SegEM: Semi automated image analysis toolkit for Connectomics

For an up-to-date version of this code, please check: http://www.segem.io, also this pdf-readme might be outdated wrt. The README.md supplied in the git repositorium there.

Prerequisites:

- A working matlab jobmanager, 'local' will usually work by default
- Make sure using 'mex -setup' that you have configured a compiler for use on Matlab mex-files (C-language)

Usage

Next start Matlab with the working directory set to the root folder of this code repositorium or navigate there using matlab 'current folder' pane and run initalSettings.m.

Please execute initalSettings.m first, this will start a GUI to set needed information:

- Data Directory = Location of data acquired via segem.io or currently pointer to Frankfurt Fileserver under \\storage.hest.corp.brain.mpg.de\Data\Data\berningm\20150205paper1submission\
- Name of Jobmanager = Name of working Matlab Jobmanager profile ('local' usually works by default)
- Output Directory = Directory to store data and figures generated by scripts
- Chosen Dataset = chose whether you want to work with cortex or retina data & code version

IMPORTANT NOTE: Only cortex is set up and tested completely, for retina volume reconstruction and contact detection based on supplied segmentation is working as well.

If you are not specifically interested in the retina dataset/code version or want to apply the scripts to your own data it is therefore advisable to use cortex.

After settings have been completed, relevant scripts will open which can be executed from top to bottom using Ctrl+Enter and are opened in an order that should be most instructive to follow (e.g. start with top- or leftmost script).

See 'Cell Mode' in Matlab help for more information on using Ctrl + Enter for executing a cell (=between line starting with %%)

The most important parts of the code for CORTEX are (always including some visualization):

+ cnnStart: load training data, train a convolutional neuronal network, paralell network training

- cnnParameterSelection: Automated hyperparameter selection and variation of learning rates for each layer
- mainSegCortex: steps from classification result by CNN to segmentation (grid parameter search for watershed segmentation) including skeleton based split-merger metrics
- bigFwdPassStart: Apply learned CNN classifier (learned in point 1) and watershed segmentation steps (with parameters optimized in 2)
- skeletonBasedCortex: Use segmentation of whole dataset (point 4 in this list) to use skeleton reconstructions for volume reconstructions or contact detection between cell pairs (a segmentation is also provided because step 4 is computationally expensive)

The most important parts of the code for RETINA are:

- main: load training data, train a convolutional neuronal network, paralell network training
- mainSeg: steps from classification result by CNN to segmentation (grid parameter search for watershed segmentation) including skeleton based split-merger metrics
- visSeg: visualize a given segmentation, e.g. object chains, segMovies, errorMovies = detected merger errors etc.
- skeletonBasedRetina: Use segmentation of whole dataset (provided) to use skeleton reconstructions for volume reconstructions or contact detection between cell pairs