

Network visualization and analysis of gene expression data using BioLayout *Express*^{3D}

Athanasios Theocharidis¹, Stijn van Dongen², Anton J Enright² & Tom C Freeman¹

¹The Roslin Institute, R(D)SVS, University of Edinburgh, Roslin BioCentre, Midlothian, Scotland, UK. ²EMBL—European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK. Correspondence should be addressed to T.C.F. (Tom.Freeman@roslin.ed.ac.uk).

Published online 1 October 2009; doi:10.1038/nprot.2009.177

Network analysis has an increasing role in our effort to understand the complexity of biological systems. This is because of our ability to generate large data sets, where the interaction or distance between biological components can be either measured experimentally or calculated. Here we describe the use of BioLayout *Express*^{3D}, an application that has been specifically designed for the integration, visualization and analysis of large network graphs derived from biological data. We describe the basic functionality of the program and its ability to display and cluster large graphs in two- and three-dimensional space, thereby rendering graphs in a highly interactive format. Although the program supports the import and display of various data formats, we provide a detailed protocol for one of its unique capabilities, the network analysis of gene expression data and a more general guide to the manipulation of graphs generated from various other data types.

INTRODUCTION

Over the past decades, researchers have painstakingly worked out the functional role of specific proteins in their system of interest and characterized details of their interaction partners and the pathways in which they function. Although of significant use in shaping our view of the underlying molecular activity that control these systems, these studies at best provide mere snapshots of the system as a whole. More recently, enormous amounts of data pertaining to the activity of genes and proteins and their interactions in the cell have been generated by a number of high-throughput techniques, including yeast two-hybrid assays, mass spectrometry, RNA interference and gene expression analysis¹. In order to understand the data generated by these functional genomics and proteomics approaches, there has been increasing emphasis on developing methods in computational biology and the emerging discipline of systems biology, to allow for the comprehensive mapping of cellular and molecular networks and pathways^{2,3}. However, one of the main difficulties we currently face is how best to integrate and visualize these disparate data types and use the information gleaned from these efforts to better understand biological systems in health and disease⁴.

Visualization and analysis of data as networks is becoming an increasingly important approach in the exploration of relationships between entities in various different fields. Shifting data into a graph/network paradigm allows one to use algorithms, techniques, ideas and statistics previously developed in graph theory, engineering and computer science. Graph and network analysis techniques allow the exploration of the position of a biological entity in the context of its local neighborhood in the graph and the network as a whole⁵. Another important advantage of such techniques is that for noisy data sets, spurious edges tend not to form structure (or cliques) in the resultant graph, but instead randomly link nodes. Because many network analysis techniques (e.g., graph clustering) exploit local structure in networks between related nodes, they are far less troubled by inherent noise, which may confound conventional pair-wise approaches⁵.

In biology, such approaches have already been used with great success in the study of sequence similarity, protein structure, protein interactions, evolutionary relationships and gene

expression^{5–8}. In classical graph theory, a graph or a network consists of nodes connected by edges. For biological networks, nodes are usually genes, transcripts or proteins, whereas edges tend to represent experimentally determined similarities or functional linkages between them⁹. The use of network-based analyses in biology is now well established and a hallmark of a systems biology paper.

There are now a range of academic and commercial programs that have been developed for the construction, display, editing and analysis of biological networks and pathways, as well as more generic network analysis and visualization platforms^{10–13}. Comprehensive surveys of these tools comparing features and availability have been published recently^{14–16}, and we recommend them for an in-depth discussion of the network-based tools. However, it is worth mentioning the common features of these tools and the current limitations in functionality. Most available tools are designed to support the visualization of protein–protein interaction networks and metabolic or signaling pathways. They all support the ability to display a biological component as a node, whose shape, size and color can be modified so as to display various characteristics of that component or the group of components to which it belongs. This might be used to visually depict groups of nodes as belonging to classes based on their functional properties, e.g., cellular location, expression level, etc. In a network graph, nodes are connected by edges, which may be directional, e.g., A interacts with B, but B does not interact with A. Edges can be undirected, e.g., based on a reciprocal similarity measure. Edges can be weighted, where the edge weight signifies the strength of the connection. Finally, edges may confer a particular type of relationship between two components, e.g., /A phosphorylates B/ or /A binds to B/. The display of such relationships is key to providing a flexible graphical platform to display biological networks and pathways. Laying out a graph of relationships is a nontrivial matter. Technically, a useful graph layout must represent the data in an ‘aesthetically pleasing’ and logical manner. This means algorithms are used to place graph nodes and edges in such a way that the number of edges crossing is minimized and that the layout represents the overall structure of the graph. Hierarchical, organic, circular and orthogonal graph

formats are commonly used for graph layout. Finally, having rendered a graph in an optimal layout, there is also a need to analyze the networks in terms of their overall graph statistics, e.g., number of nodes/edges, node degree (edges per node), graph diameter, etc., and also to interact with the data and explore the local structure and connectivity.

BioLayout *Express*^{3D} (ref. 7) is an application designed for displaying network graphs from biologically derived data and is based on an earlier program, BioLayout Java, written in Java and utilizing Java2D for graphics^{17,18}. The current version is unique in having been specifically designed and optimized for the display of very large graphs containing tens of thousands of nodes and millions of edges. To achieve this end, we used the hardware-accelerated OpenGL framework combined with an optimized layout algorithm and graph clustering, which allows the user to effectively explore graphs and data sets beyond the reach of other tools with a standard consumer computer. The three-dimensional (3D) nature of the interface is also better suited for user interaction with larger, more complicated data sets where understanding graph structure and topology is of paramount importance in data exploration. In particular, this tool has been designed for, and indeed driven by, the need to analyze graphs derived from gene expression studies, BioLayout *Express*^{3D} being the only tool to support the visualization and analysis of these large and complicated graphs ‘out of the box’.

The idea of transforming gene expression data into a network graph based on correlation measures is not new^{17–20}. The similarity between individual expression profiles may be determined by one of a number of possible statistical techniques, e.g., Pearson’s and Spearman’s Rank. Networks can be constructed by connecting transcripts (nodes) by edges that infer a degree of coexpression based on a given method of calculating correlation and defined threshold. However, this approach has not previously been widely explored because of the lack of a tool that supports these analyses. All other network analysis tools lack the functionality to support the input of raw expression data, but, in particular, are unable to display and allow the user to interact with and analyze large graphs comprising many thousands of nodes and edges. Graphs generated from large expression data sets are normally highly structured. However, graphs of this size are difficult to visualize, comprehend and navigate when only two-dimensional (2D) rendering is available. 3D rendering confers a distance onto the nodes relative to the viewer enabling subparts of the graph to be focused independently, and effectively allows the display to pack information more effectively in the viewing area. This concept gains importance proportionally to the size of the graph that is rendered. Furthermore, the ability to rotate a graph and look at it from different positions confers additional cues to the viewer, and aids navigation and user interaction with the graphs. The other advantage of displaying graphs using OpenGL and 3D graphics is that graphs can be made bigger and rendered and explored interactively in real time. BioLayout *Express*^{3D} can currently render graphs of up to 30,000 nodes and 2–3 million edges using standard workstation hardware with a 3D accelerated graphics card.

BioLayout *Express*^{3D} also supports the input of data in a number of standard graph formats (see **Box 1**), and appropriate data from any source, biological or otherwise, can be visualized using this tool. Our original publication described the network approach to analyzing gene expression data with this tool⁷, and it remains a unique feature of BioLayout *Express*^{3D}. Indeed, large repositories

of microarray data are now readily available in the public domain in databases such as ArrayExpress and GEO, and much of this data has been little explored. Here, we provide for the first time a detailed protocol on the use of BioLayout *Express*^{3D}, in particular with respect to the analysis of microarray gene expression data. Development of the program is ongoing and new features will be added in the near future as we refine the tools for existing applications, develop new applications and optimize its implementation to make the best use of rapid advances in Java/OpenGL and improvements in hardware, particularly graphics cards.

The graph paradigm for microarray expression data

Microarray expression data typically consists of many thousands to millions of measurements of relative transcript abundance taken across anything from just a few to several thousand biological samples. The data can be influenced not only by the variation in the biological system of interest but also by technical artifacts. Traditional approaches to its analysis are generally focused on identifying statistical differences between groups of samples or use a range of explorative clustering techniques to divide data into groups (clusters) of genes showing similar expression profiles. The network paradigm for the analysis of data is based on the use of a correlation measure to define similarities between expression profiles. Generation of these graphs makes no earlier assumptions regarding the experimental design, normalization method, microarray platform or indeed questions being addressed by the study, and therefore provides a truly unbiased view of the data. However, all of these factors do influence the properties of a network, and therefore it is worth spending some time to discuss them.

Experimental design

The design of a given experiment influences the resultant graph. However, unlike many statistical-based approaches to analysis where the experimental design (number of samples per grouping, balanced or not) dictates the analytical approach used, network analysis is less influenced by these criteria. A network is constructed purely on the basis of the data provided and although ultimately a user’s ability to draw meaningful analysis from this data is a function of the experimental design, this does not fundamentally affect the analytical approach.

The size of a data set is a function of the number of probes on the array and the number of samples analyzed. In terms of the input file, this translates to the number of rows and columns, respectively. As the approach is based on constructing graphs of the correlation between probes, where the number of samples is small (< 10), there will generally be a higher correlation between data derived from different probes as the space for variation is restricted. In other words, the correlation between probes representing ‘unchanging’ genes will be higher if these were measured over many more samples. This results in graphs from small unfiltered data sets being large even at high correlation cutoffs. When dealing with relatively small data sets, especially those where there is little biological or technical variation, we recommend that the data are filtered by statistical approaches to isolate the genes of interest before graph construction. In contrast, when dealing with large data sets comprised of ~20 samples or more, there is often no need to filter data at all before loading into the tool. Depending on the number of measurements included in the data file, BioLayout *Express*^{3D} should be able to load data from up to 500 to 1,000 genome-wide expression arrays.

BOX 1 | BioLayout *Express*^{3D} INPUT FORMATS

Listed below are the basic input formats for BioLayout *Express*^{3D} graphs. Example graphs of each type are available on the website. We have aimed to make data input both flexible and simple. The formats below are quite flexible and allow various different data types to be loaded and graphed. The user has considerable control over how nodes and edges are annotated and rendered in the final graph. Before we describe the type of data formats currently readable by BioLayout *Express*^{3D}, we will describe some conventions:

- BioLayout *Express*^{3D} files are usually text files representing columns of data.
- Data points are separated by tabs.
- Each node should have a unique identifier.
- Ensure text entries such as annotations are enclosed by quotations where they contain spaces (e.g., 'Protein Kinase Alpha').
- Comments may be placed in the file by preceding them with '//'.
Advanced options such as upfront definition of //NODESIZE or //NODESHAPE are usually placed at the end of the input file.

In the following section we will describe these various formats and how to load different types of data into BioLayout *Express*^{3D}.

Simple multicolumn format (.layout, .txt)

This is perhaps the easiest input format for dealing with heterogeneous data types in BioLayout *Express*^{3D}. The format allows a full range of nodes, edges and classes to be created from a simple column format that can be prepared in a spreadsheet such as Excel. The basic format to define such networks is as follows:

Simple pair-wise edges. This will create a simple directional network where each line of the input file defines two nodes that are connected to each other by a new edge. If one desires a bi-directional graph, then only one direction needs to be added and the other edge can be inferred according to the layout properties panel. The parser will create new nodes in the network as required. The format itself is shown below in bold.

```
Node1      Node2
NodeA    NodeB
NodeB    NodeC
NodeC    NodeD
```

In each line, an edge is defined by connecting the node described in the first column with the node defined in the second column. The graph constructed here would consist of four nodes (A, B, C and D) each connected to the next by a single non-weighted edge.

Pair-wise edges with weights. This is a simple extension of the simple pair-wise format, which also adds a weight to each edge. Edge weights will affect how the edge is colored and will also influence the layout algorithm. Nodes connected by higher weighted edges tend to be closer together in the resulting layout. The format is a one-column extension to the previous format adding a single numeric weight, as illustrated below:

```
Node1      Node2      Weight
NodeA    NodeB    1.0
NodeB    NodeC    0.95
NodeC    NodeD    0.86
```

Weights should normally be in linear ranges and in whatever scale is appropriate as they will be re-centered. Non-linear weights can be log-scaled if desired. Negative weights are currently not supported.

Pair-wise edges with weights and edge annotation. This format extends the previous pair-wise weighted format adding support for edge annotation. In this case, weighted edges are constructed between node pairs and a pseudo-node describing the edge is added to the edge. This allows pairs of nodes to be connected using different edge annotations of different weights, as shown in the example below:

```
Node1      Node2      Weight      EdgeType
NodeA    NodeB    1.0          'Yeast 2-hybrid'
NodeB    NodeC    0.95        'Co-immunoprecipitation'
NodeC    NodeD    0.86        'Computationally Inferred'
NodeC    NodeD    0.95        'Yeast 2-hybrid'
```

It should be noted that annotations containing spaces should be quoted in either double or single quotes for the parser to recognize them correctly. In addition, characters such as quotes and other special characters should be removed from annotation lines.

Reactome OWL format (.owl)

BioLayout *Express*^{3D} currently supports OWL format graphs generated by Reactome but not necessarily from other OWL sources. Future versions of BioLayout *Express*^{3D} will endeavor to support full OWL parsing and other markup formats.

Cytoscape SIF format (.sif)

BioLayout *Express*^{3D} supports the simple Cytoscape SIF format. In this format, nodes are connected to each other with an explicit declaration of the edge type. The first column represents the first node, the second describes the type of interaction and the last column defines the second node in the edge pair:

BOX 1 | CONTINUED

A	'Yeast 2-hybrid'	B
B	'Co-immunoprecipitation'	C
C	'Computationally Inferred'	D

The main difference between BioLayout *Express*^{3D} SIF parsing and native Cytoscape SIF parsing is that lines connecting one node to many are not currently supported. However, most online resources that provide SIF files (e.g., BOND) do not use this multi-node connection format.

Matrix files (.matrix)

We have recently implemented the support of correlation matrix file import in BioLayout *Express*^{3D}. In principle, the matrix files may be generated from any set of numbers using any correlation measure but must have a '.matrix' extension in order for BioLayout *Express*^{3D} to recognize them. On opening of a .matrix file (as illustrated below), a Matrix CutOff dialog will appear requesting the user to define the threshold above which relationships will be plotted.

	A	B	C	D	E	F
A	1.00000	0.98663	0.93504	0.93464	0.92341	0.91745
B	0.98663	1.00000	0.93365	0.92930	0.92165	0.91817
C	0.93504	0.93365	1.00000	0.98991	0.96653	0.96679
D	0.93464	0.92930	0.98991	1.00000	0.96728	0.96799
E	0.92341	0.92165	0.96653	0.96728	1.00000	0.98699
F	0.91745	0.91817	0.96679	0.96799	0.98699	1.00000

Expression data input format (.expression)

Expression files must have an '.expression' extension in order for BioLayout *Express*^{3D} to recognize them. Beware that when saving an .expression file from a Windows package, e.g., Excel, one will need to put the file name in double quotation marks ('name.expression') to avoid a .txt extension being added to the name. Files with an extension '.expression' will automatically be associated with the program and when clicked twice they will be automatically opened inside the program.

The basic format is a header row, followed by a single row for each probe (set)/gene on the array. Each row must start with the unique identifier of that row (node). Annotation columns may then follow the identifier (these are optional but very useful), followed finally by the raw data columns, which are usually numeric (integer or floating point). Columns are usually tab separated in this format and text entries are surrounded by double quotes:

Unique ProbeID	Description	Annotation1	Annotation2	Data1	Data2	Data3
Tub;gnf1m00002_f_at	tubulin, alpha 7	Term1	Term1	245.6	278.9	364.6
I116;gnf1m00009_s_at	interleukin 16	Term2	Term1	125	203	235.2
Cul17;gnf1m00122_a_at	cullin 7	Term3	Term2	302	288	134.7

Graphml (.graphml)

GraphML is an easy-to-use file format for network graphs. It was designed to describe the structural and visual properties of a network graph. Its main features include support of directed, undirected and mixed graphs, hypergraphs, hierarchical graphs, graphical representations, references to external data, application-specific attribute data and light-weight parsers. Unlike many other file formats for graphs, GraphML does not use a custom syntax. Instead, it is based on XML and hence is suited as a common denominator for all kinds of services generating, archiving or processing network graphs. There are now a number of programs that support the import and export of GraphML files. We have been using the program yEd Graph Editor (yFiles) for the construction of pathway diagrams²⁶. However, certain functions of interest to us are not supported within yEd Graph Editor; hence, we have recently implemented a parser that supports the import of GraphML files into BioLayout *Express*^{3D}. Once created, a GraphML file may be directly opened in BioLayout *Express*^{3D}.

Creation of classes

Nodes can be assigned to multiple classes so that multiple annotations may be layered on the same graph. Examples of such annotation might be Gene Ontology terms or Enzyme Classification numbers assigned to nodes in a protein graph. Node classes are differentiated from each other primarily by color and alternatively by shape or size of node. BioLayout *Express*^{3D} operates on a system of Class Sets, which refer to the overall type of classes being assigned (e.g., GO Term, EC Number). Each node may have only one class annotation within a Class Set. It is not required that all nodes have an annotation in any given Class Set. Nodes without a defined class are added to a default unannotated class.

BOX 1 | CONTINUED

Creating classes in the simple pair-wise formats is straightforward. To assign a specific node, a specific class within a Class Set is done as follows:

<i>NodeID</i>	<i>Class Set name</i>	<i>Node annotation within Class Set</i>
//NODECLASS NodeA	'Gene Ontology'	'histone deacetylase'
//NODECLASS NodeA	'Enzyme Classification'	'3.5.1'
//NODECLASS NodeB	'Enzyme Classification'	'2.7.1'

The example above will create two new Class Sets ('Gene Ontology' and 'Enzyme Classification'). NodeA will be assigned an annotation of 'histone deacetylase' in the GO Class Set and to '3.5.1' in the EC Class Set. NodeB is only assigned a specific annotation '2.7.1' (Kinase) in the EC Class Set and would be set as 'unannotated' in the Gene Ontology Class Set. The above annotation lines should be added to the bottom of the input file after edge descriptions. Once again, it is best if annotations are quoted using single or double quotes (as shown).

New classes within a Class Set are automatically assigned a random color. To change this color, one can encode the desired color at the end of an input file as follows:

<i>Class Set name</i>	<i>Class annotation</i>	<i>Color</i>
//NODECLASSCOLOR 'Gene Ontology'	'histone deacetylase'	'FF0000'

This example changes the default color for the class 'histone deacetylase' in the Class Set 'Gene Ontology' to red. The last column specifying the color is composed of standard HTML hexadecimal RGB triplets. A number of online tools exist that provide these encodings for any color desired, e.g., <http://www.keller.com/rgb.html>.

Node properties

A number of node properties can be directly encoded into the input file. Such properties include size of node, color and position. The following examples illustrate how to change various properties for a node called 'NodeA':

<i>Node name</i>	<i>Node size in units</i>
//NODESIZE NodeA	60

This sets the size of 'NodeA' to 60 units. Node sizes are rendered according to how many nodes are present in a graph; therefore, some experimentation may be required to determine an appropriate size.

Sometimes it is desirable for a node to be fixed during the layout process. It is also possible to take coordinates for all nodes generated in a different layout packages. To fix the position of a node, one adds X, Y and Z coordinates as shown below. In this case, 'NodeA' is assigned to a position of (100.0, 500.0, 400.0) in X, Y, Z space. The BioLayout *Express*^{3D} layout space is (1,000.0, 1,000.0, 1,000.0). If one desires a 2D layout, setting the Z coordinate to 500.0 will keep them on the central Z plane.

<i>Node name</i>	<i>Node position in X, Y, Z coordinates</i>		
//NODECOORD NodeA	100.0	500.0	400.0

Normalization method and platform dependency. BioLayout *Express*^{3D} does not possess the ability to normalize data, nor in principle does it matter whether the input data have been normalized, log-transformed or converted into ratio-metric data. A correlation matrix will be calculated and a graph plotted regardless. However, the size and structure of the graph will be highly influenced by these factors. For instance, the use of quantile normalization techniques such as RMA or gcRMA significantly reduces sample-to-sample and gene-to-gene variation compared with other normalization techniques, such as used for the Affymetrix MAS5 or GCOS approach, which are based on a data scaling. Less variation equates to higher correlations, and graphs of the same data normalized by two different approaches may have considerable differences in the overall size at a given correlation⁷. In particular, a graph of quantile normalized data would be larger at the same correlation cutoff, and low-intensity data, which is inherently noisy,

are more likely to be present in the graph because the approach reduces the variation in such data. Similarly, the variation between maximum and minimum values is compressed in log-transformed data, and therefore the data are less variable in terms of correlation measures. For these reasons, we use natural scale data for most analyses. Depending on the experiment size and samples to be analyzed, we have used different normalization strategies with a preference for quantile normalization methods where appropriate. Furthermore, BioLayout *Express*^{3D} is not restrained in analyzing data from any commercial or academic microarray platform; the input format is the same regardless of the platform the data were generated on.

Focus of the study. In contrast to a statistical approach to identifying genes of interest in a data set where the biological groupings and contrasts of interest need to be defined, the network paradigm

presents the structure in the data irrespective of the question asked. This is to say that data will be included in the graph based purely on whether it correlates to other data above the defined threshold. This can result in a significant proportion of the graph being composed of groups of genes that show no differential expression between sample groupings, but which are included in the graph by virtue of their expression profile being highly correlated with others. This can occur in cases where different probe sets for a given gene are represented multiple times on an array. Other examples include situations where genes are highly correlated in their expression but are not differentially expressed in the experimental contrasts. An example of such proteins would be the ribosomal proteins that are coordinately expressed and frequently form their own cluster(s) within graphs. Finally, the other major source of ‘noninteresting’ structure found in graphs is dependent on experimental variables. In situations where there is significant technical variation between data, this can give rise to spikes or troughs in measurements. This will have a marked effect on the correlation between probes affected by these issues, such that cliques of highly connected nodes in the graph may be formed due to a significant technical variation in one or more samples. The same is also true for biological variation. In clinical data, we have often found clusters of genes grouping due to their relative overexpression in one or a small number of samples. Often the nature of these gene groupings suggests that these spikes in the expression represent true biological variations, but they can also arise when one or a number of samples are ‘contaminated’ with other tissues. These relationships may not be the focus of the study and can be ignored, but their inclusion and visualization in one’s analysis at least provides a clearer insight into true diversity present in the data.

Graph clustering using the Markov clustering algorithm. Network graphs formed from expression data are often large and highly structured. This structure is a direct consequence of coordinate gene expression and the graphs provide an excellent interface to display and analyze these relationships. Integrated within BioLayout *Express*^{3D} is the Markov clustering algorithm (MCL), which represents a powerful approach to dividing graphs nonsubjectively into discrete chunks of genes sharing similarities in their expression, i.e., clusters. MCL has been shown to compare favorably with other commonly used algorithms in clustering of large graphs²¹ and as such it represents a robust state-of-the-art general purpose clustering algorithm, available also as a stand-alone open-source (GPL) software package. A full description of the MCL algorithm is provided elsewhere²².

Mining genes for overrepresentation of classes. A group of selected genes, e.g., a cluster, may be mined to see if they are enriched for a given annotation class. This approach is widely used elsewhere to explore gene lists in order to query them for their overrepresentation of certain classes of genes usually relating to gene function (e.g., GO terms)²³ or gene sets (previously derived gene lists)²⁴. This functionality is supported in BioLayout *Express*^{3D}.

Getting started with microarray gene expression analysis

Construction of an input file. The format of an input file for expression data is given in **Box 1**. In short, the minimum requirement for the program is one column (the first) of unique gene/probe identifiers followed by columns of data derived from individual

samples. The unique identifier column is searchable within the program as well as being used to support hyperlinking by a web search or linked to a specific website, and is also used for display purposes on graphs. It is therefore useful if this column contains information that supports these activities. In general, we have found that a concatenation of a gene symbol and probe ID provides a label that is both understandable and specific to the measurement. As mentioned above, BioLayout *Express*^{3D} is not restricted in the type of data that can be loaded and will accept unnormalized, normalized, natural scale, ratiometric or log-transformed data. However, for most purposes, we would recommend the use of normalized, natural scale data where technical variation has been minimized but contrasts between measurement values are maintained. Columns of data should be placed after the unique identifier, but should be ordered according to biological groupings. Although the order of the samples does not affect the Pearson correlation value and therefore resultant graph, it does affect the ease with which the expression profile of selected genes can be interpreted.

Input files can be also structured to allow the import and display of more information. Columns of annotation (class sets) may be placed between the unique identifier and the data columns. This annotation may take the form of a categorization in classes into which transcripts can be grouped, and these classes can be displayed on the graph. The program assigns a different display color to each class and color nodes according to the class they are in. Furthermore, there are built-in tools that allow selected groups of genes to be mined for over- or underrepresentation of specific classes (see Mining selected genes for overrepresentation of classes). Gene annotation may include GO terms, statistical lists, clusters, gene sets, pathway or protein family membership, etc. The import of classes with data therefore allows the overlay of information onto the graph and can provide a powerful aid to graph interpretation.

The number of probes on the array (rows) and samples (columns of data) is limited only by the ability to store the information in RAM, but tests suggest that for most configurations of modern computers it should be possible to work with hundreds of genome-wide arrays, i.e., with 20–50,000 probes worth of data. In principle, there is no need to filter the data before loading in BioLayout *Express*^{3D}. Structure in network graphs is made up of groups of genes whose expression is highly correlated and this is generally the most interesting aspect to any data set. However, one might wish to filter data before loading in order to remove low-intensity data or to remove genes whose expression does not alter over the experiment, i.e., ‘flat-line’ data. The need to do this depends on the experimental design and the hypothesis or question being addressed. Low-intensity data by its nature is noisy and therefore does not tend to correlate highly with other low-intensity data when the sample size is large (>20 arrays). Certain normalization methods, however, particularly quantile-based methods such as RMA and gcRMA, effectively regularize data by nature of their assigning of expression values to normalized rank values. With small data sets (< 20 arrays), especially where the biology is relatively similar across samples, many genes may not be changing in their expression level and their expression profiles are likely to show a high degree of correlation. This can therefore translate into a large network, even a high correlation cutoff, of genes that one is not interested in. Removal of this data or the statistical selection

of genes that are likely to be of high interest makes the graphs more manageable and focused on the research question.

Input files are tab delimited and can be assembled in a text editor or a spreadsheet software (e.g., Microsoft Excel). In order to recognize them as such, they should be saved with the extension ‘expression’. Examples of expression files are given on the BioLayout *Express*^{3D} website under the data sets section.

Worked example. In the PROCEDURE detailed below, we go through an example analysis of microarray expression data. The data used are the Genome Novartis Foundation (GNF) mouse tissue atlas²⁵. This analysis was carried out on a custom-designed Affymetrix GeneChip (named GNF1M) that possessed 36,182 probe sets designed to cover every known mouse gene and a

number of alternative transcripts of these. Run across these arrays was RNA derived from 61 different ‘tissues’ covering most major mouse organs and/or subdivisions thereof. The data set has been used and cited widely in a range of genomic investigations, including the work of the original paper describing this approach to the analysis of gene expression data and the tool BioLayout *Express*^{3D} (ref. 7). The data are derived from 122 arrays and represent a medium to large data set. However, the complexity of expression patterns that are to be found in this data due to cell/tissue-specific gene expression represents a significant challenge to analysis and visualization of the data. Indeed, it was the complexity of the analysis of this and related data that acted as the catalyst for the development of this tool. The data are also available to query from the GNF’s excellent BioGPS site (<http://biogps.gnf.org/>).

MATERIALS

EQUIPMENT

BioLayout *Express*^{3D} is platform independent and runs on Windows, Apple Mac or Unix operating systems (see **Box 2** for technical details). The *minimum* hardware requirements for the application running are:

- 512MB of main RAM
 - Single core CPU 1.5 GHz
 - Graphics card capable of OpenGL rendering
 - Monitor capable of displaying a 1,024 × 768 resolution
- With the above requirements, BioLayout *Express*^{3D} can process and display small- to medium-size graphs (< 5,000 nodes).

For processing and displaying large graphs (>5,000 nodes), the hardware specifications below are recommended for optimum performance:

- 2 GB of main RAM
- Dual-core CPU
- NVidia Geforce/Quadro series or ATI equivalent graphics card for OpenGL rendering
- Monitor capable of displaying a 1,600 × 1,200 resolution

The rule of thumb applies that the better/faster the hardware, the larger the graph that can be processed and displayed at acceptable speed/frame rates. In this protocol, the current limit for rendering large networks is somewhere around 30,000 nodes (3 million edges).

PROCEDURE

Data import

1| Download and install BioLayout *Express*^{3D}: The application should run on most PCs, Apple Macs and Linux systems. For PCs using Vista 32 bit or XP operating system, or Apple Macs running the Leopard system, we recommend downloading and using the installers available on the BioLayout *Express*^{3D} website. Java 1.6 is included in the PC installer; for Apple Macs, if not already installed, it will need to be. For Apple Macs running the Tiger operating system, which is only compatible with Java 1.5, there is also an installer available, although this will not contain the latest updates and optimizations. A jar file is also available that will run on all platforms. We recommend opening the jar file using a .bat/.cmd (PC), .command (Mac) or .sh (Linux) file using the script on the website. To install, go to BioLayout *Express*^{3D} website and select *Downloads*. Download the appropriate installation package or .jar file and install the package. Open BioLayout *Express*^{3D}.

2| Download data: Go to the BioLayout *Express*^{3D} website and select *Datasets*. Scroll down to data, click on GNF1M Mouse tissue atlas and download this ‘expression’ file. Decompress (unzip) the file.

BOX 2 | BioLayout *Express*^{3D} TECHNICAL DESCRIPTION

Implementation

BioLayout *Express*^{3D} uses many of the latest technologies in graphical rendering and network analysis to achieve its results.

- Usage of the Java 1.6 language for the main code implementation, providing OS-independent, multi-platform compatibility.
- Java 1.6 Generics and Iterator compliant code for data structure and graph handling.
- Full object-oriented programming (OOP) code is being used for graph modeling and processing, the graphic user interface (GUI) and keyboard/mouse event handling.
- Modified Fruchterman—Reingold^{27,28} layout algorithm for 2D/3D graph positioning and display.
- A heavily optimized C-based Markov clustering (MCL) algorithm for graph clustering (micans.org/mcl).
- For fast native OpenGL 2D/3D rendering engines, use the JOGL Sun API (please refer to the license section).
- Apache Xerces SAX XML parser for GraphML parsing and file creation.
- The scripting NSIS library provides installer support for the Windows platform with the added advantage of including the JRE in the package, making installing/uninstalling or running the application a seamless process to the end user (<http://nsis.sourceforge.net>).

Availability

BioLayout *Express*^{3D} is an open-source program distributed under a GNU public license (GPL).

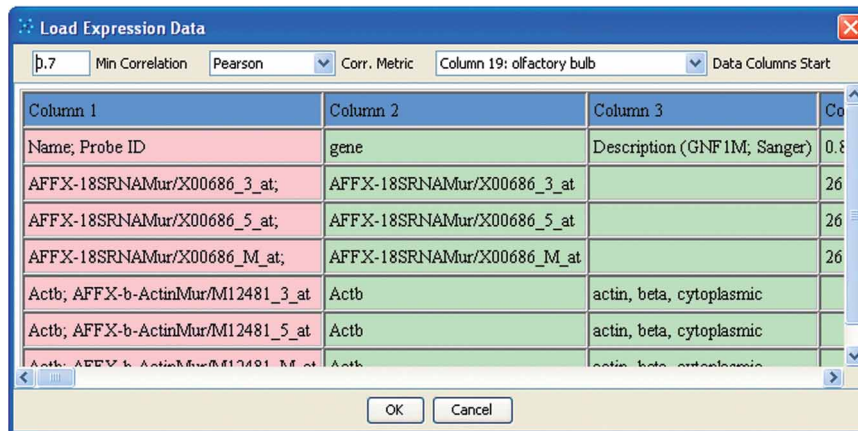


Figure 1 | The Load Expression Data dialog.

3| To open the file select *File* → *Open*. The *File Open* dialog will appear; find and select the file and click *Open*. Alternatively, a file may be opened by double-clicking on the file assuming that the file has an extension that is a recognized (associated) file type and the program is installed (as opposed to running the .jar file through the website script), or the file may be dropped into the BioLayout *Express*^{3D} window.

? TROUBLESHOOTING

4| The *Load Expression Data* dialog will appear (Fig. 1), then click OK. Generally, one will not need to change settings within this window and will be able to go straight to OK. However, there are a number of options that can be changed if required, as described in Box 3. After loading of the expression data into memory, if this data have not previously been loaded and a correlation matrix file does not already exist (in the same folder), the program will begin calculating a correlation matrix. The number of calculations that is necessary increases exponentially with the number of rows of the input file. A small file of just several thousand rows of data will therefore be calculated very quickly; this file with >36,000 probe sets will take a few minutes to perform the ~700 million calculations that are necessary to construct the matrix. However, once calculated, a correlation matrix (e.g., .pearson) file can be used for all future studies (assuming the expression file does not change in name, order or the number of probes or samples) and is kept in the same folder.

5| Once the correlation matrix file has been calculated (or an existing one located), the *Expression Graph Settings* dialog will appear. This presents two graphs derived from the data (Fig. 2). On the left of the dialog box is plotted a graph of the network size versus correlation threshold for the data. On the x axis is plotted the number of nodes and edges and on the y axis the correlation threshold range of the stored values. The two lines of dots represent the number of nodes (pink, lower) and edges (orange, higher) that would be included in the graph across the range of potentially selectable thresholds. The red vertical line denotes the currently selected value (default $r=0.85$) as determined by the slider at the bottom of the window. The lower the cutoff, the larger the graph. On the right of the dialog is plotted a graph of the distribution of node degrees (number of edges

BOX 3 | DATA INPUT DIALOG OPTIONS

min correlation

This refers to the correlation threshold above which correlations will be saved. A correlation matrix file can be very large if all correlations, i.e., -1 to +1 were saved. For example, a microarray of 50,000 probe features requires 1.25 billion calculations; therefore, only correlations above a certain value are saved, $r = 0.7$ being the default and suitable for most applications.

corr. metric.

For the work described here (and indeed for most work performed by the authors), the Pearson correlation measure has been used to generate the network graphs from expression data. However, in principle, it is possible to construct graphs based on any measure that results in a weighted edge between components and we have also implemented the Spearman rank correlation calculation as a selectable alternative to the Pearson correlation.

data columns start.

The program should recognize a file's structure (unique identifier, class/annotation columns and data columns) automatically, coloring them red, green and blue, respectively, based on the input format. However, this may not always happen, especially if the final column of annotation is a numeric. This dialog allows the user to override the automatic selection.



per node) at the selected threshold. Although not of immediate use, it does provide some clues as to the likely graph structure. Select the desired threshold, using the guidelines provided in **Box 4**, and when you have done so, click OK.

? TROUBLESHOOTING

First view of the data

6| *Navigation in the 3D interface: basic controls.* Graphs will first appear in the BioLayout Express^{3D} window and are rendered in 3D space (**Fig. 3**). Default settings will determine many of the aesthetic properties of the graph but before looking at how these may be changed to suit a user's preferences (as described in Step 7), navigate around the graph using the following controls:

- * *left mouse button* rotates of the current view;
- * *middle mouse button* allows sideways movement (translation) of the current view;
- * *right mouse button* for zoom in/out;
- * holding down *Shift* and *clicking the left mouse button* on a node makes it the center of the graph's axis of rotation;
- * holding down *Shift* while *dragging the left mouse button* is used to select nodes in the graph; and
- * holding down *Shift + Alt* while *dragging the left mouse button* is used to select even more nodes in the graph, without deselecting the previous ones.

In situations where a three-button mouse is not available, these commands are also available under the 3D menu bar.

▲ **CRITICAL STEP** Before continuing, explore the graph using these commands, as they are fundamental to the ability to interface with the data.

7| To alter the aesthetic characteristics of the 2D and 3D interfaces, use the settings described in **Box 5**.

▲ **CRITICAL STEP** For Mac and Linux users, the 'Command' key is used as an alternative to the Alt key.

? TROUBLESHOOTING

Graph clustering using the MCL

8| Go to Properties (Shift+P) and select the *MCL* tab. The top *Inflation* slider is the most important factor defining the 'granularity' of the clustering. A high inflation value, e.g., 4, will result in numerous small but 'clean' clusters, whereas a low inflation value of 1.5 will give fewer but generally less 'clean' clusters. For most purposes, we have found that an MCL inflation value between 1.7 and 2.2 works optimally. One other setting that is commonly used is the *Smallest Cluster Allowed*. In areas where networks are constructed from nodes with sparse connectivity, clusters tend to be small. This can result in the generation of many (hundreds) of small clusters that often surround and connect cliques of high connectivity. Setting *Smallest Cluster Allowed* to a number (usually between 3 and 10) will result in all clusters with sizes falling below that number being assigned to *No Class*. These clusters can then be easily filtered away (see "Removal of small clusters" in **Box 6**, which also presents additional options that can be used at various points during expression analysis).

9| Select an MCL inflation value, click OK and then go to the *Tools* menu and select bottom option *Cluster Using MCL*.

A clustering dialog will then appear to mark the progress of the operation. The bigger the graph and the lower the MCL

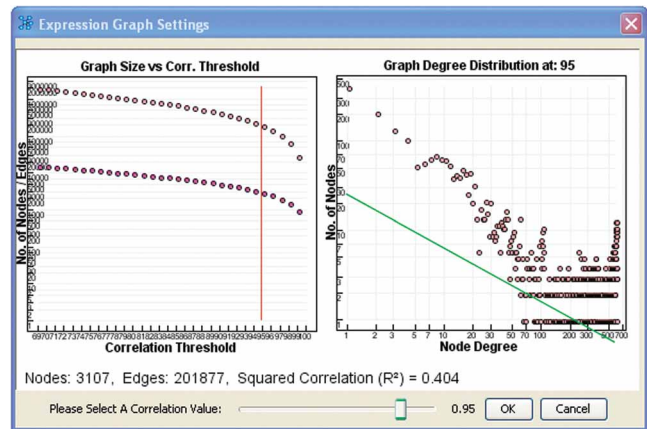


Figure 2 | The Expression Graph Settings dialog.



BOX 4 | CHOOSING A CORRELATION THRESHOLD VALUE: HARDWARE LIMITATIONS AND CONSTRAINTS

A threshold is chosen that balances graph size and complexity. The ideal graph is fully connected as every node has a measured degree of similarity with every other node, defined by correlation. In practice, however, such graphs are impractical because of their size (N^2) and because biological relationships of interest normally occur between nodes with very high degrees of correlation. Hence, some form of thresholding is generally desirable. The ideal threshold may be determined empirically by laying out graphs of increasing size but may also be determined by the maximum size of graph one is capable of rendering (if the program crashes after loading the data but before rendering the graph, see the Troubleshooting guide). Depending on the hardware configuration (in particular, the graphics card), graphs of up to 30,000 nodes or 3 million edges may be loaded. However, for most configurations, we recommend a working limit of half this. Above this size, one is more likely to experience issues in rendering. For this exercise, we use a setting of $r = 0.95$, resulting in a graph composed of 3,107 nodes 201,877 edges.

PROTOCOL

inflation value, the longer it will take. Following clustering, the graph will be re-centered and nodes will appear colored according to the cluster to which they belong. Clusters are numbered according to the number of nodes they contain (the largest cluster will be Cluster 1) and assigned an arbitrary color. A clustering at a given inflation value will be added to and displayed as a new *Class*, but will not be added to the input file.

Selecting nodes and viewing their properties

10 | There are number of basic ways to select nodes in a graph; hold down 'Shift' while dragging the mouse with left mouse button held down to select nodes in the graph. Selected genes will be highlighted in the graph by encirclement with a 'cage' (see **Fig. 4**). Hold down Shift+Alt while dragging the mouse with left mouse button held down to select additional nodes in the graph, without deselecting the previous ones. If a node belongs to a given class, e.g., cluster, use the command Ctrl+Alt+S to select nodes within the same class (see also *Selection Menu*). In order to find a specific gene or a class of genes (see later), select *Search* from the top menu bar and in the appropriate dialog box type in the gene of interest or select genes of a given class. Use Ctrl+A to select all nodes in the graph. Various attributes of the selected nodes may now be viewed.

▲ CRITICAL STEP For Mac users 'CMD/Apple' key is used as an alternative to the 'CTRL' key.

11 | Select nodes by one of the means discussed in Step 10 and open the *Class Viewer* (Ctrl+C). The left-hand side of the window will display the expression profile of the selected nodes scaled to the maximum expression value of those selected and the right hand side lists the identity of the selected genes and shows details of their associated properties/annotation/class membership (see **Fig. 4**). The view of the expression data may be plotted in *Log Scale*, shown as the mean expression

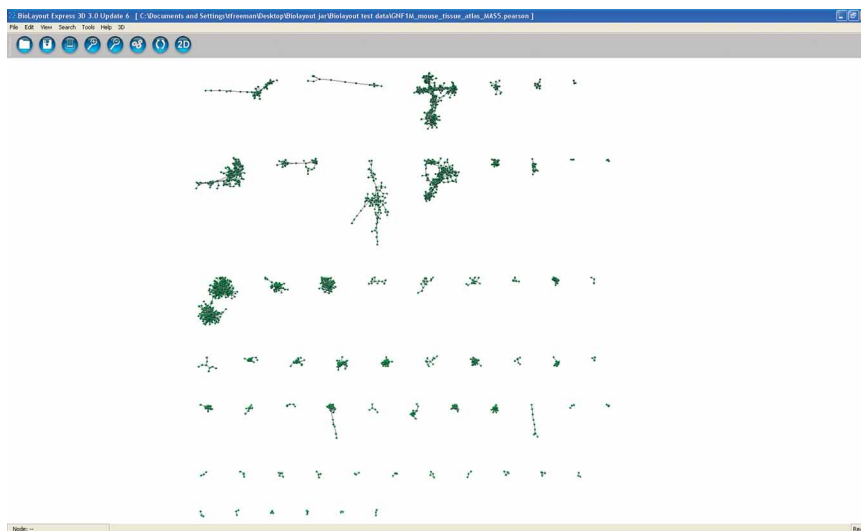


Figure 3 | The main BioLayout *Express*^{3D} graph window.

BOX 5 | PERSONALIZING YOUR DISPLAY

For a full description of options available to change the aesthetic characteristics of the 2D and 3D interfaces, please consult sections dealing with the *Properties* menu in the **Supplementary Manual**. We here detail just the most important variables, available to change by selection of the *Properties* toolbox by clicking on the icon on the menu bar or selecting it under *Tools*→*Properties* (Shift+P). This will give access to eight tabbed dialog boxes that allow the user to specify a range of options for personalizing the aesthetic characteristics of the graphs as well as options for their analysis. The most commonly customized aesthetic characteristics are highlighted below:

1. *Background color*. Click on the *General* tab and select *3D Background Colour*. Select any color from the palette provided. White or black is generally preferable but many other options are provided. (Similarly, this may be done for the 2D environment.)
2. *Tiled layout*. Under the *Layout* tab, one can choose to alter how the graph is laid out. As a default, graphs are laid out as a *Tiled Layout*. This is to say, individual graph components (groups of interconnected nodes that share no edges with other graph components) are 'tiled' side by side, with the component with the greatest diameter being initially placed in the top left-hand corner and the smallest component in the bottom right of the tiled graph. If this checkbox is clicked off, components are laid out in an organic manner and form a 'cloud' of components. For most applications, a *Tiled Layout* provides a more intuitive format.
3. *Minimum component size*. Also listed under the *Layout* tab is the option to alter this setting. Expression graphs potentially contain many small components (<5), which are formed through either chance correlations or more often arise due to a redundancy in the targets of certain probe sets. As such, they are generally not of interest and take up a lot of the plot space. They may be filtered out by setting this to an appropriate value. Changes to both the *Tiled Layout* and *Minimum Component Size* will only come into effect when new preferences are saved and the data are reloaded.
4. *Edges*. Default edges are colored to reflect the Pearson correlation that they represent. Hence, red edges represent a high correlation measure and blue a low correlation (relative to the range selected for display). Although sometimes useful to see, edges with a consistent color may produce a more aesthetically pleasing graph. To change click on *Edges* tab, select *Colour Edges By* and select *Colour*. Select your edge color of choice (preferably one that contrasts with the background color) and hit OK. *Edge Thickness* can also be adjusted here and for larger graphs (>1,000 nodes) one might find that thinner edges, e.g., 0.4 are preferable.
5. In order to ensure that alterations to the graph view are changed for future sessions select *Tools*→*Save Preferences* (Alt+P).

BOX 6 | OTHER USEFUL FUNCTIONS FOR EXPRESSION ANALYSIS

1. *Removal of small clusters (Step 8).* If a value has been entered in the *Smallest Cluster Allowed* box under the *MCL* tab, clusters below the size selected will not be assigned to cluster class, i.e., they will be listed as *No Class* for that clustering. Nodes belonging to small clusters often reside at the periphery of a graph and often represent genes only loosely related (perhaps by chance) to the main structure of the graph. If you select a node not assigned to a class (colored dark blue as default) and *Select Nodes Within The Same Class* (Ctrl+Alt+S), all nodes belonging to this class will be highlighted. To hide these nodes, go to *View*→*Hide Selected Nodes* (Ctrl+Shift+H). This will effectively give the graph a ‘haircut,’ removing nodes from around the central structure. These nodes may be added back to the graph by selecting *View*→*Unhide Nodes* (Ctrl+U).
2. *Filter by edge weight or number of edges.* Graphs can be filtered based on edge weight such that all edges below a set threshold and potentially the nodes that depend on them for their connectivity to the graph will be removed. Go to *Edit*→*Filter Edges By Weight* (Ctrl+Alt+W) and move the slider bar to desired setting. If the *Preview* button is checked, the changes to the graph will be visible, and if the *Also Hide/Unhide Nodes* box is checked, nodes will disappear when they no longer possess any edges above the selected threshold. Similarly, graphs may also be filtered to remove nodes based on the number of edges they possess. As nodes with a low node degree tend to be on the periphery of the main structure of a graph, this has the effect of giving a graph a ‘haircut,’ removing the nodes on the periphery of the main structure. Go to *Edit*→*Filter Edges By Weight* (Ctrl+W) and select the minimum node degree.
3. *Changing the visual properties of nodes.* Make a selection of a given group of nodes and open the *Properties* (Shift+P) window. The window should open directly on the *Nodes* tab where the properties of the selected nodes may be changed (if no nodes are selected, this menu will not be available). In this window, the *Shape*, *Node Size* and *Transparency* of the selected nodes may be altered. Try altering these settings and then press *Apply* or *OK*.
4. *Display of node labels.* There are number of options for displaying the names (labels) of nodes. *View*→*Show All Labels* (Ctrl+Shift+L) will display labels for all nodes and *Show Labels Of Selected Nodes* (Ctrl+L). To reverse and hide node labels, use either *Hide All Labels* (Ctrl+Alt+Shift+H) or *Hide Labels from Selected Nodes* (Ctrl+Alt+H).
5. *Collapsing nodes.* In order to simplify the view of a selected graph, nodes may be ‘collapsed’. The most common use of this function is when dealing with graphs with complex structure containing multiple clusters. Cluster the graph (or select a previous clustering) and then under the *Edit* menu select *Collapse Cluster By Class* (Ctrl+Alt+Shift+G). All nodes belonging to a given class will be collapsed into a single node where the diameter (volume) of the node is proportional to the number of nodes in the original class (cluster). The connectivity between clusters will be maintained and size of the spheres may be enlarged or reduced for aesthetic reasons using the commands Ctrl+> or Ctrl+<, respectively. The operation may be reversed by selecting *Edit*→*Uncollapse All Groups* (Ctrl+Alt+Shift+U).
6. *Image export.* To export images of expression graphs, select *3D* from the top menu and *Render Graph Image to File*. Files will be stored in the folder called ‘Screenshots’ in the same folder from which the data were launched. Files can be saved as .png or .jpg format. A high-definition image for presentation purposes can be generated by selecting *Render Hi Res Graph Image To File As...* under the *3D* menu. The resolution of the image can be adjusted using the slide bar found under *Properties*→*3D Rendering* listed as *High Resolution Image To File Render Option*. If set at high values, this operation may take a few seconds to complete.
7. *Saving graphs.* Entire graphs may be saved as a layout file whereby the plot coordinates are stored from the existing graph such that when a saved graph is reloaded, the time taken to layout the graph is omitted. When dealing with large graphs, this may represent a considerable saving in load time. If expression data are to be viewed from these graphs, the layout file must always be stored in the same folder. To save an entire graph, go to *File*→*Save Graph As...* (Ctrl+S). Similarly, subgraphs may be saved using *Save Selected Graph As...* and *Save Visible Graph As...* listed under the same *File* menu.

value of those selected (*Selection Mean*) or shown as mean values of individual classes represented in the selection (*Class Mean*) by clicking on the appropriate button above the expression window. The *Rescale* button rescales the data after calculation of the mean. *Grid Lines* may also be applied for better connection between data and sample. The list on the right-hand side of the screen by default lists all the selected nodes and displays their unique identifier, number of input and output edges (expression graphs are nondirectional; hence, input and output edges are equivalent) and the current class selected. If the graph has just been clustered, this will be the selected class.

12| Click on the box marked *View All Classes*. This will display all *Class* columns in the input file between the unique identifier and the data, plus the results of any clusterings that have been performed during the current analysis session. This may be a lot of columns of information and their simultaneous display may be undesirable. To select specific columns of interest, click on *Choose Columns To Hide* and uncheck those data that one does not wish to display and the display list will be automatically updated.

13| To browse the network based on class membership (this is particularly useful in assessing the results of a cluster analysis), click on *Find By Class*. If one wishes to browse the data, then select the first class (or cluster) and using *Next Class* one can browse the profile and content of those classes (clusters). This is a very fast way of reviewing the results clustering.

14| Having identified the ‘genes of interest,’ one can edit the selection by deselecting the boxes next to the gene/probe identifier (first column). Clicking *Refresh Selection in Table* will update the table and the expression graph accordingly.

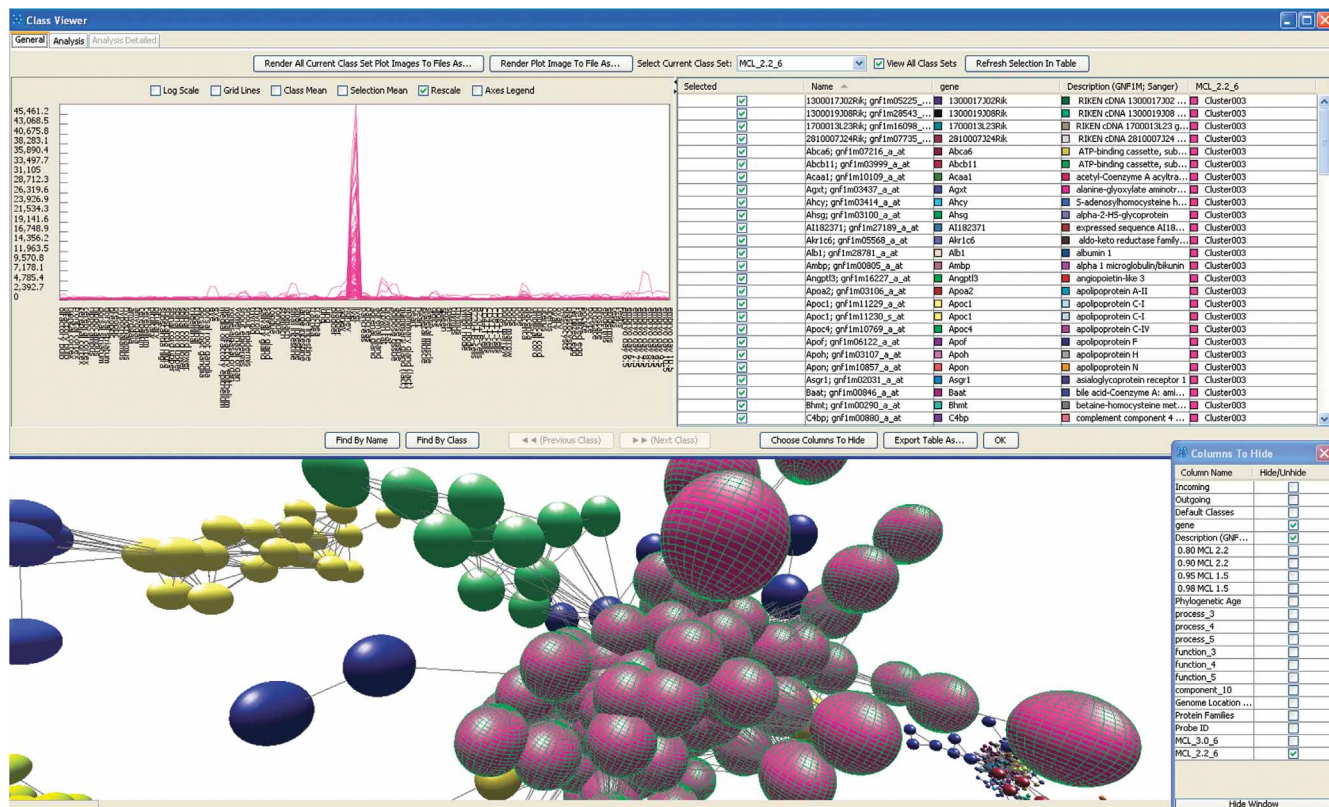


Figure 4 | The Class Viewer on top with the main BioLayout *Express*^{3D} graph window below. Selected nodes can be seen highlighted by a green 'cage'.

15 | To export a list of selected genes as a tab limited file, click on *Export List as...* and specify the location and name of the file to be saved.

Mining selected genes for overrepresentation of classes

16 | Select genes of interest and open *Cluster Viewer* (Ctrl+C) or select within this window. In the top right-hand corner click on the *Analysis* tab; the window will display all the available columns of annotation and class membership (the program does not differentiate between the two) and on the right the accumulated entropy score for each column.

17 | Select an annotation class of interest and click the *Details* button at the bottom of the page. Depending on the size of the gene list selected and the number of terms within the selected class, this may take a little time to calculate. The *Analysis Detailed* tab will appear and it will be populated with the frequencies of each term in the selected group of genes. A detailed description all the calculations performed and presented within this window is given in the **Supplementary Manual**, but in short, the most informative column is the *Adjusted Fisher's P value*, which gives a robust statistical score of the relative representation of each class term in the selected genes relative to the background of the entire chip. Overrepresentation of specific classes thereby provides clues to the biological significance of the selected gene. Clicking on the *Details For All* button in the *Analysis* tab will examine the relative enrichment for all terms in every class.

▲ CRITICAL STEP Although potentially useful, the number of calculations necessary to perform this task inevitably makes this process slow. A final cautionary note on mining data in this manner: if one is not working with a network derived from filtered data, one may have already been working with a biased selection and the mining of specific terms will be compared with this background, potentially invalidating the results. Similarly, when data include numerous probe sets designed for the same gene, all of which are likely to share the same annotation, enrichment of terms associated with these genes may appear to be artificially enriched if they are included multiple times in the same selection.

Working with other data formats; editing networks using GraphML input file

18 | BioLayout *Express*^{3D} supports the construction of network graphs from data imported in a number of standard graph formats (see **Box 1** for details). Below we describe some of the other possibilities for editing networks using a GraphML input file to illustrate some of these features. To download a sample file of macrophage signaling and effector pathways²⁶

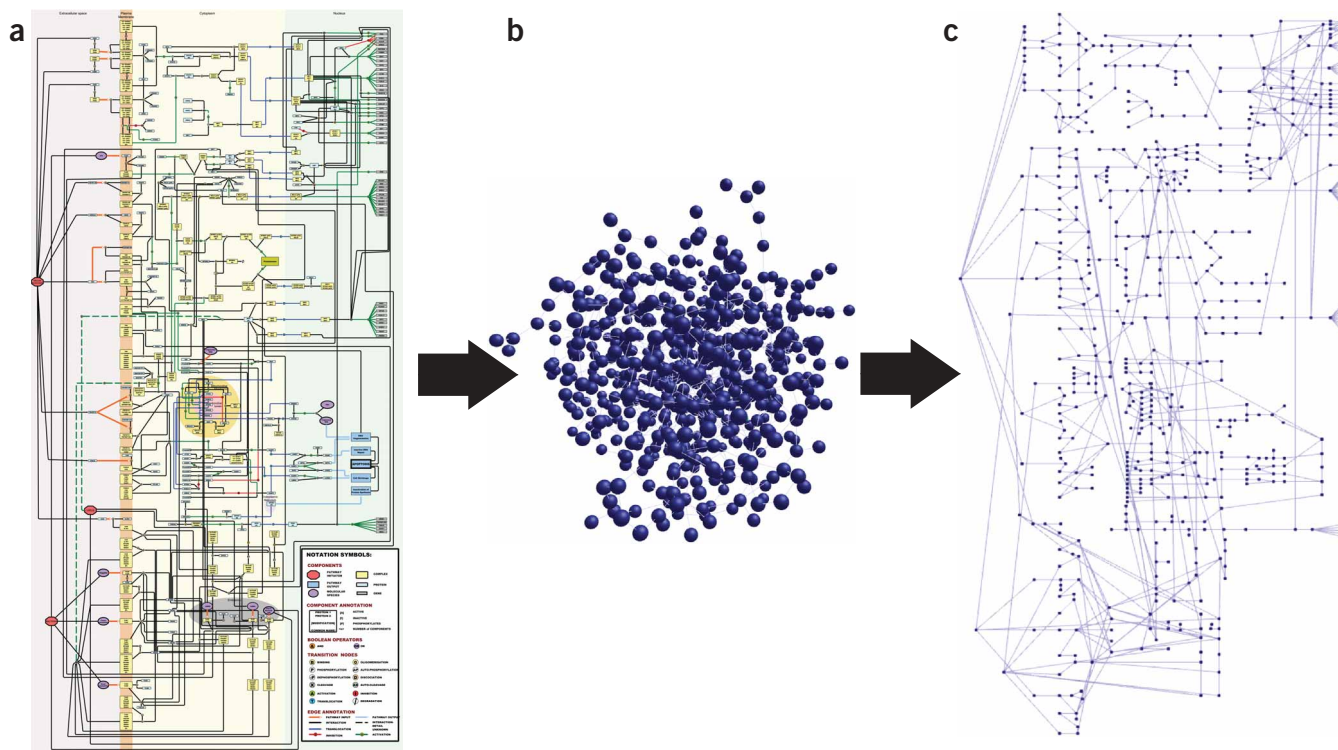


Figure 5 | Rendering of GraphML files in BioLayout *Express*^{3D}. (a) Pathway diagram encoded as a .graphml file and viewed within yEd graph editor (yFiles). (b) When loaded in BioLayout *Express*^{3D}, a .graphML parser imports all the graph and displays it in 3D as an organic layout. (c) If the 2D button on the menu bar is clicked, then the graph will be converted to a 2D representation where the original position of nodes in the graphML is used.

(Fig. 5a), go to the BioLayout *Express*^{3D} website and select *Downloads*. Scroll down to data sets, click on the pathway in GraphML file format and download. Decompress (unzip) file.

19 | To open the file, select *File*→*Open*. The *File Open* dialog will appear. Find and select file and click *Open*. Alternatively, a GraphML file may be dropped directly into the BioLayout *Express*^{3D} window.

20 | If opened in a 3D mode, a graph will be generated as shown in **Figure 5b**. At present, the BioLayout *Express*^{3D} GraphML parser imports all the graph details but will display only the component name in the 3D mode. To display component names, go to *View*→*Show All Labels*.

21 | If the 2D button on the menu bar is clicked, then the graph will be converted to a 2D representation of the 3D graph. We have, however, also implemented the import of data such that the original position of nodes in the GraphML is imported and may be used in the program's 2D mode. Open the *Properties* window (Shift+P), and under the *General* tab, click the text box *yEd-style rendering of GraphML imported files* (second row down under the subheading *General 2D Graph Settings*). The graph will then be rendered using the node locations encoded in the GraphML file (**Fig. 5c**). In this pathway diagram, the nodes in the graph represent different biological entities and various relationships between them. The various classes of nodes may be distinguished using a combination of node properties, namely node size, shape and color.

22 | There are a number of ways to select nodes for the editing of their properties. Nodes of a similar type may be searched for by name or part of their name. For example, there are multiple nodes in the example graph with the same name that represent processes, e.g., activation (A), inhibition (I), translocation (T) or logic nodes, e.g., AND, OR. To search for a given class of nodes, select *Search* from the top menu bar and select *Find By Name* (Ctrl+F). As an example, type A. This will select all nodes representing protein/gene activation.

23 | To change the visual properties of the selected nodes, open the *Properties Menu* (Shift+P) and the window will open on the *Nodes* tab. This menu provides the opportunity to change the nodes' shape (in 2D as well as in 3D), color, transparency as well as size. Try changing these settings and then click *Apply*. A dialog window will appear asking *Are you sure you want to Override Class Color?*. Click *Yes* and the changes made will take effect.

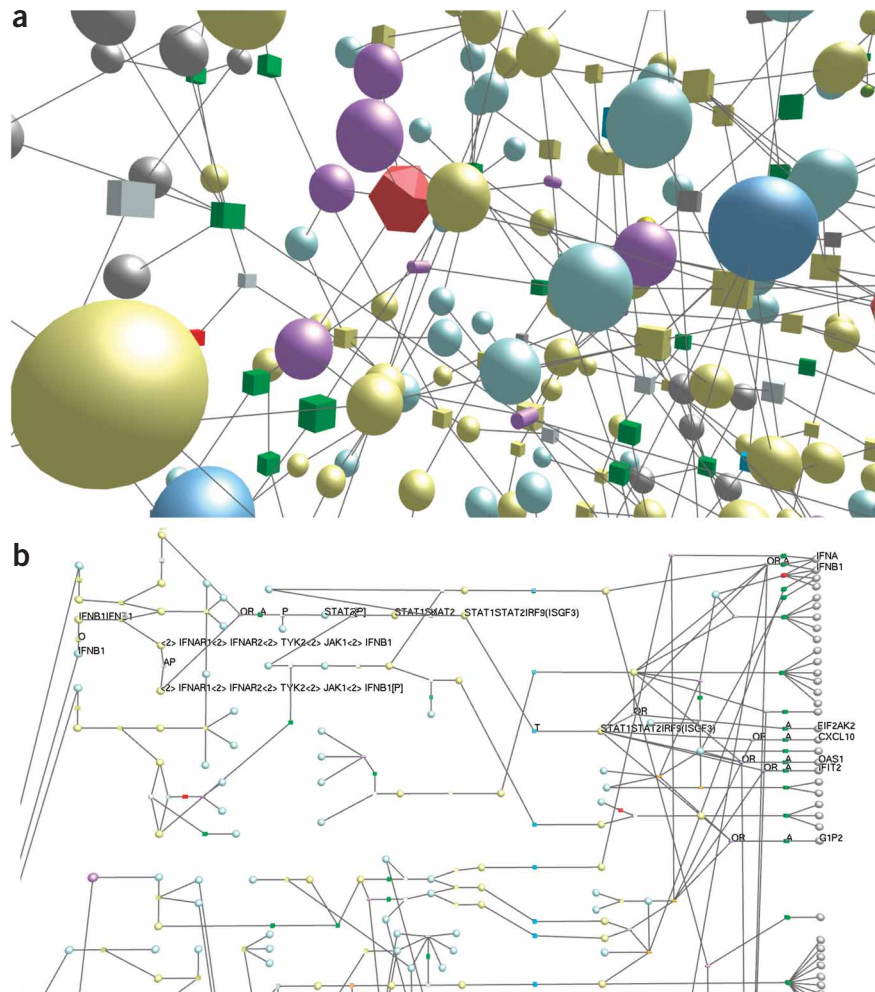


Figure 6 | Rendering of pathways in BioLayout Express^{3D}. (a) 3D mode and (b) 2D mode using node shape, size and color to distinguish between classes of nodes. In panel b, labels have been placed on nodes downstream of interferon- β (IFNB1).

24 | In this way, the visual characteristics of the graph may be changed such that nodes belonging to different classes may be assigned distinct visual properties. Saving the graph, *File*→*Save Graph As...* (Ctrl+S), will create a layout file where the changes to the node characteristics will be saved.

25 | In a similar way, nodes can also be assigned to classes such that we can select all nodes belonging to a given class of protein. Again using the macrophage activation pathway example of Raza *et al.*²⁶, select all the process nodes representing activation by selecting *Search* from the top menu bar and select *Find By*, type 'A.' Open the *Properties* window and go to the *Classes* tab. Go to *Create Class Set* and type in 'Process nodes', and in the *Create Class* box type 'Activation'. Then click *Apply*. Moving to the *Nodes* tab one may change the selected node's shape, size and color in 2D and 3D, and finally confirm their class membership by going down to the *Node Class* section of the *Nodes* tab and click on the *Containing Class* drop-down menu and select the newly defined class. When one clicks *Apply*, a dialog window will appear asking *Are you sure you want to Override Class Color?*. Click *Yes* and the changes made will take effect. This procedure may be repeated adding nodes to classes and changing their properties. A version of this pathway diagram where all the nodes in this pathway have been classified under node type and cellular location is also available on the BioLayout Express^{3D} website just below the GraphML file in the download section (**Fig. 6**). It is also worth noting that nodes can be assigned to classes by directly editing the input file (see 'Creation of Classes' in **Box 1**).

26 | In 2D mode, the directionality of edges may be shown by opening the *Properties* dialog box and, under the *General* tab, selecting *Directional Edges* under the heading *General 2D Graph Options*. The arrowhead size can be adjusted under the *Edges* tab.

27 | A useful feature when following the connectivity of components in a pathway diagram is the selection of parent (upstream) or children (downstream) nodes. To do this, select a node in the diagram, e.g., a pathway input such as IFNG and go to *Edit*→*Selection*→*Select Children* (Ctrl+Alt+C). The node(s) immediately downstream, i.e., output edges will be selected. Repeating this action will allow to follow the flow of connectivity. *Edit*→*Selection*→*Select All Children* (Ctrl+Alt+Shift+C) will select all nodes downstream of a node selection. Alternatively, nodes upstream (inputs) of nodes may be selected using the commands *Edit*→*Selection*→*Select Parents* (Ctrl+Alt+P) or *Select All Parents* (Ctrl+Alt+Shift+P).

● **TIMING**

Depending on network speed, it should take between 1 and 2 min to download BioLayout *Express*^{3D} and the data sets, and to install the program. On first opening the expression data, the program will calculate a Pearson correlation matrix. In case of the GNF1M data, corresponding to more than 36,000 probe sets, over 650 million individual calculations are required. Depending on hardware, this may take up to 6 min. Layout of large graphs, i.e., >10,000 to 30,000 nodes can take up to an hour, but for the small graph used in this example it should take only 10–30 s to render. Again the speed of graph clustering using the MCL algorithm is dependent on the size of graph but for this example it should take between 10 and 20 s. The remainder of the time it takes to go through this protocol is largely user-dependent, as all processes described will essentially occur instantaneously. For an experienced user, the protocols described here should take less than an hour to perform.

? **TROUBLESHOOTING**

Troubleshooting advice can be found in **Table 1**.

TABLE 1 | Troubleshooting guide.

Step	Problem	Possible reason	Possible solution
3	File does not load	In principle, files in any of the supported file types should load and result in a graph being displayed—the most likely explanation is therefore that the input file is incorrectly formatted or corrupted	Examples of all the supported file types are available on the BioLayout <i>Express</i> ^{3D} website. Try downloading the appropriate file type and running these. If this works, there is a problem with your file format. Check the file for missing data, data outside the normal bounds, bad formatting, etc. Also check the file extension as some Windows programs, e.g., Excel, may add a .txt extension when saving
5	Trying to load a large map and BioLayout <i>Express</i> ^{3D} crashes/does not work/is not responding	Large maps are very memory-consuming. Insufficient main RAM installed on the machine	Try running a smaller graph or using a machine with more installed main RAM. Visit the <i>Requirements</i> section of the BioLayout <i>Express</i> ^{3D} website
7	Experience rendering/image quality problems when using BioLayout <i>Express</i> ^{3D}	OpenGL drivers are out of date	Install the latest OpenGL drivers for the machine configuration
	Settings were not saved?	May not have tried to save the preferences before quitting BioLayout <i>Express</i> ^{3D} . Go to 'Tools→Save Preferences' before quitting the application	Restricted access to privileges on the machine that is running on BioLayout <i>Express</i> ^{3D} (university-managed desktops, shared machines in public areas). Unfortunately in this case, the application may not be able to save its settings because of the shared host running platform

ANTICIPATED RESULTS

Network graphs generated in BioLayout *Express*^{3D} from microarray gene expression data represent all the coexpression relationships that occur within a data set above the threshold level selected by the user. Clustering of the graphs breaks them down into units (modules) of coexpressed genes. This is not to say that all of these clusters will be made up of genes that are of interest to the user or that all of the clusters will represent 'real' coexpression clusters. Generally, biologists are interested in genes that show a high degree of differential expression between experimental conditions or biological groupings. These will be found in the graphs and indeed should form the majority of the graph, particularly when the data are viewed at high correlation threshold cutoffs and the data set is relatively large or diverse in the range of biology it represents. In the case of the GNF1M data used here as the example data set, the majority of clusters at higher correlation



thresholds are composed of genes that show a marked tissue- or cell-specific expression pattern. However, genes that share the same function and are tightly coexpressed, e.g., ribosomes, may also form cliques within a network, although they may not be necessarily differentially expressed. As the correlation threshold is dropped and graphs get larger, genes that are more weakly coexpressed but show little differential expression, i.e., that are part of the basal machinery of the cell, account for more of the graph structure. It should also be noted that technical artifacts, whether they be down to hybridization issues or contamination of samples with non-target tissues, will also form network cliques. In this way, BioLayout *Express*^{3D} allows for true exploration of the data, not just the statistically differentially expressed portion that is generally focused on. Overlay of other information, as imported as class-sets, allows this information to be viewed and mined for overrepresentation of particular biological groupings in the context of the graph. In this way, the graphs not only provide an interface for the exploration of large portions of the data but also support the integration of other information with these analyses.

Note: Supplementary information is available via the HTML version of this article.

ACKNOWLEDGMENTS We thank all those who have been involved with the development of BioLayout *Express*^{3D} over the years including Leon Goldovsky, Markus Brosch, Ildefonso Cases and Christos Ouzounis. We also thank the BBSRC who are currently funding the development of the program (BB/F003722/1) together with the Wellcome Trust (GR077040RP) who previously provided support.

AUTHOR CONTRIBUTIONS T.C.F., A.T., S.v.D. and A.J.E. wrote this paper. T.C.F. and A.T. conceived and designed the individual protocols.

Published online at <http://www.natureprotocols.com>.

Reprints and permissions information is available online at <http://npg.nature.com/reprintsandpermissions>.

1. Reed, J.L., Famili, I., Thiele, I. & Palsson, B.O. Towards multidimensional genome annotation. *Nat. Rev. Genet.* **7**, 130–141 (2006).
2. Kitano, H. Computational systems biology. *Nature* **420**, 206–210 (2002).
3. Nurse, P. Systems biology: understanding cells. *Nature* **424**, 883 (2003).
4. Cassman, M. Barriers to progress in systems biology. *Nature* **438**, 1079 (2005).
5. Enright, A.J., Van Dongen, S. & Ouzounis, C.A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30**, 1575–1584 (2002).
6. Enright, A.J., Kunin, V. & Ouzounis, C.A. Protein families and TRIBES in genome sequence space. *Nucleic Acids Res.* **31**, 4632–4638 (2003).
7. Freeman, T.C. *et al.* Construction, visualisation, and clustering of transcription networks from microarray expression data. *PLoS Comput. Biol.* **3**, 2032–2042 (2007).
8. Li, L., Stoeckert, C.J., Jr. & Roos, D.S. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.* **13**, 2178–2189 (2003).
9. Bader, G.D. & Enright, A.J. In *Bioinformatics: A Practical Analysis of Genes and Proteins* (ed. Baxeavanis, A.D.) 540 (John Wiley, New York, 2005).
10. Pavlopoulos, G.A. *et al.* Arena3D: visualization of biological networks in 3D. *BMC Syst. Biol.* **2**, 104 (2008).
11. Junker, B.H., Klukas, C. & Schreiber, F. VANTED: a system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics* **7**, 109 (2006).
12. Funahashi, A., Jouraku, A., Matsuoka, Y. & Kitano, H. Integration of CellDesigner and SABIO-RK. *In Silico Biol.* **7**, S81–S90 (2007).
13. Demir, E. *et al.* PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics* **18**, 996–1003 (2002).
14. Cline, M.S. *et al.* Integration of biological networks and gene expression data using Cytoscape. *Nat. Protoc.* **2**, 2366–2382 (2007).
15. Suderman, M. & Hallett, M. Tools for visually exploring biological networks. *Bioinformatics* **23**, 2651–2659 (2007).
16. Pavlopoulos, G.A., Wegener, A.-L. & Schneider, R. A survey of visualization tools for biological network analysis. *BioData Min.* **1**, 12 (2008).
17. Franke, L. *et al.* Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am. J. Hum. Genet.* **78**, 1011–1025 (2006).
18. Kim, S.K. *et al.* A gene expression map for *Caenorhabditis elegans*. *Science* **293**, 2087–2092 (2001).
19. Lee, H.K., Hsu, A.K., Sajdak, J., Qin, J. & Pavlidis, P. Coexpression analysis of human genes across many microarray data sets. *Genome Res.* **14**, 1085–1094 (2004).
20. Zhang, B. & Horvath, S. A general framework for weighted gene co-expression network analysis. *Stat. Appl. Genet. Mol. Biol.* **4**, Article 17 (2005).
21. Brohee, S. & van Helden, J. Evaluation of clustering algorithms for protein–protein interaction networks. *BMC Bioinformatics* **7**, 488 (2006).
22. van Dongen, S. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht (2000).
23. Dennis, G.S.B., Jr *et al.* DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biol.* **4**, P3 (2003).
24. Subramanian, A.T.P. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* **102**, 15545–50 (2005).
25. Su, A.I. *et al.* A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc. Natl. Acad. Sci. USA* **101**, 6062–7 (2004).
26. Raza, S. *et al.* A logic-based diagram of signalling pathways central to macrophage activation. *BMC Syst. Biol.* **2**, 36 (2008).
27. Fruchterman, T.M. & Rheingold, E.M. Graph drawing by force directed placement. *Softw. Exp. Pract.* **21**, 1129–1164 (1991).
28. Enright, A.J. *Analysis of Protein Function in Complete Genomes* PhD thesis, University of Cambridge (2003).