Bayesian Network Analysis of Signaling Networks: A Primer

Dana Pe'er

(Published 26 April 2005)

High-throughput proteomic data can be used to reveal the connectivity of signaling networks and the influences between signaling molecules. We present a primer on the use of Bayesian networks for this task. Bayesian networks have been successfully used to derive causal influences among biological signaling molecules (for example, in the analysis of intracellular multicolor flow cytometry). We discuss ways to automatically derive a Bayesian network model from proteomic data and to interpret the resulting model.

With the advent of high-throughput proteomic technologies, molecular signaling biology is experiencing an explosion of new experimental results. For example, intracellular multicolor flow cytometry (*1*, *2*) allows for quantitative, simultaneous observation of multiple signaling molecules in many thousands of individual cells. A major challenge is to reveal a coherent systems-level view of the signaling network from such data.

We proffer Bayesian networks (*3*) as suitable models for signaling pathways. We believe that it is essential for such pathway models to be of a probabilistic nature to accommodate the noise inherent in biologically derived data. Additionally, Bayesian networks are relatively robust to the existence of unobserved variables and can explicitly handle the uncertainty in such unobserved events (for example, current proteomic technology simultaneously measures only 12 molecules in individual cells, although there are many more molecules involved in a typical signaling response). Bayesian networks have been used for automatic reconstruction of causal signaling network models from data derived from individual primary human immune cells (*1*). The purpose of this primer is to provide a better mathematical understanding of Bayesian networks and how they can be used to derive causal influences between biological signaling molecules.

Bayesian networks provide a compact graphical representation of the joint probability distribution over the random variables $\mathbf{X} = X_1, \ldots, X_n$ (each such random variable represents the protein expression or activity level of a signaling molecule). Even for binary-valued variables (on or off), the joint distribution requires specification of the probabilities for the 2^n different assignments to $X_1, \ldots, X_n$. The key property of Bayesian networks is that they use simplifying structure in the joint distribution (by explicit encoding of conditional independencies; see below) to represent such high-dimensional data in a compact manner. Furthermore, the structural aspects of the joint distribution (modeled as a graph) might correspond to the graph structure of the signaling network itself.

Department of Genetics, Harvard Medical School, Boston, MA 02115, USA. E-mail: dpeer@genetics.med.harvard.edu

Consider a finite set $\mathbf{X} = X_1, \ldots, X_n$ of random variables, where each variable X_i may take on a value x_i from the domain Val(X_i). We use italic capital letters such as X, Y, Z for variable names; specific values taken by these variables are denoted x, y, z. Sets of variables are denoted by boldface capital letters $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$; assignments of values to the variables in these sets are denoted $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$.

At the core of Bayesian networks is the notion of conditional independence. This concept can be explained with an example from classical genetics. Assume that we are studying a certain mutation that appears in the Springfield population at a frequency of 0.001. If we sample a random resident of the city, the probability (Pr) that Bart has the mutation is 0.001. Now, assume that we know that Bart's Grandpa has the mutation. In this case, Pr(Bart has mutation given Grandpa has mutation) = 0.25. Grandpa's genotype was informative toward Bart's genotype; the two genotypes are clearly dependent. Next we learn that Homer (Bart's father) has the mutation too. In this case, Grandpa's genotype is irrelevant, so Pr(Bart has mutation given Homer and Grandpa have mutation) = Pr(Bart has mutation given Homer has mutation but Grandpa does not) = 0.5. Likewise, if Homer does not have the mutation, the probability that Bart has it is 0.001 whether Grandpa has the mutation or not. Conditioned on Homer's genotype, Grandpa's genotype does not affect the probability of Bart's genotype. Thus, we say that Bart's genotype is conditionall*y* independent of Grandpa's genotype, given Homer's genotype.

Definition 1: X is conditionally independent of Z given $\mathbf{Y}$ if the probability distribution of X conditioned on both $\mathbf{Y}$ and Z is the same as the probability distribution of X conditioned only on $\mathbf{Y}$:

$$P(X|\mathbf{Y},Z) = P(X|\mathbf{Y}) \qquad \text{(Eq. 1)}$$

We represent this statement as $(X \perp Z|\mathbf{Y})$.

Bayesian networks encode these conditional independencies with a graph structure.

Definition 2: A graph $G = (\mathbf{X},E)$ consists of a set of nodes $\mathbf{X}$, depicted as dots, and a set of edges E that connect the nodes, drawn as lines between pairs of nodes. Each edge X–Y represents a pair of nodes from $\mathbf{X}$. In a directed graph, each edge is ordered and $X \rightarrow Y$ denotes an edge from X into Y.

Model Semantics

A Bayesian network is a structured directed graph representation of relationships between variables. The nodes represent the random variables in our domain, and the edges represent the influence of one variable on another.

Definition 3: A Bayesian network (*3*) is a representation of a joint probability distribution consisting of two components. The first component, G, is a directed acyclic graph (DAG) whose nodes correspond to the random variables $X_1, \ldots, X_n$. Let Pa_i denote the parents of X_i in G (all nodes coming into X_i). The second component, θ, describes a conditional probability distribution $P(X_i|Pa_i)$ for each variable X_i in $\mathbf{X}$.

An important property of the graph G is that it represents conditional independencies between variables. In the genetics example above, Bart is independent of his ancestors conditioned on his parents. The Markov assumptions generalize this concept to any directed graph.

Definition 4: In a Bayesian network, the graph G encodes the Markov assumptions: Each variable X_i is independent of its nondescendants, given its parents in G.

As an example, a Bayesian network can represent the relations among five different proteins (Fig. 1). Assume that A is a kinase that phosphorylates protein B. If A activates B, we expect that in most cases when A is active, so is B. The Bayesian network indicates this dependency in activity levels by drawing a directed edge from A into B. The protein activities of A and B are statistically dependent; thus, knowing the value of A provides information that can help predict the value of B.

In addition, B is a kinase that phosphorylates C; thus, the network model has an edge from B into C. On the basis of pairwise correlations alone, we expect the activity of C to be correlated not only with its direct regulator (B), but also by its indirect regulator (A). In our example, if we know the value of B, A does not provide additional information that can improve our predictions for C, and so we say ‘‘The effect of A on C is mediated through B,’’ that is, A and C are conditionally independent given B. Such a relation can be inferred from interventional data; for example, A and C are correlated under normal conditions, but when B's activity is inhibited, this correlation disappears.

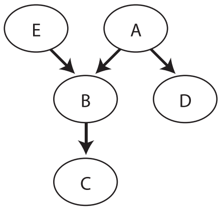

Fig. 1. An example of a simple Bayesian network structure. This network structure implies several conditional independence statements: $(A \perp E)$, $(B \perp D|A,E)$, $(C \perp A,D,E|B)$, $(D \perp B,C,E|A)$, and $(E \perp A,D)$. The joint distribution has the product form $P(A,B,C,D,E) = P(A)P(E)P(B|A,E)P(C|B)P(D|A)$.

Furthermore, kinase A also activates kinase D, creating a correlation between the activities of B and D. If B is active, we can reason that its activity might be a result of A's activity, and therefore D is more likely to be active as well. This is another example of conditional independence encoded in the graph structure: The activities of B and D are conditionally independent given their common regulator A. Such a relation can be inferred from the data if, for example, D is activated by another (unmeasured) kinase. In this case, A would be a more reliable predictor for B's activity than would D.

Finally, kinase E inhibits kinase B, providing B with two proteins that control its activity. A and E are B's parents in the Bayesian network. This leads into the second component of the Bayesian network, θ: Each node has a conditional probability θ that describes the probability of its levels conditioned on the levels of its parents.

The two components G and θ specify a unique distribution on $X_1, \ldots, X_n$. The chain rule of probabilities claims that any joint distribution can be expressed as a product of conditional probabilities, so that each variable X_i is conditioned on all the variables that precede it, $X_1, \ldots, X_{i-1}$:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \qquad \text{(Eq. 2)}$$

With the use of the conditional independencies derived from the Markov assumptions of Definition 4, the product form can be further simplified so that each variable X_i is only conditioned on its parents Pa_i:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Pa_i) \qquad \text{(Eq. 3)}$$

This is called the chain rule for Bayesian networks. This product economizes on the number of parameters, thus making the Bayesian network representation of a joint probability compact. As an example, consider the joint probability distribution $P(A,E,B,C,D)$ represented in Fig. 1. By the chain rule of probability, without any independence assumptions,

$$P(A,E,B,C,D) = P(A)P(E|A)P(B|A,E)P(C|A,E,B)\,P(D|A,E,B,C) \qquad \text{(Eq. 4)}$$

Assuming that all variables are binary, this representation requires $1 + 2 + 4 + 8 + 16 = 31$ parameters. Taking the conditional independencies into account,

$$P(A,E,B,C,D) = P(A)P(E)P(B|A,E)P(C|B)P(D|A) \qquad \text{(Eq. 5)}$$

which only requires $1 + 1 + 4 + 2 + 2 = 10$ parameters. More generally, given n binary variables and G whose indegree (that is, maximal number of parents) is bounded by k, then, instead of representing the joint distribution with $2^n - 1$ independent parameters, we can represent it with at most $2^k n$ independent parameters. This reduction in parameters is critical when estimating a model from empirical data. Robust estimation of a model with many parameters requires many more data points than does estimation of a model comprising fewer parameters. Fortunately, flow cytometry can measure protein expression/activity levels in thousands of individual cells, providing enough data for the estimation of Bayesian network models.

A graph G specifies a product form as in Eq. 3. To fully specify a joint distribution, we need to specify the conditional probability distributions $P(X_i|Pa_i)$ for each variable X_i. θ represents the parameters that specify these distributions. $P(X_i|Pa_i)$ can be viewed as a probabilistic function of X_i whose inputs are X_i's parents in G. Any distribution P satisfying the conditional independencies in Definition 4 can be encoded as a Bayesian network with structure G and associated conditional probability distributions.

In this primer, we focus on discrete variables and describe the conditional probability distributions used by Sachs *et al.* (*1*). In the more general case, these conditional distributions can be almost any computable representation. For instance, many continuous conditional probability distributions have been used with Bayesian networks (*4*–*6*).

We use a conditional probability table that specifies a probability distribution over X_i for each possible combination of value assignments to its parents Pa_i. Each joint value assignment pa_i to Pa_i corresponds to a row in the table. This row specifies the probability vector for X_i conditioned on $Pa_i = pa_i$. For example, if Pa_i consists of k binary valued variables, the table will specify 2^k distributions. This representation can describe any discrete conditional distribution. But this flexibility comes at a price: The number of free parameters is exponential in the number of parents.

As an example, assume that A and E each weakly activate protein B, and together they strongly activate protein B. Table 1 is a conditional probability table that represents such a relation.

a	e	P(b=0)	P(b=1)
0	0	0.96	0.04
0	1	0.61	0.39
1	0	0.68	0.32
1	1	0.07	0.93

Table 1. Example of a conditional probability table.

The Graph Structure: Independence, Dependence, and Causality

We now discuss the properties of the Bayesian network structure G. We describe the relationship between the graph structure and the conditional independencies it implies.

d-separation. If we assume that molecular interactions can be modeled by probabilistic dependencies, then we can use independence queries on the distribution $P(X_1, \ldots, X_n)$ to infer these interactions. The Bayesian network structure G greatly facilitates efficient multivariate independence queries. For instance, the structure $X \rightarrow Y \rightarrow Z$ implies that although X and Z are dependent, the variable Y renders them conditionally independent, $(X \perp Z | Y)$. The variable Y dismisses the dependence between X and Z, indicating that the interaction between X and Z is indirect. Thus, a Bayesian network can be used to distinguish between direct and indirect relationships. For example, consider the Raf→Mek→Erk cascade of protein kinases (*1*). The activity of Raf is informative toward the activity of Erk, but Mek activity renders the Raf and Erk activities independent. We infer that Raf influence on Erk proceeds though the intermediate Mek.

Although this independence statement was easy to infer from the simple substructure ($X \rightarrow Y \rightarrow Z$), it is natural to ask about queries that relate to variables that are further apart in G. It is possible to automatically derive such conditional independence relations between variables from the graph structure itself. Before continuing, we present a graph substructure that plays a key role both for the notion of d-separation and for the notion of equivalence described in this section.

Definition 5: A v-structure (*3*) is an induced subgraph (a subset of nodes and all the edges between these nodes in G) of the form $X \rightarrow Y \leftarrow Z$ so that no edge exists between X and Z.

The v-structure implies an interesting set of dependencies. In the previous cases, X and Z were dependent only when Y was unobserved; in a v-structure, given the value of Y, two possibly independent variables become dependent. Following is a classic example of such a dependency from genetics: Consider a random variable Y representing the existence of a rare mutation in some child, where $Y = 0$ indicates that Y does not have the mutation and $Y = 1$ indicates that Y has the mutation. Let X and Z represent the existence of the same mutation in each of that child's two biological parents. The genotypes of each of the parents X and Z are independent of one another. But if we know that $Y = 1$ (that is, the child has the rare mutation), this means that one of the two parents must have the mutation. Now, if we are also given that $X = 0$, we can infer that $P(Z = 1|Y = 1, X = 0) = 1$, and given $X = 1$ we can infer that $P(Z = 1|Y = 1, X = 1)$ is very low. Therefore, given the value of Y, the independent variables X and Z become dependent.

Intuitively, one can view dependence as a property that can ''flow'' between the nodes representing X and Z through paths that connect them in G. For instance, in $X \rightarrow Y \rightarrow Z$, dependence can ''flow'' from X to Z through Y, unless Y ''blocks'' this flow. Because $(X \perp Z|Y)$, the path is ''blocked'' only when Y is given. In an opposite case, $X \rightarrow Y \leftarrow Z$, dependence can ''flow'' from X to Z only if Y is given. To generalize these notions to longer paths, we say that the graph has a trail from X_1 to X_n, denoted X_1–...–X_n, if for every $i = 1 \ldots n - 1$, G contains either $X_i \rightarrow X_{i+1}$ or $X_i \leftarrow X_{i+1}$.

Definition 6: Let G be a Bayesian network structure and X_1–...–X_n be a trail in G. Let $\mathbf{E}$ be a subset of nodes from $\mathbf{X}$. There is an active trail between X_1 and X_n given evidence $\mathbf{E}$ if:

- Whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in $\mathbf{E}$.
- No other node along the trail is in $\mathbf{E}$.

Intuitively, this means that the dependence can ''flow'' through every triplet X_{i-1}–X_i–X_{i+1}.

Definition 7: Let $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ be three sets of nodes in G. We say that $\mathbf{X}$ and $\mathbf{Z}$ are d-separated given evidence $\mathbf{Y}$, denoted d-sep$_G$ ($\mathbf{X}$;$\mathbf{Z}$|$\mathbf{Y}$), if there is no active trail between any node X in $\mathbf{X}$ and Z in $\mathbf{Z}$ given evidence $\mathbf{Y}$ (*7*).

d-sep$_G$ is a property of the graph structure G that corresponds to the notion of conditional independence in the corresponding probability distribution P. Ind(G) is defined as the set of independence statements (of the form ''X is independent of Z given $\mathbf{Y}$'') that are implied by G. Using this formulation of d-separation, we can use an efficient graph algorithm whose running time scales linearly with the number of nodes in G (*3*) to check whether any such conditional independence statement holds.

Equivalence classes. More than one graph can imply exactly the same set of independencies. For example, consider the graphs $X \rightarrow Y$ and $X \leftarrow Y$; both imply the same set of independencies (that is, X and Y are dependent).

Definition 8: Two graphs G_1 and G_2 are equivalent if Ind(G_1) = Ind(G_2). That is, both graphs are alternative ways of describing the same set of independencies.

This notion of equivalence is crucial, because when we examine observations from a distribution, we cannot distinguish between equivalent graphs. Following the example above, given a statistical dependence between the activities of proteins X and Y, we cannot determine whether X activates Y or whether Y activates X. We can characterize equivalence classes of graphs with the use of a simple representation (*8*). Equivalent graphs have the same underlying undirected graph but might disagree on the direction of some of the arcs.

Theorem 1: Two Bayesian network structures are equivalent if and only if they have the same underlying undirected graph (termed ''skeleton'') and the same v-structures (*8*).

For example, the skeleton X–Y–Z can be partitioned into two equivalence classes, one containing three graphs representing $[(X \perp Z|Y), \neg(X \perp Z|\varnothing)]$ and the other containing the v-structure representing $[\neg(X \perp Z|Y), (X \perp Z|\varnothing)]$, where ¬ denotes the logical operator NOT (Fig. 2).

Moreover, an equivalence class of network structures can be uniquely represented by a partially directed acyclic graph (PDAG) P, where a directed edge $X \rightarrow Y$ denotes that all members of the equivalence class contain the directed edge $X \rightarrow Y$; an undirected edge X–Y denotes that some members of the class contain the directed edge $X \rightarrow Y$, whereas others contain the directed edge $X \leftarrow Y$. (In the final section, we discuss the relation between directed edges in the PDAG and causality.)

Assume that we are given some Bayesian network structure G and wish to derive the PDAG P representing G's equivalence class. Because Theorem 1 states that all equivalent graphs must agree on their v-structures, it is obvious that we need to orient any edge that participates in a v-structure. What is less obvious is that there are other edges in the graph that need to be oriented. These are the edges that form new v-structures when reversed. They can be derived through one of the following three propagation rules for compelled edges in P (Fig. 3):

- Consider the subgraph $X \rightarrow Y$–Z, where no edge exists between X and Z. Each edge direction between Y and Z defines a different equivalence class. The edge $Y \leftarrow Z$ forms a v-structure, whereas $Y \rightarrow Z$ does not. Therefore, the edge in the corresponding PDAG P is compelled to be directed as $Y \rightarrow Z$.
- Consider the subgraph $X \rightarrow Y$, $Y \rightarrow Z$, and X–Z. If we direct the edge as $Z \rightarrow X$, a cycle is formed. Therefore, to ensure acyclicity, the edge is compelled to be directed as $X \rightarrow Z$.
- Consider the subgraph X–Y, X–W, X–Z, $Y \rightarrow Z$, and $W \rightarrow Z$. The edge X–Z is compelled to be directed as $X \rightarrow Z$. Assume that the edge is directed as $Z \rightarrow X$. Then, to avoid acyclicity, the edges $X \leftarrow Y$ and $X \leftarrow W$ are compelled, thus forming a new v-structure.

Given a DAG G, the PDAG representation of its equivalence class can be constructed as follows. We begin from the underlying skeleton of G and orient all edges that participate in a v-structure. Then we continue applying the propagation rules of Fig. 3 until no more subgraphs corresponding to one of the rules exist.

Proposition 1: If we apply the procedure described above to G, the resulting PDAG represents the equivalence class of G.

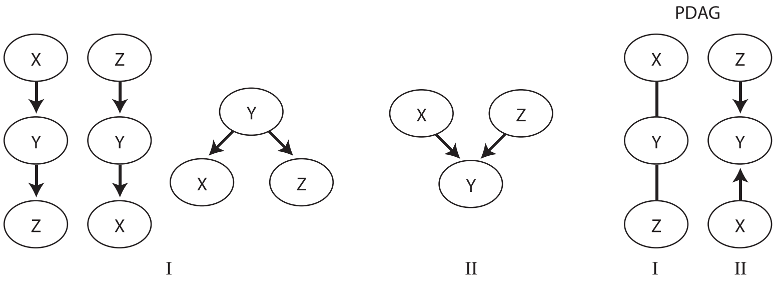

Fig. 2. The skeleton X–Y–Z is partitioned into two equivalence classes: class I, representing $[(X \perp Z|Y), \neg(X \perp Z|\varnothing)]$, and class II, the v-structure representing $[\neg(X \perp Z|Y), (X \perp Z|\varnothing)]$. The right panel illustrates the corresponding PDAGs.

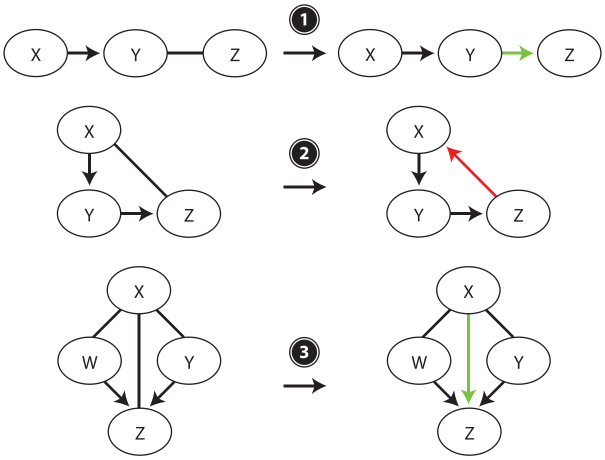

Fig. 3. Propagation rules for compelled edges: After all v-structures are oriented, a representation of the equivalence class is constructed by iteratively applying these rules to the PDAG.

An Algorithm for Inferring Bayesian Networks from Experimental Data

Our goal is to automatically infer the structure of the molecular signaling network from a large data set of phosphorylated protein activity levels. For example, flow cytometry can measure multiple protein expression levels simultaneously in many thousands of individual cells. We assume that influences between molecules in the signaling network enforce statistical dependencies between protein expression/activity levels and thus also impose a distribution P^* underlying the measurements. Recall that each Bayesian network represents a unique distribution (as defined by Eq. 3) that assigns a probability to each joint value assignment to variables (measured levels of phosphorylated proteins). The process of "learning" a Bayesian network is finding a Bayesian network that gives the samples in our data a high probability.

Thus, given flow cytometry data $D = \{X[1], \ldots, X[M]\}$, we treat each individual cell as a sample and assume that these are independently drawn from some unknown generating Bayesian network G^*, with an underlying distribution P^*. Our goal is to recover G^*. Because D is only a noisy sample derived from P^*, we cannot detect with complete reliability which dependencies are present in the underlying distribution. Instead, we search for a relatively simple model, $B = (G,\theta)$ (with few edges), that is likely to have generated the data—that is, a model whose underlying distribution is close to the empirical distribution of the data D.

More precisely, we search for an equivalence class of networks that best matches D. Recall that all structures in an equivalence class represent the same dependencies and are equally close to the empirical distribution of D. We cannot distinguish between them solely on the basis of D. Thus, the best we can hope for is to recover a structure that is equivalent to G^*.

The theory of learning networks from data has been examined extensively over the past decade. In this primer, we take a score-based approach to learning. We define a hypothesis space of potential network models, introduce a statistically motivated scoring function that evaluates each network with respect to the training data, and search for the highest scoring network.

To do so, we first assume that the graph structure G is given and describe an appropriate score for the conditional probability distribution parameters θ and a closed-form solution for the highest scoring parameters. Then we describe an appropriate score for the graph structure itself. Finally, we describe a greedy algorithmic approach that finds a high-scoring network structure. (The reader unfamiliar with probability theory may skip ahead to the section on model averaging.)

Maximum likelihood estimation of parameters. In this section, we assume that the structure of the graph G is known. Although this is not a reasonable assumption for our domain, the theory of parameter estimation is a basic building block for the structure learning to come. In the Bayesian network learning task, we implicitly assume that there is some Bayesian network B^* that generated the data D, and our goal is to use these data to try to reconstruct B^*. A good Bayesian network B is one that is likely to have generated D. If we assume that G is already known, our task then is to find the conditional probabilities θ that maximize the likelihood that D was generated by the Bayesian network $B^* = (G,\theta)$.

Definition 9: We define a likelihood function, $L(\theta{:}D)$, that measures the likelihood that the parameters θ generated the data D. Because the samples are independent, the likelihood decomposes into a product of probabilities, one for each sample:

$$L(\theta{:}D) = \prod_{m=1}^{M} P(X[m] \| \theta) \qquad \text{(Eq. 6)}$$

In maximum likelihood estimation, given a data set D, we wish to choose parameters $\hat{\theta}$ that maximize the likelihood of the data (as above):

$$\hat{\theta} = \text{argmax } L(\theta{:}D) \qquad \text{(Eq. 7)}$$

Optimizing Equation 7 could potentially be a computationally hard problem (exponential in the number of parameters) because of the high dimensionality of θ and the large number of parameters that need to be concurrently optimized. One of the big advantages of the Bayesian network representation is that this likelihood separates into local likelihood functions, one for each variable. This simplifies the calculation of the likelihood; more important, it renders finding its optimal parameters tractable. Each local likelihood can be optimized in an independent manner, thus decomposing a complex global problem into smaller subproblems:

$$\begin{aligned} L(\theta{:}D) &= \prod_{m=1}^{M} P(X[m]) \\ &= \prod_{m=1}^{M} \prod_{i=1}^{n} P(X_i[m] \| Pa_i[m],\theta) \\ &= \prod_{i=1}^{n} \left[\prod_{m=1}^{M} P(X_i[m] \| Pa_i[m],\theta) \right] \\ &= \prod_{i=1}^{n} L_i(\theta_{X_i|Pa_i}{:}D) \end{aligned} \qquad \text{(Eq. 8)}$$

where

$$L_i(\theta_{X_i|Pa_i}{:}D) = \prod_{m=1}^{M} P(X_i[m] \| Pa_i,\theta) \qquad \text{(Eq. 9)}$$

is the local likelihood function for X_i.

If we have a variable X with its parents Pa_X, the conditional probabilities for each joint assignment of values to X and Pa_X are associated with the Bayesian network. In the case of conditional probability tables, for each combination of value assignments $x \in \text{Val}(X)$ and $\mathbf{u} \in \text{Val}(Pa_X)$, a parameter $\theta_{x|\mathbf{u}}$ represents $P(X = x|Pa_X = \mathbf{u})$. In conditional probability tables, these parameters are independent for each $\mathbf{u} \in \text{Val}(Pa_X)$; therefore, the local likelihood can be further decomposed into a yet simpler form. The idea behind the decomposition is to group together all the instances in which $X = x$ and $Pa_X = \mathbf{u}$. We define $M[x,\mathbf{u}]$ as the number of instances in which $X = x$, $Pa_X = \mathbf{u}$, and $M[\mathbf{u}] = \sum_{x \in \text{Val}(X)} M[x,\mathbf{u}]$. Then, by rearranging the order of the product, we can write

$$L_i(\theta_{X|Pa_X}{:}D) = \prod_{\mathbf{u} \in \text{Val}(Pa_X)} \prod_{x \in \text{Val}(X)} \theta_{x|\mathbf{u}}^{M[x,\mathbf{u}]} \qquad \text{(Eq. 10)}$$

Proposition 2: The maximum likelihood estimate for a Bayesian network with multinomial conditional probability tables is given by

$$\hat{\theta}_{x|\mathbf{u}} = \frac{M[x,\mathbf{u}]}{M[\mathbf{u}]} \qquad \text{(Eq. 11)}$$

We call the counts $M[x,\mathbf{u}]$ and $M[\mathbf{u}]$ sufficient statistics. Given these counts, the actual data instances $X[1], \ldots, X[M]$ themselves are no longer needed. The sufficient statistics extract from the data all the relevant information needed to calculate the likelihood. Note that the optimal parameters are based on the empirical counts observed in our data. Thus, optimizing the likelihood is equivalent to finding the best approximation for the empirical distribution constrained to the independencies of G.

Bayesian approach to parameter estimation. Although the maximum likelihood estimation approach would seem to be a suitable way to measure the fit of a Bayesian network to the data, it has a number of disadvantages. Its main drawback is that it tends to overfit the model to the particular data instance at hand. This problem is especially critical when studying signaling in rare cellular subsets, for which we have a relatively small number of samples. We illustrate this problem with an example from the medical domain: Assume that we are performing a study on the effect of smoking on lung cancer. Our sample contains 30 nonsmokers, none of whom contracted lung cancer. Maximum likelihood estimation would construct a model that postulates P(lung cancer = YES|smoker = NO) = 0. However, we think that the correct answer is that there is a small chance for a nonsmoker to develop lung cancer, but our small sample did not contain such a case.

We therefore turn to the Bayesian approach, which formulates this concept of prior belief in a principled manner. The idea is that in addition to the observed data D, we have some initial distribution $P(\theta)$, termed "the prior," which encodes our beliefs regarding the domain prior to our observations.

When we have little prior knowledge of our domain, this distribution is often flat and mostly ensures that every event has some nonzero probability. On the other hand, if we do have specific information about our domain, this distribution can be more peaked over certain values.

After we observe some data D, we update the distribution $P(\theta)$ to reflect the combination of both our prior belief and our observations. This updated distribution, $P(\theta|D)$, is called the posterior distribution:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \qquad \text{(Eq. 12)}$$

The term $P(D)$, termed ‘‘the marginal likelihood,’’ averages the probability of the data over all possible parameter assignments. Because it is a normalizing constant, which is independent of θ, we ignore it in our score calculations.

In this primer, we use the Dirichlet priors (*9*) for multinomial distributions. A Dirichlet prior is specified by a set of hyperparameters $\alpha_{x^1|u}, \ldots, \alpha_{x^K|u}$, with one such hyperparameter corresponding to each $x^j \in \mathrm{Val}(X)$. The Dirichlet distribution is specified by

$$P(\theta) = \mathrm{Dirichlet}(\alpha_{x^1|u}, \ldots, \alpha_{x^K|u}) \sim \prod_j \theta_{x^j|\mathbf{u}}^{\alpha_{x^j|\mathbf{u}}-1} \qquad \text{(Eq. 13)}$$

Dirichlet priors have a number of desirable properties; they satisfy global parameter independence and local parameter independence. This means that the prior decomposes into a product of independent terms, in a manner similar to decomposition in maximum likelihood estimation.

Definition 10: A parameter prior $P(\theta)$ is said to satisfy global parameter independence if it decomposes into the following form:

$$P(\theta) = \prod_{i=1}^{n} P(\theta_{X_i|Pa_i}) \qquad \text{(Eq. 14)}$$

Definition 11: Let X be a variable with parents Pa_X. Then the prior $P(\theta_{X|Pa_X})$ has local parameter independence if

$$P(\theta_{X|Pa_X}) = \prod_{\mathbf{u}\in \mathrm{Val}(Pa_X)} P(\theta_{X|\mathbf{u}}) \qquad \text{(Eq. 15)}$$

The prior $P(\theta)$ satisfies parameter independence if it satisfies both global and local parameter independence.

In addition, Dirichlet priors are conjugate priors; that is, the posterior distribution has the same functional form as the prior. This property provides an intuitive interpretation for the hyperparameters. One can view $\alpha_{x|\mathbf{u}}$ as imaginary counts; that is, before we observed D, on $\alpha_{x|\mathbf{u}}$ occasions we ‘‘observed’’ $X = x$ and $Pa_X = \mathbf{u}$.

Proposition 3: If $P(\theta)$ is Dirichlet($\alpha_{x^1|u}, \ldots, \alpha_{x^K|u}$), then the posterior $P(\theta|D)$ is Dirichlet($\alpha_{x^1|u} + M[x^1,u], \ldots, \alpha_{x^K|u} + M[x^K,u]$), where $M[x,\mathbf{u}]$ are the sufficient statistics derived from D.

We say that $\alpha^* = \sum_j \alpha_{x^j|\mathbf{u}}$ is our effective sample size, meaning that we assume that our prior is based on α^* observations. This reflects how strongly we believe in our prior. As we accumulate more samples in D, the effect of the prior on the posterior grows weaker.

Even when our prior is flat, the full effect of the Bayesian approach comes into play when predicting the probability of future samples. In the Bayesian approach, the probability of a future observation is calculated not on the basis of only one set of parameters, but using the expectation over the entire distribution of parameters. Thus, the probability of a new sample $X[M+1]$ is

$$P(X[M+1]|D) = \int P(X[M+1]|D,\theta)P(\theta|D)P(D)dD \quad \text{(Eq. 16)}$$

When we use conditional probability tables and Dirichlet priors, this integral has a closed-form solution:

$$\begin{aligned} P(X_i[M+1]|D) &= P(X_i[M+1] = x^i|Pa_i[M+1] = \mathbf{u},D) \\ &= \frac{\alpha_{x^i|\mathbf{u}} + M[x^i,\mathbf{u}]}{\sum_j (\alpha_{x^j|\mathbf{u}} + M[x^j,\mathbf{u}])} \end{aligned} \qquad \text{(Eq. 17)}$$

Structure learning: A score-based approach. Previously, we showed how one can learn the Bayesian network parameters given a known structure G, but in a real scenario we do not know G. Our goal is to understand the structural relationships between the variables in our domain. For instance, we would like to be able to distinguish between direct and indirect interactions between molecules. Therefore, it is exactly this graph structure that we wish to reconstruct from the observed data. We can then use this reconstructed graph structure to answer queries regarding the interactions between the proteins and other structural properties. On the basis of observational data alone, it is not possible to distinguish between equivalent structures. Thus, at best our reconstruction procedure can reconstruct an equivalence class of networks.

We take a score-based approach to this problem. We define a model space of candidate models that we are willing to consider, along with a scoring function that measures how well each model fits the observed data. Then we use an optimization algorithm that searches for a high-scoring model.

Our scoring function is based on the same Bayesian principles described above, following a basic principle: Whenever we have uncertainty about something, we place a probability distribution over it. We therefore define a structure prior $P(G)$ over the different graph structures and a parameter prior $P(\theta|G)$ over the parameters once the graph is given. The particular choice of the priors $P(G)$ and $P(\theta|G)$ determines the exact Bayesian score. Our score evaluates the posterior probability of the graph given the data:

$$\mathrm{Score}_B(G{:}D) = \log P(D|G) + \log P(G) \qquad \text{(Eq. 18)}$$

where $P(D|G)$ takes into consideration our uncertainty over the parameters by averaging the probability of the data over all possible parameter assignments to G,

$$P(D|G) = \int P(D|G,\theta)P(\theta|G)\,d\theta \qquad \text{(Eq. 19)}$$

The Bayesian score inherently handles the problem of overfitting a small sample to a complex model. Because of the integration over all possible parameters, structures with many parameters (that is, many parents for each variable) are penalized, unless the probability of the true parameters is very peaked (which happens when the sample size is large). Although the Bayesian score is biased to more simple structures, as more data accumulate, the score will support more complex structures (when the generating distribution is indeed complex). We explain this intuition with the following thought experiment: Assume that we are playing a gambling game on the value of X; each time we guess the correct value, we gain \$10. Assume that for \$1000 we can purchase a subscription to the value of Y and this value improves our ability to guess X, so that each time we guess, we expect to be correct 1 out of 5 times. For only 100 games it does not pay to purchase Y, but for a series of 700 games it would be advantageous to purchase Y. This is the ''game'' our scoring function is playing. In a biological context, if we observe that Y holds a small amount of information on X in only 100 samples, this might be a spurious dependency, but if this remains consistent over 1000 samples, then we begin to believe the relationship and add the edge to our model.

We next show how to choose good priors and demonstrate how these priors lead to desirable properties in our score. An important characteristic of the Bayesian score is that when we restrict ourselves to a certain class of factorized priors (*10*, *11*), the Bayesian score decomposes.

Definition 12: A parameter prior satisfies parameter modularity when for any two graphs G_1 and G_2, if $Pa_i^{G_1} = Pa_i^{G_2}$, then

$$P(\theta_{X_i|Pa_i{}^{G_1}}|G_1) = P(\theta_{X_i|Pa_i{}^{G_2}}|G_2) \qquad \text{(Eq. 20)}$$

This relationship means that the parameter prior depends only on the local structure of the graph.

Proposition 4: If the prior $P(\theta|G)$ satisfies global parameter independence and parameter modularity, then

$$P(D|G) = \prod_i \int_{\theta_{X_i|Pa_i}} \prod_m P(X_i[m]|Pa_i[m],\theta_{X_i|Pa_i})P(\theta_{X_i|Pa_i})d\theta_{X_i|Pa_i} \qquad \text{(Eq. 21)}$$

Therefore, we can decompose the score into the local contributions of each variable (denoted FamScore), where the contribution of every variable X_i to the total network score depends only on the sufficient statistics of X_i and its parents Pa_i:

$$\text{Score}_B(G{:}D) = \sum_i \text{FamScore}_B(X_i,Pa_i{:}D) \qquad \text{(Eq. 22)}$$

As we will see, this decomposition plays a crucial role in our ability to efficiently search for high-scoring network structures.

One of the big advantages of using Dirichlet priors is that the family score has a simple closed-form formula.

Definition 13: The gamma function $\Gamma(x)$ is defined to be an extension of the factorial function to real-number arguments. The gamma function is defined as the integral

$$\Gamma(x) = \int_0^\infty t^{x-1}\exp(-x)\,dt \qquad \text{(Eq. 23)}$$

If n is a natural number, then $\Gamma(n) = (n - 1)!$

Theorem 2: Let G be a network structure and $P(\theta|G)$ be a parameter prior satisfying parameter independence. Further assume conditional probability tables and Dirichlet priors with hyperparameters $\{\alpha_{x_i^j|u}\}$. Then

$$\text{FamScore}_B(X_i,Pa_i{:}D) = \log \prod_{\mathbf{u}\in \text{Val}(Pa_i)} \frac{\Gamma(\alpha_{x_i|\mathbf{u}})}{\Gamma(\alpha_{x_i|\mathbf{u}} + M[\mathbf{u}])} \prod_{x_i^j \in \text{Val}(X_i)} \frac{\Gamma(\alpha_{x_i^j|\mathbf{u}} + M[x_i^j,\mathbf{u}])}{\Gamma(\alpha_{x_i^j|\mathbf{u}})} \qquad \text{(Eq. 24)}$$

where Γ is the gamma function and

$$\alpha_{x_i|\mathbf{u}} = \sum_{j\in \text{Val}(X_i)} \alpha_{x_i^j|\mathbf{u}} \qquad \text{(Eq. 25)}$$

(*10*). A desired property is that the score reaches its optimum on the true generating structure; that is, given a sufficiently large number of samples, graph structures that exactly capture all dependencies in the distribution will receive a higher score than all other graphs (*12*). This means that given a sufficiently large number of instances, learning procedures can pinpoint the correct equivalence class of network structures.

Definition 14: Assume that our model is generated by some true model G^*. We say that our score is consistent if, as $M \to \infty$, the following properties hold with probability asymptotic to 1 (over possible choices of data set D):

- The structure G^* will maximize the score.
- All structures that are not equivalent to G^* will have a strictly lower score.

Theorem 3: The Bayesian score is consistent.

Recall that given D it is impossible to distinguish between two different networks in the same equivalence class. Thus, another desirable property in our score is that equivalent structures receive the same score. That is, if G_1 and G_2 are equivalent graphs, they are guaranteed to have the same score. Such a property is called structure equivalence. To achieve structure equivalence, we devise a set of hyperparameters so that our prior will not bias the score between equivalent structures. This is achieved with the use of a BDe prior (*13*). We define a probability distribution P' over X and an equivalent sample size M' for our set of imaginary samples. The hyperparameters are then defined to be

$$\alpha_{x_i|\mathbf{u}_i} = M'P'(x_i,\mathbf{u}_i) \qquad \text{(Eq. 26)}$$

Theorem 4: When the data are complete and Dirichlet BDe priors are used, the score is structure-equivalent (*10*).

Search algorithm for maximizing scores. Once the score is specified and the data are given, learning amounts to finding the structure G that maximizes the score. This problem is known to be computationally hard (exponential in the number of variables) to solve exactly (*14*); thus, we resort to a heuristic search. We define a search space, so that each state in this space is a network structure. We define a set of operators that transform one network structure into another. This defines a graph structure on the states: Neighboring states are those that are one operator away. We start with some initial structure (for

```
Input
        $D$ //  a data set
        $G_o$ // initial network structure
Output
        $G$ // final network structure
Greedy-structure-search
        $G_{best} = G_o$
        repeat // apply best possible operator to G in each iteration
                $G = G_{best}$
                foreach operator o // (each edge addition, deletion, or reversal on $G$)
                        $G^o = o(G)$ // apply to $G$
                        if $G^o$ is cyclic continue
                        if $\text{score}_{BDe}(G^o : D) > \text{score}_{BDe}(G_{best} : D)$
                                $G_{best} = G^o$
        until $G == G_{best}$ // no change in structure improves score
```

Fig. 4. Outline of greedy search algorithm.

example, the empty graph) and, using the operators, traverse this space searching for high-scoring structures.

A natural choice of neighboring structures is a set of structures that are identical to the base structure except for local modifications. We use the following operators that change one edge at each step:

- Add an edge
- Remove an edge
- Reverse an edge

Note that we only consider operations that result in allowed networks. Networks must be acyclic and must satisfy any other constraints we specify (for example, maximal indegree constraints).

A local search procedure can efficiently evaluate the gains made by adding, removing, or reversing a single edge. The decomposition of the score is crucial for the efficiency of this procedure. Decomposition allows us to reevaluate only those components of the score that involve the variables affected by our local step. For instance, if we add an edge to the variable X_i, we only need recalculate X_i's family score; the family scores for all other variables remain unchanged.

We use the following greedy hill-climbing algorithm for our search procedure: At each step, we evaluate all possible local moves and perform the change that results in the maximal gain, until we reach a local maximum. (A sketch for such an algorithm appears in Fig. 4.) Although this procedure does not necessarily find a global maximum, it does perform well in practice. A number of heuristics can be included to overcome some of the local maxima (for example, random restarts). Examples of other search methods that advance by single edge changes include beam-search, stochastic hill-climbing, and simulated annealing (*15*).

Any implementation of these search methods involves caching of computed counts to avoid unnecessary passes over the data. This cache also allows us to marginalize counts. Thus, if $M[X,Y]$ is in the cache, we can compute $M[X]$ by summing over values of Y. This is usually much faster than making a new pass over the data. One of the dominating factors in the computational cost is the number of passes actually made over the data.

Model Averaging

Given a flow cytometry data set, we use the learning algorithm described above to search for the Bayesian network G that best explains the data. A simple approach would be to accept G as a correct model of our domain. Then we could use G to infer relations between proteins (for example, kinase A directly influences kinase B). Such analysis would rely on the assumption that the network G correctly represents the interactions in the underlying domain, but how reasonable is this assumption? A sufficiently large number of samples (hundreds of thousands of cells) would (almost) provide that the network inferred is a good model of the data (*12*). However, given a smaller number of training instances (thousands), there may be many models that explain the data almost equally well.

Figure 5 shows five high-scoring networks in relation to data set D. The networks vary in structure, but their scores are almost the same. In our example, G_3 is the highest scoring network. In a simple approach, we would infer a direct interaction between the proteins A and C (because the edge A→C exists in G_3), but the edge is absent from the other high-scoring networks. Therefore, it is more likely that the dependence between A and C is a spurious artifact in D. Thus,

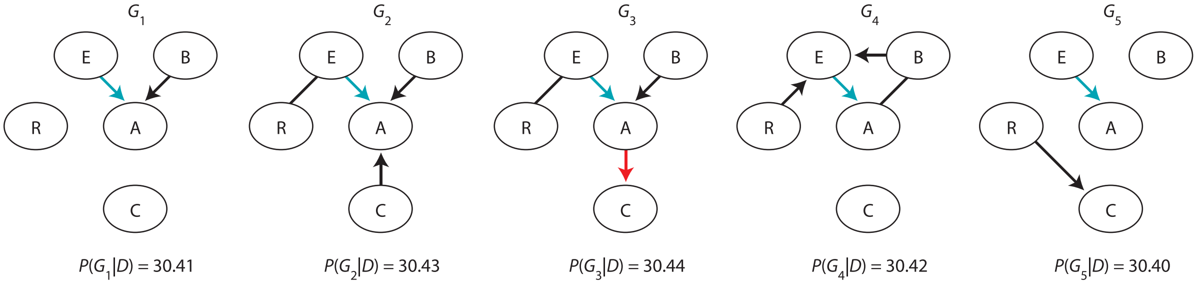

Fig. 5. An ensemble of high-scoring networks. Five different Bayesian networks are shown; the score of the network with respect to the data set D is listed below each network. G_3 is the highest scoring network, but the other networks are almost as likely. The edge A→C exists in G_3, but is absent from all other high-scoring networks. On the other hand, all five networks agree on the edge E→A.

we cannot depend on a single network to provide an accurate description of the relations in our biological domain.

Instead of querying a single structure, we search for common feature*s* that most of the high-scoring network structures agree on. For example, the ensemble of high-scoring networks in Fig. 5 all agree on the edge E→A. Therefore, it is likely to represent a real biological signal. We find such features by examining the posterior probability of the feature given the data. A network feature is a property such as "$X \rightarrow Y$ is in the network" or "$d - \text{sep}_G(X;Y|\mathbf{Z})$ is in the network." We associate the feature f with an indicator function, $f(G)$, that has the value 1 when G satisfies the feature and value 0 otherwise. The posterior probability of f is defined as

$$P[f(G)|D] = \sum_G f(G)P(G|D) \qquad \text{(Eq. 27)}$$

This probability reflects our confidence in f given D.

The straightforward way of calculating Eq. 27 is by enumerating all high-scoring networks. Unfortunately, the number of such networks can be exponential in the number of variables, so exact computation of the posterior probability is not feasible. Instead, we can estimate this posterior probability by sampling representative networks and then estimating the fraction that contain the feature of interest. Ideally, we would like to sample networks from $P(G|D)$ and use the sampled networks to estimate this quantity. The general solution to this problem is to build a Markov chain Monte Carlo sampling procedure (*16*). Unfortunately, this sampling procedure is also computationally costly.

Instead, we use an effective and relatively simple bootstrap method (*17*) as an approximation of the posterior probability. Our networks are generated using nonparametric bootstrap (*18*). We generate "perturbed" versions of the original data set and learn a Bayesian network structure from each of them. In this way, we collect many networks, all of which are fairly reasonable models of the data. These networks reflect the effect of small perturbations to the data on the learning process. We use the following procedure:

- For $i = 1, \ldots, m$, construct a data set D_i by sampling, with replacement, M instances from D. Then, apply the learning procedure on D_i to induce a network structure G_i.
- For each feature f of interest, calculate

$$\text{conf}(f) = \frac{1}{m}\sum_{i=1}^{m} f(G_i) \qquad \text{(Eq. 28)}$$

[See (*19*) for an evaluation of this bootstrap approach on simulated data.] These simulation experiments show that features induced with high confidence are rarely false positives, even in cases where the data sets are small relative to the size of the system being learned.

Interventions and Causality

Causality. Recall that a Bayesian network is a model of dependencies between multiple variables. However, we are also interested in modeling the mechanisms that generated these dependencies. Thus, we want to model the causality in the system of interest (for example, protein X activates protein Y). A causal network is a model of such causal processes. Having a causal interpretation facilitates predicting the effect of an intervention in the domain.

Although at first glance there would seem to be no direct connection between probability distributions and causality, causal interpretations for Bayesian networks have been proposed (*8*, *20*). A causal network is mathematically represented similarly to a Bayesian network, a DAG where each node represents a random variable along with a local probability model for each node. However, causal networks have a stricter interpretation on the meaning of edges: The parents of a variable are its immediate causes.

A causal network models not only the distribution of the observations, but also the effects of interventions. If X causes Y, then manipulating the value of X affects the value of Y. On the other hand, if Y causes X, then manipulating X will not affect Y. Thus, although $X \rightarrow Y$ and $X \leftarrow Y$ are equivalent Bayesian networks, they are not equivalent causal networks.

A causal network can be interpreted as a Bayesian network when we are willing to make the causal Markov assumption, which states that given the values of a variable's immediate causes, it is independent of its earlier causes. When the causal Markov assumption holds, the causal network satisfies the Markov independencies of the corresponding Bayesian network. But when can we derive a causal network from data? This issue has received a thorough treatment in the literature (*8*, *21*, *22*); we briefly review some relevant results [for a more detailed treatment of the topic, see (*20*, *23*)].

First, it is important to distinguish between an observation (a passive measurement of our domain; that is, a sample from $\mathbf{X}$) and an intervention [setting the values of some variables with the use of forces outside the causal model, such as chemical inhibition or small interfering RNA (siRNA)]. Interventions are an important tool for inferring causality, but occasionally some causal relations can be inferred from observations alone.

To learn causality, we require several assumptions. First, we require a modeling assumption: We assume that the (unknown) causal structure of the domain satisfies the causal Markov assumption. Thus, we assume that causal networks can provide a reasonable model of the domain. The second assumption is that there are no latent or hidden variables that affect several of the observable variables. Unfortunately, neither assumption holds in our domain. Thus, causal conclusions from our learning procedure must be treated with caution.

If we do make these two assumptions, then we essentially assume that one of the possible DAGs over the domain variables is the "true" causal network. However, as discussed above, observations alone do not permit us to distinguish between causal networks that specify the same independence properties (i.e., belong to the same equivalence class). Thus, at best we can hope to learn a description of the equivalence class that contains the true model. In other words, we will learn a PDAG description of this equivalence class.

Once we identify such a PDAG, we are still uncertain about the true causal structure in the domain. However, we can draw some causal conclusions. For example, if there is a directed path from X to Y in the PDAG, then X is a causal ancestor of Y in all the networks that could have generated this PDAG, including the "true" causal model. Thus, in this situation we can recover some of the causal directions.

Modeling interventions. To infer the causal direction of more edges, we need to apply external interventions to the data. These interventions can be genetic mutations, siRNA, small chemical interventions (inhibitors or activators), or any other intervention that directly influences one of the molecules observed in the data. The use of data containing external interventions contradicts a basic assumption made by our Bayesian network learning algorithm: that each data instance was sampled from the same underlying distribution. For instance, using a small chemical to inhibit X, we replace the original molecular control on X's activity by an external one. Thus, any consequent measurement (in which X's value is constantly set to 0) will behave differently from X's conditional distribution on its parents in observational data. Therefore, it is important to explicitly model this intervention into our learning algorithms.

Formally, we model an intervention, denoted $\mathrm{do}(X = x)$, as an ideal intervention (*20*) that deterministically sets the value of X to x. This intervention disables the natural causal mechanisms that affect X and replaces them with an external deterministic mechanism. In addition, we assume that the intervention only affects X's causal mechanism and leaves intact all other causal mechanisms in the model; that is, all other variables behave according to their respective conditional distribution. More formally, given a causal network G, an ideal intervention defines a new causal network $G_{\mathrm{do}(X=x)}$, identical to G except that all incoming edges into X are removed. In $G_{\mathrm{do}(X=x)}$, X becomes a root node associated with the probability distribution $\Pr(X = x) = 1$ (Fig. 6). Note that X's outgoing edges are not affected by such an intervention. When a number of different variables are manipulated in the same sample, we remove the edges incoming to each of these variables.

Such interventions can be used for causal inference. We illustrate this point with the following thought experiment: Consider the pair of networks $X \rightarrow Y$ and $Y \rightarrow X$. As Bayesian networks, the two are equivalent and cannot be distinguished on the basis of observational data alone. If we inhibit X [i.e., $\mathrm{do}(X = x)$], then, as causal models, we expect each of the two models to respond differently. If the causal model is $X \rightarrow Y$, then Y's causal mechanism remains intact.

Therefore, the same conditional distribution is measured in both the observed and inhibited samples: $P[Y|\mathrm{do}(X = x)] = P(Y|X = x)$. On the other hand, if $Y \rightarrow X$ is the causal model, the inhibition of X disables the causal mechanism responsible for the dependency between the two variables. When X is inhibited, the variables X and Y become independent: $P[Y|\mathrm{do}(X = x)] = P(Y)$. Therefore, we expect to measure a different conditional distribution in the observational and inhibited samples. Whereas we could not distinguish between the two models with the use of observations alone, we can differentiate between them with the use of interventional data.

The global nature of our reasoning allows us to reach a causal conclusion upstream of an intervention. In the above example, if we inhibit X and observe a different conditional distribution—$P[Y|\mathrm{do}(X = x)] \neq P(Y|X = x)$—we infer that the causal mechanism between X and Y has been disrupted. Because inhibition of X disrupts the mechanism of its direct incoming causes, we infer that Y causes X, or in our terminology, Y regulates X.

Scoring with interventions. The Bayesian scoring function described above assumes that all samples are drawn from the same network structure. We adapt that score to properly handle interventions (under the model of an ideal intervention). In such data sets, the underlying Bayesian network associated with each sample differs depending on the perturbation administered to the sample.

Similarly to (*24*), we make the following set of assumptions:

- Our samples are independent random samples from a causal network. This causal network represents both the probability distribution sampled and the causal relationships in the data. We note that this assumption does not hold in our domain, as this assumes acyclicity of the generating network.
- Each perturbation is an ideal intervention; that is, it disables the normal causal mechanism of the inhibited/activated protein and an independent causal mechanism deterministically sets its value. Furthermore, this intervention does not directly affect the causal mechanism of any other protein.
- The data are complete; there are no missing or hidden variables.
- Global and local parameter independence (Definitions 10 and 11).
- Parameter modularity (Definition 12).
- The prior distribution of all parameters is Dirichlet.

We define $M[m]$ as the set of interventions occurring in the mth sample. Let M be the collection of all sets of interventions

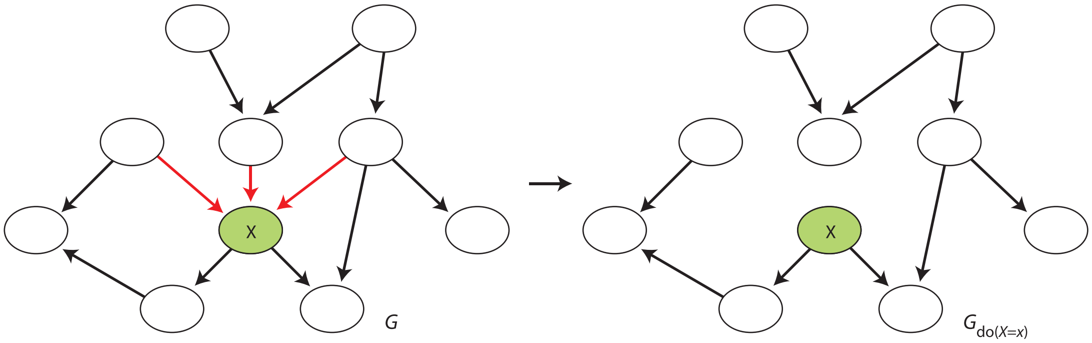

Fig. 6. Example of ideal intervention. Assume that the variable X (green oval) is inhibited in G. We construct the corresponding $G_{\mathrm{do}(X=x)}$ by removing all edges incoming from Pa_X (red edges).

occurring in D. Then, G^M is the set of all graphs derived from G based on M, that is,

$$G^M = \left\{G_{\mathrm{do}(M[m])}\right\}_{m=1}^{M} \qquad \text{(Eq. 29)}$$

One of the key properties of the Bayesian scoring function is that the score decomposes into a product of local entities, each depending only on X and Pa_X. Unfortunately, when D contains interventions, the instances are not associated with a single structure. The probability of each sample m is calculated using the appropriate graph, $G_{\mathrm{do}(X=x)}$. Therefore, at first glance, such a decomposition might seem problematic.

We briefly sketch how this problem can be overcome [see (*19*) for a full derivation of the appropriate Bayesian score under interventions]. The assumptions of parameter independence and parameter modularity allow us to calculate the contribution of each variable independently, without being affected by the choice of parent sets for other variables. Thus, the expression for each variable X_i decomposes as a separate integral that depends only on its parents in $G_{\mathrm{do}(X=x)}$ and the associated parameters. The key point to notice is that although G^M might contain many different graphs, if we focus our attention on a single variable X_i, we find only two possible sets of parents: Pa_i^G for samples where X_i is not intervened, and a root node in samples where it is intervened (Fig. 6). Furthermore, the parameters of X_i in the nonintervened samples are independent of the intervened samples.

When assuming Dirichlet priors, this leads to the same closed-form formula derived in Eq. 24. The only difference is that the counts $M[\mathbf{u}]$ and $M[x_i^j,\mathbf{u}]$ are tallied only over the samples $X_i \notin M[m]$:

$$\mathrm{FamScore}_B(X_i,Pa_i{:}D) = \log \prod_{\mathbf{u}\in \mathrm{Val}(Pa_i)} \frac{\Gamma(\alpha_{x_i|\mathbf{u}})}{\Gamma(\alpha_{x_i|\mathbf{u}}+M[\mathbf{u}])} \prod_{x_i^j\in \mathrm{Val}(X_i)} \frac{\Gamma(\alpha_{x_i^j|\mathbf{u}}+M[x_i^j,\mathbf{u}])}{\Gamma(\alpha_{x_i^j|\mathbf{u}})} \qquad \text{(Eq. 30)}$$

Inferring causality with interventions. The score of Eq. 30 is not structure-equivalent: The score of two equivalent graphs G and G' is no longer necessarily the same. This should not come as a surprise, because the score was derived with the intent of using interventional data to differentiate between equivalent graphs. We use the equivalent graphs $X\rightarrow Y$ and $Y\rightarrow X$ to demonstrate this point. As an example, assume that our domain contains the variables (X,Y) and we measure the samples

$$\begin{aligned} D = \{&(0,0),(0,1),(1,1),(1,1),(1,0),(0,0),[\mathrm{do}(X=1),1],\\ &[\mathrm{do}(X=1),1],[\mathrm{do}(X=1),0],[\mathrm{do}(X=0),0]\} \end{aligned} \qquad \text{(Eq. 31)}$$

For these data, the score for the graph structure $X\rightarrow Y$ is –6.46. In contrast, the score for the graph structure $Y\rightarrow X$ is –6.78. Because the score for $X\rightarrow Y$ is better, we conclude that it is more likely that X causes Y.

Although interventions help us to determine the causal direction of some edges, usually these leave the causal direction of the others undetermined. This motivates the development of a notation of equivalence suited for the interventional setting.

Definition 15: For a set of interventions M, the graphs G^1 and G^2 are M-equivalent if, for any set of interventions $m \in M$ (always including $m = \varnothing$), the graph structures $G^1_{\mathrm{do}(m)}$ and $G^2_{\mathrm{do}(m)}$ are equivalent.

Using the empty set of interventions, M-equivalence implies equivalence as defined in Definition 8. The notion of M-equivalence is a more restrictive refinement of Definition 8 in which a larger set of edges are compelled.

Definition 16: An edge $X\rightarrow Y$ in G is M-compelled for a set of interventions M if all graphs that are M-equivalent to G contain the directed edge $X\rightarrow Y$.

Theorem 5: The following edges are M-compelled in G:

1. All edges participating in a v-structure in G.
2. For each set of interventions $m \in M$, all edges entering or leaving any variable X intervened in m.
3. All edges compelled by the repeated application of the propagation rules specified in Fig. 3.

We can use Theorem 5 to identify the edges that can be given a causal interpretation when interventional data are used (*19*). Even when the criteria specified by Theorem 5 are met, care must be taken in their causal interpretation. Signaling networks do not meet all the assumptions required for the correctness of Theorem 5. For example, the underlying signaling network is not acyclic. Although some assumptions do not hold, these criteria do work well in practice. For example, in the network derived in (*1*), the correct directionality was inferred for all edges except one. It is interesting to note that all edges, except for this reversed edge, met the criteria of Theorem 5.

Conclusion

The task of inferring signaling pathway architectures is an important challenge of the postgenomic era, and the method described here is only a small step toward this goal. There are many ways in which the basic framework proposed here can be extended and developed further.

The method described infers an acyclic network and does not consider the timing of signaling events. Given appropriate time-series data, both of these limitations can be overcome by the use of dynamic Bayesian networks (*25*, *26*). Another important area for improvement is to develop more realistic local probability distributions that are capable of modeling the actual kinetics of interactions between signaling molecules (*27*, *28*). Finally, one of the advantages of probabilistic graphic models is that these can be explicitly used to model unobserved variables and infer their structure and activity (*29*, *30*). This feature could be used to discover previously unknown components as well as potential drug targets.

References and Notes

1. K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, G. P. Nolan, Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308**, 523–529 (2005).
2. O. D. Perez, G. P. Nolan, Simultaneous measurement of multiple active kinase states using polychromatic flow cytometry. *Nat. Biotechnol.* **20**, 155 (2002).
3. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann, San Francisco, 1988).
4. N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using Bayesian networks to analyze expression data. *J. Comput. Biol.* **7**, 601 (2000).

5. N. Friedman, I. Nachman, Gaussian process networks. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 2000), pp. 211–219.
6. D. Geiger, D. Heckerman, Learning Gaussian networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1994), pp. 235–243.
7. D. Geiger, T. Verma, J. Pearl, d-separation: From theorems to algorithms. In *Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence* (Elsevier, New York, 1989), pp. 139–148.
8. J. Pearl, T. S. Verma, A theory of inferred causation. In *KR'91: Principles of Knowledge Representation and Reasoning* (Morgan Kaufmann, San Francisco, 1991), pp. 441–452.
9. M. H. DeGroot, *Optimal Statistical Decisions* (McGraw-Hill, New York, 1970).
10. D. Heckerman, D. Geiger, D. M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1994), pp. 293–301.
11. G. F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data. *Mach. Learn.* **9**, 309 (1992).
12. N. Friedman, Z. Yakhini, On the sample complexity of learning Bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1996), pp. 274–282.
13. D. Heckerman, D. Geiger, Learning Bayesian networks: A unification for discrete and Gaussian domains. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1995), pp. 274–284.
14. D. M. Chickering, Learning Bayesian networks is NP-complete. In *Learning from Data: Artificial Intelligence and Statistics V*, D. Fisher, H.-J. Lenz, Eds. (Springer-Verlag, New York, 1996).
15. G. Elidan, M. Ninio, D. Schuurmans, N. Friedman, Data perturbation for escaping local maxima in learning. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence* (AAAI Press, Edmonton, Alberta, Canada, 2002), pp. 132–139.
16. N. Friedman, D. Koller, Being Bayesian about Bayesian network structure: A Bayesian approach to structure discovery in Bayesian networks. *Mach. Learn.* **50**, 95 (2003).
17. B. Efron, R. J. Tibshirani, *An Introduction to the Bootstrap* (Chapman & Hall, London, 1993).
18. N. Friedman, M. Goldszmidt, A. Wyner, Data analysis with Bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1999), pp. 206–215.
19. www.cs.huji.ac.il/~nir/Dissertations/DanaPHD.pdf
20. J. Pearl, *Causality: Models, Reasoning, and Inference* (Cambridge Univ. Press, Cambridge, 2000).
21. D. Heckerman, C. Meek, G. Cooper, A Bayesian approach to causal discovery. In *Computation, Causation, and Discovery*, C. Glymour, G. F. Cooper, Eds. (MIT Press, Cambridge, MA, 1999), pp. 141–166.
22. P. Spirtes, C. Glymour, R. Scheines, *Causation, Prediction and Search*, vol. 81 of *Lecture Notes in Statistics* (Springer-Verlag, New York, 1993).
23. C. Glymour, G. F. Cooper, Eds., *Computation, Causation, and Discovery* (MIT Press, Cambridge, MA, 1999).
24. G. Cooper, C. Yoo, Causal discovery from a mixture of experimental and observational data. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1999), pp. 116–125.
25. N. Friedman, K. Murphy, S. Russell, Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1999), pp. 139–147.
26. S. Kim, S. Imoto, S. Miyanom, Dynamic Bayesian network and non-parametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems* **75**, 57 (2004).
27. N. Friedman, M. Goldszmidt, Learning Bayesian networks with local structure. In *Learning in Graphical Models*, M. I. Jordan, Ed. (Kluwer, Dordrecht, Netherlands, 1998), pp. 421–460.
28. I. Nachman, A. Regev, N. Friedman, Inferring quantitative models of regulatory networks from expression data. *Bioinformatics* **20**, I248 (2004).
29. N. Friedman, The Bayesian structural EM algorithm. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, 1998), pp. 129–138.
30. G. Elidan, N. Lotner, N. Friedman, D. Koller, Discovering hidden variables: A structure-based approach. In *Proceedings of Neural Information Processing Systems 2000* (MIT Press, Cambridge, MA, 2000), pp. 479–485.
31. This primer is based on parts of a doctoral dissertation written under the guidance of N. Friedman. I thank N. Friedman, both for his mentorship and readings of this manuscript; I thank K. Sachs and I. Nachman for readings of this manuscript. Supported by a Pharmaceutical Research and Manufacturers of America Center of Excellence in Integration of Genomics and Informatics grant and a NIH Centers of Excellence in Genomic Science grant to G. Church, and by a NSF postdoctoral research fellowship in biological informatics.

Citation, D. Pe'er, Bayesian network analysis of signaling networks: A primer. *Sci. STKE* **2005**, pl4 (2005).