crcns.org hc3 processing flowchart
version 0.9 (July 1, 2014)

The following is a flowchart of the processing of neural recording data that generates different file types in the CRCNS.org hc3 data set. It is very similar to the flowchart for the hc2 data set, but with some difference because different sampling frequencies were used for some of the data in the hc3 data set.

Experiment:

Animal with 4 (or 8) shank probe. Each shank has 8 recording sites, making 32 or 64 possible recording sites (channels).

```
  |
  | Record data from channels (amplify by 1000x, record at 20Khz or 32,552Hz sample
  | rate, bandpass 1-5kHz).  Recordings made by either DataMax recording device
  | (DataMax system; RC Electronics) at 20kHz, or a NeuraLynx recording device
  | (NeuraLynx system) at 32,552Hz.  Data sets recorded using DataMax (20KHz) are:
  | ec012, ec013, ec014, ec016, f01_m, g01_m, i01_m, j01_m.  Data sets recorded using
  | NeuraLynx (32,552Hz) are: gor01, pin01, vvp01, ec014 (ec014.n329 only, all other
  | sessions from rat ec014 were recorded by DataMax).  The sampling frequency is also
  | available in .xml files.
  |
  | Also record video of animal movements.
  V
```

.dat file. Raw data or "wideband data." Contains everything – LFP (Local Field Potentials) and spikes. Has up to 32 (four shank probe) or 64 (eight shank probe) short integers (2 bytes, signed) every time step. (One integer from each recording site, i.e. channel). Actual number of channels is specified in the .xml file and is not always a multiple of 8 because bad channels (due to very high impedance or broken shank) were removed from the data.

.xml file. Has information about recording. Also used for viewing data using neuroscope, can be loaded into Matlab using the LoadXml script and can be viewed using ndmanager (a GUI to browse the xml file).

.mpg file. Video recording of animal position during the experiment. The video shows LEDs attached to the animal. This is used to generate the **.whl** file, which contains the coordinates of the position of the LEDs.

.led file. Contains recordings (at 20KHz sampling rate) of the synchronization pulse that drives a stationary flashing LED visible in the .mpg file. Format is one short integer (2 bytes, signed) per sample.

Post experiment processing of .dat file:

.dat file (Wideband data, same as .dat file above).

```
  |
  |  Downsample to 1250 Hz (DataMax recording device) or 1252 Hz (NeuraLynx recording
  |  device).  Done using script 'downsample.m' (in scripts download).
  |
  V
```

.eeg file. Contains LFP data, below 625 Hz. LFP data is that part downsampled below the spike range to include only synaptic or membrane fluctuation related currents. Has same number of channels as .dat file, but 16x less samples (1.25 Khz, vs 20Khz) or 26x less (1252 vs 32552 Hz).

.dat file (Wideband data, same as .dat file above).

```
  |
  |  High-pass filter (800 Hz high-pass).
  |
  V
```

.fil file. Contains high pass filter output. Not included because they are removed once processing is done to save space. Generated by firfilter function with parameters normally 800 Hz to 5000 Hz bandpass-filtering.

```
  |
  |  Spike detection.  Works as follows:
  |    1. Data from all sessions that are recorded on the same day with the
  |       same electrode configuration are concatenated and used for spike sorting.
```

2. A mean RMS is computed for a sample section of about 100 seconds on all channels.
3. A threshold is set to N standard deviations above mean RMS, where N is a parameter. N=7 is a typical value, but may not have been used in all cases.
4. Peaks in RMS power above the threshold are detected as spikes. Spike timing is aligned to the trough in the filtered signal and the spike wave shape is extracted from filtered signal. Timing alignment is done using oversampling (a type of interpolation beyond the 20K Hz sampling rate) to improve the precision. For an example, see: http://caton.googlecode.com/hg/doc/_build/html/detect_extract.html#alignment
5. Results of spike sorting are distributed to each session, so that the data files for each session contain the spike data for that session. (This step reverses the contationation of session data done in step 1 above.)

Produces three file types:

V

.threshold files. (Generated in step 3 above). Precise threshold values are not important because thresholds are well below what's needed to detect spikes. In addition to valid spikes, there is a lot of "hash" noise detected above the thresholds. These are sorted as noise (cluster 1 below), which includes small amplitude units. This does not affect the good units.

.res files (.res.1, res.2, res.3, res.4). Time of each spike (time stamp, in 20kHZ or 32,552 hz sample rate).

.spk files (.spk.1, spk.2, spk.3, spk.4). One .spk file for each probe. Contains waveform of every detected spikes. Each spike stored as 32 samples x 8 (each site on probe). Total time recorded for each spike waveform is: (32 / 20khz) = 1.6ms or 54/32,552 = 1.6 ms. Captures most of the AHP (afterhyperpolarization), but maybe not the full one, especially since it's filtered at 800 Hz high-pass, whereas AHP power could be much lower.

.spk (Spike waveform files)

Compute PCA (Principal Component Analysis) features of each spike. The script used to do this is similar to the "ndm_pca" script in the "ndmanager-plugins" suite. (Described in the "scripts and programs" document).

V

.fet files (Contain features of each of the spikes in the .spk files). An ascii file. First line contains the number of features, which is the number of columns of the matrix that follows. Each row has features for one spike. The number of spikes is the same as in the spk and res files (above) and the clu file (below) for this electrode. The features are:
Ch1PC1 Ch1PC2 Ch1PC3 Ch2 PC1 Ch2 PC2 ... ChMPC3 Peak2TroughAmp Peak2Baseline Trough2Baseline Width TimeOfSpike

ChXPCY corresponds to the Y's principal components of the waveshape of this spike on channel X (there would be M channels in this description for this electrode group). Typically this is 8 (all channels are good in the electrode group. Some of them might have 7, if 1 recording site was bad. The next 3 features (Peak2TroughAmp Peak2Baseline Trough2Baseline) are computed on the channel with maximal amplitude of the spike. (The order of these three values may be different). Having three values is redundant (since one can be computed from the other two) but can be useful when doing manual spike sorting using the GUI. Width is not well computed, and is not used. Lastly, the TimeOfSpike is simply the .res.N added as a column. This is used as a dimension for automatic clustering to track drifting clusters.

Run KlustaKwik to cluster the spikes (do spike sorting). See "scripts and programs" document. KlustaKwik has an option "–UseFeatures" which allows specifying which features to use. Included is a sample script "KKsubmit" which calls KlustaKwik. It's only useful to illustrate the call, because the exact options used to cluster these data were probably different than what is in the script. See the KlustaKwik manual for information about available options.

V

.clu files. (.clu.1, ...clu.4). First line is number of classes (spike types) found. Subsequent lines indicate which class the corresponding spike (in .fet files) is in.

Note: Number of rows of data in .res, .spk, .fet and .clu files should be the same, after taking into account that the first line in .fet files is the number of input dimensions and the first line in .clu files is the number of classes.

```
  |
  |  Compute some statistics.  This is an optional step.  Code to do this is similar
  |  to what is in the Kluster.m script, although that script is more general and was
  |  designed to create files that could be loaded into Klusters for other data sets.
  |
  |
  | Produces auxiliary files:
  |
  |
  V
```

.m1m2. and .mm. files. Auxiliary files created to help Klusters visualize quickly the spikes of subset of clusters in the axis of feature space. These files contain the max/min of the axis and the variance and the mean in each dimension, they are used to normalize the display. They are convenient to have computed beforehand, since recomputation takes time and the GUI doesn't need to spend this time.

```
  |
  |  Run Klusters (manual adjustment).
  V
```

.clu files, (modified). .clu files are modified by manual changes made to clustering provided by KlustaKwik. Clusters are merged, split, and all apparent artifacts/noisy clusters marked as such.

Summary.

1. dat has raw data.
2. eeg has LFP (local field potential) data.
3. res file has time of spike.
4. spk file has spike waveform (same data as in .dat, but for each spike waveshape).
5. fet has PCA features used to do spike sorting.
6. clu file has class (classification) for each spike, i.e. result of spike sorting.

dat, eeg and spk.* files are all binary. res, fet and clu are ascii files.

Scripts and programs:

See the **crcns-hc2-scripts-and-programs** document for information about how to obtain the scripts and programs referenced here.