Spike Sorting and Quality Evaluation Through Generative Model Estimation, Data Simulation, Flexible Coarse Classification and Refined Template Matching

Christophe Pouzat

2011-05-19

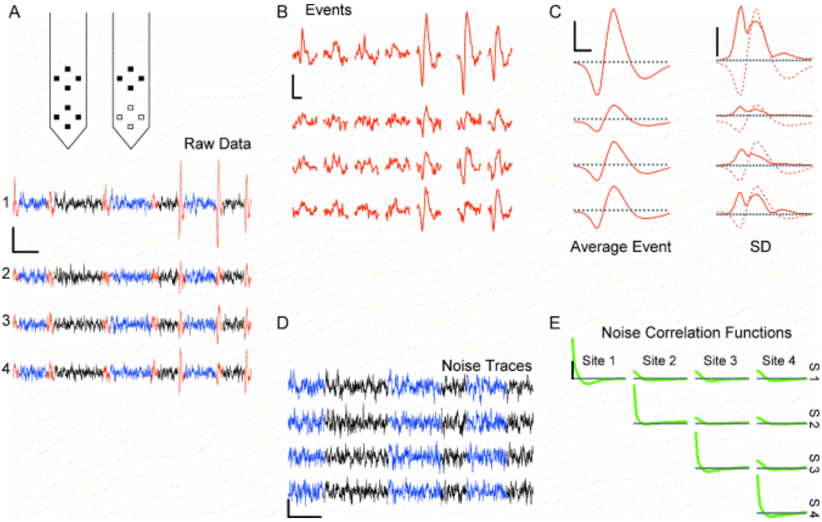
Outline

Our former fully parametric approach (1)

In a joint work with Ofer Mazor and Gilles Laurent (2002, *J Neurosci Methods*, **122**, 43-57) we proposed (following others on most of the points) to:

- Split the *continuous* raw data in 2 parts: the events and the noise.
- Build a pseudo continuous noise trace sticking together the noise parts in between events and estimate the second order properties, the *covariance matrix* of the noise.
- Check that a noise description based on its covariance matrix was good enough.

Our former fully parametric approach (2)



Our former fully parametric approach (3)

We did next:

- *Whiten* the events base on the noise covariance matrix.
- Split the events' sample into "pure" – non superposed – events and superposed events.
- Run an EM algorithm with a Gaussian mixture model – using a diagonal covariance matrix for each cluster since whitening was performed – setting the number of clusters by minimizing the BIC.
- Notice that our explicit "generative model" was: each pure event from a given neuron is generated by adding to a single mother waveform an uncorrelated white Gaussian noise.
- The idea was that if both our estimated waveforms and our noise model were correct then the sum of squares of the residuals (`RSS`) of each pure event after classification should follow a χ^2 distribution.

We then ran into problems with large spikes...

Our former fully parametric approach (4)

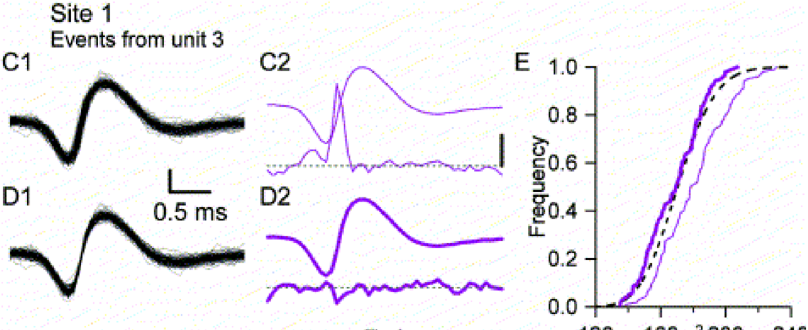
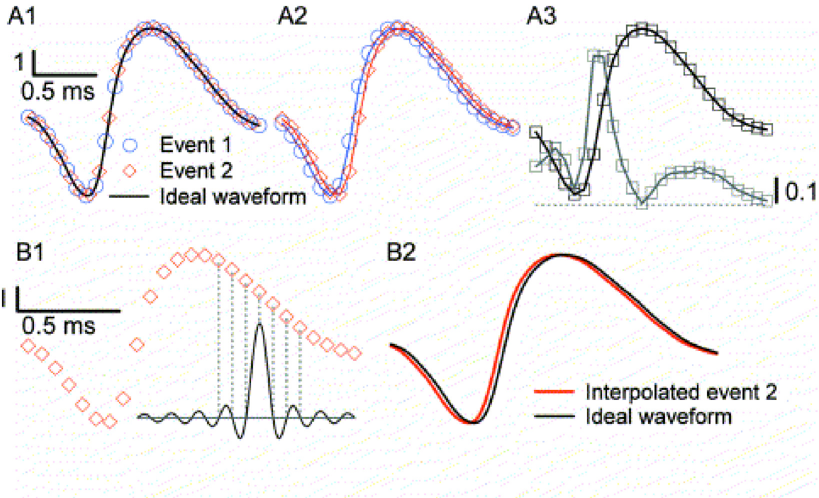


Fig. 4

The jitter problem…

Our former fully parametric approach (5)



The solution was to align events of a cluster on the "central" event with a `sinc` ($\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$) interpolation.

Our former fully parametric approach (6)

- `sinc` based alignment was done with a discrete oversampling by a factor of 10 (*i.e.* we added 9 points in between each sample).
- Full sample classification was done in two steps:
  - Try each pure waveform with each jitter value and keep the one giving the smallest `RSS`.
  - If the `RSS` is compatible with the noise, classify.
  - If not, try all combinations of 2 pure waveforms...
  - If this last step does not give a small enough `RSS`, tag the event as unclassified.

Some drawbacks of our former approach

The algorithm just outlined suffered from many drawbacks, here are some of them assuming that the key hypothesis of waveform stationarity is valid:

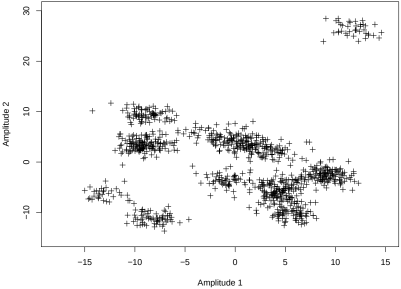
- To get a good noise description based on the covariance matrix we had to take low detection threshold leading to many badly isolated and therefore not usable units at the end.
- The `sinc` interpolation based alignment was very costly.
- Doing the alignment for each event and each waveform was therefore even costlier.

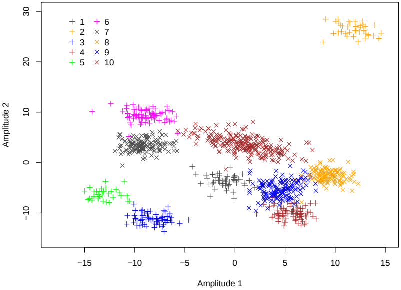
We needed the mutlivariate Gaussian noise to be able to build our goodness of fit (GOF) tests and that forced us to align the events. That means that if you don't care about GOF tests you don't need to bother with alignment.

Some afterthoughts

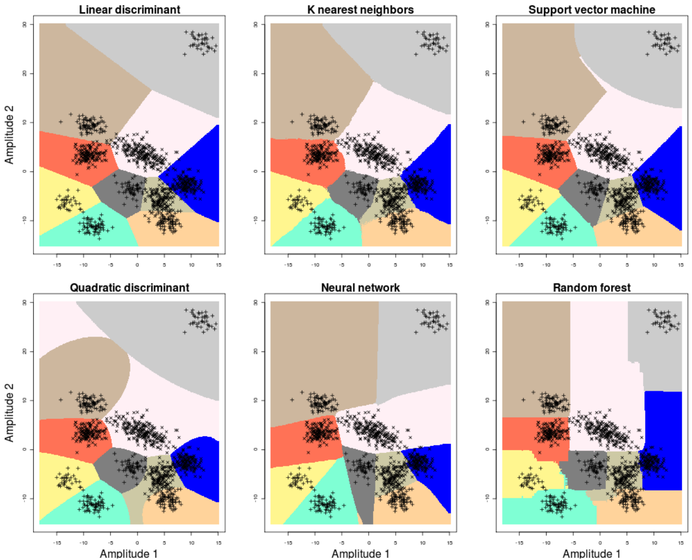
- We insisted a lot on making the analysis as automatic as possible in order to get reproduciblity.
- There are now a bunch of wonderful tools allowing us to simply and comprehensively detail every single step of an analysis:
  - The `Sweave` function of `R`.
  - The combination of `emacs` and `org-mode` and `R` or `Matlab` or `Octave` or `Python` or etc.
- So we get reproducibility by explaining in plain English the "arbitrary" choices we made.
- The automatic setting of the number of clusters by the BIC in combination with Gaussian mixture models does not compete with the human eye plus brain helped by GGobi (a dynamic multidimensional visualization tool).

Split explicitly clustering. . .





. . . and classification



More explicitly we propose

We take the approach of Chandra & Optican (Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network. *IEEE Trans Biomed Eng*, 1997, **44**, 403-412) but:

- We take an empirical noise sample, not its representation based on second order statistics.
- We don't limit ourselves to neural network based classifiers.

The key ideas here are:

- We cluster pure events and use the results of this clustering stage to generate a "full" sample with pure and superposed events (with jitter)... Clearly if we get the clustering wrong everything coming after will also be wrong.
- Classification methods / algorithm / theory are much more advanced than their clustering counterparts, so we use better tools when we can.

Quality checks

- The classification is based on simulated data with jitter. That's why we speak of "coarse" classification.
- After coarse classification, alignment is performed but with a faster method.
- `RSS` comparison with the noise sample is done with the Dvoretsky-Kiefer-Wolfowitz (DKW) inequality.

The DKW inequality states that if $X_1, \ldots, X_n \sim F$, then for any $\epsilon > 0$,

$$\text{Prob}\left(\sup_x |F(x) - \hat{F}(x)| > \epsilon\right) \leq 2\,\exp(-2n\epsilon^2)$$

where $\hat{F}$ is the empirical distribution function – these confidence bands are just simpler to compute than the Kolmogorv-Smirnov ones and are practically as tight.

Alignment without `sinc` interpolation (1)

- Let us call $g(t)$ the observed amplitude at time t within our cut.
- We have $g(t) = f(t+\delta)$, where δ is our *jitter induced by sampling and noise* and f is our "mother" waveform (ignoring additive noise).
- We are going to model δ as the realization of a random variable Δ with theoretical mean, $\mathrm{E}\Delta = 0$, and a finite variance, $\mathrm{E}\left(\Delta - \mathrm{E}\Delta\right)^2 = \mathrm{E}\Delta^2 \equiv \sigma^2_{\Delta}$.
- Assuming that f admits at least two derivatives (it does since our data are low-pass filtered), we have:

$$g(t) = f(t) + \delta\, f'(t) + \frac{\delta^2}{2} f''(t) + o(\delta^3)\,.$$

Alignment without `sinc` interpolation (2)

- From our hypothesis on Δ we have:

$$\mathrm{E}g(t) = f(t) + \frac{\sigma_\Delta^2}{2} f''(t) + o(\mathrm{E}\Delta^3) \, ,$$

- That is, to the first order in δ: $\mathrm{E}g(t) = f(t)$.
- That means that our template estimates are nearly unbiased if σ_Δ^2 is sufficiently small.
- Now what about the variance of $g(t)$? We have *to the first order in* δ :

$$\mathrm{E}\left(g(t) - \mathrm{E}g(t)\right)^2 = \mathrm{E}\left(\delta \, f'(t)\right)^2 = \frac{\sigma_\Delta^2}{2} f'^2(t) \, ,$$

- Leading to a standard deviation (to the first order in δ):

$$\sigma_{g(t)} = \frac{\sigma_\Delta}{\sqrt{2}} \left| f'(t) \right| \, .$$

Alignment without `sinc` interpolation (3)

With actual data we have:

- $g(t_i) = f(t_i + \delta) + \epsilon_i$, with ϵ_i random variables with zero mean and finite variance – not necessarily independent.
- We use a second order Taylor expansion to get: $g(t_i) \approx f(t_i) + \delta f'(t_i) + \delta^2/2 f''(t_i) + \epsilon_i$.
- Our estimated δ, $\hat{\delta}$, is defined by:

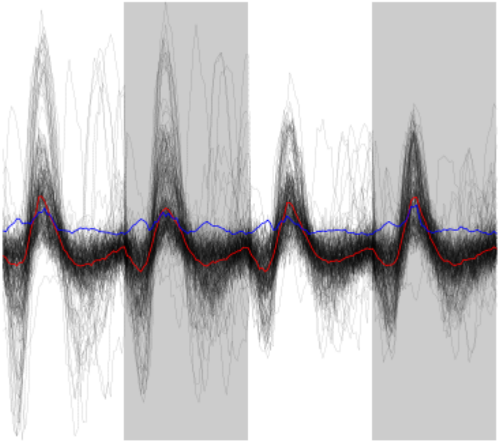
$$\hat{\delta} \equiv \mathrm{argmin}_\delta \sum_i \left(g(t_i) - f(t_i) - \delta\, f'(t_i) - \frac{\delta^2}{2}\, f''(t_i) \right)^2 .$$

- This is fast to compute once estimates of f, f' and f'' have been obtained.
- It is safe to use the pointwise median as opposed to the *mean* to get them.
- Using the first order Taylor expansion works most of the time but not always and the extra cost generated by the second order is “affordable”.

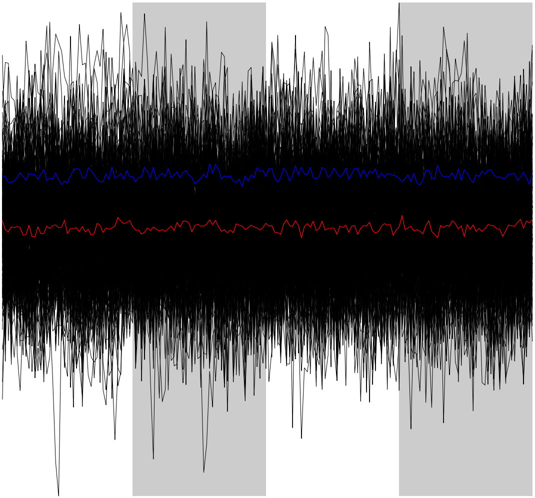
Self consistent quality checks (1)

- The basic idea is to compare empirical distribution of the `RSS` (Residual Sum of Squares) of the classified events with the empirical distribution of noise events' sum of square.
- So we start by splitting data into events and noise...

Self consistent quality checks (2): the spikes

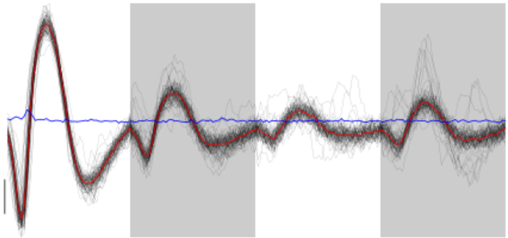


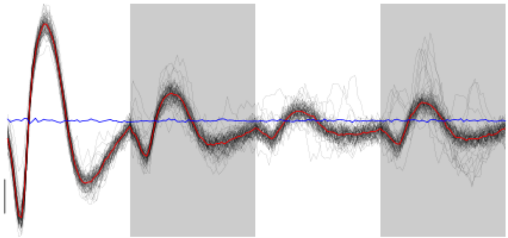
Self consistent quality checks (3): the noise



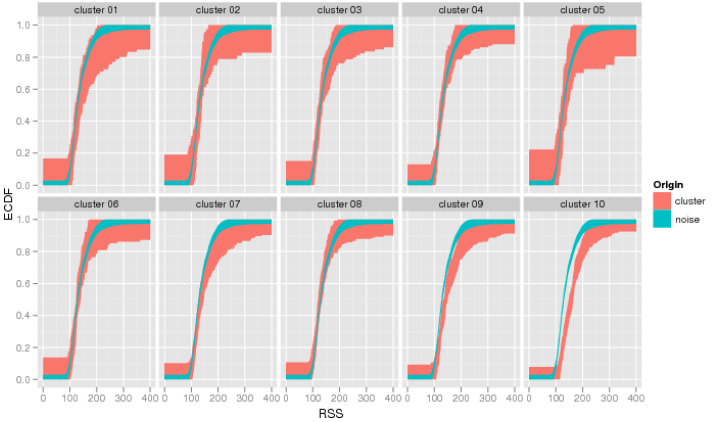
Self consistent quality checks (4)

After (coarse) classification we align the events (recursively):





Self consistent quality checks (5): RSS distributions



The end

Thank you all for listening!