AutoSOME User Manual

December 03, 2009

Written by Aaron M. Newman

Biomolecular Science and Engineering Program
University of California, Santa Barbara

Table of Contents

Introduction

AutoSOME is a powerful new unsupervised clustering method that identifies clusters of diverse geometries from potentially large, multi-dimensional, datasets without prior knowledge of cluster number or structure. In addition, fuzzy relationships among data points in complex, noisy datasets like microarrays can also be detected by AutoSOME and usefully visualized as two-dimensional fuzzy cluster networks. We have implemented the AutoSOME method as a Graphical User Interface (GUI) and command line tool to facilitate its use by the academic research community. Below, instructions are provided for using both versions of AutoSOME, and a protocol is given for generating fuzzy cluster networks in Cytoscape [Shannon et al., 2005]. Refer to the AutoSOME website for tutorials and FAQ [http://jimcooperlab.mcdb.ucsb.edu/autosome].

System Requirements

AutoSOME is coded in Java to promote platform independence. To launch the GUI from the AutoSOME website [http://jimcooperlab.mcdb.ucsb.edu/autosome], Java Web Start technology needs to be installed on your computer along with Java Standard Edition 1.6, both of which are freely available at http://java.sun.com. The command line version of AutoSOME will run in JAVA 1.5 and perhaps earlier versions. In general, the more memory and processors available, the better the performance of AutoSOME. Without user intervention, both the GUI and command line versions will greedily use all available CPUs. For 32-bit Windows systems, the maximum amount of memory that can be allocated to AutoSOME is about 1.6 GB, while 64-bit operating systems with 64-bit Java can allocate up to ~30 GB of memory. Microarray datasets like the HG U133 Plus 2.0 Affymetrix chipset (>54k probes) with dozens of samples can be run with 1.6 GB RAM.

The ‘Java Web Start’ launch button on the ‘AutoSOME Web Portal’ homepage will automatically allocate up to 1 GB of RAM. To run AutoSOME with more or less memory, download the .jar file from:
http://jimcooperlab.mcdb.ucsb.edu/autosome/downloads.

Installing and Running the AutoSOME GUI

Download AutoSOME from the website, and then unzip all contents of the downloaded archive into the same directory. Using your system console, navigate to the directory where you installed AutoSOME, and type:

```
java –Xmx1600m –Xms1600m –jar autosome_vXXXXXX.jar
```

XXXXXX = the GUI version you downloaded (each version is named by the date of compilation, e.g. 120109). The -Xmx,-Xms arguments allocate additional memory to the

Java Virtual Machine necessary for running large datasets. Allocate more or less memory as needed for your input dataset, parameters, and machine architecture.

For instructions on how to use the AutoSOME command line interface, see *AutoSOME Command Line Version* below.

Using AutoSOME

Input Format

AutoSOME accepts three kinds of numerical input files.

Standard format: The input file is a table of numerical values with optional column labels (row 1) and mandatory row labels (column 1). If column labels are specified, column 1 also needs a label in row 1. All entries must be tab, comma, or space delimited. Input data can be easily formatted using Microsoft Excel. See Table 1 below for an example.

Table 1

Probe	hESC-1	hESC-2	hESC-3	hESC-4	iPSC-1	iPSC-2	iPSC-3
212853_at	8.221449	8.297634	7.694108	8.215596	10.1284	10.25488	10.21546
212854_x_at	8.523748	8.706556	8.044123	8.992252	8.87927	8.974617	9.083473
212855_at	10.64296	10.4093	10.60148	11.03563	12.08599	11.91321	12.04592
212856_at	8.936721	9.218977	9.109346	8.425392	8.431269	8.418733	8.450901
212857_x_at	6.195244	5.929077	4.432911	6.764105	4.344787	4.786857	4.953329
212858_at	7.795604	7.565154	7.744513	7.58388	8.657005	8.613815	8.781187
212859_x_at	10.26586	10.44094	9.274517	9.467813	9.218634	9.221389	9.472309
212860_at	6.178299	6.260259	6.218979	6.416278	6.758869	6.275902	6.270324
212861_at	14.36244	14.37399	14.52238	14.38661	14.50383	14.52275	14.45975

Microarray formats:

1) **PCL** = Pre CLuster format used for the Cluster software [Eisen et al. (1998) PNAS 95:14863].
   For an example, see http://puma.princeton.edu/help/formats.shtml#pcl or the Cluster/TreeView User Manual. The first two columns are reserved for gene annotation (column 1 = row identifiers, column 2 = gene names or annotation). Column 3 is optional, is called GWEIGHT, and specifies how to weigh each gene when computing gene-gene similarity. This column is read by AutoSOME, but is ignored since AutoSOME does not cluster genes using a similarity matrix. Row 1 is mandatory and is used to provide column names including names of each array (e.g. ID, NAME, GWEIGHT, array1, array2, …). Row 2 is optional, is called EWEIGHT, and specifies how each array is weighed when computing array-array

similarity. In contrast to GWEIGHT, AutoSOME will use EWEIGHT when constructing a distance matrix for transcriptome clustering (fuzzy cluster networks option).

2) **Gene Expression Omnibus (GEO) Series Matrix**. This format contains all chipset expression data as well as user-supplied annotation in a spreadsheet style. AutoSOME can automatically extract expression information content and column names from a raw series matrix file. This allows for rapid analysis of any GEO dataset by simply downloading the archive containing the series matrix text file, unzipping it, and loading it into AutoSOME.

Missing Values

There are two strategies implemented in AutoSOME for handling missing values (blank space, '?', or 'NA').

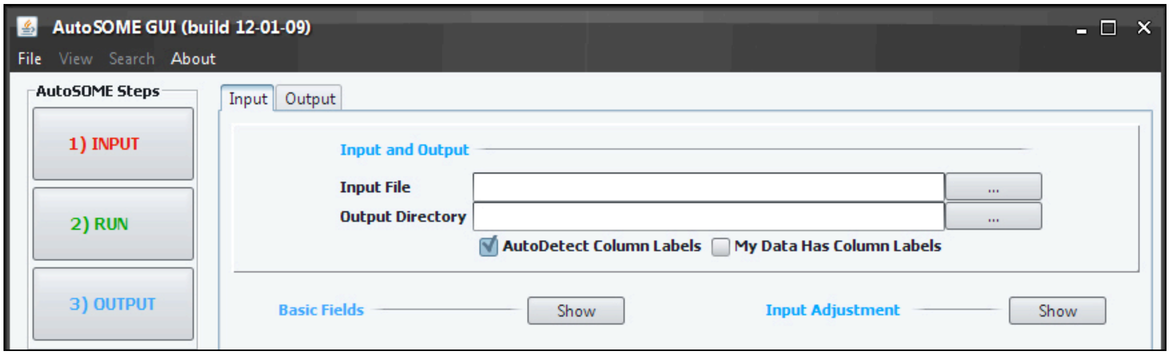
1) Replace missing values with median of columns (or rows).
2) Replace missing values with mean of columns (or rows).

Go to File>Settings to launch a window that will allow you to specify which of these two techniques to use.

AutoSOME GUI

The AutoSOME GUI is launched using Java Web Start from your browser or via the command line, and is used to run AutoSOME as well as to browse and generate publication quality visualizations of the cluster output (see Figure 1).

Figure 1

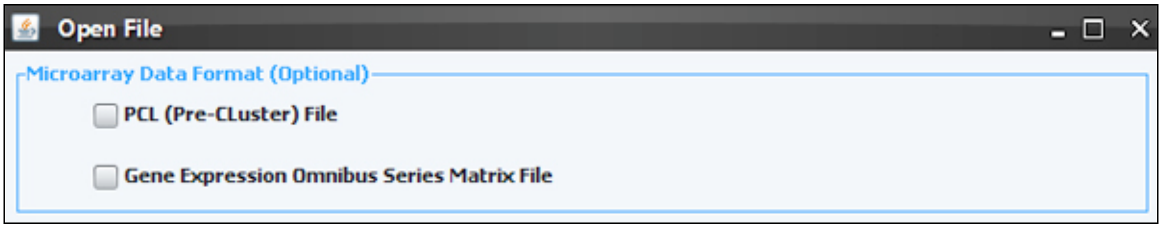


☑ AutoDetect Column Labels ☐ My Data Has Column Labels

Load Input

To begin, launch a file browser by either pressing the large '**INPUT**' button in the '**AutoSOME Steps**' panel on the top left, or by clicking the browse button adjacent to the input file text box in the '**Input and Output**' section. A file browser will appear. For standard input (see Input Format above), simply find your file and open it. For microarray input, optional data formats are shown at the top of the file browser (see Figure 2). Select the appropriate file format for your microarray dataset and load the file.

Figure 2



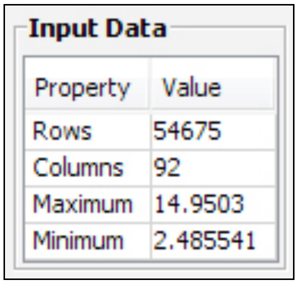
Microarray Data Format (Optional)

☐ PCL (Pre-CLuster) File

☐ Gene Expression Omnibus Series Matrix File

The GUI will attempt to automatically detect the presence of column labels. If the input has column labels containing numerical data, select the '**My Data Has Column Labels**' checkbox. Once loaded, the output directory is automatically shown, and can be changed by clicking the browse button next to the output directory text box. The GUI will deploy an error message if the input file is incorrectly formatted.

Figure 3



Input Data

Property	Value
Rows	54675
Columns	92
Maximum	14.9503
Minimum	2.485541

After loading, the number of rows and columns will be displayed in the '**Input Data**' panel on the left side of the GUI (see Figure 3). The maximum and minimum values over the entire dataset are also provided. When finished clustering, this table will dynamically update based on the contents of selected clusters in the cluster tree (see Output below).

Input Adjustment

AutoSOME implements several input normalization methods in addition to log2 scaling. For an excellent overview of these techniques, see the manual to the Cluster software available at http://rana.lbl.gov/manuals/ClusterTreeView.pdf [Eisen et al., 1998]. Press the '**Show**' button to the right of the '**Input Adjustment**' label to access input scaling and normalization options (see Figure 4 below). All input adjustment operations are performed in the order listed in the GUI, from top to bottom. In brief:

i) Log$_2$ Scaling: Logarithmic scaling is routinely used for microarray datasets to amplify small fold changes in gene expression, and is completely reversible. We recommend applying log2 scaling in cases where expression values span several orders of magnitude. All other implemented input adjustment methods irreversibly change the input to make it more suitable for analysis.

ii) Unit variance: forces all columns to have zero mean and a standard deviation of one, and is commonly used when there is no *a priori* reason to treat any column differently from any other.

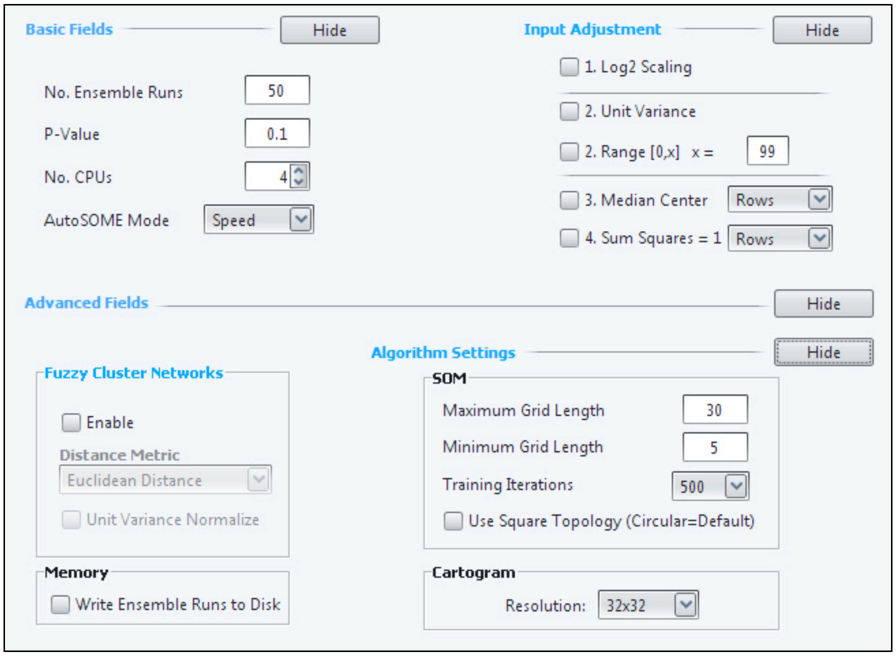
iii) Range [0,x]: Alternatively, data in all columns can be normalized to share lowest and highest values (0,*x*) by specifying an upper bound *x*.

iv) Median Center Rows/Arrays: For microarray analysis, centering each gene (row) and/or array (column) by subtracting the median value of the row/column eliminates amplitude shifts to highlight the most prominent patterns in the expression dataset. We recommend at least median centering of genes (and arrays for additional smoothing) for co-expression clustering.

v) Sum of Squares=1 Rows/Arrays: This normalization procedure smoothes the dataset by forcing the sum of squares of all expression values to equal 1 for each row/column in the dataset. The impact of this method on cluster identification can be significant and tends to result in the detection of larger clusters trailing off into genes with minimal differential expression (after clustering, the signal to noise ratio can be boosted by using the confidence filter; see Output below)

As an example, to run AutoSOME co-expression analysis, one may apply log2 scaling, unit variance normalization, and median-centering of genes (and/or arrays). To run AutoSOME transcriptome clustering (fuzzy cluster networks, see tutorial below), it is generally recommended to apply unit variance normalization to your dataset. Other normalization settings may also be desirable. Input can also be adjusted using Microsoft Excel or the Cluster software [Eisen et al., 1998] and then imported into AutoSOME.

Figure 4



No. Ensemble Runs 50
P-Value 0.1
No. CPUs 4
AutoSOME Mode Speed

☐ 1. Log2 Scaling
☐ 2. Unit Variance
☐ 2. Range [0,x] x = 99
☐ 3. Median Center Rows
☐ 4. Sum Squares = 1 Rows

☐ Enable
Distance Metric Euclidean Distance
☐ Unit Variance Normalize

Maximum Grid Length 30
Minimum Grid Length 5
Training Iterations 500
☐ Use Square Topology (Circular=Default)

☐ Write Ensemble Runs to Disk

Resolution: 32x32

Basic Fields

Press the '**Show**' button to expand the '**Basic Fields**' section (see Figure 4).

1. i) Ensemble Runs: The default of 50 ensemble iterations should be sufficient to begin investigating the cluster structure of most datasets. Although in practice, AutoSOME clustering results can be quite stable with ~50 ensemble iterations (and even as little as 20), for final clustering results, it is recommended to increase this number to at least 100.
2. ii) P-Value: AutoSOME has been extensively benchmarked on a highly diverse array of clustering problems using a P-value cutoff of 0.1. Reduce the p-value threshold to identify tighter clusters.
3. iii) No. CPUs: To liberate processor resources, decrease 'No. CPUs'. AutoSOME running time will decrease approximately linearly with respect to increasing number of dedicated CPUs.
4. iv) AutoSOME Mode: This parameter alters advanced AutoSOME algorithm settings to switch between 'Precision' and 'Speed' modes of operation. 'Precision' takes longer, but provides greater training of the SOM node lattice (2X1000 iterations) and greater resolution for density-equalization of the SOM error surface (64X64). For enhanced performance, and especially for first-pass exploratory cluster analysis, choose 'Speed' for less SOM iterations (2X500) and less resolution for density-equalization (32X32). In our experiments, 'Speed' works quite well, and in fact, often yields comparable results to 'Precision'. In addition, 'Speed' is roughly 4 times faster.

Advanced Fields

Fuzzy Cluster Networks. To cluster columns (e.g. individual microarrays), select the 'Enable' checkbox (see Figure 3). Then, pick the distance metric for calculating the input distance matrix. Euclidean distance is selected by default due to empirical testing. See the review by D'haeseleer (2005) for descriptions of Euclidean, Pearson's and Uncentered correlation distance metrics. In brief, Euclidean distance is sensitive to amplitude shifts while correlation is not. Experiment with these metrics to get a feel for how they work. If smoothing out the distance matrix is desired, 'Unit Variance Normalize' can be selected to normalize the distance matrix columns to unit variance. When finished clustering, AutoSOME will automatically generate additional output files for building fuzzy cluster networks in Cytoscape (see Fuzzy Cluster Networks below).

Memory. Write Intermediate Runs to Disk: To decrease RAM consumption, the results from individual ensemble iterations can be written to disk instead of memory. This may be necessary on systems with insufficient RAM to process large datasets over many ensemble iterations (e.g. 45k probes, 100 samples, and 100 ensemble iterations), however, overall running time will be slower due to reading and writing to disk. If AutoSOME crashes (or seems to hang for a long period of time), then 'Write Ensemble Runs to Disk' should be selected (see Figure 3) as an out of memory error is likely. All intermediate files will be written to a temporary folder in the output directory.

Algorithm Settings. Press the '**Show**' button to expand the '**Algorithm Settings**' section (see Figure 3). The GUI permits modification of the most critical algorithm parameters. For access to all algorithm parameters, use the command line version of AutoSOME.

SOM.

i) Maximum Grid Length: The maximum number of nodes for the x/y length of the SOM node lattice.

ii) Minimum Grid Length: The minimum number of nodes for the x/y length of the SOM node grid.

In general increase 'Maximum Grid Length' to provide greater resolution for cluster separation in the SOM, and to increase the number of possible clusters. Increase 'Minimum Grid Length' to force AutoSOME to use an SOM lattice with specific minimum dimensions. Set both parameters to the same value to disable automatic adjustment of grid length by the algorithm. The SOM grid length is set, by default, between 5 and 30, providing sufficient cluster resolving power for most clustering problems. *If a dataset with at least 5,000 rows is imported (e.g. whole-genome microarray), the maximum grid length will be automatically set to 20 for more efficient running time.*

iii) Training Iterations: The number of iterations for training the SOM node lattice. The value of this parameter dictates the number of iterations for each of two phases, coarse-grained followed by fine-grained training. This value can be toggled between '1000' iterations for accuracy and '500' iterations for speed.

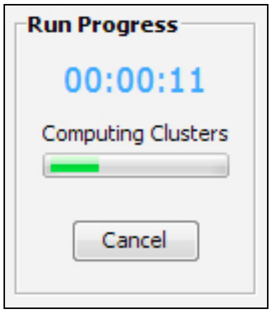
iv) Use Square Topology: Select this checkbox to use a square-shaped node lattice rather than a circular node lattice (default). Benchmarking results indicate both topologies yield comparable results. See the Manuscript for more details.

Cartogram. Resolution: The density-equalizing cartogram algorithm requires an input array with dimensions that are a power of two. Although all benchmarking tests were conducted using an array size of 64X64, greater resolutions may yield more accurate density-equalization (especially when the SOM dimensions are large). On the other hand, going from 64X64 to 32X32 may increase running time dramatically (up to ~4X), and is sufficient for accurate clustering results in many cases (especially when the SOM dimensions are less than 30X30).

Running AutoSOME

After the input file and parameters are specified, execute AutoSOME clustering by pressing the green '**RUN**' button in the '**AutoSOME Steps**' panel (see Figure 1). Progress is shown in the lower left corner of the GUI and elapsed time is displayed (see Figure 5). When finished, the GUI will automatically be redirected to the output panel for browsing the cluster output. The input and output panels can be toggled back and forth using the tabs or the '**INPUT**' and '**OUTPUT**' buttons in the control panel.

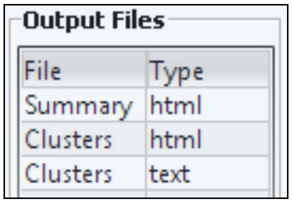
Figure 5



Output

Output Files. The 'Output Files' table, located on the center-left side of the GUI, will show links to the AutoSOME output files after clustering is finished (see Figure 6). Simply select a file from the table and the file will be automatically displayed. For details of the output files generated by AutoSOME, see *files written to disk* below.

Figure 6

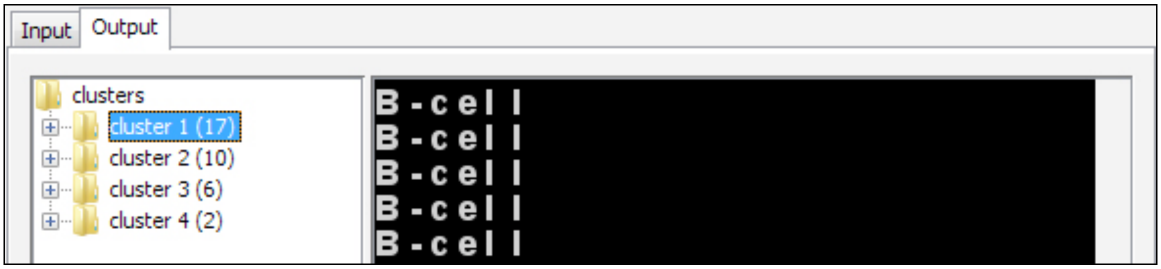


Output Files

File	Type
Summary	html
Clusters	html
Clusters	text

Cluster tree. As shown in Figure 7, all clusters are listed using a dynamic tree structure and are ordered from top to bottom by decreasing size.

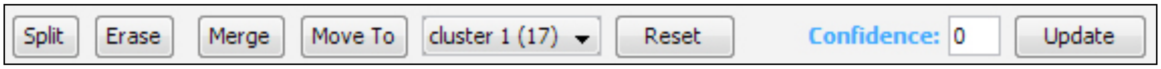
Figure 7



```
B-cell
B-cell
B-cell
B-cell
B-cell
```

Once a cluster node is selected using your mouse, the cluster tree can be rapidly traversed by pressing the '↑' and '↓' keys. To select more than one cluster, hold down the 'Shift' or 'Control' key and select using your mouse. Clusters can be manually modified using the control panel below the cluster tree (see Figure 8). Use 'Split' to make a new cluster from a selected group of data points (first expand a cluster node by double-clicking, then select data points with mouse). Use 'Erase' (or press Delete key) to erase entire clusters or specific data points. Press 'Merge' to combine the contents of all selected clusters into a single cluster. Use 'Move To' for moving selected data points from one cluster to another cluster. Press 'Reset' to return to the original clustering output. To filter clustering results by cluster confidence (metric ranging from 1-100, where 100=data point always in cluster x) type in confidence threshold ≤100 in the '**Confidence**' text box and press 'Update'.

Figure 8



Confidence: 0

By default, the contents of the selected cluster(s) will be displayed as a list of cluster labels. For additional viewing options, see below.

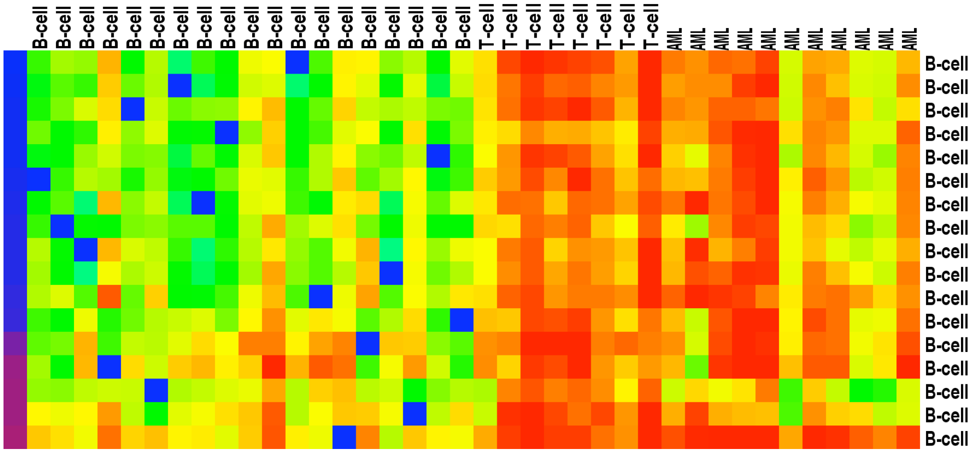
Heatmaps. Several heatmap visualizations are available. Go to 'View' in the menu bar, and select 'green red', 'rainbow', 'gray scale', or 'blue white' from the 'heatmap' submenu. Table 2 summarizes the key attributes of the available heatmap color modes.

Table 2

Heatmap Mode	High Value (R,G,B)	Low Value (R,G,B)	Middle Value(s)
green red	Green (0,255,0)	Red (255,0,0)	Black
rainbow	Red (255,0,0)	Blue (0,0,255)	Blue, Light Blue, Green, Yellow, Red (evenly spaced)
gray scale	Black (0,0,0)	White (255,255,255)	shades of Gray
blue white	Blue (0,0,255)	White (255,255,255)	shades of Blue

The mouse scrollbar is used to zoom in or out. By default, cluster confidence is shown as a vertical bar to the left of the heatmap with blue=high and red=low confidence (see rainbow-colored heatmap in Figure 9 below). For fine-grained control over heatmap visualization parameters, select '**pixel settings**'.

Figure 9



Heatmap>pixel settings. The pixel settings window provides several options for customizing the heatmap display (see Figure 10).

i) <u>Row height:</u> Determines the vertical resolution of each heatmap pixel and can be used to vertically compress the heatmap image.

ii) <u>Contrast</u>: Adjusts the maximum and minimum values used for heatmap display. Contrast C increases the original maximum value M and decreases the original minimum value m by $(C\text{-}1)*(1+M\text{-}m)/2$.

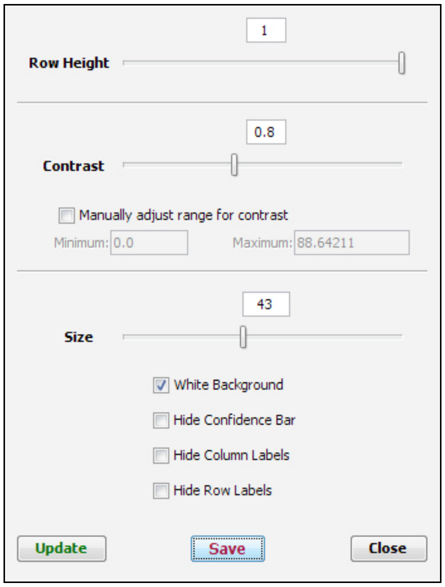
   a. <u>Manually adjust range for contrast</u>: Select this option to manually input maximum and minimum values for contrast adjustment. By default, the maximum and minimum values of the entire input dataset are used. To see the maximum and minimum values for a particular cluster (or set of clusters), check the 'Input Data' table on the left panel of the GUI.

iii) <u>Size</u>: This slider controls the size scale of each heatmap pixel and can be used for saving high-resolution images.

iv) <u>Checkboxes</u>: Self-explanatory.

To update the heatmap image after changing parameters, press 'Update'. Click 'Save' to launch a file browser for saving your heatmap output. All images are saved in the Portable Network Graphics (PNG) format.

Figure 10



Row Height 1

Contrast 0.8

☐ Manually adjust range for contrast

Minimum: 0.0 Maximum: 88.64211

Size 43

☑ White Background

☐ Hide Confidence Bar

☐ Hide Column Labels

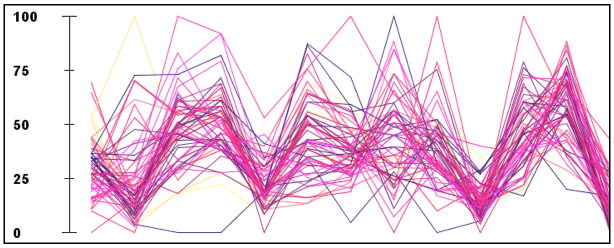
☐ Hide Row Labels

Signal plots. As indicated in Figure 11, signal plots display all numerical vectors in the selected cluster(s) as a line graph across all column categories (x-axis). Go to ‘View’ in the menu bar, and select ‘rainbow’ or ‘red’ from the ‘signal plot’ submenu.

i) Scale bar: By default, a scale bar is shown on the y-axis. Deselect the ‘scale bar’ checkbox in View>signal plot to remove. To increase or decrease the scale-bar resolution, use the up and down arrow keys in the number pad (8 and 2), respectively. To change the decimal precision of each number, use the left and right arrow keys in the number pad (4 and 6).

ii) Mean signal: Select the ‘mean signal’ checkbox in View>signal plot. Collapses signal plot into one line representing the mean of all vectors in the selected cluster(s).

Figure 11



View>Settings><u>scale using entire dataset</u>: When selected (go to View>settings), the contrast setting in the heatmap, and range of the y-axis for signal plots, are scaled using the maximum and minimum values of the dataset. When deselected, maximum and minimum values are set based on the selected cluster(s). Deselecting is useful for amplifying cluster-specific signals that are otherwise washed out in the context of the entire input dataset.

Search. To find a particular data item in the cluster output, go to 'Search>Find'. This will invoke a search window. Either find your data point from the 'Choose Identifier' list of manually enter your data item identifier, then click 'Submit'. If it is found, your data item will be highlighted in yellow using the heatmap display (you might need to scroll down to find it). Please make sure your data items are uniquely labeled for this feature to work properly.

Saving data.

i) <u>Images</u>: All images can be saved by selecting 'File>Export>save image' from the menu.

ii) <u>Tabular Data</u>: To export tabular data from the selected cluster(s), go to 'File>Export>save tabular data'. This option will save all cluster contents, along with cluster labels and cluster confidence for each data point. The output format is identical to output file 3 (see *Files written to disk* below).

Files written to disk. AutoSOME writes the following files to your hard drive:

1) AutoSOME_inputName_E**x**_Pval#_Summary.html = summary table of file name, parameters used, and cluster output table with size of each cluster, mean cluster confidence, and hyperlinks to data content of each cluster.
2) AutoSOME_inputName_E**x**_Pval#.html = all clusters and data contents, including individual cluster confidence for each data point.
3) AutoSOME_inputName_E**x**_Pval#.txt = text file with same data as file (2). First row = column labels (if provided as input), first column 'CLUST'=cluster label, second column 'CONF'=cluster confidence, third column 'NAME' =data point label, all other columns=data vectors.

If your dataset was normalized using AutoSOME, a code is stored using row 1 of output file (3) in order to properly reopen the file (*see Opening old clustering results* below). The code legend: #=input dataset has column labels, n=log2 scaling, u=unit variance, sX=normalized from [0,X], m=median centering of rows, M=median centering of columns, q=sum of squares normalization of rows, Q=sum of squares normalization of columns.

E**x** = **x** ensemble iterations (e.g. E100)

Pval# = p-value cutoff (e.g. Pval0.1)

If '**Fuzzy Cluster Networks**' is enabled (see Figure 3), three additional files are written to disk:

4) AutoSOME_inputName_E**x**_Pval#_Edges.txt = fuzzy edges among data points: first two columns denote connected nodes, third column = pairwise affinity or the fraction of times the two data points were co-clustered (pairwise affinity ranges from -.5, never co-clustered, to 0.5, always co-clustered).
5) AutoSOME_inputName_E**x**_Pval#_Nodes.txt = All clustered data points: first column= unique data point label, second column=cluster label, third column=original data point label
6) AutoSOME_inputName_E**x**_Pval#_Matrix.txt = pairwise affinity matrix of all data points compared to all data points; essentially the same information content of output file (4) presented in a form suitable for a heatmap display. This file can be immediately read as input into the Cluster 3.0 software [Eisen et al., 1998] to hierarchically reorder the matrix. Results can be visualized with Java TreeView [Saldanha, 2004].

Since Fuzzy Cluster Networks are computed from a distance matrix of all vertical data vectors (e.g. cell samples), output files (1)-(3) and (6) contain data points represented as a distance matrix (not the original input data).

Opening old clustering results. To browse previous clustering results, select 'Open AutoSOME Results' from the 'File' menu. Use the file browser to find output file (3), the text file, and press 'Open'. The GUI will be immediately redirected to the 'output' window and display the cluster tree. All original data normalization settings are automatically restored for display in the output window.

Reset. To reset all AutoSOME settings, go to File>Reset.

Enhanced GUI vs. Basic GUI. The enhanced and basic GUIs offer identical functionality. The look and feel of the enhanced GUI makes use of the Substance freeware package (https://substance.dev.java.net/) while the basic GUI uses the default system look and feel.

AutoSOME Command Line Version

>Usage:
java -jar autosome_vXXXXXX.jar [Input] [Options]

>maximum JVM memory recommended, e.g.
java -Xmx1600m -Xms1600m -jar autosome_vXXXXXX.jar [Input] [Options]

where XXXXXX=release date of AutoSOME (e.g. 120109)

To display all input parameters, run AutoSOME with the parameter –o (letter 'o' for options).

As indicated below, the command line version of AutoSOME has many options not available in the GUI, including the option to use alternative clustering algorithms (i.e. K-Means, Hierarchical Clustering with four linkage types) and alternative dimensional reduction techniques (i.e. density-equalized SOM, normal SOM, and Sammon's Mapping). For alternative clustering methods, the number of input clusters needs to be specified using the –k parameter (see below). In addition, all of these clustering methods, including AutoSOME, can be benchmarked with the –b option as long as the data item labels in the input file correspond to numerical cluster labels starting with 1 (1,2,…,No. clusters *n*).

Examples using fictitious gene expression dataset 'yeast.txt'

1) Perform co-expression clustering using log2 scaling, unit variance and median-centering of rows normalization, 100 ensemble runs, p-value threshold of 0.05, launch GUI to display results, and write files to "C:\output".

```
java -Xmx1600m -Xms1600m -jar autosome_v120109.jar yeast.txt –N –j1 –e100 –p.05 –v –DC:\output
```

2) Perform transcriptome clustering using unit variance, Uncentered correlation, 100 ensemble runs, launch GUI to display results, and write files to "C:\output".

```
java -Xmx1600m -Xms1600m -jar autosome_v120109.jar yeast.txt –n1 –Q3 –e100 –v –DC:\output
```

3) Perform transcriptome clustering using unit variance, Pearson's correlation to build the distance matrix, K-means with 5 clusters, 20 ensemble iterations, and launch GUI to display results.

```
java -Xmx1600m -Xms1600m -jar autosome_v120109.jar yeast.txt –n1 –Q2 –K –k5 –e20 –v
```

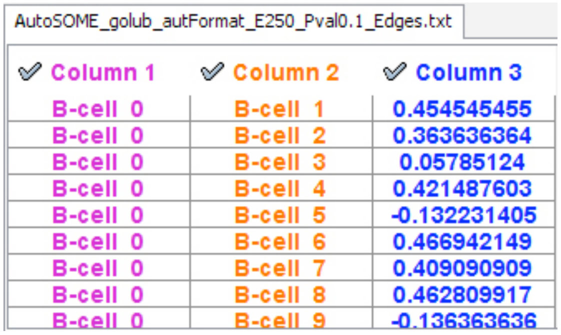
All Command Line Parameters

Parameter	Description (Default)
-t[integer]	set number of threads (available CPUs)
-e[integer]	set number of runs to merge into ensemble (10)
-p[0-1]	set p-value threshold for minimum spanning tree clustering (0.1)
-D[directory]	set output directory (same directory as input file)
-C	read in column headers from first row of input file (auto-detect otherwise)
-v	launch cluster viewer (false)
-v2	display previous clustering results: input=clustering output text file (false)
-n[integer]	normalize input by unit variance '-n1', to log2 '-n0' or into range [0,X] '-nX' (false)
-j[1,2,3]	1=perform median center normalization on all rows;2=columns;3=rows and columns
-u[1,2,3]	1=perform sum of squares=1 normalization on all rows;2=columns;3=rows and columns
-N	normalize input to log2 with unit variance (false)
-#	apply unit variance normalization to distance matrix (false)
-w	read in PCL-formatted input file (false)
-W	read in Gene Expression Omnibus Series Matrix-formatted input file (false)
-Q	transform columns from input into Euclidean distance matrix (false)
-Q[2,3]	distance matrix metric; 2 = Pearson's, 3 = Uncentered Correlation (Euclidean)
-h[1,2,3,4]	fill missing values; 1=means of rows;2=medians of rows;3=means of columns;4=medians of columns (means of rows)
-b	do benchmarking: 'F-measure, Precision, Recall, NMI, corrected Rand Index',[data items must be labeled: 1,2,3,...,total number of clusters] (false)
-c[integer]	set number of Monte Carlo simulations for MST clustering (10)
-g[integer]	set x of SOM grid xy, where y=x (square root of input size*2)
-M[integer]	set maximum x/y grid of SOM (30)
-m[integer]	set minimum x/y grid of SOM (5)
-P	set SOM distance metric to Pearson Correlation (Euclidean)
-P2	set SOM distance metric to Uncentered Correlation (Euclidean)
-s	set SOM topology to square (circle)
-i[integer]	set number of SOM iterations (1000)
-x[integer]	set SOM error surface exponent (3)
-r[power of 2]	set density-equalizing cartogram resolution (64)
-E	disable Density-Equalizing Cartogram (false)
-S	invoke Sammon Mapping instead of SOM (false)
-S[integer]	set number of Sammon Mapping iterations (100)
-k[integer]	specify number of clusters in dataset (false)
-K	invoke K-Means Clustering; requires option -k (false)
-A	invoke Agglomerative Clustering; requires option -k (false)
-A[method]	1=Single, 2=Complete, 3=Average, 4=Ward's (4)
-V	print verbose output (false)

Creating Fuzzy Cluster Networks

Fuzzy cluster networks highlight the fuzzy relationships among clustered data points using an intuitive two-dimensional network display. A powerful application of this approach is the visualization of differences between cell lines on the basis of differential gene expression. To display fuzzy cluster networks, the network visualization tool Cytoscape needs to be installed on your computer [Shannon et al., 2005]. Cytoscape is freely available from http://www.cytoscape.org/.

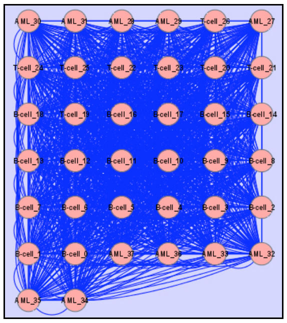
1) Import your input file into AutoSOME, set fields and adjust input.
2) Select ‘Fuzzy Cluster Networks’ and choose distance metric. (Please remember that only vertical data vectors are clustered (e.g. cell samples or time series of a microarray dataset)
3) Run AutoSOME
4) Launch Cytoscape
5) In Cytoscape (All screenshots below taken from Cytoscape 2.6.0):
    a. Go to File>Import>Network from Table (Text/MS Excel)…
    b. Go to ‘Select File(s)’ and locate AutoSOME output file (4) containing all edges (see ‘**Files written to disk**’ in Output above).
    c. Set ‘Source Interaction’ to ‘Column 1’ and ‘Target Interaction’ to ‘Column 2’. Finally, click on Column 3 in the data Preview window to activate it (it will turn blue).



AutoSOME_golub_autFormat_E250_Pval0.1_Edges.txt

✓ Column 1	✓ Column 2	✓ Column 3
B-cell 0	B-cell 1	0.454545455
B-cell 0	B-cell 2	0.363636364
B-cell 0	B-cell 3	0.05785124
B-cell 0	B-cell 4	0.421487603
B-cell 0	B-cell 5	-0.132231405
B-cell 0	B-cell 6	0.466942149
B-cell 0	B-cell 7	0.409090909
B-cell 0	B-cell 8	0.462809917
B-cell 0	B-cell 9	-0.136363636

    d. Select ‘Import’ and then ‘Close’. A raw network will appear as a grid.



e. Go to File>Import>Attribute from Table (Text/MS Excel)…

f. Go to ‘Select File(s)’ and locate AutoSOME output file (5) containing all nodes and attributes (see ‘**Files written to disk**’ in Output above).

g. Select ‘Import’.

h. Change global properties of the network:

    i. In the Control Panel, select the VizMapper^{TM} tab.

    ii. Click in the ‘Defaults’ window (shows a source and target pair with a blue background). A new window will appear.

    iii. Select the ‘Global’ tab in the bottom right.

    iv. Change the background color to white.

    v. Go back to the ‘Node’ tab and change the NODE_BORDER_COLOR property to black and increase the ‘NODE_LINE_WIDTH’ to 2.

    vi. Select ‘Apply’

i. Using the VizMapper^{TM} tab:

j. Next to ‘Node Label’, click ‘ID’ and select ‘Column 3’. All nodes should now be relabeled according to the original data labels. Minimize the ‘Node Label’ property by selecting the minus icon.

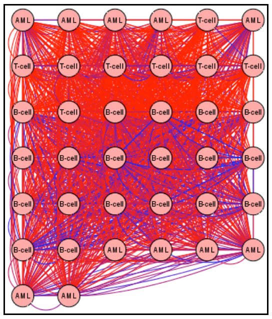
k. Under ‘Unused Properties’ find ‘Edge Color’ and double-click it.

    i. Select ‘Column 3’ as a value.

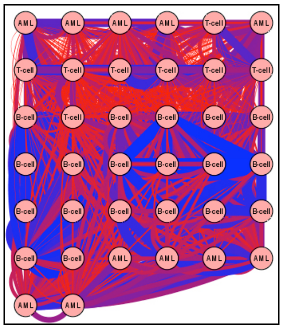
    ii. Then, select ‘Continuous Mapper’ for ‘Mapping Type’.

    iii. Click on the black-to-white gradient next to ‘Graphical View’ to launch a Gradient Editor.

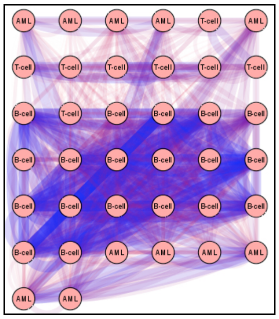
    iv. There are two fixed triangles, one on each end, and two adjustable triangles. Double-click the two leftmost triangles and set their colors to pure red (255,0,0). Double-click the two rightmost triangles and set their colors to pure blue (0,0,255). Drag the leftmost adjustable triangle all the way to the left and likewise drag the rightmost triangle to the right until it stops.



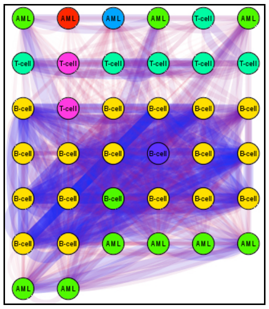
l. Find ‘Edge Line Width’ under ‘Unused Properties’ and double-click it.
   i. Select ‘Column 3’ as a value.
   ii. Then, select ‘Continuous Mapper’ for ‘Mapping Type’.
   iii. Click on the graph next to the ‘Graphical View’ property to launch the ‘Continuous Editor’.
   iv. Adjust the minimum and maximum values denoted by red squares (double-click on squares for precision, otherwise slide squares up or down). For example, set minimum to 0.5 and maximum to 20. Then, exit ‘Continuous Editor’.



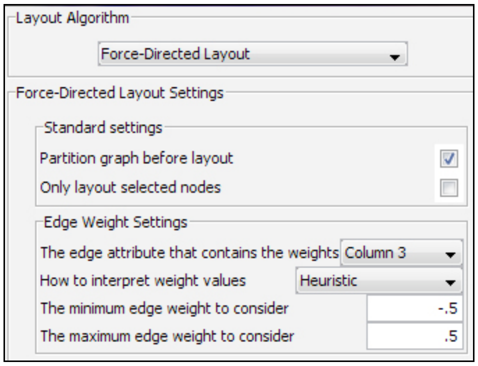
m. Find ‘Edge Opacity’ under ‘Unused Properties’ and double-click it.
   i. Select ‘Column 3’ as a value.
   ii. Then, select ‘Continuous Mapper’ for ‘Mapping Type’.
   iii. Click on the graph next to the ‘Graphical View’ property to launch the ‘Continuous Editor’.
   iv. Adjust the minimum and maximum values denoted by red squares (double-click on squares for precision, otherwise slide squares up or down). For example, set minimum to 0.5 and maximum to 60. Then, exit ‘Continuous Editor’.



n. Finally, find ‘Node Color’ under ‘Unused Properties’ and double-click it.
    i. Select ‘Column 2’ as a value.
    ii. Then, select ‘Discrete Mapping’ for ‘Mapping Type’.
    iii. Right click on ‘Discrete Mapping’ and go to ‘Generate Discrete Values’>’Rainbow 1’. All nodes are now colored according to cluster labels. Adjust colors as necessary.

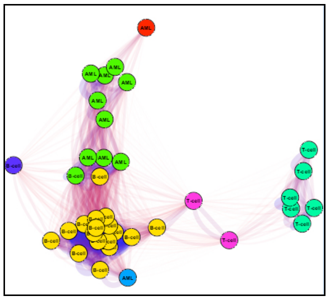


o. Go to ‘Layout’ in the main menu and select ‘Settings’
    i. Choose ‘Force-Directed Layout’ for ‘Layout Algorithm’
    ii. Under ‘Edge Weight Settings’, set ‘The minimum edge weight to consider’ to -0.5, and set ‘The maximum edge weight to consider’ to 0.5. Further, select Column 3 from ‘The edge attribute that contains the weights’.



Layout Algorithm
Force-Directed Layout
Force-Directed Layout Settings
Standard settings
Partition graph before layout ☑
Only layout selected nodes ☐
Edge Weight Settings
The edge attribute that contains the weights: Column 3
How to interpret weight values: Heuristic
The minimum edge weight to consider: -.5
The maximum edge weight to consider: .5

    iii. Press ‘Execute Layout’ to run the layout algorithm.



    iv. Although the network topology is generally preserved, different runs of the layout algorithm can yield slightly different results in terms of network rotation and local node placement. To increase the repulsion between neighboring nodes (for evenly spaced nodes within a cluster), increase ‘Default Node Mass’ under ‘Algorithm settings’.

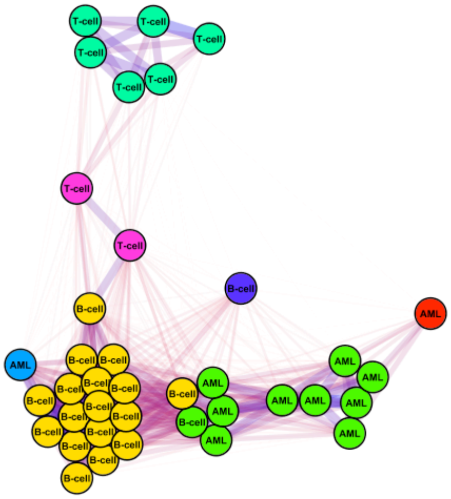
Another layout algorithm that can yield comparable results is the ‘Edge-weighted Spring Embedded’ algorithm. Before executing the layout, make sure the ‘Edge Weight Settings’ are adjusted as in (ii) above. This layout algorithm can yield more evenly spaced nodes, but is less stable than ‘Force-Directed Layout’. Run a few times.

v. Notice that all edges are slightly curved. To straighten edges, save and reopen the Cytoscape file.

vi. At this point, make any desirable fine-grained changes to the ‘Edge Color’, ‘Edge Line Width’ and ‘Edge Opacity’ parameters to emphasize the fuzziness in the network.

p. To export the final network, go to ‘File’>’Export’>’Network View as Graphics…’

q. Then, select file format and save image!



r. To expedite this process for next time, start with your saved network. All visualization parameters will still be specified. Then, simply input your edge and node files and perform the layout. Note that for the Node Color property, it is important to assign colors again using the process given in step (n).

References

1) D'haeseleer P: **How does gene expression clustering work?** *Nature Biotechnology* 2005, **23(12):**1499-1501.
2) Eisen MB, Spellman PT, Brown PO, Botstein D: **Cluster analysis and display of genome-wide expression patterns.** *Proc Natl Acad Sci USA* 1998, **95(25):**14863-14868.
3) Saldanha A: **Java Treeview—extensible visualization of microarray data.** *Bioinformatics* 2004, **20(17):**3246-3248.
4) Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks.** *Genome Research* 2003, **13:**2498-2504.