<https://github.com/cortex-lab/Suite2P>

Copied Instructions from above website

This code was written by Marius Pachitariu and members of the cortexlab (Kenneth Harris and Matteo Carandini). It is provided here with no warranty. For support, please open an issue directly on github.

**I. Introduction**

This is a complete, automated pipeline for processing two-photon Calcium imaging recordings. It is simple, fast and yields a large set of active ROIs. A GUI further provides point-and-click capabilities for refining the results in minutes. The pipeline includes the following steps

1. X-Y subpixel registration --- using a modification of the phase correlation algorithm and subpixel translation in the FFT domain. If a GPU is available, this completes in 20 minutes per 1h of recordings at 30Hz and 512x512 resolution.
2. SVD decomposition --- this provides the input to cell detection and accelerates the algorithm.
3. Cell detection --- using clustering methods in a low-dimensional space. The clustering algorithm provides a positive mask for each ROI identified, and allows for overlaps between masks.
4. Signal extraction --- by default, all overlapping pixels are discarded when computing the signal inside each ROI, to avoid using "demixing" approaches, which can be biased. The neuropil signal is also computed independently for each ROI, as a weighted pixel average, pooling from a large area around each ROI, but excluding all pixels assigned to ROIs during cell detection.
5. Automatic and manual curation --- the output of the cell detection algorithm can be visualized and further refined using the included GUI. The GUI is designed to make cell sorting a fun and enjoyable experience. It also includes an automatic classifier that gradually refines itself based on the manual labelling provided by the user. This allows the automated classifier to adapt for different types of data, acquired under different conditions.
6. Spike deconvolution --- cell and neuropil traces are further processed to obtain an estimate of spike times and spike "amplitudes". The amplitudes are proportional to the number of spikes in a burst/bin. Even under low SNR conditions, where transients might be hard to identify, the deconvolution is still useful for temporally-localizing responses. The cell traces are baselined using the minimum of the (overly) smoothed trace.
7. Neuropil subtraction --- coefficient is estimated iteratively together with spike deconvolution to minimize the residual of spike deconvolution. The user is encouraged to also try varying this coefficient, to make sure that any scientific results do not depend crucially on it.

**II. Getting started**

The toolbox runs in Matlab (+ one mex file) and currently only supports tiff file inputs. To begin using the toolbox, you will need to make local copies (in a separate folder) of two included files: master\_file and make\_db. It is important that you make local copies of these files, otherwise updating the repository will overwrite them (and you can lose your files). The make\_db file assembles a database of experiments that you would like to be processed in batch. It also adds per-session specific information that the algorithm requires such as the number of imaged planes and channels. The master\_file sets general processing options that are applied to all sessions included in make\_db, UNLESS the option is over-ridden in the make\_db file. The global and session-specific options are described in detail below.

For spike deconvolution, you need to run mex -largeArrayDims SpikeDetection/deconvL0.c (or .cpp under Linux/Mac). If you're on Windows, you will need to install Visual Studio Community in order to mex files in matlab.

Below we describe the outputs of the pipeline first, and then describe the options for setting it up, and customizing it. Importantly, almost all options have pre-specified defaults. Any options specified in master\_file in ops0 overrides these defaults. Furthermore, any option specified in the make\_db file (experiment specific) overrides both the defaults and the options set in master\_file. This allows for flexibility in processing different experiments with different options. The only critical option that you'll need to set is ops0.diameter, or db(N).diameter. This gives the algorithm the scale of the recording, and the size of ROIs you are trying to extract. We recommend as a first run to try the pipeline after setting the diameter option. Depending on the results, you can come back and try changing some of the other options.

Note: some of the options are not specified in either the example master\_file or the example make\_db file. These are usually more specialized features.

**III. Outputs.**

The output is a struct called dat which is saved into a mat file in ResultsSavePath using the same subfolder structure, under a name formatted like F\_M150329\_MP009\_2015-04-29\_plane1. It contains all the information collected throughout the processing, and contains the ROI and neuropil traces in Fcell and FcellNeu, and whether each ROI j is a cell or not in stat(j).iscell. stat(j) contains information about each ROI j and can be used to recover the corresponding pixels for each ROI in stat.ipix. The centroid of the ROI is specified in stat as well. Here is a summary of where the important results are:

cell traces are in dat.Fcell  
neuropil traces are in dat.FcellNeu  
manual, GUI overwritten "iscell" labels are in dat.cl.iscell

stat(icell) contains all other information:

iscell: automated label, based on anatomy  
neuropilCoefficient: neuropil subtraction coefficient, based on maximizing the skewness of the corrected trace (ICA)  
st: are the deconvolved spike times (in frames)  
c: are the deconvolved amplitudes  
kernel: is the estimated kernel

Less important fields of stat(icell):

xpix, ypix: x and y indices of pixels belonging to this max. These index into the valid part of the image (defined by ops.yrange, ops.xrange).  
ipix: linearized indices ((ypix, xpix) --> ypix + (xpix-1)\*Ly) of pixels belonging to this mask.  
isoverlap: whether the pixels overlap with other masks.  
lam, lambda: mask coefficients for the corresponding pixels. lambda is the same as lam, but normalized to 1.  
med: median of y and x pixels in the ROI (indices for the valid part of the image, defined by ops.yrange, ops.xrange).  
neuropilCoefficient: multiplicative coefficient on the neuropil signal, for correction based on maximal skewness of the corrected trace.  
blockstarts: the cumulative number of frames per block. Clould be useful for concatenating experiments correctly (some planes will have fewer frames/block).

footprint, mrs, mrs0, cmpct, aspec\_ratio, ellipse, mimgProj, skew, std, maxMinusMed, top5pcMinusMed: these are used by the automated classifier to label an ROI as cell or not. see section IX for details.

**IV. Input-output file paths**

RootStorage --- the root location where the raw tiff files are stored.

RegFileRoot --- location on local disk where to keep the registered movies in binary format. This will be loaded several times so it should ideally be an SSD drive.

ResultsSavePath --- where to save the final results.

DeleteBin --- deletes the binary file created to store the registered movies

RegFileTiffLocation --- where to save registered tiffs (if empty, does not save)

All of these filepaths are completed with separate subfolders per animal and experiment, specified in the make\_db file. Your data should be stored under a file tree of the form

\RootStorage\mouse\_name\session\block\*.tif(f)

If you don't want to use this folder structure, see the make\_db\_example file for alternatives. The make\_db\_example file also shows how to group together tiffs from different experiments (i.e. different subfolders within this folder structure).

The output is a struct called dat which is saved into a mat file in ResultsSavePath using the same subfolder structure, under a name formatted like F\_M150329\_MP009\_2015-04-29\_plane1. It contains all the information collected throughout the processing, and contains the fluorescence traces in dat.Fcell and whether a given ROI is a cell or not in dat.stat(N).iscell. dat.stat contains information about each ROI and can be used to recover the corresponding pixels for each ROI N in dat.stat(N).ipix. The centroid of the ROI N is specified in dat.stat(N) as well. Here is a summary of where the important results are:

**V. Options for registration**

showTargetRegistration --- whether to show an image of the target frame immediately after it is computed.

PhaseCorrelation --- whether to use phase correlation (the alternative is normal cross-correlation).

SubPixel --- accuracy level of subpixel registration (10 = 0.1 pixel accuracy)

kriging --- compute shifts using kernel regression with a gaussian kernel of width 1 onto a grid of 1/SubPixel

NimgFirstRegistration --- number of randomly sampled images to do the target computation from

NiterPrealign --- number of iterations for the target computation (iterative re-alignment of subset of frames)

smooth\_time\_space --- convolves raw movie with a Gaussian of specified size in specified dimensions; [t]: convolve in time with gauss. of std t, [t s]: convolve in time and space, [t x y]: convolve in time, and in space with an ellipse rather than circle

++ Block Registration (for high zoom/npixels - assumes scanning is in Y direction) ++

nonrigid --- set to 1 for non-rigid registration (or set numBlocks > 1)

numBlocks --- number of blocks dividing y-dimension of image (default is 6)

blockFrac --- percent of image to use per block (default is 1/(numBlocks-1))

quadBlocks --- interpolate block shifts to single line shifts (6 blocks -> 512 lines) by fitting a quadratic function (default is 1)

smoothBlocks --- if quadBlocks = 0, then smoothBlocks is the standard deviation of the gaussian smoothing kernel

++ Bidirectional scanning issues (frilly cells) taken care of automatically ++

**VI. Options for cell detection**

sig --- spatial smoothing constant: smooths out the SVDs spatially. Indirectly forces ROIs to be more round.

nSVDforROI --- how many SVD components to keep for clustering. Usually ~ the number of expected cells.

ShowCellMap --- whether to show the clustering results as an image every 10 iterations of the clustering

getROIs --- whether to run the ROI detection algorithm after registration

**VII. Options for SVD decomposition**

NavgFramesSVD --- for SVD, data has to be temporally binned. This number specifies the final number of points to be obtained after binning. In other words, datasets with many timepoints are binned in higher windows while small datasets are binned less.

getSVDcomps --- whether to obtain and save to disk SVD components of the registered movies. Useful for pixel-level analysis and for checking the quality of the registration (residual motion will show up as SVD components). This is a separate SVD decomposition from that done for cell clustering (does not remove a running baseline of each pixel).

nSVD --- how many SVD components to keep.

**VIII. Options for spike deconvolution**

imageRate --- imaging rate per plane.

sensorTau --- decay timescale.

maxNeurop --- neuropil contamination coef has to be less than this (sometimes good to impose a ceiling at 1, i.e. for interneurons)

**IX. Measures used by classifier**

The Suite2p classifier uses a number of features of each ROI to assign cell labels to ROIs. The classifier uses a naive Bayes approach for each feature, and models the distribution of each feature with a non-parametric, adaptively binned empirical distribution. The classifier is initialized with some standard distributions for these features, but is updated continuously with new data samples as the user refines the output manually in the GUI.

The features used are the following (can see values for each ROI by selecting it in the GUI).

std --- standard deviation of the cell trace, normalized to the size of the neuropil trace  
skew --- skewness of the neuropil-subtracted cell trace  
cmpct --- mean distance of pixels from ROI center, normalized to the same measuree for a perfect disk  
footprint --- spatial extent of correlation between ROI trace and nearby pixels  
mimgProjAbs --- whether this ROI shape is correlated to the shape on the mean image  
aspect\_ratio --- of an ellipse fit to the ROI

**X. Example database entry**

Look into make\_db\_example for more detailed examples.

The following is a typical database entry in the local make\_db file, which can be modelled after make\_db\_example. The folder structure assumed is RootStorage/mouse\_name/date/expts(k) for all entries in expts(k).

i = i+1; db(i).mouse\_name = 'M150329\_MP009'; db(i).date = '2015-04-27'; db(i).expts = [5 6]; % which experiments to process together

Other (hidden) options are described in make\_db\_example.m, and at the top of run\_pipeline.m (set to reasonable defaults).