



ExpressCluster **v1.3**

Microarray gene expression clustering

Written by Scott P. Davis

Benoist/Mathis Lab
Department of Microbiology & Immunobiology
Harvard Medical School
77 Avenue Louis Pasteur
Boston, MA 02115

Developed for use with...

GenePattern

www.GenePattern.org

OVERVIEW

Quickly and easily cluster genes by their expression profiles across any number of conditions. With a full, user-friendly graphical interface, ExpressCluster allows researchers to visualize results instantly in order to generate complex gene lists/signatures, heatmaps, and new expression datasets based on computed clusters. While the default settings will accommodate most people's needs, more advanced parameter options are available, giving maximum flexibility and control over how data are treated.

REQUIREMENTS



[Java 1.6](#) update 10 or newer.

1GB RAM.

Recommended: 4GB RAM, 64bit Java runtime.

INPUT

GenePattern expression dataset	*.gct	Required	Contains the samples/populations over which the expression value profiles will be clustered. NOTE: ExpressCluster cannot operate on sparse matrices! Missing values will generate an error.
GenePattern class file	*.cls	Optional	If this file is provided, clustering will be done on the class means (average of all replicates) instead of individual samples. Accordingly, the "Available" column will be updated to contain the names of the classes.

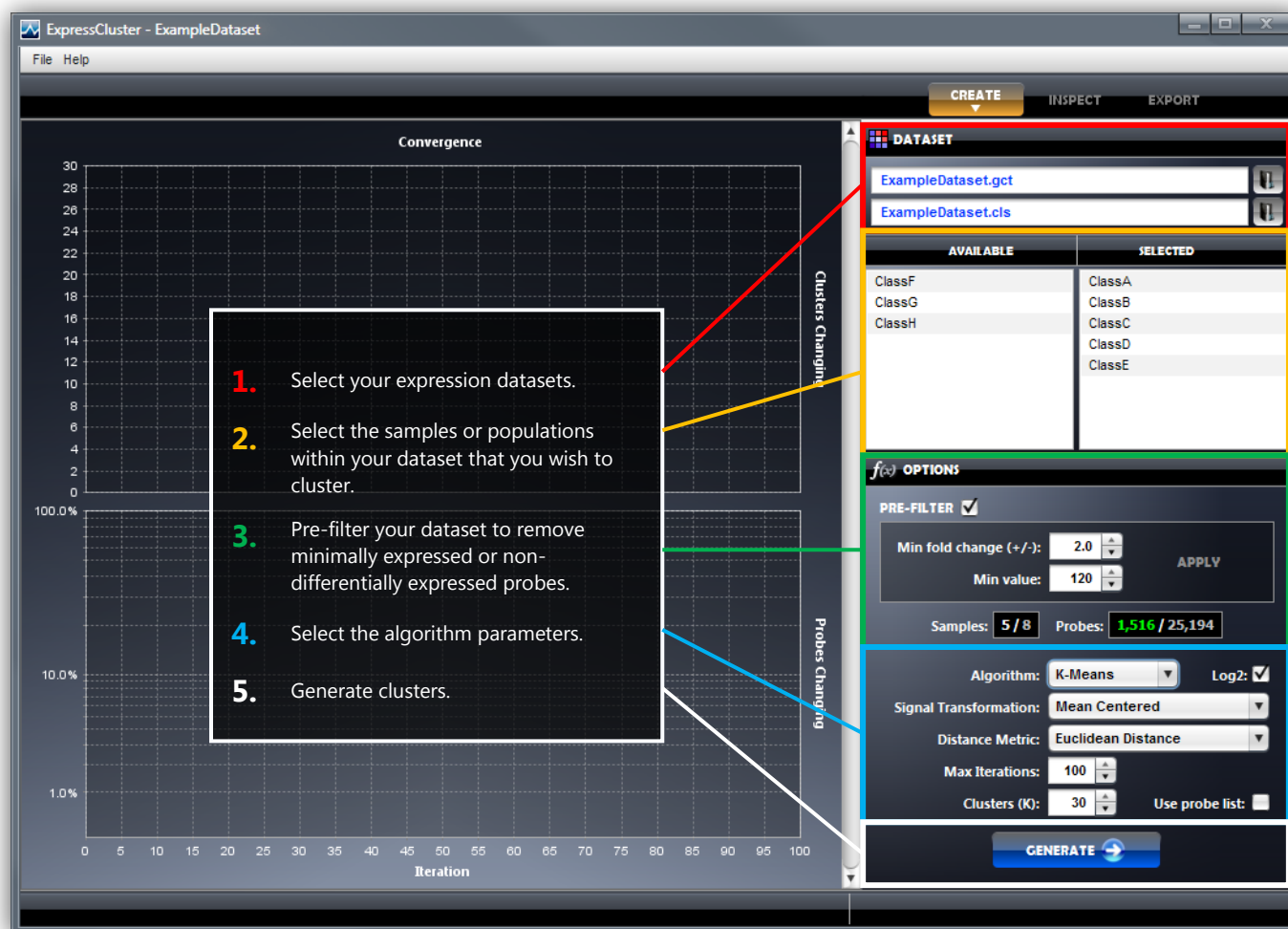
OUTPUT

Execution summary	*.txt	Required	A summary file containing the settings used for the current run.
Cluster expression datasets	*.gct	Optional	A separate GCT file is created for each exported cluster. These can be directly imported into any number of GenePattern modules.
Cluster probeset ID lists	*.txt	Optional	A separate probeset ID list is created for each exported cluster. These can be used, for example, to define highlights/filters in Multiplot or used in other GenePattern modules.
Cluster plots	*.png *.svg *.eps	Optional	Plots of each cluster in either PNG, SVG or EPS format (the latter two are vector based formats which allow for further editing/rescaling when generating publication-quality plots).
Cluster matrix	*.pdf	Optional	A single, multi-page PDF containing a matrix of the selected clusters.
Export archive	*.zip	Optional	A single ZIP file containing all of the above output files.
Cluster heatmaps	*.png *.svg	Optional	A heatmap of the expression intensity for a particular cluster.
Convergence plot	*.png *.svg	Optional	Plot illustrating the rate of convergence of the clustering algorithm.

KEY FEATURES

- Drag & drop samples/classes within a dataset to rearrange their ordering.
- Pre-filter probesets based on minimum differentiation and/or minimum expression.
- Numerous options for distance metric and signal transformation.
- Real-time convergence charts: see how quickly clusters have converged (useful for identifying over-/under-fitting and determining the most optimal starting parameters).
- Specify desired centroids (in addition to random) via probe ID list.
- Sort created clusters using a number of different criteria: size, name, mean correlation to centroid, mean variance, or based on how similar they are to a particular cluster.
- Merge or split (deterministically) clusters.
- Interactive highlighting of specific probes within a cluster based on user-selection or by searching for probe ID/description.
- Relative/global scaling of expression profiles.
- Dynamic heatmaps for each cluster. Color scale reflects either the relative or global transformed signal intensity, or the original expression values as found in the original dataset. Several pre-defined color schemes to choose from.
- Search for genes within other clusters, or find information on a gene online by its annotation or probe ID.
- Copy any cluster plot or heatmap to the clipboard, or save to disk.
- Batch export cluster plots, probe ID lists, GCTs, or a PDF containing a matrix of plots for the selected clusters.

GETTING STARTED – In 5 easy steps



GETTING STARTED – In detail

1. SELECT DATASETS	ExpressCluster requires a GenePattern expression dataset (*.gct) as the primary input. If the dataset contains multiple replicates for a population/class, then the user may also want to specify an accompanying class file (*.cls) which allows ExpressCluster to group replicates together in order to calculate and cluster on the class means. This is often preferred as it results in cleaner clusters.
2. SELECT SAMPLES	Here, there are two lists named "Available" and "Selected". Depending on whether or not a CLS file has been given, these will contain either the samples as they appear in the GCT or the names of the classes as they appear in the CLS. The "Available" column contains all the elements found in the dataset that have not yet been selected. When the user highlights one or more items in this list, a green ➡ icon will appear allowing them to be selected for clustering. Items in "Selected" list can be reordered by drag & drop. Highlighting any item in the "Selected" column will invoke a red ➡ icon – clicking this icon removes that sample from the list and puts it back in the "Available" column.
3. PRE-FILTER	<p>It is generally advisable to pre-filter your data in order to improve clustering results by removing unexpressed and/or uninformative probes. ExpressCluster provides two ways of doing this; by optionally removing probes that do not significantly change between the items being clustered (i.e. they are uninformative), and by removing probes whose maximum expression value never exceeds some threshold (i.e. they are not expressed).</p> <p>Any filtering is carried out prior to actually generating clusters so that the user knows beforehand how many probes will be included. If the "Apply" button is enabled, it means the filter needs to be applied before the user can continue.</p> <p>Min fold change: Removes uninformative probes - fold change between the highest and lowest values <i>across all items being clustered</i> must be \geq this value.</p> <p>Min value: Removes unexpressed probes - must be \geq this value in at least one class.</p>
4. PARAMETERS	<p>This is the step with the most number of options and the one where the user may spend the most amount of time making adjustments. Nevertheless, the default parameters should be appropriate for the vast majority of scenarios, and usually only the value for the number of clusters (K) will need to be changed.</p> <p>NOTE: One of the goals of ExpressCluster was to provide one of the most robust, highly-advanced, yet still user-friendly frameworks for clustering high-dimensional data. While primarily intended for microarray data, it is actually a powerful tool for clustering data from any source. Accordingly, in addition to the typical parametric choices, the user will find a number of other, more advanced options. These options exist for people who fully understand their implications. Therefore, as a general rule: <i>If you do not understand what a particular parameter choice means, you should not select it!</i></p> <p>Algorithm: This is the algorithm used to perform the actual clustering. The options for this are:</p> <ul style="list-style-type: none">▶ K-Means - default▶ K-Means++ <p>K-means: Random probes in a dataset are chosen as the <i>initial</i> cluster centers (called centroids). Every probe is then assigned to the cluster whose centroid it is closest to (according to the chosen distance metric). After assigning all probes to a cluster, each cluster's centroid is recalculated as the mean of its members. Based on these new centroids, probes are then assigned to a cluster again. The first few iterations will see many probes being reassigned to new clusters, but this number decreases with each subsequent iteration. Ultimately, no more clusters will change and the algorithm will have settled into a locally optimal solution (unless it reaches the maximum number of iterations first, in which case it stops where it is). The plots on the main page will chart this convergence in real-time.</p> <p>K-means++ differs from the normal algorithm only in the way the initial centroids are chosen. This method uses weighted probabilistic seeding to maximize how different each centroid is, which frequently results in a more globally optimal solution. The initial seeding takes longer (exponentially with K), but convergence is almost always quicker.</p> <p>NOTE: Because the initial choice of centroids is random, two independent runs of K-means will yield slightly different results.</p>

4. PARAMETERS (cont...)

Signal Transformation: Specifies how expression profiles should be scaled in order to bring them into a more directly comparable state. There are numerous choices for this parameter, but for most expression data, simply logging and Mean Centering the data should suffice. The full list of choices are:

- Mean Centered - **default**
- L_2 Norm (Euclidean)
- L_p Norm (Minkowski)
- Z-Norm (mean=0, var=1)
- Min Max (range = [0,1])
- Sigmoid (adaptive)
- Sigmoid (/w contrast)
- None

For technical details, see [Computational Methods](#).

Distance Metric: This is the calculation used to determine how similar or (dissimilar) any two probes are according to their (transformed) expression profiles. The choice for distance metric can significantly affect clustering results. For this parameter, there are five options:

- Euclidean Distance - **default**
- Pearson's Correlation
- Rank Correlation
- Cosine Θ Similarity
- Minkowski Distance
- Manhattan Distance
- Canberra Distance

For this parameter, it is recommended that most users stick to either Euclidean, Correlation or Cosine Θ . The basic difference between the first two is that Euclidean tends to place more of an emphasis on the magnitude of a signal at each stage, whereas Correlation places more of an emphasis on how similar the pattern is with respect to the direction of change at each stage. Correlation does a better job of grouping together probes that are all changing in the same direction at the same time, but it also requires much more rigorous pre-filtering because it will cluster together probes that are changing significantly with those that change very little, so long as their patterns are similar enough (even probes with a nearly flat-line expression profile still technically have a pattern). Depending on the signal transformation being used, the Cosine Θ metric is usually somewhere between Euclidean and Correlation (i.e. mostly focused on pattern of change, but more sensitive to magnitude than Correlation is). For technical details, see [Computational Methods](#).

Log2? Whether or not to log-transform the values before doing any calculations. This should always be set to **true** for expression data that is on an exponential scale (which is usually the case).

Max Iterations: The maximum number of iterations allowed before terminating. K-means is an iterative algorithm that "settles" into a locally optimal solution. It is generally advisable to allow the algorithm settle into this solution rather than stopping it prematurely. That being said, the default value of 100 represents an *ideal* upper limit for most scenarios, and if an optimal solution is not found by then, it may be an indication that the number of clusters is too high or that sufficient pre-filtering has not been applied (see [Optimizing Parameters](#) for more details).

Clusters (K): The number of clusters to generate. **There is no universally correct value for this parameter!** The optimal number of clusters that one's data should be divided into is dependent on a number of factors, the most relevant being:

- Number of probes being clustered.
- Number of samples/classes being clustered.
- The amount of variability inherent within the data.

(see [Optimizing Parameters](#) for more details).

4. PARAMETERS (cont...)

Use probe list? If this is selected, the user will be required to specify a probe ID list (*.txt) to use as fixed centroids in addition to the random centroids generated by the algorithm. This can be useful when one has a few genes of interest for which they would like to identify any other genes that are similarly co-expressed. Clusters based on given probes will be named after the probe ID instead of the usual systematic convention "Cluster ###".

The probe ID list should be a standard txt file containing 1 probe ID per line. This is best created in Notepad or Excel (save as "tab-delimited text").

NOTE: It is recommended that any probe IDs present in this list have uniquely different expression profiles. Putting multiple probes with similar profiles in this list will only result in redundant clusters being generated that are too similar in nature.

OPTIMIZING PARAMETERS

What do good results look like? The overall quality of the clusters generated is determined by how optimal the final solution is, which depends on several factors. One of the most important factors is the value for K (the number of clusters). It is possible for this value to be either too high, or too low. Setting K too high will result in *over-fitting* the data. Setting K too low will result in *under-fitting* the data. Fortunately, both types of "error" have properties that can often be visually identified in the final result, which makes fine-tuning this particular parameter a little easier.

Over-fitting:	K is too high. There are many more clusters than there are unique patterns in your data to fill them. The result is that several clusters end up looking nearly identical – very similar patterns are forcibly separated into different clusters when they would otherwise have been clustered together with a lower value for K.
Under-fitting:	K is too low. There are many unique, salient patterns within your data, but not enough clusters for them to have their own space. The result is that very dissimilar patterns are forced to share the same cluster.

Pre-filtering: When a microarray dataset is not properly pre-filtered, there will always be too many "junk" probes for good clustering. These are probes whose expression profiles are totally uninformative – meaning that they are either flat-lined, or do not express enough to be considered true signal. The inclusion of such probes yields poor results, so they should always be filtered out prior to clustering. Conveniently, ExpressCluster automatically sorts newly-created clusters by the variance of their centroid. This means that the most variable clusters are listed first, while flat-line clusters are listed last. Accordingly, one easy method of quickly identifying whether or not you've applied sufficient pre-filtering is to simply scroll to the bottom of the list to see if you have many clusters with flat profiles. If you do, then you may want to go back and modify your filtering thresholds.

Number of classes: The number of classes being clustered is a very important variable because the total number of possible patterns increases exponentially with the dimensionality of the input vector (expression profile). For 4-8 classes, an optimal value for K might be anywhere from 16 to 120, depending on how much unique variability there is in the data. For very large datasets many classes, it's not unheard of to set K as high as 400.

Signal transformation: The options for this parameter are described in greater detail in the [Computational Methods](#) section, but there are a couple things to note with respect to achieving optimal results. Specifically, there are two options for this particular parameter which can drastically change a probe's input pattern: **Min|Max** and **Z-Norm**. Both will stretch out all input patterns to the same extent, meaning flat-line profiles will end up having the same variance or range of expression as profiles that genuinely change a lot. This can yield misleading results if proper pre-filtering has not been applied because you will cluster noise together with meaningful patterns. Most microarray data should not be transformed using these particular algorithms, so only choose these if you are clustering other types of data, and be sure that you understand how these particular transformations will affect the input signals.

ANALYSING RESULTS

After you have generated your clusters, the program will switch to the INSPECT tab and automatically select the first cluster in the list. You will also notice that these clusters will be sorted by how variable their members' profiles are – that is, by the variance of the centroid, putting the least eventful clusters at the bottom. Selecting a cluster in the list will show that cluster in the inspection window on the right and list all of its members in the table below it. Individual probes within the cluster can be selected in two ways: 1) by highlighting one or more rows in the table; 2) clicking on the probe's profile in the inspection window (hold ctrl to select multiple probes). Additional features of the interface are annotated below:

The screenshot shows the ExpressCluster software interface with the following annotations:

- Show / hide cluster names or centroids:** Points to the 'Names' and 'Centroids' checkboxes in the top left.
- Search by probe ID or description:** Points to the search bar in the top center.
- Double-click to rename a cluster:** Points to the 'Cluster 8' label in the top right.
- Right-click to show options for merging & splitting clusters:** Points to a right-click context menu in the inspection window.
- Right-click a selected probe to show options searching online:** Points to a right-click context menu in the probe table.
- Choose relative or global scaling (maximize each cluster or use one range for all):** Points to the 'Y Axis: Global' dropdown at the bottom left.
- Change number of clusters per row, and the relative aspect ration:** Points to the 'Layout: 5' and '0.75' controls at the bottom left.
- Sort clusters by:**
 - Centroid Variance
 - Cluster Name
 - Mean Correlation
 - Number of Probes
 - Natural Order
 - Correlation to current
 Points to the 'Sort: Centroid Variance' dropdown at the bottom left.

The inspection window on the right displays 'Cluster 8' with 15 probes. A line plot shows the profile of probe 'Rc1tb2'. Below the plot is a table of probes with their IDs, descriptions, and correlations.

Probe ID	Description	Correlation
10565627	Aqp11	0.99015
10453062	Atp10	0.97285
10538979		0.99627
10410124		0.99627
10408693		0.99627
10591263		0.99627
10361897		0.99627
10576946		0.99627
10467139	Lipa	0.97825
10416371	Lpar6	0.99074
10416355	Rc1tb2	0.94579
10382		0.9679
10359		0.9008
10362		0.9342
10584		0.9425

At the bottom right, the status bar shows 'Samples: 8 / 8' and 'Probes: 2,092 / 25,194'.

ANALYSING RESULTS – Common functions

Rename clusters: Double-click on a cluster's name in the inspection window to rename it. This is the name that will appear in plot titles and in the names of any output files associated with the cluster.

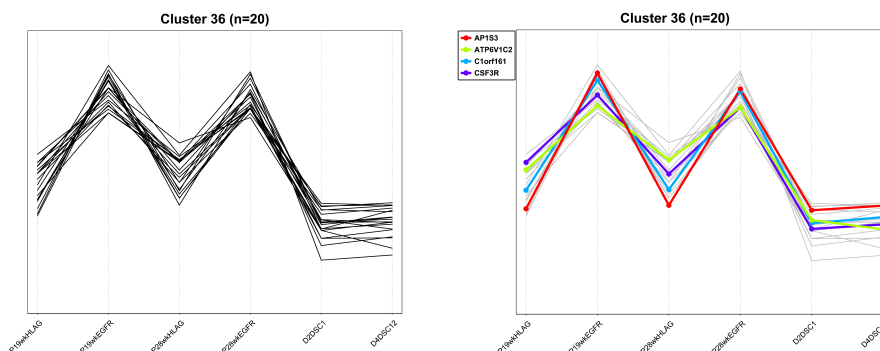
Reorder clusters: You can change the ordering of the clusters by specifying the sort key in the bottom toolbar. The default criterion is centroid variance, which is generally the most useful.

Find mirror image profiles – To find the mirror image of a particular cluster, select it then choose "Correlation to current" as the sort key. This orders the clusters by how correlated they are to the current one, which means that the last cluster in the list will be the one that is most negatively correlated (i.e. the cluster whose profile is closest to the mirror image of the current one).


Searching clusters: You can search for probes within clusters by either their probe ID or by whatever identifier is listed in the "description" column of your input GCT file. For most microarray datasets, the description field will contain Gene Symbol, Refseq, Ensembl or some other universal identifier. As you begin to type, the search field will suggest matching search terms that are currently present in your GCT. If a search term is found within a cluster, the cluster will be highlighted yellow and the matching probe(s) will be selected. If the search term is an ID from the description column, then it's possible for there to be multiple hits for a single search.

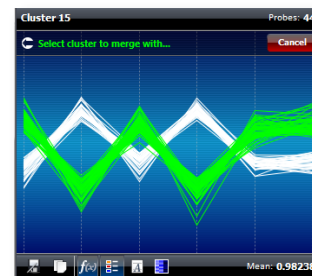
Lookup probes/genes online – Right-clicking on a cluster's probe in the Inspect table will invoke a popup menu with options to search for the current annotation term on [NCBI](#) or search for the current probe ID on [NetAffx](#) (the latter is specific to Affymetrix arrays).

Save/copy plots: You can save (or copy to the system clipboard) any plot by clicking the corresponding button on the toolbar. Saved/copied plots will look slightly different than how they appear in the GUI; they are formatted to make them more suitable for presentations or publications. The background is changed to white, and depending on whether or not the user has highlighted any probes, the lines will be either black or light gray. If probes are highlighted, a legend is added to the left side of the plot and regular probes are made light gray so that the highlighted probes stand out in contrast. The highlight colors of any selected probes will be the same as they appear in the GUI. Below is an example of how the same cluster would appear in copied/saved plots depending on whether probes are highlighted:





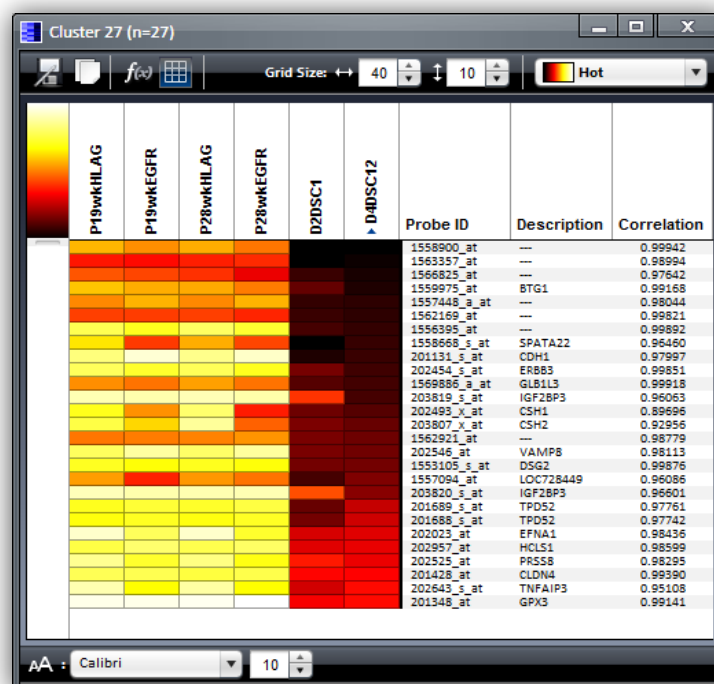
Merge/split clusters: In certain circumstances, users may want to merge or split clusters manually. Unlike the normal behavior of K-means, which chooses centroids randomly, splitting is always deterministic in that it chooses the two most distal probes to be new centroids. This ensures that the two new clusters are as different as possible. When merging two clusters, the user will be asked to select the second cluster, at which point moving the mouse over a potential cluster will overlay its probes on the current plot so that the user can see what the merger would look like.

 ***WARNING:** The clusters generated by K-means represent unbiased, mathematically optimal solutions. Once the user has manually intervened to change clusters, this is no longer true. That fact may be important depending on what conclusions the user is trying to draw from this sort of analysis. Accordingly, any clusters that have been modified will be permanently marked with a red *.



ANALYSING RESULTS – Heatmaps

Invoking:	The user can bring up an interactive heatmap of the current cluster by clicking the  icon. The heatmap window will always contain data for the currently-selected cluster in the main interface – selecting a new cluster will change the contents of this window.
Intensity scale:	<p>There are three possible ways to compute the <i>intensity</i> of the colors shown in this heatmap:</p> <ul style="list-style-type: none"> • Transformed expression value – relative • Transformed expression value – global • Original expression value (logged, if specified) – global <p>“Transformed” refers to the values after signal transformation has been applied. These are the values on which the clusters were computed. The user can toggle between Transformed (default) and Original by clicking the  icon. If original values are being shown, tooltips will display those values in addition to probe ID/description.</p> <p>“Relative” and “global” refer the current setting of the Y-Axis. Relative means that the Y-Axis (and color range) is scaled to the highest and lowest value of each cluster. Global means that the Y-Axis is scaled according to the highest and lowest value of all data being clustered.</p>
Color scale:	The user can quickly choose between 10 pre-defined color scales from the drop down list on the toolbar.
Save/copy:	Save (as PNG/SVG) or copy the current heatmap image via toolbar/popup menu. The produced image will be identical to the current view. The user can also copy the data of a heatmap, by selecting rows of the table and pressing Control-C. The contents can be pasted into Excel or Notepad as tab-delimited text.
Sorting:	Sort heatmaps by the value of any column simply by clicking on the column header. This is particularly useful when using correlation as the distance metric in that sorting by the expression of a primary class has a similar effect to sorting by each probe’s variance.
Coordinated selection:	Selecting any probe(s) in the heatmap will result in those same probes being selected in the primary interface (and vice versa). This allows the user to easily match a probe’s profile with its heatmap representation even though the tables may have different sort orders.
Adjust layout:	Users can also modify other visual aspects of the heatmap, such as the font, column width/height, and whether or not to draw grid lines.



EXPORTING RESULTS

ExpressCluster has a number of different options for exporting the results of a particular analysis. In addition to the following optional export items, a summary file is automatically generated and saved any time data is exported.

Probe ID lists

Creates a separate probe ID list for each of the selected clusters. This is simply a standard TXT file with one ID per line (used in many other GenePattern modules).

Datasets (GCT)

Creates a separate *.gct file for each of the selected clusters. The expression values will be the untransformed values as they appear in the original dataset. There are several options for which samples – or classes, if a *.cls file was specified – to include:

Current Samples: Expression data for the currently clustered samples.

All Samples: Expression data for all samples in the dataset, including those that were not part of the clustering.

Current Classes: Class means data for the currently clustered classes.

All Classes: Class means data for all samples in the dataset, including those that were not part of the clustering.

Individual plots (PNG/SVG/EPS)


A separate plot for each of the selected clusters. Users can specify the output format by clicking the down ▼ arrow. PNG is a common, high-quality image format. SVG and EPS are both vector based and should be used when needing to produce publication-quality images.

Cluster matrix (PDF)

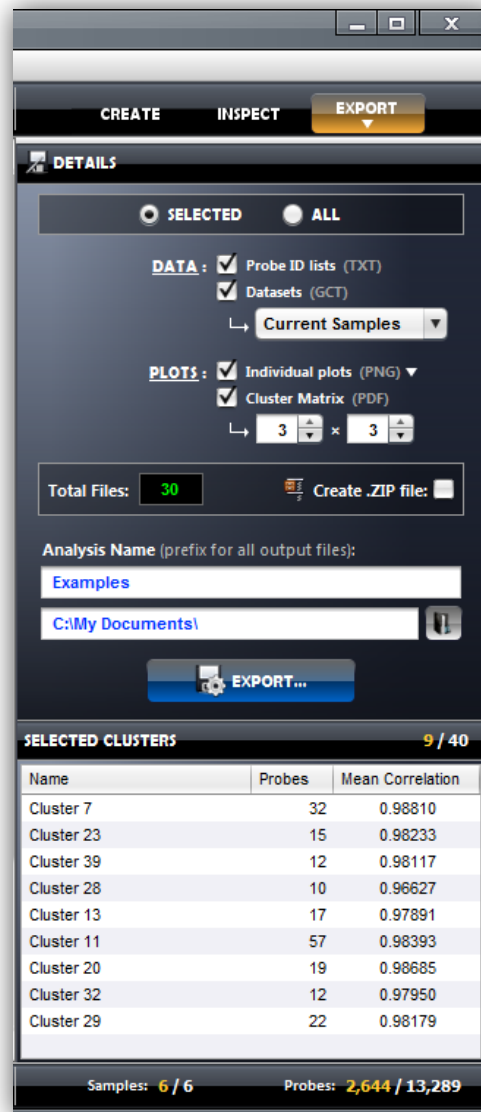
This creates a multi-page PDF file containing a matrix of the selected clusters, similar to how they appear in the program, but with the x-axis annotated. User can specify the number of rows/columns per page.

DATA

PLOTS

The total number of output files created, based on the currently selected clusters and options, is displayed for the user. Sometimes this number can get quite high, at which point a warning will appear to ensure that the user realizes how many files are about to be created. Particularly in these cases, the user may wish to bundle all of the output files into a single ZIP file, by checking the box next to the  icon.

NOTE (Heatmaps): Currently, heatmaps can only be exported one at a time via the heatmap frame (either by clicking the save icon on the toolbar or via context menu). Batch export of heatmaps will be added in a future version.



Name	Probes	Mean Correlation
Cluster 7	32	0.98810
Cluster 23	15	0.98233
Cluster 39	12	0.98117
Cluster 28	10	0.96627
Cluster 13	17	0.97891
Cluster 11	57	0.98393
Cluster 20	19	0.98685
Cluster 32	12	0.97950
Cluster 29	22	0.98179

Samples: 6 / 6 Probes: 2,644 / 13,289

COMPUTATIONAL METHODS – Signal Transformations

Mean Centered	This method simply subtracts the mean (-1) of each row from every value in that row, which results in every vector having the same center. If the user has selected the option to pre-log the data, then this is equivalent to dividing the unlogged values by the geometric row means.	$x_i^* = x_i - \bar{x}$
L_2 Norm (Euclidean)	Divides each component of a vector \mathbf{x} by its L_2 norm (i.e. Euclidean vector length). The result is that each vector has unity length in Euclidean space.	$x_i^* = \frac{x_i}{\ \mathbf{x}\ } \quad \text{where} \quad \ \mathbf{x}\ = \sqrt{\sum_{i=1}^n x_i^2}$
L_p Norm (Minkowski)	Divides each component of a vector \mathbf{x} by its L_p norm (i.e. Minkowski vector length). The result is that each vector has unity length in Minkowski space.	$x_i^* = \frac{x_i}{\ \mathbf{x}\ } \quad \text{where} \quad \ \mathbf{x}\ = \left(\sum_{i=1}^p x_i^p \right)^{1/p}$
Z-Norm (mean=0, var=1)	Transforms each vector to have zero mean and unit variance. The fact that this method normalizes vector variance means that nearly flat-lined profiles will end up having the same variance as significantly changing expression profiles. Consequently, when using this method, background noise will get clustered with meaningful signal unless adequate pre-filtering is applied.	$x_i^* = \frac{x_i - \bar{x}}{\sigma}$
Min Max (range = [0,1])	Transforms each vector to have unit range of exactly [0,1]. The fact that this method normalizes vector range means that nearly flat-lined profiles will end up having the same range as significantly changing expression profiles. Consequently, when using this method, background noise will get clustered with meaningful signal unless adequate pre-filtering is applied.	$x_i^* = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$
Sigmoid (adaptive)	Non-linear, adaptive sigmoidal transformation. Rescales vector components to be between 0 and 1, with a variance that is weighted by the range of expression over the entire dataset. Tends to saturate extreme high and low values while accentuating mid-level differences.	$x_i^* = \frac{(x_i - \bar{x})}{\text{range}(\mathbf{X}) + (x_i - \bar{x})^2}$
Sigmoid (/w contrast)	Non-linear sigmoidal transformation with enhanced contrast. Similar to above, but is additionally weighted by the initial variation of the input vector. Patterns with low variation are suppressed, while those with greater variation are enhanced. The result is an increase in contrast between signal and noise.	$x_i^* = \frac{\sigma(x_i - \bar{x})}{\text{range}(\mathbf{X}) + \sigma(x_i - \bar{x}) ^2}$

NOTE: With the exception of Z-norm, any transformation algorithm that normally results in a vector with 0-mean will, in addition to the stated transformation, shift the final input vector by 1. This does not change the signal pattern in any way, but allows for a more useful application of Cosine Θ similarity as a distance metric, which would otherwise be identical to Correlation if the input vector were centered about 0.

COMPUTATIONAL METHODS – Distance Metrics

Euclidean Distance This is one of the two most commonly used distance metric in data clustering algorithms. It tends to be more affected by the difference in magnitude at each point than to the direction of change. It is defined as the square root of the sum of squared differences between corresponding components of two vectors, and represents a specific case of the Minkowski distance (below), with $p=2$.

$$d_{euclidean}(a, b) = \sqrt{\sum_{i=1}^n |a_i - b_i|^2}$$

Pearson's Correlation Another common distance metric, the Pearson's correlation coefficient best captures how similar to patterns are in their trends, regardless of magnitude. As stated before, this is a very powerful distance metric for clustering, but requires more rigorous pre-filtering of low-variance probes, as they will end up being clustered with high-variance probes if the patterns are similar.

$$corr(a, b) = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{a_i - \bar{a}}{\sigma_a} \right) \left(\frac{b_i - \bar{b}}{\sigma_b} \right)$$

Rank Correlation Spearman's rank correlation coefficient is defined as the Pearson correlation between the ranked variables. Expression values are converted to ranks, and the correlation is computed from these values. A rank correlation of 1 results when the two vectors being compared are monotonically related, even if their relationship is not linear. This is similar to Pearson's correlation, but much less sensitive to outliers.

(see Pearson's correlation)

Cosine Θ Similarity Cosine Θ similarity is a measure of the angle between two N-dimensional vectors (in this case, an expression profile). This is also referred to as the uncentered correlation. When two input vectors both have a mean of 0, this metric is equal to the Pearson's correlation coefficient. For this reason, most of the signal transformations used in ExpressCluster will shift the final vector off of 0 by 1. This allows Cosine similarity to serve as somewhat of a compromise solution between Pearson's correlation and Euclidean distance.

$$d_{cosine}(a, b) = \cos \theta = \frac{a \cdot b}{\|a\| \|b\|}$$

Minkowski Distance The Euclidean distance is actually a specific case of the Minkowski distance, with $p = 2$. The Minkowski distance is identical in form to the former, but the value of p is determined by the dimensionality of the input vectors, a and b . Thus, for a 6 dimensional vector (i.e. clustering 6 samples or classes), $p = 6$.

$$d_{minkowski}(a, b) = \left(\sum_{i=1}^p |a_i - b_i|^p \right)^{1/p}$$

Manhattan Distance This is a special case of the Minkowski distance, where $p=1$.

$$d_{euclidean}(a, b) = \sum_{i=1}^n |a_i - b_i|$$

Canberra Distance The Canberra metric is a weighted version of the classic Manhattan distance. It is often used for data scattered around an origin.

$$d_{canberra}(a, b) = \sum_{i=1}^n \frac{|a_i - b_i|}{|a_i| + |b_i|}$$