# 1
# Transcriptional Network Inference based on Information Theory

*Patrick E. Meyer, Catharina Olsen and Gianluca Bontempi*

Machine Learning Group,
Computer Science Department,
Université Libre de Bruxelles, Belgium

## Preface

The reverse engineering of transcriptional regulatory networks (TRNs) from expression data is known to be a very challenging task because of the large amount of noise intrinsic to the microarray technology, the high dimensionality and the combinatorial nature of the problem [33]. Furthermore, a gene-to-gene network inferred on the basis of transcriptional measurements returns only an approximation of a complete biochemical regulatory network since many physical connections between macromolecules might be hidden by short-cuts.

In spite of these evident limitations the bioinformatics community made important advances in this domain over the last few years using network inference methods like Boolean networks, Bayesian networks and Association networks [10].

This chapter will focus on information-theoretic approaches which typically rely on the estimation of mutual information and conditional mutual information from data in order to measure the statistical dependence between genes expression. The adoption of mutual information in network inference can be traced back to Chow and Liu's tree algorithm [7]. Nowadays, two main categories of information-theoretic network inference methods hold the attention of the bioinformatics community: i) methods based on conditional mutual information that are able to infer a larger set of relationships between genes but at the price of a higher algorithmic complexity [6, 15, 37] and ii) methods based on pairwise mutual information that infer undirected networks up to thousands of genes thanks to their low algorithmic complexity [5, 9, 17, 20]. The strengths and weaknesses of these information-theoretic methods are depicted in this chapter.

## 1.1 Introduction

Networks, like signal transduction networks or transcriptional regulatory networks, play a central role in biological systems. In order to get insights about their structure and functionality from experimental data (e.g. microarray data), various methods of network inference have been proposed in the literature.

Network inference consists in representing the dependencies between the variables of a dataset by a graph [34]. The semantics of an arc in the graph may differ from one inference method to another. When network inference is used to reconstruct a transcriptional regulatory network from microarray data [33], arcs are usually representing a regulator/regulated gene interaction where the genes are represented by nodes in the graph. This chapter will focus on a specific family of network inference methods which rely on the estimation of mutual information and conditional mutual information from data in order to measure the statistical dependence between genes expression.

Before introducing the basic notion of information theory , in order to make easier the understanding of the reader, we introduce the notation which will be used throughout the chapter to denote the gene expressions (i.e. the variables of our model) and to identify subsets of genes.

- $A = \{1, \ldots, n\}$ index set
- $X_A = (X_1, X_2, ..., X_n)$ : a set of $n$ random variables
- $X_k \in X_A$ : the k-th variable of the set
- $\mathcal{X}_k$: the domain of values of the k-th variable
- $X_K \subset X_A$: a subset of variables
- $X_{-k} = X_A \setminus X_k$ : subset of $X_A$ with the variable $X_k$ set aside
- $X_{-K}$ : subset of $X_A$ with the subset of variables $X_K$ set aside
- $X_{i,j} = \{X_i, X_j\}$ : two variables of the set $X_A$
- $X_{-(i,j)}$: set of variables $X_A$ without $X_i$ and $X_j$
- $\mathcal{D}$: dataset

The concepts which will be presented in this chapter are mainly based on two conventional information-theoretic measures.

**Definition 1.1** [28] Let $X_i$ and $X_j$ be two (discrete) random variables, $x_i \in \mathcal{X}_i$ and $x_j \in \mathcal{X}_j$ their respective realizations and $p(x_i, x_j)$ their joint probability

distribution. The *mutual information* between two random variables $X_i$ and $X_j$ is defined as

$$I(X_i; X_j) = \sum_{x_i \in \mathcal{X}_i} \sum_{x_j \in \mathcal{X}_j} p(x_i, x_j) \log \left( \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \right).$$

The main characteristics of mutual information are:

- It is a symmetric and non-negative quantity.
- It measures the divergence between the joint and the product of the two marginal distributions.
- $I(X_i; X_j)$ is maximal if $X_i$ ($X_j$) is perfectly predictable once $X_j$ ($X_i$) is known (e.g. deterministic dependence).
- $I(X_i; X_j) = 0$ if $X_i$ ($X_j$) returns no information about $X_j$ ($X_i$), that is $X_i$ and $X_j$ are independent.

**Definition 1.2** [28]
Let $X_i$ and $X_j$ be two (discrete) random variables and $x_i \in \mathcal{X}_i$ and $x_j \in \mathcal{X}_j$ their respective realizations.Once a third random variable $X_k$ is given, the *conditional mutual information* between $X_i$ and $X_j$ knowing $X_k$ is

$$I(X_i; X_j | X_k) = I(X_{i,k}; X_j) - I(X_k; X_j).$$

The main characteristics of the conditional mutual information are

- It measures the amount of information that $X_j$ ($X_i$) provides about $X_i$ ($X_j$ ), once the value of $X_k$ is known.
- $I(X_i; X_j | X_k) \geq 0$ with equality iff $X_i$ and $X_j$ are conditionally independent given $X_k$.

### 1.1.1 Formalization

Let us consider a transcriptional network composed of $n$ genes where $X_j$, $j = 1, \ldots, n$ is the random variable representing the expression of the j-th gene and $X_A = (X_1, X_2, ..., X_n)$ is the set of variables representing the expression of all genes. Let $X_{S_j} \subset X_{-j}$ denote the (potentially empty) subset of genes (whose products are) regulating the expression $X_j$ of the j-th gene.

This induces a network $T$ of relationships between the $n$ variables which can be represented either by a directed graph or equivalently by a square matrix where the $i, j$ element is

$$t_{ij} = \begin{cases} 1 & \text{if } X_i \in X_{S_j} \\ 0 & \text{else} \end{cases} \tag{1.1}$$

Note that the adopted convention for the directionality of directed edges in a graph is $X_i \rightarrow X_j \Leftrightarrow t_{ij} = 1$.

Once a set $\mathcal{D}$ of measurements of the expressions of the $n$ genes is collected, the objective of a network inference algorithm $\mathcal{N}$ is to estimate a network $\hat{T}$ from $\mathcal{D}$,

$$\hat{T} = \mathcal{N}(\mathcal{D}) \tag{1.2}$$

that is as "close" (according to a given performance measure $\Phi$) as possible to the unknown (or partially known) true network $T$:

$$\hat{T}^{max} = \arg\max_{\hat{T}} \Phi(\hat{T}, T) \tag{1.3}$$

Qualitatively, $\hat{T}$ is inferred from a dataset $\mathcal{D}$ that is generated according to the probability distribution $p(X)$ which is itself governed by the underlying network $T$ of genetic interactions.

Typically the network inference task is decomposed into two steps: the first step consists in inferring from data the undirected network (also known as skeleton) and the second one consists in orienting the arcs of the undirected network.

### 1.1.2 Performance Measures in Undirected Network Inference

An undirected network inference problem can be seen as a binary decision problem where the algorithm $\mathcal{N}$ plays the role of a classifier. Each pair of nodes $\hat{t}_{ij}$ is thus assigned either a positive label (1, i.e., an edge) or a null one (0, i.e., no edge). A reference network $T$ is then required to assess the performances of the inferred network $\hat{T}$. Two strategies are usually adopted to define a reference network:

1. The dataset is synthetically generated by simulating a known reference network.

2. The inference concerns data where several variable interactions are known (i.e., genetic interactions that have already been discovered by researchers). This list of known interactions can then be used as a reference network, in order to discover new potential interactions.

A positive label (an edge) predicted by the algorithm is considered as a true positive ($tp$) or as a false positive ($fp$) depending on the presence or not of the corresponding edge in the reference network. Analogously, a null label is considered as a true negative ($tn$) or a false negative ($fn$) depending on whether the corresponding edge is absent or not in the underlying true network $T$, respectively (Table 1.1). Most of the inference methods which will be discussed in this chapter return a weighted adjacency matrix $W = (w_{ij})_{i,j \in A}$ representing the network. Hence, a threshold value $\theta$ is used to remove the arcs of the network that have a too low score [4, 9, 10].

$$\hat{t}_{ij} = \begin{cases} 1 & \text{if } w_{ij} \geq \theta, \\ 0 & \text{otherwise.} \end{cases} \tag{1.4}$$

For each threshold value $\theta$, a different inferred network $\hat{T}(\theta, W)$ based on the weighted adjacency matrix $W$ can be computed. As a result, a specific confusion matrix CM (Table 1.1) is obtained for each $\theta$.

**Tab. 1.1:** Confusion matrix CM

| Inferred \Truth | True edge (1) | No edge (0) |
|---|---|---|
| Inferred edge (1) | $tp = \#(\hat{t}_{ij} = t_{ij} = 1)$ | $fp = \#(\hat{t}_{ij} = 1 \neq t_{ij} = 0)$ |
| Deleted edge (0) | $fn = \#(\hat{t}_{ij} = 0 \neq t_{ij} = 1)$ | $tn = \#(\hat{t}_{ij} = t_{ij} = 0)$ |

The two following sections will introduce two accuracy measures $\Phi$ to assess the performance of a network inference technique.

#### 1.1.2.1 Precision-Recall curves

Precision and recall are two useful measures, borrowed from the information retrieval litterature, to assess the quality of an inferred network.

The *precision* quantity is given by

$$pre = \frac{tp}{tp + fp} = \frac{tp}{\#(\hat{t}_{ij} = 1)} \tag{1.5}$$

and measures the fraction of real edges among the ones classified as positive.

The *recall* quantity (also called true positive rate ($tpr$)) is given by

$$rec = tpr = \frac{tp}{tp + fn} = \frac{tp}{\#(t_{ij} = 1)} \tag{1.6}$$

and denotes the fraction of real edges that are correctly inferred. The objective of an inference method is to maximize both precision and recall. These quantities depend on the chosen threshold $\theta$.

The PR curve is a diagram which plots precision (*pre*) against recall (*rec*) for different values of the threshold [30](see Figure 1.1 for a PR curve). In a network inference setting, this diagram illustrates the trade-off between returning a small amount of arcs (low recall due to high threshold) with high confidence (high precision) and returning many arcs (high recall) with low confidence (low precision).

#### 1.1.2.2 F-Scores

The multicriteria nature of the inference problem can also be addressed by defining univariate measures which weight precision and recall. An example is given by the area under the PR-curve [2].

Another well-known measure is the F-score quantity [30] which is a weighted harmonic average of precision and recall:

$$F_\beta(\hat{T}(\theta, W),\, T) = \frac{(1+\beta^2)(pre)(rec)}{\beta^2 pre + rec} \tag{1.7}$$

where $\beta$ is a non-negative real parameter weighting the importance of recall versus that of precision. The F-score takes value in the interval $[0, 1]$.

Three common values for the parameter $\beta$ are

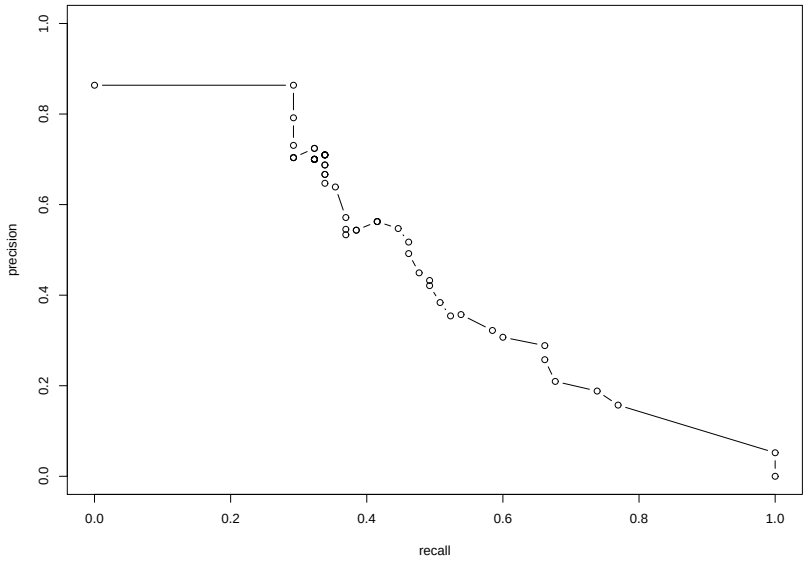


**Fig. 1.1** Example of PR curves (generated with the R package 'minet' [21])

- $\beta = 1$: assigning equal weight to precision and recall
- $\beta = 2$: (the $F_2$-measure), which weighs recall twice as much as precision
- $\beta = 0.5$: (the $F_{0.5}$-measure), which weighs precision twice as much as recall.

Note that in transcriptional network inference, precision is often valued higher than recall since the experimental cost to check for possible interactions is high.

A compact representation of the PR diagram can be returned by the maximum and/or the average (avg) $F_\beta$-score.

$$\begin{aligned} F_\beta^{max}(W, T) &= \max_\theta \left(F_\beta(\hat{T}(\theta, W), T)\right) \\ F_\beta^{avg}(W, T) &= avg_\theta \left(F_\beta(\hat{T}(\theta, W), T)\right) \end{aligned} \tag{1.8}$$

where $\theta$ is the threshold parameter.

### 1.1.3 Causal Subset Selection

The inference of a network from data requires for each gene the identification of its own family of regulators. This problem is extremely complex since, given $n-1$ variables in $X_{-j}$, there is an exponential number $2^{n-1}$ of candidate subsets $X_{S_j}$.

An approach to solve this problem is given by *causal Bayesian networks* that provide a theoretical framework to identifying a causal subset $X_{S_j}$ of a variable $X_j$.

**Definition 1.3** $X_i$ is a *cause* of $X_j$, denoted by $X_i \rightarrow X_j$, if there exists a value $x_i \in \mathcal{X}_i$ such that setting $X_i = x_i$ leads to a change in the probability distribution of $X_j$ [24].

The definition of causality states that a causal relation between two variables creates a stochastic dependency between the probability distributions of causes and effects. Thus, two causally linked variables are not independent and therefore the mutual information is larger than zero.

$$X_i \leftrightarrow X_j \Rightarrow I(X_i; X_j) > 0 \tag{1.9}$$

where $X_i \leftrightarrow X_j$ denotes an *undirected causal link*, i.e., $X_i \rightarrow X_j$ or $X_i \leftarrow X_j$. Unfortunately, since mutual information, unlike causality, is a symmetric measure, it is not possible to derive the direction of an edge.

However, the bivariate dependency stated in (1.9) is not always true. Two known examples are the cancellation of two causal pathways [24] and the XOR problem [26]. Since this occurs only under very special conditions, in the

following we will keep on assuming that *causality implies stochastic dependency*.

However, the converse is not true, dependency does not imply causality

$$I(X_i; X_j) > 0 \not\Rightarrow X_i \leftrightarrow X_j. \tag{1.10}$$

One counter-argument to the idea of *dependency implying causality* relies on the *common-cause effect*. That is, a dependency between two variable $X_i$ and $X_j$ can be created by a common cause, i.e., $X_i \leftarrow X_k \rightarrow X_j$. These two variables can be dependent but manipulating one of them does not influence the other.

An illustration of this case is the well-known sentence "curing the symptoms does not cure the disease".

In this context, the definition of direct causality provides a solution to the problem of deriving causal dependency from stochastic dependency.

**Definition 1.4** $X_i$ is a *direct cause* of $X_j$ if $X_i$ is a cause of $X_j$ and there is no other variable $X_k$ such that once the value of $X_k$ is known, a manipulation of $X_i$ no longer changes the probability distribution of $X_j$ [24].

This definition states that

if there are no sets of variables that cancel the dependency between two other variables, then one of the variables is a direct cause of the other.

In other words, it can be stated that if *two variables are dependent in every context*, then these variables have a causal relationship.

Once we make an additional assumption (known as *causal sufficiency condition*) the definition of direct cause provides the following implication in information-theoretic terms [6]:

$$\forall X_K \subseteq X_{-(i,j)} : \; I(X_i; X_j | X_K) > 0 \Rightarrow X_i \leftrightarrow X_j \tag{1.11}$$

The causal sufficiency condition requires that all variables that are causes to at least two effects (two variables in the dataset) be present in the set of measured variables. Indeed, if there is a common cause to two observable effects, the two effects are dependent in every context except when conditioning on the common cause. If the common cause is hidden then (1.11) can lead to false conclusions about the causal relationships between the variables.

However, it should be noted that the causal sufficiency condition does not concern intermediate unidentified variables along the *causal direction* $X_i \rightarrow X_k \rightarrow X_j$, as illustrated by this example from [31].

**Example 1** *If A is the event of striking a match and C is the event of the match catching on fire and no other events are considered, then A causes C. However, if we add B which stands for the sulfur on the match tip that achieved sufficient heat to combine with oxygen, then A no longer causes C directly. Rather A causes B and B causes C, i.e.,* $A \rightarrow B \rightarrow C$.

There is a fundamental difference between having hidden variables along each edge in the causal direction and hidden variables that are common causes of several effects. Along a causal direction it is intuitive to accept that the causality between a grandparent and a grandson is preserved once the parent is removed. This corresponds to make an assumption of *causal transitivity*, that is

$$\left.\begin{array}{l} X_i \rightarrow X_k \\ X_k \rightarrow X_j \end{array}\right\} \Rightarrow X_i \rightarrow X_j.$$

However, if a common parent is missed and a link between a variable and its sibling is added, then such link is no longer causal since acting on the sibling does not change the distribution of the variable.

Hence, the causal sufficiency condition can be rephrased by assuming that *there is no hidden common cause (to at least two effects)* in the set of considered variables, i.e.,

$$\forall (X_i, X_j) \in X,\ \nexists (X_h \notin X) :\ X_i \leftarrow X_h \rightarrow X_j \tag{1.12}$$

The notions of information theory and causality introduced so far will be used in the rest of the chapter to present two main network inference approaches which hold the attention of the bioinformatics community: i) methods based on conditional mutual information that are able to infer a larger set of relationships between genes but at the price of a higher algorithmic complexity and ii) methods based on bivariate mutual information that infer undirected networks up to thousands of genes thanks to their low algorithmic complexity.

## 1.2 Inference Based on Conditional Mutual Information

Once causal sufficiency is assumed, the notion of conditional information allows the definition of a simple algorithm to infer an undirected network from observed data. The algorithm consists in

setting an undirect causal link between all couples $(X_i, X_j)$ such that

$$\forall X_K \subseteq X_{-(i,j)} :\ I(X_i; X_j | X_K) > 0 \Rightarrow X_i \leftrightarrow X_j$$

In other terms this means to remove a link between the couples $(X_i, X_j)$ if at least a conditioning set $X_K$ such that $I(X_i, X_j | X_K) = 0$ exists. This is the

strategy adopted in constraint-based Bayesian network algorithms which will be briefly discussed in the following section.

### 1.2.1 Constraint-based methods

**Algorithm 1**: pseudo code of IC-algorithm

```
Start from an empty graph
foreach pair of variables (X_i, X_j) do
    if there exists no subset X_K in X_{-(i,j)} such that I(X_i; X_j | X_K) = 0 then
        set an edge connecting (X_i, X_j)
    end
end
```

**Algorithm 2**: pseudo code of the SGS algorithm

```
Start from the complete (fully connected) graph
foreach pair of variables (X_i, X_j) do
    if there exists a subset X_K in X_{-(i,j)} such that I(X_i; X_j | X_K) = 0 then
        remove the edge connecting (X_i, X_j)
    end
end
```

IC (Algorithm 1, [26]) and SGS (Algorithm 2, [31]) are two state-of-the-art contraint-based algorithms which infer the network by carrying out a set of conditional independence tests. Note that IC proceeds in a forward manner by starting with an empty graph while SGS proceeds in a backward manner by removing progressively edges from a fully connected graph. However in both algorithms, the IF instruction in the third line requires a computationally expensive procedure related to the search in the space of conditioning sets $X_K \subseteq X_{-(i,j)}$. This means that, given $n$ variables, there are $2^{n-2}$ potential subsets for each couple $(X_i, X_j)$. In order to address this issue, the PC algorithm has been proposed by [26] to speed up the SGS algorithm by replacing the IF test with the pseudo code detailed in Algorithm 3. The rationale of the PC algorithm is to make conditional independence tests by using growing conditioning sets.

In spite of this improvement, these algorithms are not affordable in problems where the number $n$ of variables is in the order of several thousands, like in transcriptional network inference from microarray data.

**Algorithm 3**: pseudo code of the subset search procedure in PC-algorithm

```
foreach subset size going from |K| = 0 to |K| = n − 2 do
    foreach subset of variables X_K of size |K| do
        if I(X_i; X_j|X_K) = 0 then
            remove the edge connecting (X_i, X_j)
            quit the two for loops
        end
    end
end
```

### 1.2.2 Approximated Conditional Mutual Information

In [37], the authors propose an information-theoretic translation of the PC algorithm that uses:

- a single conditioning variable $X_k$ instead of the set of variables $X_K$, replacing $I(X_i; X_j|X_K)$ by $I(X_i; X_j|X_k)$.
- two thresholds $\theta_1$ and $\theta_2$ representing estimation biases in the independence tests.

**Algorithm 4**: pseudo code of [37]

```
Start from an empty graph
foreach pair of variables (X_i, X_j) do
    if I(X_i; X_j) > θ_1 then
        if there exists no variable X_k in X_{−(i,j)} such that I(X_i; X_j|X_k) ≤ θ_2 then
            set an edge connecting (X_i, X_j)
        end
    end
end
with θ_1 and θ_2 thresholds (parameters) of the method
```

Both modifications of the PC algorithm render this method adapted to real microarray datasets.

### 1.2.3 Variable Selection Algorithms

An alternative to the use of conditional independence tests is the use of variable selection methods for network inference. The idea is to perform network

inference by repeating a variable selection step in the space $X_{-j}$ for each variable variable $X_j \in X$.

A generic information-theoretic objective of variable selection can be formulated as follows [22]:

*Given an output variable $X_j$ and $n-1$ input variables $X_{-j}$ , find the smallest subset $X_S \subseteq X_{-j}$ that maximizes the mutual information $I(X_S; X_j)$ between inputs and output.*

Note that maximizing the (mutual) information of a subset of variables is equivalent to reducing the uncertainty (entropy) of the target variable.

Using variable selection strategies for network inference has many practical and theoretical advantages [11,32]. For instance:

1. Some variable selection algorithms, like filters, can deal with thousands of variables in a reasonable amount of time. This makes inference scalable to large networks.

2. Variable selection algorithms may be easily executed in parallel, since each of the $n$ subset selection tasks is independent.

3. Variable selection algorithms can use a priori knowledge. For example, knowing the list of regulator genes of an organism can improve the selection speed and the inference quality by limiting the search space of the selection step to a smaller list of genes.

Another advantage of a variable selection approach is that the subset that maximizes mutual information with the target contains all direct (causal) interactions [32].

The disadvantage is that this subset can also contain non-causal variables [32]. This results from the *explaining away effect*.

**Definition 1.5** [The explaining away effect] [24] Once the value of a common effect is given, it creates a dependency between its causes because each cause explains away the occurrence of the effect, thereby making the other cause less likely.

This is a common mechanism used by medical doctors when doing their diagnoses.

**Example 2** *Some cancer can cause headache but a lot of more probable diseases, such as a cold, can also cause headache. Once a doctor has evidence that headaches are caused by a cold, he stops searching for a cancer although having cold and having a cancer are two independent events (see Figure 1.2).*

As a result, a variable selection algorithm should select the variables *cold* and *headache* since they reduce the uncertainty on the target variable *cancer* while *cold* in this case, is not a causal variable. Indeed, acting on *cold* does not modify the probability distribution of *cancer*.

In information-theoretic terms, the explaining away effect can be expressed by a conditional mutual information higher than the mutual information [6, 12, 15]:

$$X_i \rightarrow X_k \leftarrow X_j \Rightarrow I(X_i; X_j | X_k) > I(X_i; X_j) \tag{1.13}$$

This effect, is also known as negative interaction [13], complementarity [22] or synergy [1].

In order to avoid this problem, one should determine which of the selected variables are indirect links and eliminate them from the selection. This can be done by modifying properly the PC algorithm (Algorithm 3) and using it to explore, instead of the whole search space $X_{-(i,j)}$, the selected subset $X_S$ [3]. It follows that the procedure is still exponential but in the size $|S|$ of the subset selected.

## 1.3 Inference Based on Pairwise Mutual Information

This section will present a set of algorithms for inferring networks from observed data relying only on the computation of pairwise mutual information. In these methods, a link between two nodes is set if its corresponding score (based on pairwise mutual information) is higher than the chosen threshold.

All these methods require the computation of the mutual information matrix $\text{MIM} = (\text{mim}_{ij})_{i,j \in A}$, a square matrix whose $ij$-th element is given by,

$$\text{mim}_{ij} = I(X_i; X_j) \tag{1.14}$$

This is the mutual information between $X_i$ and $X_j$, where $X_i$ and $X_j$ are random variables denoting the expression level of the $i$th/$j$th gene in a transcriptional regulatory network inference.

These methods have major advantages which enable them to deal with microarray data:

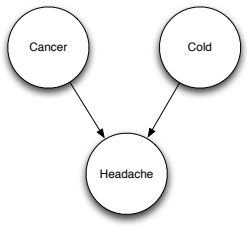


**Fig. 1.2** Illustration of the explaining away effect.

- an affordable computational complexity. This results from the fact that only $\frac{n}{2}(n-1)$ computations of mutual information, based on bivariate probability distributions, are required to obtain the MIM [17].
- they do not require a large amount of samples, since only bivariate distribution are to be estimated. Hence even basic entropy estimators perform well with these methods [25]. Most of these methods can be tested using the Bioconductor *minet* package [21].

### 1.3.1 Relevance Network (RELNET)

The relevance network approach [5] has been introduced for gene clustering and successfully applied to infer relationships between RNA expression and chemotherapeutic susceptibility [4]. This method infers a network in which a pair of genes $\{X_i, X_j\}$ is linked by an edge if the mutual information $I(X_i; X_j)$ is larger than a given threshold $\theta$. The complexity of the method is $O(n^2)$ since all pairwise interactions are considered.
This method relies on the assumption *causality implies dependency* (Section 1.1.3): $X_i \leftrightarrow X_j \Rightarrow I(X_i; X_j) > 0$. However, it does not eliminate all indirect interactions between genes. For example, if gene $X_1$ regulates both gene $X_2$ and gene $X_3$, this would cause a high mutual information between the pairs $\{X_1,X_2\}$, $\{X_1,X_3\}$ and $\{X_2,X_3\}$. As a consequence, the algorithm will set an edge between $X_2$ and $X_3$ although these two genes interact only through gene $X_1$.

Note that if one considers correlation instead of mutual information this approach boils down to building a correlation network [14].

### 1.3.2 Context Likelihood of Relatedness (CLR)

The CLR algorithm [9] is an extension of the RELNET algorithm. This algorithm derives a score from the empirical distribution of the mutual information for each pair of genes. In particular, instead of considering the information $I(X_i; X_j)$ between genes $X_i$ and $X_j$, it takes into account the score $w_{ij} = \sqrt{z_i^2 + z_j^2}$ where

$$z_i = \max\left(0, \frac{I(X_i; X_j) - \mu_i}{\sigma_i}\right) \tag{1.15}$$

where $\mu_i$ and $\sigma_i$ are respectively the mean and standard deviation of the empirical distribution of the mutual information values $I(X_i, X_k)$, $X_k \in X_{-i}$. The pseudo code of CLR is given in Algorithm 5. The CLR algorithm was successfully applied to decipher the *E. Coli* transcriptional regulatory network [9]. CLR has a complexity in $O(n^2)$ once the MIM is computed.

**Algorithm 5**: pseudo code of the normal version of CLR algorithm

**Input**: $I(X_i; X_j),\ \forall i, j \in A = \{1, 2, ..., n\}$
**Output**: the weighted adjacency matrix $W$ (having elements $w_{ij}$)
**foreach** *input* $X_i$ *in the input space* $X$ **do**
  $\mu_i \leftarrow mean(I(X_i; X_j), j \in \{1, 2, ..., n\})$
  $\sigma_i \leftarrow variance(I(X_i; X_j),\ j \in \{1, 2, ..., n\})$
**end**
**foreach** *pair of variables* $X_{i,j}$ *in the input space* $X$ **do**
  $w_{ij} \leftarrow \max\left(0, \frac{1}{\sqrt{2}} * \left\{\frac{I(X_i;X_j)-\mu_i}{\sigma_i} + \frac{I(X_i;X_j)-\mu_j}{\sigma_j}\right\}\right)$
**end**

### 1.3.3 Chow-Liu Tree

The Chow and Liu approach consists in finding the maximum spanning tree on the complete graph whose edge weights are the mutual information between two nodes [7].

In graph theory, a tree is a graph in which any two vertexes are connected by exactly one path. A spanning tree is a tree that connects all the vertexes of the graph. The maximum spanning tree is the spanning tree whose sum of edge weights is greater than or equal to that of every other spanning tree.

A maximum spanning tree can be computed in $O(n^2 \log n)$ using, for example, Kruskal's algorithm [23]. The drawback of this method lies in the fact that the resulting network has typically a low number of edges. Also precision and recall cannot be studied as a function of a parameter.

### 1.3.4 The Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE)

ARACNE [17] is based on the Data Processing Inequality [8]. If gene $X_i$ interacts with gene $X_j$ through gene $X_k$, then $I(X_i; X_j) \leq \min\left(I(X_i; X_k), I(X_j; X_k)\right)$. ARACNE begins by assigning to each pair of nodes a weight equal to their mutual information. Then, as in RELNET, all edges for which $I(X_i; X_j) < \theta$ are removed, with $\theta$ a given threshold. Eventually, the weakest edge of each triplet is interpreted as an indirect interaction and is removed (see pseudo code Algorithm 6).

An extension of ARACNE removes the weakest edge only if the difference between the two lowest weights lies above a threshold $\eta$. Hence, increasing $\theta$ lowers the number of inferred edges while the opposite happens when increasing $\eta$.

If the network is a tree including only pairwise interactions, the method guarantees the reconstruction of the original network, once it is provided with the exact MIM (see [17]). ARACNE's complexity is $O(n^3)$ since the algorithm considers all triplets of genes. In [17] the method has been able to recover components of the transcriptional regulatory network in mammalian cells and has outperformed Bayesian networks and relevance networks on several inference tasks [17]. Chow-Liu tree is proved to be a subnetwork of the network reconstructed by the ARACNE algorithm [17].

**Algorithm 6**: pseudo code of ARACNE algorithm

**Input**: the MIM, i.e., $I(X_i; X_j), \forall i, j \in \{1, 2, ..., n\}$
**Output**: the weighted adjacency matrix $W$ (having $w_{ij}$ as elements)
**foreach** *pair of variables* $X_{i,j}$ *in the input space* $X$ **do**
  **foreach** *variable* $X_k$ *in the space* $X_{-(i,j)}$ **do**
    **if** $I(X_i; X_j) < \theta$ **then**
      $w_{ij} \leftarrow 0$
    **else if** $(I(X_i; X_j) < I(X_i; X_k) \quad and \quad I(X_i; X_j) < I(X_j; X_k))$ **then**
      $w_{ij} \leftarrow 0$
    **else**
      $w_{ij} \leftarrow I(X_i; X_j)$
    **end**
  **end**
**end**

### 1.3.5 Minimum Redundancy Networks (MRNET)

MRNET [20] is a method based on the Maximum Relevance Minimum Redundancy (MRMR) variable selection procedure [27]. In other words, MRNET can be situated at the intersection of network inference methods based on pairwise mutual information (Section 1.1 ) and variable selection strategies (Sec. 1.2.3). Let $X_j$ be the output variable, the MRMR methods ranks the set $X_{-j}$ of inputs according to a score that is the difference between the mutual information with the output variable $X_j$ (the relevance) and the average mutual information with the previously ranked variables (the redundancy) $X_S$. The rationale is that direct interactions should be well ranked whereas indirect interactions (i.e. the ones with redundant information with the direct ones) should be badly ranked by the method. The MRMR method relies on a greedy search that starts by selecting the variable $X_k$ providing the highest mutual information to the target $X_j$. The second selected variable $X_i$ will be the one with a high information $I(X_i; X_j)$ to the target and at the same time a

low information $I(X_i; X_k)$ to the previously selected variable. In the following steps, given a set $X_S$ of selected variables, the criterion updates $X_S$ by choosing the variable which maximizes the MRMR score.

$$X_i^{MRMR} = \arg \max_{X_i \in X_{-(i,j)}} \left( I(X_i; X_j) - \frac{1}{|S|} \sum_{k \in S} I(X_i; X_k) \right) \tag{1.16}$$

At each step of the algorithm, the selected variable is expected to allow an efficient trade-off between relevance and redundancy. The network inference approach MRNET, consists in repeating this selection procedure for each target gene $X_j \in X$. For each pair $\{X_i, X_j\}$, MRMR returns two (not necessarily equal) scores $s_i$ and $s_j$ according to (1.16). The score of the pair $\{X_i, X_j\}$ is then computed by taking the maximum between $s_i$ and $s_j$. A specific network can then be inferred by deleting all the edges whose score lies below a given threshold $\theta$ (as in RELNET, CLR and ARACNE). Thus, the algorithm infers an edge between $X_i$ and $X_j$ either when $X_i$ is a well-ranked predictor of $X_j$ ($s_i > \theta$), or when $X_j$ is a well-ranked predictor of $X_i$ ($s_j > \theta$).

An effective implementation of the greedy search based on a similarity matrix is given in [19]. This implementation demands an $O(f \times n)$ complexity for selecting $f$ variables. It follows that MRNET has an $O(f \times n^2)$ complexity since the variable selection step is repeated for each of the $n$ genes. In other terms, the complexity ranges between $O(n^2)$ and $O(n^3)$ according to the value of $f$. In practice, the selection of variables is stopped when the average redundancy term $\frac{1}{|S|} \sum_{k \in S} I(X_i; X_k)$ exceeds the relevance term $I(X_i; X_j)$.

Although MRNET is based on a variable selection strategy, it does not suffer from the explaining away effect as most variable selection methods (see Section 1.2.3): $X_i \rightarrow X_k \leftarrow X_j \Rightarrow I(X_i; X_j | X_k) > I(X_i; X_j)$. Indeed, the MRMR criterion only relies on pairwise interactions, hence it does not measure the increase in information due to conditioning $I(X_i; X_j | X_k)$. Instead, it will consider the score $s_i = I(X_i; X_j) - I(X_i; X_k)$ where $I(X_i; X_j) = 0$ since $X_i$ and $X_j$ are independent and $I(X_i; X_k) > 0$ since $X_i$ is a cause of $X_k$ (compare Example 1.2.3). This score $s_i$ is negative and $X_i$ will be badly ranked by MRMR. The same will happen for ranking $X_j$ as a predictor of $X_i$.

Several papers [16,20,25,29] have experimentally shown the accuracy of the MRNET method both in simulated and real data network inference tasks.

## 1.4 Arc Orientation

So far, we have presented several strategies to infer an undirected network from data. The remaining step to be performed consists in the orientation of the inferred arcs. So, the question that arises, is the following: *can the direction of an arc be inferred in an observational context*? Surprisingly the answer to that

**Algorithm 7**: Detailed pseudo-code of the MRNET algorithm (given the $MIM$).

**Input**: a matrix of weights MIM (with elements $I(X_i; X_j)$) of size $n$
**Output**: the weighted adjacency matrix $W$
Initialize the weighted adjacency matrix $W$ to $n \times n$ zeros
**foreach** *variable* $j \in A = \{1, 2, ..., n\}$ **do**
  Initialize search space: $R \leftarrow A \setminus \{j\}$
  Initialize selected variable: $S \leftarrow \emptyset$
  Initialize *relevance* vector: $relevance_j \leftarrow I(X_i; X_j), i \in R$
  Initialize *redundancy* vector: $redundancy_j \leftarrow 0, i \in R$
  **while** $w_{kj} > 0$ **do**
    Select best variable: $k \leftarrow \arg\max_{i \in R}(relevance_i - redundancy_i/|S|)$
    Update subset : $S \leftarrow \{S, k\}$
    Update matrix: $w_{kj} \leftarrow relevance_k - redundancy_k/|S|$
    Update search space : $R \leftarrow R \setminus k$
    Update *redundancy* vector:
    $redundancy_i \leftarrow redundancy_i + I(X_i; X_k), i \in R$
  **end**
**end**

question is positive. This can be done in some cases thanks to the previously seen explaining away effect $X_i \rightarrow X_k \leftarrow X_j \Rightarrow I(X_i; X_j|X_k) > I(X_i; X_j)$.

Let us first remark that because of the following equality

$$I(X_i; X_j) - I(X_i; X_j|X_k) = I(X_i; X_k) - I(X_i; X_k|X_j) = I(X_j; X_k) - I(X_j; X_k|X_i) \tag{1.17}$$

the reversal statement of (1.13) is not necessarily true,

$$I(X_i; X_j|X_k) > I(X_i; X_j) \not\Rightarrow X_i \rightarrow X_k \leftarrow X_j$$

since $I(X_i; X_j|X_k) > I(X_i; X_j)$ could also imply $X_k \rightarrow X_i \leftarrow X_j$ or $X_i \rightarrow X_j \leftarrow X_k$. Notwithstanding, given particular configurations of the undirected network, arcs can be oriented thanks to the explaining-away effect. This can be done in two ways:

First, if the variable $X_k$ is linked to $X_i$ and $X_j$, and an explaining away effect between them has been detected, then the variable $X_k$ is a common effect of the two other variables. More formally,

$$\left.\begin{array}{c} X_i \leftrightarrow X_k \leftrightarrow X_j \\ I(X_i; X_j|X_k) > I(X_i; X_j) \end{array}\right\} \Rightarrow X_i \rightarrow X_k \leftarrow X_j \tag{1.18}$$

Second, if the variable $X_k$ is a consequence of $X_i$ and is linked to $X_j$ and no explaining away effect occurs between them, then $X_j$ is a consequence of $X_k$:

$$\left.\begin{array}{c} X_i \rightarrow X_k \leftrightarrow X_j \\ I(X_i; X_j | X_k) \leq I(X_i; X_j) \end{array}\right\} \Rightarrow X_i \rightarrow X_k \rightarrow X_j \tag{1.19}$$

The two rules (1.18) and (1.19) are an information-theoretic translation of the arc orientation criteria used in [26,31].

Note that the explaining away effect $X_i \rightarrow X_k \leftarrow X_j \Rightarrow I(X_i; X_j | X_k) > I(X_i; X_j)$ is true in general, but not always. Consider the following example:

**Example 3** *Disease A can cause the skin covered by eczema and disease B can cause the skin have wounds. Let the variable S (skin), have three possible values (eczema, wounds, normal). Although disease A and B are two causes of skin injuries S, there is no more information brought by an evidence in favor of one of the diseases than the information already given by the state of the skin.*

This example illustrates that, in order to observe an explaining away effect, the two causes should have the same effect on the same target variable [24].

Although directed cycles can represent feed-back and feed-forward effects, there is no suitable joint probability to model these situations. Distributions such as $p(A|B)p(B|C)p(C|A)$ are not well defined probability distributions, apart from very special cases [34]. Furthermore, the notion of loops is related to dynamics whereas the probability distribution modeled here comes from samples with no temporal dependencies. Hence, a third rule commonly used to orient arcs in a partially oriented network consists in removing cycles in triplets of variables [18].

In order to guarantee a correct inference of an oriented network, additional assumptions are required. Commonly, the causal Markov and the causal faithfulness assumptions are made [26,31]. These two properties are defined as follow.

**Definition 1.6** The *causal Markov condition* holds if every variable is statistically independent of its non-effects conditional on its direct causes.

**Definition 1.7** The *faithfulness condition* holds if the only existing conditional independencies are those specified by the causal Markov condition.

Note, that these two conditions include the assumptions defined above such as causality implies dependency or causal transitivity (for a detailed analysis of these assumptions see [35].)

### 1.4.1 Assessing arc orientation methods

Similar to the undirected case, the quality of the arc orientation process has to be assessed. For this task, there are three quantities of interest: the number

**Algorithm 8**: pseudo code of SGS/IC arc orientation step for conditioning sets of size one

**foreach** *triplet* $X_i \leftrightarrow X_k \leftrightarrow X_j$ *with no link connecting* $X_i \leftrightarrow X_j$ **do**
  $I(X_i; X_j|X_k) > I(X_i; X_j) \Rightarrow X_i \rightarrow X_k \leftarrow X_j$
**end**
**while** *there remain undirected edges, orient edges subject to* **do**
  avoiding new colliders
  avoiding cycles
**end**

**Algorithm 9**: pseudo code of the "remove cycles" step of [3] algorithm

compute the list of cycles in the graph
**while** *there exists cycles in the graph* **do**
  reverse edge that belongs the highest amount of cycles
  update the list of cycles
**end**

of correctly oriented arcs, the ones which are oriented in the wrong direction and those arc which have not been oriented at all. Depending on the problem, different strategies are applied in the literature. If one is interested in the overall performance, the number of correctly oriented arcs is compared to the number of wrongly oriented arcs. The problem with this method is, that the more arcs are oriented, the less significant the orientation is (due to the decreasing score of a arc). Another evaluation possibility is a weighted matrix, which is interesting when searching for the directionalities in a local network, e.g. the arcs in a certain "neighborhood" of the target variable. In this case, the orientation of arcs close to the target can be higher rated than that of arcs further away.

When taking into consideration the first possibility, one can use the analog measures to the ones presented in Section 1.1.2 as F-score or PR curves. In order to compare the true network and the inferred causal network, the notions of precision and recall have to be adopted to the partially oriented networks. In [36], four additional categories to the already introduced quantities fn, tp, tn and fp are defined: pfn (partial false negative) = ptp (partial true positives), ptn (partial true negatives) = pfp (partial false positives). The former ones denote the number of arcs for which $X - Y$ is in the obtained network and either $X \rightarrow Y$ or $Y \rightarrow X$ is the true network. The latter one denotes the case when $X - Y$ is in the obtained graph and neither $X \rightarrow Y$ nor $Y \rightarrow X$ are in the

original network. Precision and recall are then defined as follows

$$rec = \frac{|tp| + \frac{|ptp|}{2}}{|fn| + \frac{|pfn|}{2} + |tp| + \frac{|ptp|}{2}} \tag{1.20}$$

$$pre = \frac{|tp| + \frac{|ptp|}{2}}{|fp| + \frac{|pfp|}{2} + |tp| + \frac{|ptp|}{2}} \tag{1.21}$$

### Summary

Transcriptional network inference aims at representing interactions between transcription factors and regulated genes with a graph. Most methods introduced in this chapter are able to distinguish between dependency (e.g. corregulated gene) and causality (e.g. transcription factor) by relying on assumptions such as *causality implies dependency* or *causal sufficiency*. Among inference methods based on information theory, mutual information networks, relying on the matrix of pairwise mutual information, are particularly adapted to large number of variables and low number of samples typically encountered in microarray data. However, these methods only infer an undirected graph. In order to orient the arcs, a second step is required such as methods based on the *explaining away effect*. Finally, validation measures such as PR-curves and F-scores have been introduced here in order to assess inferred networks. Many tools and algorithms introduced in this chapter can be tested using the Bioconductor package *minet*.

### Bibliography


**1** Dimitris Anastassiou. Computational analysis of the synergy among multiple interacting genes. *Mol Syst Biol*, 3, February 2007.

**2** Sahely Bhadra, Chiranjib Bhattacharyya, Nagasuma R Chandra, and Saira Mian. A linear programming approach for estimating the structure of a sparse linear genetic network from transcript profiling data. *Algorithms for Molecular Biology*, 4:5+, February 2009.

**3** Facundo Bromberg and Dimitris Margaritis. Improving the reliability of causal discovery from small data sets using argumentation. *Journal Machine Learning Research*, 10, 2009.

**4** A. J. Butte, P. Tamayo , D. Slonim, T.R. Golub, and I.S. Kohane. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences*, 97(22):12182–12186, 2000.

**5** A. J. Butte and I. S. Kohane. Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurments. *Pacific Symposium on Biocomputing*, 5:415–426, 2000.

**6** J. Cheng, R Greiner, J. Kelly, D. Bell, and W. Liu. Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137(1), 2002.

**7** C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14, 1968.

**8** T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1990.

**9** J.J. Faith, B. Hayete, J.T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J.J. Collins, and T.S. Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, 5, 2007.

**10** T. S. Gardner and J. Faith. Reverse-engineering transcription control networks. *Physics of Life Reviews* 2, 2005.

**11** K. Hwang, J. W. Lee, S. Chung, and B. Zhang. Construction of large-scale bayesian networks by local to global search. In *7th Pacific Rim International Conference on Artificial Intelligence*, 2002.

**12** A. Jakulin and I. Bratko. Quantifying and visualizing attribute interactions, 2003.

**13** A. Jakulin and I. Bratko. Testing the significance of attribute interactions. In *Proc. of 21st International Conference on Machine Learning (ICML)*, pages 409–416, 2004.

**14** B. H. Junker and F. Schreiber. *Analysis of Biological Networks*. Bioinformatics. Wiley-Interscience, 2008.

**15** K. Liang and X. Wang. Gene regulatory network reconstruction using conditional mutual information. *EURASIP Journal on Bioinformatics and Systems Biology*, 2008.

**16** F. M. Lopes, D. C. Martins, and R. M. Cesar. Comparative study of grns inference methods based on feature selection by mutual information. In *IEEE International Workshop on Genomic Signal Processing and Statistics*, 2009.

**17** A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7, 2006.

**18** C. Meek. Strong completeness and faithfulness in Bayesian networks. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence,* Montreal, QU, pages 411–418. M. Kaufmann, August 1995.

**19** P. Merz and B. Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8(2):1381–1231, 2002.

**20** P. E. Meyer, K. Kontos, F. Lafitte, and G. Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP Journal on Bioinformatics and Systems Biology*, Special Issue on Information-Theoretic Methods for Bioinformatics, 2007.

**21** P. E. Meyer, F. Lafitte, and G. Bontempi. Minet: An open source r/bioconductor package for mutual information based network inference. *BMC Bioinformatics*, 2008.

**22** P. E. Meyer, C. Schretter, and G. Bontempi. Information-theoretic feature selection using variable complementarity. *IEEE Journal of Special Topics in Signal Processing*, 2(3), 2008.

**23** B. M. E. Moret and H. D. Shapiro. An empirical analysis of algorithms for constructing a minimum spanning tree. In Springer, editor, *Lecture Notes in Computer Science*, volume 519, 1991.

**24** R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

**25** C. Olsen, P. E. Meyer, and G. Bontempi. On the impact of missing values on transcriptional regulatory network inference based on mutual information. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009.

**26** J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

**27** H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

**28** C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 1948.

**29** T. Shimamura, S. Imoto, R. Yamaguchi, A. Fujita, M. Nagasaki, and S. Miyano. Recursive regularization for inferring gene networks from time-course gene expression profiles. *BMC Systems Biology*, 3, 2009.

**30** Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Proceedings of the AAAI'06 workshop on Evaluation Methods for Machine Learning*, 2006.

**31** P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2001.

**32** I. Tsamardinos, C. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. In *The 16th International FLAIRS Conference*, 2003.

**33** E. P. van Someren, L. F. A. Wessels, E. Backer, and M. J. T. Reinders. Genetic network modeling. *Pharmacogenomics*, 3(4):507–525, 2002.

**34** J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, 1990.

**35** J. Zhang and P. Spirtes. Detection of unfaithfulness and robust causal inference. *Minds Mach.*, 18(2):239–271, 2008.

**36** Xin Zhang, Chitta Baral, and Seungchan Kim. An algorithm to learn causal relations between genes from steady state data: Simulation and its application to melanoma dataset. *Artificial Intelligence in Medicine*, pages 524–534, 2005.

**37** W. Zhao, E. Serpedin, and E. R. Dougherty. Inferring connectivity of genetic regulatory networks using information-theoretic criteria. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2), 2008.