Artificial gene networks for objective comparison of analysis algorithms

Pedro Mendes¹,*, Wei Sha¹ and Keying Ye²

¹Virginia Bioinformatics Institute, USA and ²Statistics Department, Virginia Polytechnic Institute and State University, 1880 Pratt Drive, Blacksburg, VA 24061, USA

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT
Motivation: Large-scale gene expression profiling generates data sets that are rich in observed features but poor in numbers of observations. The analysis of such data sets is a challenge that has been object of vigorous research. The algorithms in use for this purpose have been poorly documented and rarely compared objectively, posing a problem of uncertainty about the outcomes of the analyses. One way to objectively test such analysis algorithms is to apply them on computational gene network models for which the mechanisms are completely known.

Results: We present a system that generates random artificial gene networks according to well-defined topological and kinetic properties. These are used to run in silico experiments simulating real laboratory microarray experiments. Noise with controlled properties is added to the simulation results several times emulating measurement replicates, before expression ratios are calculated.

Availability: The data sets and kinetic models described here are available from http://www.vbi.vt.edu/~mendes/AGN/ as biochemical dynamic models in SBML and Gepasi formats.

Contact: mendes@vt.edu

INTRODUCTION

Whole-genome gene expression analysis has become one of the most popular techniques used in genomics, either in the form of spotted microarrays (Schena et al., 1995) or oligonucleotide chips (Lockhart et al., 1996). These technologies have changed molecular biology research significantly, as they allow experiments to be carried out without preliminary biases on which factors (genes, etc.) are determinant in the phenomenon of interest. Most experiments are based on observing the changes in gene expression levels of most genes of an organism after being challenged with some external perturbation, or comparing such levels between wild type and mutant organisms. A combination of the two can also be used, comparing the gene expression levels of wild type and mutants in time after a perturbation. The advantage of these analyses is also their Achilles heel: the generation of very high-dimensional data sets. An invariant problem is that there are typically over two orders of magnitude fewer observations than factors observed. The data matrices have typically a small number of rows for a very large number of columns. Traditional (univariate) statistical analysis is not adequate and other algorithms are required.

A plethora of analysis and classification algorithms has been proposed to extract information from whole-genome (or large-scale) gene expression data sets, going from fairly standard multivariate statistics to machine-learning and heuristics (Quackenbush, 2001; Slonim, 2002). In many cases, the justifications for their application, their objectives, and the expected outcomes, have been poorly documented. Two approaches have been used to support their usefulness: a) application to experimental datasets where the algorithms were able to recover information that was previously established by independent methods, or b) application to synthetic data sets. Eisen and colleagues used the first approach when they proposed hierarchical clustering for gene expression data (Eisen et al., 1998). While this documentation by example can create confidence in the algorithm, it often suffers from our lack of knowledge about the phenomena under study; therefore conclusions about the method itself are entangled with our uncertainty about the underlying biology. The second approach, which uses synthetic data, has then the advantage that the process that generated the data is well known and so one is able to judge the success or failure of the algorithm (Mendes, 1999). Unfortunately, in the few cases that employed this approach the data sets were different from case to case, not allowing one to judge their relative merits in an objective way. An exception is a comparison made between several clustering algorithms with a synthetic data set (Quackenbush, 2001), but as stated there, the data set used did not reflect the complexity of real gene expression data.

*To whom correspondence should be addressed.
We are interested in objective comparisons between algorithms, in particular with data sets that contain, in a controlled way, the features and complexity of gene expression data. A system to provide such benchmarks must be flexible enough to allow different experiments to be performed, because each analysis method may require different type of experiments. It must, however, do this while preserving the characteristics of the underlying data generation system. For example, if a comparison is to be made between a method that requires time series data with another that uses mutant comparisons, then it must produce a time series for the former, and comparative mutant data for the latter. Here we present a collection of models and synthetic data sets that were constructed specifically to be used as a benchmark for comparing gene expression analysis algorithms. The strategy is based on creating dynamic models of gene networks and emulate any required experiments using a biochemical simulator. These models are artificial gene networks, created taking particular attention to the need to mimic, as much as possible, the characteristics of known gene networks and the existing (real) gene expression data sets. The following sections describe the software system used to generate the data, the supporting theory and algorithms, and the actual data sets.

SYSTEM AND METHODS
To generate the artificial gene expression data sets, a software system was developed that automatically generates gene networks with particular topological and kinetic properties. These networks are embodied in kinetic models, which are used for in silico experiments with the simulator Gepasi (Mendes, 1993, 1997) producing synthetic gene expression data. These data are noiseless in the first instance, but well-defined noise is added using a program written within the SAS® system (SAS Institute, Carey, NC). These noisy data are equivalent to single-channel gene expression measurements (e.g. fluorescence), and thus can be used as a surrogate for Affymetrix™ data. Noise that originates at the level of the array, such as caused by fluctuations in humidity at the time of spotting, is emulated by adding the same random value to all the gene levels of two single-channel measurements. Finally data from two such single-channel sets are combined in ratios, mimicking hybridization ratios like those obtained from spotted microarrays. We make these synthetic data sets available in their final and intermediate stages, allowing for the study of the effect of noise on the performance of the algorithms. Because some analysis algorithms might be dependent on experiments that we simulated yet, we also supply the actual kinetic models in Gepasi input files (which we use) and in the SBML format (Hucka et al., 2002). The latter is becoming a standard exchange format for biochemical models and can be imported by an increasing number of other software packages.

Figure 1 depicts the flow of data in the system, and the individual processes involved. The gene network generator can be one of four optional programs, each generating one specific type of topology. These programs, written in C++, can generate any number of random networks, each one different from the previous but following a common topology. The SBML files generated are provided on the web site, but are otherwise not used further in the system; they are intended for use with simulators other than Gepasi. The Pajek input files are designed for visualization of the networks with the software Pajek (Batagelj and Mrvar, 1998), which is capable of producing high-quality graphics, though only through interactive use. A PERL script constructs a HTML report page for each network generated, including several network metrics. This script uses the GraphViz package (Gansner and North, 1999) for rendering images of the networks, and gnuplot (http://www.gnuplot.info/) to draw a plot of the distribution of connectivity in the network. Another PERL script was designed to carry out ‘experiments’ with each network. It uses one Gepasi file (the ‘wild type’) describing a full dynamic model of one gene network and alters it in defined ways to produce a series of other Gepasi files, each representing a different ‘mutant’ or environmental perturbation. It then calls the simulator Gepasi to produce steady state and/or time course data that are noiseless. Finally, a SAS® program takes each of the result files and adds noise to the data according to the Gaussian or gamma distributions, and calculates ratios of gene expression of the mutant or perturbed networks against the wild type.

ALGORITHMS
Dynamic models of gene networks
Gene networks are conceptual models of genetic regulation where each gene is considered to be directly affected by a number of other genes (Brazhnik et al., 2002), and can be represented as directed graphs. In the present approach genes are represented by their expressed messages, and the model explicitly includes the processes of transcription and message degradation. The interactions between genes are then represented as inhibitions or activations of the rate of transcription of one mRNA by other mRNA species. Unlike other popular gene network models (Kauffman, 1969b) the rates of transcription and degradation are here represented as continuous variables. Thus, the levels of mRNA species are also continuous variables and depend on the balance of transcription and degradation. Mathematically, these network models are based on sets of coupled differential equations. There is one differential equation for each gene in the network, and their
derivations of an overall rate law for transcription is outside the scope of this work. Instead, we opted to construct an empirical rate law to represent it. Given that free nucleotides are not represented explicitly in the models (which is equivalent to assuming their concentrations are constant), then the rate law of transcription does not take include them. In this framework, changes in the rate of transcription of any one gene come about by changes in concentration of a few other gene products. Given that the number of gene copies for each gene is limited, and transcription is catalyzed by a limited number of transcription complexes, the rate law must saturate both with activators or inhibitors. We have encapsulated these characteristics in a general rate law for transcription:

\[
s_i(G_1, K, G_n) = V_i \cdot \prod_j \left( \frac{K_i^{n_j}}{I_j^{n_j} + K_i^{n_j}} \right) \times \prod_k \left( 1 + \frac{A_k^{n_k}}{A_k^{n_k} + K_a^{n_k}} \right). \tag{2}\]

A variation of this rate law can also be used:

\[
s_i(G_1, K, G_n) = V_i \cdot \prod_j \left( \frac{K_i^{n_j}}{I_j^{n_j} + K_i^{n_j}} \right) \times \prod_k \left( \frac{A_k^{n_k}}{A_k^{n_k} + K_a^{n_k}} \right). \tag{3}\]

In both forms, the inhibitors \( I_j \) and activators \( A_k \) act independently of each other. \( V_i \) is a basal rate of transcription, i.e. that when there is no action of inhibitors or activators. The constants \( K_i \) and \( K_a \) represent concentrations at which the effect of the inhibitor or activator is half of its saturating value. The exponents \( n_i \) and \( n_k \) regulate the sigmoidicity of the curve, in the same way as Hill coefficients in enzyme kinetics: when unity the curve is a hyperbola, values larger than unity result in a sigmoidal shape. In fact the overall behavior of these rate laws is very similar to the behavior of Hill kinetics (Hill, 1910; Hofmeyr and Cornish-Bowden, 1997). Figure 2 represents the behavior of these two rate laws in the presence of one inhibitor or one activator. The actual number of inhibitors \( (j) \) and activators \( (k) \) depends on the topology of the network, discussed below. For the mRNA degradation steps we use plain mass action kinetics (which has a single rate constant). The total number of kinetic parameters of these networks is then \( 2(n+l) \) where \( n \) is the number of genes, and \( l \) is the total number of gene interactions.

**Network topologies**

To fully define a gene network model it is also necessary to create a network topology, or wiring diagram. This means...
deciding the number of genes involved in the network, and which genes act as inhibitors and activators of the transcription of each gene. In essence, to decide how many and which \( j \) and \( k \) terms of equation (2) or (3) are included in the rate laws, and finally the values of all the constants. Past research in gene networks has concentrated on a particular topological class: random gene networks, for which Kauffman was a pioneer (Kauffman, 1969a,b, 1974). These random networks follow a topology studied earlier by mathematicians Erdős and Rényi (1959), where each vertex of a graph is equally likely to be connected to any other vertex in the graph. Kauffman’s networks contained solely two parameters: the number of genes and the number of connections (modifiers) per gene. One major difference between Kauffman’s networks and ours is that we use continuous kinetics, while he used discrete (Boolean) functions. This topology is interesting, at least for historical reasons, and it is one of the topologies that our models can follow.

Research in theory of networks has been increasing at a fast pace, in part because of the growth of computer networks, both in size and their importance in society. Watts and Strogatz described a type of network whose properties lie between random networks and regular lattices (Watts and Strogatz, 1998). The property known as ‘six degrees of separation’ is the major property of this topology: the average distance between any two vertices in the graph is small, and the networks are known as ‘small-worlds’. Wagner and Fell studied the metabolic network of *E.coli* and provided evidence that it displays a similar property (Wagner and Fell, 2001). Given that many links of gene networks pass through metabolic reactions, it seems important to study gene networks organized as small-worlds.

Barabási and Albert proposed another type of networks to be pervasive in sociology, technology, and science, which they named ‘scale-free’ (Barabasi and Albert, 1999). Such networks are characterized by the majority of vertices having only a few connections, while a small number of vertices have a very large number of connections (hubs). To be more precise, the distribution of vertices according to their connectivity in these networks follows a power law. The same group showed that metabolic networks also display this property (Jeong et al., 2000). Featherstone and Broadie provide some evidence, based on expression profiles of yeast mutants, that this is also true at the level of gene networks (Featherstone and Broadie, 2002). Although this topic may still be controversial, it seems that scale-free topologies are perhaps the best suited for describing gene networks, and so they are important for the present study.

Our software system includes four different programs to generate gene networks. Each of them uses a different algorithm producing networks that follow one of the topologies mentioned above. A fourth topology, regular grids (Watts and Strogatz, 1998), is also included for completeness, though perhaps not so relevant for gene networks. For Erdős-Rényi networks, the algorithm first creates all
the genes, and then chooses a predetermined number of other genes, at random (with uniform distribution), that interact with it. To create small-world gene networks, we implemented the algorithm described by Watts and Strogatz (1998); this is also used to create 1-dimensional regular grid gene networks, as that is the first step in the small-world algorithm. The scale-free networks are created with the algorithm described by Albert and Barabasi (2000). The networks can be generated with rate laws following equations (2) or (3). In all cases, the generators set all basal rate constants ($V_i$ in equations (2) and (3)) to the same value; all affinity constants ($K_a$ and $K_i$) to the same values; all Hill coefficients ($n_a$ and $n_i$) to the same value; and all degradation rate constants to the same value. Thus, all the genes in the network have similar kinetics, differing only in the number and nature of the interactions. It is possible, and straightforward, to set all these constants to different values within the program Gepasi, so the networks are not necessarily constrained in this way. All four gene network generators chose if each interaction should be an activation or an inhibition at random (sampled from a uniform distribution) with a specified probability. The user can then control the proportion of inhibitions versus activations. It is not clear what proportion of genetic regulatory interactions are inhibitions or activations. Savageau has developed a theoretical basis that established some properties of either mode of regulation (Savageau, 1976, 1977), but it is not clear yet what is this proportion in genomes.

**Experimental strategies**

The network generators described above create kinetic models that we consider to be wild types (as if each was a different ‘biological species’). The wild type model can be used in many different ways, just as actual biological material can be manipulated in different experiments. For this work, we subject our wild-type models to two different in silico experiments. The first is the generation of null mutants: for a particular network, each single gene can be silenced by simply setting the parameter $V_i$ of the corresponding differential equation to zero. This prevents the mRNA of that gene ever being expressed. In this experiment, we generate all null mutants, one at a time, to later compare their steady-state gene expression levels to the wild type’s. The second experiment is a time course of the network following an environmental perturbation. To emulate the external perturbation a number of kinetic parameters of the rate laws (Equations (2) and (3)) are changed with a random magnitude. In this case we a zero-centered random value is added to a number of the $K_a$ and $K_i$ constants. The perturbation would be one that would change the inhibition and activation status of some transcription steps in different directions, which is possibly the case of some environmental factors, such as exposure to UV radiation slightly above what cells are adapted to. In this case the systems are followed for a rather small number of time points, just as in most realistic settings.

**Biological and analytical variance**

One of the objectives of this exercise is to study the effects of noise on the performance of gene expression analysis algorithms. As such it is important to insert sources of variance in the data, given that the numerical solution of ODEs is deterministic. Two sources of variance are important for this study: biological variance, which in real biological systems arises from genetic polymorphisms and from different environmental conditions. In the case of our artificial gene networks there is no difference between these two sources (both would affect the kinetic parameters). As such biological variance is created by adding random values to the $K_a$, $K_i$, $n_a$ and $n_i$ constants. These random values are to be small, as opposed to the environmental perturbations mentioned in the previous section.

A second type of variance appears due to measurement noise. In this case we do not add the variance in the model before the simulations, but rather after them, directly on the results. Because in microarrays one detects the fluorescence of each dye separately, each of these measurements suffers from Gaussian noise. Microarray technology is sufficiently complex that several sources contribute noise in significant amounts. To reproduce this, we add noise at the level of the array, i.e. a common factor for all genes in one particular replicate. We also add noise at the level of the gene, which would have been generated by uneven surface coating of the slide, by defects in the scanner, by differences in the sample preparation reactions, among others. This gene-level noise also includes all other random effects not taken into account. In this case one random number is drawn per gene. Both sources are present at the same time. Given that noise is added before calculating expression ratios, it will suffer all distortions introduced by the analysis, just like in real settings. Microarray data often displays non-Gaussian noise, and so it may be useful to add noise of different types to these synthetic data. A possibility would be the gamma function, which is not symmetric around the mean.

**IMPLEMENTATION**

To properly assess the differences of analysis algorithms, and to identify the causes of any discrepancies, it is important to create data sets that vary in different factors, one at a time. Several data sets were created where a single, or at most two, factors vary, the group of data sets that follow one such variation is named a series. Table 1 lists the data sets that are available on our website, together with the corresponding ‘wild-type’ models (http://www.
Artificial gene networks

Fig. 3. Graphs of two artificial gene networks in the data set. (A) a 1000-gene network generated with the Erdős–Rényi topology (JumboRND-001); (B) a 1000-gene scale-free network (JumboSF-001). Both networks contain 1000 interactions, to an average of one interaction per gene.

The series of data sets provided attempt to cover a number of features that might be important factors in gene expression data. The Century series compares three topologies and provides a large number of networks of each type, becoming the prime candidate for any benchmarks. The Jumbo series provides networks of 1000 genes, which approximates the dimension of microarray data sets. The Web series is designed to test the effect of density of connectivity, as it includes networks with an increasing average number of connections per gene. The Coop series tests the effect of increasing nonlinear kinetics, which is made by increasing the exponents $n_j$ and $n_k$ of equation (2).

Because these gene networks are generated with random connectivity, using single networks as benchmarks is essentially meaningless (especially at high noise levels). Benchmarks should concentrate on the average results from each data set, and it is for this reason that the most of the series are composed of several networks. We strongly recommend that use of these data be made with proper statistical analysis.

DISCUSSION

The kinetics of the networks generated is arguably simpler than that of real systems. They can easily be improved by setting different values for each single kinetic constant. This would generate a hierarchy of time scales, which has indeed been observed (Selinger et al., 2003). Additionally, different rate laws for transcription could include non-additive interactions between the inhibitors and activators. In real genetic systems the products of some genes are absolutely required for transcription to take place (e.g. RNA polymerase), and their absence cannot be counteracted by increased expression of other activators (e.g. transcription factor). Another detail that could be improved is the rate law for mRNA degradation, which is largely dependent on enzymes but in our models is spontaneous. All these improvements may add realism to the networks and render them more suitable for the purposes described herein. These gene network models, as simplistic as they may seem, are already capable of generating synthetic gene expression data with characteristics like those of experimental results. As such they are a good starting point for providing the kind of objective evaluation of algorithms intended.

The work described here is similar to ongoing research in computer networking. Several computer network simulators are available that use methods similar to our gene network simulators (Calvert et al., 1997; Jin et al., 2000; Medina et al., 2001). To study the Internet one needs to
Table 1. The benchmark data collection

<table>
<thead>
<tr>
<th>Series</th>
<th>Data Set</th>
<th>Topology</th>
<th>Models</th>
<th>Genes</th>
<th>Links</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Century</td>
<td>CenturySF</td>
<td>AB</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 1$</td>
</tr>
<tr>
<td></td>
<td>CenturySW</td>
<td>WS</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 2$</td>
</tr>
<tr>
<td></td>
<td>CenturyRND</td>
<td>Erdős</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 4$</td>
</tr>
<tr>
<td>Jumbo</td>
<td>JumboSF</td>
<td>AB</td>
<td>5</td>
<td>1000</td>
<td>1000</td>
<td>$n_i = n_a = 1$</td>
</tr>
<tr>
<td></td>
<td>JumboRND</td>
<td>Erdős</td>
<td>5</td>
<td>1000</td>
<td>1000</td>
<td>$n_i = n_a = 2$</td>
</tr>
<tr>
<td>Web</td>
<td>Web50</td>
<td>AB</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>$n_i = n_a = 4$</td>
</tr>
<tr>
<td></td>
<td>Web100</td>
<td>AB</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>$n_i = n_a = 8$</td>
</tr>
<tr>
<td></td>
<td>Web200</td>
<td>AB</td>
<td>50</td>
<td>50</td>
<td>200</td>
<td>$n_i = n_a = 8$</td>
</tr>
<tr>
<td>Coop</td>
<td>CoopSF1</td>
<td>AB</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 1$</td>
</tr>
<tr>
<td></td>
<td>CoopSF2</td>
<td>AB</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 2$</td>
</tr>
<tr>
<td></td>
<td>CoopSF4</td>
<td>AB</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 4$</td>
</tr>
<tr>
<td></td>
<td>CoopSF8</td>
<td>AB</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 8$</td>
</tr>
<tr>
<td></td>
<td>CoopSW1</td>
<td>WS</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 1$</td>
</tr>
<tr>
<td></td>
<td>CoopSW2</td>
<td>WS</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 2$</td>
</tr>
<tr>
<td></td>
<td>CoopSW4</td>
<td>WS</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 4$</td>
</tr>
<tr>
<td></td>
<td>CoopSW8</td>
<td>WS</td>
<td>10</td>
<td>100</td>
<td>200</td>
<td>$n_i = n_a = 8$</td>
</tr>
</tbody>
</table>

Data sets are organized in series; each data set contains a number of models. Each model represents a wild-type network and undergoes one or two experiments with or without ‘biological’ replicates. All models were generated using the kinetics of equation (2). Parameters are $n_i = n_a = 1.5$, $K_i = K_a = 0.01$, $V_i = 1$, equal probability of inhibitions and activations, except where noted. Topologies: AB created with the Albert-Barabási algorithm, WS with the Watts–Strogatz algorithm, and Erdős with the Erdős–Rényi topology.

use methods much like those of biological research, even though it is a man-made system. Its level of complexity is such that requires sampling and experimentation (Paxson and Floyd, 1997). It is not surprising then, that similar methods be applicable to biochemical networks. This is the strategy that was followed here; the gene network generators were developed to study how topological and kinetic properties of such networks may influence their behavior. In the future, this may facilitate the discovery of underlying principles of biochemical regulation at the level of transcription. A first step is indeed to use such networks to investigate the performance of analysis algorithms.

The system presented here generates data sets that hopefully will help establishing objective comparisons of algorithms for large-scale gene expression analysis. The system is also useful to carry out computational studies of gene regulatory networks, and the properties that may be associated with particular features of these networks, such as the topology or kinetics of the interactions. Future developments in this research include exploring large numbers of diverse artificial gene networks to search for particular properties similar to real gene networks, and expanding the system to protein interaction networks and eventually metabolic networks.

ACKNOWLEDGEMENTS

We thank Alberto de la Fuente and Paul Brazhnik for helpful discussions. We are grateful to the NSF Plant Genome Research Program (grants DBI-0109732 and DBI-0217653) and the NIGMS (1 RO1 GM068947-01) for financial support of this work.

REFERENCES


