in press for Computational Biology of Embryonic Stem Cells, edited by Ming Zhan

Chapter X. Exploring Stem Cell Gene Expression Signatures using AutoSOME Cluster Analysis

Aaron M. Newman and James B. Cooper
University of California, Santa Barbara

Abstract: Stem cell laboratories around the world routinely generate whole-genome expression data to study systems-level processes in stem cell biology, and computational clustering methods are critical for the genome-wide analysis of such large data sets. To address major limitations with commonly used clustering approaches, we developed a novel computational method called AutoSOME to automatically cluster large, high-dimensional data sets, such as whole-genome microarray expression data, without prior knowledge of cluster number or structure. In previous work we demonstrated that AutoSOME clustering is an effective method for studying genome-wide expression patterns in stem cells. Here we present a primer that describes how to use this new method to perform comprehensive cluster analyses of stem cell gene expression data. We include two detailed protocols illustrating the identification of gene co-expression modules and clusters of cellular phenotypes in a single step (Protocol 1), and the visualization of transcriptome variation among stem cells using an intuitive network display (Protocol 2). The workflow described in this chapter is sufficiently general for use with a wide variety of in-house and publicly available genomics data sets.

Introduction

Stem cells have significant potential for elucidating fundamental mechanisms of developmental and disease biology, and are widely believed to hold great promise for regenerative medicine. In recent years, activity in stem cell research has greatly accelerated, owing largely to the advent of cellular reprogramming [1], an increase in funding (see [2] and [3]), and the use of systems-level technologies to characterize the pluripotent state (e.g. [4-9]). For example, since the successful generation of induced pluripotent stem cells (iPSCs) from mouse fibroblasts in 2006 [1], increasingly effective strategies have been devised for creating iPSCs from various progenitor cells (e.g. [10-12]), and viable mice were born from iPSC-derived embryos [13]. Important insights have also been made with regard to similarities and differences in gene expression [6, 14-17] and methylation patterns of iPSCs compared to embryonic stem cells (ESCs) [8, 18-19]. In addition, key components of pluripotency regulatory networks are being defined (e.g. Oct4 [20], p53 [21], miRNA-145 [22]), and the cellular and molecular aspects of tissue regeneration are being elucidated [23, 24].

Many laboratories now utilize powerful functional genomics approaches to dissect the systems-level processes underlying stem cell biology. Such high-throughput strategies, including conventional microarrays, SNP [9] and miRNA [6] arrays, ChIP-on-chip [4], CHARM

methylation profiling [8], and massively parallel sequencing [7] yield very large data sets that are generally deposited with online repositories such as the Gene Expression Omnibus (GEO) (http://www.ncbi.nlm.nih.gov/geo/) or the ArrayExpress archive (http://www.ebi.ac.uk/microarray-as/ae/). Primarily consisting of whole-genome microarray outputs, these pluripotent stem cell data sets have grown in number about three-fold every two years, from five GEO data sets in 2005 to fifty in 2009. Importantly, archived data sets can be reanalyzed to gain new insights into systems-level stem cell biology (e.g. [14, 16-18]), and thus represent a valuable resource for the stem cell community.

Without the analytical power of computational and statistical methods, large genomics data sets are virtually impossible to interpret and thus of limited utility. Unsupervised clustering is one widely used strategy applied to large data sets for identifying groups of similar patterns, such as, for example co-regulated transcripts (for review, see [25]). Although many kinds of unsupervised clustering methods are available, commonly used approaches, like K-Means and Hierarchical clustering, have major limitations for identifying natural data clusters. For instance, K-Means is restricted to symmetrical clusters, is unable to detect outlier data points, and requires prior knowledge of cluster number or use of an external cluster number prediction method [26]. Hierarchical clustering methods, such as those implemented in Eisen's widely used Cluster 3.0 software [27], are also unable to identify the number of clusters [28], make irreversible local decisions that can decrease cluster quality [29], and are inefficient on large genomics data sets [28]. These methods, and many others (see [30]), are less than ideal for researchers seeking to identify and study biologically meaningful patterns from the large amounts of data generated by high-throughput technologies.

To address limitations with the most commonly used clustering strategies, we recently developed and validated a new computational method, called AutoSOME, capable of identifying data clusters of diverse geometries from large high-dimensional data sets without prior knowledge of cluster number or structure [31]. The AutoSOME method is based on a serial application of well-established techniques from different fields, including machine learning, cartography, and graph theory. As demonstrated by the finding of a large protein-protein interaction (PPI) network up-regulated in pluripotent stem cells [31] and by the finding that pluripotent stem cells exhibit lab-specific gene expression signatures [17], AutoSOME provides a valuable approach for analyzing the genetic relationships among different cell lines from large genomics data sets. The AutoSOME method is implemented in Java and packaged within a Graphical User Interface (GUI) to accommodate end-users with diverse backgrounds using diverse computer operating systems (http://jimcooperlab.mcdb.ucsb.edu/autosome).

Here we propose a standard protocol for whole-genome expression analyses based on AutoSOME clustering, and present a primer illustrating the use of the AutoSOME GUI for exploring stem cell gene expression data. Although the protocols in this chapter utilize specific

publicly available microarray data sets, the workflow is sufficiently general for use with any number of diverse in-house or publicly available gene expression data. Key steps preceding cluster analysis are described first, including how to import, filter, and normalize microarray gene expression data using built-in GUI functions. Major cluster parameters are subsequently reviewed followed by two detailed examples that demonstrate how to perform an AutoSOME cluster analysis on stem cell microarray data.

Importing Gene Expression Data

AutoSOME accepts two major input file formats. The first input format is a table of numerical values, as shown in table 1, with one column of unique gene labels (left column) and one row of array labels (top row). All entries must be tab, comma, or space delimited. The second major input format, called a Gene Expression Omnibus Series Matrix File, available online at http://www.ncbi.nlm.nih.gov/geo/, consists of normalized gene expression data generated from a microarray experiment deposited in the GEO archive. AutoSOME can automatically extract expression data and column names from a series matrix file, allowing for rapid microarray re-analysis and meta-analysis. Once imported, gene expression data are represented as a matrix composed of *n* data rows, or gene probes, and *m* data columns, or arrays.

Table 1: Basic Input Format.

Probe	hESC-1	hESC-2	hESC-3	hESC-4	iPSC-1	iPSC-2	iPSC-3
212853_at	8.22	8.29	7.69	8.22	10.13	10.26	10.22
212854_x_at	8.52	8.71	8.04	9.00	8.88	8.97	9.08
212855_at	10.64	10.41	10.60	11.04	12.09	11.91	12.05

Microarray Data Preprocessing

To reduce noise and increase cluster quality, several preprocessing procedures for gene probe filtration and microarray normalization are available in the AutoSOME GUI. The procedures described below are tailored for *intensity* microarray data. For *two-colored* expression data, different normalization steps will be required.

Data Filtration

Gene probe filtration is a common preprocessing step for whole-genome microarray data cluster analysis. By removing gene probes corresponding to transcripts with low background-level expression or low variance across experiments, filtration can decrease algorithm running time and increase the overall signal-to-noise ratio. Filtration options currently available in the

AutoSOME GUI are the removal of transcripts with a fold change less than *X* and/or the removal of transcripts with a mean expression value below some threshold *Y*.

Normalization

Normalizing microarray data is an *essential* preprocessing step to remove technical bias and mitigate the impact of outlier data points on cluster identification. Unfortunately, the most appropriate normalization protocol is not always straightforward (especially to end-users without knowledge of the various normalization techniques commonly employed). The AutoSOME GUI implements several major normalization methods for data clustering, each of which is described below with recommendations for proper usage (G=recommended for clustering genes, or data rows; A=recommended for clustering arrays, or data columns). All operations are performed in the order listed, from top to bottom. (For technical descriptions, see the manual to the Cluster software [27] at http://rana.lbl.gov/manuals/ClusterTreeView.pdf)

Log_2 Scaling (G, A)

By amplifying small-scale changes in gene expression, log_2 scaling prevents transcripts with low levels of expression from being overshadowed by more highly expressed transcripts. Since AutoSOME works best in the log space, this data adjustment strategy should be used whenever expression values span several orders of magnitude over the entire microarray data set (e.g. 0.3-20,000). Importantly, unlike data normalization methods, log_2 transformation is a scaling procedure that is completely reversible.

Unit Variance (G, A)

Based on the assumption that all arrays have a normal distribution, this technique standardizes all arrays to zero mean expression and a standard deviation of one. When there is no *a priori* reason to treat any array differently from any other, we strongly recommend using unit variance normalization (even after raw microarray data have been pre-normalized by RMA, MAS5, etc.).

Median Centering (G)

Median centering sets the median of each row and/or column equal to zero. In the context of microarrays, this procedure centers the expression pattern, or "waveform," of each gene (or array) so that expression patterns can be isolated and compared without being affected by differences in transcript abundance. We highly recommend applying median-centering to rows in *all* cases where AutoSOME will be used to identify genes with similar co-expression signatures.

Sum of Squares=1 (G)

Sum of Squares=1 normalization yields substantial data smoothing by setting the sum of squares (x^2) of all expression values equal to 1 for each gene over all arrays and/or each array over all genes. The impact of this method on cluster identification can be significant and tends to result in the detection of large coherent clusters trailing off into genes with minimal differential expression (background noise can be removed by filtering the data prior to clustering or by using

the AutoSOME confidence filter after clustering, see *confidence filter* in the online AutoSOME manual at http://jimcooperlab.mcdb.ucsb.edu/autosome/files/AutoSOME_Manual.pdf. For an example of clusters identified using sum of squares normalization, see the heat map presented in Figure 6A of [31]. We recommend applying sum of squares normalization to both rows and columns whenever AutoSOME is used to cluster a microarray data set that has been previously filtered to remove genes with minimal variance (see Data Filtration above).

For additional data adjustment options, one can use Microsoft Excel or the Cluster 3.0 software tool [17] before importing the data into AutoSOME.

Materials

For the protocols described herein, the following software and data sets are needed:

AutoSOME and Java

Download AutoSOME from http://jimcooperlab.mcdb.ucsb.edu/autosome/download.jsp. Since AutoSOME is coded in Java, in principle, it can be run using any operating system with Java Standard Edition (Java SE) 1.6+, available from http://www.oracle.com/technetwork/java/index.html. In general, we recommend using a computer with at least 1.6GB RAM and at least a dual-core CPU. Of course, the more memory and dedicated cores, the better the performance. Microarray data sets like the Affymetrix HG-U133plus2 chipset (>54k probes) with dozens of samples can be run with 1.6GB RAM (maximum RAM that can be allocated for 32-bit Java systems), however, large data sets with many arrays (e.g. >200) will benefit from the additional RAM made possible by systems with the 64-bit Java Runtime Environment (up to ~30GB RAM).

Cytoscape

To run the second protocol, Cytoscape 2.6.0 [32], a network visualization tool, needs to be installed on your computer (download from http://cytoscape.org/download_list.php). Although recent versions of Cytoscape are available (2.6.1 to 2.7.0), to display straightened edges in the fuzzy cluster network visualization introduced in Protocol 2, Cytoscape 2.6.0 is currently needed.

Microarray Data

The protocols in this book chapter make use of two publicly available GEO data sets, GSE22651 (http://www.ncbi.nlm.nih.gov/projects/geo/query/acc.cgi?acc=GSE22651) and GSE19164 (http://www.ncbi.nlm.nih.gov/projects/geo/query/acc.cgi?acc=GSE19164). The hyperlink for the Series Matrix File corresponding to each data set is located at the bottom of the GEO data set page under *Download family*. Unzip each Series Matrix File using a decompression tool that can handle the '.gz' format (e.g., 'WinRaR', available at http://www.rarlab.com) and save it to your hard drive.

Running AutoSOME

AutoSOME can be launched from the AutoSOME website via Java Web Start, or by downloading the executable. To use the downloadable version (http://jimcooperlab.mcdb.ucsb.edu/autosome/download.jsp), unzip all contents to the same directory. If you are using Windows, you can run AutoSOME by double-clicking on one of the two batch files that come with the download (e.g. runautosome-win32-maxRAM.bat). Otherwise, using your system terminal, navigate to the directory where you installed AutoSOME, and run the following command:

java -Xmx1600m -Xms1600m -jar autosome_vXXXXXX.jar

(where XXXXXX = MMDDYY represents the date of compilation (e.g. 080110)). Since AutoSOME has not been internationalized, if you are using a computer with a primary language other than English, insert '-Duser.language=en' before '-jar' (otherwise number representation issues can arise). The -Xmx, -Xms arguments allocate additional memory (in megabytes) to AutoSOME for running large datasets. All, or most, available memory should be allocated. In operating systems running 32-bit Java, the maximum amount of memory that can be allocated to AutoSOME is about ~1.6 GB, while operating systems running 64-bit Java can allocate up to ~30 GB of memory. To see which Java version is installed on your computer, launch a terminal window and type "java -version".

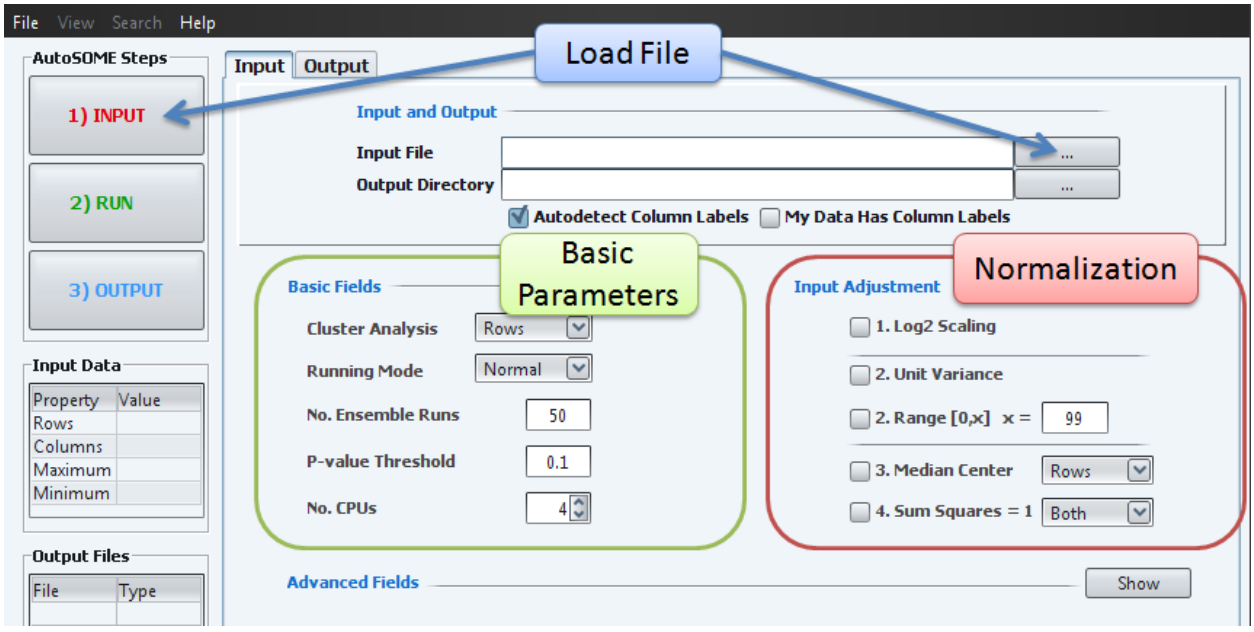


Figure 1: Layout of AutoSOME GUI main window.

Basic Parameters

A screenshot of the AutoSOME GUI layout is presented in Fig. (**1**). We recommend reviewing the basic AutoSOME parameters, described in this section, before running the protocols in this chapter. These parameters control key aspects of the AutoSOME cluster analysis, including whether AutoSOME will cluster genes and/or array experiments, how long AutoSOME will take to run, and the statistical significance, or granularity, of the cluster output. This section reviews the 'Basic Fields' parameters: Cluster Analysis, Running Mode, No. Ensemble Runs, P-value Threshold, and No. CPUs (see Basic Parameters in Fig. **1**).

Cluster Analysis

Use the Cluster Analysis combo box to switch among clustering rows (genes), columns (arrays), or both (genes followed by arrays).

Running Mode

Use the Running Mode combo box to select among 'Precision', 'Normal', or 'Speed' modes of operation. Each mode specifies different parameters for two major components of the AutoSOME method, Self-Organizing Map (SOM, see [33]) and density-equalization. Greater training of the SOM and greater resolution of density-equalization can lead to more accurate delineation of cluster boundaries. The 'Precision' mode takes longest (SOM=2X1000 iterations, density-equalization resolution=64X64), but has the best chance of resolving difficult cluster borders. On the other hand, 'Speed' (SOM iterations=2X250, density-equalization resolution=16X16) is very rapid, and is useful for first-pass exploratory cluster analysis. In our experiments, a compromise between the two extremes, 'Normal' (SOM iterations=2X500, density-equalization resolution=32X32), generally yields comparable results to 'Precision' with the benefit of increased speed. Depending upon desired expediency, we recommend selection of either 'Normal' or 'Precision' for final clustering results.

Ensemble Runs

AutoSOME stochastically samples a large cluster space and makes use of an ensemble averaging procedure to stabilize the cluster output. As demonstrated in Newman and Cooper, 2010a, increasing ensemble iterations can dramatically reduce output variance and increase cluster quality. Additional ensemble stability tests indicate that gene co-expression clusters in noisy whole-genome microarray data exhibit the greatest gain in cluster stability by 50 ensemble iterations (data not shown). Co-expression clusters continue to gradually stabilize with increasing iterations past 50. While the default of 50 ensemble iterations is enough to investigate the cluster structure of most data sets, we recommend using less ensemble iterations (e.g. 10-20) for a first-pass exploratory analysis and using 100-500 iterations for a final clustering.

P-Value Threshold

A critical step of the AutoSOME method involves partitioning a graph containing all input data points into a set of data clusters. The p-value threshold allows the data graph to be cut into statistically significant clusters based on a simulated null hypothesis of random data points. The smaller the p-value the tighter (and smaller) the resulting clusters. A default threshold of ≤ 0.1 has been extensively benchmarked to yield consistently good accuracy on a wide variety of clustering problems (see [31]). Lower the p-value threshold for increasingly challenging datasets or increasingly fine-grained clusters.

No. CPUs

Due to the ensemble averaging step, AutoSOME running time will decrease linearly with respect to an increasing number of dedicated CPUs, and thus, all available CPU cores are allocated by default.

Protocols

This section consists of two general protocols that illustrate the use of AutoSOME for large-scale exploration of gene expression signatures. Both protocols make use of the publicly available software and data sets described in the Materials section (above). The first protocol demonstrates how to use AutoSOME to identify both gene co-expression modules and transcriptome clusters from publicly available microarray data. The second protocol shows how to use Cytoscape [32] to visualize stem cell transcriptome variation using an intuitive two-dimensional network schematic called a "fuzzy cluster network" (e.g. see Figure 3 in [17]).

Protocol 1: AutoSOME Co-Expression and Transcriptome Clustering

Here we show how to cluster both transcripts and transcriptomes in a single efficient step with the AutoSOME GUI. By executing this protocol, you will also be introduced to important output features of the AutoSOME GUI, including displaying and adjusting heat maps, and saving publication quality figures of the cluster output. Due to the inherent noise in microarray data sets and the stochastic component of AutoSOME, your cluster results may vary slightly (and only slightly) from those presented here.

1) Launch AutoSOME and press the large 'INPUT' button (see Fig. **1**). A file browser will appear. Select the 'Gene Expression Omnibus Series Matrix File' checkbox located in the data format box over the browsing window (otherwise there will be an input error). Browse to where you saved the GSE22651 text file (should have been saved as 'GSE22651_series_matrix.txt', see Materials), select it, and press the 'Open' button.

2) The number of arrays (=65) and gene probes (=48,786), along with maximum and minimum data values, will be shown. You are asked whether you want to filter your data. Press 'Yes'. A 'Filter Data' window will open.

3) Since the expression values span a range of 26 to 27,446, the data have not been $\log_2$ scaled. We can leave the corresponding checkbox deselected. To reduce the computational load and generate cleaner clusters, remove gene probes with a fold change (maximum over minimum value) less than 4. Press 'Apply' to preview the filtered data results. The filtered data set has 14,990 rows (gene probes). Press 'Accept'.

4) In the main GUI window, expand 'Basic Fields' by pressing 'Show'. Since AutoSOME will be used to cluster both filtered transcripts and transcriptomes, select 'Both' from the 'Cluster Analysis' combo box. 'Basic Fields' will expand to 'Basic Fields (Rows)' and 'Basic Fields (Columns)'. Under 'Basic Fields (Rows)', set 'Running Mode'=Normal, 'No. Ensemble Runs' to 100, 'P-value Threshold' =0.1, and underneath in 'Basic Fields (Columns)', set 'No. Ensemble Runs' to 200 and 'P-value Threshold' =0.1.

5) Expand 'Input Adjustment'. Under 'Input Adjustment (Rows)', select 'Log_2 Scaling', 'Unit Variance', 'Median Centering' of 'Rows', and 'Sum Squares = 1' 'Both'. Under 'Input Adjustment (Columns)', select 'Log_2 Scaling' and 'Unit Variance'.

   Note: AutoSOME can also be used to cluster unfiltered microarray data (and even larger data sets). In this case, 'Sum of Squares=1' normalization is **not** recommended.

6) Press the large 'RUN' button. Progress and elapsed time are shown in the 'Run Progress' box located in the lower-left region of the main GUI window. Note that running time will vary depending upon the number of dedicated CPUs and CPU clock speed. (For this data set a typical run time of ~10 minutes should be expected using a 3GHz dual CPU computer.)

   Note: if AutoSOME does not finish, there may not be enough memory available. If you have more RAM on your computer, simply allocate additional RAM to AutoSOME at start-up (1.6GB is sufficient; see Running AutoSOME). There is also an option to write intermediate ensemble runs to disk (go to 'Advanced Fields'>Memory from the main window of the GUI). If selected, a temporary folder will be created in the current working directory.

7) Once clustering is finished, AutoSOME will write output files to disk (see table 2), and the main window will be redirected to the 'Output' tab. AutoSOME will identify 3 array (or transcriptome) clusters and approximately 42 gene (or co-expression) clusters. Co-expression clusters are displayed as a list in the 'Cluster Output' tree with the number of transcripts in each cluster shown in parentheses. Select 'cluster 1' with your mouse. By default, a table of all gene probe labels in cluster 1 (along with cluster confidence values, see Newman and Cooper 2010a) will be displayed.

Table 2: AutoSOME Output Files. By default, all output files will be written to the parent directory of your input file. ('MyInput' = name of input file, 'X' = the number of ensemble runs, 'Y' = the p-value, and 'Z' = 'rows' or 'columns' depending upon what whether genes or arrays were clustered, respectively (e.g. AutoSOME_GSE22651_E100_Pval0.1_rows.txt)).

Cluster Rows or Columns?	File Name	Description	Open in GUI?
Either	AutoSOME_**MyInput**_E**X**_Pval**Y**_**Z**_summary.html	List of all AutoSOME parameters, and cluster summary table for **either** row **or** column clustering	No
Either	AutoSOME_**MyInput**_E**X**_Pval**Y**_**Z**.html	HTML version of all clusters and confidence values for **either** row **or** column clustering	No
Either	AutoSOME_**MyInput**_E**X**_Pval**Y**_**Z**.txt	Text file of all AutoSOME clusters and confidence values for **either** row **or** column clustering (stores original data prior to normalization)	**Yes**
Both	AutoSOME_**MyInput**_Pval**Y**_rows_columns.txt	Text file of all AutoSOME clusters and confidence values for **both** row **and** column clustering (stores original data prior to normalization)	**Yes**
Columns	AutoSOME_**MyInput**_E**X**_Pval**Y**_Edges.txt	Fuzzy cluster network **edges and edge weights** for use with Cytoscape [Shannon et al., 2003]	No
Columns	AutoSOME_**MyInput**_E**X**_Pval**Y**_Nodes.txt	Fuzzy cluster network **nodes** for use with Cytoscape [Shannon et al., 2003]	No
Columns	AutoSOME_**MyInput**_E**X**_Pval**Y**_Matrix.txt	Fuzzy cluster network edge weights in matrix form (can be hierarchically clustered using Cluster 3.0 [27] and visualized in Java TreeView [34])	No

8) While cluster 1 is still selected, select 'View'>'heatmap'>'green red' from the dropdown menu. A traditional heat map will be displayed. A scale-bar above the heat map shows the maximum and minimum normalized expression values. A cluster confidence vertical bar on the left side of the heat map indicates the confidence of each gene probe for its cluster (Fig. **2A**, blue=100% confidence, red=0%, see Figure 2 in [31]). Since we clustered arrays as well as gene probes, white vertical bars are shown separating different array clusters (Fig. **2A**).

9) To display heat maps of more than one gene co-expression cluster simultaneously, hold Shift or Ctrl to select multiple clusters from the cluster list using your mouse. Let's select clusters 1-10 (hold Shift). Since the heat map is too large to see all clusters without scrolling, go to 'View'>'fit to screen'. Now all 10 co-expression clusters are visible separated by white bars (see Fig. **2A**). (If some clusters extend beyond the display after using the fit screen function, the clusters at the bottom may be viewed by using the scroll bar, changing the image dimensions using the 'image settings' window (see below), or using the mouse scroll wheel.)

10) To more clearly visualize expression differences in the heat map, we can change the normalization settings and heat map contrast. Go to 'View'>'settings'>'image settings' to launch a new window. The 'Image Settings' window will appear showing a wide variety of adjustable display settings (Fig. **2B**). To use the heat map to accurately display expression differences, we need show the data prior to normalization. To do this, select 'Display Original Data'. Since the raw expression data span a very large range, select Log_2 scaling. Next, select 'Median Center Rows' to center all gene probes. Finally, let us adjust the heat map contrast. You can slide the 'Heat Map Contrast' bar to the left (e.g. to 0.2) and let go, or for more precise contrast adjustment, select 'Manually adjust range for contrast' and input explicit minimum and maximum values, such as, for example, -2 to 2. For the typed-in manual adjustments to take effect it is then necessary select the update button. Note that the color-bar in the heat map updates to reflect the new value range. Finally, switch to the 'Display Options' tab and select 'Hide Heat Map Row Labels' to remove gene probe labels from the right side of the heat map (these cannot be seen at the current resolution). The heat map should look similar to the one shown in Fig. **2B**.

11) To save a high-resolution image of this heat map, increase the 'Zoom Factor' bar to 50 (zoom factor is the width in pixels of each column in the heat map). Let's leave the 'Adjust Height' bar the same since it was determined by the 'fit to screen' command (the height of each row in the heat map is determined by multiplying this number by the zoom factor). Press the 'Save' button at the bottom of the 'Image Settings' window to write the heat map to disk (Portable Network Graphics (PNG) format only). Although the image will overflow the display window, the entire image will be saved to file. In addition to being able save the heat map images created in the output window, AutoSOME automatically saves additional output files (see table 2).

Anticipated Results

The filtered GSE22651 data set has two large gene co-expression clusters that distinguish pluripotent stem cell lines from somatic cell lines (see Fig **2C**). There are also three filtered transcriptome clusters, an undifferentiated pluripotent stem cell cluster and two somatic cell line clusters: a cluster of fibroblasts, mesenchymal cells, keratinocytes, and human umbilical vein endothelial chord cell lines, and a smaller cluster composed of transcriptomes representing lung, adipose, bladder, and ureter tissue samples.

Next Step

Try repeating this protocol using a p-value threshold of 0.05 for both rows and columns. You will notice that AutoSOME identifies tighter clusters, including a distinct keratinocyte transcriptome cluster, and a bladder and ureter transcriptome cluster. In addition, several tighter co-expression clusters are now resolved. Heat maps displaying clusters obtained with p-value thresholds 0.1 and 0.05 are shown in Figs. (**2C**) and (**2D**), respectively.

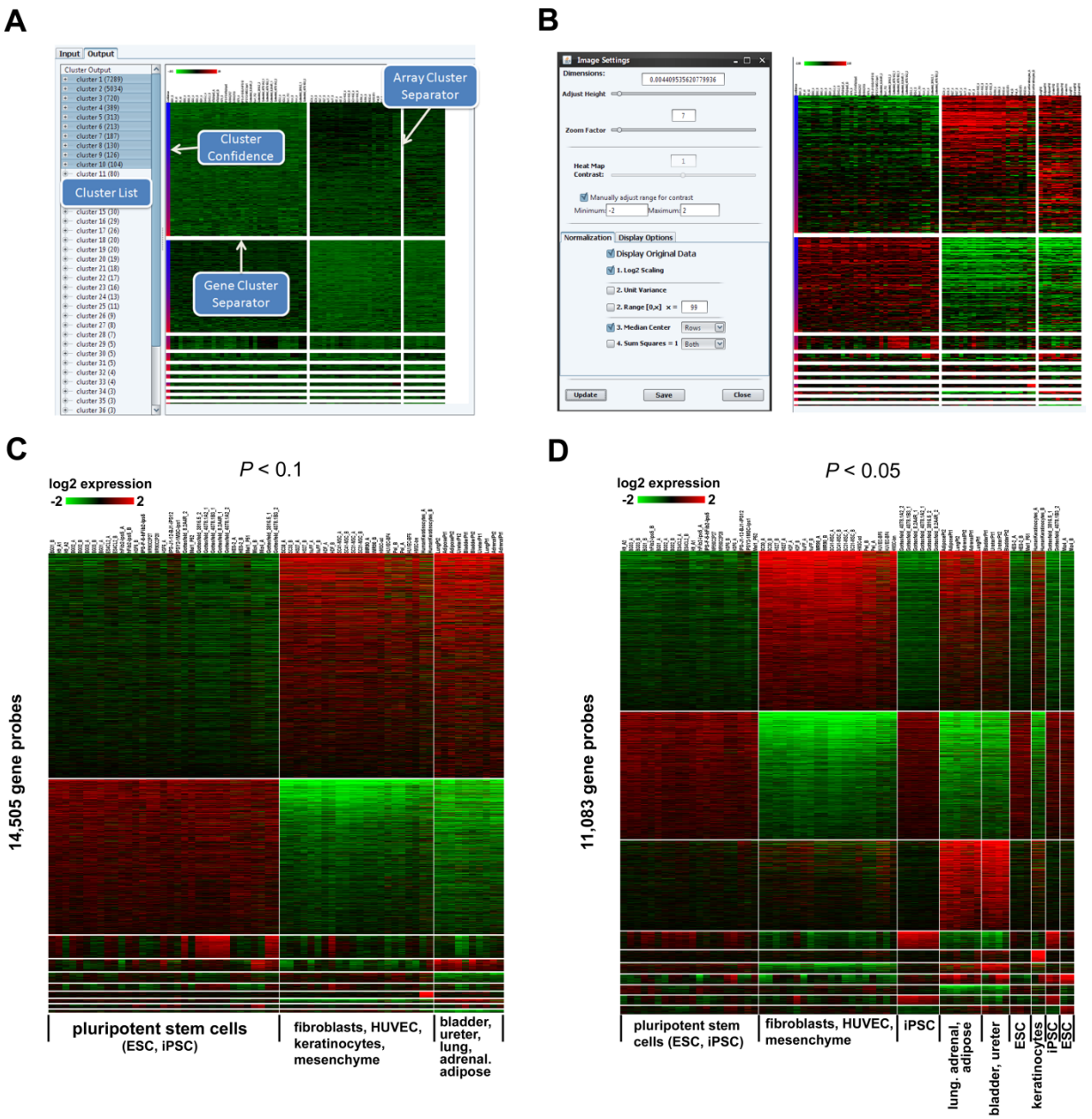


Figure 2: AutoSOME co-expression and transcriptome clustering, related to Protocol 1. (A) Cluster list and heat map output, (B) Image settings window and renormalized cluster heat map, (C) Final heat map for 10 largest gene co-expression clusters at $P < 0.1$ with cellular phenotypes corresponding to each transcriptome cluster shown underneath, (D) Final heat map for 10 largest gene co-expression clusters at $P < 0.05$ with cellular phenotypes corresponding to each transcriptome cluster shown underneath,. To render heat maps shown in panels (C) and (D), follow protocol 1 through step 10 using the appropriate p-value, then reorder each cluster by decreasing variance: go to the 'Display Options' tab in the Image Settings window (panel B) and select 'Sort by Decreasing Variance' (you may need to press 'Update' if the heat map fails to refresh), and finally, follow step 11.

Protocol 2: Exploring Stem Cell Transcriptome Variation using a Fuzzy Cluster Network

A powerful application of AutoSOME clustering, in addition to identifying discrete clusters of co-expressed genes (e.g. Fig **2C**), is identifying data points with fractional membership to one or more clusters. Such "fuzzy clusters" are a natural way of representing the inherent noise in gene expression data, and importantly, when visualized as a network diagram, provide an intuitive schematic for displaying the relationships among the transcriptome clusters identified by AutoSOME. In the first part of this protocol, AutoSOME is used to cluster the transcriptomes of several pluripotent stem cell lines, including iPSCs generated from three different combinations of reprogramming factors [12]. Using the cluster results from part one, the second part of this protocol details how to create a fuzzy cluster network. Before proceeding, it would be useful to become familiar with the transcriptome clustering strategy utilized by AutoSOME, which involves the construction of a distance matrix of transcriptome profiles.

Transcriptome Clustering using a Distance Matrix

The number of microarray gene probes *n* is usually much greater than the number of arrays *m*. To decrease the computational load for clustering transcriptomes, AutoSOME does not directly cluster transcriptome profiles, but instead clusters a matrix representing pair-wise similarities of all transcriptomes. (This amounts to performing an All-against-All comparison of *m* array expression vectors to generate a similarity matrix of size *m* by *m* used for clustering.) Three common distance metrics (Euclidean, Pearson's, and uncentered correlation) for calculating transcriptome *similarity* are provided as a user-adjustable parameter. To access the 'Distance Metric' combo box, expand 'Advanced Fields' in the GUI main window and go to 'Fuzzy Cluster Networks'. Euclidean distance is chosen by default due to superior results obtained from empirical testing (using microarray datasets with previously known classes of cell lines). With regard to Euclidean distance, similar transcriptomes have smaller distances between them. AutoSOME also implements Pearson's correlation and uncentered correlation metrics, both of which have a maximum of 1 (completely correlated) and minimum of -1 (inversely correlated). Unlike uncentered correlation, Pearson's correlation is insensitive to amplitude shifts, meaning that two transcriptomes with similar expression patterns but different amplitudes can still be highly correlated using Pearson's method. For an excellent review of these three distance metrics, see [25].

Protocol 2 Part 1: AutoSOME Transcriptome Clustering

1) Launch AutoSOME and press the large 'INPUT' button (see Fig. **1**). A file browser will appear. Select the 'Gene Expression Omnibus Series Matrix File' checkbox located in the data format box over the browsing window (otherwise there will be an input error). Browse to where you saved the GSE19164 text file (should be saved as 'GSE19164_series_matrix.txt', see Materials), select it, and press the *Open* button.

2) You will be asked whether you want to filter the data. Press 'No'.

3) In the main GUI window, expand 'Basic Fields' by pressing 'Show'. Since we are going to cluster cellular transcriptomes, select 'columns' from the 'Cluster Analysis' combo box. Set 'Running Mode'=Normal, 'No. Ensemble Runs' to 500, and 'P-value Threshold' =0.1.

4) Expand 'Input Adjustment'. Since these expression data have not been $\log_2$ scaled, select 'Log_2 Scaling'. Also, select 'Unit Variance'.

5) At this point, the distance matrix used for column clustering could be changed by expanding 'Advanced Fields' and picking another metric from the 'Distance Metric' combo box. We will use the default setting, Euclidean distance.

6) Press the large 'RUN' button.

7) After clustering is finished, output files will be written to disk (see table 2), and the main window will be redirected to the 'Output' tab. Select all clusters in the cluster list using your mouse (hold Shift), and select 'View'>'heatmap'>'rainbow' from the dropdown menu. The display should look like Fig. (**3**). The heat map shows the clustering of the Euclidean distance matrix of all arrays in the GSE19164 data set (see Choice of Distance Metric). Each transcriptome cluster is separated by a white horizontal bar. The Euclidean distance between any given pair of cellular transcriptomes can be inferred from the heat map by using the color bar. A distance of zero means that the two cell lines are identical by Euclidean distance (colored blue).

Anticipated Results

Four transcriptome clusters are identified, as shown in Fig. (**3**). Notice that ESCs and iPSCs cluster separately, and each cell type also has two distinct sub-clusters. To explore relationships among each cluster, along with individual transcriptomes, proceed to part two (below).

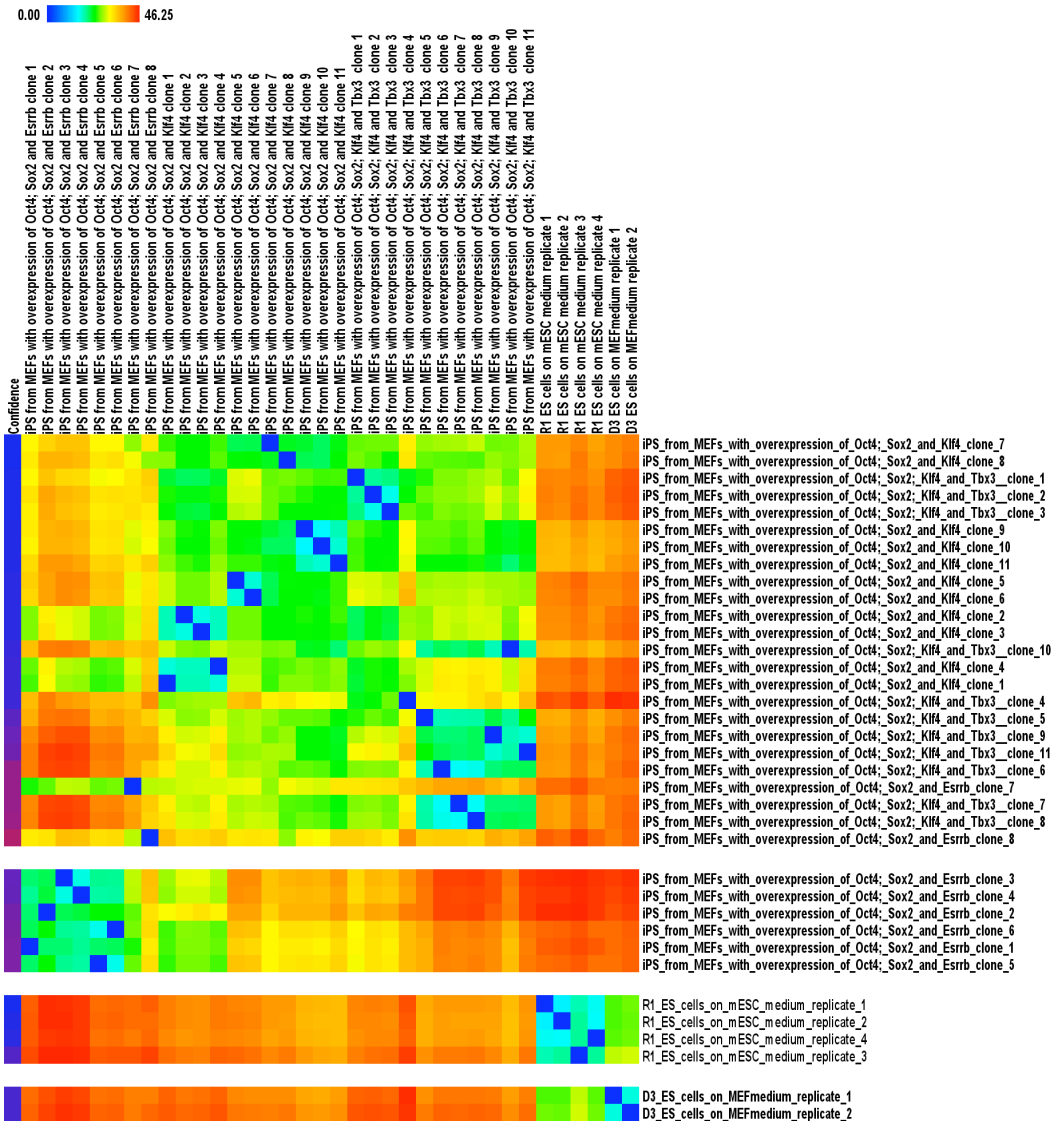


Figure 3: Clustered Euclidean distance matrix of iPSC and ESC cell lines, related to Protocol 2 Part 1.

Protocol 2 Part 2: Fuzzy Cluster Network

The fuzzy cluster network has three major components: nodes, edges, and clusters. A **node** represents a particular transcriptome, usually labeled by cellular phenotype or time point. An **edge** is defined as a link between two transcriptomes (nodes). The edge weight represents the fraction of times over all ensemble runs that the pair of transcriptomes clustered together, minus 0.5. For example, if two transcriptomes clustered together 80% of the time, or 0.8, the edge value will be 0.3. The full range of edge weights is thus -0.5 to 0.5. Finally, the **clusters** represent the

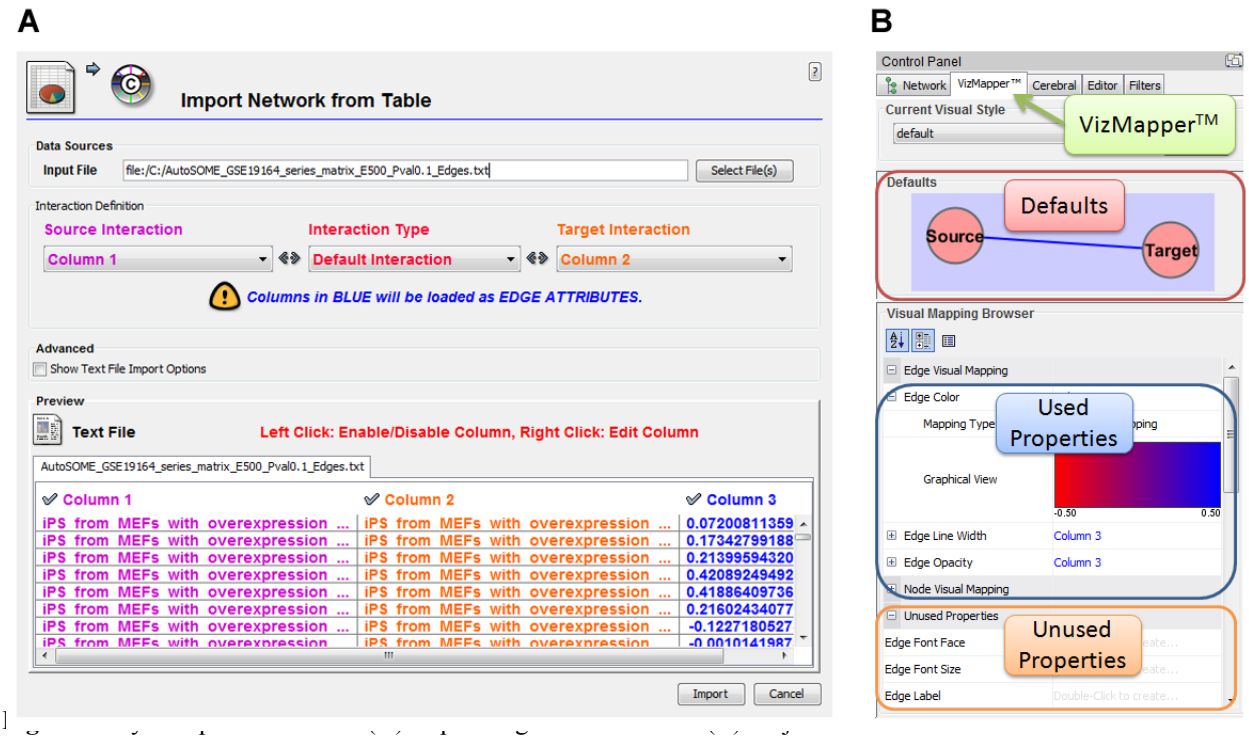
discrete clusters of transcriptomes identified by AutoSOME (e.g., see Fig. **3**). For further details, see [31].

1) First, let's modify the cell line labels from the GSE19164 data set so that they are short enough to display in the network. Using a spreadsheet editing program (e.g. Microsoft Excel), open the output file 'AutoSOME_GSE19164_series_matrix_E500_Pval0.1_Nodes.txt' (for details of this output file, see table 2). Column 1 contains identifiers that link this file to the edges file. Column 2 contains cluster numbers and Column 3 contains cell line labels. In column 3, replace all iPSC lines with the first letter of each transcription factor so that lines overexpressing Oct4, Sox2, and Klf4 are denoted 'OSK', lines overexpressing Oct4, Sox2, Klf4, and Tbx3 are denoted 'OSKT', and lines overexpressing Oct4, Sox2, and Esrrb are denoted OSE. For ESC lines, let's keep the cell type R1 and D3, and remove the remaining text to yield R1_ES and D3_ES. Save all three columns as a new text file, e.g. 'AutoSOME_GSE19164_series_matrix_E500_Pval0.1_Nodes_reformat.txt'.

2) Launch Cytoscape (for download information, see Materials). In the 'File' dropdown menu, select the option to import a network from a table (Fig. **4A**). Go to 'Select File(s)' and locate the AutoSOME output file containing all edges called 'AutoSOME_GSE19164_series_matrix_E500_Pval0.1_Edges.txt' (for details of this output file, see table 2). Press 'Open'. 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). Select 'Import', and finally, press 'Close'. A raw network will appear as a grid. To render a network with detailed graphics, go to the dropdown menu and select 'View'→'Show Graphics Details'.

3) In the 'File' dropdown menu, select the option to import attributes from a table. Go to 'Select File(s)' and locate the AutoSOME output file containing all nodes and modified cell line labels, 'AutoSOME_GSE19164_series_matrix_E500_Pval0.1_Nodes_reformat.txt'. Select 'Open' and then 'Import'.

4) Change global properties of the network: In the Control Panel, select the VizMapper™ tab (Fig. **4B**). Click in the 'Defaults' window (Fig. **4B**, shows a source and target pair with a blue background). A new window will appear. Select the 'Global' tab in the bottom right. Change the background color to white. Go back to the 'Node' tab and change the NODE_BORDER_COLOR property to black and increase the 'NODE_LINE_WIDTH' to 2. Change NODE_FONT_SIZE to 8. Press 'Apply'. The network should look like Fig. (**5A**).



5) Using the VizMapper™ tab: 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. The network will now look like Fig. (**5B**).

6) Under 'Unused Properties' (Fig. **4B**) find 'Edge Color' (top of list) and double-click it. Select 'Column 3' as a value. Then, select 'Continuous Mapper' for 'Mapping Type'. Click on the black-to-white gradient next to 'Graphical View' to launch a Gradient Editor. 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. Exit 'Continuous Editor'.

7) Find 'Edge Line Width' under 'Unused Properties' and double-click it. Select 'Column 3' as a value. Then, select 'Continuous Mapper' for 'Mapping Type'. Click on the graph next to the 'Graphical View' property to launch the 'Continuous Editor'. 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 10. Then, exit 'Continuous Editor'.

8) Find 'Edge Opacity' under 'Unused Properties' and double-click it. Select 'Column 3' as a value. Then, select 'Continuous Mapper' for 'Mapping Type'. Click on the graph next to

the ‘Graphical View’ property to launch the ‘Continuous Editor’. 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 150. Then, exit ‘Continuous Editor’.

9) Finally, find ‘Node Color’ under ‘Unused Properties’ and double-click it. Select ‘Column 2’ as a value. Then, select ‘Discrete Mapper’ for ‘Mapping Type’. 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 desired. The network should now look like Fig. (**5C**).

10) Select ‘Layout’ in the dropdown menu (top of main window) and select ‘Settings’. Choose ‘Force-Directed Layout’ for ‘Layout Algorithm’. Under ‘Edge Weight Settings’, select Column 3 from ‘The edge attribute that contains the weights’, set ‘The minimum edge weight to consider’ to -0.5, and set ‘The maximum edge weight to consider’ to 0.5. Press ‘Execute Layout’ to run the layout algorithm (see, e.g., Fig. **5D**). To increase the repulsion between neighboring nodes (for evenly spaced nodes within a cluster), increase ‘Default Node Mass’ under ‘Algorithm settings’. Although the network topology is generally preserved, different runs of the layout algorithm will yield slightly different results in terms of network rotation and local node placement. Additionally, with different data sets, it is not uncommon for the layout algorithm to generate a network that is logically flawed (i.e. with closely related nodes, as indicated by thick, dark blue edge lines, being distantly removed from each other). When this happens, simply re-execute the layout until a logically consistent network is rendered.

    Note: 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 above. This layout algorithm can yield more evenly spaced nodes, but is less stable than ‘Force-Directed Layout’. Run a few times.

11) Notice that all edges are slightly curved. To straighten edges, save and reopen the Cytoscape file (make sure you are using Cytoscape version 2.6.0 for this to work). If desired, rotate (go to ‘Select’ in the dropdown menu and press ‘Rotate’) and/or zoom the network until it looks aesthetic (see, e.g., Fig. **5E**). Some nodes may overlap with others. You may be able to manually nudge them into view without substantially altering the network topology (e.g., Fig. **5E**).

12) To export the final network, go to ‘File’→’Export’→’Network View as Graphics…’. Then, select file format and save the image.

Anticipated Results

The final fuzzy cluster network figure is shown in Fig. (**5F**). Note that nodes represent transcriptomes, edge weights represent the fraction of times that each pair of transcriptomes clustered together over all ensemble runs, and differently colored nodes represent discrete clusters from Protocol 2 Part 1 (Fig. **3**). Clearly, iPSCs are more similar to each other than to ESCs, and vice versa. Two clusters distinguish iPSCs overexpressing Oct4, Sox2, and Esrrb (OSE) from iPSCs overexpressing Oct4, Sox2, and Klf4 (OSK) or Oct4, Sox2, Klf4, and Tbx3 (OSKT). In addition, OSE lines 7 and 8 bridge both the OSE and OSK/OSKT lines, and OSK and OSKT lines are indistinguishable (at least at $P < 0.1$). These results are comparable to the hierarchical tree shown in Figure 3 of [12].

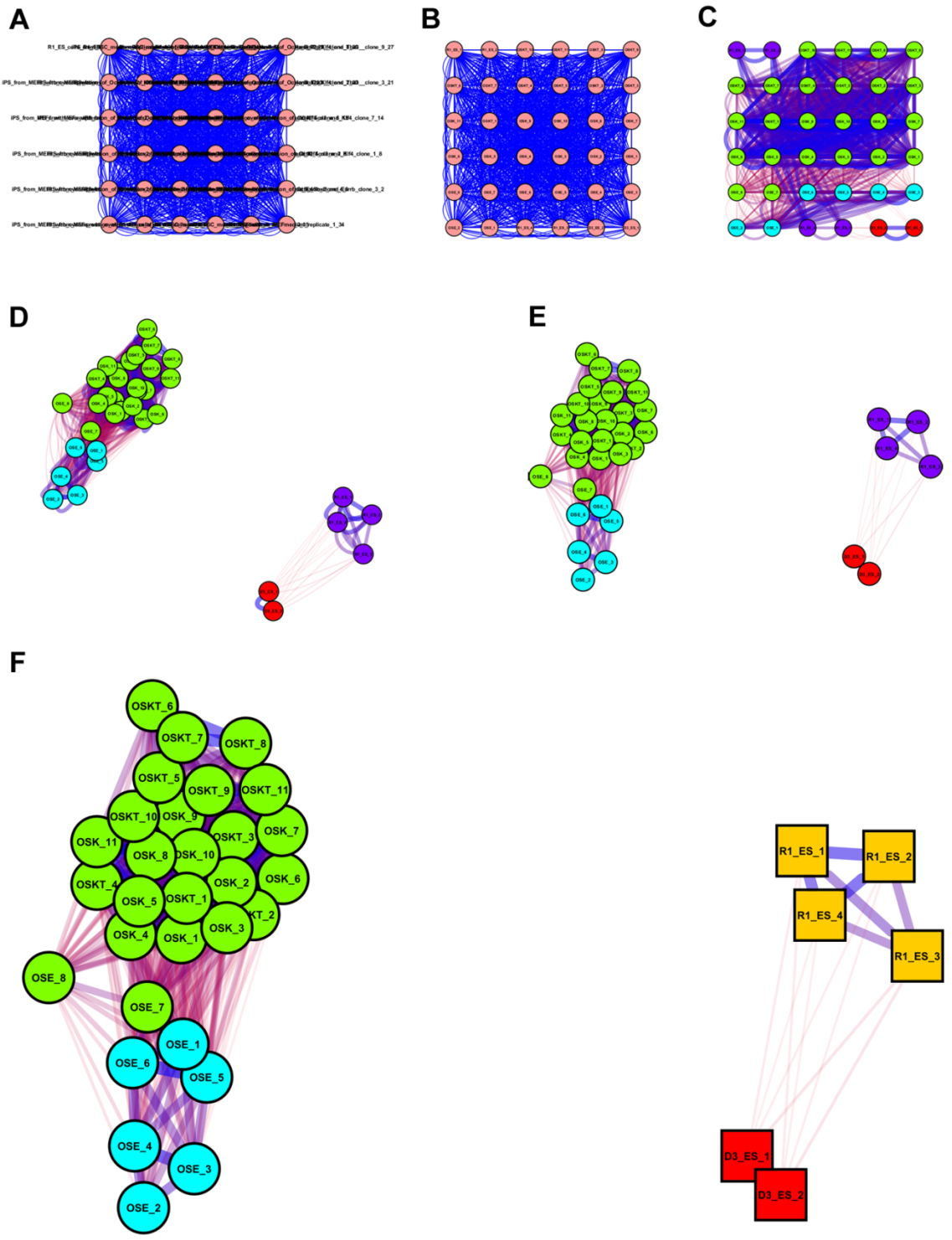


Figure 5: How to render an AutoSOME fuzzy cluster network using Cytoscape, related to Protocol 2 Part 2. (A) After importing edges. (B) After importing nodes and changing node labels. (C) After adjusting node and edge settings. (D) After performing layout algorithm. (E) After re-opening file (to straighten edges) and rotating network. (F) Final network with iPSCs denoted by circles, ESCs denoted by squares, and color adjustment of R1_ESCs to orange.

Conclusions

Clustering is the process of partitioning information into useful categories. Although the human brain is endowed with powerful classification tools, our innate faculties for identifying data clusters are challenged by the massive, often high-dimensional, data sets made possible by twenty-first century high-throughput technologies. We developed a new unsupervised clustering method for genomics research, called AutoSOME, to overcome important limitations of common clustering methods, including poor scalability to large data sets, cluster shape restrictions, lack of outlier detection, and most importantly, inability to determine the number of data clusters [31]. In this chapter, we demonstrated how AutoSOME clustering can be applied to stem cell genomics research. Specifically, we presented a primer illustrating how to use the AutoSOME GUI for microarray filtration and normalization, and for "single step" co-expression and transcriptome clustering. In addition, we showed how one can visualize transcriptome variation among stem cell lines by rendering an AutoSOME fuzzy cluster network diagram. Taken together, the workflow proposed in this chapter has utility for studying gene expression signatures in diverse cellular phenotypes and systems, and should have broad application for clustering genomics data generated by diverse microarray platforms and massively parallel sequencers.

Acknowledgements

We thank Dr. Monte Radeke, Dr. Don Anderson, and Dr. Chris Banna for testing the protocols and Dr. Monte Radeke for critically reading this manuscript.

References

[1] Takahashi K, Yamanaka S. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. Cell 2006; 126: 663-676.

[2] Hayden EC. California stem-cell grants awarded. Nature 2009; 462: 22.

[3] Wadman M. Most popular cell lines close to approval for US federal funding. Nature 2010; 464: 967.

[4] Komashko VM, Acevedo LG, Squazzo SL, *et al*. Using ChIP-chip technology to reveal common principles of transcriptional repression in normal and cancer cells. Genome Res 2008; 18: 521-532.

[5] Müller FJ, Laurent LC, Kostka D, *et al*. Regulatory networks define phenotypic classes of human stem cell lines. Nature 2008; 455: 401-405.

[6] Wilson KD, Venkatasubrahmanyam S, Jia F, Sun N, Butte AJ, Wu JC. MicroRNA profiling of human-induced pluripotent stem cells. Stem Cells Dev 2009; 18: 749-758.

[7] Guttman M, Garber M, Levin JZ, *et al.* Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. Nat Biotechnol 2010; 28: 503-510.

[8] Kim K, Doi A, Wen B, *et al.* Epigenetic memory in induced pluripotent stem cells. Nature doi:10.1038/nature09342. 2010 July 19. Available from: http://www.nature.com/nature/journal/vnfv/ncurrent/full/nature09342.html

[9] Närvä E, Autio R, Rahkonen N, *et al.* High-resolution DNA analysis of human embryonic stem cell lines reveals culture-induced copy number changes and loss of heterozygosity. Nat Biotechnol 2010; 28: 371-377.

[10] Yu J, Hu K, Smuga-Otto K, *et al.* Human induced pluripotent stem cells free of vector and transgene sequences. Science 2009; 324: 797-801.

[11] Kim D, Kim CH, Moon JI, *et al.* Generation of human induced pluripotent stem cells by direct delivery of reprogramming proteins. Cell Stem Cell 2009; 4: 472-476.

[12] Han J, Yuan P, Yang H, *et al.* Tbx3 improves the germ-line competency of induced pluripotent stem cells. Nature 2010; 463: 1096-1100.

[13] Zhao XY, Li W, Lv Z, *et al*. iPS cells produce viable mice through tetraploid complementation. Nature 2009; 461: 86-90.

[14] Chin MH, Mason MJ, Xie W, *et al.* Induced pluripotent stem cells and embryonic stem cells are distinguished by gene expression signatures. Cell Stem Cell 2009; 5: 111-123.

[15] Marchetto MCN, Yeo GW, Kainohana O, Marsala M, Gage FH, Muotri AR. Transcriptional signature and memory retention of human-induced pluripotent stem cells. PLoS ONE 2009; 4: e7076.

[16] Ghosh Z, Wilson DK, Wu Y, Hu S, Quertermous T, Wu JC. Persistent donor cell gene expression among human induced pluripotent stem cells contributes to differences with human embryonic stem cells. PLoS ONE 2010; 5: e8975.

[17] Newman AM, Cooper JB. Lab-specific gene expression signatures in pluripotent stem cells. Cell Stem Cell 2010; 7: 258-262.

[18] Guenther MG, Frampton GM, Soldner F, *et al.* Chromatin structure and gene expression programs of human embryonic and induced pluripotent stem cells. Cell Stem Cell 2010; 7: 249-257.

[19] Polo JM, Liu S, Figueroa ME, *et al.* Cell type of origin influences the molecular and functional properties of mouse induced pluripotent stem cells. Nat Biotechnol 2010; 28: 848-855.

[20] van den Berg DL, Snoek T, Mullin NP, *et al.* An Oct4-centered protein interaction network in embryonic stem cells. Cell Stem Cell 2010; 6: 369-381.

[21] Hong H, Takahashi K, Ichisaka T, *et al.* Suppression of induced pluripotent stem cell generation by the p53-p21 pathway. Nature 2009; 460: 1132-1135.

[22] Xu N, Papagiannakopoulos T, Pan G, Thomson JA, Kosik KS. MicroRNA-145 regulates OCT4, SOX2, and KLF4 and represses pluripotency in human embryonic stem cells. Cell 2009; 137: 647-658.

[23] Kragl M, Knapp D, Nacu E, *et al.* Cells keep a memory of their tissue origin during axolotl limb regeneration. Nature 2009; 460: 60-65.

[24] Pajcini KV, Corbel SY, Sage J, Pomerantz JH, Blau HM. Transient inactivation of Rb and ARF yields regenerative cells from postmitotic mammalian muscle. Cell Stem Cell 2010; 7: 198-213.

[25] D'haeseleer. How does gene expression clustering work? Nat Biotechnol 2005; 23: 1499-1501.

[26] Xu R, Wunsch II D. Survery of clustering algorithms. IEEE T Neural Networ 2005; 16: 645-678.

[27] Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 1998; 95: 14863-14868.

[28] Giancarlo R, Scaturro D, Utro F. Computational cluster validation for microarray data analysis: experimental assessment of Clest, Consensus Clustering, Figure of Merit, Gap Statistics and Model Explorer. BMC Bioinformatics 2008; 9: 462.

[29] De Souto MCP, Costa IG, de Araujo DSA, Ludermir TB, Schliep A. Clustering cancer gene expression data: a comparative study. BMC Bioinformatics 2008; 9: 497.

[30] Andropoulos B, An A, Wang X, Shroeder M. A roadmap of clustering algorithms: finding a match for a biomedical application. Briefings Bioinf 2009; 10: 297-314.

[31] Newman AM, Cooper JB. AutoSOME: a clustering method for identifying gene expression modules without prior knowledge of cluster number. BMC Bioinformatics 2010; 11: 117.

[32] Kohonen T. The self-organizing map. Proc of the IEEE 1990; 78: 1464-1480.

[33] Shannon P, Markiel A, Ozier O, *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 2003; 13: 2498-2504.

[34] Saldanha AJ. Java Treeview-extensible visualization of microarray data. Bioinformatics 2004; 20: 3246-3248.