Inferring Cellular Networks Using Probabilistic Graphical Models

Nir Friedman

Science **303**, 799 (2004);
DOI: 10.1126/science.1094068

41. S. Nee, *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **355**, 1607 (2000).
42. R. M. May, *Stability and Complexity in Model Ecosystems* (Princeton Univ. Press, Princeton, NJ, 1973).
43. B. Sinervo, C. M. Lively, *Nature* **380**, 240 (1996).
44. B. Kerr, M. A. Riley, M. W. Feldman, B. J. M. Bohannan, *Nature* **418**, 171 (2002).
45. S. A. Frank, in *Foundations of Social Evolution*, J. R. Krebs, T. H. Clutton-Brock, Eds. (Princeton Univ. Press, Princeton, NJ, 1998).
46. E. Sober, D. S. Wilson, *Unto Others: The Evolution and Psychology of Unselfish Behavior* (Harvard Univ. Press, Cambridge, MA, 1998).
47. M. A. Nowak, R. M. May, *Nature* **359**, 826 (1992).
48. A. Sasaki, W. D. Hamilton, F. Ubeda, *Proc. R. Soc. London Ser. B* **269**, 761 (2002).
49. P. D. Taylor, L. Jonker, *Math. Biosci.* **40**, 145 (1978).
50. J. Hofbauer, K. Sigmund, *Evolutionary Games and Population Dynamics* (Cambridge Univ. Press, Cambridge, 1998).
51. J. W. Weibull, *Evolutionary Game Theory* (MIT Press, Cambridge, MA, 1996).
52. C. Hauert, S. De Monte, J. Hofbauer, K. Sigmund, *Science* **296**, 1129 (2002).
53. T. Vincent, B. Brown, *Evolutionary Game Theory, Natural Selection, and Darwinian Dynamics* (Cambridge Univ. Press, Cambridge, 2003).
54. D. Fudenberg, K. Levine, *The Theory of Learning in Games* (MIT Press, Cambridge, MA, 1998)
55. R. Ferriere, R. Michod, *Am. Nat.* **147**, 692 (1996).
56. U. Dieckmann, R. Law, J. A. J. Metz, Eds., *The Geometry of Ecological Interactions: Simplifying Spatial Complexity* (Cambridge Univ. Press, Cambridge, 2000).
57. R. Durrett, *SIAM (Soc. Ind. Appl. Math.) Review* **41**, 677 (1999).
58. R. Cressman, *Evolutionary Dynamics and Extensive Form Games* (MIT Press, Cambridge, MA, 2003).
59. J. A. J. Metz, R. M. Nisbet, S. A. H. Geritz, *Trends Ecol. Evol.* **7**, 198 (1992).
60. S. D. Mylius, O. J. Diekmann, *J. Theor. Biol.* **211**, 297 (2001).
61. R. Boyd, J. Lorberbaum, *Nature* **327**, 58 (1987).
62. M. A. Nowak, K. Sigmund, *Acta Appl. Math.* **20**, 247 (1990).
63. J. Hofbauer, K. Sigmund, *Appl. Math. Lett.* **3**, 75 (1990).
64. U. Dieckmann, R. Law, *J. Math. Biol.* **34**, 579 (1996).
65. A. P. Hendry, J. K. Wenburg, P. Bentzen, E. C. Volk, T. P. Quinn, *Science* **290**, 516 (2000).
66. U. Dieckmann, P. Marrow, R. Law, *J. Theor. Biol.* **176**, 91 (1995).
67. I. Eshel, *J. Theor. Biol.* **103**, 99 (1983).
68. F. B. Christiansen, *Am. Nat.* **138**, 37 (1991).
69. P. D. Taylor, *Theor. Pop. Biol.* **36**, 125 (1989).
70. M. A. Nowak, *J. Theor. Biol.* **142**, 237 (1990).
71. J. A. J. Metz, S. A. Geritz, G. Meszena, F. J. A. Jacobs, J. S. van Heerwarden, in *Stochastic and Spatial Structures of Dynamical Systems*, S. J. Van Strien, S. M. Verduyn Lunel, Eds. [Koninklijke Nederlandse Academie van Wetenschappen (KNAW), Amsterdam, 1996], pp. 183–231.
72. S. A. Geritz, J. A. J. Metz, E. Kisdi, G. Meszena, *Phys. Rev. Lett.* **78**, 2024 (1997).
73. K. Parvinen, *Bull. Math. Biol.* **61**, 531 (1999).
74. I. Eshel, *J. Math. Biol.* **34**, 485 (1996).
75. R. Bürger, *Am. Nat.* **160**, 661 (2002).
76. P. Hammerstein, *J. Math. Biol.* **34**, 511 (1996).
77. M. Slatkin, *Genetics* **93**, 755 (1979).
78. U. Dieckmann, M. Doebeli, *Nature* **400**, 354 (1999).
79. C. Matessi, A. Gimelfarb, S. Gavrilets, *Select. Mol. Genes Memes* **2**, 41 (2001).
80. M. Gyllenberg, K. Parvinen, U. Dieckmann, *J. Math. Biol.* **45**, 79–105 (2002).
81. L. Van Valen, *Evol. Theory* 1, 1 (1973).
82. For an interactive Web page, see www.univie.ac.at/virtuallabs.
83. K. M. Page, M. A. Nowak, *J. Theor. Biol.* **219**, 93 (2002).
84. Support from the Austrian Science Fund WK 10008, the Packard Foundation, and J. Epstein is gratefully acknowledged.

REVIEW

Inferring Cellular Networks Using Probabilistic Graphical Models

Nir Friedman

High-throughput genome-wide molecular assays, which probe cellular networks from different perspectives, have become central to molecular biology. Probabilistic graphical models are useful for extracting meaningful biological insights from the resulting data sets. These models provide a concise representation of complex cellular networks by composing simpler submodels. Procedures based on well-understood principles for inferring such models from data facilitate a model-based methodology for analysis and discovery. This methodology and its capabilities are illustrated by several recent applications to gene expression data.

Research in molecular biology is undergoing a revolution. The availability of complete genome sequences facilitates the development of high-throughput assays that can probe cells at a genome-wide scale. Such assays measure molecular networks and their components at multiple levels. These include mRNA transcript quantities, protein-protein and protein-DNA interactions, chromatin structure, and protein quantities, localization, and modifications. These rich data illuminate the working of cellular processes from different perspectives and offer much promise for novel insights about these processes (*1*).

The challenge for computational biology is to provide methodologies for transforming high-throughput heterogeneous data sets into biological insights about the underlying mechanisms. Although high-throughput assays provide a global picture, the details are often noisy, hence conclusions should be supported by several types of observations. Integration of data from assays that examine cellular systems from different viewpoints (for instance, gene expression and protein-protein interactions) can lead to a more coherent reconstruction and reduce the effects of noise. To perform such an integration, however, we must understand the biological principles that couple the different measurements. In addition, the conclusions of the analysis should go beyond a mere description of the data and should provide new knowledge about the relevant biological entities and processes, ideally in the form of concrete, testable hypotheses.

To answer this challenge, we need to build models of the biological system. A model is a simplifying abstraction. It generates predictions of system behavior under different conditions (as reflected by observations) and illuminates the roles of various system components in these behaviors. We focus on probabilistic models, which use stochasticity to account for measurement noise, variability in the biological system, and aspects of the system that are not captured by the model.

In a model-based approach to data analysis, we start by defining the space of possible models that we are willing to consider. This modeling decision depends on the phenomena we wish to describe and how they are reflected by the observations. We then use a learning procedure to select the model that best fits the actual observations. (Such procedures are referred to by different names in different disciplines, including inference, estimation, reverse engineering, and system identification.) Finally, we use the learned model to reason about the data, make predictions, and glean insights and hypotheses.

An important aspect of model-based approaches is the shift from a procedural methodology to a declarative one. In a procedural method, we focus on the sequence of steps from the data to the conclusions. For example, when relating transcription factor binding sites in the promoter regions

School of Computer Science and Engineering, Hebrew University, 91904 Jerusalem, Israel. E-mail: nir@cs.huji.ac.il

of genes to their expression profiles, we can start by finding clusters of coexpressed genes and then search for overrepresented elements in the promoters of the genes in each cluster (*2*). Alternatively, we can group genes with similar binding sites in their promoter regions, and then test whether they are coexpressed (*3*).

In contrast, in a declarative approach we start by designing a model that encompasses both gene expression and binding sites. As explained below, such a model explicitly describes the assumptions we make about the relations between the two types of data. By doing so, we clarify what kinds of predictions we can perform with each model, after we learn its parameters from the data and how these parameters relate to the biological phenomena we are attempting to capture. Next, we apply well-understood principles such as maximum likelihood estimation to fit the model to the data. This can be done using a general-purpose strategy (such as Expectation Maximization, gradient ascent, or Gibbs sampling) in the context of the particular model. By treating different data sets within one model, the learning procedure can combine evidence from multiple data sets and reach more robust conclusions.

The model-based approach is widely used in biological sequence analysis (*4*), for which there is a range of established sequence models such as Hidden Markov Models. For cellular networks, the structure of the underlying processes that generate the observed measurements is not fully characterized, and the question of which mechanism to model—and at what granularity—is open-ended and depends on the biological question we are attempting to answer.

In this review, I examine the use of a class of mathematical models known as probabilistic graphical models (*5*, *6*) for model-based analysis of cellular networks. These models were developed in the fields of machine learning and statistics for modeling complex systems with multiple interacting entities. They are closely related to probabilistic sequence models but are not restricted to sequential observations.

Probabilistic graphical models are suitable for this task for several reasons. They provide a concise language for describing probability distributions over the observations. The computational procedures for reasoning in graphical models are derived from basic principles of probability theory. In addition, the literature on graphical models provides approaches to learning from data that are derived from basic well-understood principles in statistics. These approaches allow the use of observations to "fill in" many model details. Furthermore, they provide principles for combining multiple local models into a joint global model. This provides flexibility when constructing models for novel data sets or experimental designs. Using graphical models, one can construct simple submodels and then combine them for the full model. Finally, the declarative nature of graphical models provides an advantage when we need to extend the model to account for additional aspects of the system or new observations.

My emphasis is on the modeling choices and how they facilitate different analysis tasks. For each of these models, we also need to apply inference and learning procedures. In some cases, we can adopt generic strategies. In others, finding computationally efficient algorithms is a major challenge, and the details of the algorithms (not discussed here) greatly affect the results.

Probabilistic Graphical Models

When modeling a biological system, we are interested in entities that are involved in the system under study (e.g., genes) and their different attributes (e.g., expression level). In a probabilistic model, we treat each of these attributes as a random variable (*7*). Random variables include observed attributes, such as the expression level of a particular gene in a particular experiment, as well as hidden attributes that are assumed by the model, such as the cluster assignment of a particular gene. A model embodies the description of the joint probability distribution of all the random variables of interest.

Probabilistic graphical models represent multivariate joint probability distributions via a product of terms, each of which involves only a few variables. The structure of the product is represented by a graph that relates variables that appear in a common term. This graph specifies the product form of the distribution and also provides tools for reasoning about the properties entailed by the product (*5*). For a sparse graph, the representation is compact and in many cases allows effective inference and learning.

In Bayesian Networks, the joint distribution over a set $X = \{X_1, \ldots, X_n\}$ of random variables is represented as a product of conditional probabilities. A Bayesian network associates with each variable X_i a conditional probability $P(X_i|\mathbf{U}_i)$, where $\mathbf{U}_i \subseteq X$ is the set of variables that are called the parents of X_i. Intuitively, the values of the parents directly influence the choice of value for X_i. The resulting product is of the form

$$P(X_1, \ldots, X_n) = \prod_i P(X_i|\mathbf{U}_i) \qquad (1)$$

The graphical representation is given by a directed graph where we put edges from X_i's parents to X_i (Fig. 1, A to C). If the graph is acyclic, the product decomposition of Eq. 1 is guaranteed to be a coherent probability distribution.

Bayesian networks appear naturally in several domains in biology. In pedigree analysis, for example, the joint distribution of genotypes in a pedigree is a product of conditional probabilities of the genotype of each individual given the genotypes of its two biological parents. In phylogenetic models, the probability over all possible sequences during evolution is the product of the conditional probability of each sequence given its latest ancestral sequence in the phylogeny.

To specify a model completely, we need to describe the conditional probability associated with each variable. In general, any statistical regression model may be used. For example, we can consider models where each $P(X_i|\mathbf{U}_i)$ is a linear regression of X_i on $\mathbf{U}_i$. Alternatively, we can use decision trees to represent the probability of a discrete variable X_i given the values of its parents. The choice of a specific parametric representation of the conditional probabilities is often dictated by our knowledge or assumptions about the domain.

Another class of models are Markov Networks, which represent a joint distribution as a product of potentials. Each potential captures the interactions among a (small) set of variables and specifies the "desirability" of joint value assignments to these variables. This results in a product of the form

$$P(X_1, \ldots, X_n) = \frac{1}{Z}\prod_j \pi_j[\mathbf{C}_j] \qquad (2)$$

where $\pi_j[\mathbf{C}_j]$ is the jth potential over the variables $\mathbf{C}_j \subseteq X$, and Z is a normalizing constant that ensures that the total probability mass is 1 (Fig. 1, D to F). A canonical example of an undirected model is an Ising model, where each random variable represents the orientation of an element (e.g., magnetic particle) and the potential between pairs of elements captures the compatibility of two elements. The joint probability is determined by the overall compatibility of each assignment of values according to all the potentials.

Another related class of models are Chain Graphs, which involve a product of conditional probabilities and potentials. In many domains there is additional structure, beyond the product form, that can be exploited for concise representation (*8*). Below, we discuss one such class of models that captures additional structure.

A crucial question for the tasks we examine here is inferring, or "learning," models from observations (*6*, *8*). The general aim is to learn a model that is as close as possible to the underlying distribution from which the observations were made. We distinguish two main tasks: parameter estimation and model selection. In parameter estimation, we learn the parameters of the conditional probabilities for a given model structure. This task is often addressed as a

maximum likelihood problem. In model selection, we select among different model structures to find one that best reflects the dependencies in the domain. This task is often addressed as a discrete optimization problem where we try to maximize a score that measures how well each candidate structure matches the observed data (*22*).

Once we specify or learn a model that describes a joint distribution, we can use it to compute the likelihood of our observations and make predictions about the value of unobserved random variables given these observations. There is a wide choice of exact and approximate methods for answering such queries. These exploit, when possible, the structure of the product form for efficient computation (*5*, *9*).

From Clustering to Regulation

We now consider a graphical model for gene expression data and then examine how to extend it to the modeling of cis-regulatory elements. The main sources of high-throughput data on the behavior of cellular networks are gene expression profiles, obtained using DNA microarrays. A typical data set reports the expression level of thousands of genes as measured by several dozens or hundreds of arrays. We treat these as observations of the values of random variables $X_{g,a}$, where g is an index over genes and a is an index over arrays.

A fairly simple modeling assumption is that genes can be partitioned into clusters of coexpressed genes, and that the genes in each cluster have a typical expression level in each array. We might also assume that arrays are partitioned into array clusters, which capture relevant biological context, and that the expression of a gene is roughly the same in arrays that belong to the same array cluster. We can pose this model by adding random variables, so that $GeneCluster_g$ denotes the cluster assignment of gene g, and $ArrayCluster_a$ denotes the cluster assignment of array a. The underlying assumption is that the expression of gene g in array a depends on the value of $GeneCluster_g$ and $ArrayCluster_a$. This model assumes that all measurements that correspond to a particular gene cluster–array cluster pair are governed by the same conditional distribution.

We can describe such a model as a Bayesian network. The structure of the Bayesian network is regular, in the sense that each expression attribute $X_{g,a}$ has parents $GeneCluster_g$ and $ArrayCluster_a$. The actual network structure depends on the number of genes and arrays in the data set (see Fig. 2A for a small example). The description by a Bayesian network, however, does not explicitly represent two important aspects. First, the random variables denote attributes of different entities, such as genes and arrays. Second, a general scheme or template is shared by all entities of the same type. For example, the conditional probability of $P(X_{g,a}|GeneCluster_g, ArrayCluster_a)$ is similar for different choices of g and a. By capturing such regularities, we can provide a more concise representation of the model. One such representation is the language of relational Bayesian networks (*10*, *11*). Figure 2B shows a template model for the clustering problem, from which we can generate the Bayesian network once we are given the set of genes and arrays.

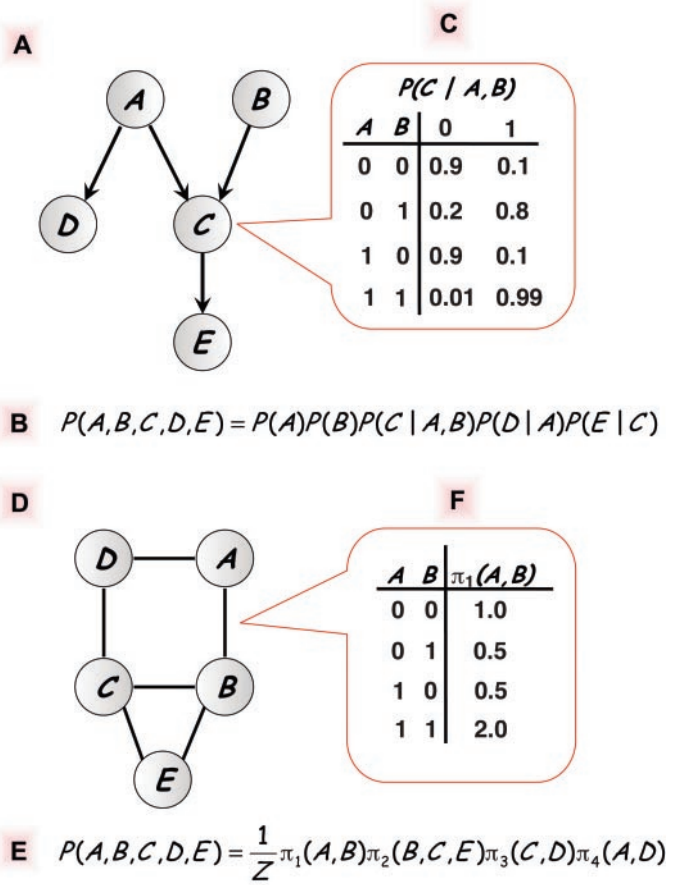

Fig. 1. (**A**) A Bayesian network over five binary random variables. Vertices are labeled with random variable names (*A* to *E*); edges correspond to direct dependencies. (**B**) The product form specified by this Bayesian network structure. A full specification of the joint distribution for these random variables requires 31 parameters; this product form requires 10 parameters. (**C**) An example of one conditional distribution in the product form that specifies $P(C|A,B)$. (**D**) A Markov network over the same five variables. (**E**) A product form that induces this Markov network structure. This is a product of four potential functions, each a function of a subset of the variables. (**F**) One potential in this product form.

The model just described can achieve high likelihood if the cluster and gene assignment partitions the original measurements into blocks with approximately uniform expression within each block (*11*). We can learn such a partition by using an Expectation Maximization procedure that iterates between an E-step, which uses current parameters to find the probabilistic cluster assignment of genes and arrays, and an M-step, which reestimates the distribution within each gene/array cluster combination on the basis of this assignment.

This basic model can be extended to capture additional insight about the biological mechanisms. We consider one example here. It is common to assume that coexpression of genes reflects coregulation. A key regulation mechanism involves binding of transcription factors to promoter regions of genes. Thus, we aim to identify the transcription factor binding sites in the promoter region of genes that can explain observed coexpression. To do this, we extend the model to include observations about promoter sequences. A straightforward approach is to maintain the clustering model, where genes in the same cluster have similar expression patterns, and in addition associate each cluster with the transcription factors that regulate it. Although such a description oversimplifies the biological mechanism, it can capture the first-order signal while ignoring many differences between the expression and regulation of specific genes.

There are several ways to translate this intuition into a mathematical model. One approach is to annotate promoters with characterized binding sites, and then use these as new attributes of the gene entity. The random binary variable $R_{g,j}$ denotes whether gene g has a binding site of transcription factor j. Then, we can design a model where the cluster assignment of each gene directly influences the associated binding sites and expression attributes (*12*, *13*).

Alternatively, we can attempt to maximize the likelihood of gene expression profiles given the promoter region content. Again, we introduce binding site attributes to the gene entity, but now we assume that the gene cluster depends on these attributes (*14*, *15*). Such a model focuses on binding sites that predict expression and does not attempt

to account for the occurrences of other binding sites, which might be relevant under conditions that are not represented in the expression data. We can augment this model by modeling the probability of a binding site given the actual promoter sequence. We introduce new entities that denote the promoter regions, and model $R_{g,j}$ as depending on the promoter sequence Seq_g (Fig. 2D). The parameters of this conditional probability characterize the specific motif recognized by the transcription factor. This extension allows us to learn the characterization of the binding site while learning how its presence influences gene expression.

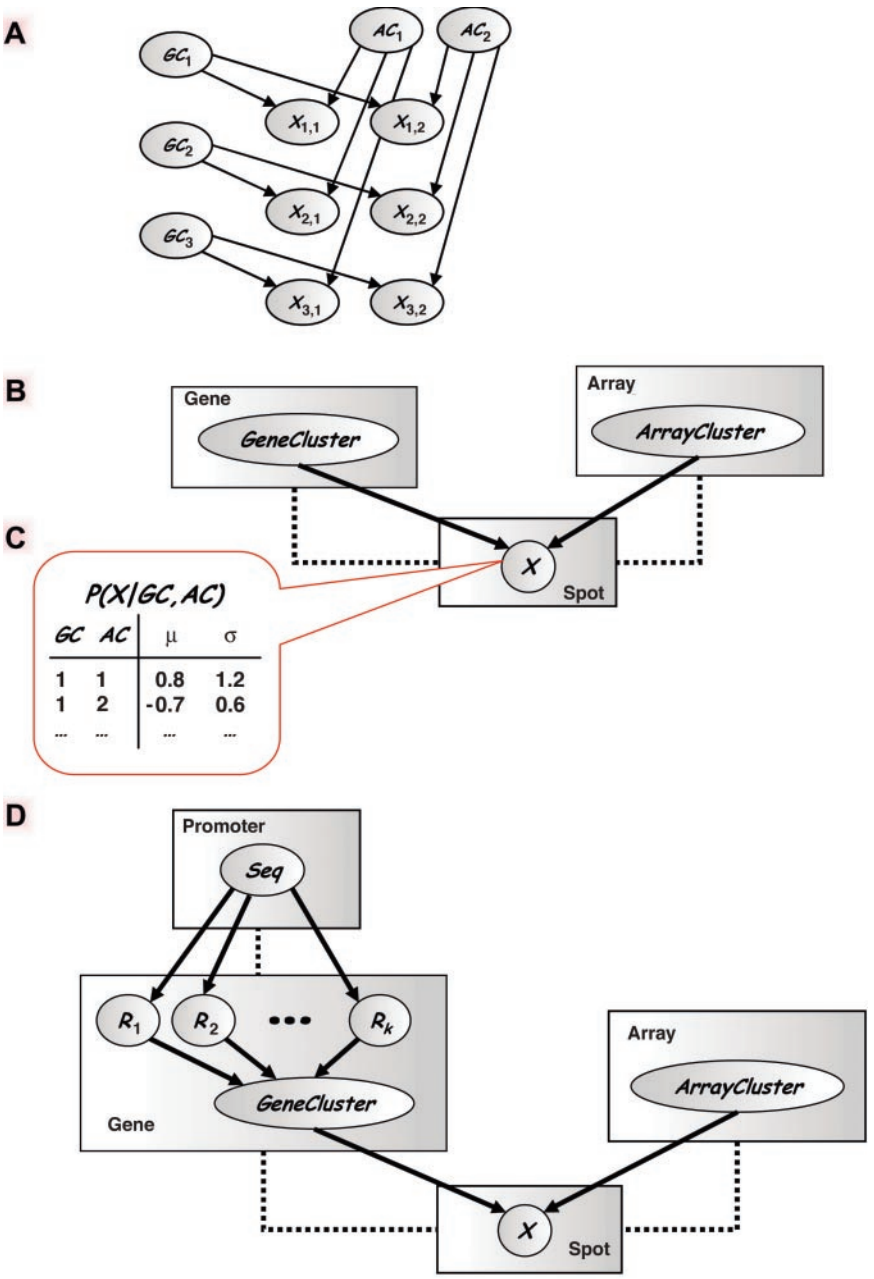

Fig. 2. (**A**) A Bayesian network for the clustering problem in a simple data set with three genes and two arrays. The random variable $X_{g,a}$ denotes the spot that measures expression of gene g in array a, GC_g denotes the cluster of gene g, and AC_a denotes the array cluster of array a. (**B**) A relational Bayesian network template for the basic clustering model. Boxes delimit entity types; dotted lines correspond to relations between entities (e.g., the relation between a spot entity and a gene entity denotes that the spot measures the expression of the particular gene). Each spot entity has a single attribute that measures an expression level and is associated with a gene entity and an array entity. Each gene entity is associated with a gene cluster, and each array entity with an array cluster. (**C**) Representation of the conditional distribution of expression levels given the clusters of the corresponding gene and array. Each combination of values of the respective gene-cluster and array-cluster variables is associated with parameters of a Gaussian distribution (mean μ and standard deviation σ). (**D**) A template model that also includes promoter sequences. Each gene entity is associated with a promoter entity that has a sequence attribute that reports the DNA sequence of the promoter. The gene entity now has new attributes, where R_j indicates whether the gene is regulated by the jth transcription factor. This indicator depends on the promoter sequence. The cluster of the gene depends on the combination of these indicators.

A crucial detail in building such a model is the representation of the conditional distributions associated with $GeneCluster_g$. This distribution describes how the existence of binding sites in the promoter region determines (or predicts) what cluster the gene belongs to. The conditional probabilities explored so far involve fairly generic representation of decision trees (*14*) or additive votes (*15*). Both representations manage to reconstruct some aspects of yeast transcriptional circuits. However, it is not clear whether either one matches the underlying logic in biological regulation.

Learning in this type of model combines similarity of genes in terms of expression and in terms of their promoters. In training the model, there are steps where we find new binding sites that explain assignments of genes to clusters, steps that reassign genes to new clusters on the basis of both their expression profile and their promoter region, and steps that reestimate the distributions of expression within each cluster. Thus, learning involves information flow between the two types of data and allows the combination of weak evidence from both sources. This information is channeled through the gene cluster variables. Thus, to achieve high likelihood, the gene cluster variables must represent clusters of genes with both coherent expression profiles and similar promoters. At the same time, the learning procedure must identify the binding site motifs that are most predictive of these cluster assignments.

Segal *et al*. (*15*) applied such a model to two data sets of yeast gene expression profiles. One involved ~800 genes in 77 arrays of different yeast cell cycle stages (*16*); the other involved ~1000 genes in 173 arrays of yeast under environmental stress conditions (*17*). They showed that, without using prior knowledge, their procedure identified several dozen binding site motifs. More than half of these corresponded to motifs that have been characterized in the literature; the resulting gene clusters corresponded to known biological processes and function annotations. This correspondence is significantly more pronounced than for standard clusters learned from gene expression alone. In addition, their model suggests a specific cis-regulatory circuit that in many cases corresponds to prior knowledge about regulation in yeast.

The clustering model can be extended to involve other types of mechanisms and ob-

servations. For example, we can assume that active transcription factor binding sites should correspond to observations of transcription factor location data (*18*). We can extend the model to view these observations as a noisy sensor (*14*). Another orthogonal extension can incorporate protein-protein interactions. For example, a pair of interacting proteins are more likely to belong to the same coregulated cluster (because they might act together). To capture this, we can assign a Markov network pairwise potential that prefers coordinated cluster assignments for pairs of interacting proteins (*19*). This model can combine annotation of proteins (e.g., cluster assignment) with protein-protein interaction data. This pairwise interaction model can be applied to other types of protein annotations. Deng *et al*. (*20*) recently used a similar model for predicting functional annotations of proteins.

Reconstruction of Regulatory Networks

A key challenge in gene expression analysis is the reconstruction of regulatory networks. A simple thought experiment suggests the following formulation. If the expression of gene *A* is regulated by proteins *B* and *C*, then *A*'s expression level is a function of the joint activity levels of *B* and *C*. Because of variability in underlying biology and measurement noise, we treat the expression of *A* as a stochastic function of its regulators. This suggests a Bayesian network where the expression level of each gene depends on the activity levels of its regulators. In most current biological data sets, however, we do not have access to measurements of protein activity levels. Hence, we resort to using expression levels of genes as a proxy for the activity level of the proteins they encode. This is a problematic assumption, as there are numerous examples where an activation or silencing of a regulator is carried out by posttranscriptional protein modifications.

With this caveat in mind, we set out to find a Bayesian network that relates the expression level of a gene to those of its regulators (*21*). That is, we search for a Bayesian network that specifies for each gene *g* a set of regulators, so that in each array $X_{g,a}$ depends on the expression level of the regulators in that array. We then use tools for structure learning in Bayesian networks (*22*, *23*) to determine the network architecture. This involves considering different network structures and evaluating the likelihood that they have generated the observations.

This general outline faces two main challenges. The first challenge involves statistical robustness. Building a network that involves thousands of genes from several dozen examples of their joint expression levels (arrays) is extremely problematic. Such a small number of examples does not suffice to distinguish between true correlations and spurious ones. There are several strategies to deal with this challenge. Methods such as the bootstrap can identify significant network features that are robust to perturbations of the observations (*21*, *24*). Another approach is to use prior knowledge about biological principles to restrict the set of network structures we are willing to consider (*25*, *26*). This reduces the number of competing "false" structures and increases robustness. Alternatively, we can restrict ourselves to evaluating a much smaller set of structures on the basis of prior biological knowledge about specific genes (*27*). Finally, we can rely on biological principles for restricting the stochastic function of a gene, given its regulators, to be of a particular form (*26*–*29*).

The second and more difficult challenge is the biological interpretability of the results. Can we really distinguish regulation from coexpression? Do these methods discover direct or indirect regulation? How do unobserved posttranscriptional events affect the conclusions? Whereas our ultimate goal is to identify the direct regulation of targets by transcription factors, experience shows that the methods also find many other indirect relations.

As a specific example, we applied (*24*) the bootstrap procedure to a data set of the expression profiles of 565 genes in 300 knockout variants of yeast (*30*). This was done in an ab initio fashion, without any prior knowledge about which genes might be regulators and without making strong assumptions about the network structure (Fig. 3A). We used the bootstrap procedure to assign confidence to different relations in the learned networks, and compiled subnetworks of genes that have high-confidence interconnections among them (Fig. 3B). We then compared the significant regulation relations to the experimental literature. This analysis showed that many regulator-target pairs correspond to known regulatory relationships that involve some intermediate steps—for example, components of mitogen-activated protein kinase signaling cascades and their downstream transcriptional targets.

Two studies have used this insight as a justification to focus on regulators that include components of signal pathways, receptors, and transcription factors. Pe'er *et al*. (*25*) introduced a method that examines only those networks in which a small number of regulators explain the expression of all other genes (Fig. 3C). The restriction to such a network architecture forces the learning procedure to identify the most pronounced regulators in the data set. It also simplifies the learning procedure, which leads to statistical and computational advantages. They applied this method to several yeast gene expression data sets (*16*, *17*, *30*) and then performed a systematic validation of the regulators in the learned networks by examining the process and function annotation of the target set of each regulator (Fig. 3D). In most cases, the significant annotations matched the known literature about the regulators.

Segal *et al*. (*26*) introduced a method that examines networks composed of modules of coregulated genes (Fig. 3E). In these networks, all the genes within a single module are controlled by a shared regulatory program. The learning procedure simultaneously identifies the composition of the modules and finds regulators for each module that best predict the expression of its genes. The module network representation forces the learning procedure to find patterns that are shared by many genes. Such a representation is congruent with biological principles that suggest that the cellular regulatory circuits coordinate activation or repression of groups of genes that are involved in the same process. This is in contrast to the two approaches we described above, where each gene is associated with an individual regulatory program. The benefit of the module network approach is that shared regulation programs require far fewer parameters and increase the robustness of the learned model. The downside is that we lose flexibility and might miss fine points in the regulation of specific genes. Another important aspect of the module networks is that the representation leads to an easier interpretation. Instead of examining the regulation program of hundreds of genes, we focus on a much smaller number of modules.

This approach was applied to the 2450 genes in 173 arrays of yeast under environmental stress conditions (*17*) to learn a network with 50 modules (*26*). This was followed by a systematic evaluation against the literature, which showed that regulators of 35 modules agree with experimental results in the literature and with evidence based on gene annotation and cis-regulatory motifs (Fig. 3F). Moreover, the model learned by this method leads to testable hypotheses of the form "protein *X* regulates a module of genes *G* under condition *C*." Segal *et al*. confirmed such hypotheses for three unknown regulators (one transcription factor and two kinases) by examining gene expression of knockout strains in the specific conditions where the regulator is predicted to be active.

These results show that regulation can be learned from expression profiles. Clearly, because these methods examine expression profiles, they detect coordinated changes in transcript levels of the target genes and their regulators. This suggests that to understand the successes of such methods, we need to examine how the discovered regulators are

themselves transcriptionally regulated. These regulation mechanisms often involve feed-forward loops and feedback mechanisms (*18*, *31*) that change the mRNA expression level of regulators when their protein activity changes. As a consequence, we can detect coordinated changes in the expression levels of regulators and their targets. This hypothesis is supported by an analysis of the discovered regulatory relations against a database of protein-DNA and protein-protein interactions (*26*).

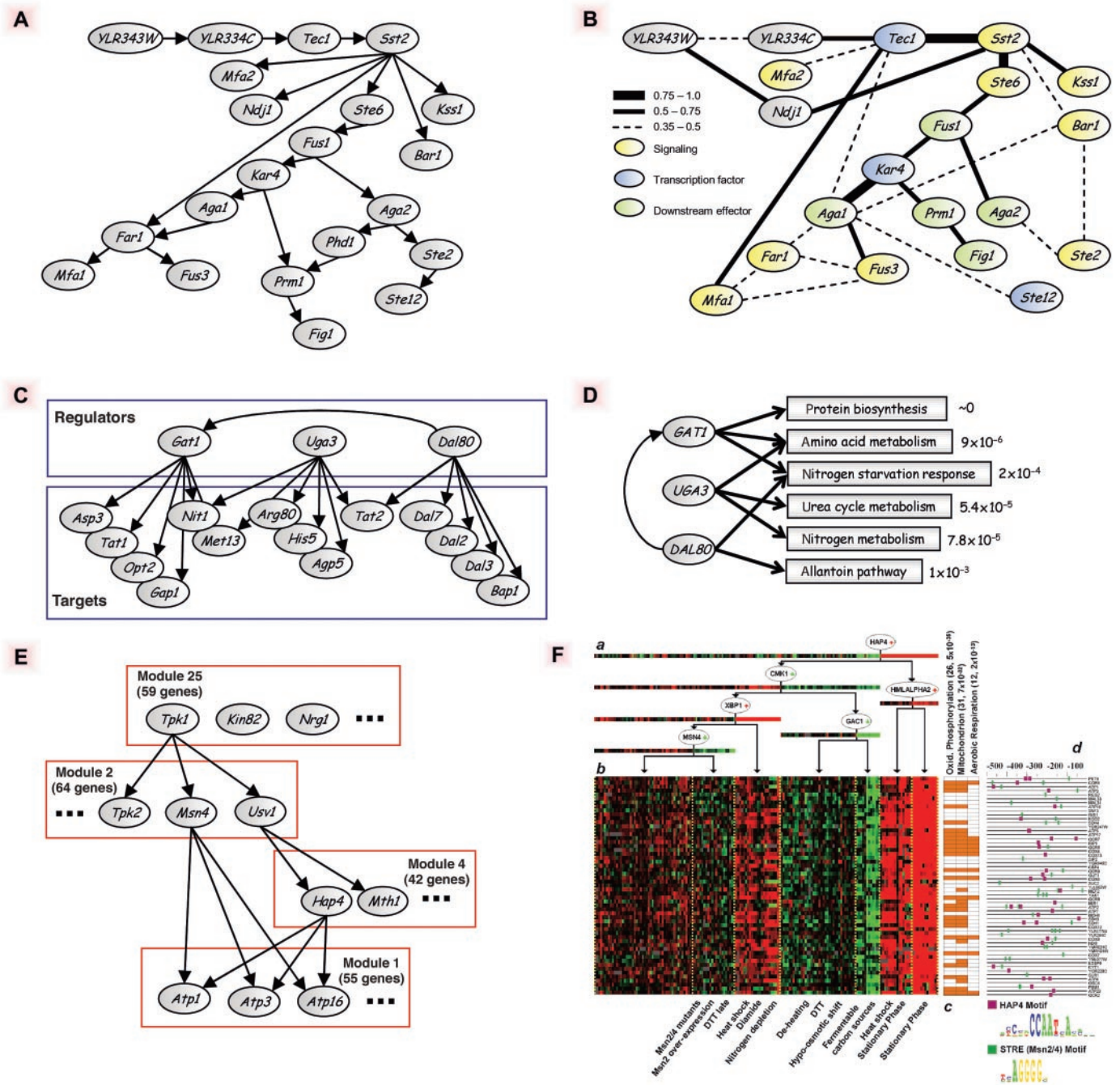

Fig. 3. Different regulatory network architectures. (**A**) An unconstrained acyclic network where each gene can have a different regulator set. This is a fragment of a network learned in the experiments of Pe'er *et al*. (*24*). (**B**) A summary of direct neighbor relations among the genes shown in (A) based on bootstrap estimates. Degrees of confidence are denoted by edge thickness. We automatically identify a subnetwork of genes, with high-confidence relations among them, that are involved in the yeast-mating pathways. The colors highlight genes with known function in mating, including signal transduction (yellow), transcription factors (blue), and downstream effectors (green). (**C**) A fragment of a two-level network described by Pe'er *et al*. (*25*). The top level contains a small number of regulators; the bottom level contains all other genes (targets). Each gene has different regulators from among the regulator genes. (**D**) Visualization of significant Gene Ontology (*42*) annotations of the targets of different regulators. Each significant annotation for the targets of a regulator (or pairs of regulators) is shown with the hypergeometric *p*-value. (**E**) A fragment of the module network described by Segal *et al*. (*26*). Each module contains several genes that share the same set of regulators and share the same conditional regulation program given these regulators. (**F**) Visualization of the expression levels of the 55 genes in Module 1 (b) and their regulators (a). Significant Gene Ontology annotations (c) and cis-regulatory motifs in promoter regions of genes in the module (d) are shown. [See figure 3 of (*26*); reproduced with permission]

One of the key challenges is to distinguish regulation from coexpression. We can improve these distinctions by careful experimental design and by combining additional data sources. An important strategy for dissecting the direction of regulation is by gene knockout perturbations. The intuition is that by knocking out a gene, we can pinpoint genes downstream from it. The Bayesian network semantics can use such perturbations to also infer the direction of regulation in genes upstream of the knockout (*32*, *24*), and modeling such perturbations can lead to stronger conclusions (*24*). An alternative experimental strategy for distinguishing correlation from causation is to examine how the system changes over time (*33*–*35*). For example, Kim *et al*. (*34*) showed that a learning temporal model of yeast cell cycle expression data could reduce the errors made by a model that did not take temporal observations into account. Finally, we can bias the model to prefer regulator-target pairs that are consistent with additional data sources, such as transcription factor location data (*36*).

Conclusion

We have discussed several model-based approaches for learning cellular networks from data. These approaches represent the state of the art for this task and were evaluated through extensive validation against prior biological knowledge and independent experimental assays. A recurring theme in these approaches is an exploration of the tradeoff between two contradictory aims. On one hand, we aim to reconstruct detailed models; on the other hand, we must be able to learn these models from the available data. As a result of this tradeoff, the methods we discussed capture only the aspects that were deemed most crucial to the biological question at hand.

There are two main strategies for obtaining models that provide deeper biological insight. Much current research focuses on unified models that combine evidence from different levels of the cellular machinery. A key strength of the graphical models is the ability to specify models that account for such heterogeneous observations. These models can reach conclusions that are not supported by either data source considered independently. We discussed a unified model that finds transcription factor binding sites and simultaneously characterizes the behavior of their target genes. Similarly, we expect that modeling promoter sequences will lead to more accurate regulatory models [e.g., (*37*)]. Another direction is to incorporate data from assays in proteomics. For example, recent works (*38*, *39*) combine protein-DNA and protein-protein interaction maps to reconstruct regulatory circuits that explain differential expression in gene knockout experiments.

Another key ingredient for improving our models is our current understanding of the biological regulatory mechanisms. As we have seen, ab initio exploration can lead to insights about the nature of the data, such as the ability to infer indirect regulation. However, when possible, incorporating biological principles into the design of the models (e.g., constraining regulatory network structure) can restrict the degrees of freedom during learning and result in better models. One of the most intriguing prospects for research is to develop models that capture aspects of the actual details of the regulatory machinery. This includes combining tools from a large body of research on kinematic equations and stochastic differential equations for modeling cellular pathways (*40*, *41*).

In the near future, we expect to see an explosion in the quantity and diversity of high-throughput data sets, including new experimental assays, new experimental designs, and examinations of systems at the levels of a single cell, a composite organ, a whole organism, and a society. Computational analysis methods will be critical for gleaning biological insight from these data sets. It is clear that no single tool will meet all the analysis needs; instead, we need a range of tools tailored to specific assays and experimental designs. To cope with these challenges, the field of computational biology must develop methodologies and concrete implementations that empower researchers to explore different models of varying detail and rapidly apply them to diverse data sets. The language of graphical models is well suited for composing different submodels in a principled and understandable fashion. The declarative semantics of graphical models provide foundations for building a "modeling toolbox" and unified learning algorithms that apply well-understood statistical principles. Such tools will enable researchers to combine components and to tailor learning procedures. The challenge is to gain an understanding of the modeling choices for different cellular components and their suitability for different types of assays, and to extend the methods for inference and learning in such models.

References and Notes

1. E. Lander, *Nature Genet.* **21**, 3 (1999).
2. S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, G. M. Church, *Nature Genet.* **22**, 281 (1999).
3. Y. Pilpel, P. Sudarsanam, G. Church, *Nature Genet.* **29**, 153 (2001).
4. R. Durbin, S. Eddy, A. Krogh, G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* (Cambridge Univ. Press, Cambridge, 1998).
5. J. Pearl, *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufmann, San Francisco, 1988).
6. M. I. Jordan, Ed., *Learning in Graphical Models* (MIT Press, Cambridge, MA, 1998).
7. A glossary of this and other technical terms appears on *Science* Online.
8. W. Buntine, *J. Artif. Intel. Res.* **2**, 159 (1994).
9. F. V. Jensen, *An Introduction to Bayesian Networks* (Univ. College London Press, London, 1996).
10. N. Friedman, L. Getoor, D. Koller, A. Pfeffer, *Relational Data Mining*, S. Dvzeroski, N. Lavrac, Eds. (Springer-Verlag, Berlin, 2001), pp. 307–337.
11. E. Segal, B. Taskar, A. Gasch, N. Friedman, D. Koller, *Bioinformatics* **17** (suppl. 1), S243 (2001).
12. Y. Barash, N. Friedman, *J. Comp. Biol.* **9**, 169 (2002).
13. I. Holmes, W. Bruno, *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 202 (2000).
14. E. Segal, Y. Barash, I. Simon, N. Friedman, D. Koller, *Proceedings of the 6th Annual International Conference on Computational Molecular Biology*, G. Myers *et al.*, Eds. (ACM Press, New York, 2002).
15. E. Segal, R. Yelensky, D. Koller, *Bioinformatics* **19** (suppl. 1), I273 (2003).
16. P. T. Spellman *et al.*, *Mol. Biol. Cell* **9**, 3273 (1998).
17. A. P. Gasch *et al.*, *Mol. Biol. Cell* **11**, 4241 (2000).
18. T. Lee *et al.*, *Science* **298**, 799 (2002).
19. E. Segal, H. Wang, D. Koller, *Bioinformatics* **19** (suppl. 1), I264 (2003).
20. M. Deng, T. Chen, F. Sun, *Proceedings of the 7th Annual International Conference on Computational Molecular Biology*, M. Vingron *et al.*, Eds. (ACM Press, New York, 2003), pp. 95–103.
21. N. Friedman, M. Linial, I. Nachman, D. Pe'er, *J. Comp. Biol.* **7**, 601 (2000).
22. D. Heckerman, D. Geiger, D. M. Chickering, *Mach. Learn.* **20**, 197 (1995).
23. P. Spirtes, C. Glymour, R. Scheines, *Causation, Prediction, and Search* (Springer-Verlag, Berlin, 1993).
24. D. Pe'er, A. Regev, G. Elidan, N. Friedman, *Bioinformatics* **17** (suppl. 1), S215 (2001).
25. D. Pe'er, A. Regev, A. Tanay, *Bioinformatics* **18** (suppl. 1), S258 (2002).
26. E. Segal *et al.*, *Nature Genet.* **34**, 166 (2003).
27. A. Hartemink, D. Gifford, T. S. Jaakkola, R. A. Young, *Pac. Symp. Biocomput.* **6**, 422 (2001).
28. S. Imoto, T. Goto, S. Miyano, *Pac. Symp. Biocomput.* **7**, 175 (2002).
29. H. Toh, K. Horimoto, *Bioinformatics* **18**, 287 (2002).
30. T. R. Hughes *et al.*, *Cell* **102**, 109 (2000).
31. S. Shen-Orr, R. Milo, S. Mangan, U. Alon, *Nature Genet.* **31**, 64 (2002).
32. G. Cooper, C. Yoo, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, K. Laskey, H. Prade, Eds. (Morgan Kaufmann, San Francisco, 1999), pp. 116–125.
33. I. Ong, J. Glasner, D. Page, *Bioinformatics* **18** (suppl. 1), S241 (2002).
34. S. Y. Kim, S. Imoto, S. Miyano, *Brief. Bioinform.* **4**, 228 (2003).
35. B. E. Perrin *et al.*, *Bioinformatics* **19** (suppl. 2), II138 (2003).
36. A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, R. A. Young, *Pac. Symp. Biocomput.* **7**, 437 (2002).
37. Y. Tamada *et al.*, *Bioinformatics* **19** (suppl. 2), II227 (2003).
38. T. Ideker, O. Ozier, B. Schwikowski, A. Siegel, *Bioinformatics* **18** (suppl. 1), S233 (2002).
39. C. H. Yeang, T. Jaakkola, *Proceedings of the 7th Annual International Conference on Computational Molecular Biology*, M. Vingron *et al.*, Eds. (ACM Press, New York, 2003), pp. 312–321.
40. H. H. McAdams, A. Arkin, *Proc. Natl. Acad. Sci. U.S.A.* **94**, 814 (1997).
41. W. Vance, A. Arkin, J. Ross, *Proc. Natl. Acad. Sci. U.S.A.* **99**, 5816 (2002).
42. M. Ashburner *et al.*, *Nature Genet.* **25**, 25 (2000).

43. I thank L. Garwin, R. Nelken, D. Pe'er, A. Regev, and I. Wapinski for useful comments on earlier versions of this manuscript, and N. Kaminski, I. Nachman, D. Pe'er, A. Regev, E. Segal, and in particular D. Koller for many insightful discussions. Supported by the Bauer Center for Genomics Research, Harvard University, and a grant from the Israel Science Foundation.

Supporting Online Material

www.sciencemag.org/cgi/content/full/303/5659/799/DC1

Glossary