©2017 LabMonti, University of Arizona
Software Created and Written by Ben Wu
Manual Written by Nathan Bamberger

Manual for Distance-Conductance Hierarchical Clustering Software v1.1

Original Article: Uncovering Hierarchical Data Structure in Singe Molecule Transport, B.H. Wu, J.A. Ivie, T.K. Johnson, O.L.A. Monti, J. Chem. Phys., 146, 092321 (2017)

1. Purpose of this Manual

This manual describes how to run the hierarchical clustering software used in Wu et al.; it does *not* describe how the software works or the theory and rationale behind its design. For this background information, please refer to Wu et al.

2. Starting the Software

To start the program, open and run the file "MainAnalyzer.m" in MATLAB. This should cause the main GUI window to appear (see Figure 1). For best results, maximize this window (otherwise certain text labels and outputs may be truncated).

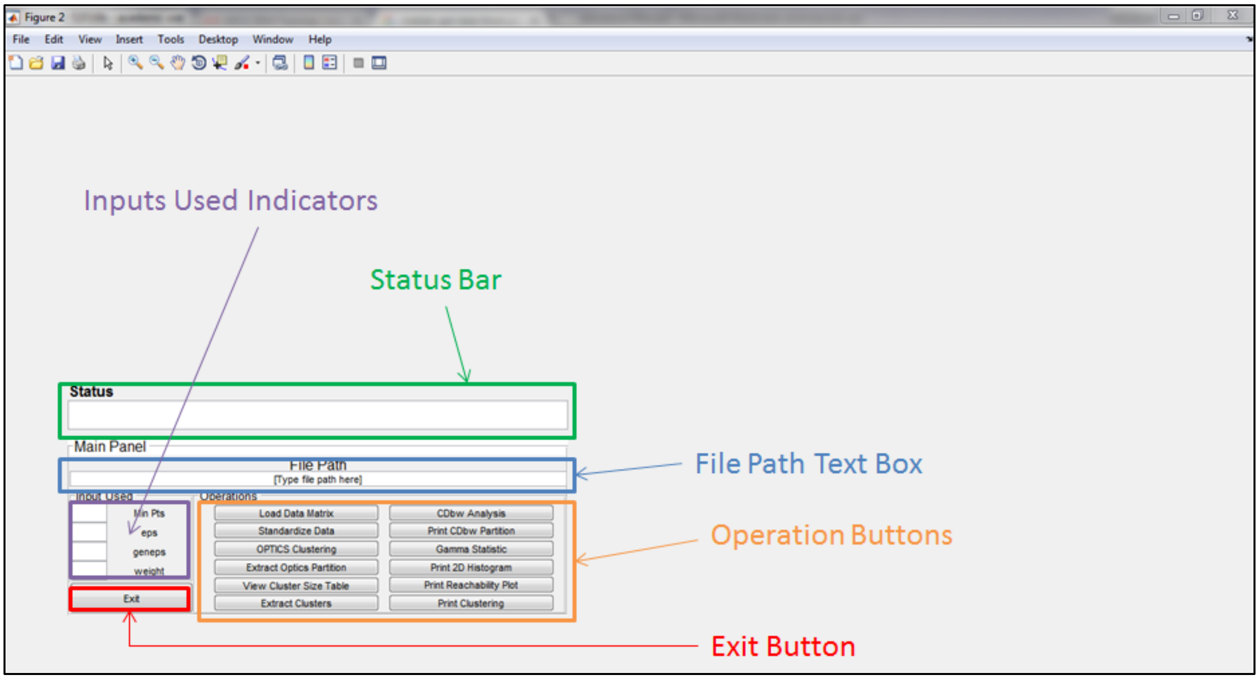

Figure 1. How the main GUI window appears when "MainAnalyzer.m" is run in MATLAB, with different sections of the GUI labeled in color. The empty space in the GUI window will be populated with different plots and tables as the program runs.

3. Running the Software

3.1. Operation Dependencies/Order of Execution

The different operations available in the main GUI window *cannot* be run in an arbitrary order, since certain functions require data that are produced by other functions. The flow chart in Figure 2 displays these dependencies visually; each operation can only be run successfully if all operations "upstream" of it have already been run. If operations are run without their dependencies satisfied, the program will encounter an error. Note that this error will not necessarily be displayed in the GUI.

Figure 2 also shows where in the program's logical flow certain key user-specified parameters are obtained. To change one of the parameters shown in Figure 2, all operations downstream of where the parameter was specified must be re-run. For example, if a user wishes to run the "View Cluster Size Table" operation using a different value of geneps, they must first re-run "Optics Clustering" and "Extract Optics Partition" since both operations appear between geneps and "View Cluster Size Table" in the flow chart. Note that as soon as "Optics Clustering" is re-run, the geneps value displayed in the main GUI window will be updated, but the displayed size table will still correspond to the original value of geneps; to avoid confusion, whenever one of the parameters shown in Figure 2 is changed all operations "downstream" of the operation in which it was changed should be immediately re-run.

Assuming all operations have been re-run after any parameter changes, the output of all operations will correspond to using the values of the input parameters displayed in the main GUI window.

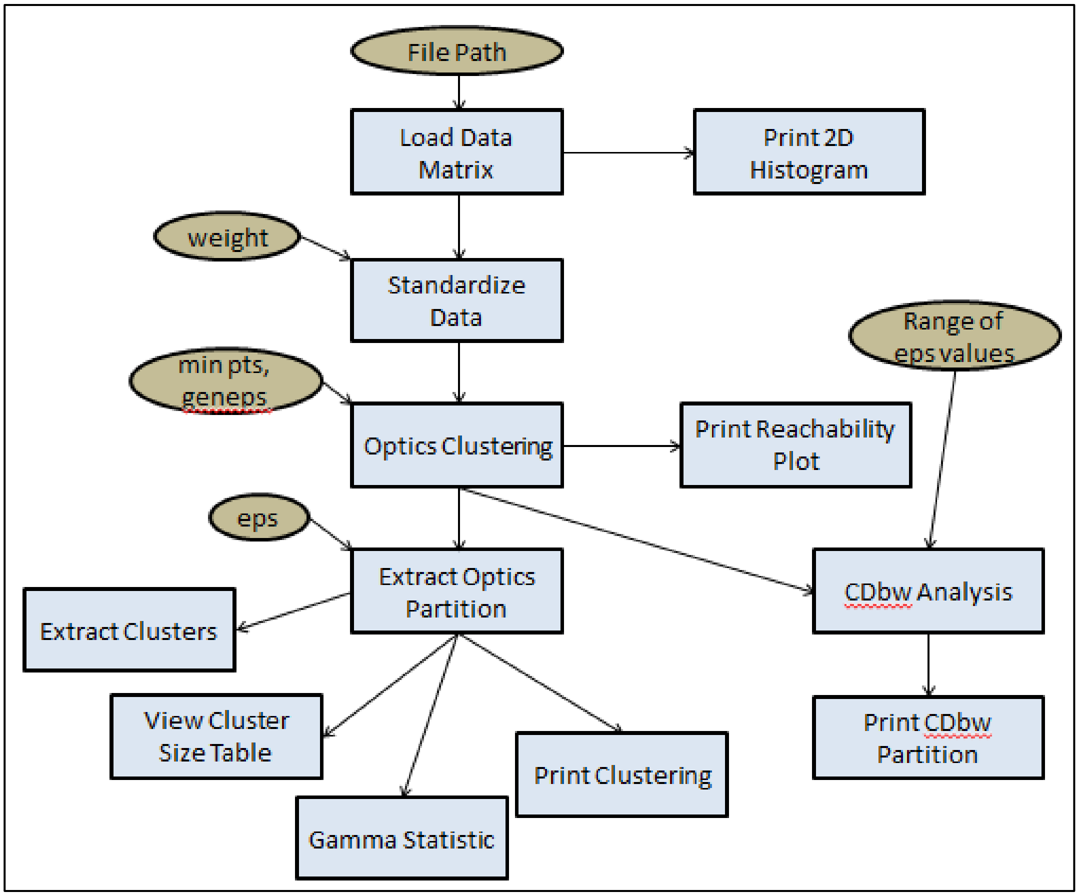

Figure 2. Flow chart showing how data flows through the different operations (blue boxes) as well as key user inputs (tan ellipses). User inputs that do not affect more than one operation are not shown. To avoid errors, operations should only be executed after all "upstream" operations have already been performed. Whenever one of the user inputs shown in the figure is changed, all operations "downstream" of it should be immediately re-run to avoid confusion.

3.2. GUI Functionality

This section briefly describes each of the GUI elements in Figure 1.

3.2.1. Status bar

Displays current status of the program (e.g. "Running File I/O", "Completed File I/O", etc.). Numeric outputs such as the CDbw index value are sometimes displayed in the status bar as well (see descriptions of specific functions below).

3.2.2. File Path text box

A textbox that will display the full file path for the data file being used as input. The file path is only used by the "Load Data Matrix" function (see below).

3.2.3. Exit button

Press this button to close all GUI windows and exit the program.

3.2.4. Inputs Used indicators

All user inputs are attained using dialogue boxes. The purpose of the "Inputs Used" subpanel is to display the values of critical inputs the user has previously entered for later reference. Each input box begins blank and updates when the appropriate input has been requested and received via a dialogue box.

3.2.4.1. min pts

The minimum number of points required to be inside an epsilon-neighborhood for it to be considered a core point (called k in Wu et al.). Recommended value is 25 and is set when "Optics Clustering" is run.

3.2.4.2. eps

The value of epsilon used when a specific clustering solution (partition) is extracted from the OPTICS reachability plot (called ε' in Wu et al.). Must be smaller than geneps. The appropriate value of this parameter will be different for each new data set considered, but values in the range 0.5 – 0.225 have tended to work best for single molecule transport data. Is set when "Extract Optics Partition" is run.

3.2.4.3. geneps

The value of epsilon used when the OPTICS algorithm constructs the reachability plot (called ε in Wu et al). Ideally, this should be high enough so that every point is considered a core point. Recommended value is 1 and is set when "Optics Clustering" is run.

3.2.4.4. weight

The scaling weight which multiplies the conductance data coordinates after standardization (called w in Wu et al.). Recommended value is 1.5 and is set when "Standardize Data" is run.

3.2.5. Operations

The operations subpanel contains buttons to execute the different sections of the software. *These sections cannot be executed in arbitrary order* (see section 3.1 for details).

3.2.5.1. Load Data Matrix

This function prompts the user to enter in a file path, then reads in data from that specified file and displays the file path used in the file path text box (see above). The input file should contain the data from a 2D distance-conductance histogram, formatted into three space-delimited columns: distance, $\log_{10}$(conductance), and bin count. Zero-count bins need not be included but the code will work either way. See sample data files for examples.

3.2.5.1.1. Preparing Input Files from Raw Data

Raw transport data will likely be in the form of hundreds to thousands of “traces” each containing a series of (distance, conductance) points. To transform these data into a viable input file, all data points must first be binned in both the distance and conductance dimension. Each row in the input file should then be in the form: [discrete distance value] [discrete log$_{10}$(conductance) value] [# of data points rounded to that discrete pair of values]. No header is expected in the input file.

3.2.5.2. Standardize Data

Transforms, standardizes, and weights the histogram data as appropriate. Running this function will prompt the user for a value for the scaling factor, weight (see above).

3.2.5.3. Optics Clustering

Runs the OPTICS algorithm on the standardized data set to produce a cluster ordering of the points. A “wait bar” is created to display the progress of this function, which may be time-consuming for large data sets. When this function finishes, the cluster ordering will be displayed via a reachability plot in the top left quadrant of the main GUI window. Running this function will prompt the user for values for the minimum number of points in a neighborhood, min pts, and the generating value of epsilon, geneps (see above).

3.2.5.4. Extract Optics Partition

This function prompts the user to enter a value of epsilon (eps; see above), and then uses that value and the reachability plot created by the OPTICS algorithm to extract a single clustering solution by assigning each data point to a particular cluster ID number. Note that the ID# -1 is always reserved for the noise cluster and ID#0 is never used. Additionally, any clusters originally representing <2% of the total number of data points and any clusters consisting entirely of points with counts <6 have all of their data points reassigned to the noise cluster. When finished, the extracted clustering solution (partition) will be displayed in the top right quadrant of the main GUI window in the form of a 2D distance-conductance plot with points color-coded by cluster ID#. Additionally, the CDbw value associated with the clustering solution will be displayed in the status bar.

3.2.5.5. View Cluster Size Table

Displays a table in the bottom right quadrant of the main GUI window that lists the ID# of each cluster, the number of data points assigned to each cluster, and the percentage of the total number of data points that each cluster accounts for. Note that the ID# -1 is always reserved for the noise cluster, ID#0 is never used, and any clusters originally representing <2% of the total number of data points or consisting entirely of points with counts <6 have all of their data points reassigned to the noise cluster.

3.2.5.6. Extract Clusters

Running this function will prompt the user to list the cluster ID#s of the subset of clusters which they wish to have plotted (use of View Cluster Size Table to help choose clusters is encouraged). A new window will then be generated to display the extracted clustering solution (in the form of a 2D distance-conductance plot with points color-coded by cluster ID#) with only data points from the specified clusters shown. In addition, a second new window will be created to display a reachability plot in which all data points not assigned to the specified clusters have been omitted.

3.2.5.7. CDbw Analysis

Running this function will prompt the user for a range of epsilon (eps) values by specifying a starting value, ending value, and the number of steps to take between. In addition, the user is prompted for a "comparison" epsilon value. First, for each epsilon value in the specified range only, the function will find the clustering solution (partition) using that value and calculate the CDbw index value for said solution. The clustering solution with the highest CDbw value will be displayed in a new window, and the epsilon value leading to this optimal clustering solution will be stated in the status bar. In addition, a second new window will be created to display a plot of the CDbw value for each clustering partition from the specified epsilon range. All of these partitions, not just the optimal one, have been stored by the program and can be viewed using the "Print CDbw Partition" function (see below). Finally, the function will also calculate the CDbw value for the clustering solution found using the comparison epsilon value. A dialogue box will then appear stating how many of the CDbw values calculated for the range of epsilon value are greater than or equal to the CDbw value for the comparison epsilon value. This information can be used to calculate the statistical significance of the CDbw value for the comparison epsilon in regards to the range of epsilon values tried.

3.2.5.8. Gamma Statistic

This function calculates the value of Hubert's Gamma Statistic for the clustering solution displayed in the main GUI window, and displays this value in the status bar. In addition, the function uses Monte Carlo methods to simulate a sampling distribution for the Gamma Statistic by permuting the cluster membership randomly while preserving the number of clusters and the size of each cluster. Upon running, the function prompts the user to specify the number of such permutations to include in the sampling distribution. The recommended value is 100, but note that this will require several minutes to run on most computers and may slow down the computer significantly while running (recommended to only run on a computer with at least 8GB of available memory). A "wait bar" indicates the progress of the Monte Carlo simulation. Upon completion, a new window is created to display the Gamma Statistic sampling distribution. The number of Monte Carlo clustering solutions with Gamma Statistics greater than or equal to the Gamma Statistic of the clustering solution displayed in the main GUI window is also written to the status bar. This information can be used to calculate the statistical significance of the Gamma Statistic for the clustering solution displayed in the main GUI window.

3.2.5.9. Print 2D Histogram
Creates a 2D histogram of the raw distance-conductance data from the input file and displays it in a new window.

3.2.5.10. Print Reachability Plot
Displays the reachability plot derived from the OPTICS algorithm in a new window.

3.2.5.11. Print Clustering
Creates a new window to display the clustering solution corresponding to the epsilon value listed in the "eps" input text box in the main GUI window. The clustering solution is displayed in the form of a 2D distance-conductance plot with points color-coded by cluster ID#.

3.2.5.12. Print CDbw Partition
This function allows the user to select one of the clustering solutions calculated by the "CDbw Analysis" function (see above) by specifying its partition ID# (which runs from 1 to the number of epsilon values in the user-specified range). That clustering solution is then displayed in a new window in the form of a 2D distance-conductance plot with points color-coded by cluster ID#. Both the epsilon value used to extract the specified clustering solution and the CDbw value for that solution are displayed in the status bar.

4. Outputting Data from Plots

To easily output data from the plots created by the software, users should first use one of the functions that create a plot in their own window rather than in a subsection of the main GUI window (see above). Next, using the menu buttons in that separate plot window, the user can save the plot as a MATLAB file. This file can then be opened in MATLAB at any later time to extract the data.

5. Validating the Software

To help confirm that the software is behaving correctly, four sample data files are included along with this software package: Fig5.dat, Fig6.dat, Fig8.dat, and Fig9.dat, which contain the data sets used for the corresponding figures in Wu et. al. In addition, a "solution" to each sample data set has been included in the form of a screen shot of the main GUI window after processing of those data up to the determination of a clustering solution at a particular epsilon value. Within these screen shots can be found all of the parameters used during processing (min pts, eps, geneps, and weight) as well as quantitative outputs such as the CDbw value and exact cluster sizes for the clustering solution. To validate that the software is performing correctly, a user can simply process one or more of the sample data sets and compare their results to the appropriate screen shots.

6. Log of Known Bugs and Fixes

6.1. April 2017

6.1.1. A minor bug in the calculation of distances in the version of the OPTICS clustering software used to produce the figures in Wu et al. was resolved. This bug has been corrected in the published software version associated with this manual, and we have confirmed that the removal of this bug does not affect any of the paper's conclusions nor does it qualitatively change any of the reported clustering solutions. However, removing the bug does lead to minor quantitative differences in extraction epsilon values and final cluster sizes, so users should not expect to be able to precisely reproduce the figures in Wu et al. Instead, the sample data sets and their solutions (see above) should be used as a direct point of comparison since these solutions were produced using the fixed version of the software (i.e. the version included with this manual).

6.1.2. There is a typo in Wu et al. that reports the cluster size cut-off as 1%; 2% is the correct value used in Wu et al. and implemented in the code.

7. Version Log

7.1. Version 1.0

Original version, released April 24, 2017.

7.2. Version 1.1

Released May 5, 2017. Changes include: load data matrix function was modified to automatically remove zero-count bins; wait bar was added to optics clustering function; and the memory requirement for calculating the CDbw index was reduced.