

© 2005 Nature Publishing Group http://www.nature.com/naturebiotechnology

How does gene expression clustering work?

Patrik D'haeseleer

Clustering is often one of the first steps in gene expression analysis. How do clustering algorithms work, which ones should we use and what can we expect from them?

Our ability to gather genome-wide expression data has far outstripped the ability of our puny human brains to process the raw data. We can distill the data down to a more comprehensible level by subdividing the genes into a smaller number of categories and then analyzing those. This is where clustering comes in.

The goal of clustering is to subdivide a set of items (in our case, genes) in such a way that similar items fall into the same cluster, whereas dissimilar items fall in different clusters. This brings up two questions: first, how do we decide what is similar; and second, how do we use this to cluster the items? The fact that these two questions can often be answered independently contributes to the bewildering variety of clustering algorithms.

Gene expression clustering allows an open-ended exploration of the data, without getting lost among the thousands of individual genes. Beyond simple visualization, there are also some important computational applications for gene clusters. For example, Tavazoie *et al.*[1] used clustering to identify *cis*-regulatory sequences in the promoters of tightly coexpressed genes. Gene expression clusters also tend to be significantly enriched for specific functional categories—which may be used to infer a functional role for unknown genes in the same cluster.

In this primer, I focus specifically on clustering genes that show similar expression patterns across a number of samples, rather than clustering the samples themselves (or both). I hope to leave you with some understanding of clustering in general and three of the more popular algorithms in particular. Where possible, I also attempt to provide some practical guidelines for applying cluster analysis to your own gene expression data sets.

Patrik D'haeseleer is in the Microbial Systems Division, Biosciences Directorate, Lawrence Livermore National Laboratory, PO Box 808, L-448, Livermore, California 94551, USA.
e-mail: patrikd@llnl.gov

A few important caveats

Before we dig into some of the methods in use for gene expression data, a few words of caution to the reader, practitioner or aspiring algorithm developer:

- It is easy—and tempting—to invent yet another clustering algorithm. There are hundreds of published clustering algorithms, dozens of which have been applied to gene

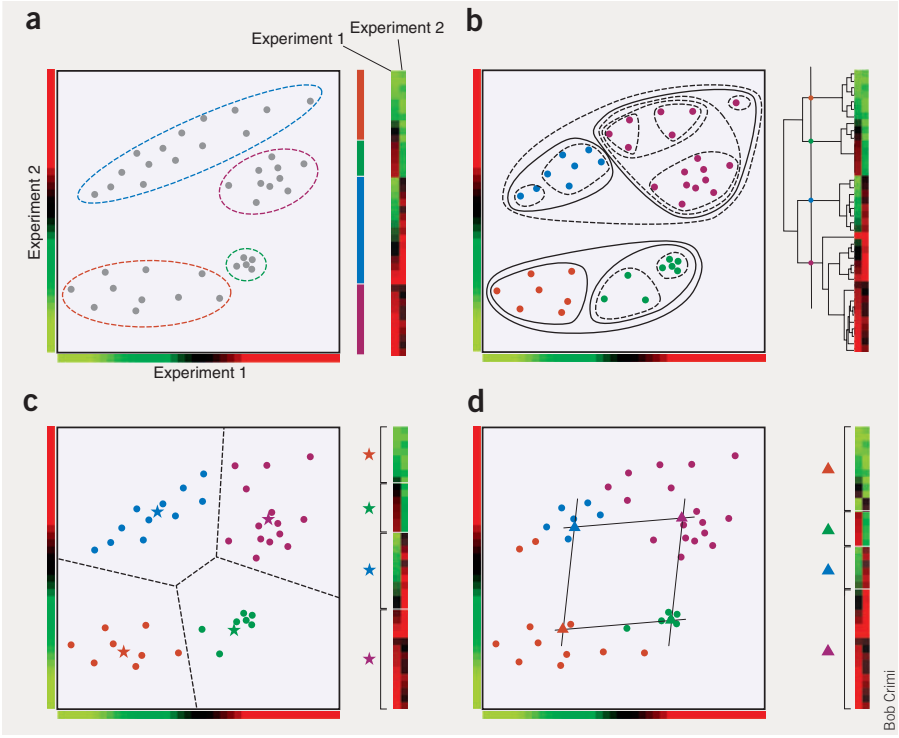


Figure 1 A simple clustering example with 40 genes measured under two different conditions. (**a**) The data set contains four clusters of different sizes, shapes and numbers of genes. Left: each dot represents a gene, plotted against its expression value under the two experimental conditions. Euclidean distance, which corresponds to the straight-line distance between points in this graph, was used for clustering. Right: the standard red-green representation of the data and corresponding cluster identities. (**b**) Hierarchical clustering finds an entire hierarchy of clusters. The tree was cut at the level indicated to yield four clusters. Some of the superclusters and subclusters are illustrated on the left. (**c**) *k*-means (with $k = 4$) partitions the space into four subspaces, depending on which of the four cluster centroids (stars) is closest. (**d**) SOM finds clusters, which are organized into a grid structure (in this case a simple 2×2 grid).

Table 1 Gene expression similarity measures

Manhattan distance (city-block distance, L1 norm)	$d_{fg} = \sum_c \lvert e_{fc} - e_{gc} \rvert$
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (\mathbf{e}_f - \mathbf{e}_g)' \Sigma^{-1} (\mathbf{e}_f - \mathbf{e}_g)$, where $\boldsymbol{\Sigma}$ is the (full or within-cluster) covariance matrix of the data
Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (angular separation, cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}^2 \sum_c e_{gc}^2}}$
Spellman rank correlation	As Pearson correlation, but replace e_{gc} with the rank of e_{gc} within the expression values of gene g across all conditions $c = 1 \ldots C$
Absolute or squared correlation	$d_{fg} = 1 - \lvert r_{fg} \rvert$ or $d_{fg} = 1 - r_{fg}^2$

d_{fg}, distance between expression patterns for genes f and g. e_{gc}, expression level of gene g under condition c.

© 2005 Nature Publishing Group http://www.nature.com/naturebiotechnology

expression data. It is much harder to do a fair evaluation of how well a new algorithm will perform on typical expression data sets, how it compares with those dozens of other published algorithms and under which circumstances one algorithm should be preferred over another.

- There is no one-size-fits-all solution to clustering, or even a consensus of what a 'good' clustering should look like. In the words of Jain and Dubes[2]: "There is no single best criterion for obtaining a partition because no precise and workable definition of 'cluster' exists. Clusters can be of any arbitrary shapes and sizes in a multidimensional pattern space. Each clustering criterion imposes a certain structure on the data, and if the data happen to conform to the requirements of a particular criterion, the true clusters are recovered."

In other words, each algorithm imposes its own set of biases on the clusters it constructs, and whereas most sensible clustering algorithms may yield similar results on trivial test problems, in practice they can give widely differing results on messy real-world gene expression data.

Similarity measures

So, how do we decide how similar the expression patterns of two genes are? Note that this really boils down to which types of expression patterns we would like to see fall into the same clusters —something that may go well beyond which patterns look visually similar and is directly related to the question 'what do we want to achieve by clustering?'

Two of the easiest and most commonly used similarity measures for gene expression data are Euclidean distance and Pearson correlation coefficient. Note that Euclidean distance is sensitive to scaling and differences in average expression level, whereas correlation is not. See **Table 1** for these and other similarity measures and variants.

A brief intro to clustering methods

The two most important classes of clustering methods are hierarchical clustering and partitioning (**Fig. 1**). In hierarchical clustering, each cluster is subdivided into smaller clusters, forming a tree-shaped data structure or dendrogram. Agglomerative hierarchical clustering (also used in phylogenetics) starts with the single-gene clusters and successively joins the closest clusters until all genes have been joined into the supercluster. In fact, there is a whole family of clustering methods, differing only in the way intercluster distance is defined (the 'linkage function'). Some of the more common ones are single linkage (the distance between clusters is the shortest distance between any two members of the cluster), complete linkage (largest distance between any two members), average linkage/UPGMA (unweighted pair-group method using arithmetic averages; average distance between any two members) and centroid linkage/UPGMC (unweighted pair-group method using centroids; distance between the cluster centroids).

Partitioning methods, on the other hand, subdivide the data into a typically predetermined number of subsets, without any implied hierarchical relationship between these clusters. How many clusters are actually present in the data is a thorny issue. A common approach is to rerun the clustering with different numbers of clusters, in the hopes of being able to distinguish the optimal number of clusters. A hierarchical clustering can also be reduced to a partitioning, by cutting the dendrogram at a given level (**Fig. 1b**).

If you're interested in delving deeper, see Jain and Dubes[2] for a comprehensive overview of clustering algorithms or Aldenderfer and Blashfield[3] for a more concise treatment. Jiang *et al.*[4] provide a survey of methods used specifically for gene expression data.

Three popular clustering methods

Eisen *et al.*[5] applied hierarchical clustering (using uncentered correlation distance and centroid linkage) to analyze some of the first yeast microarray data sets. Because of the early availability of free clustering and visualization software, this is probably the single most often used clustering algorithm—to the extent of being dubbed 'Eisen clustering' by some. For visualization, the resulting dendrogram can be sorted according to a predefined criterion (e.g., average expression), and the dendrogram and the gene expression heat map are displayed side by side (**Fig. 1b**).

Another popular clustering method is *k*-means, a partitioning method, that subdivides the genes into a predetermined number (*k*) of

© 2005 Nature Publishing Group http://www.nature.com/naturebiotechnology

clusters[1]. The algorithm is initialized with k randomly chosen cluster centroids, and each gene is assigned to the cluster with the closest centroid (**Fig. 1c**). Next, the centroids are reset to the average of the genes in each cluster. This process is continued until no more genes change cluster. Different initial centroid positions may yield different cluster results, and it is important to run the algorithm several times with different random seeds.

The self organizing map (SOM) method[6] also starts with a predetermined number of cluster centroids, except here the centroids are linked in a grid structure. At each iteration, a gene is chosen, and the closest centroid is moved toward the gene—as well as its neighboring centroids on the grid. The grid of centroids initially behaves like a flexible sheet stretching across expression space, but as the radius of this neighborhood gradually shrinks over time, each centroid will focus more and more on its own cluster. The end result is a grid of clusters, in which neighboring clusters show related expression patterns (**Fig. 1d**).

How good is my clustering?

The quality of a clustering result can be evaluated based on internal criteria (that is, based on various statistical properties of the clusters) or external criteria (that is, based on additional information that was not used in the clustering process itself).

Internal validation seems straightforward: we would like clusters to be compact and well separated. Unfortunately, this reverts back to the question 'what would we like clusters to look like'? At least a dozen different measures have been developed to test the quality of a cluster, and for many of these there exists a clustering method that will optimize that measure. For example, k-means optimizes the variance of the clusters, whereas complete linkage minimizes the radius of the clusters. Other measures test the within-group versus between-group variance, the separation between clusters and the stability of clusters with respect to noise, random initializations (such as for k-means or SOM) and leaving out conditions for example (see Handl *et al.*[7] for an overview of internal validation measures and their biases).

Ultimately, the real test of the pudding is in the eating, not just in what the pudding looks like. The most reliable quality measure of a clustering method is how well it actually performs the task at hand. For example, if our goal is to cluster together genes with similar function, then we can use existing functional annotations to verify how well that goal has been achieved[8,9]. If our goal is to extract *cis*-regulatory elements from the clusters, then we can check how well genes with known regulatory sequences are clustered together.

Some guidelines for gene expression clustering

Although much more work needs to be done to compare the performance of clustering algorithms on real expression data, some general trends are emerging from the few comparative studies available.

- Single-linkage clustering performs abysmally on most real-world data sets, and gene expression data is no exception[7–9]. It is included in almost every single clustering package 'for completeness,' but should probably be removed altogether.

- Contrary to conventional expectation, complete linkage seems to outperform average linkage[8].

- k-means and SOM outperform hierarchical clustering[8–10]. In particular, this means that the traditional 'Eisen clustering', despite having become a *de facto* standard for visualization of expression data, is likely to be a poor choice for further computational analysis of the resulting clusters.

- There are some indications that SOM may outperform k-means[8,11], especially for larger numbers of clusters[8].

- Euclidean distance, Pearson correlation and 'uncentered' correlation (angular separation) all seem to work reasonably well as distance measures[8,9]. Euclidean distance may be more appropriate for log ratio data, whereas Pearson correlation seems to work better for absolute-valued (e.g., Affymetrix) data[8].

More exhaustive comparisons will be needed to verify the usefulness of these approaches on different types of data sets. In addition, preprocessing details, such as normalization of the data, filtering of unreliable data points and how missing values are dealt with (simply setting them to zero is almost never a good idea) can also affect the clustering outcome.

For now, the best advice is to use more than one clustering algorithm. Methods that may give different results based on initial conditions should be rerun multiple times to find the best possible solution and to check for stability. Don't forget to compare the results with those on randomized data, because just as the human mind can see patterns in random noise, most clustering methods will find clusters even when no actual structure is present!

The use of these very basic gene expression analysis tools is still surprisingly poorly formalized. Perhaps over the next few years we will see a set of best practices emerge for expression clustering. But no doubt there will always be room to hunt for those unexpected patterns that might show up if only you look at the data from just the right angle.

Further study

Try out some clustering algorithms online at http://ep.ebi.ac.uk/EP/EPCLUST/ or http://gepas.bioinfo.cnio.es/

1. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J. & Church, G.M. Systematic determination of genetic network architecture. *Nat. Genet.* **22**, 281–285 (1999).
2. Jain, A.K. & Dubes, R.C. *Algorithms for Clustering Data.* (Prentice Hall, Englewood Cliffs, New Jersey, 1988).
3. Aldenderfer, M.S. & Blashfield, R.K. Cluster Analysis. (Sage Publication, Newbury Park, California, 1984).
4. Jiang, D., Tang, C. & Zhang, A. Cluster analysis for gene expression data: a survey. *IEEE Trans. Know. Data Eng.* **16**, 1370–1386 (2004).
5. Eisen, M.B., Spellman, P.T., Brown, P.O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* **95**, 14863–14868 (1998).
6. Tamayo, P. *et al.* Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA* **96**, 2907–2912 (1999).
7. Handl, J., Knowles, J. & Kell, D.B. Computational cluster validation in post-genomic data analysis. *Bioinformatics* **21**, 3201–3212 (2005).
8. Gibbons, F.D. & Roth, F.P. Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res.* **12**, 1574–1581 (2002).
9. Costa, I.G., de Carvalho, F.A. & de Souto, M.C. Comparative analysis of clustering methods for gene expression time course data. *Genet. Mol. Biol.* **27**, 623–631 (2004).
10. Datta, S. & Datta, S. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics* **19**, 459–466 (2003).
11. Gat-Viks, I., Sharan, R. & Shamir, R. Scoring clustering solutions by their biological relevance. *Bioinformatics* **19**, 2381–2389 (2003).