COMPUTATIONAL ANALYSIS OF MICROARRAY DATA

John Quackenbush

Microarray experiments are providing unprecedented quantities of genome-wide data on gene-expression patterns. Although this technique has been enthusiastically developed and applied in many biological contexts, the management and analysis of the millions of data points that result from these experiments has received less attention. Sophisticated computational tools are available, but the methods that are used to analyse the data can have a profound influence on the interpretation of the results. A basic understanding of these computational tools is therefore required for optimal experimental design and meaningful data analysis.

COMPUTATIONAL GENETICS

The advent of the genome project has vastly increased our knowledge of the genomic sequences of humans and other organisms, as well as the genes that they encode. Various techniques have been developed to exploit this growing body of data, including serial analysis of gene expression (SAGE)[1], oligonucleotide arrays[2] and cDNA microarrays[3,4], that provide rapid, parallel surveys of gene-expression patterns for hundreds or thousands of genes in a single assay. These transcriptional profiling techniques promise a wealth of data that can be used to develop a more complete understanding of gene function, regulation and interactions.

The most powerful applications of transcriptional profiling involve the study of patterns of gene expression across many experiments that survey a wide array of cellular responses, phenotypes and conditions. The simplest way to identify genes of potential interest through several related experiments is to search for those that are consistently either up- or downregulated. To that end, a simple statistical analysis of gene-expression levels will suffice. However, identifying patterns of gene expression and grouping genes into expression classes might provide much greater insight into their biological function and relevance. Several techniques have been used for the analysis of gene-expression data, including hierarchical clustering[5–9], mutual information[5,10] and self-organizing maps (SOMs)[11,12].

The implementation of a successful programme of expression analysis requires the development of various laboratory protocols, as well as the development of database and software tools for efficient data collection and analysis. Although detailed laboratory protocols have been published[13,14], the computational tools necessary to analyse the data are rapidly evolving and no clear consensus exists as to the best method for revealing patterns of gene expression. Indeed, it is becoming increasingly clear that there might never be a 'best' approach and that the application of various techniques will allow different aspects of the data to be explored. Furthermore, without a more complete understanding of the underlying biology, particularly of gene regulation, there might never be a single technique that will allow us to find all the relationships in the data. Consequently, choosing the appropriate algorithms for analysis is a crucial element of the experimental design. The purpose of this review is to provide a general overview of some existing computational approaches. This review is not comprehensive, as new, more sophisticated techniques are rapidly being developed, but instead represents a tutorial on some of the more basic tools. Although the focus here is on spotted DNA microarrays, the techniques described are generally applicable to expression data generated using oligonucleotide arrays, Affymetrix GeneChips^{TM}, or SAGE, provided the data is presented in an appropriate format.

The Institute for Genomic Research, 9,712 Medical Center Drive, Rockville, Maryland 20850, USA.
e-mail: johnq@tigr.org

Box 1 | Normalization

There are three widely used techniques that can be used to normalize gene-expression data from a single array hybridization. All of these assume that all (or most) of the genes in the array, some subset of genes, or a set of exogenous controls that have been 'spiked' into the RNA before labelling, should have an average expression ratio equal to one. The normalization factor is then used to adjust the data to compensate for experimental variability and to 'balance' the fluorescence signals from the two samples being compared.

Total intensity normalization

Total intensity normalization data relies on the assumption that the quantity of initial mRNA is the same for both labelled samples. Furthermore, one assumes that some genes are upregulated in the query sample relative to the control and that others are downregulated. For the hundreds or thousands of genes in the array, these changes should balance out so that the total quantity of RNA hybridizing to the array from each sample is the same. Consequently, the total integrated intensity computed for all the elements in the array should be the same in both the Cy3 and Cy5 channels. Under this assumption, a normalization factor can be calculated and used to re-scale the intensity for each gene in the array.

Normalization using regression techniques

For mRNA derived from closely related samples, a significant fraction of the assayed genes would be expected to be expressed at similar levels. In a scatterplot of Cy5 versus Cy3 intensities (or their logarithms), these genes would cluster along a straight line, the slope of which would be one if the labelling and detection efficiencies were the same for both samples. Normalization of these data is equivalent to calculating the best-fit slope using regression techniques[27] and adjusting the intensities so that the calculated slope is one. In many experiments, the intensities are nonlinear, and local regression techniques are more suitable, such as LOWESS (LOcally WEighted Scatterplot Smoothing) regression[29].

Normalization using ratio statistics

A third normalization option is a method based on the ratio statistics described by Chen *et al.*[20]. They assume that although individual genes might be up- or downregulated, in closely related cells, the total quantity of RNA produced is approximately the same for essential genes, such as 'housekeeping genes'. Using this assumption, they develop an approximate probability density for the ratio $T_k = R_k/G_k$ (where R_k and G_k are, respectively, the measured red and green intensities for the *k*th array element). They then describe how this can be used in an iterative process that normalizes the mean expression ratio to one and calculates confidence limits that can be used to identify differentially expressed genes.

Selecting the array probes

The first step in any microarray assay is starting with a well-characterized and annotated set of hybridization probes (the sequences that are arranged on the microarray). For prokaryotes and simple eukaryotes, such as yeast, this is most easily accomplished by designing PCR primers to amplify gene-specific probes directly from genomic DNA. In most eukaryotic genomes, the large number of genes, the existence of introns and the lack of a complete genome sequence, makes direct amplification impractical. In these species, the EST data collected in the public DNA sequence databases are a valuable representation of the transcribed portion of the genome, and the cDNA clones from which the ESTs are derived have become the primary reagents for expression analysis.

However, clone selection is a significant challenge; there are more than three million human ESTs in the dbEST database, from which a single representative clone needs to be selected for each gene included in the array. There are several publicly available analyses of human ESTs, including UniGene[15] and TIGR Gene Indices (TGI)[16] (TIGR is The Insititute for Genomic Research), the STACK database[17] and the Database of Transcribed Sequences (DoTS) (see links box). Each database attempts to group ESTs from the same gene and to provide a common annotation. Although the precise approaches taken by the databases vary, they all generally provide high-quality annotation for the cDNAs represented in the public databases. Regardless of which resource is used to select clones, the cDNAs being arrayed should have their sequences verified to validate the clone identities. The consensus emerging in the community is that stable descriptors, either accession numbers or DNA sequence, should be used as the primary identifier for arrayed cDNA clones.

For other array-based assays, such as pre-spotted filter arrays and Affymetrix GeneChips™ (REF. 2), the researcher has little, if any, control over the probe content of the chip. These arrays do have the advantage of providing a platform that can be used to more easily compare results between laboratories. However, at least in the case of Affymetrix GeneChips™, the lack of precise knowledge of the probe sequences used to represent each gene forces users to rely on the annotation provided by the manufacturer.

After clone selection, amplification and purification, the probes are loaded in microtiter plates into an arraying robot and are mechanically spotted onto chemically modified glass slides. The robotic arrayers provide a reproducible and precise mathematical map from spots on the arrays to wells in the microtiter plates, and therefore to the cDNA clones and the genes that they represent.

One important element for assuring data quality, and maximizing opportunities for comparison of expression data between experiments, is the establishment and use of an integrated laboratory information management system (LIMS) database to track all aspects of the process. Early in the process, various data must be tracked, including clone selection, slide information and scanner settings. Although many arraying labs have developed their own internal relational databases, there are several commercially available products, as well as published prototypes, available from the National Center for Biotechnology Information (NCBI)[18] and Stanford[19] (see links box at end).

Data collection and normalization

Once a collection of microarray slides is printed, each slide represents a potential experiment. The arrayed genes are probes that can be used to query pooled, differentially labelled targets derived from RNA samples from different cellular phenotypes to determine the relative expression levels of each gene. The two RNA samples from the tissues of interest are typically used to generate first-strand cDNA targets labelled with the fluorescent dyes Cy3 and Cy5. These are then purified, pooled and hybridized to the arrays. After hybridization, slides are scanned and independent images for the control and query channels are generated. These images must then be analysed to identify the arrayed spots and to measure the relative fluorescence intensities for each element.

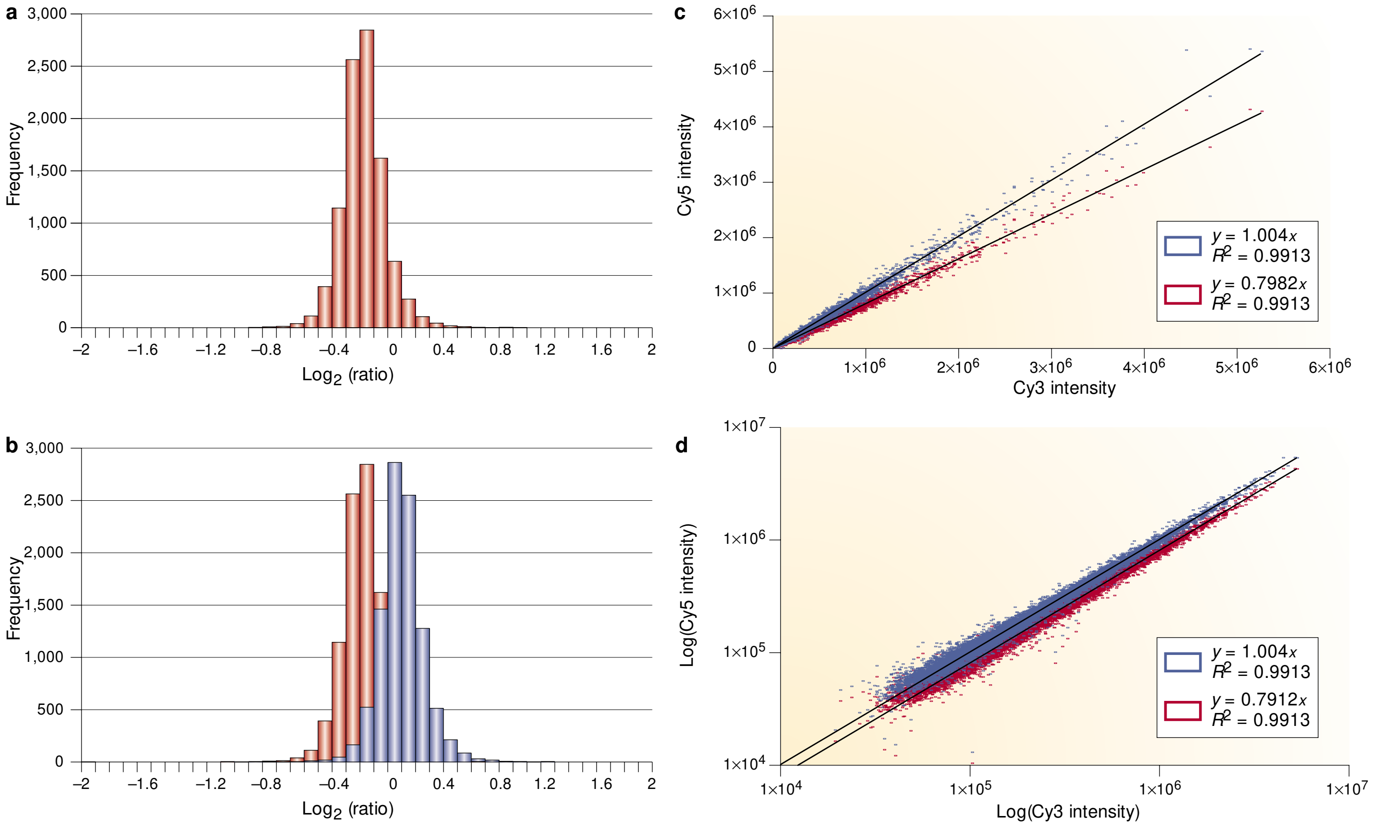


Figure 1 | **Data normalization. a** | A histogram representing the distribution of $\log_2$(ratio) values for a 'self–self' hybridization, in which the measured Cy5 intensity is generally less than the measured Cy3 intensity. Consequently, the $\log_2$(ratio) histogram is centred to the left of zero (as are, indeed, the vast majority of the data). **b** | The same data set shown before (red) and after (blue) normalization to illustrate how the data are transformed. The normalized distribution, shown in blue, is shifted and centred about zero. The perceived change in the shape of the distribution is an artefact of the process of placing data into 'bins' when making the histogram. Scatter plots before (red) and after (blue) normalization of the **c** | measured intensities and **d** | log(intensities) also illustrate the transformation of the data.

Most commercially available microarray scanner manufacturers provide software that handles image processing; there are several additional image-processing packages available (see links box at end).

After image processing, it is necessary to normalize the relative fluorescence intensities in each of the two scanned channels. Normalization adjusts for differences in labelling and detection efficiencies for the fluorescent labels and for differences in the quantity of initial RNA from the two samples examined in the assay. These problems can cause a shift in the average ratio of Cy5 to Cy3 and the intensities must be rescaled before an experiment can be properly analysed. Three normalization algorithms are described in BOX 1. There are several sophisticated, nonlinear approaches to normalization that correct, for example, for variation between the individual spotting pens and for nonlinear relationships between the dye intensities.

After normalization, the data for each gene are typically reported as an 'expression ratio' or as the logarithm of the expression ratio. The expression ratio is simply the normalized value of the expression level for a particular gene in the query sample divided by its normalized value for the control. The advantage of using the logarithm of the expression ratio is simple to understand. Genes that are upregulated by a factor of 2 have an expression ratio of 2, whereas those downregulated by the same factor have an expression ratio of one-half (0.5) — downregulated genes are 'squashed' between 1 and 0. By contrast, a gene upregulated by a factor of 2 has a $\log_2$(ratio) of 1, whereas a gene downregulated by a factor of 2 has a $\log_2$(ratio) of –1, and a gene expressed at a constant level (with a ratio of 1) has a $\log_2$(ratio) of 0 (FIG. 1).

At this point in the analysis of a single experiment, we typically look for genes that are differentially expressed. Most published studies have used a post-normalization cut-off of twofold increase or decrease in measured level to define differential expression, although there is no firm theoretical basis for selecting this level as significant. Alternatively, the approach defined by Chen *et al.*[20] provides confidence intervals that can be used to identify differentially expressed genes.

It should be noted that there are disadvantages to using only expression ratios for data analysis. Although ratios can help to reveal some patterns in the data, they remove all information about the absolute gene-expression levels. Various parameters depend on the measured intensity, including the confidence limits that are placed on any microarray measurement. Although most of the techniques developed for analysis of microarray data use ratios, many of them can be adapted for use with measured intensities.

Box 2 | Distance metrics

In any clustering algorithm, the calculation of a 'distance' between any two objects is fundamental to placing them into groups. Analysis of microarray data is no different in that finding clusters of similar genes relies on finding and grouping those that are 'close' to each other. To do this, we rely on defining a distance between each gene-expression vector. There are various methods for measuring distance; these typically fall into two general classes: metric and semi-metric.

Metric distances

To be classified as 'metric', a distance measure d_{ij} between two vectors, *i* and *j*, must obey several rules:

- **The distance must be positive definite, $d_{ij} \geq 0$ (that is, it must be zero or positive).**
- **The distance must be symmetric, $d_{ij} = d_{ji}$, so that the distance from *i* to *j* is the same as the distance from *j* to *i*.**
- **An object is zero distance from itself, $d_{ii} = 0$.**
- **When considering three objects, *i*, *j* and *k*, the distance from *i* to *k* is always less than or equal to the sum of the distance from *i* to *j*, and the distance from *j* to *k*, $d_{ik} \leq d_{ij} + d_{jk}$. This is sometimes called the 'triangle' rule.**

The most common metric distance is Euclidean distance, which is a generalization of the familiar Pythagorean theorem. In a three-dimensional space, the Euclidean distance, d_{12}, between two points, (x_1,x_2,x_3) and (y_1,y_2,y_3) is given by EQN 1:

$$d_{12} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}, \qquad (1)$$

where (x_1,x_2,x_3) are the usual Cartesian coordinates (x,y,z). The generalization of this to higher-dimensional expression spaces is straightforward. For our *n*-dimensional expression vectors, the Euclidean distance is given by EQN 2:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}, \qquad (2)$$

where x_i and y_i are the measured expression values, respectively, for genes X and Y in experiment *i*, and the summation runs over the *n* experiments under analysis.

Semi-metric distances

Distance measures that obey the first three consistency rules, but fail to maintain the triangle rule are referred to as semi-metric. There are a large number of semi-metric distance metrics and these are often used in expression analysis. Mathematical descriptions of various distance metrics are available as supplementary material (see supplementary Box 1 online).

CLUSTER ANALYSIS
The term 'cluster analysis' actually encompasses several different classification algorithms that can be used to develop taxonomies (typically as part of exploratory data analysis). Note that in this classification, the higher the level of aggregation, the less similar are members in the respective class.

The true power of microarray analysis does not come from the analysis of single experiments, but rather, from the analysis of many hybridizations to identify common patterns of gene expression. Based on our understanding of cellular processes, genes that are contained in a particular pathway, or that respond to a common environmental challenge, should be co-regulated and consequently, should show similar patterns of expression. Our goal then is to identify genes that show similar patterns of expression and there exists a large group of statistical methods, generally referred to as 'CLUSTER ANALYSIS', that can be used to achieve this. Before comparing the clustering methods, I first discuss a mathematical definition for what we mean by 'similar', in the context of gene expression.

Comparing expression data

For expression data, we can begin to address the problem of 'similarity' mathematically by defining an 'expression vector' for each gene that represents its location in 'expression space'. In this view of gene expression, each experiment represents a separate, distinct axis in space and the $\log_2$(ratio) measured for that gene in that experiment represents its geometric coordinate. For example, if we have three experiments, the $\log_2$(ratio) for a given gene in experiment 1 is its *x*-coordinate, the $\log_2$(ratio) in experiment 2 is its *y*-coordinate, and the $\log_2$(ratio) in experiment 3 is its *z*-coordinate. So, we can represent all the information we have about that gene by a point in *x*–*y*–*z*-expression space. A second gene, with nearly the same $\log_2$(ratio) values for each experiment will be represented by a (spatially) nearby point in expression space; a gene with a very different pattern of expression will be far from our original gene. The generalization to more experiments is straightforward (although harder to draw): the dimensionality of expression space grows to be equal to the number of experiments. In this way, expression data can be represented in *n*-dimensional expression space, where *n* is the number of experiments, and where each gene-expression vector is represented as a single point in that space.

Having been provided with a means of measuring distance between genes, clustering algorithms sort the data and group genes together on the basis of their separation in expression space. It should also be noted that if we are interested in clustering experiments, we could represent each experiment as an 'experiment vector' consisting of the expression values for each gene; these define an 'experiment space', the dimensionality of which is equal to the number of genes assayed in each experiment. Again, by defining distances appropriately, we could apply any of the clustering algorithms defined here to analyse and group experiments.

To interpret the results from any analysis of multiple experiments, it is helpful to have an intuitive visual representation. A commonly used approach relies on the creation of an expression matrix in which each column of the matrix represents a single experiment and each row represents the expression vector for a particular gene. Colouring each of the matrix elements on the basis of its expression value creates a visual representation of gene-expression patterns across the collection of experiments. There are countless ways in which the expression matrix can be coloured and presented. The most commonly used method colours genes on the basis of their $\log_2$(ratio) in each experiment, with $\log_2$(ratio) values close to zero coloured black, those with $\log_2$(ratio) values greater than zero coloured red, and those with negative values coloured green. For each element in the matrix, the relative intensity represents the relative expression, with brighter elements being more highly differentially expressed. For any particular group of experiments, the expression matrix generally appears without any apparent pattern or order. Programmes designed to cluster data generally re-order the rows, or columns, or both, such that patterns of expression become visually apparent when presented in this fashion.

Before clustering the data, there are two further questions that need to be considered: first, should the data be adjusted in some way to enhance certain relationships? And second, what distance measure should be used to group together related genes? In many microarray experiments, the data analysis can be dominated by the variables that have the largest values, obscuring other, important differences. One way to circumvent this problem is to adjust or re-scale the data and there are several methods in common use with microarray data. For example, each vector can be re-scaled so that the average expression of each gene is zero — a process referred to as 'mean centring'. In this process, the basal expression level of a gene is subtracted from each experimental measurement. This has the effect of enhancing the variation of the expression pattern of each gene across experiments, without regard to whether the gene is primarily up- or downregulated. This is particularly useful for the analysis of time-course experiments, in which one might like to find genes that show similar variation around their basal expression level. The data can also be adjusted so that the minimum and maximum are ±1, or so that the 'length' of each expression vector is one.

The manner in which we measure distance between gene-expression vectors also has a profound effect on the clusters that are produced. (Several distance metrics are reviewed in BOX 2 and supplementary Box 1 online; others have been proposed[21].)

Box 3 | Hierarchical clustering algorithms

There are various hierarchical clustering algorithms[30] that can be applied to microarray data analysis[5–9,21]. These differ in the manner in which distances are calculated between the growing clusters and the remaining members of the data set, including other clusters. Clustering algorithms include, but are not limited to:

- ***Single-linkage clustering*. The distance between two clusters, *i* and *j*, is calculated as the minimum distance between a member of cluster *i* and a member of cluster *j*. Consequently, this technique is also referred to as the minimum, or nearest-neighbour, method. This method tends to produce clusters that are 'loose' because clusters can be joined if any two members are close together. In particular, this method often results in 'chaining', or the sequential addition of single samples to an existing cluster. This produces trees with many long, single-addition branches representing clusters that have grown by accretion.**
- ***Complete-linkage clustering*. Complete-linkage clustering is also known as the maximum or furthest-neighbour method. The distance between two clusters is calculated as the greatest distance between members of the relevant clusters. Not surprisingly, this method tends to produce very compact clusters of elements and the clusters are often very similar in size.**
- ***Average-linkage clustering*. The distance between clusters is calculated using average values. There are, in fact, various methods for calculating averages. The most common is the unweighted pair-group method average (UPGMA). The average distance is calculated from the distance between each point in a cluster and all other points in another cluster. The two clusters with the lowest average distance are joined together to form a new cluster. Related methods substitute the CENTROID or the median for the average.**
- ***Weighted pair-group average*. This method is identical to UPGMA, except that in the computations, the size of the respective clusters (that is, the number of objects contained in them) is used as a weight. This method (rather than UPGMA) should be used when the cluster sizes are suspected to be greatly uneven.**
- ***Within-groups clustering*. This is similar to UPGMA except that clusters are merged and a cluster average is used for further calculations rather than the individual cluster elements. This tends to produce tighter clusters than UPGMA.**
- ***Ward's method*. Cluster membership is determined by calculating the total sum of squared deviations from the mean of a cluster and joining clusters in such a manner that it produces the smallest possible increase in the sum of squared errors[31].**

CENTROID
The centroid of a cluster is the weighted average point in the multidimensional space; in a sense, it is the centre of gravity for the respective cluster.

Clustering algorithms

Various clustering techniques have been applied to the identification of patterns in gene-expression data. Most cluster analysis techniques are hierarchical; the resultant classification has an increasing number of nested classes and the result resembles a phylogenetic classification. Non-hierarchical clustering techniques also exist, such as *k*-means clustering, which simply partition objects into different clusters without trying to specify the relationship between individual elements. Clustering techniques can further be classified as divisive or agglomerative. A divisive method begins with all elements in one cluster that is gradually broken down into smaller and smaller clusters. Agglomerative techniques start with (usually) single-member clusters and gradually fuse them together. Finally, clustering can be either supervised or unsupervised. Supervised methods use existing biological information about specific genes that are functionally related to 'guide' the clustering algorithm. However, most methods are unsupervised and these are dealt with first.

Although cluster analysis techniques are extremely powerful, great care must be taken in applying this family of techniques. Even though the methods used are objective in the sense that the algorithms are well defined and reproducible, they are still subjective in the sense that selecting different algorithms, different normalizations, or different distance metrics, will place different objects into different clusters. Furthermore, clustering unrelated data will still produce clusters, although they might not be biologically meaningful. The challenge is therefore to select the data and to apply the algorithms appropriately so that the classification that arises partitions data sensibly.

Hierarchical clustering. Hierarchical clustering has the advantage that it is simple and the result can be easily visualized[13]. It has become one of the most widely used techniques for the analysis of gene-expression data. Hierarchical clustering is an agglomerative approach in which single expression profiles are joined to form groups, which are further joined until the process has been carried to completion, forming a single hierarchical tree. The process of hierarchical clustering proceeds in a simple manner. First, the pairwise distance matrix is calculated for all of the genes to be clustered. Second, the distance matrix is searched for the two most similar genes (see above) or clusters; initially each cluster consists of a single gene. This is the first true stage in the 'clustering' process. If several pairs have the same separation distance, a predetermined rule is used to decide between alternatives. Third, the two selected clusters are merged to produce a new cluster that now contains at least two objects. Fourth, the distances are calculated between this new cluster and all other clusters. There is no need to calculate all distances as only those involving the new cluster have changed. Last, steps 2–4 are repeated until all objects are in one cluster. There are several variations on hierarchical clustering (BOX 3) that differ in the rules governing how distances are

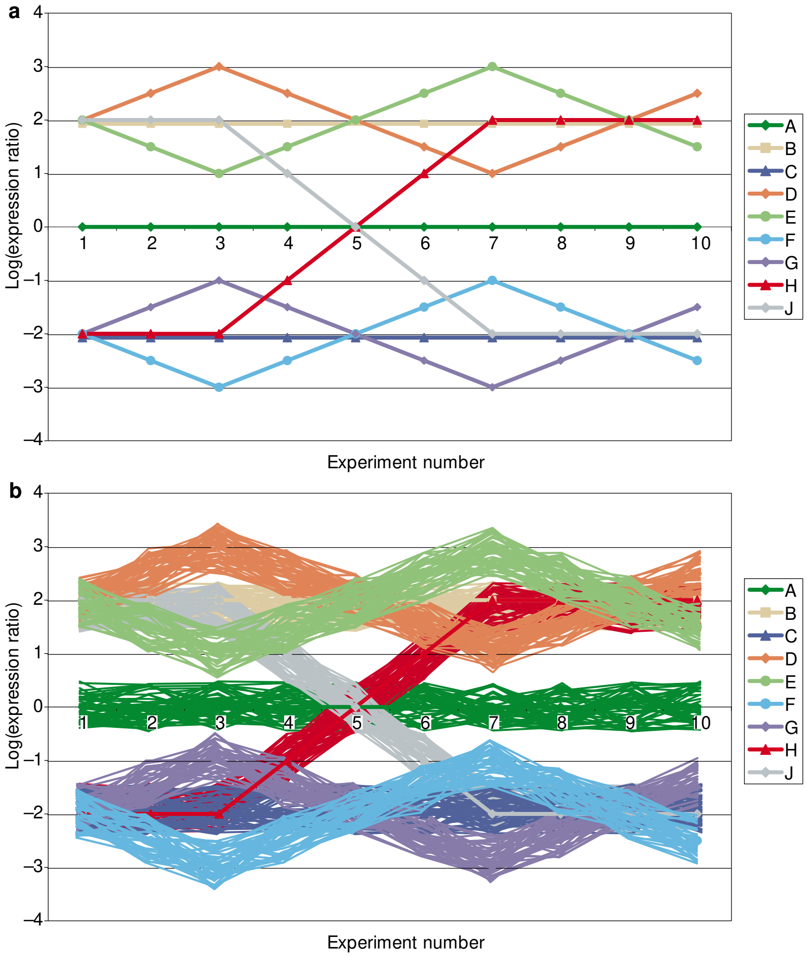


Figure 2 | **A synthetic gene-expression data set.** This data set provides an opportunity to evaluate how various clustering algorithms reveal different features of the data. **a** | Nine distinct gene-expression patterns were created with $\log_2$(ratio) expression measures defined for ten experiments. **b** | For each expression pattern, 50 additional genes were generated, representing variations on the basic patterns.

measured between clusters as they are constructed. Each of these will produce slightly different results, as will any of the algorithms if the distance metric is changed. Typically for gene-expression data, average-linkage clustering gives acceptable results.

One potential problem with many hierarchical clustering methods is that, as clusters grow in size, the expression vector that represents the cluster might no longer represent any of the genes in the cluster. Consequently, as clustering progresses, the actual expression patterns of the genes themselves become less relevant. Furthermore, if a bad assignment is made early in the process, it cannot be corrected. An alternative, which can avoid these artefacts, is to use a divisive clustering approach, such as *k*-means or self-organizing maps, to partition data (either genes or experiments) into groups that have similar expression patterns.

***k*-means clustering.** If there is advanced knowledge about the number of clusters that should be represented in the data, *k*-means clustering is a good alternative to hierarchical methods[22]. In *k*-means clustering, objects are partitioned into a fixed number (*k*) of clusters, such that the clusters are internally similar but externally dissimilar; no DENDROGRAMS are produced (but one could use hierarchical techniques on each of the data partitions after they are constructed). The process involved in *k*-means clustering is conceptually simple, but can be computationally intensive. First, all initial objects are randomly assigned to one of *k* clusters (where *k* is specified by the user). Second, an average expression vector is then calculated for each cluster and this is used to compute the distances between clusters. Third, using an iterative method, objects are moved between clusters and intra- and inter-cluster distances are measured with each move. Objects are allowed to remain in the new cluster only if they are closer to it than to their previous cluster. Fourth, after each move, the expression vectors for each cluster are recalculated. Last, the shuffling proceeds until moving any more objects would make the clusters more variable, increasing intra-cluster distances and decreasing inter-cluster dissimilarity.

Some implementations of *k*-means clustering allow not only the number of clusters, but also seed cases (or genes) for each cluster, to be specified. This has the potential to allow, for example, use of previous knowledge of the system to help define the cluster output. For example, an attempt to classify patients with two morphologically similar but clinically distinct diseases using microarray expression patterns can be imagined. By using *k*-means clustering on experiments with $k = 2$, the data will be partitioned into two groups. The challenge then faced is to determine whether there are really only two distinct groups represented in the data or not. In this case, *k*-means clustering is particularly useful with other techniques, such as principal component analysis (PCA, described below). PCA allows visual estimation of the number of clusters represented in the data. This can be used to specify *k* and to group genes (or experiments) into related clusters.

Self-organizing maps. A self-organizing map (SOM) is a NEURAL-NETWORK-based divisive clustering approach[11]. A SOM assigns genes to a series of partitions on the basis of the similarity of their expression vectors to reference vectors that are defined for each partition. It is the process of defining these reference vectors that distinguishes SOMs from *k*-means clustering. Before initiating the analysis, the user defines a geometric configuration for the partitions, typically a two-dimensional rectangular or hexagonal grid. Random vectors are generated for each partition, but before genes can be assigned to partitions, the vectors are first 'trained' using an iterative process that continues until convergence so that the data are most effectively separated. First, random vectors are constructed and assigned to each partition. Second, a gene is picked at random and, using a selected distance metric, the reference vector that is closest to the gene is identified. Third, the reference vector is then adjusted so that it is more similar to the vector of the assigned gene. The reference vectors that are nearby on the two-dimensional grid are also adjusted so that they are more similar to the

DENDROGRAM
A branching 'tree' diagram representing a hierarchy of categories on the basis of degree of similarity or number of shared characteristics, especially in biological taxonomy. The results of hierarchical clustering are presented as dendrograms, in which the distance along the tree from one element to the next represents their relative degree of similarity.

NEURAL NETWORKS
Neural networks are analytic techniques modelled after the (proposed) processes of learning in cognitive systems and the neurological functions of the brain. Neural networks use a data 'training set' to build rules capable of making predictions or classifications on data sets.

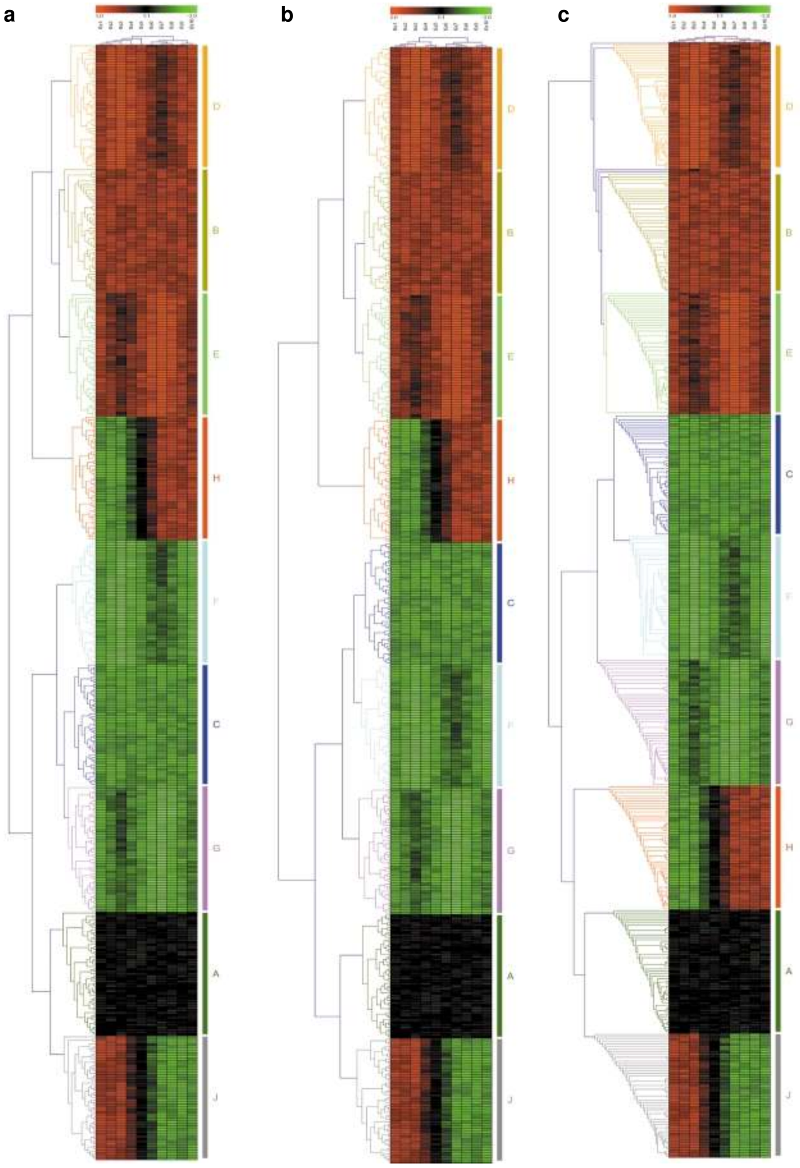


Figure 3 | **Hierarchical clustering.** Genes in the demonstration data set were subjected to **a** | average-linkage, **b** | complete-linkage and **c** | single-linkage hierarchical clustering using a Euclidean distance metric and gene-expression families (A–J) that were colour coded for comparison. Genes that are upregulated appear in red, and those that are downregulated appear in green, with the relative $\log_2$(ratio) reflected by the intensity of the colour. This method of clustering groups genes by reordering the expression matrix allows patterns to be easily visualized.

vector of the assigned gene. Fourth, steps 2 and 3 are iterated several thousand times, decreasing the amount by which the reference vectors are adjusted and increasing the stringency used to define closeness in each step. As the process continues, the reference vectors converge to fixed values. Last, the genes are mapped to the relevant partitions depending on the reference vector to which they are most similar.

In choosing the geometric configuration for the clusters, the user is, effectively, specifying the number of partitions into which the data is to be divided. As with *k*-means clustering, the user has to rely on some other source of information, such as PCA, to determine the number of clusters that best represents the available data.

FACTOR ANALYSIS
Factor analysis is a data reduction and exploratory method similar to pincipal component analysis. Factor analysis techniques seek to reduce the number of variables and to detect structure in the relationships between elements in an analysis.

Principal component analysis. An analysis of microarray data is a search for genes that have similar, correlated patterns of expression. This indicates that some of the data might contain redundant information. For example, if a group of experiments were more closely related than we had expected, we could ignore some of the redundant experiments, or use some average of the information without loss of information.

PCA (also called singular value decomposition) is a mathematical technique that exploits these factors to pick out patterns in the data, while reducing the effective dimensionality of gene-expression space without significant loss of information[23]. PCA is one of a family of related techniques that include FACTOR ANALYSIS and PRINCIPAL COORDINATE ANALYSIS that provide a 'projection' of complex data sets onto a reduced, easily visualized space.

Although the mathematics is complex, the basic principles are straightforward. Imagine taking a three-dimensional cloud of data points and rotating it so that you can view it from different perspectives. You might imagine that certain views would allow you to better separate the data into groups than other views. PCA finds those views that give you the best separation of the data. This technique can be applied to both genes and experiments as a means of classification.

In most implementations of PCA, it is difficult to define accurately the precise boundaries of distinct clusters in the data, or to define genes (or experiments) belonging to each cluster. However, PCA is a powerful technique for the analysis of gene-expression data when used with another classification technique, such as *k*-means clustering or SOMs, that requires the user to specify the number of clusters.

Analysis of a demonstration data set

The performance of these varied algorithms, normalization strategies and distance metrics is best shown by examining a demonstration data set (FIG. 2). Although this sample data set does not reflect the complexity of real biological data, its analysis can help to provide an understanding of how the data are handled and interpreted by the various methods.

The first analysis involves the three most commonly used variations on hierarchical clustering using a Euclidean distance metric without any data filtering (FIG. 3). Each performed quite well with respect to grouping together genes from a single expression class, although the branch lengths produced by each algorithm, and the structures of the individual clusters, differ quite a bit. It should be noted that, relative to the other algorithms, single-linkage clustering places expression group 'H' differently with respect to the other expression groups on the tree. The difference is due to the manner in which the growing clusters are linked together. In single-linkage clustering, growing clusters are joined on the basis of the distance between the closest members of their two respective clusters, whereas complete linkage uses the greatest distance between any two members of the groups and average linkage uses a group

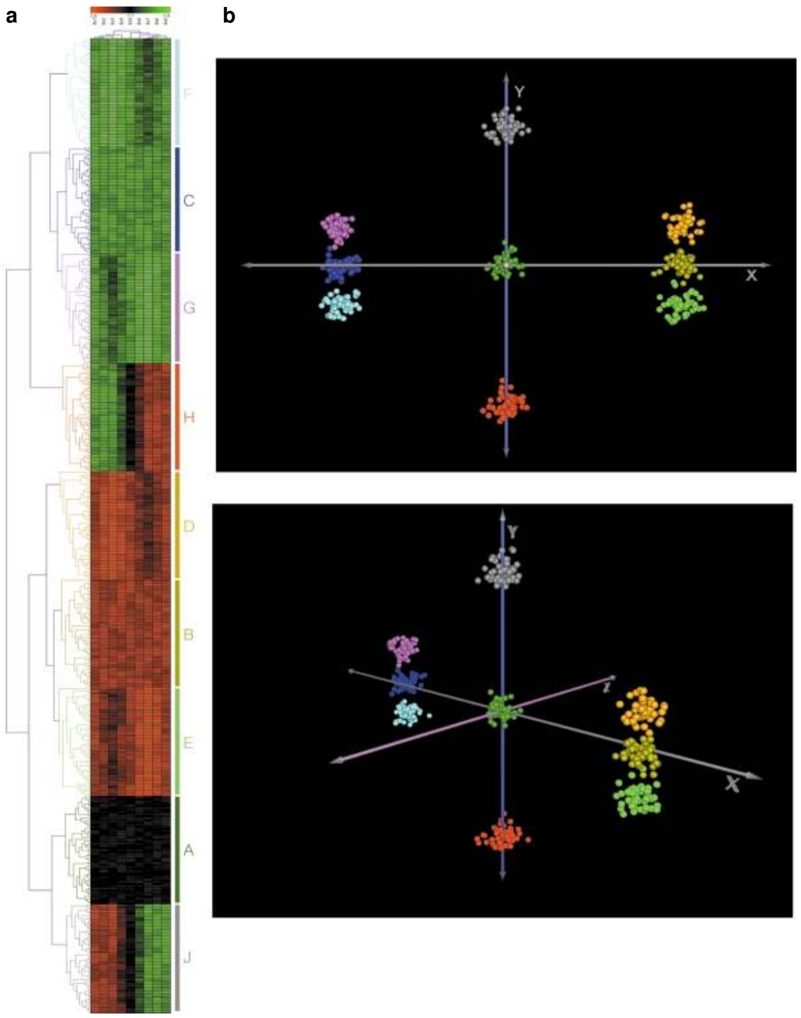


Figure 4 | **Principal component analysis.** The same demonstration data set was analysed using **a** | hierarchical (average-linkage) clustering and **b** | principal component analysis using Euclidean distance, to show how each treats the data, with genes colour coded on the basis of hierarchical clustering results for comparison.

PRINCIPAL COORDINATE ANALYSIS
Like principal component analysis, principal coordinate analysis seeks to reduce the dimensionality of a spatial representation of a data set by creating new coordinate axes that are a combination of the originals, and projecting the data onto those new axes.

average (BOX 3). Without a biological basis for interpreting these results, there is no way to decide which grouping is right and which is wrong. Depending on the actual experiment, any of the three approaches might provide a 'correct' order. As average-linkage clustering is the most commonly used approach, and because it grouped genes together in expression groups as well as the others, we will use this method as the basis for comparison with the non-hierarchical clustering algorithms.

Average-linkage clustering and PCA applied to the same data set are shown in FIG. 4. The nine groups of genes found in the hierarchical clustering can be clearly seen in the PCA analysis, although without previous knowledge of the results of the hierarchical analysis, one might argue that the data set only contains five distinct groups of genes. Application of *k*-means clustering and SOM analysis to this data set with more than five clusters produces five principal groups of genes with small numbers of genes assigned to the remaining groups. If the data set is analysed by *k*-means clustering, using *k* = 5, as one might expect from the PCA results, gene-expression groups B, D and E form one cluster, C, F and G form another, and A, H and J remain as distinct groups (see supplementary Figure 1 online). The results from this *k*-means analysis are also consistent with the results of hierarchical clustering in that genes that consistently up- or downregulated are grouped together, whereas those that vary around zero appear separately from these.

If, however, we were looking at time-course data and wanted to identify genes with expression levels that varied in a time-regulated fashion, this analysis would not have allowed such variations to be identified. One way to help identify coordinated fluctuations in the data is to first 'centre' the gene-expression vectors by subtracting the average across all experiments from each data point (FIG. 5). The results from both PCA and average-linkage clustering reflect this data 'filtering'. In the hierarchical clustering, genes with similar changes relative to their baseline expression patterns are grouped. The 'constant' A, B and C genes are now placed in a single cluster even though, in the original data, B genes were generally upregulated and C genes were generally downregulated. Similarly, the D and G genes cluster together, as do the E- and F-gene groups. Although the H and J groups appear next to each other in the dendrogram, they remain as separate groups, distinct from the others.

Although this data set does not reflect the full complexity of the real data sets, this analysis helps to show some of the complexity in the analysis of real microarray expression data. There often is no 'correct' way to analyse any data set; the application of various techniques, including algorithms and data filters, can help to reveal different features in the data. One must remember that the results of any analysis have to be evaluated in the context of other biological knowledge.

Supervised methods

The techniques discussed so far are unsupervised methods for identifying patterns of gene expression. Supervised methods represent a powerful alternative that can be applied if one has some previous information about which genes are expected to cluster together. One widely used example is the support vector machine (SVM)[24]. SVMs use a training set in which genes known to be related by, for example function, are provided as positive examples and genes known not to be members of that class are negative examples. These are combined into a set of training examples that is used by the SVM to learn to distinguish between members and non-members of the class on the basis of expression data. Having learned the expression features of the class, the SVM can then be used to recognize and classify the genes in the data set on the basis of their expression. In this way, SVMs use biological information to determine expression features that are characteristic of a group and to assign genes to that group. The SVM can also identify genes in the training set that are outliers or that have been previously assigned to the incorrect class.

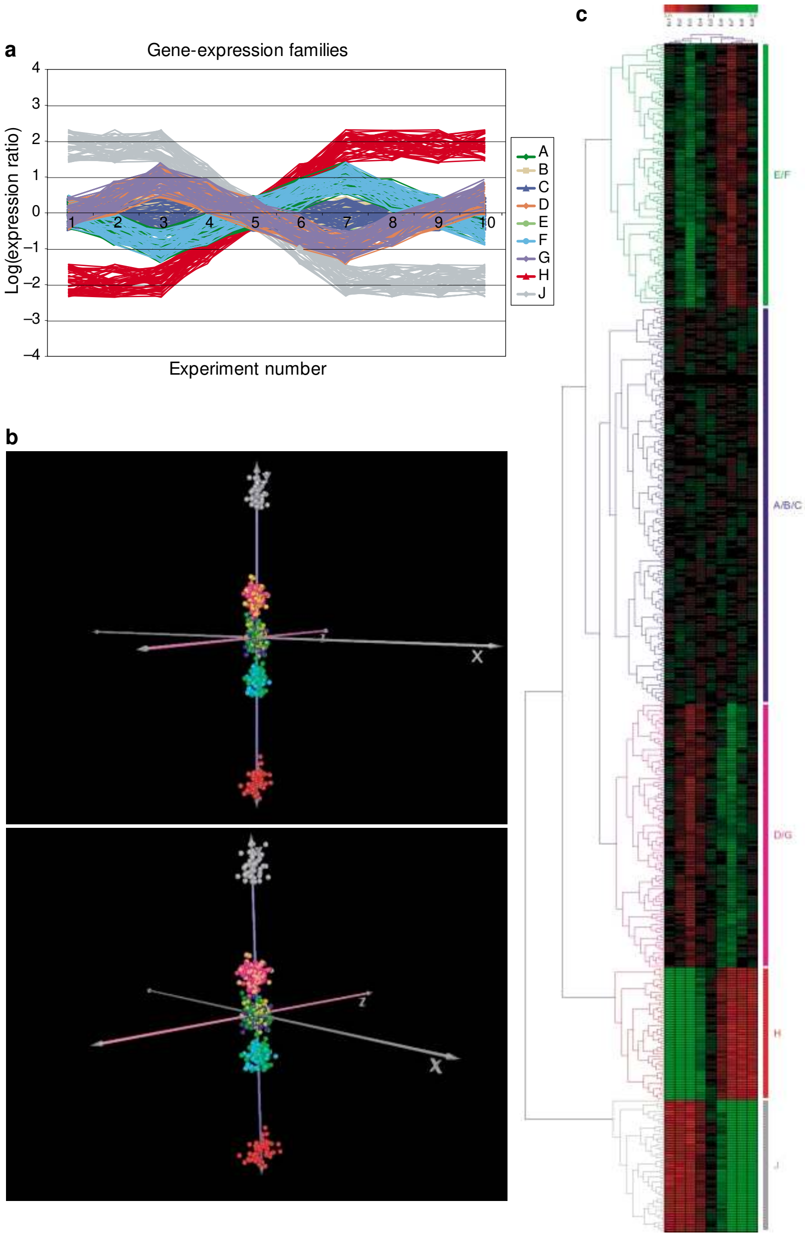


Figure 5 | **The effect of data filtering.** Application of various data filters or changes in the distance metric can change the results derived from any clustering algorithm. **a** | Mean centring of the data removes 'constant' expression, which reveals changes in expression patterns for the nine gene families across the ten experiments. The changes can be seen in the results of **b** | principal component analysis and **c** | average-linkage hierarchical clustering.

HYPERPLANE
A hyperplane is an *N*-dimensional analogy of a line or plane, which divides an '*N* + 1' dimensional space into two.

KERNEL FUNCTION
In support vector machines, the kernel function is a generalization of the distance metric; it measures the distance between two expression vectors as the data are projected into higher-dimensional space.

As discussed previously, gene-expression data can be thought of as an *m*-dimensional space, in which expression vectors are represented as points in that space. An SVM is a binary classifier that attempts to separate genes into two classes (in the positive training set, or outside it) by defining an optimal HYPERPLANE separating class members from non-members. However, for most real examples, there is no simple solution to this problem in expression space. SVM solves the problem by mapping the gene-expression vectors from expression space into a higher-dimensional 'feature space', in which distance is measured using a mathematical function known as a 'KERNEL FUNCTION', and the data can then be separated into two classes. For some data sets, SVMs might not achieve clean separation, either because of errors in classification in the training set, or noise in the data, or an improperly chosen kernel function. For this reason, most implementations also allow users to specify a 'soft margin' that allows some training examples to fall on the wrong side of the separating hyperplane. Completely specifying a SVM therefore requires specifying both the kernel function and the magnitude of the penalty to be applied for violating the soft margin.

As with the other techniques described here, this is one of the challenges of using SVMs. It is often difficult to choose the best kernel function, parameters and penalties. Different parameters often yield completely different classifications. It is therefore often necessary to successively increase kernel complexity until an appropriate classification is achieved[24].

SVMs are one of a group of supervised algorithms that have been applied to the classification of gene-expression patterns. Although they might be of use in the identification of genes that share related expression patterns, an application of potentially greater impact is the use of supervised methods for the classification of samples[25–27]. If we measure gene-expression patterns using RNAs collected from various patients for which there is, for example, disease-stage classification or survival data, we can use the microarray data to 'train' an algorithm that can then be applied to the classification of other previously unclassified samples. This approach could lead to the development of 'molecular expression fingerprinting' for disease classification. In cancer diagnosis, the ability to produce a molecular expression fingerprint of each tumour might prove to be extremely important as histologically similar tumours might in fact be the result of substantially different genetic changes, which might profoundly influence the progression of the tumour and its response to treatment.

Discussion and conclusions

Microarray expression analysis offers an opportunity to generate functional data on a genome-wide scale and consequently, should provide much-needed data for the biological interpretation of genes and their functions. It has also shown promise for classifying physiological and disease states. As the discussion presented here should show, the careful handling and interpretation of microarray expression data is not yet an exact science.

The hypothesis behind using clustering techniques is that genes in a cluster must share some common function or regulatory elements. However, classifications based on clustering algorithms are dependent on the particular methods used, the manner in which the data are normalized within and across experiments, and the manner in which we measure similarity; any and all of these factors can have a tremendous effect on the outcome of any analysis. Consequently, there is no such thing as a single correct classification, although different techniques might be more or less appropriate for different data sets. Furthermore, the application of more than one technique to the analysis of a particular data set might illuminate different relationships between the data. For example, a technique that allows us to find cell-cycle-regulated genes might obscure the expression response to whatever technique was used to synchronize the cells. As with experimental design, analysis techniques must be selected and tuned to best show the relationships in the data. Cluster analysis does not give absolute answers. Instead, these are data-mining techniques that allow relationships in the data to be explored. Some of the most exciting and promising applications are those that classify human disease states using patterns of gene expression[25–27].

The tools and techniques described here are by no means comprehensive and many new algorithms and software tools are under development. As the analysis presented here has shown, the ultimate guide to the use and applicability of any laboratory technique or data analysis method must be our biological understanding. If an analysis provides insight into the data that is consistent with our understanding of the system under study, then any extension it provides is more likely to be valid (for example, by classifying novel genes that might be involved in a known pathway).

Finally, new means for studying gene and protein expression are continuing to be developed, as are the tools for data analysis and its applications. Many of the techniques applied to the analysis of microarray data will be applicable to the data sets provided by these new techniques and will allow the interactions represented in those data to be explored. As with microarray analysis, these explorations will provide hypotheses that can be tested in the laboratory using more standard biological and biochemical methods.

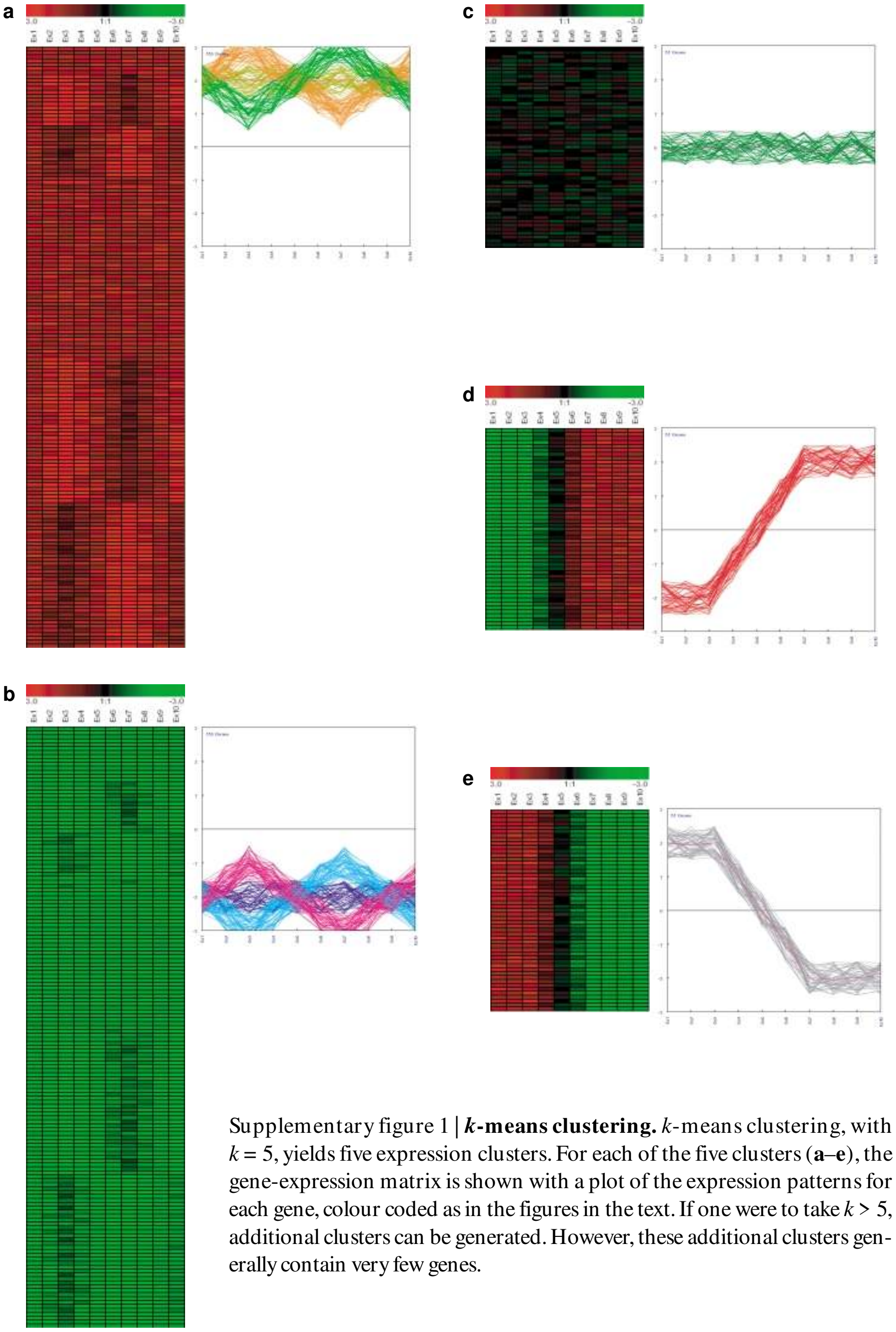
Links

PUBLIC EST SEQUENCES **dbEST**
cDNA DATABASES **UniGene | TIGR Gene Indices | STACK | DoTS**
IMAGE-PROCESSING SOFTWARE **Axon | BioDiscovery | Imaging Research | NHGRI Microarray Project | TIGR software tools | Eisen lab**
DATA ANALYSIS TOOLS **BioDiscovery | European Bioinformatics Institute (EBI) Expression Profiler | Eisen lab | Silicon Genetics | Spotfire | X Cluster | TIGR software tools | J-express**
META-LISTS OF OTHER AVAILABLE SOFTWARE **EBI | National Center for Genome Resources | Rockefeller | École Normale Supérieure | Stanford**

1. Velculescu, V. E., Zhang, L., Vogelstein, B. & Kinzler, K. W. Serial analysis of gene expression. *Science* **270**, 484–487 (1995).
2. Lockhart, D. J. *et al.* Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnol.* **14**, 1675–1680 (1996).
3. Schena, M., Shalon, D., Davis, R. W. & Brown, P. O. Quantitative monitoring of gene expression patterns with complementary DNA microarray. *Science* **270**, 467–470 (1995).
4. Schena, M. *et al.* Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc. Natl Acad. Sci. USA* **93**, 10614–10619 (1996).
5. Wen, X. *et al.* Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl Acad. Sci. USA* **95**, 334–339 (1998).
This is one of the first analyses of large-scale gene expression — in this case, RT–PCR data — using clustering and data-mining techniques. It elegantly shows how integrating the results derived using various distance metrics can reveal different but meaningful patterns in the data.
6. Michaels, G. S. *et al.* Cluster analysis and data visualization of large-scale gene expression data. *Pacific Symp. Biocomput.* **1998**, 42–53 (1998).
7. 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).
This is an excellent demonstration of the power of hierarchical clustering to the analysis of microarray data. The authors also provide software — Cluster and Treeview — which became the standard for analysing microarray data.
8. Weinstein, J. N. *et al.* An information-intensive approach to the molecular pharmacology of cancer. *Science* **275**, 343–349 (1997).
Weinstein and colleagues present one of the first and most elegant applications of hierarchical clustering and other data-mining and visualization techniques to the analysis of large-scale data in molecular biology.
9. Sokal, R. R. & Michener, C. D. A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.* **38**, 1409–1438 (1958).
10. Shannon, C. C. & Weaver, W. *The Mathematical Theory of Communication* (Illinois Univ. Press, Illinois, 1963).
11. Kohonen, T. *Self Organizing Maps* (Springer, Berlin, 1995).
12. 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).
Tamayo and colleagues use self-organizing maps (SOMs) to explore patterns of gene expression generated using Affymetrix arrays, and provide the GENECLUSTER implementation of SOMs.
13. Eisen, M. B. & Brown, P. O. DNA arrays for analysis of gene expression. *Meth. Enzymol.* **303**, 179–205 (1999).
14. Hegde, P. *et al.* A concise guide to microarray analysis. *Biotechniques* **29**, 548–560 (2000).
15. Boguski, M. S. & Schuler, G. D. ESTablishing a human transcript map. *Nature Genet.* **10**, 369–371 (1995).
16. Quackenbush, J. *et al.* The TIGR gene indices: analysis of gene transcript sequences in highly sampled eukaryotic species. *Nucleic Acids Res.* **29**, 159–164 (2001).
17. Burke, J., Wang, H., Hide, W. & Davison, D. B. Alternative gene form discovery and candidate gene selection from gene indexing projects. *Genome Res.* **8**, 276–290 (1998).
18. Ermolaeva, O. *et al.* Data management and analysis for gene expression arrays. *Nature Genet.* **20**, 19–23 (1998).
19. Sherlock, G. *et al.* The Stanford Microarray Database. *Nucleic Acids Res.* **29**, 152–155 (2001).
20. Chen, Y., Dougherty, E. R. & Bittner, M. Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J. Biomed. Opt.* **2**, 364–374 (1997).
21. Heyer, L. J., Kruglyak, L. & Yooseph, S. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.* **9**, 1106–1115 (1999).
22. Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J. & Church, G. M. Systematic determination of genetic network architecture. *Nature Genet.* **22**, 281–285 (1999).
23. Raychaudhuri, S., Stuart, J. M. & Altman, R. B. Principal components analysis to summarize microarray experiments: application to sporulation time series. *Pac. Symp. Biocomput.* **2000**, 455–466 (2000).
24. Brown, M. P. *et al.* Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl Acad. Sci. USA* **97**, 262–267 (2000).
This paper shows the power of supervised techniques, in this case support vector machines, to provide additional insight into gene expression and function.
25. Golub, T. R. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–537 (1999).
26. Furey, T. S. *et al.* Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16**, 906–914 (2000).
27. Hedenfalk, I. *et al.* Gene-expression profiles in hereditary breast cancer. *N. Engl. J. Med.* **344**, 539–548 (2001).
28. Chatterjee, S. & Price, B. *Regression Analysis by Example* (John Wiley and Sons, New York, 1991).
29. Cleveland, W. S. & Devlin, S. J. Locally weighted regression: an approach to regression analysis by local fitting. *J. Am. Stat. Assoc.* **83**, 596–610 (1988).
30. Sokal, R. R. & Sneath, P. H. A. *Principles of Numerical Taxonomy* (W. H. Freeman & Co., San Francisco, 1963).
31. Ward, J. H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **58**, 236–244 (1963).

Acknowledgements

Cluster analysis was done using the The Institute for Genomic Research MeV software package developed by A. Sturn, A. I. Saeed and J.Q., which is available at http://pga.tigr.org/tools.shtml, along with the sample data set used here. The author also thanks A. Sturn, N. H. Lee, R. L. Malek and E. Snesrud for valuable discussions and comments. This work is supported by grants from the US National Science Foundation, the US National Cancer Institute, and the US National Heart, Lung, and Blood Institute.



Supplementary figure 1 | ***k*-means clustering.** *k*-means clustering, with $k = 5$, yields five expression clusters. For each of the five clusters (**a**–**e**), the gene-expression matrix is shown with a plot of the expression patterns for each gene, colour coded as in the figures in the text. If one were to take $k > 5$, additional clusters can be generated. However, these additional clusters generally contain very few genes.

Supplementary box 1 | Distance metrics

In addition to the Euclidean distance, there are various distance measures that are used in the analysis of gene-expression data. The Manhattan distance (or city block distance) is an example of a non-Euclidean metric distance measure. The name comes from the distance one would travel in crossing a large city, such as Manhattan, in which the streets are laid out in a regular, rectangular grid. In most cases, this distance measure yields results similar to the simple Euclidean distance. The Manhattan distance is calculated as the sum of the absolute distances between the components of each expression vector, given by EQN 1:

$$d = \sum_{i=1}^{n} \left| x_i - y_i \right. . \qquad (1)$$

The most commonly used semi-metric distance measure in the analysis of gene-expression data is the Pearson correlation coefficient (also known as the centred Pearson correlation coefficient), *r*, given by EQN 2:

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}\sqrt{\sum_{i=1103}^{n} (y_i - \bar{y})^2}} , \qquad (2)$$

where $\bar{x}$ and $\bar{y}$ are the mean expression values for the X and Y genes, respectively.

The values of the Pearson correlation coefficient range between –1 and +1, with *r* = 1 when the two vectors are identical (perfect correlation), *r* = –1 when the two vectors are exact opposites (perfect anti-correlation), and *r* = 0 when the two vectors are completely independent (uncorrelated or orthogonal vectors).

The Pearson correlation coefficient is very useful if the 'shape' of the expression vector is more important than its magnitude. If, however, the relative expression level is important, it is better to use the uncentred Pearson correlation coefficient, given by EQN 3:

$$r_{\mathrm{un}} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} , \qquad (3)$$

for curves of the same shape, but different relative magnitudes.

The centred and uncentred Pearson correlation coefficients are useful for examining correlations in the data, but are not useful for identifying genes, the expression levels of which are anti-correlated. One might imagine an instance, for example, in which the same transcription factor can cause both enhancement and repression of expression. In this case, a better alternative is the squared Pearson correlation coefficient, given by EQN 4:

$$r_{\mathrm{sq}} = \left(\frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} {x_i}^2}\ \sqrt{\sum_{i=1}^{n} {y_i}^2}} \right)^2 . \qquad (4)$$

Although the Pearson correlation coefficient takes on values, the squared Pearson correlation coefficient takes values in the range $0 \le r_{\mathrm{sq}} \le 1$, where uncorrelated vectors have $r_{\mathrm{sq}} = 0$, and both perfectly correlated and anti-correlated expression vectors have $r_{\mathrm{sq}} = 1$.