Analyses of Neural Ensembles

Last updated: 31 October 2014

Mark Humphries

Contents

1 Introduction

So you've found some ensembles of neurons (or "cell assemblies") in your population recording data - now what? This toolbox tackles this problem by laying out a set of tools for analysing neural ensembles.

This report documents some of the ideas and algorithms designed for *classifying* neural ensembles that are currently realised in the toolbox.

2 Fit-space: finding types of neurons and ensembles

Having recorded from many individual neurons and detected the neural ensembles in each recording, the key challenge is building a picture of the neural system using this data. Often each recording represents a partial, noisy sample of the system, so we need robust tools to integrate across the samples. The first step is to identify common characteristics of firing properties across the recordings, to build a database of types of ensembles in the neural system.

Our goals here include:

1. Ensemble types, existence: ensembles are defined by the temporal correlations of their neurons' spike-trains; but each ensemble will then also have a characteristic set of spike-train statistics; how many types of ensemble exist when we take these statistics into account? (e.g. tonic, bursting, oscillatory, other?)

2. Ensemble types, use of: by identifying ensemble types, we can then refine all further analyses of ensemble-level dynamics by selecting the type of ensembles we are interested in (e.g. for sequential firing, we can select out only oscillatory types).

3. Cataloguing neurons and ensembles: large-scale recordings allow us to rapidly sample across a large subset of simple neural systems, reducing the time to gather data by orders of magnitude compared to traditional single cell extra- or intra-cellular recordings. For example, our first application of these tools was to population recording data from *Aplysia* pedal ganglion during fictive locomotion (Bruno et al, in revision). To that point, the only well-characterised large sample of neuron types were the motorneurons of the rostro-medial quadrant (Hening et al., 1979) and neurons contributing axons to pedal nerve 9 (Fredman and Jahan-Parwar, 1980). So we had the opportunity to map the largely unexplored majority of the ganglion.

4. Novelty of neurons and ensembles: we would like to know if there exists any particularly unusual types of ensembles and neurons, which would merit further investigation on their own.

To achieve these goals, we want to characterise the ensembles using a range of measurements, and use unsupervised clustering on these measurements to identify groups of similar ensembles. That is, we might imagine there exists a small set of "prototype" ensembles of which the actual recorded ensembles are noisy realisations.

We should note that this problem of unsupervised electrophysiological classification is of very general interest in neuroscience – a well-known example being the efforts of cortical electrophysiologists to automate categorisation of the broad range of interneurons and pyramidal neurons (Cauli et al., 2000; Petilla Interneuron Nomenclature Group, 2008; Defelipe et al., 2013). A particular challenge facing the use of large-scale recording technology (Bartho et al., 2004; Fujisawa et al., 2008) is that we only have access to spike-trains, and not other electrophysiological parameters that usefully distinguish neuron types.

Thus the challenge here is to use only measures of spike-train structure to define these types, and find a combination of these measures' values to uniquely define each type. The key problem to solve is that each neuron's spike-train potentially has a complex structure, so we need a way to succinctly capture this structure without obscuring differences between neurons, yet in a way that allows for unsupervised clustering.

2.1 Spike-train measures for defining types

We defined types of neural ensembles based on the structural properties of their spike trains in terms of the firing rate and firing regularity. The former gives us clues about the excitability of the neuron, the latter gives us clues about the intrinsic dynamics of the neuron; together they form a picture of how a neuron integrates its inputs into its output. We refer to this throughout as the rate-regularity type of an ensemble.

For firing rate, we simply use the inter-spike interval (ISI), and calculate all ISIs for each spike-train. Typically spike-train rates are summarised by the mean ISI or its reciprocal f = 1/mean(ISI) spikes/s. However, the mean ISI is not broadly distributed for the neurons in the pedal ganglion, and so is a poor indicator of differences in spike-train structure.

Many measures of spike-train regularity are available. The most common choice is to use the coefficient of variation (CV) of the ISI: CV(ISI) = std(ISI)/mean(ISI). This has the nice property that CV(ISI) = 1 for a Poisson point process, and so CV(ISI) $<$ 1 indicates greater regularity and CV(ISI) $>$ 1 indicates greater irregularity. However, as it is measured from ISIs taken over the whole recording, it is sensitive to changes in firing rate over the recording because the standard deviation of the ISIs will reflect those changes.

To overcome this problem, we used the so-called CV_2 (Holt et al., 1996), which is both less sensitive to rate changes and allows for examination of the changes of irregularity over

time (Ponce-Alvarez et al., 2010; Wohrer et al., 2013). This is a local measure of regularity between adjacent ISIs I_i and I_{i+1}:

$$CV_2(i) = \frac{2|I_i - I_{i+1}|}{I_i + I_{i+1}}, \tag{1}$$

which is the ratio of the twice the difference to the sum of those adjacent ISIs. Typically the spike-train is then summarised by taking the mean $\bar{CV}_2$ over all $CV_2(i)$ with $i = 1 \ldots n-1$ for n ISIs. Again this has the nice property that $\bar{CV}_2 = 1$ for a Poisson point process, and so $\bar{CV}_2 < 1$ indicates greater regularity and $\bar{CV}_2 > 1$ indicates greater irregularity. Moreover, we also have the option of examining the distribution of the $CV_2(i)$s themselves, which we use below.

2.2 Defining types by distributions of metrics

We compute the above measures for every spike-train in the data-set. Spike-train structure is difficult to summarise using simple aggregates of these measures. Our applications to real data-sets indeed showed clear problems with using simple aggregate measures to characterise neurons: visual inspection showed that many neurons had bimodal distributions of ISIs and of $CV_2(i)$s, and so mean rate (mean ISI) or mean regularity ($\bar{CV}_2$) did not capture even the central peak of the data, let alone the shape of the distribution. Consequently, we wished to use the complete distribution of ISIs and of $CV_2(i)$s to capture the complexity of each neuron's spike-train structure.

To identify ensemble types we needed to also characterise their rate and regularity properties. One option would be to use an aggregate over all neurons in the ensemble – such as the mean firing rate – but given the problems with using such aggregate measures for individual neurons, pooling them across multiple neurons would cause further issues in capturing the structure of the ensembles' spike-trains. We thus characterised each ensemble by a single ISI and a single $CV_2(i)$ distribution created by pooling the distributions for each neuron in the ensemble.

2.3 Summarising distributions of spike-train metrics

To directly use these distributions of rate and regularity to characterise each ensemble's spike-train structure we must first quantitatively summarise them. We thus fit models for a range of possible distributions (see below). However, this seems incompatible with our ultimate goal of finding ensemble types by unsupervised clustering: first, for unsupervised clustering each object must be defined by the same length N vector of properties; for each ensemble that would be something like: [mean(rate), std(rate), mean(regularity), std(regularity),...]. We then cluster in this N-dimensional space. But fitting a complete distribution model would provide either different numbers of parameters or incomparable parameters for each distribution, depending on the fitted model (for example, one free parameter for an exponential distribution versus two for a normal distribution; the mean and standard deviation for the normal distribution versus the a,b parameters for the Gamma distribution). Second, even if we arrange for each distribution to be characterised by the same set of parameters, that leaves the problem of how to choose the right distribution model(s).

2.4 Fit space

We introduce here the idea of "fit space" to solve both problems simultaneously: we fit a range of m models to a distribution, and find for each model M the probability $p(M)$

that it is the best fitting of the m candidates. For that distribution we now have a length m vector $P(model)$ defining the relative fits of all the models – for example for $m = 5$ we might have $P(model) = [0, 0.1, 0, 0.7, 0.2]$. By repeating for all distributions, we then have every ensemble represented as a point in this m-dimensional space of "fits". We can then cluster to determine if there are distinct types of ensemble present in that space.

This approach has two particularly appealing properties. First, we can go on to include other properties we wish to use to define "types" (for example, the mean phase of spiking with respect to some global oscillation) by simply concatenating their measured values to the length m vector and increasing the dimensions of the space. We can even concatenate two or more fit-spaces, as we do in the main text for the rate and regularity vectors for $P(model)$. Second, amongst the m candidates we do not need to guess the correct model for every single distribution in the data. The m-dimensional vector of probabilities can also be interpreted as mixture weights for combining the candidate distributions to obtain the true distribution. If a particular type of ensemble has a distribution of, say, ISIs that is not in the m candidates, then it is likely that all ensembles of that type will have approximately the same vector of probabilities giving the mixture of models closest to the true distribution. Thus all ensembles of that "type" will still have points in approximately the same region of fit space, and will still be found by the unsupervised clustering.

2.4.1 Choosing and fitting candidate models

Our choice of models was guided by the twin needs to broadly sample the range of possible distributions (to allow a mixture interpretation to make sense) and to capture the evident dominant distributions of ISIs and $CV_2(i)$s in the data from visual inspection of a sample of neurons. Our final set of $m = 6$ candidates were: (1) Normal ($n = 2$ parameters); (2) Log-normal ($n = 2$); (3) Gamma ($n = 2$); (4) Uniform ($n = 2$); (5) Bimodal Normal ($n = 5$); (6) Bimodal Gamma ($n = 5$). Note that both bimodal models include a mixture parameter.

Two are particularly unusual. The uniform distribution was included because this is the expected distribution of CV_2(i)s for a Poisson process (as each possible pair of adjacent ISIs occurs with equal probability, so their ratios as expressed in $CV_2(i)$ must also occur with equal probability). The bimodal distributions of ISIs and $CV_2(i)$s visually observed in the data often were not symmetric around the two modes, and thus a bivariate Gamma model was included in an attempt to capture this asymmetry.

All models were fitted to each distribution using maximum likelihood estimation (MLE). Models (1)-(4) have analytical MLE solutions for their parameters, and are also available in MATLAB's Statistics Toolbox. Models (5) and (6) do not have analytical MLE solutions for their parameters, so we used numerical optimisation of their maximum-likelihood fits to the distribution using MATLAB's mle and fmincon functions.

2.4.2 Calculating $P(model)$

For each distribution the fitting process gives us a vector of negative log-likelihoods L, one per model. One option is to interpret the model with the lowest L as the best-fit to the distribution. However, this does not take into account that the models have different numbers of parameters, and we must always take care that a model with more parameters is not simply over-fitting the data. Thus we choose some measure of model-selection that computes the trade-off between the goodness-of-fit and the number of parameters. A wide-range of model-selection approaches are available (see e.g. (Wasserman, 2004)). Here we will use the Bayesian Information Criterion (BIC) as it tends to more heavily

penalise the number of parameters, and thus makes the selection of the bimodal models (5-6) more convincing. But equally we could have used Akaike's Information Criterion (AIC) or others.

Given that L is the negative log-likelihood of model M_i, the BIC is

$$\mathrm{BIC}(M_i) = 2L + k\ln(n) \tag{2}$$

where k is the number of free parameters and n is the number of data-points in the distribution. The model with the lowest BIC score is thus considered the best-fit in that it best trades-off the goodness-of-fit (as measured by L) with the number of free parameters (as given by the second term above). However, the absolute BIC values cannot tell us whether that model is notably better than the next-best model: as these values scale with the number of data-points n, so their differences might be small on the scale of the BIC values. Instead, using the BIC values we can compute the approximate posterior probability that each model is correct (Wasserman, 2004):

First for each model we compute the difference between its BIC value and the minimum of the BIC values: $\Delta_i = \mathrm{BIC}(M_i) - \min_{j \in m} \mathrm{BIC}(M_j)$.

Second, we compute the posterior probability that this model is correct by:

$$p(M_i) = \frac{\exp(-\Delta_i/2)}{\sum_{j=1}^{m} \exp(-\Delta_j/2)}. \tag{3}$$

Thus we now have our fit-space vector: $P(model) = [p(M_1), p(M_2), ..., p(M_m)]$ computed for each of the m models applied to the distributions. We do this for the distribution of ISIs and the distributions of $CV_2(i)$s for each ensemble.

2.4.3 Finding "types" in the fit-space

Our overarching goal was to find properties of ensemble activity that uniquely define "types". This is another unsupervised clustering problem in an N-dimensional space: we take N metrics for spike-train structure, and n objects (here, ensembles) with scores for each metric, and we ask if this n-dimensional data-set can be reduced to C clusters, where each cluster is a defined "type". In the main text we use a 12-dimensional vector, made by concatenating the 6-dimensional $P(model)$ vector for rate (ISI distribution) with the 6-dimensional $P(model)$ vector for regularity ($CV_2(i)$ distribution).

We use our consensus community detection algorithm to also tackle this unsupervised clustering problem (the toolbox includes a version of this algorithm; the latest version is at https://github.com/mdhumphries/SpikeTrainCommunitiesToolBox).

We construct a weighted network $\mathbf{D}$ for the entire dataset using the Euclidean distance between the fit-space vectors for each pair of objects. Note that we use Euclidean distance and not a correlation-type measure due to the likely small number of non-zero entries in the vectors. We then convert this to a similarity matrix $\mathbf{W}$ by $W_{ij} = \exp(-D_{ij}^2)$. This similarity matrix is then passed to the consensus community detection algorithm.

2.5 Code

All MATLAB functions for creating and clustering in the "fit-space" are in the toolbox. Each function has extensive help text within it; here we give an overview of each function's role.

Main functions:

fitMLEdistribution Takes a distribution of some variable, and fits the specified models to it using maximum likelihood; also computes the AIC and BIC scores for the fits, and corresponding estimates of $P(model)$; the latter specify position of that distribution in the fit-space. This function calls:

bimodalGaussianPDF computes the PDF for a bimodal Gaussian model

bimodalGaussianCDF computes the cumulative distribution function (CDF) for a bimodal Gaussian model

bimodalGammaPDF computes the PDF for a bimodal Gamma model

bimodalGammaCDF computes the CDF for a bimodal Gamma model

BICL Compute BIC scores from a set of (negative) log-likelihoods

AICL Compute AIC scores from a set of (negative) log-likelihoods

Akwgts Compute the approximate posterior probability $P(model)$

These functions are used by top-level scripts in the toolbox:

Analyse_Spike_Train_Properties computes statistical properties of each neuron's and ensemble's spike-trains, and fits models to their distributions using fitMLEdistribution

Types_of_Ensemble_Across_Dataset Clusters in the fit-space. Computes the similarity between each pair of ensembles, and uses the consensus community detection algorithm to cluster ensembles in that space.

References

Bartho, P., Hirase, H., Monconduit, L., Zugaro, M., Harris, K. D., Buzsaki, G., 2004. Characterization of neocortical principal cells and interneurons by network interactions and extracellular features. J Neurophysiol 92, 600–608.

Cauli, B., Porter, J. T., Tsuzuki, K., Lambolez, B., Rossier, J., Quenet, B., Audinat, E., 2000. Classification of fusiform neocortical interneurons based on unsupervised clustering. Proc Natl Acad Sci U S A 97, 6144–6149.

Defelipe, J., Lpez-Cruz, P. L., Benavides-Piccione, R., Bielza, C., Larraaga, P., Anderson, S., Burkhalter, A., Cauli, B., Fairn, A., Feldmeyer, D., Fishell, G., Fitzpatrick, D., Freund, T. F., Gonzlez-Burgos, G., Hestrin, S., Hill, S., Hof, P. R., Huang, J., Jones, E. G., Kawaguchi, Y., Kisvrday, Z., Kubota, Y., Lewis, D. A., Marn, O., Markram, H., McBain, C. J., Meyer, H. S., Monyer, H., Nelson, S. B., Rockland, K., Rossier, J., Rubenstein, J. L. R., Rudy, B., Scanziani, M., Shepherd, G. M., Sherwood, C. C., Staiger, J. F., Tams, G., Thomson, A., Wang, Y., Yuste, R., Ascoli, G. A., 2013. New insights into the classification and nomenclature of cortical GABAergic interneurons. Nat Rev Neurosci 14, 202–216.

Fredman, S. M., Jahan-Parwar, B., 1980. Role of pedal ganglia motor neurons in pedal wave generation in Aplysia. Brain Res Bull 5, 179–193.

Fujisawa, S., Amarasingham, A., Harrison, M. T., Buzski, G., 2008. Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. Nat Neurosci 11, 823–833.

Hening, W. A., Walters, E. T., Carew, T. J., Kandel, E. R., 1979. Motorneuronal control of locomotion in Aplysia. Brain Res 179, 231–253.

Holt, G. R., Softky, W. R., Koch, C., Douglas, R. J., 1996. Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. J Neurophysiol 75, 1806–1814.

Petilla Interneuron Nomenclature Group, 2008. Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex. Nat Rev Neurosci 9, 557–568.

Ponce-Alvarez, A., Kilavik, B. E., Riehle, A., 2010. Comparison of local measures of spike time irregularity and relating variability to firing rate in motor cortical neurons. J Comput Neurosci 29, 351–365.

Wasserman, L., 2004. All of Statistics: A Concise Course in Statistical Inference. Springer, New York.

Wohrer, A., Humphries, M. D., Machens, C., 2013. Population-wide distributions of neural activity during perceptual decision-making. Prog Neurobiol 103, 156–193.