Separating multiple structures using SLICEM

Eric Verbeke

November 22, 2020

Abstract

SLICEM is a tool designed to help sort single particle cryo-EM data into consistent subsets when there are multiple distinct structures present in the data (e.g. from non-purified samples or cell lysates). This tutorial introduces how SLICEM can be combined with the standard single particle processing pipeline. There are two main steps in using this program: (1) generate a score file comparing 2D class averages using the command line, (2) cluster the class averages using the scores in the SLICEM GUI. Before running SLICEM, you will need to particle pick and perform 2D classification. SLICEM was designed to work with the STAR file format and with class averages that are zero-mean normalized to background. Required inputs to SLICEM are the class averages (.mrcs) and the corresponding particles (.star) file *from 2D classification*, as class membership number must be present for each particle.

1) Generate scores comparing 2D class averages

First install the software following the instructions on the SLICEM GitHub page (https://github.com/marcottelab/SLICEM). If you do not have Anaconda you can download and install it from here (https://www.anaconda.com/download). After creating and activating the SLICEM environment, you can see the command line arguments with:

python slicem.py --help

```
usage: slicem.py [-h] -i MRC_INPUT -o OUTPATH [-m {Euclidean,L1,cosine,EMD}]
                 [-s SCALE_FACTOR] [-c NUM_WORKERS]

compare similarity of 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
  -m {Euclidean,L1,cosine,EMD}, --metric {Euclidean,L1,cosine,EMD}
                        choose scoring method, Euclidean default
  -s SCALE_FACTOR, --scale_factor SCALE_FACTOR
                        scale factor for downsampling. (e.g. -s 2 converts
                        100pix box --> 50pix box)
  -c NUM_WORKERS, --num_workers NUM_WORKERS
                        number of CPUs to use
```

The only required arguments are the path to the mrcs file and a path to save the scores. Basic command line example:

python slicem.py -i path/to/input.mrcs -o path/to/output/

We recommend using the default metric of Euclidean distance (**Figure S1**). A scale factor (*-s*) can be added to downsize class averages. This will improve processing speed but can decrease accuracy (**Figure S2**). For initial testing, we recommend scaling to ~100 pix or less (e.g. if the input class averages are 200x200 pix, a scale factor of 2 would convert to 100x100). See **Figure S3** for examples of different scaling. You can use multiple processors to improve speed (*-c*). Command line example using scaling and multiple processors:

python slicem.py -i path/to/input.mrcs -o path/to/output/ -s 2 -c 10

The output score file will be saved to the path you designate as:

slicem_scores_YearMonthDay_HourMinuteSecond.txt

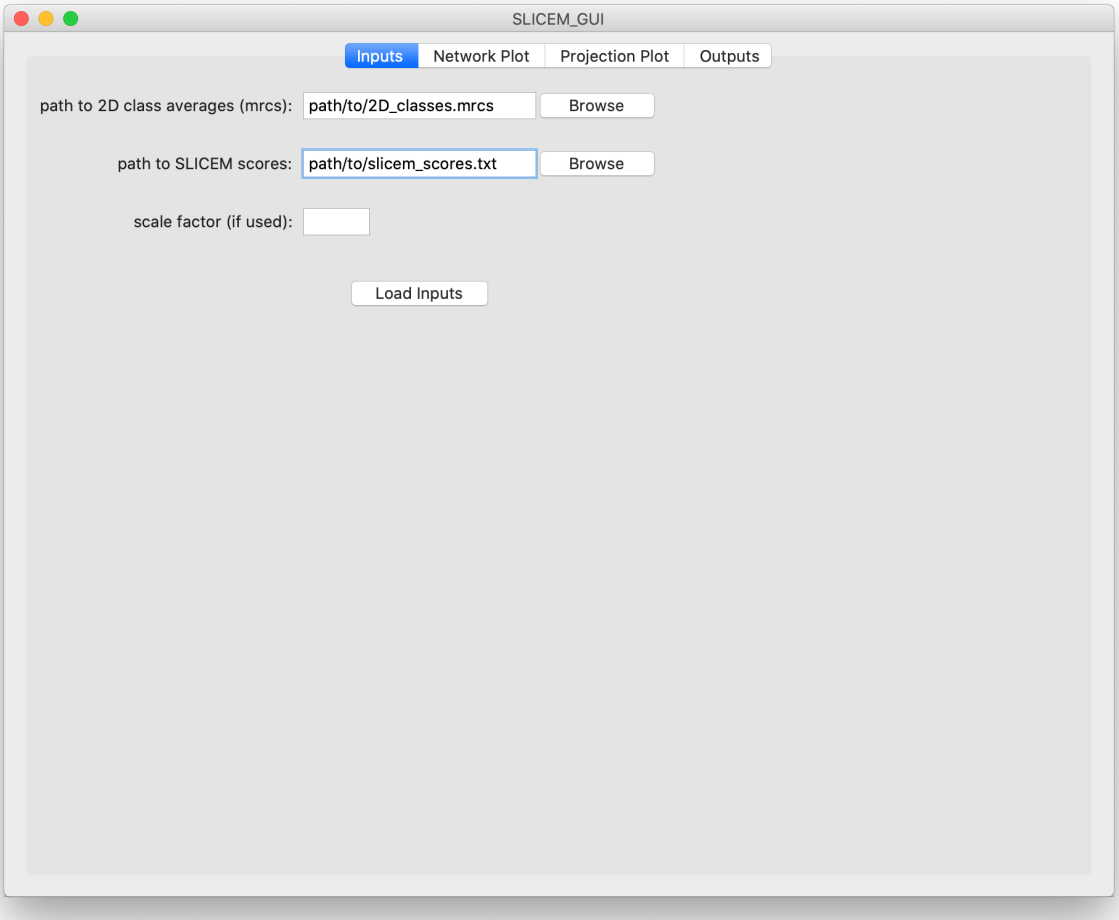
You can then use the GUI for clustering.

2) Visualization and clustering using the GUI

After generating the scores, launch the GUI from the command line:

python slicem_gui.py

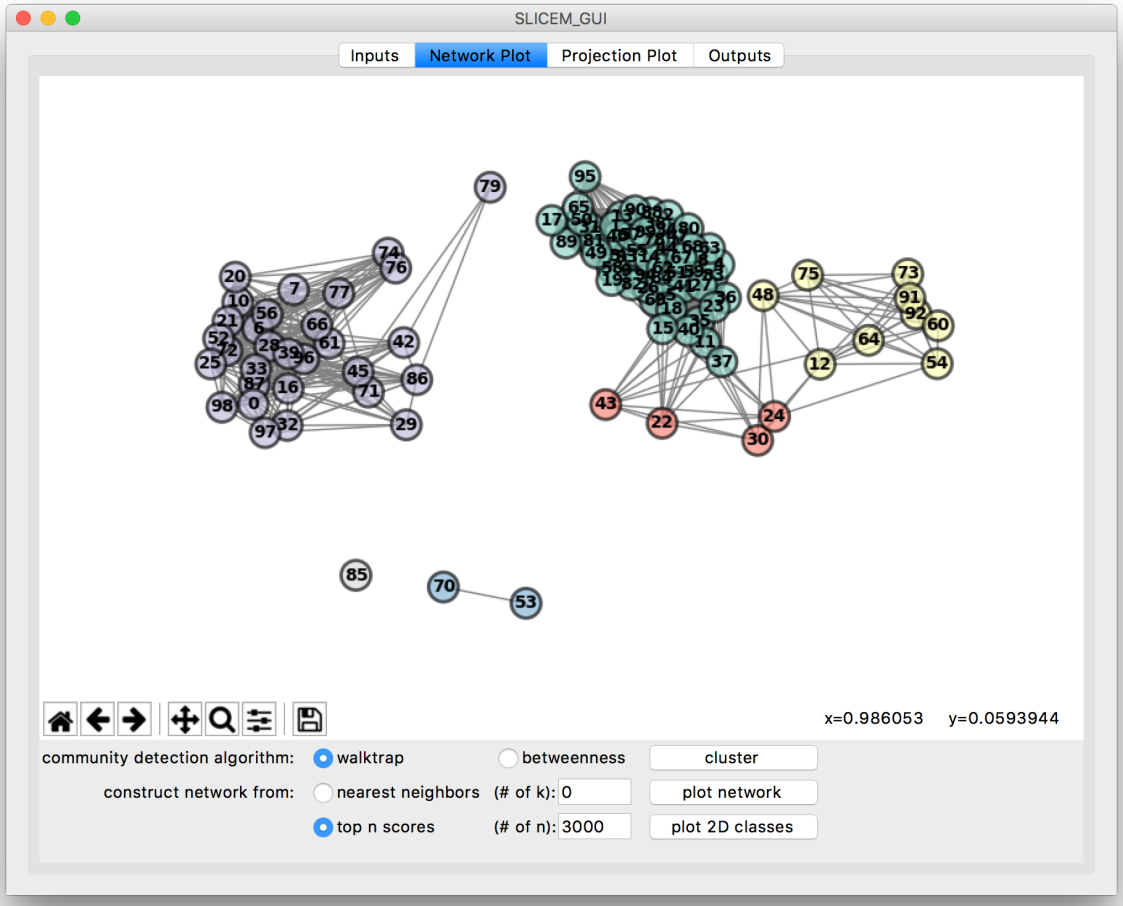
The goal of the GUI is to help visualize the relations between class averages in a network format and to cluster related class averages for subsequent 3D classification. For the Inputs tab, fill in the input boxes and press “Load Inputs”. You can then move on to the “Network Plot” tab. **Note:** the GUI can be slow if running through XQuartz. One solution is to generate the scores on a compute cluster then transfer the results to a local computer to use the GUI.



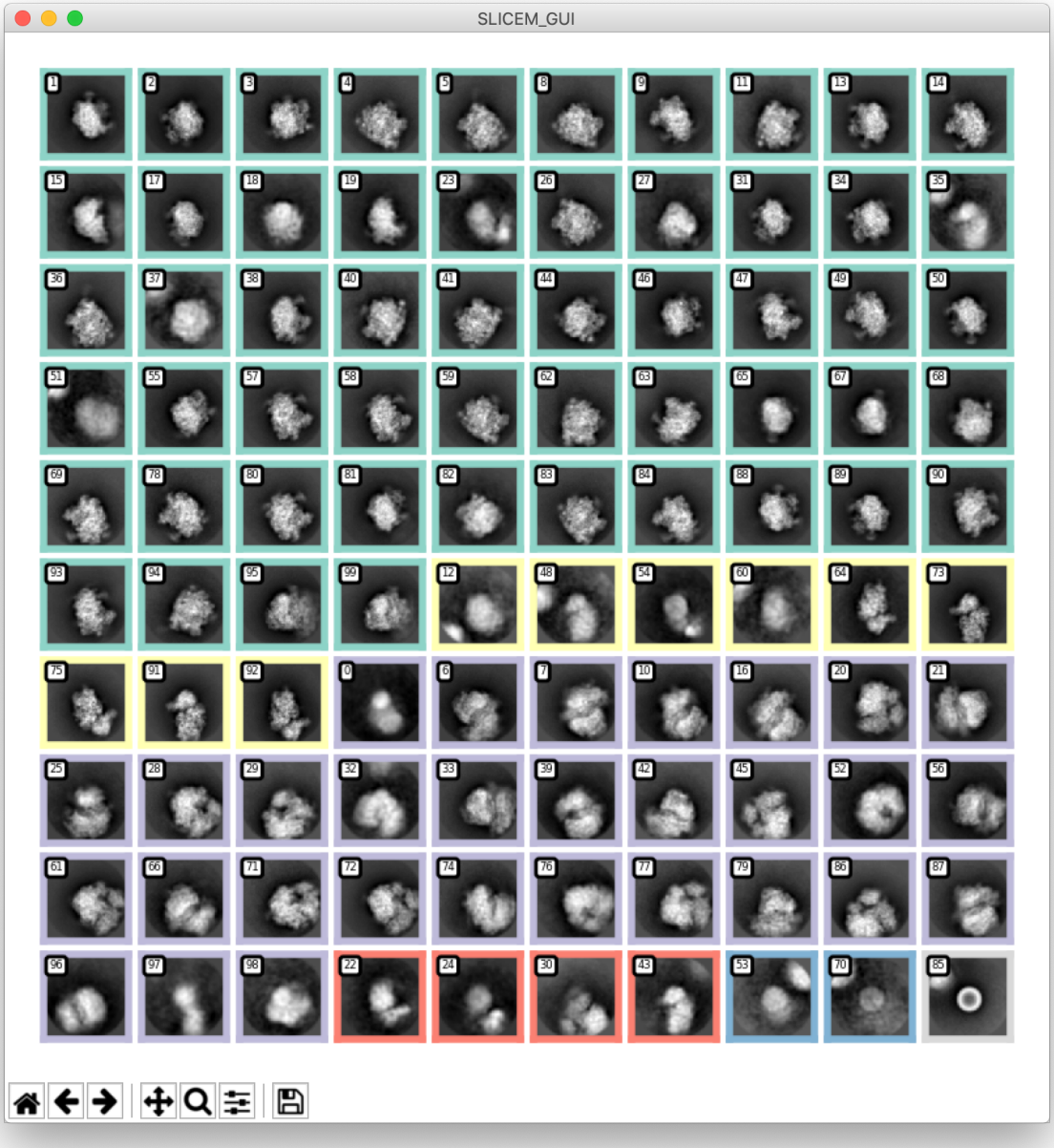
In the Network Plot tab, you can cluster your 2D class averages using various parameters.

First select which clustering algorithm you want to use. We recommend trying “walktrap”. 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 starting with 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.

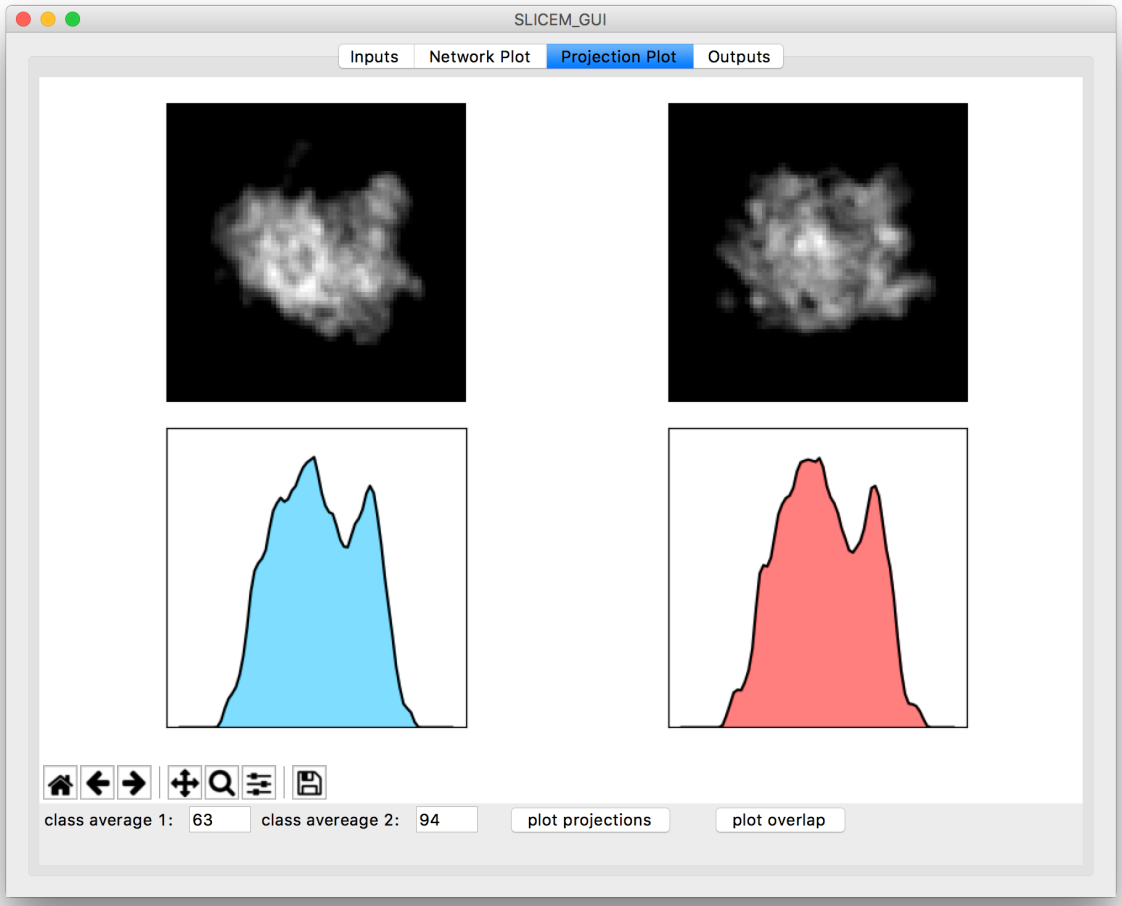
Now you can view the network by pressing “plot network”. As the network is plotted with a stochastic layout algorithm, you might consider pressing "plot network" multiple times to select a clear visual layout. You will have the option to save a file containing the edges for use in other programs later. 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.

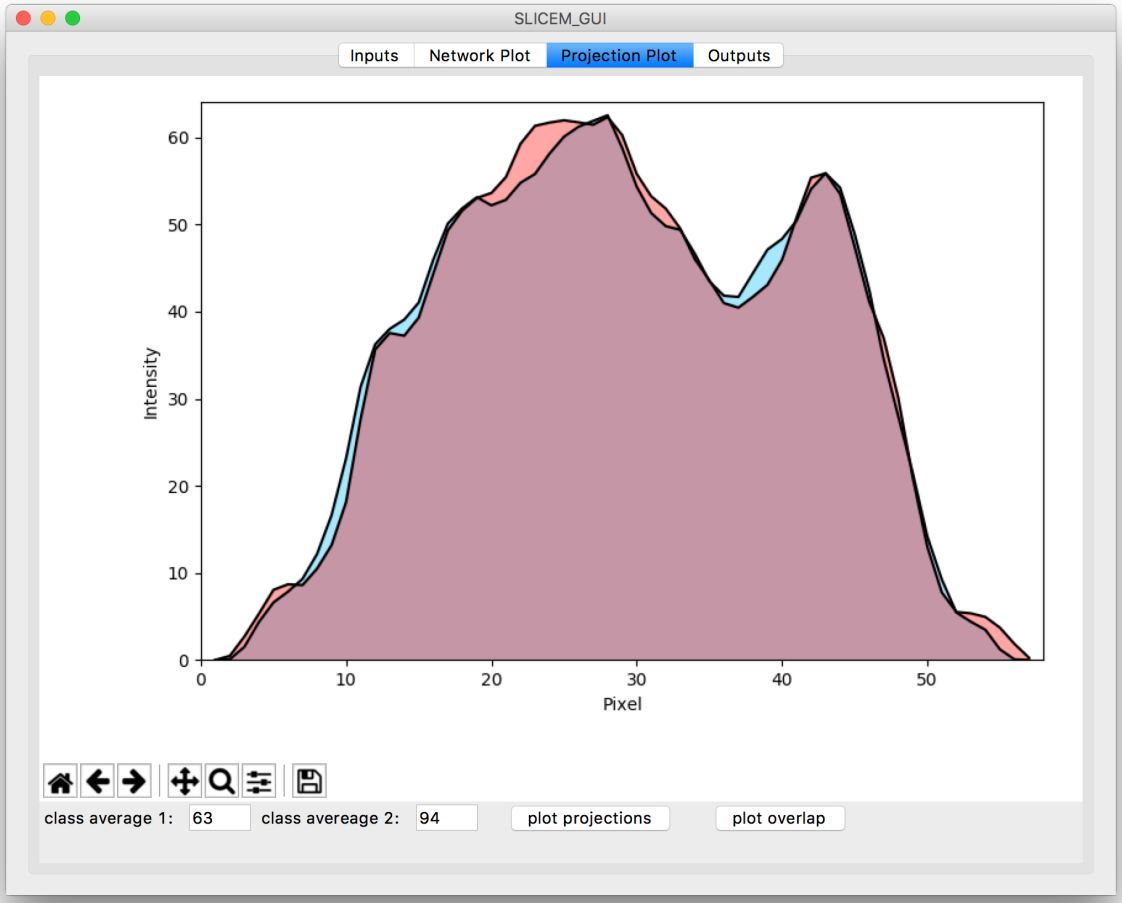


To further inspect the clustering, press the “plot 2D classes” button which will spawn a new window with your 2D class averages colored by cluster. This will take a few seconds to generate. **Note:** you can save any of the plots by clicking the save icon. The images look best if you maximize the window and save the figure as a pdf.



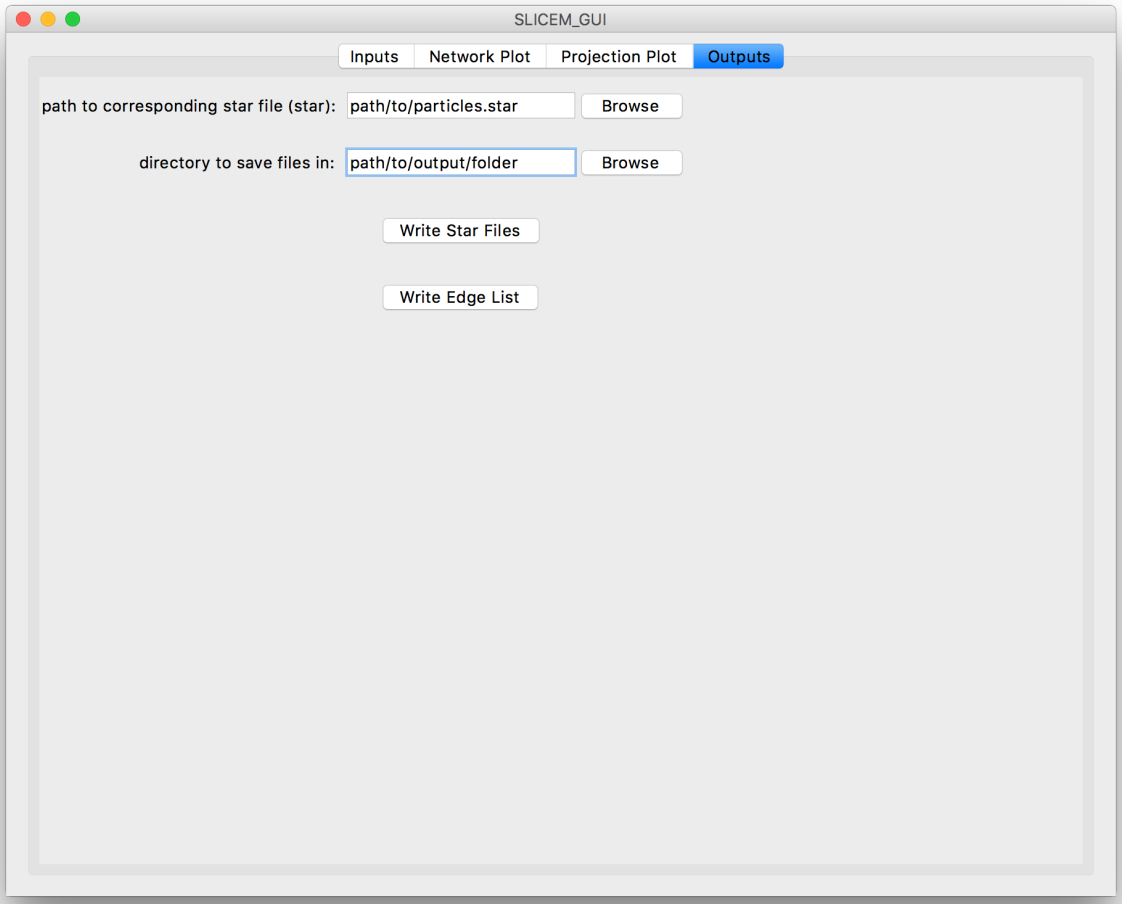
In the Projection Plot tab, you can manually inspect the most similar line projection between two different class averages. You can either look at the line projections with their class averages or with the line projections overlaid.




SLICEM_GUI
Inputs
Network Plot
Projection Plot
Outputs
Intensity
Pixel
0
10
20
30
40
50
60
class average 1:
63
class avereage 2:
94
plot projections
plot overlap

3) Saving new star files from clusters

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!

4) Supplemental Figures

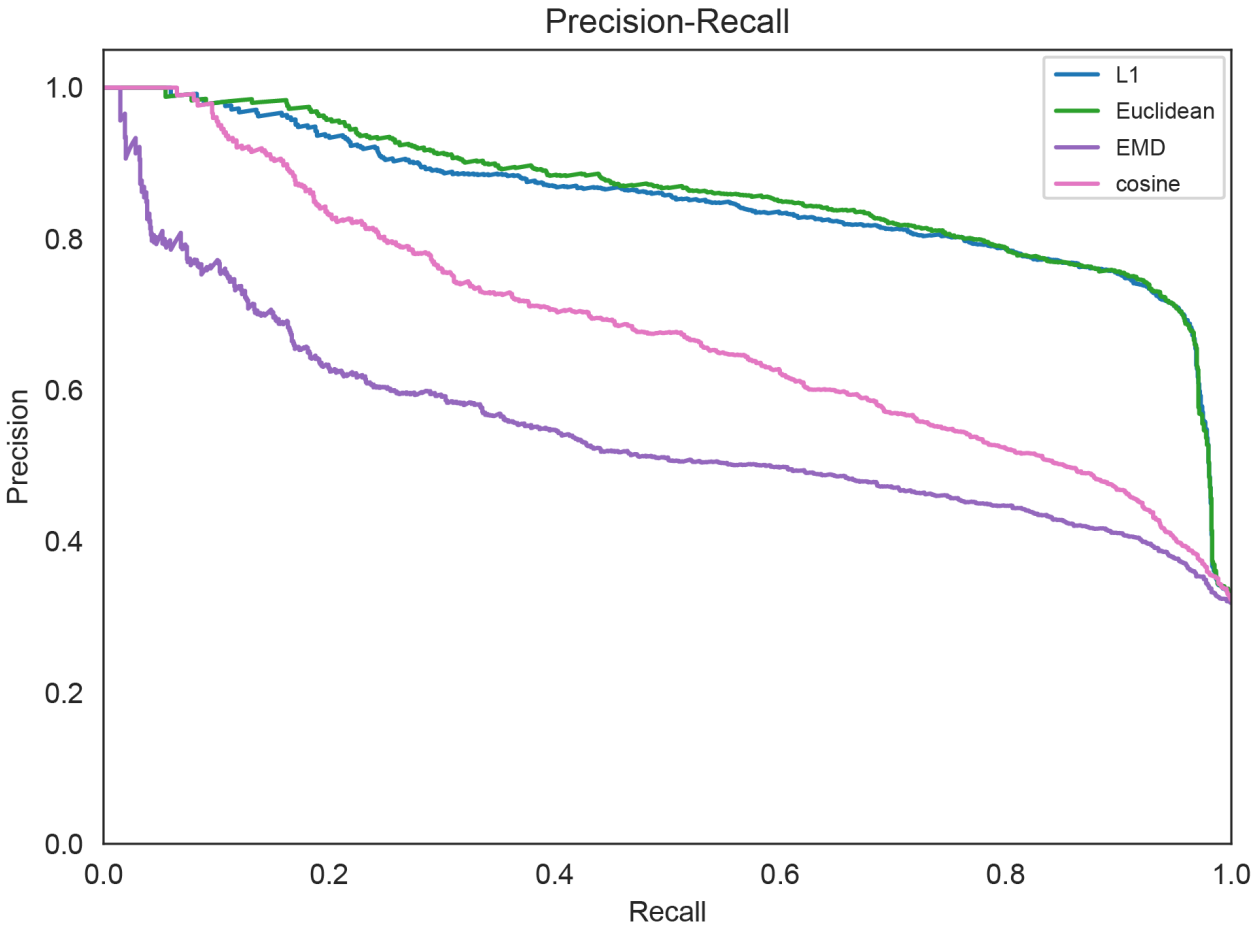


Figure S1. Precision-Recall curves comparing different scoring metrics. Scores were computed from data provided in the manuscript. Euclidean distance shows the best performance.

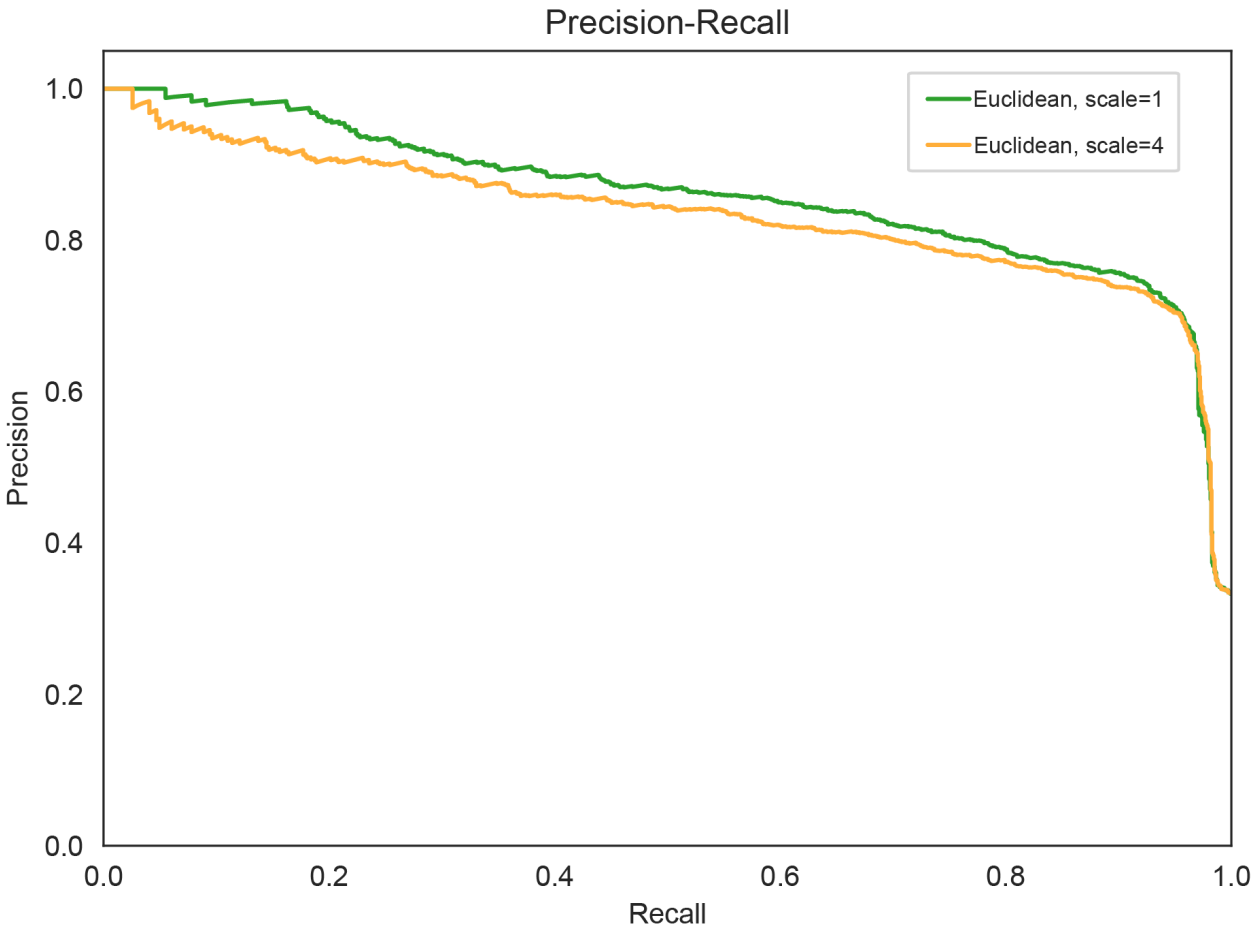


Figure S2. Precision-Recall curves comparing different scaling factors. Scores were computed from data provided in the manuscript.

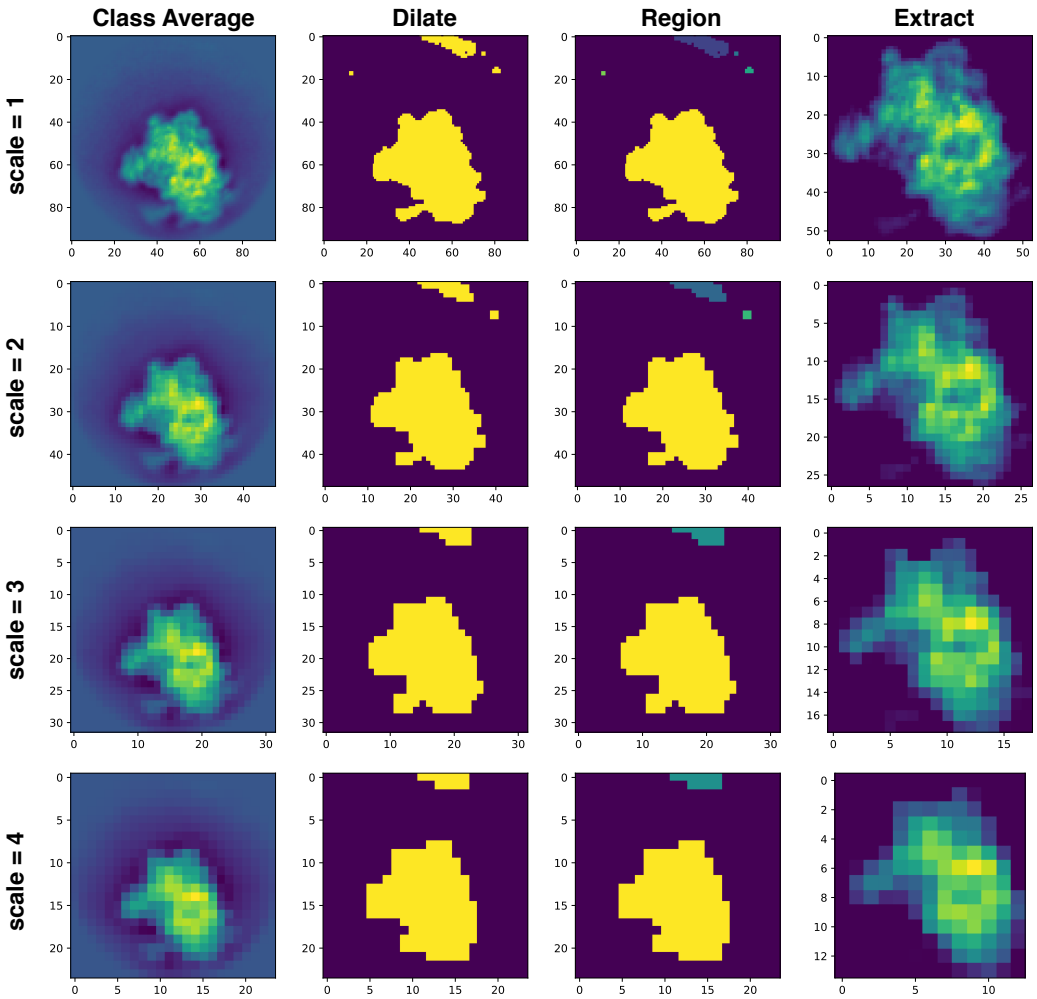


Figure S3. Scaling of class averages. Original class average size 96x96 pixels.