

Expander: from expression microarrays to networks and functions

Igor Ulitsky^{1,5}, Adi Maron-Katz¹, Seagull Shavit¹, Dorit Sagir², Chaim Linhart¹, Ran Elkon³, Amos Tanay⁴, Roded Sharan¹, Yosef Shiloh² & Ron Shamir¹

¹Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel. ²Department of Human Molecular Genetics and Biochemistry, Tel Aviv University, Tel Aviv, Israel. ³Division of Gene Regulation, The Netherlands Cancer Institute, Amsterdam, The Netherlands. ⁴Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. ⁵Current address: Whitehead Institute for Biomedical Research, Cambridge, Massachusetts, USA. Correspondence should be addressed to R.S. (rshamir@post.tau.ac.il).

Published online 28 January 2010; doi:10.1038/nprot.2009.230

A major challenge in the analysis of gene expression microarray data is to extract meaningful biological knowledge out of the huge volume of raw data. Expander (EXpression ANalyzer and DisplayER) is an integrated software platform for the analysis of gene expression data, which is freely available for academic use. It is designed to support all the stages of microarray data analysis, from raw data normalization to inference of transcriptional regulatory networks. The microarray analysis described in this protocol starts with importing the data into Expander 5.0 and is followed by normalization and filtering. Then, clustering and network-based analyses are performed. The gene groups identified are tested for enrichment in function (based on Gene Ontology), co-regulation (using transcription factor and microRNA target predictions) or co-location. The results of each analysis step can be visualized in a number of ways. The complete protocol can be executed in ~1 h.

INTRODUCTION

Gene expression microarrays provide accurate and highly accessible experimental data for functional genomics. Microarrays are available for a large variety of organisms and are used in a plethora of experimental designs. Powerful yet convenient tools for mining meaningful biological knowledge out of the raw data are essential for successful use of microarrays. Many algorithms were developed for various types of microarray data analysis, but most of them are inaccessible to nonspecialists, and others require extensive input preparation and output processing. Thus, integrative software tools that allow easy use of various algorithms within the same package are needed.

Expander 5.0 (<http://acgt.cs.tau.ac.il/expander>) is designed as a 'one-stop shop' tool that gives the user access to a range of microarray analysis algorithms covering the complete analysis process: processing of raw data, normalization, filtering, clustering, and functional, regulatory (i.e., promoter and 3'-untranslated region (UTR)), network and location analyses^{1–10}. The high-level analysis layer includes tests for enrichment of functionally related genes (based on gene characteristics recorded in Gene Ontology (GO, see **Box 1** for glossary)), co-regulated genes (using transcription factor (TF) and microRNA (miRNA) target predictions) and co-localized genes (i.e., genes that are close together in the genome). In addition, Expander implements custom enrichment analysis, which allows testing of gene groups against user-defined annotations. Finally, networks of protein interactions or signaling can be used for the detection and analysis of active gene modules. Expander 5.0 is pre-compiled with up-to-date sequence and annotation data for 13 model organisms (**Table 1**), and is freely available for academic users. Altogether, Expander provides a powerful software suite for comprehensive analysis of gene expression microarray experiments.

Comparison with other microarray data analysis tools

Many excellent alternatives are available for performing basic microarray data analysis^{11–18}. Some examples are GenePattern¹¹, dChip¹², TM4¹³, J-Express¹⁴ and GeneSpring. Each such tool

supports a distinct set of possible analyses, as shown in **Table 2**, but combining together the analysis steps from different tools requires considerable expertise and file format manipulation, and is error prone. Expander fully supports the entire basic gene expression analysis pipeline including normalization, multiple clustering algorithms and functional enrichment of small- to large-scale gene expression data sets. However, beyond the basics, Expander is unique in providing an integrated, comprehensive suite of high-level gene expression analysis tools. Using Expander, the user can apply state-of-the-art *cis*-regulatory sequence motif finding, miRNA target enrichment, network-based module finding, biclustering, co-location analysis and custom (user-defined) enrichment analysis, using unique and proven algorithms all in the same application.

The current limitations of Expander include a relatively narrow repertoire of normalization methods and the lack of support for classification and feature selection (**Table 2**). In addition, the methods for promoter and miRNA analysis currently support only known motifs of TF or miRNA binding, and the promoter and 3'-UTR data that are supplied by Expander. Finally, Expander is currently supported only on Windows and Linux platforms. We are continuing the development of Expander toward removing these limitations as well as adding new features.

This protocol is organized into three modules: loading and organizing expression data, identifying groups of co-regulated genes, and uncovering functions and regulatory networks (**Fig. 1**). Each of these analysis steps, as implemented in Expander, has proved useful in multiple studies, such as analysis of mRNA from yeast¹⁹, *Leishmania*²⁰, fly²¹, mouse⁷ and human^{22–24} and analysis of human miRNA data²⁵. Two murine microarray data sets are provided to illustrate this protocol: time course of bone marrow-derived macrophages (BMMs) exposed to various stimulators (taken from InnateImmunity-SystemsBiology website, <http://www.innateimmunity-systemsbiology.org/>), which will be referred to as EX1, and a comparison between wild-type and miR-155-deficient Th1 cells²⁶, which we will be referred to as EX2.

BOX 1 | GLOSSARY

- BMM—bone marrow-derived macrophages
- CDF—chip description file
- CLICK—CLuster Identification via Connectivity Kernels²
- FAME—Functional Assignment of MiRNAs via Enrichment (I. Ulitsky and R. Shamir, unpublished data)
- GO—Gene Ontology⁵⁰
- MATISSE—Module Analysis via Topology of Interactions and Similarity SETs¹⁰
- miRNA—microRNA
- PCA—principal component analysis³²
- PPI—protein–protein interaction
- PRIMA—PRomoter Integration in Microarray Analysis³
- PWM—position weight matrix
- RMA—robust multi-array average²⁷
- SAM—significance analysis of microarrays⁵⁴
- SAMBA—statistical-algorithmic method for bicluster analysis⁶
- SIF—simple interaction format
- SPIKE—Signaling Pathway Integrated Knowledge Engine⁸
- SOM—Self-Organizing Map³³
- TANGO—Tool for ANalysis of GO enrichments⁴
- TF—transcription factor
- TFBS—transcription factor binding site
- UTR—UnTranslated Region

Loading and organizing expression data. This section describes the ways in which data can be imported into Expander and then organized, normalized and filtered. The basic data representation

in Expander is a matrix, in which rows correspond to microarray *probes* and columns corresponds to studied *conditions* (samples). There are two ways to import microarray data into Expander.

TABLE 1 | Species and analyses supported in Expander.

Species	Name in Expander	Gene IDs	Supported analyses				
			GO analysis	Promoter analysis	miRNA analysis	Location analysis	Network analysis
<i>Homo sapiens</i>	Human	Entrez Gene	√	√	√	√	√*
<i>Mus musculus</i>	Mouse	Entrez Gene	√	√	√	√	√*
<i>Rattus norvegicus</i>	Rat	Entrez Gene	√	√		√	√*
<i>Gallus gallus</i>	Chicken	Ensembl	√	√		√	√
<i>Danio rerio</i>	Zebrafish	Ensembl	√	√		√	√
<i>Drosophila melanogaster</i>	Fly	Flybase	√	√	√	√	√*
<i>Caenorhabditis elegans</i>	C. elegans	Wormbase	√	√	√	√	√*
<i>Arabidopsis thaliana</i>	Arabidopsis	AGI	√	√		√	√*
<i>Solanum lycopersicum</i>	Tomato	Ensembl	√			√	√
<i>Listeria monocytogenes</i>	Listeria	Entrez Gene	√			√	√
<i>Schizosaccharomyces pombe</i>	S. pombe	Entrez Gene	√	√		√	√
<i>Saccharomyces cerevisiae</i>	S. cerevisiae	ORFs	√	√		√	√*
<i>Escherichia coli</i>	E. coli	Ensembl	√				√*













Name in Expander: the organism name as it appears in Expander's 'Load Study' dialog (Step 2).



Gene IDs: the gene identifiers used and expected by Expander for each organism. These identifiers must be used in user input data, unless a conversion file is provided (see Step 2).

√*in the network analysis column indicates that a sample protein–protein interaction network for the organism is available as part of Expander installation. Other networks can be loaded by the user.



TABLE 2 | Comparison to other tools.

Tool	Expander	GP	dChip	EP	TM4	GEPAS	Gen	Cyto	CT	GS	JE	GM
Software type												
Free for academic users	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Free for commercial users		✓	✓	✓	✓	✓		✓				
CEL file preprocessing	✓	✓	✓	✓	✓	✓				✓	✓	✓
Microarray image analysis			✓		✓							
Normalization	✓	✓	✓	✓	✓	✓				✓		✓
Principal component analysis	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓
Clustering	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Biclustering	✓			✓								
Classification		✓	✓		✓	✓	✓			✓		✓
GO analysis	✓	✓	✓	✓		✓		✓		✓	✓	✓
Chromosomal location analysis	✓	✓	✓	✓								
Network-based analysis	✓	✓						✓				
Network visualization	✓							✓		✓		
Promoter analysis	✓							✓				
MicroRNA analysis	✓											
Custom enrichment analysis	✓											

Software types:  stand-alone application,  web server. Compared tools: Expander; GP, GenePattern¹¹; dChip¹²; EP, EBI Expression Profiler¹⁵; TM4¹³; GEPAS¹⁶; Gen, Genesis¹⁷; Cyto, Cytoscape¹⁸; CT, Cluster/Treeview³¹ (<http://rana.lbl.gov/EisenSoftware.htm>); GS, GeneSpring; JE, J-Express¹⁴; GM, GeneMaths. Only tools that support multiple types of analyses, including some high-level analysis (e.g., GO analysis) were included in the comparison. Partek was not included in the comparison because of license restrictions. Comparisons of the individual algorithms available in Expander (CLICK, MATISSE, FAME, etc.) to other extant methods have been reported in detail in the original publications introducing these algorithms.

The first way is to directly load raw Affymetrix CEL files, which are then processed using the robust multi-array average (RMA) or the GC robust multi-array average method (utilizing their BioConductor implementation)²⁷. The second way is to load a processed text file containing the data matrix (**Box 2**). Expander allows significant flexibility in the structure of this text file. Such files can be obtained from low-level analysis programs and edited using any spreadsheet editor such as Microsoft Excel. In order to apply the diverse Expander annotation capabilities, the microarray probes must be mapped to organism-specific gene identifiers. Several mapping files are provided on the Expander download page and others can be obtained from various web-based services, such as BioMart²⁸.

The analysis of microarray data frequently requires normalization, to remove technical biases present in the data. Expander implements two methods for normalization of oligonucleotide arrays: nonlinear regression²⁹ and quantile normalization³⁰. Dual-channel array data loaded into Expander are assumed to be already

normalized (this is because Expander accepts such data as red/green intensity log ratios, while normalization requires separate raw intensities for these two channels).

Before running most high-level analysis algorithms, it is important to first identify those genes that have meaningful (i.e., non-static) gene expression patterns and filter out the rest. Expander provides several commonly used filters for this task, including filters based on fold-change factors, expression pattern variance, detection calls (for Affymetrix microarrays) or differential expression between condition subsets. Guidelines for the choice of a filter are described in **Box 3**. The set of probes that pass the filter can be used either directly as a gene group (e.g., if the microarray data contain cases and controls, and a *t*-test filter was applied) or further divided into groups using a gene grouping algorithm.

Throughout preprocessing, diverse views of the data are available, including box-plots, heat maps, hierarchical clustering³¹ and principal component analysis³².

PROTOCOL

Identifying groups of co-regulated genes.

In this section, we describe how to execute algorithms that identify probe/gene groups that share similar expression patterns. Three basic approaches for gene grouping are available in Expander as follows:

Clustering Clustering is the process of partitioning the probes into distinct groups (clusters) based on the similarity of their expression patterns across all the profiled conditions. The goal is to assign probes with similar expression patterns to the same cluster, and probes with dissimilar patterns to different clusters. The quality of the clustering solution reflects these two criteria, and is usually measured in terms of homogeneity and separation of the clusters. Expander implements some of the most widely used clustering algorithms: SOM³³, *k*-means³⁴ and CLuster Identification via Connectivity Kernels (CLICK), a graph-theoretic clustering algorithm that we developed^{1,2}. CLICK was shown to be superior to several other methods according to several figures of merit². In addition, Expander contains an implementation of hierarchical clustering, which organizes the probes (or the conditions) into a tree structure based on expression pattern similarities³¹.

Biclustering The basic assumptions of clustering algorithms are that co-expressed genes exhibit a global expression pattern similarity across the entire condition set, and that each gene belongs to only one group. This assumption becomes too restrictive when many diverse conditions (e.g., over 20) are studied. A *bicluster* is a set of genes that show significant similarity over a *subset* of the conditions. A *biclustering algorithm* can detect a collection of biclusters in a large gene expression data set. In this collection, genes or conditions can take part in more than one bicluster. Expander implements the Statistical-Algorithmic Method for Bicluster Analysis (SAMBA 2.0) biclustering algorithm, which can handle data sets with hundreds to thousands of conditions^{5,6,35}. Detailed examples of analysis using SAMBA have been described previously^{4,5}, and it is not included in the protocol presented here.

Network-based gene grouping Gene networks, such as protein–protein interaction (PPI) networks, can improve the interpretation of microarray data^{9,10,36–39}. In particular, by combining network and expression data, it is possible to identify *network modules*: connected subnetworks with similar expression patterns^{9,10,36–38}. Such modules are frequently more insightful than co-expression clusters, as they account also for the network relationships between the genes. Furthermore, the genes in such a module are more likely to be functionally related¹⁰. Expander contains an implementation of Module Analysis via Topology of Interactions and Similarity SETs (MATISSE), a graph-theoretic algorithm that detects significant co-expressed connected subnetworks^{9,10}.

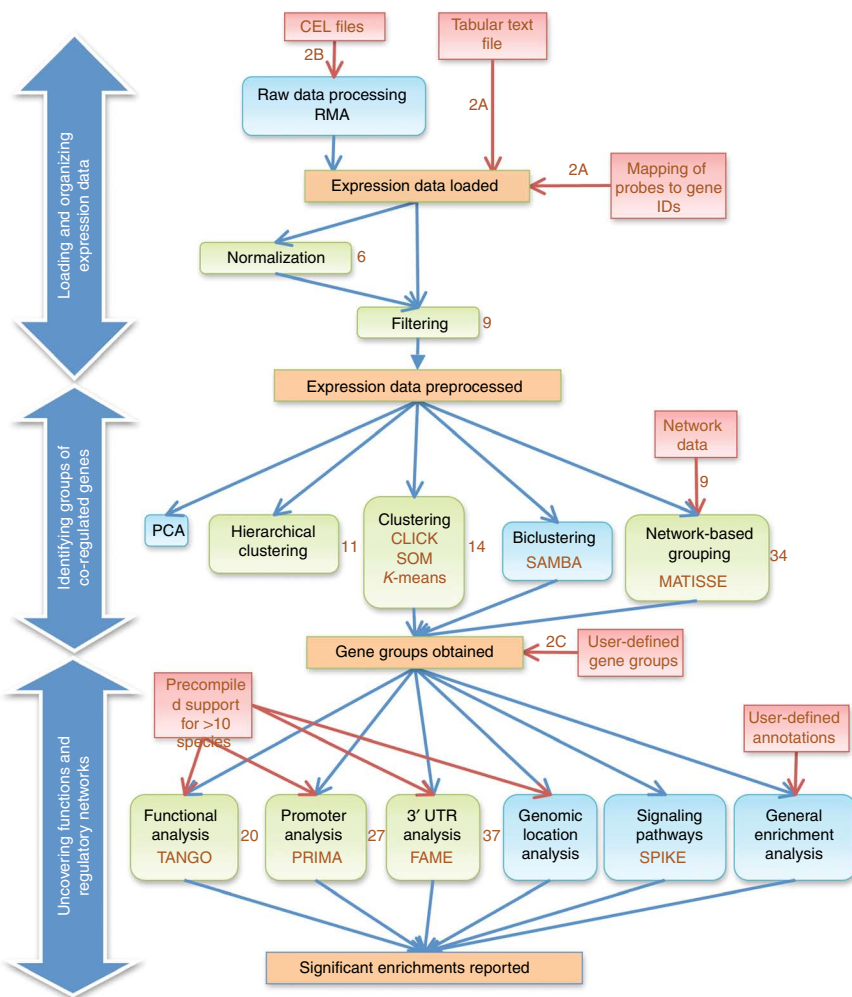


Figure 1 | Flow chart of the Expander workflow. The steps in green are included in this protocol, whereas the steps in blue represent additional Expander capabilities. Red rectangles indicate input files and the algorithms names are written in orange. Numbers next to arrows and rectangles indicate protocol step numbers.

In addition, it is also possible to load user-defined gene groups directly into Expander.

Uncovering functions and regulatory networks. This section describes how gene groups, identified in the previous section, or imported directly into Expander, can be analyzed and connected to functions and to regulatory networks. The following methods are available in Expander:

Functional analysis Successful analysis of microarray experiments is facilitated by the association of gene sets with biological functions. For example, when a cluster of co-expressed genes contains an unusually high proportion of genes sharing some annotation (e.g., a GO category), the other genes in the cluster can also be predicted to share this function. A major concern when testing functional enrichment for a gene set using GO is the thousands of repeated and related statistical tests that are performed for each gene group. Expander implements Tool for ANalysis of GO enrichments (TANGO)⁴, which addresses this problem by computing empirical *P*-values corrected for multiple testing and also filters similar annotations from the reported results (**Box 4**). TANGO

BOX 2 | DATA FORMATS USED IN EXPANDER

Tabular expression data

The most common input to Expander is a tabular text file, in which entries are separated by tab delimiters. Other formats can also be loaded using the 'Advanced Input Dialog,' which can be accessed by clicking the *Advanced* button in the 'Load Tabular Data' dialog box (Step 2A).

The data in the file should be in a matrix format, where rows correspond to probes and columns to studied conditions. The first row provides the *header*, which contains condition names (these could be any string). The first column provides the probe IDs and the second column provides the gene symbols. If gene symbols are not available from the software used to generate the file, this column can be omitted, and Expander will automatically assign gene symbols using Entrez Gene. In the first row, by default, the first two columns are ignored and can contain any string. For example, these are the first eight columns of the first four lines in BMM.txt:

Probeset	Gene Symbol	CPG_0	CPG_20	CPG_40	CPG_60	CPG_80	CPG_120
1415670_at	Copg	9.25	9.10	9.24	9.112	9.20	8.94
1415671_at	Atp6v0d1	11.17	11.25	11.16	11.10	11.07	10.91
1415672_at	Golga7	10.97	10.94	11.01	11.01	10.93	10.80

Using the 'Advanced Input Dialog' (**Figure 3A**), it is possible to specify that gene symbols appear in a different column or that the file does not contain a symbol column at all. In the latter case, measurements should start from the second column. It is also possible to specify that the header appears in a different row. In that case, the measurements should start from the row following the header row. Missing values should be presented as blank entries, dashes, 'NA' or 'NA'.

For Affymetrix oligonucleotide microarrays, the file may also contain information about the detection call for each measurement. If this is the case, one has to select the *File contains detection calls (A, M, P flags)* option in the 'Load Tabular Data' dialog box (Step 2A). The format of such a file is similar to the standard tabular file, but each condition is represented by two columns: the first containing the measurement and the second containing the detection call. In the header row, the title of each condition should be followed by the title of its detection column (which is ignored). A sample input file with detection calls, *'expressionWithDetection.txt'*, appears in the Expander home directory under *sample_input_files*.

Identifier (ID) conversion files

A conversion file is a tab-delimited file containing two columns. The first column should contain the probe ID exactly as it appears in the data file and the second column should contain the corresponding gene ID (e.g., an Entrez Gene ID for mouse and human genes). If no conversion is available for a probe, the second column can be either left blank or the probe can be omitted from the file. Note that several probe IDs can be mapped to the same gene ID. Several conversion files are available in the *organisms* directory under Expander home directory. This directory contains a separate subdirectory for each species, and within these directories the conversion files can be found in the *conversionFiles* subdirectory. **Supplementary Data 1** contains a sample conversion file, *MG430_2.0_Probe2EntrezGene.txt*.

Gene groups, clustering solutions and annotation files

The same format is used for loading gene groups (Step 2B) or annotation data, and for loading and saving clustering solutions. These files are tab-delimited files that contain two columns. The first column contains the probe/gene identifier (supported in Expander) and the second column the gene group/cluster (unique name or index). Network node IDs have to match the gene IDs supported by Expander (**Table 1**). Note that for a clustering solution, cluster 0 is reserved for probes that are left unclustered (singletons). A sample gene groups file, *geneSetsData1.txt*, is found in the *sample_input_files* directory under the Expander home directory.

SIF network file format

The SIF format is a straightforward and widely used format for definition of network data. A SIF file is a tab-delimited file, containing three columns. Each row represents a single network edge. The first column contains the ID of the source node and the third column contains the ID of the target node. The middle column specifies the interaction type. For a full list of possible interaction types see http://www.cytoscape.org/manual/Cytoscape2_5Manual.html. Expander currently distinguishes between edge types only for visualization purposes. Protein→DNA edges (denoted by 'pd' in SIF files) appear as directed edges, whereas all other interactions appear as undirected ones. Note that the network node IDs have to match the gene IDs supported by Expander (**Table 1**).

allows functional testing using the entire GO hierarchy and provides reliable and parsimonious results.

Promoter analysis Microarray experiments hold great potential for revealing the regulatory networks that determine the expression patterns of individual genes. As co-expressed genes are likely to be co-regulated, computational analysis of the promoter

sequences of co-expressed genes can identify the signatures of TFs that regulate specific transcriptional programs. Several algorithms for detecting overrepresented TF binding sites (TFBSs) have been successfully used to delineate transcriptional networks in organisms ranging from yeast to human^{3,34,40}. Expander uses PRomoter Integration in Microarray Analysis (PRIMA)³ for promoter analysis. Given the promoter sequences of the genes in each

BOX 3 | FILTERING PROBES USING EXPANDER

Probe filtering allows one to extract a set of informative probes from large gene expression studies. Filtering often facilitates more rapid and accurate downstream analysis. In addition, in many common scenarios, the set of all the genes passing a filter (e.g., *t*-test) is of interest by itself and can be used as an input into various Expander group analyses (e.g., Step 38).

Six probe filtering methods are available in Expander under the *Preprocessing*→*Filter Probes* menu. The selection of the filter depends on the experimental design used to generate the data (and on the level of statistical rigor desired from this initial step in the analysis):

A. Two or more subsets of conditions (and each subset contains at least two conditions) are compared. This scenario occurs if the experiment was designed for:

1. comparison of samples before and after some treatment (and replicates are available for both cases);
2. comparison of specimen from different genetic backgrounds. For example, mir155.txt in **Supplementary Data 1** represents a comparison between five WT and five mir-155-deficient Th1 cells; and
3. comparison of cohorts of cases and controls (e.g., in disease studies).

In this case, the *t*-test and SAM filters are appropriate. Expander allows the user to partition conditions in the data into subsets, and these filters test, for each probe, whether its expression levels differ among the condition subsets. During parameter setting (Step 9B), it is possible to specify whether the filter should be one-sided (looking for genes that are only upregulated or only downregulated in one subset compared with the other) or two-sided (retaining all differentially expressed genes). The *t*-test filter supports comparisons of two group subtypes, whereas SAM can be used to compare any number of condition subsets. For details on additional differences between SAM and *t*-test, see the SAM manuscript⁵⁴.

B. Data contain several time points after a specific treatment

In such cases, informative probes are those that show a considerable change relative to a reference time point (usually time point 0). For such data, the *Fold change* filter is appropriate. The fold change can be calculated in relation to (a) a selected baseline condition, (b) the minimal expression level of the probe, (c) the reference value, when working on relative intensities data in which a biologically meaningful reference sample was used in one of the channels, or on absolute intensities data that have already been divided before by some reference condition.

The fold-change filtering scheme is usually not statistically controlled. However, as the number of replicates in microarray experiments is typically very low, rigorous statistical tests usually have limited power only in detecting genes whose expression is significantly changed in the examined data set. Therefore, in many cases, applying fold-change filter gives better results, and statistical rigor of the entire analysis comes from subsequent enrichment tests.

C. Data compare different individuals, tissues or treatments

In this case, informative probes are those that show significant variance among conditions. In this case, there are two possible filters:

1. The *Variation* filter, which extracts the genes with the most variant patterns (*variation* is defined as variance for relative intensities data and as coefficient of variation for absolute intensity data).
2. The *Fold change* filter, in which the reference is defined as the minimal expression level of the probe (i.e., for each probe the maximum and the minimum values are compared).

D. Data were obtained using Affymetrix microarrays, and contain detection calls

In this case, it is possible to use the detection calls to filter out probes for which the fraction of conditions in which they were called as present ('P') is below a certain threshold. It is recommended to follow the application of this filter by one of the filtering steps described above.

E. Custom filter

Finally, the user can load any custom filtering file (a simple text file containing a probe name in each row) using *Preprocessing*→*Filter Probes*→*Load Probes Subset*.

target set (e.g., co-expression cluster) and of a background set (typically, all the genes represented on the microarray), PRIMA performs statistical tests aimed at identifying TFBS signatures that are significantly more prevalent in any of the target sets than in the background set. Expander provides promoter sequences for most of its supported species (**Table 1**) and the length of the promoter can be defined by the user.

miRNA analysis miRNAs are believed to have a major impact on mRNA expression^{41,42}. Analysis of 3'-UTR sequences of co-expressed genes can provide evidence of miRNA regulation^{41,43}. The Functional Assignment of MiRNAs via Enrichment (FAME) algorithm (I. Ulitsky and R. Shamir, unpublished observations), implemented in Expander, evaluates empirically the significance

of over- or underrepresentation of miRNA binding sites in the 3'-UTRs of gene group members. FAME uses TargetScan 5.0 predictions of the target sites of miRNA families⁴⁴, and utilizes target site context scores, which quantify the strength of the regulation conferred by each site⁴⁵. Importantly, FAME accounts for possible biases owing to 3'-UTR sequence length (**Box 5**).

Changes in transcription levels in the cell are determined by the activity of diverse signaling pathways that regulate the TF and miRNA activity. The analysis of gene expression in the context of these pathways can thus be insightful. To address this, Expander is integrated with the Signaling Pathway Integrated Knowledge Engine (SPIKE) knowledge base⁸, which contains diverse and rich information on human signaling pathways (**Box 6**). By invoking SPIKE from within Expander, it is possible to view the

BOX 4 | DETECTING FUNCTIONAL ENRICHMENT USING TANGO

Expander's functional enrichment analysis places the gene groups (such as the results of a clustering analysis) in a biological context. This is done by looking for statistically significant large intersection between a target group of genes and sets of genes that are annotated with some function. The standard statistical test for enrichment of a group for genes annotated with a particular function employs the hypergeometric (HG) distribution. According to this test (also called *Fisher's exact test for independence*), we are observing a background set of n genes, m of which are annotated with a certain function (the set A). Given a gene group T with m' genes, the probability that the intersection of T and A is of size k is

$$HG(n, m, m', k) = \binom{m}{k} \binom{n-m}{m'-k} / \binom{n}{m'}$$

The P -value for intersection of size k or larger between A and T is $\sum_{j \geq k} HG(n, m, m', j)$. TANGO uses this formula to identify significant enrichments (and PRIMA uses a generalized version of it).

When testing several sets A_i corresponding to different terms, and especially when the number of terms is large (e.g., when using the GO database), the P -values we derive have to be corrected, as we test many hypotheses and can obtain low P -values even if T is a random set of genes. If the sets A_i were independent, one could have applied a standard correction procedure (e.g., Bonferroni or FDR⁵⁵). However, when the sets A_i are highly dependent (e.g., GO terms 'cell cycle' and 'cell cycle regulation'), such correction may be too stringent. To cope with this problem, TANGO takes a simple approach, and computes the empirical distribution of the minimal annotation P -value by sampling a large number of random gene sets and computing their P -values versus each of the annotation sets. When annotating several gene sets T_i in a single analysis (e.g., all the clusters identified in a data set), TANGO also corrects for the additional multiple testing that takes place.

In practice, TANGO computes corrected P -values for a collection of gene sets T_j and an annotation database A_i by estimating the distribution of enrichment P -values in permuted genes sets of the same sizes. The idea is that we keep all of the relations among annotation sets A_i and among target sets T_j , but we decouple any dependency between them by applying a random permutation on the gene IDs used by the T_j s.

Filtering redundancies

GO is designed for maximum flexibility and provides a rich biological vocabulary that can support diverse species. As a result, when testing functional enrichment, one can detect several annotation terms with significant corrected P -values, such that all of the terms reflect essentially the same function or process. To avoid reporting such redundant terms, TANGO performs a greedy redundancy filtering procedure based on an approximated conditional HG test. Formally, given a target set T that is enriched with genes from the set A' , we test if T is enriched with genes from another set A , assuming we already know the size of intersection between A' and T and between A and A' :

$$\text{CondP}(T, A | A') = \sum_{k \geq |T \cap A \cap A'|} HG(|A'|, |A \cap A'|, |T \cap A'|, k) \times \sum_{l \geq |(T-A') \cap A|} HG(n - |A'|, |A - A'|, |T - A'|, l)$$

TANGO starts with all annotation terms that got a significant enrichment for a certain target set, sorted by their P -values $A_{i1}, A_{i2}, \dots, A_{ik}$. Then, the list is traversed and only sets A_j for which $\text{CondP}(T, A_j | A_i) < p_{\min}$ for all $j < i$ are reported, where p_{\min} is a pre-defined cutoff.

We note that if the analysis generating the target sets included only part of the genome (e.g., the ensemble of genes present on the microarray), one should perform all the analysis with an appropriately chosen background set, which can be specified in the *Background Set* panel in the TANGO, PRIMA and FAME configuration dialogs. Finally, note that TANGO tests for the significance of the overlap between gene groups and GO categories, unlike GSEA⁵⁶, which tests for enrichment of annotation sets in ranked lists of genes.

signaling interactions among the genes in a specific group and identify enrichment of specific pathway maps in gene groups. As we are currently working on significantly expanding the interface between Expander and SPIKE, these features will not be described as part of the protocol.

Finally, Expander also supports *location analysis*, in which gene groups can be tested for exhibiting unusually high concentration in their genomic locations (chromosome, chromosome arm or band), and *custom enrichment analysis*, in which gene groups can be tested using any user-defined annotations.

MATERIALS

EQUIPMENT SETUP

- PC with an internet connection
- *Hardware requirements.* Expander hardware requirements depend on the size of the expression data and on the organism that is analyzed. For a typical expression data set of 22,000 probes in 20 conditions, we recommend a ≥ 2 GHz CPU, 500 MB of available hard disk space, at least 1 GB of free physical RAM and a minimum screen resolution of 1,024 \times 768.
- *Operating system.* Expander is currently supported on the Windows XP or Vista and Linux operating systems.

- *Java 2 Runtime Environment (JRE)* version 5.0 or higher (<http://java.sun.com/javase/downloads/index.jsp>).
- *R Software* (R Project for Statistical Computing⁴⁶). It should be installed if the user wants to import CEL files or use the SAM filter.
- *Expander.* It has to be installed on a local computer by following the steps in **Box 7**.
- *Pre-compiled data for different organisms.* Pre-compiled data on gene symbols, genomic positions, functional annotations, promoter sequences and miRNA target predictions are available in organism-specific bundles. These data for the analyzed organism can be downloaded from within Expander, as described in **Box 7**.

BOX 5 | TESTING FOR MIRNA TARGET ENRICHMENT OR DEPLETION USING FAME

Expander's miRNA target enrichment tests help linking groups of co-expressed genes with possible miRNA regulation^{41,57}. In addition, if a miRNA's targets are underrepresented in such a group, it is likely that the genes have evolved to avoid miRNA targeting, e.g., because the miRNA and the group are highly expressed in some of the same conditions^{41,58}. A large number of algorithms for sequence-based prediction of miRNA targets have been described in the literature⁵⁹. Expander uses TargetScan 5.0 predictions of miRNA targets⁴⁴, as they were recently shown to be more accurate than other tools⁶⁰. These predictions are used to test whether the targets of some miRNA families (as defined in TargetScan) are enriched or depleted in any of the analyzed gene groups. The standard statistical method for identifying such enrichment or depletion is the hypergeometric test (used also in TANGO and PRIMA). However, this test treats equally all the predicted miRNA targets, while in reality target sites can be quite diverse in their efficacy. Several recent studies have shown that specific features of the 3'-UTR can influence the efficacy of individual miRNA target sites^{45,61}. These features can be combined into 'context scores' that can be used to assign confidence to the predicted targets of each miRNA. In addition, the hypergeometric test does not take into account the very uneven distribution of 3'-UTR lengths. The 3'-UTRs of genes expressed in some tissues, such as brain and the neural systems, are very long, whereas the 3'-UTRs of genes expressed in proliferating cells are much shorter^{58,62}. To identify significant over- or underrepresentation of miRNA targets in gene groups, Expander uses FAME (I. Ulitsky and R. Shamir, unpublished observations), a permutation-based statistical method that uses the confidence values for individual miRNA-target pairs and accounts for the number of miRNAs regulating each target. FAME computes an empirical *P*-value for each miRNA-gene group pair, which is more accurate than the hypergeometric *P*-value (the 'raw *P*-value', which is also computed by Expander). The accuracy of the FAME *P*-value is limited by the number of random iterations, which is specified by the user (e.g., if 100 random iterations are performed, the *P*-value cannot be below 0.01). The FAME settings dialog box allows the user to specify the following parameters (only parameters not shared by TANGO or PRIMA are explained here):

- Enrichment direction—as described above, underrepresentation of miRNA targets in a set of genes can often be even more informative than over-representation. Select the direction of the enrichment using this dropdown menu.
- Use context score—this option specifies whether FAME will use the context scores (confidence scores for individual miRNA target sites⁴⁵) to weight miRNA-target pairs in FAME.
- Minimal overlap between targets and group—it specifies the minimum number of predicted miRNA targets that need to appear in the group, in order for the miRNA to be considered by FAME.

- **Data files.** This protocol begins with an expression data set to be analyzed. Expander installation is accompanied by several sample files, which are described in the Expander manual. Loading new data into Expander is described in Step 2. Several data files are available in Supplementary Data 1–4 for readers wishing to follow this protocol as a tutorial:
 - BMM.txt contains a microarray data set constructed by the Innate Immunity Systems Biology project, in which expression profiles were recorded in murine BMMs at several time points after exposure to six agents. We will refer to this data set as EX1 (**Supplementary Data 1**).
 - MG430_2.0_Probe2EntrezGene.txt contains a mapping of Affymetrix

Murine Genome (MG) U430 2.0 probes to Entrez Gene identifiers, taken from BioMart²⁸ (**Supplementary Data 2**).

- Mir155.txt contains a microarray data set developed by Rodriguez *et al.*²⁶, in which five repeats of Th1 cells deficient for mir-155 are compared with five controls (obtained from the ArrayExpress database, accession number E-TABM-232). We will refer to this data set as EX2 (**Supplementary Data 3**).
- mouse.IntAct.sif, a mouse PPI network taken from the IntAct database⁴⁷ in simple interaction format (SIF) (**Supplementary Data 4**) (see **Box 2** for format specifications).
- Additional data files and accompanying step-by-step tutorials are available at Expander homepage (<http://acgt.cs.tau.ac.il/expander>) under the heading 'Hands on.'

BOX 6 | SPIKE

Cellular responses and processes are coordinated by diverse signaling pathways, data on which are constantly accumulating. It is frequently desirable to analyze gene expression data in the context of these pathways. Expander installation includes the SPIKE knowledge-base of human signaling pathways⁸. SPIKE includes an extensive database of human signaling interactions curated from the literature, with a specific focus on DNA damage response, as well as interactions from other pathway databases, including KEGG⁶³ and Reactome⁶⁴. Using SPIKE, it is possible to analyze either entire gene groups or the subset of a group that shares a GO annotation. Several types of analysis are possible for each such set of genes:

- View the known signaling interactions among genes in the set. The user can view a map of interactions and explore additional signaling interactions involving the genes using SPIKE's dynamic network exploration capabilities⁸.
- Identify intermediate signaling pathway members that connect the genes in the cell. It is possible to add to the gene set additional genes that lie on shortest paths among the selected genes.
- Test whether the gene set is enriched for genes from pathway maps defined in SPIKE. If such enrichment is found, it is possible to view the pathway and highlight the genes in it that belong to the gene set.

Analysis using SPIKE can be initiated by selecting in the *Group Analysis* menu *Network*→*SPIKE*. Full details on the analysis are available in the Expander manual.

BOX 7 | EXPANDER INSTALLATION

1. Go to the Expander homepage (<http://acgt.cs.tau.ac.il/expander>) and click on the *Download* link.
2. Fill in the details on the registration page. Read carefully the Expander license and accept its terms.
3. Download the Expander installation package for the operating system (Windows or Linux).
4. After downloading, unzip the installation bundle and place its contents in a directory on the computer. When done, make careful note of the directory in which Expander is installed: this will be the *Expander home directory*.
- ▲ **CRITICAL STEP** Owing to a problem in modules that use the R program, the path of the Expander directory should not contain spaces.
5. *Optional step:* Download pre-compiled annotation files for the organisms that will be studied. To follow this protocol as a tutorial, one will need to download the annotation files for mouse. The files for each species are available as a separate .zip archive. After downloading, place the contents of the .zip files into the *organisms* subdirectory in the Expander home directory. Alternatively, you can download and extract the files automatically from within the software (see Step 9).
6. *Optional step:* In order to perform CEL file preprocessing and use the SAM filter (both of which are not part of this protocol), one will need to install R, a free software environment for statistical computing and graphics. After installing R, do the following to install the Bioconductor *affy*, *aroma.affymetrix* and *samr* packages:
 - A. Run R.
 - B. In the R window, type the following lines:
 - i. `source("http://bioconductor.org/biocLite.R")`
 - ii. `biocLite("affy")`
 - iii. `install.packages("samr")`
 - iv. `source("http://www.braju.com/R/hbLite.R")`
 - v. `hbInstall("aroma.affymetrix")`
7. If using the Linux operating system, ensure that one has *rwX* permission for the Expander home directory and for the directory in which the data are located. Also ensure that one has *rx* permissions for all '.exe' files in the Expander directory.
8. Navigate to the Expander home directory. Two execution files are available for Expander. If the PC contains at least 2 GB of RAM, double click `Expander_2GB.bat` (under Microsoft Windows) or use `./Expander_2GB.bat` (under Linux). Otherwise, double click `Expander.bat`.
9. If not done so in Step 5, download the annotations files for the species that will be analyzed. In the *Help* menu, click on *Download Data for Organism*, select an organism and press *OK*. To follow this protocol as a tutorial with EX1 (BMM.txt) and EX2 (mir155.txt) (**Supplementary Data 1**), download the mouse annotation files. Once the annotation data are downloaded, they will remain installed and will be available during subsequent executions of Expander.
10. For more information, refer to the manual on the Expander homepage.

PROCEDURE

Import of expression data

1| Follow the steps in **Box 7** to install and execute Expander and to download the pre-compiled annotation data for the organism that you wish to analyze (mouse, if one would like to follow this protocol as a tutorial).

▲ **CRITICAL STEP** Organism-specific annotation data have to be installed and placed at the right directory for successful execution of Steps 20–39.

? TROUBLESHOOTING

2| After Expander is started, the Expander desktop image (the frame of **Fig. 2**) can be seen. Before data are loaded, the main window remains blank. In the beginning of a session, one has to load data into Expander. Three possible data types can be imported: a tabular data file (option A), a collection of CEL files (option B) or a collection of gene groups (option C).

(A) Tabular data file

- (i) A tabular data file is a tab- or space-delimited file describing gene expression measurements (for details on the format, see **Box 2**), which can be obtained from various image processing and low-level analysis programs, and edited using any spreadsheet editor, such as Microsoft Excel. Data has to be assembled in a tabular file, as described in **Box 2** (to follow the protocol as a tutorial, BMM.txt in **Supplementary Data 1** can be used).
- (ii) To load the file, select the *File* menu and then *New Session*→*Expression Data*→*Tabular Data File*. A 'Load Tabular Data' dialog box will appear.
- (iii) Select the desired species in the *Organism* pull-down menu. For EX1, select *mouse*.
- (iv) In the *Raw data file* field, specify the location and the name of the tabular text file (specify the full path). To use this protocol as a tutorial, load BMM.txt (EX1) or mir155.txt (EX2).
- (v) For each organism, Expander supports the most common gene identifier type, which is specified in **Table 1**. Each row in the data file can correspond either to a microarray probe or to a single gene. If each row in the data

file corresponds to a probe, in order to perform the analyses in Steps 20–39, one has to load a simple two-column file that contains a mapping of probes to the gene identifiers supported by Expander for the analyzed species (see **Box 2** for file format). For EX1, such a mapping file for Affymetrix MG U430 2.0 microarray is available in **Supplementary Data 1** (MG430_2.0_Probe2EntrezGene.txt). Similar files for several popular microarrays are available for download on the Expander download page (see the Expander manual for details). Select the *IDs conversion file* option and specify the full path of the mapping file in the corresponding field. Alternatively, if rows in the data file correspond to genes rather than probes (and the first column contains gene IDs), select the *Use probe IDs as gene IDs* option (do not select this option for EX1 or EX2).

▲ CRITICAL STEP Failure to properly match probes to gene identifiers will lead to failure in executing Steps 20–39.

- (vi) Expander supports two types of data files, as described below. When loading data one needs to specify the type of array analyzed. Experiments using single-channel technology, usually based on oligonucleotide arrays (such as Affymetrix or Illumina arrays), measure the absolute intensities of the probes on the array. These values are always positive. If the data describe a single-channel array experiment, select *Absolute Intensities* option in the *Data type* dropdown menu. The array intensities reported in the file can be either raw or base 2 log-transformed (depending on the software that was used to produce the data file). The range of raw intensity values is typically 1–40,000, whereas log-transformed values are in the range of 0–15. Select the scale of the data in the *Data scale* dropdown submenu. For EX1 or EX2, select *Absolute Intensities* as data type and *Log2* as data scale. Dual-channel technology is used in spotted cDNA arrays as well as most of Agilent arrays. In a data file resulting from dual-channel arrays, each measurement corresponds to the logarithm of the ratio of the intensities of the two channels. These values are both positive and negative and usually fall in the range of –10 to 10. If the data file describes a dual-channel experiment, select *Relative Intensities* in the *Data type* dropdown menu.

▲ CRITICAL STEP The array type affects various aspects of data visualization and normalization in Expander. The specified data scale affects various aspects of data normalization. If the measurements in the file are log-transformed and this is not specified, the flooring procedure (see below) is likely to discard most of the measurements, significantly affecting all subsequent analyses.

- (vii) If the data taken from an Affymetrix array, detection calls may appear as separate columns in the tabular file (for details, see **Box 2**). If the file contains such columns, select the *File contains detection calls* option. For EX1 or EX2, leave this option unselected.
- (viii) If the file contains missing values, they will be imputed by Expander upon loading the data. There are two options for imputing missing data. First, set the missing values to a specified value by selecting *Set missing values to* and fill the appropriate field. By default, the missing values in ‘Absolute Intensities’ data files are set to 40 (5.3 if the data are in log scale). Select this option for EX1 or EX2. Another option is to estimate missing values using the K-Nearest Neighbors (KNN) algorithm⁴⁸ by selecting *Estimate missing values with KNN*.
- (ix) For data files containing absolute intensities, a flooring procedure will set all the values below a certain threshold (40 for raw values and 5.3 for log2 data by default) to that threshold. Specify the flooring threshold in the *Floor value* field. For EX1 or EX2, use the default parameters.

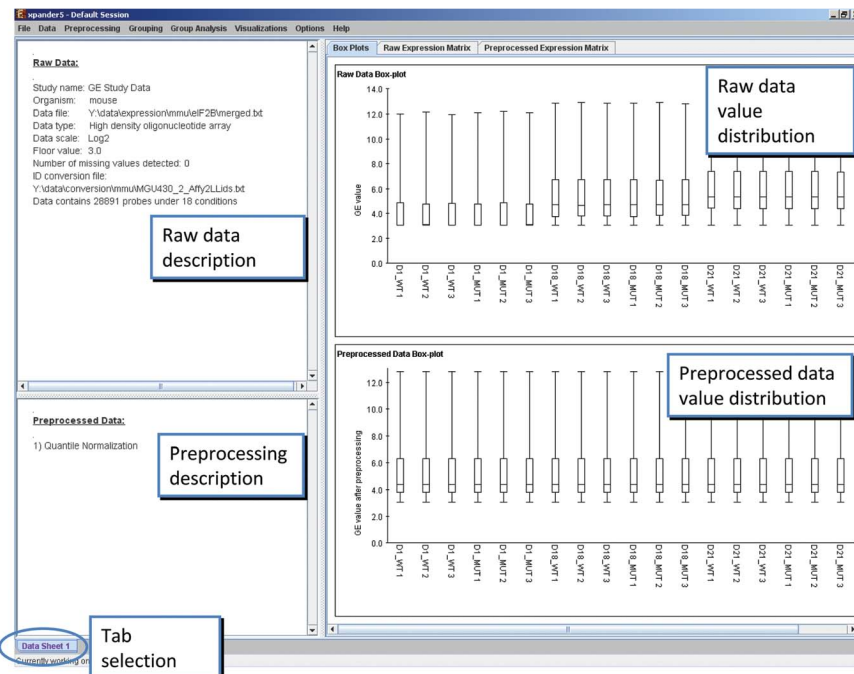


Figure 2 | Data sheet view. The top left panel contains the description of the loaded data. The bottom left panel contains a log of the preprocessing steps that were done. The right panel contains several views describing the entire data set. The displayed tab describes the distribution of the intensity values in each condition in the raw and in the preprocessed data.

(x) Click on the *Advanced* button. An 'Advanced Input Dialog' will appear (**Fig. 3a**). This dialog box can be used for loading files that deviate from the standard tab-delimited format (**Box 2**). In the bottom part of the dialog, the first few rows and columns of the data are displayed in a table, showing the way the data will be parsed by Expander, based on the current parameter settings. If this table has red background, it indicates a problem with data input. The row containing the header information (from which the condition names will be taken), the first data row and the columns specifying the probe/gene IDs and the gene symbols can be altered. The first data column is automatically set to the column following the gene symbols column. If additional columns should be ignored, specify some part of their names in the *Ignore columns* fields. After setting the parameters, press the *Preview* button. The preview will be updated. Both the sample files, EX1 and EX2, follow the standard tab-delimited file format, and no parameter adjustments are required in order to load them.

? TROUBLESHOOTING

(xi) Select the *Auto-fill symbols* option. This will enable Expander to use its gene database to update the gene symbols corresponding to each probe. This option is helpful when the data file does not contain a gene symbol column. Gene symbols appear in various Expander visualizations, such as heat maps (**Fig. 3b**) and probe lists. After finishing specifying the file format, press *OK* to close the Advanced Input Dialog and then press *OK* again to load the data.

(B) Affymetrix CEL files

- (i) Before loading CEL files, ensure that R software along with the Bioconductor 'affy' package (**Box 7**) has been installed. An open internet connection is also required for this operation.
- (ii) In the *File* menu, select *New Session*→*Expression Data*→*CEL files*. A dialog box will appear. Specify the species and the type of the microarray for the CEL files.
- (iii) Enter the folder where the CEL files are located into the *Files location* field. Specify the name and the location of the tabular file that will be generated from the processing CEL files in the *Save data into file* field.
- (iv) For new-generation Affymetrix chips, which are not part of the 3' Gene Expression chip family, the chip description file (CDF) has to be provided, which can be obtained from <http://bioinf.wehi.edu.au/folders/mrobinson/CDF/> or <http://www.affymetrix.com/support/technical/libraryfilesmain.affx>. It is also possible to use custom CDF files for 3' gene expression microarrays (see Expander manual for details).
- (v) To view the detection calls in Expander, select the *Retrieve detection calls* option (available only for the 3' Gene Expression chip family).
- (vi) Press *OK*.
- (vii) If Expander cannot find the location of the installation of R, specify the location. Browse to the location of the R software (the full path of the R.exe file). In Windows, R.exe file is likely to be located in the *bin* folder of R software.

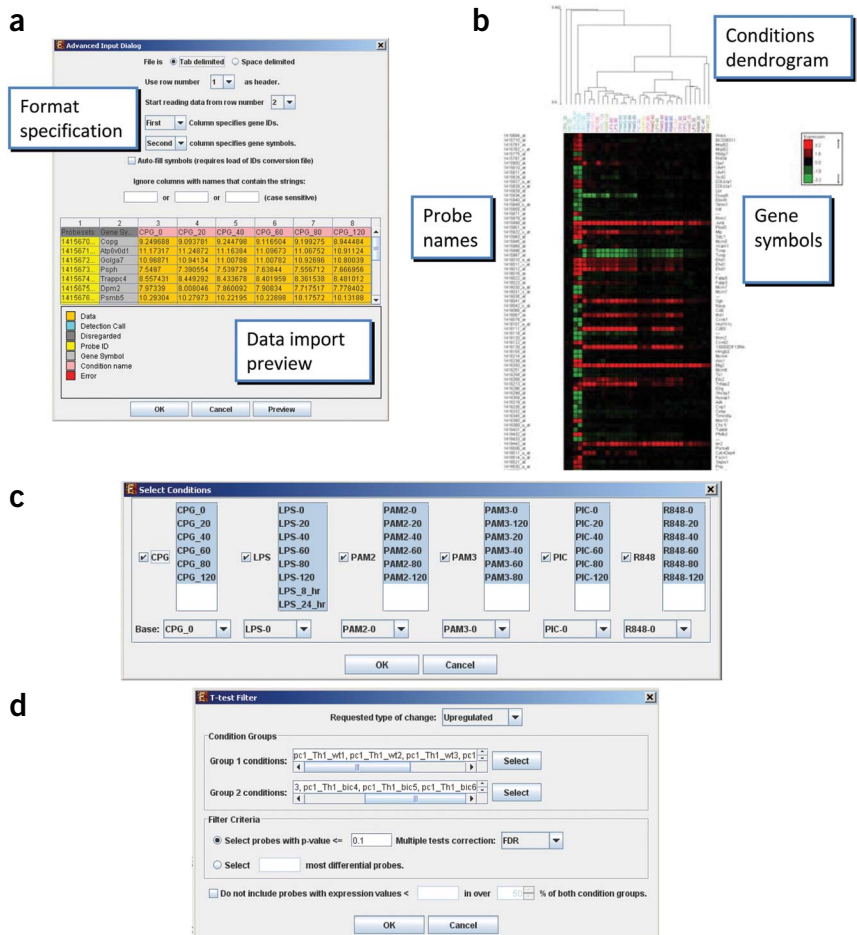


Figure 3 | Expander views. (a) Advanced data import dialog and (b) hierarchical clustering (partial view). Note that the red/green coloring scheme can be changed to blue/yellow using Options→Settings→Display; (c) normalization relative to reference conditions and (d) *t*-test configuration.

PROTOCOL

In Linux, type 'which R' in the terminal to find the R path. If more than one version of R is installed, ensure to point Expander to a version in which the relevant R packages have been installed.

(viii) Once CEL files preprocessing is complete, quantile normalization will be applied, and a corresponding tabular data file is generated and a 'Load Tabular Data' dialog box will appear, as described in Step 2A.

(C) Custom gene groups

- (i) Prepare a gene groups file in a two-column tab-delimited format, which maps the probe identifiers to gene group names (**Box 2**). A sample gene groups file appears in the Expander home directory under `sample_input_files/geneSetsData1.txt`.
- (ii) Select the *File* menu and then *New Session* → *Gene Groups*. A 'Load Gene Groups' dialog box will appear.
- (iii) Select the desired species in the *Organism* pull-down menu and specify the file name in the *File Name* field. Press *OK*. After this step proceed directly to Step 20.

3| Grouping the conditions into subsets improves the heat map visualization and allows easier selection of multiple conditions (e.g., when using the *t*-test filter, Step 9B). In the *Data* menu, select *Define Condition Subsets*. In the dialog box, for each condition subset, use the condition list on the left to select the conditions that should be grouped, type the subset name in the *Group name* field and then press the >> button. EX1 contains several time series, each describing exposure to a different TLR stimulator⁷. For example, all the conditions starting with 'LPS' describe a time series of exposure to LPS. Note that each time point is treated by Expander as a separate condition. Group the conditions in each time series into a separate condition subset. When analyzing EX2, group the first five conditions into a 'control' subset and the rest of the conditions into a 'mir-155' subset.

4| After expression data are loaded, a new tab, called 'Data Sheet 1,' will appear in the main window (**Fig. 2**). Subsequent analyses will open additional tabs, and the active tab can be selected by clicking its name on the lower part of the screen (**Fig. 2**). A data sheet can be closed by right clicking the tab and selecting 'Close.' A data sheet tab consists of three panels, as shown in **Figure 2**: (a) The top left panel presents information about the raw data file. (b) The bottom left panel is initially blank, and presents information about the preprocessing procedures applied to the raw data. (c) The right panel consists of several tabs (selected at the top on the panel), which contain different visualizations of the data set. Initially, they include 'Box Plots,' a visualization describing the distribution of intensities in each sample and 'heat maps' of the raw and preprocessed data matrices. In the heat maps, the conditions are grouped and color-coded based on the condition subsets that were defined in Step 3. If detection calls are available (for Affymetrix microarrays only), the detection statistics for each probe appear in three columns in the raw data heat map, on a 0–1 scale, corresponding to the relative appearance frequency of each detection call (P, M and A).

Organization and normalization of the data set

5| *Optional step:* This step is relevant only for the analysis of oligonucleotide arrays. An *M* versus *A* plot compares two conditions *A* and *B*, by plotting the log intensity ratios $M = \log_2(A_i/B_i)$ versus average log intensities $\log_2(A_i B_i)/2$, where A_i and B_i represent the measurements for probe *i* in conditions *A* and *B*, respectively³⁰. View an *M* versus *A* plot by selecting in the *Preprocessing* menu *Normalization* → *View Scatter Plots*. Select the two conditions that will be compared and select the *M* versus *A* plot option. Ideally, if the data are properly normalized, the points in the plot should be scattered around the horizontal line $M=0$. If this is not the case, it is likely that an intensity-dependent bias is present in the data, which can be corrected using normalization, as described in the next step.

6| *Optional step:* Use this step if the raw data are suspected to contain technical biases (based on visual inspection of intensity box plots and *M* versus *A* plots). If EX1 or E2 is being used, no normalization is needed as the data were already normalized. Expander supports two normalization algorithms:

(A) Quantile normalization

- (i) This method makes the distribution of probe intensities to be the same for all arrays in the data set³⁰. To perform quantile normalization, in the *Preprocessing* menu select *Normalization* → *Quantile*.

(B) Nonlinear baseline normalization

- (i) This normalization is based on pseudo-locally weighted scatterplot smoothing (LOESS) regression of the *M* versus *A* scatter plot²⁹. In the *Preprocessing* menu, select *Normalization* → *Nonlinear baseline*. The other conditions will be normalized relative to a reference condition that can be specified in the *Select base condition* dropdown menu. The subset of genes that are used to evaluate the normalization function can be set to 'all genes' (recommended when most genes in the data set are expected to be constantly expressed) or a 'rank invariant set' of genes (recommended when it is likely that a large number of genes are differentially expressed).

7| *Optional step:* If the data contains technical or biological repeats, it is possible to average them using the *Merge Conditions* option in the *Preprocessing* menu. Merging is not required when using sample files.

8| It is possible to further normalize the data relative to reference condition(s), by dividing each value by the corresponding value in the reference condition. The time courses in EX1 should each be normalized in relation to time point 0 in each time course. In the *Preprocessing* menu, select *Divide by Base*. If the data are on a \log_2 scale, a message box will raise the alert that the values in the base condition(s) will be subtracted from each sample (rather than dividing each sample by the relative condition). Press *OK*. Select the checkbox next to each condition subset (**Fig. 3c**). In the *Base* dropdown menus, select the appropriate time point 0 for each condition subset (**Fig. 3c**) and press *OK*. In the *Preprocessed Data Box-plot* panel in the main window, the values in time point 0 of each time course should be always 0. Details on the preprocessing now appear in the bottom left panel of the main window.

Filtering probes and conditions

9| Expander implements various filtering methods (**Box 3**). In this protocol, we will describe two filtering options: filtering by fold change, which is useful when analyzing time course data, and filtering by *t*-test, which is appropriate when the data compare two groups of samples. If one is following this protocol as tutorial, use option A for analyzing EX1 and option B for analyzing EX2.

▲ **CRITICAL STEP** Filtering of uninformative expression patterns is crucial for successful application of hierarchical clustering and gene grouping algorithms. For optimal performance in terms of execution time and solution quality, the number of probes passing the filtering step should typically not exceed 5,000.

(A) Filtering by fold change

- (i) In the *Preprocessing* menu, select *Filter Probes*→*Fold Change*. One can specify the baseline relative to which the fold change will be computed. When Step 8 is performed, the data points are already normalized by a reference condition. In this case, the fold change criterion should be computed relative to time 0; therefore, select the *Use reference as base value* option. When analyzing absolute intensities data, if the data were not divided before by any reference conditions, it is possible to compute the fold relative to either the minimum expression level of each probe or by its level in a base condition selected by the user. This is done by selecting either the *Use minimal value as base value* or the *Select base condition* options, respectively.
- (ii) Specify the fold parameter and the minimal number of conditions in which this fold-change has to be observed. For EX1, specify a fold change of 4 in at least one condition.
- (iii) Note the *Preprocessed Data Box-plot*, located at the bottom of the right panel. One would see that the values in time 0 of each time course are always 0, and the variance increases with time in each time course.

(B) Filtering by differential expression

- (i) Expander contains two options for filtering by differential expression: *T*-test and SAM. Using *T*-test, it is possible to compare two groups of conditions, whereas using SAM any number of groups can be compared, while correcting for multiple testing. In order to use the *T*-test filter, in the *Preprocessing* menu select *Filter Probes*→*T-test*. A dialog box will appear (**Fig. 3d**).
- (ii) Select the type of *t*-test (one-sided or two-sided) in the *Requested type of change* dropdown menu. For EX2, select *Up-regulated*.
- (iii) Press the *Select* button to specify the conditions in the two groups. In case of a case–control data, group 1 conditions are the ‘controls’ and group 2 conditions are the ‘cases.’ For EX2, select the conditions in the ‘control’ subset as group 1 and the conditions in the ‘mir- 155’ subset as group 2 (**Fig. 3d**).
- (iv) Specify the significance threshold in the *Select probes with P-value ≤* field and specify the multiple testing correction. Alternatively, one can specify that a fixed number of genes with the most significant *P*-values will pass the filter. For EX2, set the *P*-value threshold to 0.1 and select *FDR* as the multiple testing correction. Press *OK*.
- (v) Note the *Preprocessed Data Box-plot*, located at the bottom of the right panel. One would see a clear difference between the distributions of the intensity levels in the two condition subsets.

10| *Optional step:* In the *Preprocessing* menu, select *Filter Conditions*. Select the conditions one wants to use in subsequent analysis (use the Ctrl and Shift keys to select multiple conditions). When analyzing EX1, select all the conditions except for time point 0 in each condition subset (as these time 0 points are not informative after all conditions are divided by their matching time point 0 (Step 8)).

11| Perform hierarchical clustering of the conditions. In the *Grouping* menu, select *Hierarchical Clustering*. Select the clustering method⁴⁹ in the *Linkage* dropdown menu (select *Average linkage* for EX1) and check only the *Conditions* checkbox. Press *OK*.

BOX 8 | HOW TO CHOOSE THE GROUPING ALGORITHM

Gene grouping using clustering is a hard task and no single algorithm is superior over others. In the number of clusters is known, *k*-means or SOM have a potential advantage. Otherwise, several values of *k* can be tried, or CLICK, which does not require tuning *k* can be used. CLICK was shown to outperform the other algorithms on some expression data sets².

When the number of studied conditions is very large (e.g., over 30), biclustering can have an advantage over clustering, as it does not assume that genes belonging to the same group exhibit similar expression across all the conditions⁶.

12| A new tab called 'Hierarchical 1.1' will now appear in the main window. The panel on the left contains information about the clustering and the panel on the right shows the clustered gene expression matrix (**Fig. 3b**).

13| Save the clustered heat map. In the *File* menu, select *Save As Image*. Check only the *dendrogram with matrix* option. Enter the name of the directory where the image file should be placed. Press *OK*.

Gene grouping

14| General guidelines on the choice of clustering algorithm are available in **Box 8**. Cluster the gene expression patterns using the CLICK algorithm. In the *Grouping* menu, select *Clustering*→*CLICK*. Select *Default homogeneity* and press *OK*. For details on the homogeneity parameter, refer to the CLICK manuscript². Clustering using CLICK takes a few minutes for data sets with several thousands of genes. Note that most advanced analysis algorithms (gene grouping, functional enrichment and promoter analysis) have a random component, so results can vary slightly when the same algorithm is repeated.

? TROUBLESHOOTING

15| Once clustering is completed, a new tab named 'CLICK 1.1' will appear in the main window (**Fig. 4**). The top left panel contains information about the clustering algorithm, the quality of the results (overall homogeneity and separation), the number of clusters and the number of singletons (the number of genes that were not assigned to any cluster). In addition, a table is shown, specifying the names of the clusters, their sizes and homogeneity (the average Pearson's correlation among the expression patterns of probes in the cluster). The bottom left panel contains the average expression pattern of each cluster with error bars (± 1 s.d.).

16| In the top left panel, click on one of the table rows. The corresponding cluster display will appear on the right. It contains a list (shown as a table) of the probes in the cluster, and the genes they correspond to. Click on one of the rows. The expression pattern of the probe described in the row that is selected will appear on the bottom. Click on one of the Entrez Gene identifiers in the 'Gene ID' column. This will open an internet browser window with the Entrez Gene webpage describing the gene.

17| In the cluster display (right panel), switch the tab to *Expression matrix*. A heat map of the cluster will appear (**Fig. 4c**). This image can be saved by selecting *Save as Image* in the *File* menu (see Step 13).

18| In the right panel, switch to the *Positions* tab. Chromosomes appear as vertical lines, and the genes in the cluster that is selected appear as horizontal bars on them, at their genomic locations. Zoom in by clicking the magnifying glass button and then clicking on the area of the panel that one would like to magnify. For EX1, in the dropdown menu on the toolbar of the right panel, select 'LPS-120'. The length and the direction of the bars now correspond to the expression levels of the genes in this condition.

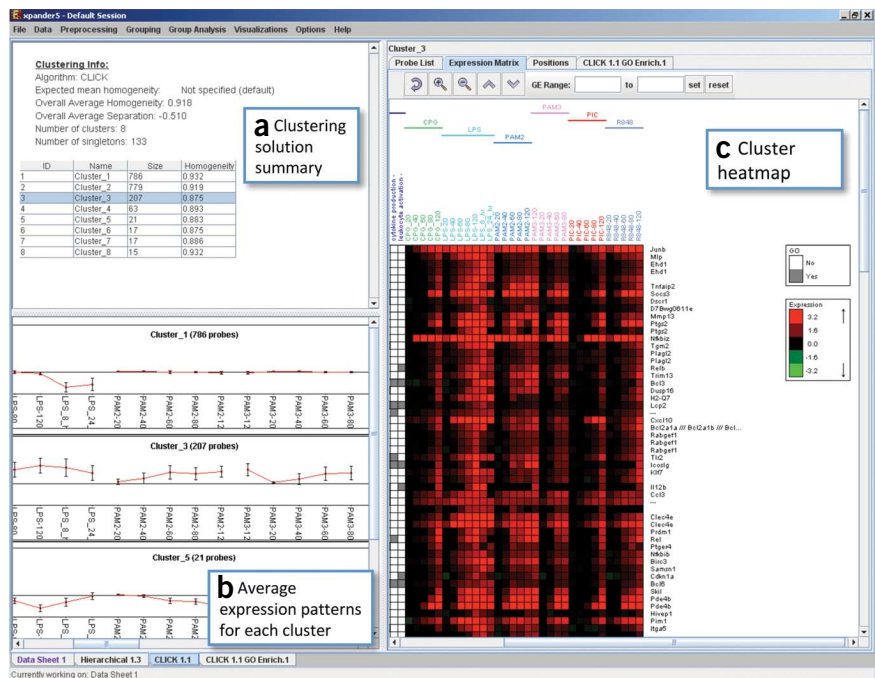


Figure 4 | Clustering solution visualization. (a) Statistics of the clustering solution and a table with cluster sizes and homogeneities. (b) Average expression patterns in each cluster. Error bars indicate 1 s.d. (c) A heat map of the cluster currently selected in panel a.

19| In the *Visualizations* menu, select *Clustered Expression Matrix*. A new tab will appear in the main window. In the top right panel (*Coloring scheme*), select 'CLICK 1.1'. The rows in the heat map in the left panel are now sorted by the assignment of the probes to clusters. The labels of the probes and the gene symbols are now color-coded based on their cluster assignment and the color legend appears in the bottom right panel. This view can also be saved using *File*→*Save As Image* (see Step 13).

Functional enrichment analysis

20| At this point, gene groups have been identified using one of the methods offered in *Expander* or loaded using Step 2. These groups will be tested for enrichment with GO annotations⁵⁰. If not already done so, the organism-specific pre-compiled annotation data should be downloaded, as described in **Box 7**. *Expander* uses the TANGO algorithm (**Box 4**) to identify GO enrichments that are statistically significant after correction for multiple testing. To run TANGO, go to the *Group Analysis* menu and select *Functional Analysis*→*TANGO*.

? TROUBLESHOOTING

21| The dialog box that will appear specifies the TANGO parameters: (a) In the *Perform analysis on* dropdown menu, select the gene groups that are to be analyzed (select CLICK 1.1 if Step 14 has been followed). (b) The option *Include back nodes* is visible only if the analysis is performed on MATISSE results (see Step 35 for definition of back nodes). (c) In the *Focus on* option, select the GO subsets that will be used in the analysis. For EX1, select only the *Process* option. (d) The *Ignore classes over size of* field allows to filter out GO terms that are too general. For EX1, set this parameter to 2,000. (e) The *Number of iterations by the algorithm* option specifies the number of randomizations (see **Box 4** for details). (f) In the *Background set* panel, specify the background against which enrichments will be computed. When analyzing gene expression data, it is recommended to use the set of genes that are represented on the microarray as background. To do so, select *Original Data*. (g) The *Corrected P-value threshold* field specifies the threshold below which enrichment results will be reported. When all the parameters are set, press *OK*. Execution of TANGO can take up to 10 min for large data sets.

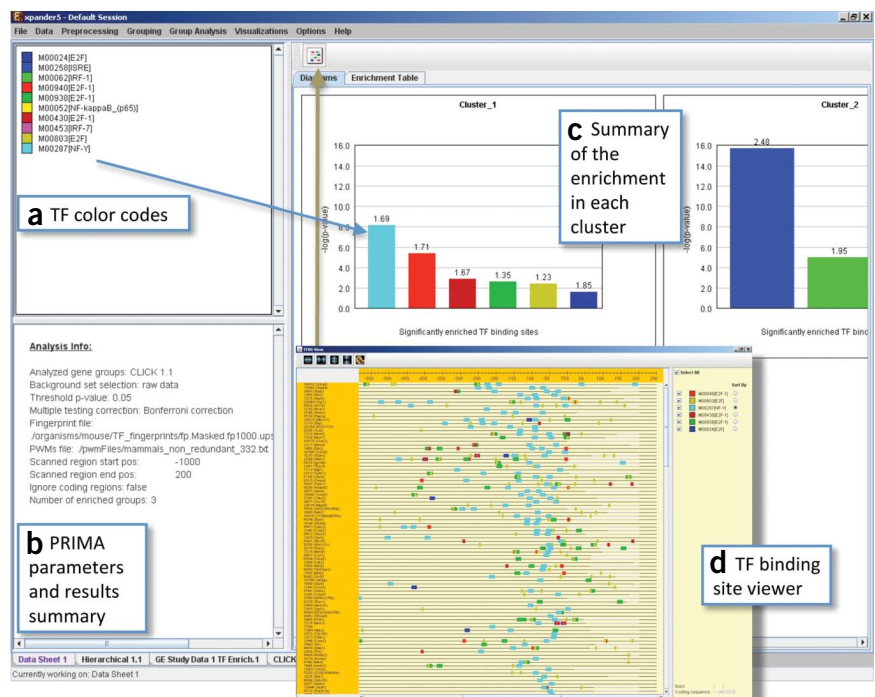
? TROUBLESHOOTING

22| When TANGO execution is completed, a new tab will appear in the main window. The structure of all the tabs showing enrichment results (produced by TANGO, PRIMA and FAME) is the same (**Fig. 5**). The tab consists of three panels: (a) The top left panel, which lists the color codes of the annotations that are significantly enriched in at least one gene group. (b) The bottom left panel, which describes the input parameters and the number of gene groups in which significant enrichment was found. (c) The right panel, which contains one bar diagram for each gene group in which enrichment has been detected. Each chart contains a bar for each significantly enriched GO annotation (after correction for multiple testing). The height of the bar is proportional to the significance of this enrichment (i.e., height = $-\log_{10}(\text{raw } P\text{-value})$).

The frequency in the group (the fraction of the genes in the group that were annotated with the GO annotation) appears on top of the column.

? TROUBLESHOOTING

Figure 5 | Enrichment analysis results. The presented results are for PRIMA analysis. TANGO or FAME results are presented in a similar way. (a) Legend of the color codes for each annotation (TF in this case). (b) Summary of the parameters and the results of the enrichment algorithm. (c) Bar charts showing the enrichment in each cluster. Each column indicates the enrichment of a different annotation (TF in this case). Bars indicate enrichment *P*-value (base 10 log-scale). Numbers above the bars are relative abundance in the target versus background sets. (d) TF binding site viewer. This viewer that can be opened by clicking on the button pointed to by the arrow. Each row presents a single promoter.



PROTOCOL

23| Click on one of the diagram bars. A dialog box with information about the specific enrichment will appear. The following information is available in this dialog box: (a) *Raw P-value*, the hypergeometric enrichment *P*-value, not corrected for multiple testing. (b) *Corrected P-value*, the significance level after the correction performed by TANGO (note that the smallest possible corrected *P*-value that can be reported is $1/N_{\text{rand}}$, where N_{rand} is the number of randomizations carried out by the algorithm). (c) *Number of genes*, the number of genes that appear in the gene group and that are also annotated with the GO annotation. (d) *Frequency in set*, the percentage of the genes in the group that are annotated with the GO annotation.

24| In the right panel, switch to the *Enrichment Table* tab. This table lists all the significant enrichments that were detected. Click on the header of the *Raw P-value* column to sort the rows in ascending order of *P*-values. Clicking on other headers will sort the table by the respective columns.

25| Save the results to a text file by selecting *Group Analysis*→*Functional Analysis*→*Export Results*. Specify the name of the output file and click *OK*.

26| Switch to the clustering results tab (e.g., CLICK 1.1) and select one of the clusters in which enrichment was found. In the right panel, switch to the new *GO Enrich* tab (e.g., 'CLICK 1.1 GO Enrich 1'), which shows a bar chart displaying the enrichment found in the selected cluster. Switch to the *Expression matrix* tab. Columns in the left part of the heat map (*Enriched functions*) correspond to GO terms that were significantly enriched in the gene group. Gray cells in these columns indicate the genes annotated with the GO term.

Promoter analysis

27| Examine the enrichment of TFBSs in the promoters of genes in each group using the PRIMA algorithm, which performs a statistical analysis on the distribution of TFBS motifs in the promoters of genes within each gene group, using a generalized hypergeometric test⁷. In the *Group Analysis* menu, select *Promoter Analysis*→*PRIMA*. Some of the PRIMA parameters are identical to the TANGO parameters described in Step 21. The additional parameters are (a) *Fingerprint, PWM* (position weight matrix) and *Promoter sequence files*, which are pre-compiled files used by PRIMA. The default values of these fields are filled automatically, based on the organism that is selected in the beginning of the session. (b) *Hits range* determines which region of the promoter will be analyzed. The relevant ranges vary among the supported organisms (see the Expander manual for details). (c) *Ignore coding regions* if this option is selected, TFBS falling within coding regions will be excluded from the analysis. (d) In *Multiple testing correction* in this dropdown menu, one can specify whether the significance levels should be corrected for multiple hypothesis testing. (e) *Save results as* field allows to specify the name of a file which will contain a textual description of the results. When analyzing EX1, set the background set to *Original data*, *P*-value threshold to 0.05 and multiple testing correction to *Bonferroni*. Press *OK*. Note that PRIMA execution can take up to 20 min, depending on the number and size of the gene groups.

? TROUBLESHOOTING

28| After the analysis is completed, the results appear in a new tab in the main window (**Fig. 5**). The structure of this tab and the operations that can be performed are similar to the functional analysis results tab described in Step 22. Each TF PWM is represented by its TRANSFAC database identifier⁵¹. Unlike the functional analysis results, this display also contains the frequency ratio of TF prevalence (the frequency in the group divided by the frequency in the background set). It is displayed on top of the corresponding bar.

29| In the right panel, click on one of the chart bars. A dialog box with details about the specific enrichment and the list of genes in the group containing the motif (similar to the dialog in Step 23) will appear.

30| Press the button at the top of the right results panel (**Fig. 5d**). The *View binding sites* dialog box will appear. In the dialog box, select the set of PRIMA results (e.g., 'CLICK 1.1 TF enrich.1') as the *Analysis*. In the *Set* dropdown menu, select one of the gene groups and press *OK*. The 'TFBS View' frame will be displayed. This view shows the locations of the TFBS hits on the promoters of the genes in the group. Each line represents a promoter and each colored rectangle represents a putative TFBS. A color index appears on the right, mapping each color to the corresponding TF. Click on the checkbox next to each of the entries in the color index to hide/show the sites of the corresponding TF. Click on the radio button next to one of the entries in the color index to sort the genes in the display according to the number of hits of the corresponding TF. Use the buttons in the toolbar to adjust the display. When the zoom factor allows it, the actual promoter sequence is displayed (**Fig. 5**).

PROTOCOL

described in Step 21. Other FAME parameters are described in **Box 5**. For analysis of EX2, use the default parameters. Execution of FAME can take up to 3 min for large data sets.

? TROUBLESHOOTING

39| After FAME execution is over, a new results tab will appear in the main window. The results are presented in the same way as the PRIMA results described in Step 28 (**Fig. 5**). Note that each color represents a different TargetScan 5.0 family (**Box 5**).

● TIMING

A large fraction of the time required to execute this protocol is taken by the time required to download the Expander pre-compiled annotation data (which depends on the internet connection speed). Once the annotation data have been downloaded, it will remain installed and will be available during subsequent executions of Expander. The download of the mouse annotation data takes about 20 min for a standard home internet connection.

The times of the remaining steps of the protocol depend on the size of the data set analyzed. The clustering algorithms are affected by the number of probes that are clustered, and the group analysis methods are dependent on the number of the groups and on their sizes. On a PC with 3 GHz CPU and 2 GB of memory, with the sample files EX1 and EX2, execution of hierarchical clustering takes 2 min, CLICK 5–10 min, TANGO 1 min, PRIMA ≈20 min and FAME 3 min. The rest of the steps take only a few seconds. Once the annotation data are downloaded, an experienced user can execute the full protocol described within 1 h.

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 3**.

TABLE 3 | Troubleshooting table.

Step	Problem	Possible reason	Solution
1	Executing Expander.bat or Expander_2GB.bat does not open the Expander window	Java is not installed properly	Check the Java installation. If necessary, reinstall Java 5.0 (or higher)
2A(x)	When loading tabular data, the preview table has red background	The data file cannot be read	Ensure that the part of the table that contains the data contains only numbers or one of the supported missing value indicators – “-”, “<NA>” or “NA”
14 and 34	Execution of gene grouping algorithms takes a very long time	Preprocessed data contain too many probes/genes	Reduce the size of the data set by applying a more stringent filtering (Step 9). For optimal performance, the number of probes that pass the filtering step should not exceed 5,000
All	An ‘Out of memory’ error box appears	Expander has exhausted all the available computer memory	Close unnecessary visualization tabs by right clicking the tab name If the PC allows it, run Expander with more memory (using Expander_2GB.bat) If possible, apply filtering to the data and remove unnecessary probes/conditions Save the session and try re-opening Expander or rebooting the computer.
14, 21 and 27	CLICK, TANGO, SAMBA and/or PRIMA fail when running on Linux	User permissions are not set correctly	Make sure that you have write permission in the Expander directory, and execution permissions for the files: click.exe, samba.exe, annot_sets.exe and analyze Fingerprints.exe, which are under the Expander directory
21, 27 and 38	TANGO, PRIMA or FAME produce no results	A problem with the probe→gene mapping file The organism field in the input dialog is not set correctly (the selected organism appears in the top left panel in the ‘Data sheet’ tab) The threshold <i>P</i> -value is too rigid or no significant enrichments exist	Check that the conversion file maps the probe identifiers exactly, as they appear in the data file (e.g., lower case) to the gene identifiers supported by Expander (Table 1) Open a new session with the correct organism specified Try increasing the <i>P</i> -value threshold

ANTICIPATED RESULTS

If EX1 file (BMM.txt) is loaded properly, it should contain 45,101 probes and 38 conditions. Out of these, 2,038 probes show a fourfold change relative to the reference conditions and pass the filtering in Step 9A. Most of the algorithms used in the protocol contain some random components and results may vary slightly among runs.

Hierarchical clustering of the conditions shows four main groups of similar conditions: (a) response to LPS after 8 and 24 h; (b) Response to all the agents after 120 min and LPS, PAM2, PAM3 and PIC after 80 min; (c) Response to CPG and R848 after 40, 60 and 80 min, LPS, PAM2 and PAM3 after 40 min and PIC after 60 and 80 min; (d) Response to LPS, R848 and PAM2 after 20 min and to PIC after 40 min.

CLICK clustering identifies 8 clusters—three large clusters with more than 100 genes, and five smaller ones (cluster sizes are shown in **Fig. 4a**). Note that the order, exact sizes and number of clusters can vary slightly between runs of CLICK, due to stochastic elements in it. Cluster 1 contains ~700 genes up-regulated following treatment with LPS in time points 8 and 24 h. Cluster 2 also contains ~700 genes down-regulated following treatment with LPS in the same time points. Cluster 3 contains ~200 genes up-regulated following all six treatments.

TANGO identifies enrichments in the three large clusters. Cluster 1 is enriched with immune response-related terms ('immune response', 'defense response', 'response to virus' etc.). Cluster 2 is enriched with cell cycle-related terms ('cell cycle', 'mitosis' etc.), and Cluster 3 is also enriched with immune-response related terms, as well as with cytokine production and apoptosis. TANGO also contains a random component (see **Box 4**), and results may vary slightly among runs.

PRIMA identifies significant TFBS enrichments in the three large clusters. The promoters of Cluster 1 are enriched with sites corresponding to ISRE, IRF-1 and IRF-7, known regulators of the interferon response. Cluster 2 is enriched with targets of cell cycle TFs E2F and NF-Y, and to a lesser extent with targets of Nrf1, ZF5 and ETF. Cluster 3 is enriched with targets of NF-κB, another key regulator of the immune response. For a detailed discussion of these results, see our previous publication about the TLR signaling transcriptional response⁷.

MATISSE detects eight modules containing 4–19 genes each. The largest module (**Fig. 6**) contains eight members of the DNA replication initiation machinery (Mcm2-7, Gmnn and Ris2). Two modules contain 10 genes each: one with four genes involved in hemopoiesis (Syk, Jak2, Csf1 and Fas) and another with two subunits of DNA polymerase delta (Pold1 and Pold2). Another module (8 genes) contains four cyclin genes (Ccn2, Ccnb1, Ccnd1 and Cdc2a). MATISSE thus identifies at least three cell-cycle related complexes, all of which are down-regulated as part of the late response to LPS exposure. The hemopoiesis module is also interesting, as this function is not enriched in any of the gene expression clusters. The two main hubs in this module, Csf1 and Fas, both have known hemopoiesis-related roles in response to LPS^{52,53}. Note that one of the reasons the identified modules are rather small is that the mouse PPI network is very limited in scope. Analysis of human or *S. cerevisiae* data typically uncovers larger modules^{9,10}.

In EX2, 33 probes are upregulated with FDR <0.05 in cells deficient for mir-155. FAME analysis identifies a significant enrichment for mir-155 targets (raw *P*-value $1.38 \cdot 10^{-7}$, corrected *P*-value 0.004), and a less significant enrichment for mir-142-3p.

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

ACKNOWLEDGMENTS We thank Israel Steinfeld for his role in the early development of Expander, Metsada Pasmanik-Chor for useful discussions and Akshay Krishnamurthy for commenting on an early version of the protocol. Igor Ulitsky was partially supported by the Edmond J Safrá Bioinformatics Program at Tel Aviv University and by the Legacy Heritage Fund. Yosef Shiloh is a Research Professor of the Israel Cancer Research Fund. This study was supported in part by the Israel Science Foundation (Grant No. 802/08) and by the European Community's Seventh Framework Programme (Grants HEALTH-F4-2007-200767 for the APO-SYS project and HEALTH-F4-2009-223575 for the TRIREME project).

COMPETING INTERESTS STATEMENT The authors declare competing financial interests (see the HTML version of this article for details). Expander is available for commercial licensing through Tel Aviv University's technology transfer company.

AUTHOR CONTRIBUTIONS R.S. conceived and led the project; A.M.-K., S.S. and D.S. developed Expander using software code contributed by R.S., A.T., C.L., R.E. and I.U.; I.U. and R.S. wrote the paper. All the authors contributed to the design of Expander, and all have read and approved the paper.

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

1. Sharan, R., Maron-Katz, A. & Shamir, R. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics* **19**, 1787–1799 (2003).
2. Sharan, R. & Shamir, R. CLICK: a clustering algorithm with applications to gene expression analysis. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 307–316 (2000).
3. Elkon, R., Linhart, C., Sharan, R., Shamir, R. & Shiloh, Y. Genome-wide *in silico* identification of transcriptional regulators controlling the cell cycle in human cells. *Genome Res.* **13**, 773–780 (2003).
4. Shamir, R. *et al.* EXPANDER—an integrative program suite for microarray data analysis. *BMC Bioinformatics* **6**, 232 (2005).
5. Tanay, A., Sharan, R., Kupiec, M. & Shamir, R. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl. Acad. Sci. USA* **101**, 2981–2986 (2004).
6. Tanay, A., Sharan, R. & Shamir, R. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **18** (Suppl 1): S136–S144 (2002).
7. Elkon, R., Linhart, C., Halperin, Y., Shiloh, Y. & Shamir, R. Functional genomic delineation of TLR-induced transcriptional networks. *BMC Genomics* **8**, 394 (2007).
8. Elkon, R. *et al.* SPIKE—a database, visualization and analysis tool of cellular signaling pathways. *BMC Bioinformatics* **9**, 110 (2008).
9. Muller, F.J. *et al.* Regulatory networks define phenotypic classes of human stem cell lines. *Nature* **455**, 401–405 (2008).
10. Ulitsky, I. & Shamir, R. Identification of functional modules using network topology and high-throughput data. *BMC Syst. Biol.* **1**, 8 (2007).

11. Kuehn, H., Liberzon, A., Reich, M. & Mesirov, J.P. Using GenePattern for gene expression analysis. *Curr. Protoc. Bioinformatics* **Chapter 7** Unit 7 12 (2008).
12. Li, C. & Wong, W.H. Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc. Natl. Acad. Sci. USA* **98**, 31–36 (2001).
13. Saeed, A.I. *et al.* TM4: a free, open-source system for microarray data management and analysis. *Biotechniques* **34**, 374–378 (2003).
14. Stavrum, A.K., Petersen, K., Jonassen, I. & Dysvik, B. Analysis of gene-expression data using J-Express. *Curr. Protoc. Bioinformatics* Chapter 7 Unit 7 3 (2008).
15. Rustici, G. *et al.* Data storage and analysis in ArrayExpress and Expression Profiler. *Curr. Protoc. Bioinformatics*. **Chapter 7** Unit 7 13 (2008).
16. Tarraga, J. *et al.* GEPAS, a web-based tool for microarray data analysis and interpretation. *Nucleic Acids Res.* **36**, W308–W314 (2008).
17. Sturn, A., Quackenbush, J. & Trajanoski, Z. Genesis: cluster analysis of microarray data. *Bioinformatics* **18**, 207–208 (2002).
18. Cline, M.S. *et al.* Integration of biological networks and gene expression data using Cytoscape. *Nat. Protoc.* **2**, 2366–2382 (2007).
19. Stern, S., Dror, T., Stolovicki, E., Brenner, N. & Braun, E. Genome-wide transcriptional plasticity underlies cellular adaptation to novel challenge. *Mol. Syst. Biol.* **3**, 106 (2007).
20. Rosenzweig, D. *et al.* Retooling *Leishmania* metabolism: from sand fly gut to human macrophage. *FASEB J.* **22**, 590–602 (2008).
21. Oron, E. *et al.* Genomic analysis of COP9 signalosome function in *Drosophila melanogaster* reveals a role in temporal regulation of gene expression. *Mol. Syst. Biol.* **3**, 108 (2007).
22. Blum, R. *et al.* Gene expression signature of human cancer cell lines treated with the ras inhibitor salirasib (S-farnesylthiosalicylic acid). *Cancer Res.* **67**, 3320–3328 (2007).
23. Elkon, R. *et al.* Dissection of a DNA-damage-induced transcriptional network using a combination of microarrays, RNA interference and computational promoter analysis. *Genome Biol.* **6**, R43 (2005).
24. Blum, R. *et al.* E2F1 identified by promoter and biochemical analysis as a central target of glioblastoma cell-cycle arrest in response to Ras inhibition. *Int. J. Cancer* **119**, 527–538 (2006).
25. Laurent, L.C. *et al.* Comprehensive microRNA profiling reveals a unique human embryonic stem cell signature dominated by a single seed sequence. *Stem Cells* **26**, 1506–1516 (2008).
26. Rodriguez, A. *et al.* Requirement of bic/microRNA-155 for normal immune function. *Science* **316**, 608–611 (2007).
27. Irizarry, R.A. *et al.* Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.* **31**, e15 (2003).
28. Smedley, D. *et al.* BioMart—biological queries made easy. *BMC Genomics* **10**, 22 (2009).
29. Schadt, E.E., Li, C., Ellis, B. & Wong, W.H. Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J. Cell. Biochem. Suppl.* (Suppl 37): 120–125 (2001).
30. Bolstad, B.M., Irizarry, R.A., Astrand, M. & Speed, T.P. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19**, 185–193 (2003).
31. 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).
32. Raychaudhuri, S., Stuart, J.M. & Altman, R.B. Principal components analysis to summarize microarray experiments: application to sporulation time series. *Pac. Symp. Biocomput.* 455–466 (2000).
33. 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).
34. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J. & Church, G.M. Systematic determination of genetic network architecture. *Nat. Genet.* **22**, 281–285 (1999).
35. Tanay, A., Steinfeld, I., Kupiec, M. & Shamir, R. Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium. *Mol. Syst. Biol.* **1**, 2005.0002 (2005).
36. Hanisch, D., Zien, A., Zimmer, R. & Lengauer, T. Co-clustering of biological networks and gene expression data. *Bioinformatics* **18** (Suppl 1): S145–S154 (2002).
37. Ideker, T., Ozier, O., Schwikowski, B. & Siegel, A.F. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* **18** (Suppl 1): S233–S240 (2002).
38. Liu, M. *et al.* Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genet.* **3**, e96 (2007).
39. Luscombe, N.M. *et al.* Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature* **431**, 308–312 (2004).
40. Suzuki, H. *et al.* The transcriptional network that controls growth arrest and differentiation in a human myeloid leukemia cell line. *Nat. Genet.* **41**, 553–562 (2009).
41. Farh, K.K. *et al.* The widespread impact of mammalian microRNAs on mRNA repression and evolution. *Science* **310**, 1817–1821 (2005).
42. Lim, L.P. *et al.* Microarray analysis shows that some microRNAs down-regulate large numbers of target mRNAs. *Nature* **433**, 769–773 (2005).
43. Halperin, Y., Linhart, C., Ulitsky, I. & Shamir, R. Allegro: analyzing expression and sequence in concert to discover regulatory programs. *Nucleic Acids Res.* **37**, 1566–1579 (2009).
44. Friedman, R.C., Farh, K.K., Burge, C.B. & Bartel, D.P. Most mammalian mRNAs are conserved targets of microRNAs. *Genome Res.* **19**, 92–105 (2009).
45. Grimson, A. *et al.* MicroRNA targeting specificity in mammals: determinants beyond seed pairing. *Mol. Cell* **27**, 91–105 (2007).
46. Ripley, B. The R project in statistical computing. *MSOR Connections. The newsletter of the LTSN Maths, Stats & OR Network* **1**, 23–25 (2001).
47. Hermjakob, H. *et al.* IntAct: an open source molecular interaction database. *Nucleic Acids Res.* **32**, D452–D455 (2004).
48. Troyanskaya, O. *et al.* Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**, 520–525 (2001).
49. Quackenbush, J. Computational analysis of microarray data. *Nat. Rev. Genet.* **2**, 418–427 (2001).
50. Ashburner, M. *et al.* Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
51. Wingender, E. *et al.* TRANSFAC: an integrated system for gene expression regulation. *Nucleic Acids Res.* **28**, 316–319 (2000).
52. Warren, M.K. & Ralph, P. Macrophage growth factor CSF-1 stimulates human monocyte production of interferon, tumor necrosis factor, and colony stimulating activity. *J. Immunol.* **137**, 2281–2285 (1986).
53. Um, H.D., Orenstein, J.M. & Wahl, S.M. Fas mediates apoptosis in human monocytes by a reactive oxygen intermediate dependent pathway. *J. Immunol.* **156**, 3469–3477 (1996).
54. Tusher, V.G., Tibshirani, R. & Chu, G. Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA* **98**, 5116–5121 (2001).
55. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* 289–300 (1995).
56. Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* **102**, 15545–15550 (2005).
57. Gaidatzis, D., van Nimwegen, E., Hausser, J. & Zavolan, M. Inference of miRNA targets using evolutionary conservation and pathway analysis. *BMC Bioinformatics* **8**, 69 (2007).
58. Stark, A., Brennecke, J., Bushati, N., Russell, R.B. & Cohen, S.M. Animal microRNAs confer robustness to gene expression and have a significant impact on 3'UTR evolution. *Cell* **123**, 1133–1146 (2005).
59. Rajewsky, N. microRNA target predictions in animals. *Nat. Genet.* **38** (Suppl): S8–S13 (2006).
60. Baek, D. *et al.* The impact of microRNAs on protein output. *Nature* **455**, 64–71 (2008).
61. Nielsen, C.B. *et al.* Determinants of targeting by endogenous and exogenous microRNAs and siRNAs. *RNA* **13**, 1894–1910 (2007).
62. Sandberg, R., Neilson, J.R., Sarma, A., Sharp, P.A. & Burge, C.B. Proliferating cells express mRNAs with shortened 3' untranslated regions and fewer microRNA target sites. *Science* **320**, 1643–1647 (2008).
63. Kanehisa, M. & Goto, S. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).
64. Matthews, L. *et al.* Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res.* **37**, D619–D622 (2009).

