec frame work exists at the meta level. It is a novel mashup of pre-existing algorithms, its steps are as follows:

1. Spike trains are divided into N equally sized time windows.
2. In each window spike times are converted by subtracting the window start time, such that spike time variability is now mapped onto the local time frame in the smaller scope of each window (ie the time each window occurred is subtracted from each window, making any variation of spike times inside the window relevant to the windows scale). Each of these windows is then stored in an array.
3. The maximum firing rate of all the windows is found.

4. A single artificial spike train in a window is constructed by taking the maximum firing rate from step 3. And constructing a spike train that has regular Inter Spike Intervals (ISIs) occurring at the maximum firing rate. We call this the reference window, or surrogate.
5. For every N windows sampled in 1, the observed spike times is compared to the uniform reference window using the Thomas Kreuz spike Distance algorithm implemented in Julia by George Datseris. <https://github.com/JuliaNeuroscience/SpikeSynchrony.jl/commits?author=Datseris>
6. The Kreuz spike distance is a way of measuring the cost of converting observed spike train **A**, to a different spike train **B**. By measuring the Kreuz spike distance between a variation free regular spiking window, and a window with observed spike time variability, we get a picture of each neurons current unique local variability at each window (note that the method for comparing reference to observed doesn't have to uniquely encode unique spike sequences, it just has to be sufficiently unique to make states appropriately distinguishable but also recognizable across a population of multiple cells). As there are **M** number of neurons we then build a vector of coordinate of **M** dimensions, at each of N time windows. **X_m**, is an M by N tensor consists of M neurons and N time windows.
7. Since each column vector of **X_m** encodes a time window, we get the euclidian distance between each column vector and every other column vector, across the columns of the whole matrix.
8. We take these new distance values we fill a new matrix, between every window, and every other window at row and column location of the matrix. It's important to recognize that here we are not comparing spike distances between neurons (as has occurred in established work, we are comparing spike train distance vectors within the same neurons along time).
9. We perform unsupervised clustering on this temporally encoded dissimilarity matrix.
10. We discard all cluster labels that correspond to just a single time window, and retain the set of cluster labels, that have at least one repeating label. We regard these duplicated cluster labels as repeated temporal spatial patterns.

In the figure below we show the a reference window of uniform spikes described above.



A plot of the regular periodic spike reference window. A unvarying uniform surrogate spike train is used as a comparison in order to compute the transformation cost of transforming spike train uniform to spike train varying.

2) Common Coordinates of Spike2Vec Algorithm Details:

The output of Spike2vec, is a reduced set of vectors. Only replayed events are encoded as vectors, and everything else that can be regarded as a non repeating or a state transition is disregarded.

1. When each individuals recorded spike session is encoded as a matrix of column spike2vec vectors, between pairs of matrices, iterate over pairs of full recording matrices where each matrix is derived from one of two different individuals.
2. Consider pairs of matrices between each of all possible individuals.
3. At each step find the a reorganization of the sequence of vector axis, which maximises the commonality between each pair.

3) Discussion:

It is possible that the windows which were disregarded because they didn't repeat, may well repeat given long enough neuronal recordings. This is an unavoidable problem, caused by fact that only a limited duration of recording data is: a. available and b, computationally tractable to analyse. States and state transitions cannot be distinguished from each other.

In the absence of infinite recording data, we can draw no conclusions about whether one of observations would have re-occurred given more time. Another caveat is that inside this frame work it is impossible to distinguish between brain states that are relevant to the organism, and state transitions, which may also look similar from trial to trial.

Each conscious recall of a memory may involve the brain approximately repeating a pattern; ie recall may mean re-visiting a previous pattern of neuronal activity. Each recalled memory may retain significant traces of activity that is well correlated with the brain experience that caused the memory to be encoded. Such "replay" is has been observed in the hippocampus and prefrontal cortex in rats during sleep.

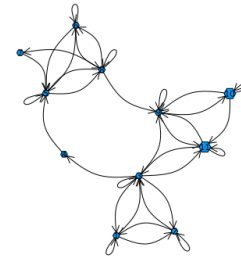
When replayed events are detected a sequential map of states can be derived from a spike train, and unrecognized state transitions and anomalies can also be sorted and labelled in discrete chunks.

Under spike2vec framework, spike patterns which are approximately the same as patterns in previous windows are detected because, in the geometric coordinate representation of vectors, spike trains which are close together will have be seperated by a small Euclidean Distance in the vector space.

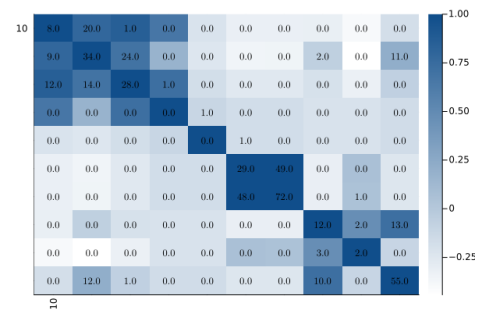
The spike2vec frame work can be used to convert spike trains to markov transition matrices, or simply state transition network maps. In such a state network, we can see that not all spike trains are equally stateful, some emperical recordings may have few replay events. When a spike recording is particularly stateful, there may be certain states which are only entered into given the appropriate sequence of prior states.

III. RESULT ANALYSIS

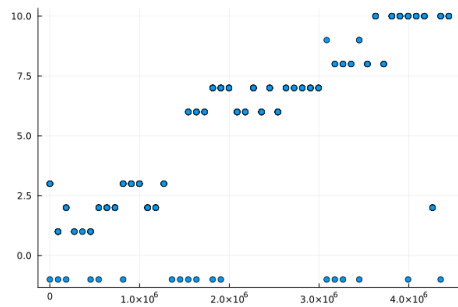
A. Evidence for Network Attractor Theory



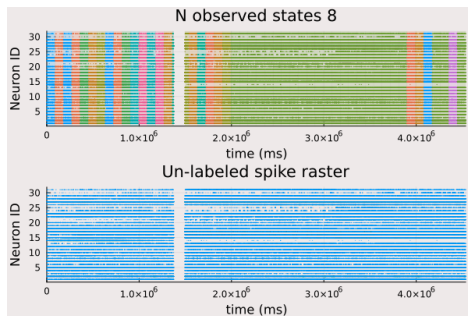
The output of the framework is a sequential state transition network of the spike train. Spontaneous network activity which didn't get repeated was simply not included in the state transition diagram. Two state transition diagrams are output, one with non repeating states, and one with repeating states.



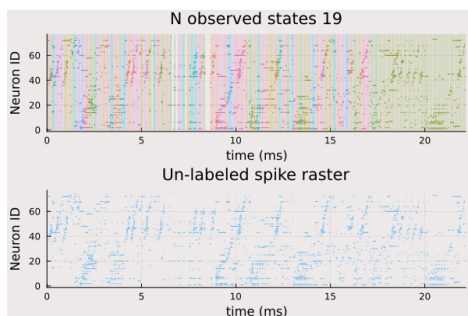
cluster_horizontal_vectors_sort_song_birds.png



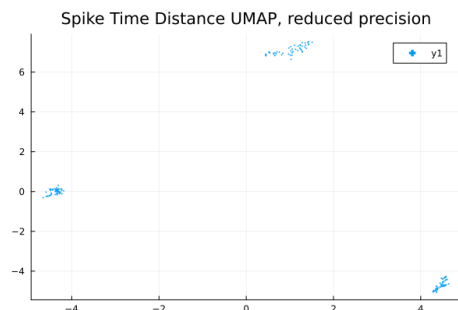
cluster_horizontal_vectors_sort_song_birds.png



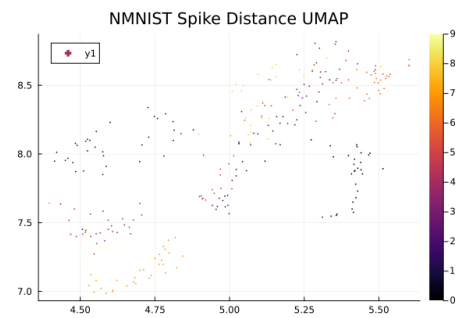
A scatter plot of state transition trajectories in the pre-frontal cortex spike train.



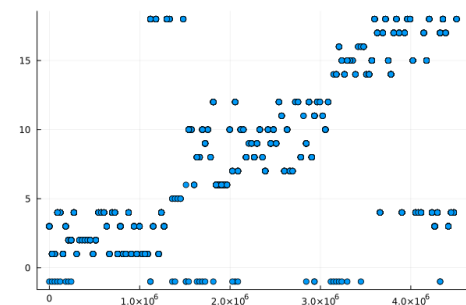
cluster_horizontal_vectors_sort_song_birds.png



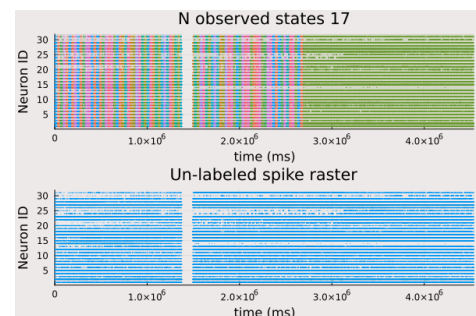
cluster_horizontal_vectors_sort_song_birds.png
align(center + bottom)[



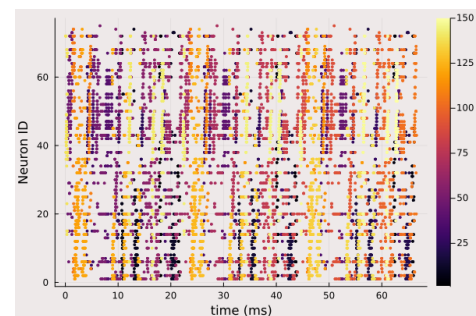
As a matter of routine UMAP dimensional embedding of spike distances was applied to all spike difference vectors in the matrix, since population level spike train local variation evolves over time. UMAP of the spike distance matrix, UMAP of the spike2vec matrix should allows data points to cluster in a time dependent manner.]



cluster_horizontal_vectors_sort_song_birds.png

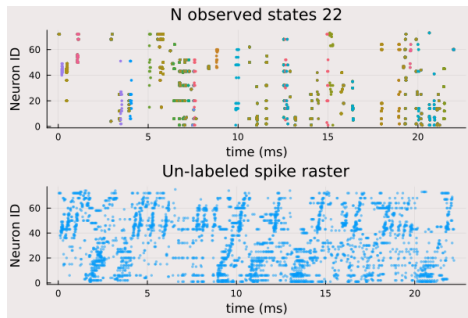


cluster_horizontal_vectors_sort_song_birds.png

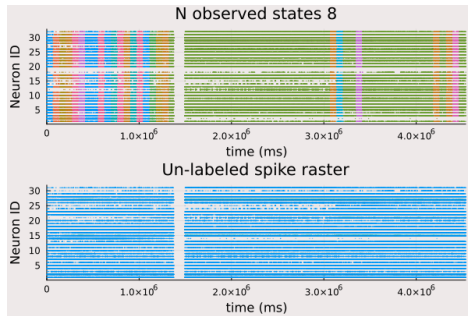


In order to test if the spike2vec framework worked as expected, we downloaded a

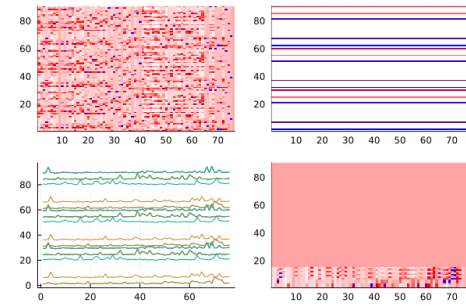
alcium imaging recording from Zebra finch (a song bird's) High Vocal Centre (brain region) source [19]. Although the actual data source was from (https://github.com/lindermanlab/PPSeq.jl/blob/master/demo/data/songbird_spikes.txt) The downloaded data set was then simply augmented, by duplicating the spike time raster plot in a manner that appended the full repeated recording to the end of the first recording, the process was iterated 3 times yielding a highly repititive data set 4 times the length of the original. The intention of this exercise was simply to show that spike2vec could identify and label such obvious repeating patterns.



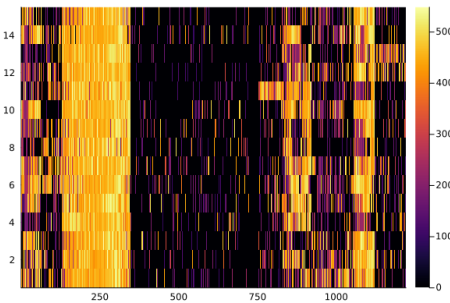
We take the single un augmented Zebra finche data and label repeating patterns using the discussed frame work, since there are no obvious explicit repitions caused by duplicating spike patterns, the algorithm is forced to consider the similarity between more disparate population level patterns.



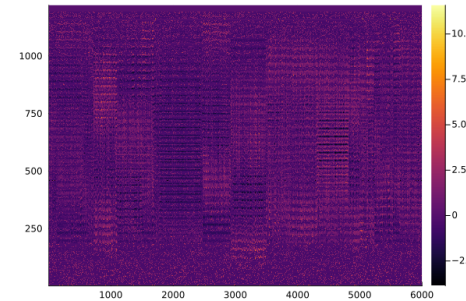
We took a data set from rat prefrontal cortex. The data set was captured, under experimental conditions explicitly designed to maximise the probability of recording replay ie dreaming and slow wave sleep.



Figures from left top to bottom right: A: Top left: 75 MNIST channels were recorded and time binned in a manner which yielded 85 vectorized time bins. Bottom right: Once the vectorized time bins had been vectorized, a clustering algorithm was applied to the entire matrix of vector coordinates. Cluster centres could then be used as reference points, such that it was possible to compare all

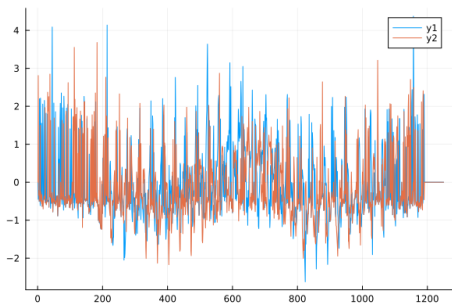


Normalized prefrontal cortex vectors for over 1000 time steps 15 neurons



We sampled the MNIST data set more broadly using 6,000 samples to create vectors. The entire data set consists of 60,000 samples.

align(center + bottom)[



Two spike time encoded numerals, were read in to the Julia-lang namespace, then the spiking neuromorphic data were converted to vectors over 1200 channels. Orange and Blue plots are vectors corresponding to two distinct MNIST data labels. Rapid positive alternating deflections are visible in both vectors, because the MNIST data is caused by pixel activations, when 2D pixel derived data sources are converted into a 1D vector, sparse clusters of activated pixels, have regular gaps between them. Herein lies a heatmap of dissimilarity matrices constructed using the MNIST dataset, ie the heatmap above, comes from analysing spike train distance across the MNIST data set numbers: 0-9 represented as spiking events. There are 300 total presentation number presentations. All nine numbers are incrementally cycled through. Number presentations within the one number are contiguous, (the dataset is not shuffled), and this contiguity is reflected in the heatmap too.]

REFERENCES

- [1] N. Brunel, "Hebbian learning of context in recurrent neural networks," *Neural Computation*, vol. 8, no. 8, pp. 1677–1710, 1996.
- [2] A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. ii. coding of the direction of movement by a neuronal population," *J. Neuroscience*, vol. 8, no. 8, pp. 2928–2937, 1988.
- [3] J. Pérez-Ortega, J. Araya, et al., "Parallel processing of natural images by overlapping retinal neuronal ensembles," *Biorxiv*, pp. 2021–2, 2021.
- [4] Y. N. Billeh, B. Cai, et al., "Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex," *Neuron*, vol. 106, no. 3, pp. 388–403, 2020.
- [5] A. Peyrache, M. Khamassi, K. Benchenane, S. I. Wiener, and F. P. Battaglia, "Replay of rule-learning related neural patterns in the prefrontal cortex during sleep," *Nature Neuroscience*, vol. 12, no. 7, pp. 919–926, 2009.
- [6] C. Eliasmith, "Attractor network," *Scholarpedia*, 2007.
- [7] A. Stella, P. Quaglio, E. Torre, and S. Grün, "3d-spade: significance evaluation of spatio-temporal patterns of various temporal extents," *Biosystems*, vol. 185, p. 104022, 2019.
- [8] P. Quaglio, A. Yegenoglu, E. Torre, D. M. Endres, and S. Grün, "Detection and evaluation of spatio-temporal spike patterns in massively parallel spike train data with spade," *Frontiers Comput. Neuroscience*, vol. 11, p. 41, 2017.
- [9] S. Shinomoto, K. Miura, and S. Koyama, "A measure of local variation of inter-spike intervals," *Biosystems*, vol. 79, no. 1-3, pp. 67–72, 2005.
- [10] F. Vasile, E. Smirnova, and A. Conneau, "Meta-prod2vec: product embeddings using side-information for recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 225–232.
- [11] T. C. Potjans, and M. Diesmann, "The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model," *Cerebral Cortex*, vol. 24, no. 3, pp. 785–806, 2014.
- [12] A. K. Dhawale, R. Poddar, et al., "Automated long-term recording and analysis of neural activity in behaving animals," *Elife*, vol. 6, 2017.
- [13] J. Pérez-Ortega, T. Alejandro-García, and R. Yuste, "Long-term stability of cortical ensembles," *Elife*, vol. 10, 2021.
- [14] J. L. Riquelme, M. Hemberger, G. Laurent, and J. Gjorgjieva, "Single spikes drive sequential propagation and routing of activity in a cortical network," *Elife*, vol. 12, 2023.
- [15] H. Lin, C. Wang, et al., "A review of chaotic systems based on memristive hopfield neural networks," *Math.*, vol. 11, no. 6, p. 1369, 2023.
- [16] T. Grootswagers, A. K. Robinson, and T. A. Carlson, "The representational dynamics of visual objects in rapid serial visual processing streams," *Neuroimage*, vol. 188, pp. 668–679, 2019.
- [17] B. Illing, W. Gerstner, and J. Brea, "Biologically plausible deep learning—but how far can we go with shallow networks?," *Neural Networks*, vol. 118, pp. 90–101, 2019.
- [18] C. M. Kim, U. Egert, and A. Kumar, "Dynamics of multiple interacting excitatory and inhibitory populations with delays," *Physical Rev. E*, vol. 102, no. 2, p. 22308, 2020.
- [19] E. L. Mackevicius, A. H. Bahle, et al., "Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience," *Elife*, vol. 8, 2019.
- [20] G. K. Cohen, G. Orchard, et al., "Skimming digits: neuromorphic classification of spike-encoded images," *Frontiers Neuroscience*, vol. 10, p. 184, 2016.