**IMPORTANT**

Much of the code provided has parameters of our optical setup explicitly or implicitly coded. If you are using this package for your own data analysis, many segments of the code will need to be adjusted to match your optical setup.

**Analysis Instructions**

1. Revise read\_cellLists.m to load corresponding data
2. Keep the output ‘signals’ from read\_cellLists in your workspace and run analyzeData

See headers and comments within each function for additional documentation. Below are terse descriptions of each component.

**analyzeSignalData**

This script ties all functions together and requires the cell array ‘signals’ to have previously been constructed with read\_cellLists.m

A similarity metric is constructed for all cells based on cytoplasmic GFP, Draq5 and HADA signals and cells are aggregated in bins with weights determined by their similarity to the bin center. Both the bins (with colormapped weights) and the aggregated data are plotted as output.

**center\_interp\_signal**

Centers a signal and returns an interpolant with 300 sampled points.

**findPeaks**

A specialized function for locating peaks for the fieldname provided in the signals cell array (see read\_cellLists.m to generate a signals cell array).

Peaks must be approximately near the cell center, and local maxima in that region are used as initial guesses for peak locations. The bounds of a local maxima are identified by finding the smallest derivative on either side of the maxima and trimming the peak to that region. The position of the peak will then be identified either with a center of mass based method or a fitting routine as specified by the user. The fitting routine fits a biased Gaussian to the peak. The output are peak locations such that peaks(C) is the peak for signals{C}.(fieldname)

**gaussianWeightX**

Returns a smoothed version of x, y determined by the input sigma. For each x(k), weights are given to its neighbors according to a Gaussian profile. These weights are used to return the weighted average of points around each point k.

**loadIMstack**

Helper function to load image stacks into matlab

**pchip\_interp**

Convenience function for matlab’s pchip interpolation

**read\_cellLists**

Load cellLists and images. Identify any cells which have pixels saturating the camera chip and remove. Compile the remaining cells into the signals cell array – that is a matlab cell array containing data about biological cells.

**reformatCellList**

provided a cellList and a cell array of signalNames, extract those signals and recompile into a linear cell array.

**removeSaturatedCells**

Go through the provided cellList and remove any cells from the cellList which have pixels saturating the camera.

**signalConstriction**

Calculate cell constriction in each signal provided for the indicated field with smoothing performed over window size sWin