Reverse engineering cellular networks

Adam A Margolin[1,2,4], Kai Wang[1,2,4], Wei Keat Lim[2], Manjunath Kustagi[2], Ilya Nemenman[3] & Andrea Califano[1,2]

[1]Department of Biomedical Informatics, Columbia University, New York, New York 10032, USA. [2]Joint Centers for Systems Biology, Columbia University, New York, New York 10032, USA. [3]Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA. [4]These authors contributed equally to this work. Correspondence should be addressed to A.C. (califano@c2b2.columbia.edu).

Published online 27 June 2006; doi:10.1038/nprot.2006.106

We describe a computational protocol for the ARACNE algorithm, an information-theoretic method for identifying transcriptional interactions between gene products using microarray expression profile data. Similar to other algorithms, ARACNE predicts potential functional associations among genes, or novel functions for uncharacterized genes, by identifying statistical dependencies between gene products. However, based on biochemical validation, literature searches and DNA binding site enrichment analysis, ARACNE has also proven effective in identifying *bona fide* transcriptional targets, even in complex mammalian networks. Thus we envision that predictions made by ARACNE, especially when supplemented with prior knowledge or additional data sources, can provide appropriate hypotheses for the further investigation of cellular networks. While the examples in this protocol use only gene expression profile data, the algorithm's theoretical basis readily extends to a variety of other high-throughput measurements, such as pathway-specific or genome-wide proteomics, microRNA and metabolomics data. As these data become readily available, we expect that ARACNE might prove increasingly useful in elucidating the underlying interaction models. For a microarray data set containing ~10,000 probes, reconstructing the network around a single probe completes in several minutes using a desktop computer with a Pentium 4 processor. Reconstructing a genome-wide network generally requires a computational cluster, especially if the recommended bootstrapping procedure is used.

INTRODUCTION

High-throughput technologies have allowed the simultaneous measurement of the concentrations of thousands of molecular species in a biological system, such as mRNA[1], microRNA[2], proteins[3] and metabolites[4]. As the dynamics of each molecular species are influenced by the concentration of several other species, a number of statistical approaches have been developed to infer functional relationships within large sets of biochemical variables[5–9] based on correlations among the available data modalities. In particular, gene expression profiles, which represent the average concentrations of mRNA in a cellular population, have emerged among the most readily available genome-wide measurements for a variety of organisms.

Computational methods used to infer biochemical interactions from gene expression profile data hold the promise of elucidating functional mechanisms underlying cellular processes[10], as well as identifying molecular targets of pharmacological compounds[11]. Several different approaches have been applied successfully to dissect bacterial[11] and yeast[12] networks; however, a variety of limitations have impeded their generalization to the inference of genome-wide networks in higher eukaryotes. These approaches fall generally into four major categories. Optimization methods[6,13,14], such as Bayesian networks, maximize a scoring function over alternative network models. Challenges with this approach include exponential complexity in the local network connectivity necessitating heuristic search procedures, reliance on unrealistic network models (i.e., directed acyclic graphs) and the need to discretize expression data for most commonly used methods. Regression techniques[10,11] fit the data to *a priori* models and are limited to relatively simple models, because the number of parameters becomes much larger than the number of experimental constraints as the network complexity increases. Integrative bioinformatics approaches[15] combine data from a number of independent experimental clues, which are only now starting to become available for higher eukaryotes. Finally, statistical methods[7] rely on a variety of pairwise gene expression correlation measures, and are generally subject to exceedingly high false-positive rates for molecular species that interact indirectly (i.e., via one or more intermediaries). Furthermore, they suffer from difficulty in defining a rational basis for choosing statistical significance thresholds, as pairwise statistical independence cannot be used as the only criterion in the context of a highly interconnected network.

ARACNE overcomes many limitations of existing algorithms: it has a low polynomial computational complexity, it uses the full dynamic range of the data instead of relying on (arbitrary) discretizations and it does not make assumptions about the underlying network topology. These properties have enabled ARACNE to be successfully applied to a system-wide reconstruction of complex transcriptional networks in human B cells[16]. In contrast to many methods that have not been biochemically validated[17], ARACNE's predictions have been validated for the *MYC* protooncogene by chromatin immunoprecipitation assays (ChIPs), which have shown that *MYC* binds *in vivo* to the regulatory region of 11 out of 12 genes selected among those inferred by the algorithm[16]. When further combined with literature analysis, over 50% of the *MYC* targets inferred by the algorithm were validated. More recently, similar results were achieved for other transcription factors (TFs), including BCL6 (R. Dalla-Favera, unpublished results) and NOTCH1 (A. Ferrando, unpublished results). ARACNE's performance has also been studied on the reconstruction of synthetic biochemical networks, and it has been shown to significantly outperform other algorithms in this setting[9]. Finally, the theoretical limitations of the algorithm have been characterized to asymptotically reconstruct networks exactly under certain assumptions[9]. In particular, ARACNE has been shown to have a low false-positive rate, which makes it

appealing in terms of further biochemical validation of its predictions.

Given the extreme complexity of cellular networks, we do not expect these results to generalize to all cases. For example, due to the focus on reducing false positives, ARACNE might miss a significant number of targets of a TF that is involved in a large number of feedback or feedforward loops. Additionally, ARACNE is sensitive to the ranking of the mutual information (MI) estimates. Thus inhomogeneous noise sources that change the rankings might lead to reconstruction errors. Furthermore, ARACNE is not designed to directly reconstruct complex combinatorial regulation patterns involving multiple independent TFs, although it might identify such interactions one TF at a time. For instance, in the B-cell network[16], interactions of a gene with multiple TFs are frequent, suggesting a cooperative regulation mechanism. An additional limitation, germane to all microarray expression profile analysis methods, is that ARACNE relies on the assumption that the mRNA of a TF is correlated with that of its targets. This assumption might be violated for many TFs that are post-transcriptionally regulated, or if the cells under investigation have not reached equilibrium. Furthermore, as microarray expression profiles only monitor a subset of the interacting species in a biochemical network, many transcriptional interactions might be undetectable.

Due to these limitations, as with any biological assay, predictions made by ARACNE should be used in conjunction with prior knowledge and with additional data (such as promoter region sequence information, ChIP-on-Chip[18] and existing interactomes) to provide a useful tool to biologists attempting to dissect specific transcriptional pathways. This protocol is thus an attempt to provide a straightforward guide, so that ARACNE can be readily used for this purpose by people with relatively little computational expertise.

So far, ARACNE applications in the literature have been strictly based on microarray expression profile data. However, high-throughput technologies for measuring concentrations of other molecular species, such as microRNA, proteins and their phosphorylated isoforms, phospholipids and metabolites, are being rapidly developed. The methodology described in this protocol should also be applicable to any data set containing measurements of interacting species. In fact, the algorithm might produce even better results when applied to data sets that include more direct measures of interacting species. Thus this protocol could further assist scientists in the analysis of a variety of types of new emerging data. Here, for brevity and coherence, we will discuss only the application to gene expression profile data.

ARACNE can be run either as a command-line executable or through a graphical user interface (GUI). Here we describe the GUI version of ARACNE. Users wishing to employ the command-line version should consult the online **Supplementary Manual**. The command-line version must be used for some more advanced operations that require access to a computational cluster.

Algorithm

ARACNE, including its advantages and limitations, is fully described in ref. 9. Several algorithmic improvements have been introduced since the original version was launched, such as the use of bootstrapping to address the issue of a limited sample size, integration of prior knowledge about genes encoding TFs (Step 8) and several statistical improvements to the MI estimator (online **Supplementary Technical Report**). Overall, these have led to improvements in the statistical significance of the validated target enrichment. The original algorithm[16] is still available using the 'fast' option of the command-line program (online **Supplementary Manual**).

Given the scope of this manuscript, we limit ourselves to the definition of the procedural steps necessary to generate an interaction network from microarray expression profile data. For a set of gene expression measurements that characterize a specific cellular system across diverse phenotypic conditions, the method described herein can be used to infer candidate direct regulatory relationships between gene products, as well as to predict broader functional relationships. ARACNE generates a putative transcriptional network in two computational steps.

First, gene pairs that exhibit correlated transcriptional responses are identified by measuring the MI between their mRNA expression profiles. MI is arguably the best measure of statistical correlation in a non-linear setting[19]. Key elements in this step are determination of the parameters for computation of the MI (i.e., the kernel width of the estimator), and of the MI threshold for statistical independence.

In the second step, ARACNE eliminates those statistical dependencies that might be of an indirect nature, such as between two genes that are separated by intermediate steps in a transcriptional cascade. Such genes will likely have correlated expression profiles, resulting in high MI, and might otherwise be selected as candidate interacting genes. Indirect interactions are eliminated by applying a well known property of MI called the data processing inequality (DPI)[19]. Given a TF, application of the DPI, under appropriate assumptions[9], will thus generate predictions about which other genes might be its direct transcriptional targets or its upstream transcriptional regulators.

After this step, some additional filtering and post-processing procedures might be applied. The final result is a matrix of candidate interactions, also called an adjacency matrix, which can be used for further network visualization and analysis, as discussed in ANTICIPATED RESULTS.

MATERIALS

EQUIPMENT

- ARACNE (http://amdec-bioinfo.cu-genome.org/html/caWorkBench/upload/aracne.zip): ARACNE source code can be downloaded from http://amdec-bioinfo.cu-genome.org/html/caWorkBench/upload/aracne_source.zip
- JDK 1.5 (http://java.sun.com/j2se/1.5.0/download.jsp)
- Computer operating systems: Windows, GNU Linux or Mac OS X (version 10.4 or higher, on a PPC architecture)
- Perl (http://www.perl.org), for users wishing to use the provided scripts described in **Boxes 1** and **2**
- Matlab 7 or higher (http://www.mathworks.com/products/matlab), for users wishing to use our methodologies for mapping *P* values to MI thresholds and for calculating the optimal kernel width for MI estimation
- Cygwin (www.cygwin.com), for command line usage under the Windows operating system

REAGENT SETUP

ARACNE should be used on data sets containing a minimum of 100 microarray expression profiles (or other high-throughput assays). This represents an empirical lower bound on the amount of data needed to estimate the MI reliably. Algorithm accuracy drops rapidly below this threshold. This requirement is not entirely due to ARACNE's reliance on MI as a measure of statistical dependence. In fact, even for simple Pearson correlation-based methods, one would need approximately the same number of samples to establish the significance of the difference between two close correlation values.

It is critical that the cellular systems under investigation explore a significant range of their expression dynamics. This can be achieved either by sampling a variety of naturally occurring cellular phenotypes, or by systematic experimental perturbation with chemical or genomic stimuli, such as a set of pharmacological compounds or small interfering RNAs. Gene expression profiles should be generated using the same experimental protocol, array platform and data pre-processing method, avoiding variability originating from manipulation of the samples. If two channel arrays are used, it is critical that the control RNA is homogeneous across the entire data set.

Microarray data should be formatted as a tab-delimited matrix, with each column representing a microarray experiment and each row representing a microarray probe (throughout this text, we use the term probe to mean the label associated with a variable to be analyzed, e.g., a probe set in Affymetrix microarrays). **Figure 1** demonstrates this data format. The same format can be used if other experimental data are used instead of expression data. For instance, the gene expression values might be replaced by the concentration of specific phospholipids or phosphoproteins as measured by a multicolor flow-cytometry experiment.

Due to the noisy nature of microarray technology, most researchers will typically filter probes that are considered 'uninformative'. For example, genes with low mean expression cannot be accurately measured by microarray technology, and probes that do not display a dynamic range of expression throughout expression profiles cannot be correlated with other probes. Therefore, although this step is not required, before beginning an ARACNE analysis, we suggest that uninformative probes be eliminated from the data set using criteria appropriate for the given microarray technology and data pre-processing procedure. Alternatively, the ARACNE GUI provides input boxes to specify thresholds for the mean and the coefficient of variation for probes to be included in the analysis. Values for these thresholds depend heavily on the data pre-processing procedure employed, as well as the level of noise in each microarray. Duplicate probes on a microarray chip might be used to determine these thresholds.

Col header 1	Col header 2	Sample name 1	Sample name 1	...
Description				
...				
Description				
ProbeID 1	Probe annot 1	3.6	0.5	2.8
ProbeID 2	Probe annot 2	4.5	9.8	5.6
...	...	...	...	...

Figure 1 | Input data format. ARACNE input is formatted as a tab-delimited text file with rows representing variables (e.g., ProbeSets for Affymetrix microarrays) and columns representing samples or observations (e.g., a single microarray experiment). No 'Tab' character should be contained in any entry. The first and second column in the first row contain arbitrary text, such as 'ProbeID' and 'Annotation', and the remaining columns contain a textual representation of the individual microarray expression profiles or experiments (e.g., 'Centroblast sample 1'). After this header row, an arbitrary number of rows can be inserted as long as they start with the string 'Description'. These can be used to store additional information about each sample, such as clinical annotations. For each of the following rows, the first and second column contain a unique identifier (in green) and an annotation (in orange), respectively. For example, the first column might contain the Affymetrix ProbeSet ID, while the second contains the HUGO gene symbol or the Entrez Gene ID associated with the probe set. If multiple variables have the same annotation field (match by string, case sensitive), they will be treated as duplicates and no MI will be computed between them. If an annotation is not available for a variable, use the string '---' in the corresponding field. Alternatively, if no annotations are available, the unique identifiers from the first column can be copied into the second column. No values in the data set may be left blank.

EQUIPMENT SETUP

ARACNE source files are written in the C++ programming language and are compiled to run under Windows, GNU Linux and Mac OS X operating systems. A standard desktop computer is sufficient for analysis of small data sets or reconstruction of networks surrounding a small subset of probes in a larger data set (see ● TIMING). Advanced ARACNE usage, as described in **Boxes 1** and **2**, requires access to a computational cluster, as well as some minimal cluster programming experience. Users wishing to utilize the scripts described in these boxes must have Perl installed on their computers. Users wishing to use our methodologies for mapping *P* values to MI thresholds and for calculating the optimal kernel width for MI estimation must have access to Matlab.

The GUI is implemented in Java and requires JDK 1.5, which should be downloaded (see EQUIPMENT) by clicking on the link 'Download JDK 5.0 update 7' (note that the update number might change if new updates are released after publication of this protocol). The Java installer should set all system variables correctly, including the JAVA_HOME variable, which is required by the program. The version of Java installed on a computer can be determined by typing 'java version' at a command prompt (see ? TROUBLESHOOTING).

ARACNE should be downloaded (see EQUIPMENT), saved to an appropriate directory and uncompressed. This will create a directory called 'aracne' containing all files contained in the ARACNE distribution.

PROCEDURE

Starting the application

1| Launch the application. Navigate to the directory where the ARACNE download has been uncompressed. On Windows machines, double-click the 'launch_aracne.bat' file. On Linux or Mac machines, double-click the 'launch_aracne.sh' file. Alternatively, using a command prompt, navigate to the location where the ARACNE download has been uncompressed and type 'launch_aracne.bat' or 'launch_aracne.sh' for Windows or Linux/Mac machines, respectively. **Figure 2** shows a screenshot of the GUI after it is launched.

2| Load the input data file. After starting the GUI, the user is presented with an empty framework. Right-click on the 'Workspace' icon in the top left of the GUI and click 'New Project'. Right-click on the newly created 'New Project' icon and click 'Open file(s)'. Alternatively, from the menu bar at the top of the GUI, click File → Open → File. A file dialog box will appear. Navigate to the location of the input data ('.exp') file and select it. After the data are loaded, the path to the '.exp' file will appear in the 'Input File' input box in the ARACNE window. Alternatively, this file can be loaded by clicking the 'Load' button located next to the input box. If a dialog box appears that says 'Choose a chip type', either click 'Cancel' or select the appropriate microarray platform (e.g., HG_U95Av2 for the sample data set) from the dropdown box and click 'OK'.

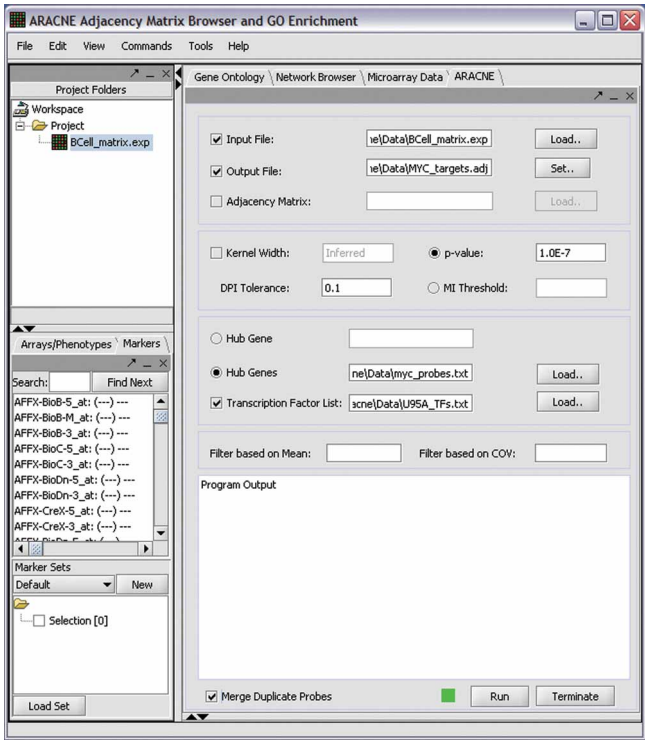

Figure 2 | GUI screenshot with example usage parameters for inferring targets of probes representing *MYC*.

Computing MI scores

3| Estimate pairwise MI values. Any ARACNE analysis begins by identifying the statistical relationships between gene expression profiles by estimating the pairwise MI, an information-theoretic measure of relatedness that is zero if and only if no statistical dependency exists between the variables. MI calculation can be performed by option (**A**), (**B**) or (**C**) depending on the scope of the analysis.
(**A**) To calculate the entire *N*-by-*N* matrix of MI scores, where *N* is the number of probes in the microarray, press the 'Run' button on the GUI with the default parameters. The data are output as an adjacency matrix ('.adj') file, as described in **Figure 3**. To specify the output file, click the 'Set' button next to the 'Output File' input box. Each input box is activated by clicking on the checkbox located next to it. If no output file is specified, one will be created by appending the analysis parameter values to the name of the '.exp' file and changing the extension to '.adj'. For large data sets, computing the entire *N*-by-*N* MI matrix is generally infeasible on a single processor computer (see ● **TIMING**). (**B**) To study probes correlated with a specific probe of interest, type the corresponding probe name into the 'Hub Gene' input box to calculate an *N*-dimensional vector of MI scores between this probe and all other probes in the data set. (**C**) To study a subset of the probes, specify the location of a file containing a list of these probes by clicking the 'Load' button next to the 'Hub Genes' input box. Each line of this file contains one probe identifier, which corresponds to the name of a probe in the input data file. Using this feature computes MI scores between each probe in the list and every other probe in the data set (i.e., if the list contains *K* probes, the program will compute a *K*-by-*N* adjacency matrix).

4| Set the MI estimation parameter. The customizable parameter for the MI estimation algorithm used in ARACNE is the kernel width of the Gaussian estimator, and optimal choices of this parameter will depend on the sample size and statistics of the data set. If none is specified in the 'Kernel Width' input box, default values are automatically generated based on the sample size, using the method described in the online **Supplementary Technical Report**. Consult this report and the Matlab script 'generate_kernel_width_configuration.m', which is provided with the ARACNE distribution, to use our procedure for inferring optimal kernel widths. However, we recommend that the automatically computed values be used for most applications, as ARACNE has been demonstrated to be resilient to suboptimal choices for this parameter[9].

5| Set significance thresholds. The '*p*-value' input box allows the user to specify a '*p*-value' threshold for an MI score to be reported in the output file. There are two options for data analysis that can be used at this point. The first option is to calculate MI values and apply a stringent significance threshold in a single step. However, as MI calculations are computationally intensive, the MI thresholding can also be applied as a post-processing step to a pre-computed adjacency matrix. This option allows pairwise MIs to be calculated only once, using a low threshold, and for different networks to be rapidly generated by applying different values for this parameter.
To perform analysis with a single high significance threshold use option (**A**) or to apply significance thresholds to a pre-computed data set use option (**B**). (**A**) Input the high significance threshold number (e.g., 1e-7) into the '*p*-value' input box to report only MI values that are above the 1e-7 level of significance (i.e., if the genes were independent, there is a probability of 1e-7 of obtaining an MI score equal to or greater than the threshold by chance). Note that scientific notation

>	Parameter name 1	Parameter value 1			
>	...	...			
ProbeID 1	ProbeId 2	0.08	ProbeId 5	0.15	...
ProbeID 2	ProbeId 1	0.08	ProbeId 3	0.22	...
...	...	...	...	...	...

Figure 3 | Adjacency matrix format. The output from ARACNE is a tab-delimited file containing an adjacency list representation of all inferred interactions and their MI scores. The file begins with a number of lines starting with the character '>', each of which contains a parameter value that was used by the algorithm (e.g., the MI threshold). For the remaining rows, the first column (in green) contains the identifier of the probe the interactions of which are being reported in the row. The remaining entries in each row consist of identifier (in orange)–MI value pairs. For example, the row corresponding to the 'ProbeId1' can be read as follows: the MI between 'ProbeId1' and 'ProbeId2' is 0.08, the MI between 'ProbeId1' and 'ProbeId5' is 0.15, and so on. Interactions can be stored symmetrically, as the interaction between 'ProbeId1' and 'ProbeId2' can also be reported in the row corresponding to 'ProbeId2'. Each row can have a different number of entries, depending on the number of interactions for the corresponding probe. Probes that have no ARACNE-inferred interactions will be absent from the output file.

can be used. This *P* value is converted to an MI threshold internally by the program, using the results from our Monte Carlo analysis and extrapolation method on the human B-cell data set (described in ref. 9 and online **Supplementary Technical Report**). To generate alternative conversion parameters, consult this report and the Matlab script 'generate_mutual_threshold_configuration.m' provided with the ARACNE distribution. To directly input an MI threshold rather than a *P* value, type this threshold into the 'MI threshold' input box, which is mutually exclusive with the '*p*-value' input box. The choice of significance threshold will depend on the desired tradeoff between false-positives and false-negatives. However, as the number of tests performed is quadratic in the number of genes, we recommend using a stringent threshold to avoid excessive false-positives. We suggest that a sensible *p*-value threshold can generally be determined by dividing the desired expected number of false-positives (generally a small integer) by the number of tests performed, calculated as the number of distinct probe pairs. For example, a threshold of 1e-7 will lead to about five expected false-positives for a data set with around 10,000 probes, because 10,000 choose 2 (i.e., about 5e7) probe pairs are tested. (**B**) Specify the location of the pre-computed adjacency matrix by clicking the 'Load' button next to the 'Adjacency matrix' input box, and specify a higher significance threshold in the '*p*-value' input box. For example, in this scenario, a low significance threshold (e.g., 1e-2) could have been used in Step 5A and the resulting adjacency matrix used as input here. If this option is specified, MI values will be read from the file rather than calculated.

6| At this point, the user will have a list of probes that are correlated with each probe of interest in the data set. This is similar to many statistical algorithms (in particular, ref. 7) that attempt to infer functional relationships between genes based on similarities in their expression profiles. Thus further analysis can be performed on these results before proceeding to the next step (e.g., analyze co-expressed genes for enriched gene ontology (GO)[20] categories; see ANTICIPATED RESULTS).

Inferring direct statistical interactions

7| Apply the DPI. Many statistical dependencies between gene expression profiles arise from cascades of transcriptional interactions that correlate the expressions of many genes that do not interact directly. ARACNE provides an option to eliminate interactions that are likely to be indirect by applying an information-theoretic property known as the DPI (described in detail in ref. 9). The DPI requires accurate estimation of MI ranks; as MI values cannot be estimated exactly with finite data, a tolerance is used to compensate for errors in the estimate that might affect these ranks. Empirically, values between 0 (no tolerance) and 0.15 (15%) tolerance should be used, as larger values tend to cause high false-positive rates. To apply the DPI, type this tolerance value into the 'DPI Tolerance' input box. By default, the tolerance is set to 1 (100%), so all edges are accepted (i.e., the DPI is not used). Similar to the procedure described in Step 5, the DPI can be applied either during the initial calculation or to a pre-computed adjacency matrix; multiple values for this parameter can be tested using the latter option.

8| Infer transcriptional only networks. We suggest reconstructing transcriptional networks using only genes that are annotated as TFs, so that the DPI cannot eliminate TF–target interactions in favor of interactions consisting of two non-TFs. This partially alleviates the problem associated with highly correlated non-interacting genes, such as those involved in stable complex formation, which violate some of the assumptions required for application of the DPI. This feature is described in greater detail in the online **Supplementary Manual**. The 'Data' folder of the program download includes example files named 'U95A_TFs.txt' and 'U133A_TFs.txt', which contain lists of probes representing TFs in the Affymetrix HG-U95A and HG-U133A microarrays, respectively. If desired, it might be appropriate to use an expanded list that also includes signaling proteins. This allows identification of interactions of a signaling molecule that affects gene expression by post-transcriptional modifications of a TF. To use this feature, click 'Load' next to the 'Transcription Factor List' input box, and select the file containing the list of TFs. If this option is used, the DPI will only be applied using edges containing at least one probe specified in the list.

Data post-processing

9| Merge targets of the same genes. Many microarray platforms contain multiple probes representing the same gene. Selecting the 'Merge Duplicate Probes' checkbox for any run of the program will print out a second adjacency matrix by merging together all probes that represent the same gene, as determined by the second column in the input data ('.exp') file. Common choices for annotations are the Human Genome Organisation (HUGO) gene symbols or Entrez Gene identifiers. The resulting '.adj' file has the same format as the original '.adj' file, but with labels now representing gene annotations (as specified in the '.exp' file) rather than probe identifiers. This file will contain a symmetric matrix; thus each edge will be represented twice in the file. This file has the same name as the original '.adj' file, with '.fused.adj' appended to the end of the file name. If multiple edges are found representing the same gene pair, the highest MI score is reported.

10| Assign edge directionality. One limitation of ARACNE is the lack of edge directionality (i.e., which gene is the regulator in an inferred interaction), although we believe this to be a limitation of most statistical methods that do not use time series data or targeted perturbations. In general, however, we expect that the number of genes regulated by a TF greatly exceeds the

number of TFs that regulate it. Thus when reconstructing a transcriptional only network, most of the inferred interactions should be targets of the TF of interest. In cases where a TF has a characterized binding matrix, these predictions can be strengthened by searching for binding sites in the promoter regions of the predicted targets, using a number of freely available software tools (e.g., Match[21]). However, as with any statistical inference algorithm, predictions should be biochemically validated.

● TIMING

ARACNE is an algorithm with polynomial time complexity; however, it can still be computationally intensive for large data sets (e.g., tens of thousands of genes and hundreds of expression profiles). Here we describe the time usage of ARACNE runs in two different scenarios.

For reconstruction of the entire network, the most time-consuming step is computing MIs between all probes in the data set. This step has complexity $O(N^2M^2)$, where N and M are the numbers of probes and samples, respectively. For example, for a data set with 1,000 probes and 200 samples, this step took about 2,000 s (33 min) on a personal computer (PC) with a Pentium 4 processor, and this number scales quadratically with the numbers of probes and samples. The application of the DPI has complexity $O(N^3)$. The time taken by this step will depend on the MI threshold. However, in a typical ARACNE analysis, the time for this step will be negligible compared with the MI calculation step.

Intuitively, reconstructing the network around a single probe is much faster than reconstructing the whole network. In this scenario, pairwise MIs are first computed between the hub probe and all other probes in the data set, which has complexity $O(NM^2)$. To apply the DPI, MIs must be calculated between all probes that have significant MI with the hub probe, so that the triangle inequality can be applied to all triplets that might potentially eliminate an edge containing the hub probe. Thus if K (out of a total of N) probes survive the MI thresholding, this step will have complexity $O(K^2M^2)$. Unlike the above case, the time spent by this DPI step might be comparable to, or even longer than, the first step, because new MIs need to be computed as triplets of probes are being examined. The value of K depends on both the underlying connectivity of the probe under consideration and the MI threshold.

We provide two general suggestions with respect to the analysis timeline for readers planning to use ARACNE for network reconstruction. First, if the user is interested in only a subset of probes, we recommend reconstructing only the networks around these probes. Second, if the entire network needs to be reconstructed (e.g., for study of the topological properties of the full network or the connectivity beyond first neighbors), we provide scripts (described in **Boxes 1** and **2**) that can make use of a computational cluster for parallel processing.

? TROUBLESHOOTING

See **Table 1**.

TABLE 1 | Troubleshooting table

PROBLEM	POSSIBLE REASON	SOLUTION
The GUI will not start.	The 'JAVA_HOME' environmental variable is not set correctly.	The 'JAVA_HOME' environmental variable is necessary for the program to locate the Java installation. This is usually set automatically by the Java installer. To check whether this variable has been set, for Windows machines open the DOS prompt and type 'echo %JAVA_HOME%', and for Linux/Mac OS X machines, open a terminal and type 'echo $JAVA_HOME'. If this variable is not set or is set incorrectly, it can be set manually using the following procedures. For Windows: (i) Select Start → All Programs → Control Panel. (ii) Double-click 'System'. Select the 'Advanced' tab and click 'Environment variables'. (iii) In the section 'User variables for…', click 'New'. (iv) In the 'Variable name' section enter 'JAVA_HOME'. (v) In the 'Variable value' section enter 'c:\Program Files\Java\jdk1.5.0_07' (or the location where the JDK is installed). (vi) Click 'OK' in all dialog boxes. For Linux/Mac OS X: (i) Open a terminal console. (ii) Type 'echo $SHELL' and press 'Enter'. (iii) If you see '/bin/tcsh', then edit the '.tcshrc' file and add 'setenv JAVA_HOME <path_to_java>'; here, <path_to_java> is the absolute path where the JDK is installed (this path would have been specified when installing the JDK). (iv) If Step (ii) outputs '/bin/bash' then edit the '.bashrc' and add 'export JAVA_HOME=<path_to_java>'; for '/bin/csh', the resource file to edit would be '.cshrc', and so forth.

BOX 1 | GENERATION OF A GENOME-WIDE TRANSCRIPTIONAL NETWORK (ADVANCED USAGE FOR THOSE WITH ACCESS TO A COMPUTATIONAL CLUSTER)

This example and the one in **Box 2** describe how a computational cluster can be used to run the more computationally intensive features of ARACNE. We provide example scripts, located in the 'Scripts' directory of the ARACNE distribution, which are designed for the Rocks 3.2 distribution of Linux and the Sun Grid Engine (SGE) job scheduler. Thus these scripts use the 'qsub' command to submit jobs to the cluster, and this command must be modified based on the user's job-scheduling software. We provide these scripts as a guideline for users who wish to modify them or to write their own scripts to function on their computational system.

Compute MI vectors for each probe

ARACNE can be used to generate large transcriptional networks, either genome-wide or for a particular subset of probes of interest. While the GUI might be used to reconstruct larger networks, to avoid excessive computational time we recommend that a computational cluster be used for the MI estimation step, so that MI vectors for individual probes can be computed in parallel. For users with programming experience, this is done relatively easily using the command-line version of ARACNE (see online **Supplementary Manual**) and submitting a job for each individual probe using the '–h' option. We provide an example Perl script, called 'splitaracne.pl', which demonstrates how such a script might be written and executed. This script takes the same arguments as the ARACNE command-line program, and the '–s' option is required. This script reads the list of probes given by the '–s' option, and, for each probe, submits a job to the cluster with the same program parameters, but with the corresponding probe specified with the '–h' option. An example usage of this script is as follows: 'Scripts/splitaracne.pl -i Data/BCell_matrix.exp -s Data/U95A_TFs.txt -p 1e-2' assuming the user has navigated to the ARCANE download directory.

The file 'U95A_TFs.txt' contains a list of all known TFs in the Affymetrix U95A microarray. Therefore, each job submitted to the cluster will compute MI values for a single probe representing a TF and all other probes in the data set, and print out an '.adj' file containing edges with MI above the specified significance value.

Merge cluster results

The individual files printed in the above step can simply be concatenated, with the header lines removed, to form the complete '.adj' file. An example Perl script, called 'concatadj.pl', to perform this concatenation is also provided. This script takes a single argument, which is the directory path containing the '.adj' files to be concatenated.

Threshold MI scores

The previous step used an MI *P* value of 0.01 to create the adjacency matrix. Once the full matrix is created by merging the individual cluster results, it can be further filtered using either the GUI, as described in ANTICIPATED RESULTS, or an ARACNE command-line option such as the following: 'aracne –i Data/BCell_matrix.exp –j ClusterResults/MI_matrix_p_1e-2.adj –o ClusterResults/MI_matrix_p_1e-7.adj –p 1e-7'. Here, 'MI_matrix_p_1e-2.adj' is the name of the file resulting from the concatenation of the cluster results, and 'ClusterResults' is the name of the directory containing this file. The output file, 'MI_matrix_p_1e-7.adj', will contain only MI scores above the 1e-7 level of significance. Alternatively, the more stringent *P* value could have been specified during the cluster submission step.

Apply the DPI

Once the complete MI matrix has been formed, the DPI can be applied using either the GUI, as described in ANTICIPATED RESULTS, or an ARACNE command-line option such as the following: 'aracne –i Data/BCell_matrix.exp –j ClusterResults/MI_matrix_p_1e-7.adj –o ClusterResults/FilteredMatrix.adj –l Data/U95A_TFs.txt –e 0.10'. This command would produce a new '.adj' file, called 'FilteredMatrix.adj', by applying the DPI with 10% tolerance. To construct a transcriptional only network, we recommend using the '–1' option, followed by a list of all probes on the microarray representing TFs and/or signaling proteins. This option can also be used with a subset of TFs (e.g., those known to be active in a particular tissue type or cellular process). Subsequent analysis on the network can then be performed as described in ANTICIPATED RESULTS.

ANTICIPATED RESULTS

ARACNE has been used to study the transcriptional network associated with the *MYC* proto-oncogene using a data set of Affymetrix HG-U95A microarrays performed on human B lymphocytes derived from normal, tumor-related and experimentally manipulated populations[22]. The file called 'BCell_matrix.exp' in the 'Data' folder of the ARACNE distribution contains a subset of 254 of these microarrays, excluding the experimentally manipulated cell lines and including a small group of new primary tumor samples (16 Burkitt lymphomas, eight follicular lymphomas and 12 splenic lymphomas with villous lymphocytes). *MYC* provides a convenient test case, because its targets have been intensively studied, it has a characterized DNA binding matrix and known activity in B cells. The following section describes how ARACNE can be used for this analysis, and should assist users who wish to adapt this procedure to their own data sets. **Boxes 1** and **2** describe more advanced options for users with access to a computational cluster. If possible, for best results, we recommend using the bootstrapping procedure described in **Box 2**. **Figure 4** displays a flowchart representation of the different ways in which ARACNE can be used.

Reconstruction of the *MYC* transcriptional network

Predictions of transcriptional targets for a single probe (or a small list of probes) can be rapidly generated with a single run of ARACNE. The screenshot in **Figure 2** shows example program parameters. Inputting these parameters and pressing 'Run' will use the 'BCell_matrix.exp' data set to generate predictions of targets for the three probes representing *MYC* on the Affymetrix U95A microarray (as specified in the 'myc_probes.txt' file), using a *P* value of 1e-7 for MI estimation and a DPI tolerance of 10%.

BOX 2 | GENERATION OF A CONSENSUS BOOTSTRAPPING NETWORK

The recommended usage of ARACNE is to employ bootstrapping to generate a consensus network. ARACNE uses bootstrapping, a method for assessing statistical confidence, to build networks that are more robust to errors in the data, and in the estimation of MIs, than regular ARACNE reconstructions. Briefly, experiments (arrays) are randomly sampled from the original data set with replacement and assembled into new 'bootstrapped' data sets containing the same number of experiments as the original. ARACNE is then applied to a large number of such pseudo-data sets to generate a set of bootstrap networks. A consensus network is then constructed that includes edges (i.e., interactions) that are supported across many of the bootstrap networks. We devised a permutation test to determine the inclusion of an edge in the consensus network as follows. We start with edges that are inferred in at least one of the bootstrap networks. For each bootstrap network, we randomly shuffle the positions of its edges while preserving the total number. The distribution of the supports of each edge across such shuffled networks can then serve as a null distribution, against which we can assess the statistical significance of an observed edge support. If the *P* value from this test is smaller than a predefined threshold, the edge will be preserved in the consensus network. Because of its computationally intensive nature, the bootstrapping procedure is only implemented in the command-line version of ARACNE, facilitating submission of jobs to a computational cluster. Two sample Perl scripts have been provided along with the ARACNE program to demonstrate the generation of a consensus bootstrapping network.

Generate bootstrap networks

The first script, 'qsubbootstrap.pl', is used for the submission of bootstrap jobs to computational clusters. The first argument to this script is the name of the output directory, and the second and third arguments specify the range that is used to number the resulting files. The remaining arguments are the same as for a regular ARACNE command-line program. For example, the command 'perl Scripts/qsubbootstrap.pl BootstrapNetworks 1 100 -i Data/BCell_matrix.exp -s Data/U95A_TFs.txt -l Data/U95A_TFs.txt -p 1e-6 –e 0.1' will submit 100 jobs (numbered 1–100) to the cluster nodes and save all of the output bootstrap networks in the directory 'BootstrapNetworks'.

Generate consensus network

Upon completion of the jobs, executing the second script, 'getconsensusnet.pl', will combine all the adjacency matrices and randomly permute the inferred edges in each bootstrap network to perform significance tests. It will then construct the consensus network based on the user-specified *P*-value threshold. This script takes as arguments the directory containing the bootstrap networks and the significance threshold. For example, 'perl Scripts/getconsensusnet.pl BoostrapNetworks 1e-7' will construct a consensus network from all bootstrap networks contained in the 'BootstrapNetworks' directory using significance threshold '1e-7'. Subsequent analysis of the consensus network can be performed as described in ANTICIPATED RESULTS.

The DPI will only be applied to edges containing a probe representing a TF (as specified in the 'U95A_TFs.txt' file) and the results will be output to a file named 'MYC_targets.adj'. Because the 'Merge duplicate probes' checkbox is selected, the program will also output a file named 'MYC_targets.adj.fused.adj' that provides a gene-level representation of the network by merging probes that represent the same gene. The program uses the gene annotations provided in the second column of the '.exp' file to merge probes. Users wishing to obtain a fast result by following the canonical ARACNE pipeline can simply run this procedure and progreed to the analysis described in the section titled 'Analyze putative targets'. Below, we describe a step-by-step approach, which allows users to test multiple program parameters as well as to analyze the intermediate output produced before application of the DPI.

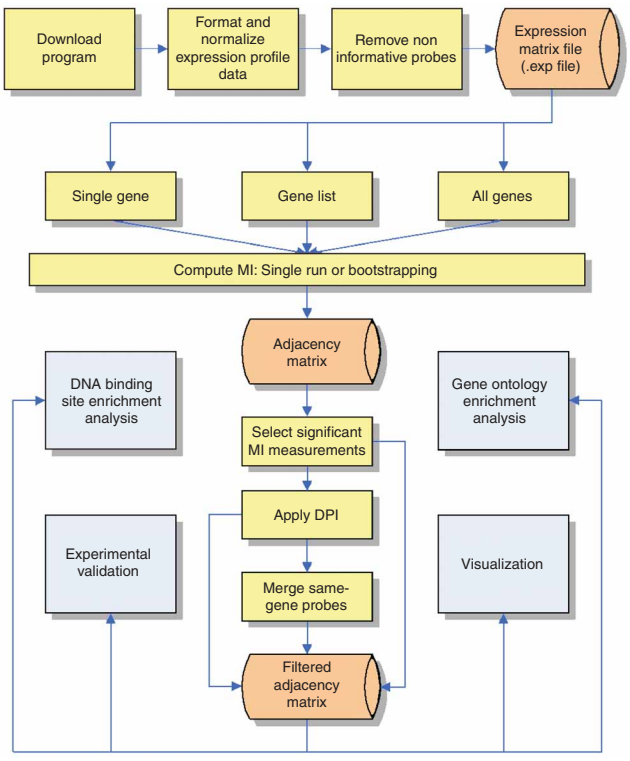

Figure 4 | ARACNE flowchart.

Generate MI matrix

The initial step is to compute MI values between all probes representing *MYC* and all other probes in the data set. The parameters will be similar to those in **Figure 2**. However, in this scenario, the DPI will not be applied, so the 'DPI Tolerance' input box will be left blank. Because MI calculations are computationally intensive, it is often useful to calculate the adjacency matrix initially using a low threshold, so that different significance thresholds can be tested rapidly in subsequent steps. For example, a value of 1e-2 might be used in the '*p*-value' input box (scientific or numerical notation can be used). The MI threshold corresponding to this *P* value is automatically determined as described in the online **Supplementary Technical Report**. The name of the output file should also be changed, for example to 'MYC_MIs_1e-2.adj'. The program completes in ~5 min (all time estimates are for a Pentium 4, 3 GHz processor). The green-colored box, located

next to the 'Run' button, will flash green and red until the program completes, at which point it will remain green. Alternatively, as described above (Step 5A), a more stringent threshold can be specified during this step.

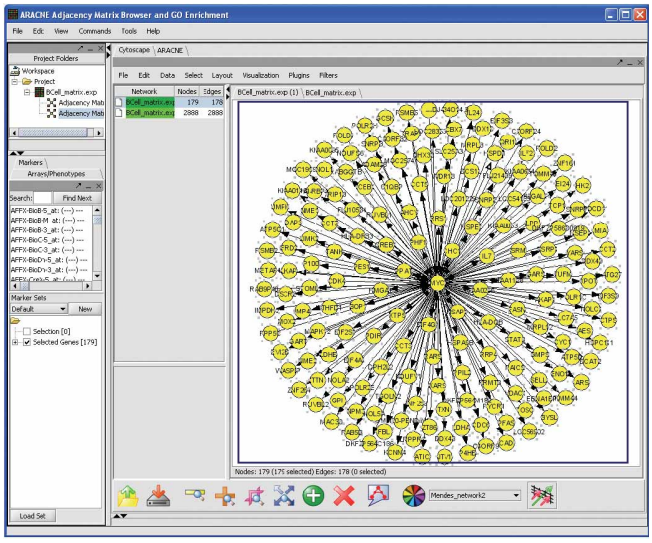

Figure 5 | Cytoscape rendering of the *MYC* network.

Filter genes for statistical significance

The next step is to filter genes using a more stringent significance threshold for MI values. In this step, the adjacency matrix generated in the previous step will be used as an input to the program. Thus this file should be selected by pressing the 'Load' button located next to the 'Adjacency Matrix' input box and navigating to the location of the file. Note that an adjacency matrix used as input for further processing should always contain probe-level data (i.e., it should not contain the suffix '.fusedprobe.adj'). The other parameters are the same as in the previous step, except a lower *P* value (e.g., 1e-7) is used. A different name for the output file should also be specified (e.g.,'MYC_MIs_1e-7.adj'). It is often useful to test a range of thresholds in order to generate a performance curve for the algorithm. This can be done by running this procedure multiple times, although users might want to use the command-line version of ARACNE instead (see online **Supplementary Manual**), and employ a scripting language to automatically test a large range of parameters. If the DPI is not used, the algorithm is much more sensitive to the choice of the significance threshold, because the DPI usually removes many edges with low MI scores.
If further analysis is to be performed on these results, it is often useful to select the 'Merge duplicate probes' check box to print out an '.adj' file in which all probes representing the same gene are merged into single entries. This step completes in several seconds.

Analyze correlated genes

Many traditional microarray analysis algorithms seek to identify gene expression profiles that are statistically correlated within a gene expression profile data set. By analyzing the results produced at this point, ARACNE can be used similarly to these traditional algorithms, while taking advantage of its sophisticated statistical methods, particularly its MI calculation machinery.

One common type of analysis is studying correlated genes to determine whether any GO categories are enriched at a statistically significant level. This can be performed using the ARACNE software, as described in the online **Supplementary Tutorial**.

Identify direct transcriptional interaction candidates

The MI network produced at this point can be processed using the DPI to predict direct transcriptional interactions. The parameters will be similar to those shown in **Figure 2**, except that the (probe level) '.adj' file produced above should be selected in the 'Adjacency Matrix' input box. In order to apply the DPI, for each *MYC* probe, MIs must be calculated between all probes that are significantly correlated with it. Thus this step will be time consuming, as *MYC* is correlated with many other genes. This step completes in ~16 min. Again, to analyze the entire performance space of the algorithm, it is often useful to test a range of parameters for the DPI. The 'filterNetworks.pl' script in the 'Scripts' directory of the distribution provides an example of how a cluster can be used to perform this type of analysis.

Analyze putative targets

The resulting putative *MYC* targets can be analyzed using several methods. Here we describe three commonly used techniques that we recommend in this protocol. First, if the hub TF has a characterized binding matrix in a database such as Transfac[23] or Jaspar[24], testing for enrichment of this binding site in the promoter regions of the putative targets can give an indication of the quality of the predicted targets. Many free software applications exist for searching for binding sites in promoter regions (e.g., ref. 21). Second, it is often useful to obtain a visual representation of the network. **Figure 5** shows such a visualization of the *MYC* network, and the online **Supplementary Tutorial** describes how this figure can be generated using the ARACNE software, which incorporates the Cytoscape software package[25]. Third, searching for GO categories that are enriched among the putative targets of a TF can be useful for characterizing the function of the TF in the cellular context of a particular microarray set or for inferring new functions of uncharacterized TFs. We suggest that this analysis can be a useful starting point for researchers interested in studying the function of an uncharacterized TF. This analysis is implemented within the ARACNE software and is described in the online **Supplementary Tutorial**.

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

ACKNOWLEDGMENTS This work was supported by the National Cancer Institute, the National Institute of Allergy and Infectious Diseases, and the National Centers for Biomedical Computing NIH Roadmap Initiative. A.A.M. is supported by an IBM Ph.D. fellowship and by the National Library of Medicine Medical Informatics Research Training Program. I.N. is supported by the Department of Energy/National Nuclear Security Administration. We would like to thank R. Dalla-Favera for continued support and insight, K. Basso and U. Klein for contributions to the experimental validation of the original ARACNE algorithm, and K. Smith for help in reviewing the manuscript.

COMPETING INTERESTS STATEMENT The authors declare that they have no competing financial interests.

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

1. Schena, M., Shalon, D., Davis, R.W. & Brown, P.O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270**, 467–470 (1995).
2. Lu, J. *et al.* MicroRNA expression profiles classify human cancers. *Nature* **435**, 834–838 (2005).
3. Perez, O.D. & Nolan, G.P. Simultaneous measurement of multiple active kinase states using polychromatic flow cytometry. *Nat. Biotechnol.* **20**, 155–162 (2002).
4. Lu, W., Kimball, E. & Rabinowitz, J.D. A high-performance liquid chromatography-tandem mass spectrometry method for quantitation of nitrogen-containing intracellular metabolites. *J. Am. Soc. Mass Spectrom.* **17**, 37–50 (2006).
5. van Someren, E.P., Wessels, L.F., Backer, E. & Reinders, M.J. Genetic network modeling. *Pharmacogenomics* **3**, 507–525 (2002).
6. Friedman, N. Inferring cellular networks using probabilistic graphical models. *Science* **303**, 799–805 (2004).
7. Butte, A.J. & Kohane, I.S. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac. Symp. Biocomput.* 418–429 (2000).
8. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A. & Nolan, G.P. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308**, 523–529 (2005).
9. Margolin, A.A. *et al.* ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* **7**, S7 (2006).
10. Tegner, J., Yeung, M.K., Hasty, J. & Collins, J.J. Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. USA* **100**, 5944–5949 (2003).
11. Gardner, T.S., di Bernardo, D., Lorenz, D. & Collins, J.J. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* **301**, 102–105 (2003).
12. Segal, E. *et al.* Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.* **34**, 166–176 (2003).
13. Hartemink, A.J., Gifford, D.K., Jaakkola, T.S. & Young, R.A. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pac. Symp. Biocomput.* 422–433 (2001).
14. Gat-Viks, I. & Shamir, R. Chain functions and scoring functions in genetic networks. *Bioinformatics* **19** (Suppl. 1): i108–i117 (2003).
15. Ideker, T. *et al.* Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science* **292**, 929–934 (2001).
16. Basso, K. *et al.* Reverse engineering of regulatory networks in human B cells. *Nat. Genet.* **37**, 382–390 (2005).
17. Hartemink, A.J. Reverse engineering gene regulatory networks. *Nat. Biotechnol.* **23**, 554–555 (2005).
18. Ren, B. *et al.* Genome-wide location and function of DNA binding proteins. *Science* **290**, 2306–2309 (2000).
19. Cover, T.M. & Thomas, J.A. *Elements of Information Theory* (John Wiley & Sons, New York, 1991).
20. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
21. Kel, A.E. *et al.* MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.* **31**, 3576–3579 (2003).
22. Klein, U. *et al.* Gene expression profiling of B cell chronic lymphocytic leukemia reveals a homogeneous phenotype related to memory B cells. *J. Exp. Med.* **194**, 1625–1638 (2001).
23. Matys, V. *et al.* TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.* **34**, D108–D110 (2006).
24. Vlieghe, D. *et al.* A new generation of JASPAR, the open-access repository for transcription factor binding site profiles. *Nucleic Acids Res.* **34**, D95–D97 (2006).
25. Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).