Welcome to the SLICEM manual!

SLICEM is meant to help visualize the compositional heterogeneity in your sample and split your particles into groups of related structures. First you will need to perform 2D classification on your data. We prefer to use RELION for this. **You will then need the class averages (.mrcs) and the corresponding particles.star file from the Class2D folder** as inputs for the SLICEM main script and GUI. For the main script, run *SLICEM.py* from your command line with the following options:

```
usage: SLICEM.py [-h] -i MRC_INPUT -o OUTPATH -d DESCRIPTION -m METRIC [-n] -p
                 PIXEL_SIZE [-c NUM_WORKERS]

compare and cluster 2D class averages based on common lines

optional arguments:
  -h, --help            show this help message and exit
  -i MRC_INPUT, --input MRC_INPUT
                        path to mrcs file of 2D class averages
  -o OUTPATH, --outpath OUTPATH
                        path for output files
  -d DESCRIPTION, --description DESCRIPTION
                        name for output file
  -m METRIC, --metric METRIC
                        choose scoring method (Euclidean, L1, cross-
                        correlation, cosine)
  -n, --normalize       zscore normalize 1D projections before scoring (not
                        recommended)
  -p PIXEL_SIZE, --pixel_size PIXEL_SIZE
                        pixel size of 2D class averages in A/pixel
  -c NUM_WORKERS, --num_workers NUM_WORKERS
                        number of CPUs to use
```

We recommend binning your images if they are larger than 100x100 pixels to speed up computation. For metric, either 'Euclidean' or 'L1' perform the best. With our benchmark data, using 8 CPUs took ~1hr and using 50 CPUs took ~5 min. Example input:

$ *python SLICEM.py -i path/to/mixture_2D.mrcs -o path/to/output -d mixture_Euclidean -m Euclidean -p 4 -c 12*

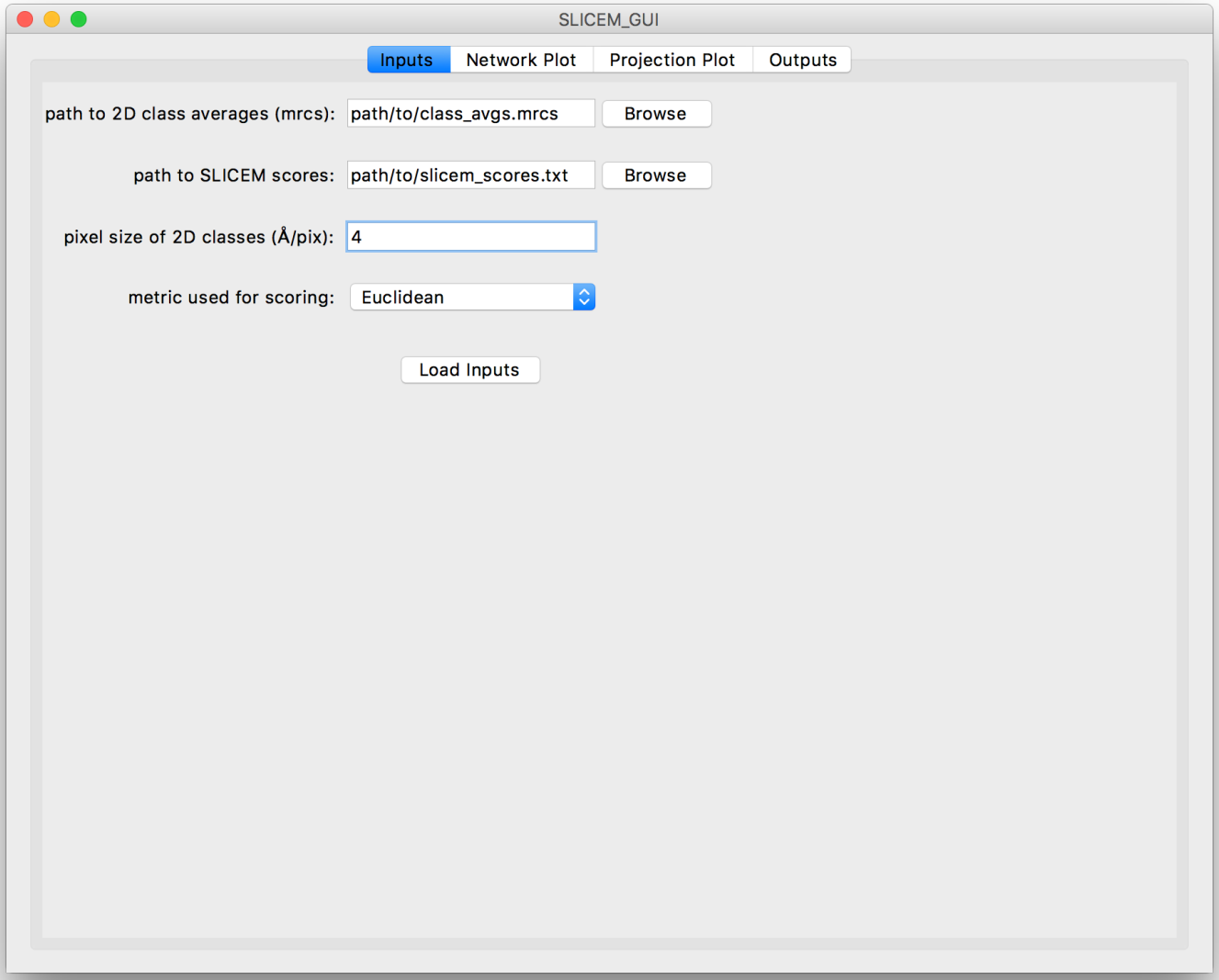
SLICEM was written in python 3+ and requires several non-standard libraries which you might need to install (e.g. the mrcfile library).

After running the main script, you should have a score file in your output directory that you will need to load into the SLICEM GUI.

After running the SLICEM main script, you can use the GUI to visualize different clusterings before generating separate star files for each of the clusters. Start from the command line with:

$ *python SLICEM_GUI.py*

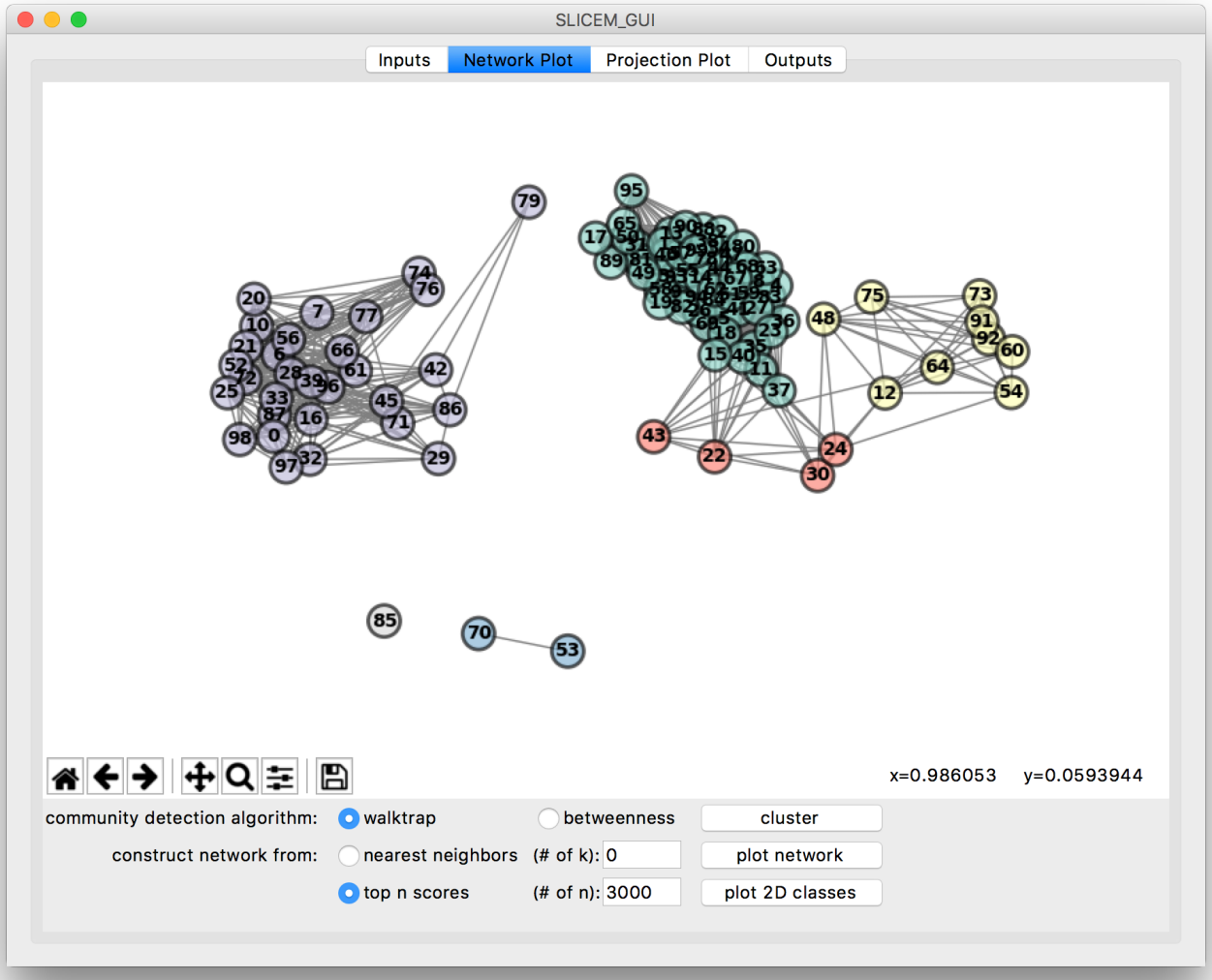
For the Inputs tab, you simply need to fill in the input boxes and press “Load Inputs”. You can then move on to the “Network Plot” tab.



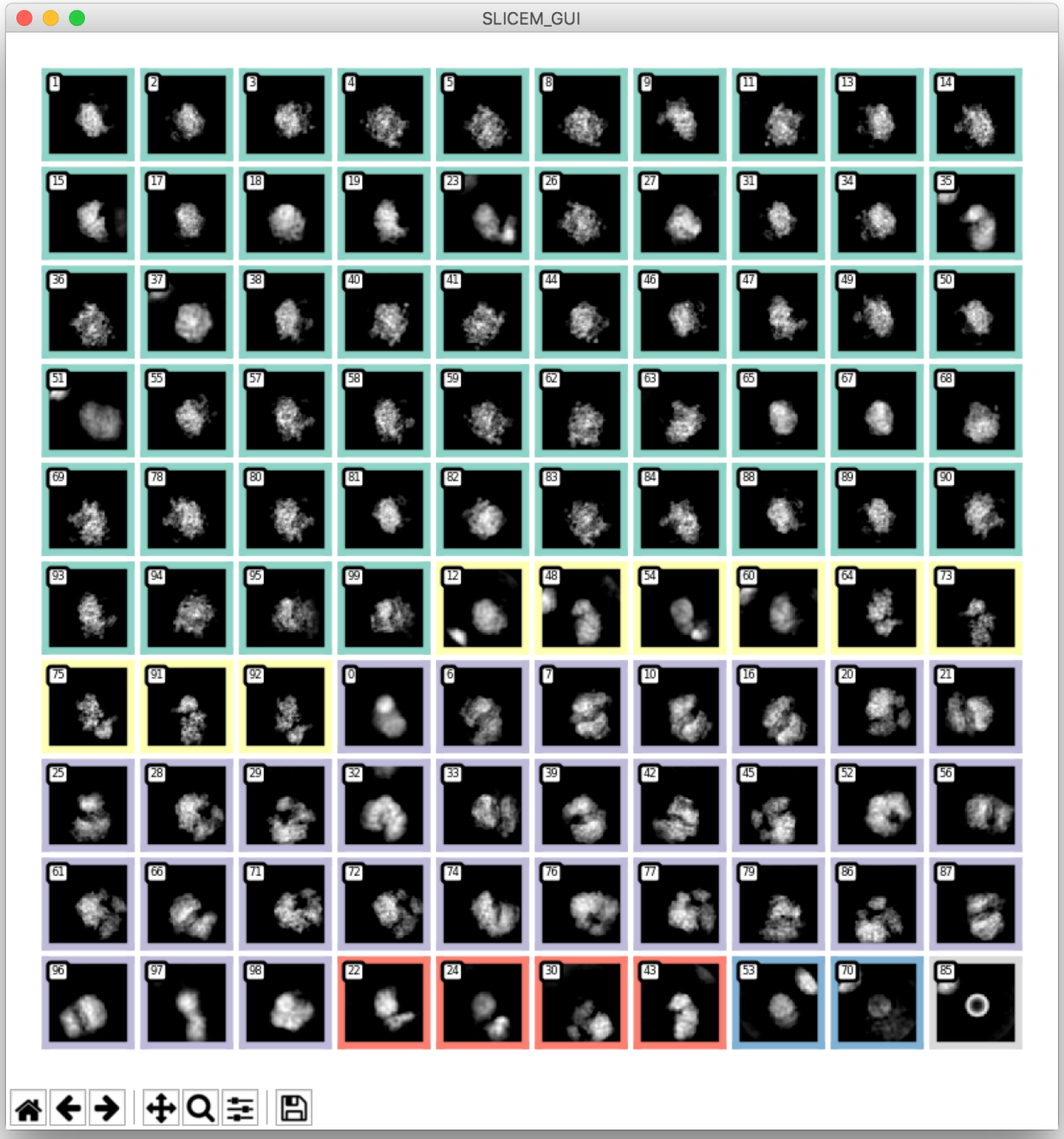
In the Network Plot tab, you can cluster your 2D class averages using various parameters to see how the clusters change. There is no best option so visualization can be helpful.

First select which clustering algorithm you want to use. Then select if you want to use k-nearest neighbors or the top-n scores to pick how many edges are in the network. You will also need to input k or n respectively. A good starting value for k is 5. For n, try between 15-30% of the total scores. 100 class averages have 9,900 pairwise scores (100*100)-100, we don't score self. After picking these parameters press "cluster". **If you change parameters, you will need to press "cluster" again.**

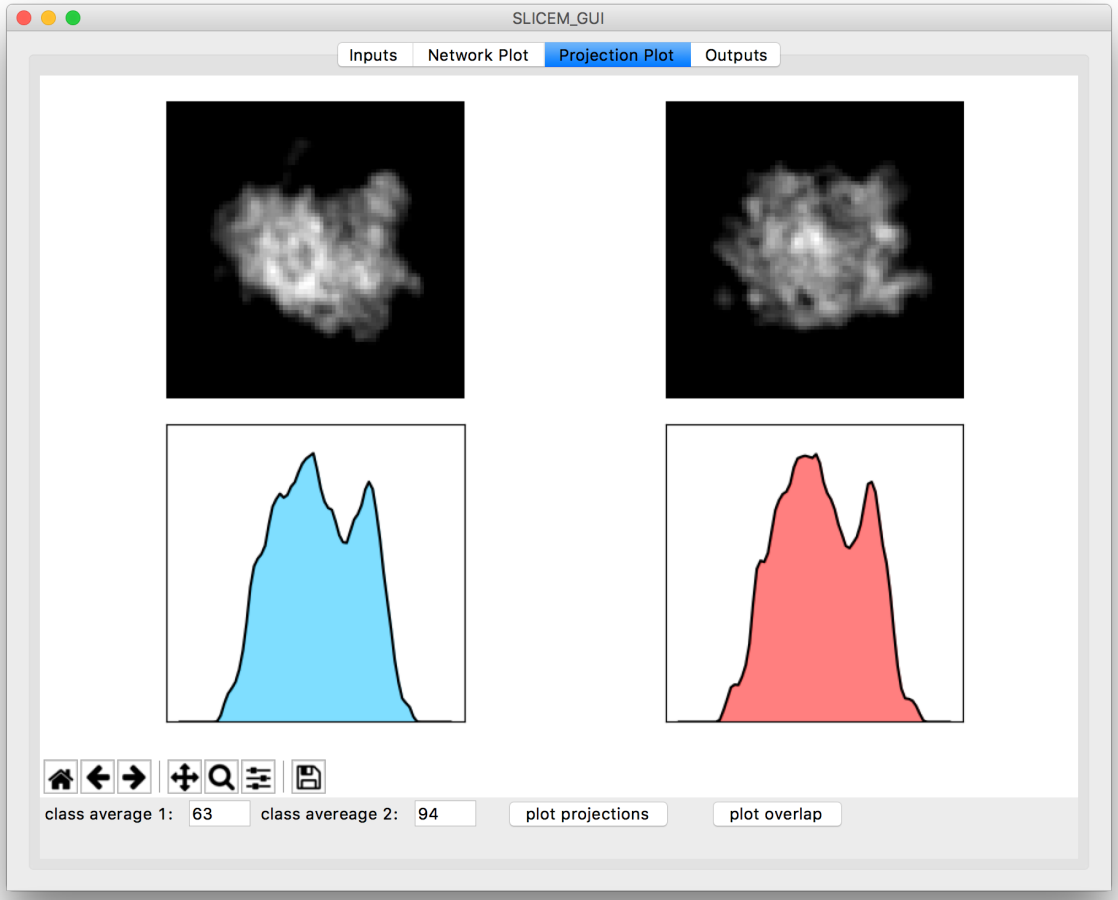
Now you can view the network by pressing "plot network". Plotting networks with nice layouts can be difficult, **if the graph looks bad, press "plot network" again** until you get one you like. You will have the option to save a file containing the edges to use in other programs later. Nodes are colored according to cluster. Dark grey nodes in a cluster are outliers and white nodes are 'singles' (i.e. they do not have a connecting edge to another node). This can occur when using the top-n option as not all nodes will have a top scoring edge.

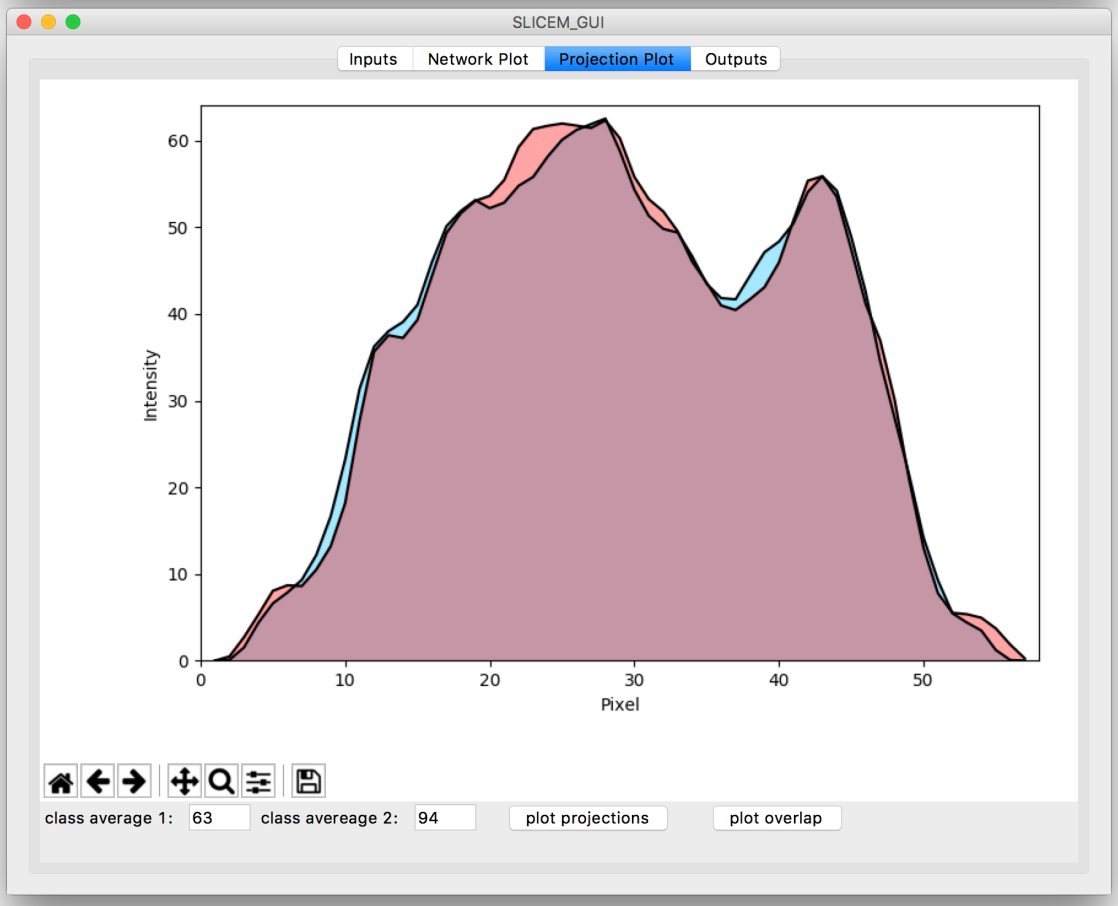


To further inspect the clustering, press the “plot 2D classes” button which will spawn a new window with your 2D class averages colored by cluster. This will take a few seconds to generate.

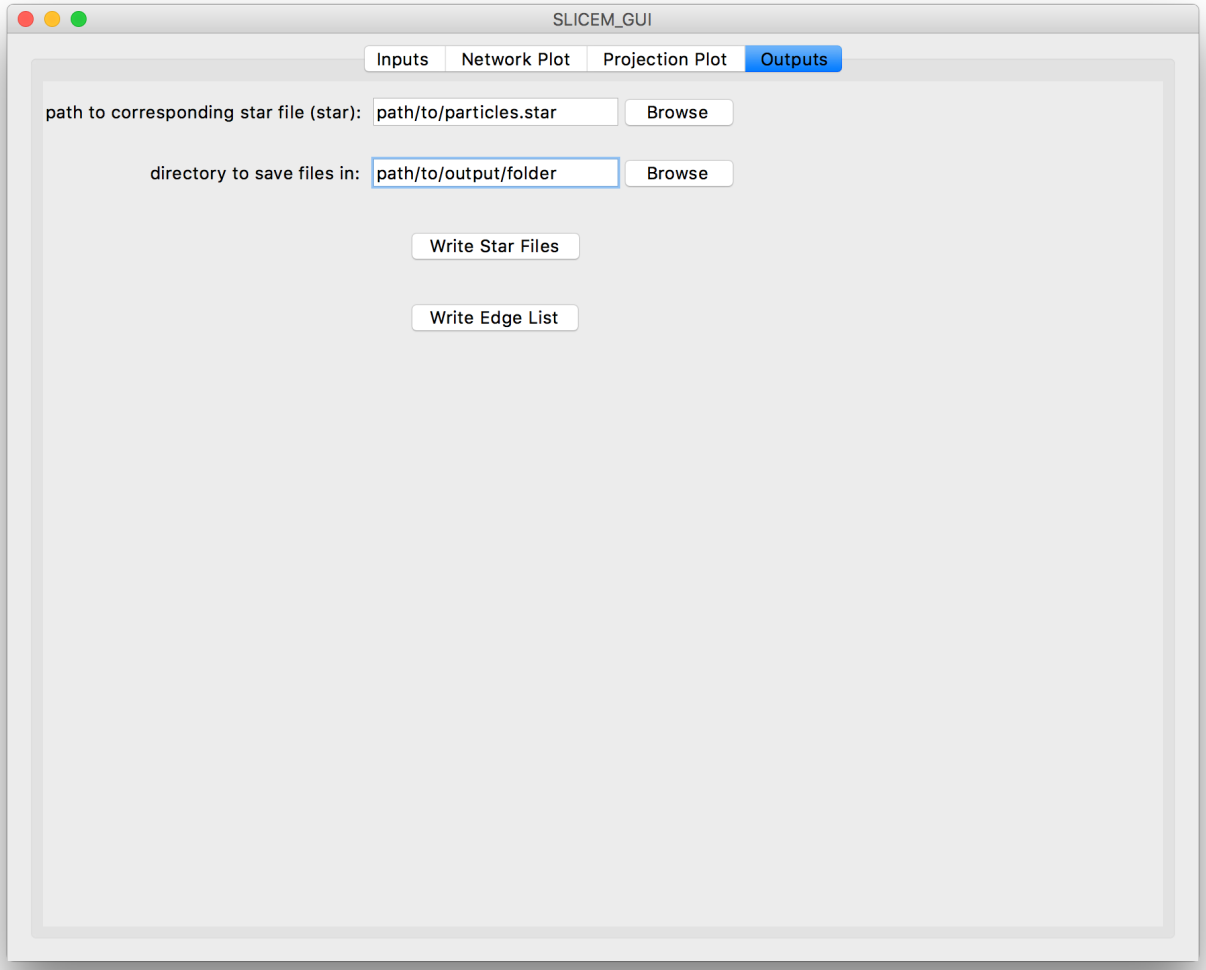


In the Projection Plot tab you can manually inspect the most similar line projection between two different class averages. You can either look at the line projections with their class averages or with the line projections overlaid.





After you have found a set of clustering parameters you like, you can write new star files based on the clusters to use for 3D classification. Make sure you have not changed any parameters and pressed “cluster” since finding the conditions you like. You will need to use the particle.star file that was generated during 2D classification because it contains a column linking the particle to the class average. Once you have filled in the inputs you can generate the new split star files by pressing “Write Star Files”. You can also generate an edge list file that you can use with other programs like Cytoscape to make better looking networks by pressing “Write Edge List”.



Finally, if you have any questions, comments or suggestions please send me an email at:

eric.verbeke@utexas.edu

and thanks for trying SLICEM!