Abstract: Spike detection and spike sorting techniques are often difficult to assess because of the lack of ground truth data (i.e. spike timings for each neuron). This is particularly important for in vitro recordings where the signal to noise ratio is poor (as is the case for multi-electrode arrays at the bottom of a cell culture dish). We present an analysis of the transmission of intracellular spike events from neurons to an extracellular electrode, and a set of MATLAB functions based on this analysis. These produce realistic signals from neighboring neurons as well as interference from more distant neurons, and Gaussian noise. They thus generate realistic but controllable synthetic signals (for which the ground truth is known) for assessing spike detection and spike sorting techniques. They can also be used to generate realistic (non-Gaussian) background noise. We use signals generated in this way to compare two automated spike sorting techniques. The software is available freely on the web at http://www.cs.stir.ac.uk/~lss/noisyspikes.

The software
The software (available from http://www.cs.stir.ac.uk/~lss/noisyspikes) is a set of MATLAB m-files. It needs the statistics and signal processing toolboxes. As well as the m-files, the website also contains a reasonably comprehensive user manual, and an extended paper[1].

The software allows:
• user selectable (and user definable) intracellular spike shapes[2]
• a user-selectable number of target neurons (including 0, allowing pure interference generation)
• a user selectable number of interference neurons, whose spike times are correlated with one of the target neurons
• a user-selectable number of uncorrelated interference neurons

Neural spikes may have Gaussian or Poisson distributions. Spike times may be re-used so the same experiment may be run with different amounts of interference. Virtually all the parameters in the simulation can be set (and are detailed in the user manual).

Figure of Merit
In an attempt to summarise the effectiveness of spike sorting we define a figure of merit, the number misclassified, the number not classified. P is preprocessing type, FiO is figure of merit, MC is misclassified, UC is unclassified spikes (either in the correct group, or in some other cluster apart from the one selected in the figure of merit). M is spikes missed, and I is spikes inserted.

Using KlustaKwik on wavelet data has some wrinkles: if the maximum number of clusters permitted is not set, far too many are produced. Setting a limit of (5) sometimes results in only one cluster being produced (line 2). Best results are obtained using PCA: SPC and KK applied to PCA perform equally.

Figure 1 shows equivalent circuit description for transfer of charge from a point on a neuron to an electrode for an extracellular and a dish based electrode. The extracellular electrode is assumed to be near the neuron, but the dish electrode may be a layer of glia between it and the neural culture. There are a number of simplifications in this circuit: the distinct resistances and capacitances have been lumped together, for example.

What is the effect of the integrating this over the extent of the neuron?
1. Because the time taken for spike movement from the spike initiation point on the soma through the axon is comparable to or larger than the spike duration, this integration will have a major effect on the shape of the voltage recorded at the electrode.
2. Because the different parts of the spiking neural surface will be at differing distances, the shape of the voltage at the electrode corresponding to a spike will depend on the precise geometry of the electrode and neuron.

We have lumped these together. We use

\[ V_{\text{peak}}(t) = \sum_i \left( V_{\text{intra}}(t) - \delta(t - \delta_{\text{init}}) \right) b_i + \sum_j V_{\text{spike}}(t) \]

where \( V_{\text{intra}}(t) \) is the voltage recorded at the electrode from neuron \( j \) at time \( t \), \( \delta(t) \) is the intracellular voltage inside Neuron \( j \) (assumed constant over its extent), and \( b_i \) the characteristic behavior of each cluster.

In this case, preprocessing using Weavelots followed by clustering using SPC is best.

Correlated and Uncorrelated Noise results:

Table rows show the figure of merit, the number misclassified, and the number not classified. P is preprocessing type, FiO is figure of merit, MC is misclassified, UC is unclassified spikes (either in the catch group, or in some other cluster apart from the one selected in the figure of merit). M is spikes missed, and I is spikes inserted. Using KlustaKwik on wavelet data has some wrinkles: if the maximum number of clusters permitted is not set, far too many are produced. Setting a limit of (5) sometimes results in only one cluster being produced (line 2). Best results are obtained using PCA: SPC and KK applied to PCA perform equally.

Testing similar spike shapes:
The spike shapes used in dataset 1 are shown in figure 3. For the other datasets, the T1 spike shape was made more and more like the T2 spike shape in nine steps using simple linear interpolation. In dataset 10 the T1 and T2 spike shapes are identical.

Even for dataset 1, PCA preprocessed data failed. The table shows only wavelet preprocessing. KK is better at separating the two clusters particularly when they are very similar.

Conclusions:
The synthetic data enables informed experimentation with spike sorting techniques. For very similar spike recording shapes, it is clear that wavelet preprocessing followed by KlustaKwik is the best technique. For other data, no clear winner emerges from these experiments. Perhaps this is not unexpected: the best method of separating the different clusters depends on the actual shapes of the clusters. The synthetic data emphasizes the importance of trying out different techniques, and different parameters to these techniques.

References:

Fifteenth Annual Computational Neuroscience Meeting CNS*2006, July 16-30, Edinburgh

http://www.cs.smu.edu/~dst/Hhsim