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

Dr Russell Jarvis

Mr Pablo de Abreu Urbizagastegui

Mr Yeshwanth Bethi

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 repitition should influence the mindset of the scientists undertaking an analysis of spike trains. Forinstance several established cortical network models have assumed that realistic cortical neuronal activity should be Asynchronous and Irregular Activity (AI) in character. Often simulatations of such activity using [1] Brunel's balanced model of cortex. However, new data sets prominently capture replayed states, and previously collected of spike trains may too have hidden states of replay. The limited recordings from limited species may have biased previous recordings in a way that under represented the prevalence of replay. recordings or otherwise have been misleading, as too their limited ability to capture replay, and detect it in analysis.

Although the dynamic systems view of the brain is old, a survey of spiking datasets which can detect and labels network attractor states in large spike count data is merited, as this would bolster the dynamic systems view of the neuronal learning.

By quantifying repetitions large spiking datasets, using geometric representations of complex spike patterns, we can quantify the frequency of repitition, 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 interrogate the state-fullness of both biologically recorded spike trains and their digitally simulated counterparts. Furthermore, there is reason to believe that when mammal brains enact visual object recognition encoded memories guide cortical neurons to "replay" previously observed neural states, as replayed spiking states may cohere with the visual brains perceptual recognition of a familiar scene.

I. Introduction

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. By using chaos attractor trajectories as an explanation the dynamic systems view of the brain can at least explain why some of the brains activity patterns are able to repeat, and why the not all brain activity is high entropy, asychronous irregular activity. The network attractor theory is compatible with recent spike train datasets that demonstrate "replay" in the hippocampus and prefrontal cortex of rats.

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 SPADE [2, 3, 4], can detect repeating spatial temporal sequences, however it is only available in the Python language, and it 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 prominant feature of this work.

There may be latent evidence for the Dynamic Sytems/ Network Attractor theory of neuronal computation, 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.

Multivariate approaches to spike train network analysis often involve inferrring a causal network diagram from spiking acitivity. 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. The computation of some kind of statistic between each possible pair of neurons in the network. 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 multivariate statistics is not computationally tractible at the scale of billions of neurons, and adding time lagged analysis of network cross-correlation, or transfer entropy makes the prospoect of scaled temporal analysis even worse. Auto-covariance acts on anolog signals (dense vectors), however autocovariance analysis of continuous membrane potentials would be another way to arrive at a network state description.

Two common models of cortical spiking networks are the, Potjan's and Diesmon model and the Brunel model, both of these models are said exist within a fluctuation driven regime, when these are simulated, observed spike times are typically chaotic and random, but some fine grained recognizable repeating patterns also occur. Under the dynamic systems view of the brain neuronal memories are analogous to attractor basins [Hopfield,Lin, Hairong, et al] The authors 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.

The application of STDP learning within the fluctuation driven regime necessitates 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 simulators.

It is the authors view, that the fast algorithm described above is functionally similar to the RecurrenceAnalysis approach, and that it leads to a faster and more interprebable network transition matrices.

A. Evidence for Network Attractor Theory

Reoccurance analysis was used to characterize vector encoded spike train matrices for repeating patterns. Re-currence analysis was able to give us numbers quantify the degree of repitition of states, and the entropy of state transition matrices. Steps convert the spike encoded vector matrices to "state space sets" as defined in the Julia Package Dynamical Systems.jl

1) *Discusssion:*

The attractor network view of the mammal cortex is consistant with phenomological 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.

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.

In the absence of infinite recording data, we can draw no conclusions about wether one of observations would have reoccured 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 transistion 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.

II. Theoretical Framework

III. Methodological Framework

1) *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 windows 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 occured is subtracted from each window, making any variation of spike times inside the window relevant to the windows scale). Each of the converted time, time windows is 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 maxmimum firing rate from step 3. And constructing a spike train that has regular Inter Spike Intervals (ISIs) occuring 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 unique enough). There is M number of neurons we can build a vector of coordinate of **M** dimensions, at each of N time windows. An M by **N** matrix consists of M neurons and N time windows.

7. Since each column vector 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 occured in established work, we are commparing spike train distance vectors within the same neurons along time).

9. We perform unsupervised clustering on this temporaly encoded dissimalirity 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.

IV. Discussions

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 satisfys 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 sim-

ilar 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. [5] Julia simulation of learning. We simulated NMNIST learning. [6] 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.

Kreuz Distance from Uniform ISI	Auto Covariance	Local Variation
Kreuz spike distance number 0	Autocovariance number 1	local variation number 0

V. RESULT ANALYSIS
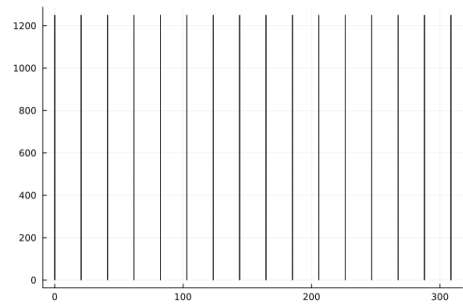
A plot of the regular periodic spike reference window. A unvarying uniform surrogate spike train is used as a comparison inorder to compute the transformation cost of transforming spike train uniform to spike train varying.
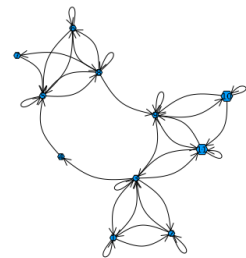
The output of the framework is a sequential state transition network of the spike train. Spontaneous network activity which didn't get repeated was simply not included in the state transition diagram. Two state transition diagrams are output, one with non repeating states, and one with repeating states.
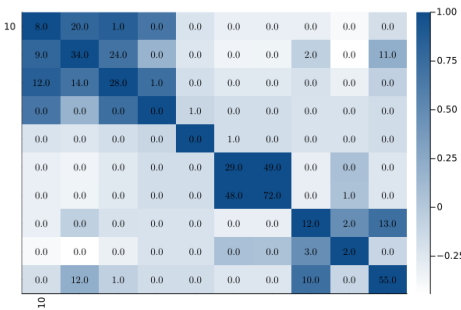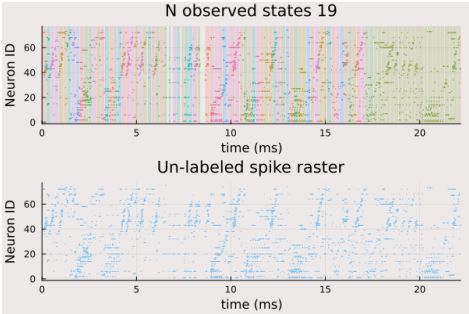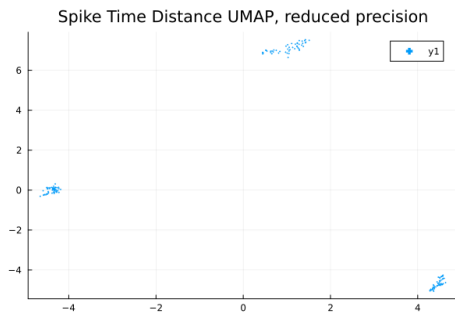

cluster_horizontal_vectors_sort_song_birds.png
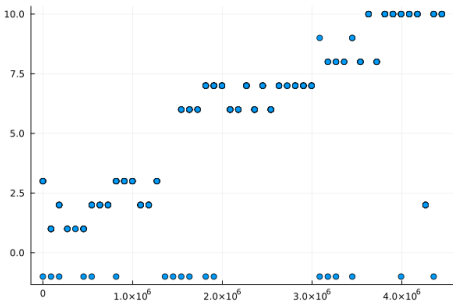
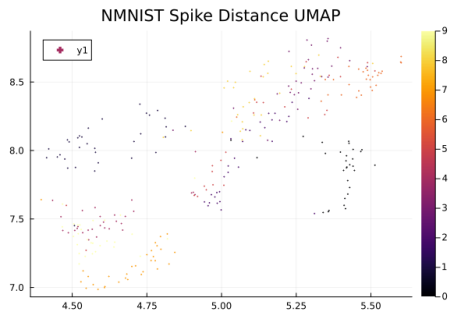**cluster_horizontal_vectors_sort_song_birds.png**
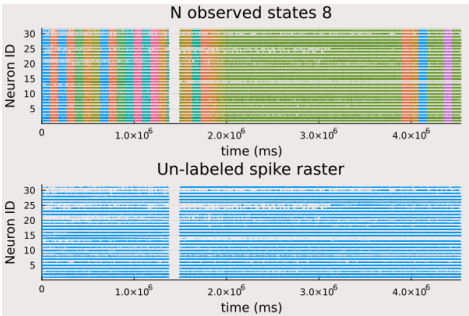
A scatter plot of state transition trajectories in the prefrontal cortex spike train.
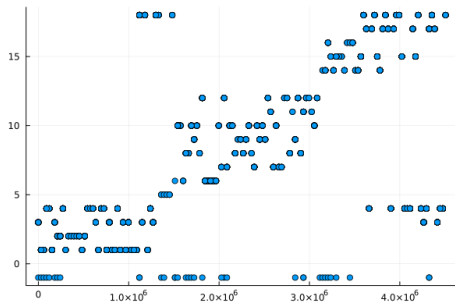

cluster_horizontal_vectors_sort_song_birds.png
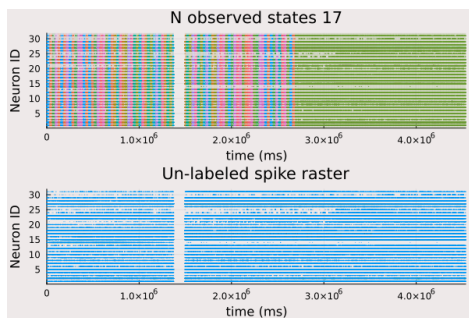
cluster_horizontal_vectors_sort_song_birds.png

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**
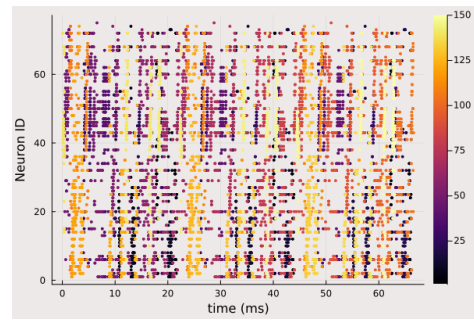
In order to test if the spike2vec framework worked as expected, we downloaded a alcium imaging recording from Zebra finch (a song bird's) High Vocal Centre (brain region) source [7]. Although the actual data source was from (https://github.com/lindermanlab/PPSeq.jl/blob/master/demo/data/songbird_spikes.txt) 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.
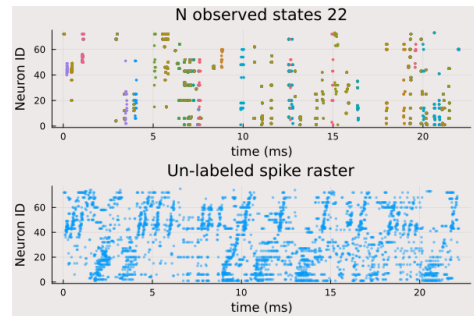

We take the single un augmented Zebra finche data and label repeating patterns using the discussed frame work, since there are no obvious explicit repitions caused by duplicating spike patterns, the algorithm is forced to consider the similarity between more disparate population level patterns.
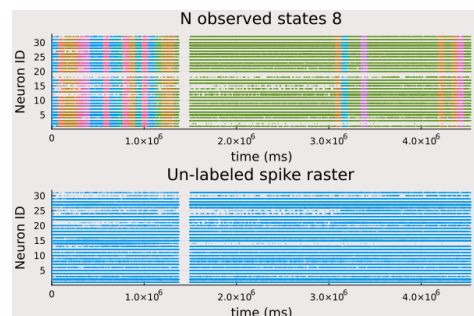
We took a data set from rat prefrontal cortex. The data set was captured, under experimental conditions explicitly designed to maximise the probability of recording replay ie dreaming and slow wave sleep.
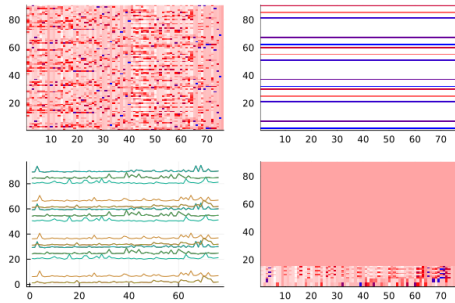
Figures from left top to bottom right:A: Top left: 75 NMNIST channels were recorded and time binned in a manner which yielded 85 vectorized time bins. Bottom right: Once the vectorized time bins had been vectorized, a clustering algorithm was applied to the entire matrix of vector coordinates. Cluster centres could then be used as reference points, such that it was possible to compare all
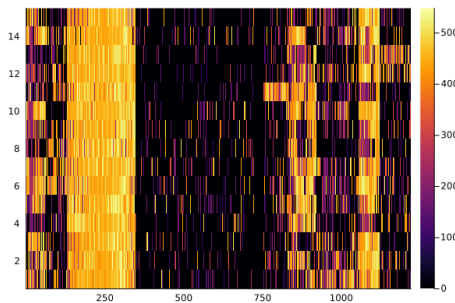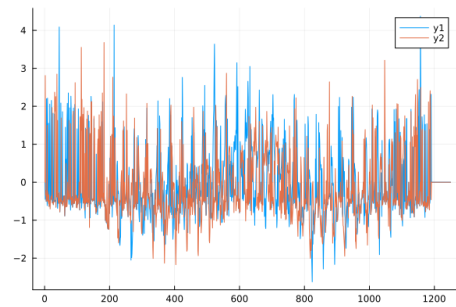

Normalized prefrontal cortex vectors for over 1000 time steps 15 neurons
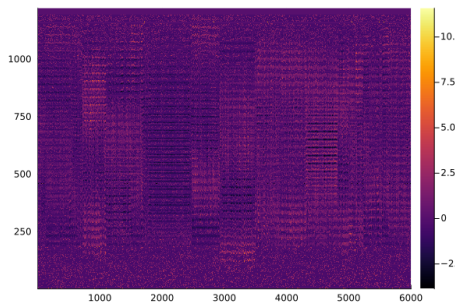
We sampled the NMNIST data set more broadly using 6,000 samples to create vectors. The entire data set consists of 60,000 samples.

align(center + bottom)[

Two spike time encoded numerals, where read in to Julia, then the spiking neuromorphic data were converted to vectors over 1200 channels. Orange and Blue plots are vectors corresponding to two distinct NMNIST data labels. Rapid positive alternating deflections are visible in both vectors, because the NMNIST 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 dis-simarity matrices constructed using the NMNIST dataset, ie the heatmap above, comes from analysing spike train distance across the NMNIST 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 data set isn't shuffled), and this contiguity is reflected in the heatmap too.]

A. References

REFERENCES

[1] N. Brunel, "Hebbian learning of context in recurrent neural networks," *Neural Computation*, vol. 8, no. 8, pp. 1677–1710, 1996.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.