**Detecting Spikes**

Several methods for spikes detection are implemented in the class, the one that is actually used is the one implemented in detectTimes. Its basic idea is to find points in the data which satisfy one of several conditions regarding the amplitude after bandpass, the unfiltered amplitude and the gradient at that point. Other methods which were implemented and not used are spike detection using wavelet analysis (based on West et al 2003) and using Taeger energy (based on Zaveri et al 2014), these methods were not validated on data, so should be used with caution.

detectTimes – The method detects spikes by finding the points in which at least one of five conditions is met: 1. The envelope of the signal after bandpass is above a set threshold, 2. The amplitude of the signal is above a set threshold, 3. The gradient (between consecutive time points) of the signal is above a set threshold, 4. The amplitude and the gradient are both above set thresholds which are smaller than the thresholds used in conditions 2 and 3. 5. The amplitude and the envelope of the signal after bandpass are both above thresholds smaller than the ones used in conditions 1 and 2.

These conditions are a mix of methods used in the papers Staresina et al Nat Neuroscience 2015, Andrillon et al J of Neuroscience 2011 and Nir et al Nat medicine 2017 (the latter two use the same method). Condition 1 is used in Andrillon et al, conditions 2-5 are used in Staresina et al (where 5 is a variant, as they use the amplitude after a high pass rather than the amplitude of the envelope after a band pass). The default parameters for the thresholds for the standard deviation are taken from these papers. (Note: as they appear in the code in Shdema’s folder in the dropbox, may have later been changed by Maya).

Not all the conditions have to be applied, the choice of which of them are applied is controlled by changing the value of the Boolean properties: useEnv (condition 1), useAmp (condition 2), useGrad (condition 3), useConjAmpGrad (condition 4), useConjAmpEnv (condition 5).

The method also returns values of the various parameters calculated in the detection process for each detected spike (the output variable peakStats).

The method follows the following steps:

1. NaN points in the data are turned to 0 (otherwise the Hilbert transform will not work).
2. The data is divided to blocks of duration blockSizeSec, the method iterates over the blocks and performs the detection separately for each block.
3. The method checks which points pass the amplitude conditions. The threshold for the amplitude is set in units of standard deviation (relative to the data in the current block), SDthresholdAmp for condition 2, SDthresholdConjAmp for conditions 4 and 5.
4. The method checks which points pass the gradient conditions. The threshold for the gradient is set in units of standard deviation (relative to the data in the current block), SDthresholdGrad for condition 3, SDthresholdConjGrad for condition 4.
5. The amplitude of the signal envelope is calculated by first performing bandpass filter and then finding the absolute value of the Hilbert transform of the resulting signal, the result is compared to the thresholds. The threshold for the envelope amplitude is set in units of standard deviation (relative to the data in the current block), SDthresholdEnv for condition 1, SDthresholdConjEnv for condition 5.
6. The points suspected as spikes are detected by finding the points where any of the stated conditions is true (either all 5 or less as set by class properties). Note: If the property isDisjunction is set to false, points are detected if the conjunction of the conditions is true (i.e. all the conditions need to be met).
7. In the method findSequences:
   1. Spikes who are too short in duration (i.e. number of consecutive sample points which satisfy the detection condition) are removed. The duration threshold is set by minLengthSpike.
   2. Points which passed the detection condition and are close to one other are merged as one spike (distance is less than the duration set in the property minDistSpikes).
8. Spikes which have too many NaNs or zeros in their vicinity (more than percentageOfNansAllowedArounsSpike in the minDistSpikes/2 ms before and after the peak) are removed.
9. Parameters calculated in the detection process are stored in in the struct peakStats:
10. indsPerPeak is a cell array element i contains all the indices in the data of the i’th spike (i.e. all the indices from the beginning to the end of the spike).
11. zscoresPerPeaksEnv, zscoresPerPeaksAmp, zscoresPerPeaksGrad are cell arrays where each element i contains the z-score values at all the i’th spike indices for the amplitude of the signal envelope, the amplitude, and the gradient correspondingly.
12. zscoresPerPeaksMax is an array, where each row corresponds to a spike and each column to a parameter (z-score for the envelope, gradient and amplitude). The array contains the maximal value over the corresponding spike’s indices for each parameter.
13. passedConditions is an array of Boolean values, where each row corresponds to a spike and each column to a condition (in total 5 conditions). The array stores for each spike what conditions have been satisfied in its detection process.

**Plotting the results**

plotSpikeWaves plots the detected spikes (within a short segment of surrounding data), and can also plot the zscores of each of the parameters of interest (envelope amplitude, amplitude, gradient) with their corresponding thresholds for a visual inspection of which of the parameters passed the threshold. The parameter blockSizeToPlot controls how many spikes are plotted per panel. If the zscores are plotted only one spike should be plotted per panel.