Determining the direction of causal influence in large probabilistic networks: a constraint-based approach

Liviu Badea [1]

Abstract. Determining the direction of *causal* influence from observational data only is essential in many applications, such as the reconstruction of genetic networks from microarray data. As opposed to many probabilistic network inference algorithms which were designed to induce just *statistical* models of the data, Conditional Independence (CI) based algorithms are theoretically able to infer true *causal* models from observational data only. But unfortunately, the small sample sizes available from current microarray experiments render the determination of causal direction highly inaccurate. Here we show that this essential aspect of CI-based algorithms can be significantly improved by double-checking certain key statistical tests and by reconciling potential inconsistencies using a simple constraint propagation scheme.

1 INTRODUCTION AND MOTIVATION

The problem of inferring the *structure* of *very large* probabilistic networks has recently received a significant boost due to bioinformatics applications, especially those dealing with the reconstruction of genetic networks from microarray data. While *temporal* gene expression data (produced for example in the study of developmental processes) contains enough causal information in the temporal data sequence to allow the reconstruction of the causal networks with a reasonable accuracy, microarray data from *steady states* of the cell (for example, associated to various disease states) has proved, as far as we know, intractable for current structure inference algorithms. The main difficulties are related to the very large numbers of variables (i.e. genes – of the order of hundreds to thousands), the presence of many hidden (latent) variables, the small sample sizes available (tens to a few hundreds), as well as to the tough requirement of reconstructing the true *causal* structure rather than just a statistically equivalent one.

As the small sample sizes available are not enough to *completely* determine the network, more sophisticated approaches using e.g. Bayesian model averaging [2] have been proposed to deal with network structure in a Bayesian manner, especially when there might be many models (usually exponentially many) with a non-negligible posterior. However, model averaging cannot deal with the large number of variables in microarray data.

On the other hand, state of the art scoring-based algorithms (either based on simple model selection or on model averaging) were designed to induce *statistical* models of the data, rather than true *causal* models. Thus, the edges induced by such algorithms and especially their orientations do not necessarily reflect (the direction of) causal influence, as several distinct causal models, which differ in edge orientation, can be statistically equivalent [4].

Conditional-independence (CI) based algorithms [4,6] infer entire *equivalence classes* of graph models, thereby enabling a *causal interpretation of the resulting edge orientations*. (Edges with the same orientation in all statistically equivalent models represent true causal influences since the true causal model must be among the statistically equivalent models.) Even with small sample sizes and large numbers of variables (e.g. 73 samples and 1000 variables), we have been able to use such CI-based algorithms for recovering at least the most influential parts of given probabilistic networks. However, existing CI algorithms tend to be highly inaccurate in orienting edges (i.e. in determining the direction of causal influence) [6], especially in the case of few samples. In this paper we show that edge orientation can be significantly improved by double-checking certain key statistical tests and by reconciling potential inconsistencies using a simple *constraint propagation* scheme.

2 AN IMPROVED CONSTRAINT-BASED ALGORITHM

Conditional-independence based algorithms like IC* of Pearl and Verma [4] or the more efficient Fast Causal Inference (FCI) algorithm of Spirtes et al. [6] start with a completely connected network and simply use conditional independence (CI) tests to find separators for edges representing indirect influences. Finally, edge endpoints are oriented based on the separators found.

Although IC* and FCI are very close to the requirements of our bioinformatics application domain, they still have certain important drawbacks: as they construct causal structures by *categorical inference* based on the results of conditional independence tests, they are sensitive to the high amount of noise in the microarray data as well as to the small sample sizes.

In the following we show how CI-based methods (and especially their edge orientation phase) can be made more robust when dealing with small and noisy samples. Since the small sample size may support several potentially conflicting models, we provide means for coping with such inconsistencies by strengthening the collider and non-collider tests of FCI while preserving their efficiency, and by eliminating the remaining inconsistencies (anomalies) as well as all the features inferred from these.

We refer to [4] for the basic notions on Bayesian networks. The output of our QFCI algorithm described below will be a *Partial Ancestral Graph* (PAG), which is a concise representation of an entire equivalence class of graph models. Unlike standard PAGs, ours have confidence factors attached to the undirected edges, as well as to directed edge endpoints.

[1] AI group, National Institute for Research and Development in Informatics, 8-10 Averescu Blvd. Bucharest, Romania. E-mail: *badea@ici.ro*. The author is grateful to ECCAI for receiving an ECAI-2004 travel award.

In the following, we use the notations of [6] for describing PAGs. Briefly, edges can have three kinds of *endpoints* in a PAG: '–', '>' and 'o'. We also use the additional meta-symbol '*' that stands for any of the three kinds of endpoints. An '–' endpoint at Y for an edge X *— Y denotes the fact that Y is an *ancestor* of X in every graph of the equivalence class represented by the PAG, while an '>' endpoint at Y for X *–> Y means that Y is *not* an ancestor of X. Finally, an 'o' endpoint places no restriction on the ancestor relationships. (See [6] for more details.)

A *collider* is a structure of the form X *–> Y <–* Z. A collider is called *unshielded* iff X and Z are not adjacent in the PAG.

In the following, we present a constraint-based causal inference algorithm, QFCI, which aims at improving the robustness of the FCI algorithm in the face of noise and small sample sizes.

Employing a two-valued logic for combining the results of conditional independence tests in noisy domains may lead to inconsistencies, or *anomalies*. In fact, we have observed the occurrence of anomalies not only in microarray datasets (such as the Garber lung carcinoma study [3], the *Rosetta Compendium* of yeast microarray experiments and the *Spellman yeast cell cycle data*), but also in synthetic data. The most important type of anomaly observed was a so-called "*collider anomaly*", which is due to the inconsistencies between different colliders at a given node Y.

Recall that FCI recognizes colliders as follows: for non-adjacent X and Z, X*–*Y*–*Z is a *collider* iff $Y \notin \mathrm{Sep}(X,Z)$, where $\mathrm{Sep}(X,Z)$ is the first separating set found for X and Z: $X \perp Z \mid \mathrm{Sep}(X,Z)$.

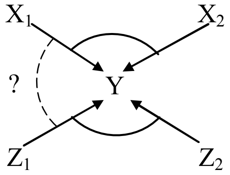
Definition (collider anomaly) Two unshielded colliders detected by the FCI algorithm

X_1 *–> Y <–* X_2 (for which $Y \notin \mathrm{Sep}(X_1,X_2)$) and

Z_1 *–> Y <–* Z_2 (for which $Y \notin \mathrm{Sep}(Z_1,Z_2)$)

are *inconsistent* w.r.t. the current set of separators Sep (or short, Sep-*inconsistent*) iff $\exists i,j \in \{1,2\}$ such that X_i and Z_j are not adjacent and X_i *–> Y <–* Z_j is not a collider w.r.t. Sep, i.e. $Y \in \mathrm{Sep}(X_i,Z_j)$.

As can be seen in the following Figure, a collider anomaly appears whenever a pair of arrowheads from different colliders (such as X_1 *–> Y <–* Z_1) doesn't form a collider according to Sep.



Example. An example of a collider anomaly (in a dataset of size 1000 sampled from a synthetic network with 40 variables and 35 edges) involves the colliders *X7**–>*X32*<–**X22* (*X32* $\notin$ Sep(*X7*,*X22*)=∅) and *X7**–>*X32*<–**X36* (*X32* $\notin$ Sep(*X7*,*X36*)={*X39*}) for which *X22* *–> *X32* <–* X36 is not a collider w.r.t. Sep (since *X32* ∈ Sep(*X22*,*X36*) ={*X32*}).

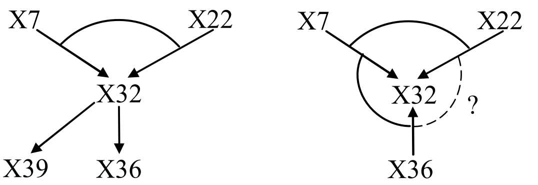


Figure 2. (a) The true graph (b) The collider anomaly

In other words, we have to place an arrow *X22* *–> *X32* (because *X7* *–> *X32* <–* *X22* is a collider w.r.t. Sep) and an arrow *X36* *–> *X32* (since *X7* *–> *X32* <–* *X36* is also a collider w.r.t. Sep), but these two arrows are inconsistent since *X32* ∈ Sep(*X22*,*X36*).

As can be seen by looking at the true graph in Figure 2(a), the inconsistency was due in this case to wrongly recognizing *X7* *–> *X32* <–* *X36* as a collider based on Sep(*X7*,*X36*)={*X39*} which does not contain *X32*. The fact that Sep records only a single separator set (among potentially many others) makes the collider recognition rule of FCI sensitive to errors in the independence test. In this specific case, the error in Sep(*X7*,*X36*) was due to a type-II error in the test $X7 \perp X36 \mid X39$, which succeeded (p-value=0.728 > α=0.05, for N=1000) despite the fact that *X39* does *not* d-separate *X7* from *X36*.

Since in the presence of many variables it would be very inefficient to recompute all the separators of *X7* and *X36*, we strengthen the FCI collider test by *double checking* whether adding *X32* to the current separator Sep(*X7*,*X36*) makes *X7* and *X36* dependent: $X7 \not\perp X36 \mid X39,X32$. (If *X32* were a true collider, conditioning on it would d-connect *X7* and *X36*.) If however, *X7* and *X36* remain independent, we cannot safely declare *X32* a collider.

Definition (strong collider test) For X *–* Y *–* Z, Y passes the *strong collider test* iff $Y \notin \mathrm{Sep}(X,Z)$ and $X \not\perp Z \mid \mathrm{Sep}(X,Z) \cup \{Y\}$, while Y passes the *strong non-collider test* iff $Y \in \mathrm{Sep}(X,Z)$ and $X \not\perp Z \mid \mathrm{Sep}(X,Z) \setminus \{Y\}$.

The strong non-collider test is dual to the strong collider test: we double check whether removing Y from the separator $\mathrm{Sep}(X,Z)$ makes X and Z dependent (as it should if Y were not a collider). If it doesn't, we refrain from declaring Y a non-collider.

Collider anomalies that are removed by the stronger definition of (non)collider are called *reducible*. The others are called irreducible.

Definition (irreducible collider anomaly) An *irreducible collider anomaly* is a pair of strong colliders X_1 *–> Y <–* X_2 and Z_1 *–> Y <–* Z_2 such that X_i *–> Y <–* Z_j is a strong non-collider for some $i,j \in \{1,2\}$.

Our constraint-based algorithm QFCI works as follows.

QFCI

1. Initialize the undirected graph by computing unconditional independencies

```
start with an empty PAG
for all pairs of variables $X,Y$
    perform the unconditional independence test $X \perp Y$ and set
    $p_u(X,Y)$ to its p-value[1] and $p(X,Y) = p_u(X,Y)$[2]
    if $p_u(X,Y) < \alpha$ (the test failed w.r.t. the significance level $\alpha$)
        add an undirected edge $X$ o–o $Y$ to the PAG
    else ($p_u(X,Y) \geq \alpha$, i.e. the test succeeded)
        set $\mathrm{Sep}(X,Y) = \emptyset$
```

2. Refine the undirected graph by conditional independence tests

```
for $k = 1..k_{\max}$ (consider conditioning sets of increasing size)
```

[1] $p_u(X,Y)$ will be used later to quantify the degree of *unconditional* correlation of X with Y.

[2] $p(X,Y)$ will be the largest p-value of a *conditional* independence test performed so far on X and Y: $p(X,Y) = max_S$ p_value($X \perp Y \mid S$). We use $p(X,Y)$ to quantify our confidence in the undirected edge X*–*Y.

```
  for all undirected edges $X$ o–o $Y$ (in decreasing order of their
  labels $p_u(X,Y)$, i.e. increasing order of the associated
  unconditional correlations)
    let $N$ = neighbors($X$) $\cup$ neighbors($Y$) [3]
    if $|N| \geq k$
      for all subsets $S \subseteq N$ of size $k$ (constructed by adding $k$
      nodes $Z \in N$ to $S$ in increasing order of their minimal p-
      labels[4] $min\{p_u(Z,X), p_u(Z,Y)\}$)
        perform the conditional independence test $X \perp Y \mid S$
        and let $p$ be its p-value
        if $p \geq \alpha$ (the test succeeded, i.e. $S$ is a separator)
          delete the undirected edge $X$ o–o $Y$
          set Sep($X,Y$) = $S$ and $p(X,Y) = p$
          break
        else if $p > p(X,Y)$ then set $p(X,Y) = p$
          (i.e. set $p(X,Y)$ to the maximal p-value of the
          $X \perp Y \mid S$ tests performed so far)
3. Search for potential colliders and non-colliders
for all variables $Y$
  for all pairs $X,Z$ of non-adjacent neighbors of $Y$
    if $X$ *–* $Y$ *–* $Z$ passes the strong collider test
      add the positive assertion $X$ *–> $Y \wedge Z$ *–> $Y$ : $cf$
      with confidence factor
      $cf = p(X,Z)(1-p_d)(1-p(X,Y))(1-p(Y,Z))$, where
      $p_d$ = p_value($X \perp Z \mid$ Sep($X,Z$) $\cup \{Y\}$) $< \alpha$ is the p-value
      of the failed independence test performed during the
      strong collider test[5]
    else if $X$*–*$Y$*–*$Z$ passes the strong non-collider test
      add the negative assertion $\neg$ ($X$ *–> $Y \wedge Z$ *–> $Y$) : $cf$
      with confidence factor
      $cf = p(X,Z)(1-p_d)(1-p(X,Y))(1-p(Y,Z))$, where
      $p_d$ = p_value($X \perp Z \mid$ Sep($X,Z$) $\setminus \{Y\}$) $< \alpha$ is the p-value
      of the failed independence test performed during the
      strong non-collider test
4. Eliminate collider anomalies
for all pairs of positive assertions
  $X_1$ *–> $Y \wedge X_2$ *–> $Y$ : $cf_1$  and  $Z_1$ *–> $Y \wedge Z_2$ *–> $Y$ : $cf_2$
  if there exists a negative assertion
    $\neg$( $X_i$ *–> $Y \wedge Z_j$ *–> $Y$ ) : $cf$ for some $i,j \in \{1,2\}$
    remove these positive and negative assertions
5. Constraint propagation of assertions
repeat
  propagate assertions (using the propagation rules below)
until no more propagations are possible
remove potential inconsistencies
```

The worst-case *complexity* of the algorithm is exponential in the number of variables n, because in principle it has to consider all subsets of variables as conditioning sets (there are $2^{n-2} \cdot n(n-1)/2$ such subsets). Fortunately however, genetic networks typically have small in- and out-degrees, so that searching for separating subsets S in increasing order of their size $|S|$ will avoid many unnecessary (and unreliable) CI tests. Thus, in practice the run-time is dominated by the independence tests conditional on size 1 subsets.

A further heuristic, but very effective improvement restricts the search for separator subsets S among the direct neighbors of the nodes to be separated. Thus, since we initially start with a completely connected graph, it is essential to reduce the number of direct neighbors of nodes as quickly as possible. This is achieved by our ordering heuristic which tries to separate the pairs of variables (X,Y) in increasing order of their unconditional correlation $|r_u(X,Y)|$. This heuristic assumes that (unconditionally) less correlated variables will be easier to separate conditionally. Scheduling independence tests that are more likely to succeed earlier reduces node neighborhoods as quickly as possible, thereby reducing the number of candidate neighbors in the later phases.

Quantitative information is also used in phase 2 when exploring potential separator sets S for a pair of nodes (X,Y). Variables Z with a higher (unconditional) correlation with one of X or Y are more likely to be true neighbors (as opposed to just temporary neighbors at this stage of the algorithm[6]) and are selected with priority as members of S.

The search for colliders in phase 3 employs the strong collider and non-collider tests. But since even these stricter tests may not eliminate all collider anomalies, we need to explicitly remove the colliders involved in such anomalies.

To allow a more precise evaluation of the results, the discovery of potential colliders and non-colliders produces assertions labeled by *confidence factors* (based on quantitative information from the independence tests).

Definition (assertions) *Assertions* can be either *positive*

X *–> $Y \wedge Z$ *–> Y : cf (p2)

X *–> Y : cf (p1)

or *negative*

$\neg$ (X *–> $Y \wedge Z$ *–> Y) : cf (n2)

$\neg$ X *–> Y : cf (n1)

Assertions of the form (p2), (p1), or (n1) are called *definite*, while those of the form (n2) are called *disjunctive* (since they are equivalent to $\neg$ X *–> Y $\vee$ $\neg$ Z *–> Y : cf).

A positive assertion of the form (p2) means that we are confident with degree cf that both arrowheads at Y (X *–> Y and Z *–> Y) should appear in the partial graph. A negative assertion of the form (n2) means that the arrowheads X *–> Y and Z *–> Y cannot both appear in the partial graph.

Collider anomalies are inconsistencies in the assertions. Under the usual assumptions (such as faithfulness and the representability of the observed JPD by a single graph model), the most likely explanation for such inconsistencies is the small sample size, which cannot exclude several potentially conflicting models.

While some anomalies disappear when using our stronger (non)collider test, the remaining irreducible ones need to be eliminated by removing the conflicting assertions (phase 4).

The remaining assertions, which are now guaranteed to be consistent, are subsequently propagated in phase 5.

Propagation (for example of Z *–> Y with $\neg$(X *–>$Y \wedge Z$ *–> Y)) can produce definite (unary) negative assertions of the form $\neg$ X *–> Y, which can be automatically converted to X *— Y (recall that an '>' arrowhead into Y means that Y is *not* an ancestor of X,

[3] For simplicity, we do not reproduce here the more complex determination of a complete set of candidate separators used in FCI (based on *Possible-D-Sep*), which might not be reliable for small sample sizes.

[4] i.e. in decreasing order of their maximal unconditional correlations $max\{|r_u(Z,X)|, |r_u(Z,Y)|\}$.

[5] Note that $p(X,Z)$ = p_value($X \perp Z \mid$ Sep(X,Z)) $\geq \alpha$ and $p(X,Y) = max_S$ p_value($X \perp Y \mid S$) $< \alpha$. (Similarly, $p(Y,Z) < \alpha$.)

[6] Recall that initially, nodes may be connected to many more other nodes than their direct neighbors.

while an '–' endpoint says that *Y is* an ancestor of *X*). But in the absence of hidden *selection* variables, we cannot have edges with '–' endpoints at both ends, so X *— Y could be immediately turned into X <— Y. Unfortunately, placing new '<' arrowheads may lead to new inconsistencies, [7] for example involving *U* *–> *X* <— *Y* and the negative assertion (non-collider) ¬(*U* *–> *X* <–* *Y*). To make things even more complicated, the arrow *X* <— *Y* may propagate another arrow, for example *V* **<—** *X* <— *Y before* the discovery of the inconsistency with *U* *–> *X*.

Using the terminology of non-monotonic logics, we adopt a "*skeptical*" attitude towards inferring new edge orientations, which amounts to withholding from propagating an arrowhead that may be involved in a conflict with another one. As all assertions involved in such potential inconsistencies must be eliminated, we have to keep track of the inferences (propagations) made from these assertions, in order to enable their subsequent removal. In our previous example, removing the arrowhead at *X* in *X* <— *Y* will have to invalidate the *V* <— *X* arrow as well (of course, only if *V* <— *X* has no other "justification").

More generally, we attach a "justification" to each assertion, representing the successive insertions of arrowheads (for avoiding *X* —— *Y* edges) that have lead to placing the current arrowhead.

Definition (justification of an assertion) The *justification of a primitive assertion* (i.e. an assertion generated in phase 3 and based on CI tests) is empty. The justification of a *derived assertion* (i.e. an assertion *propagated* in phase 5) is a set of atomic labels $j = \{l_1, l_2, ..., l_n\}$ representing arrowheads placed for avoiding *X* —— *Y* edges.

We use the notation $A : cf :: j$ for an assertion *A* with justification *j* (empty justifications can be omitted).

An ATMS could be used to manage assertions and their justifications. But the *propagation rules* in our domain are very simple due to the very constrained form of assertions (in the following, *a* and *b* stand for edge arrowheads of the form *X* *–> *Y*):

$$\begin{aligned} a \wedge b : cf \quad &\Rightarrow \quad a : cf,\ \ b : cf \\ a : cf_a :: j_a,\ \neg(a \wedge b) : cf \quad &\Rightarrow \quad \neg b : cf_a \cdot cf :: j_a \\ \neg(X *\!\!-\!\!> Y) : cf :: j \quad &\Rightarrow \quad Y -\!\!-\!\!> X : cf :: j \cup \{l_i\} \end{aligned}$$

(with l_i a new atomic label)

An *arrowhead inconsistency* is treated by the rule:

$$a :: j_1,\ \neg a :: j_2 \quad \Rightarrow \quad \text{remove\_inconsistency}(j_1, j_2)$$

which deletes all assertions $A :: j$ with justifications containing j_1 or j_2: $j \supseteq j_1$ (but only if $j_1 \neq \emptyset$) or $j \supseteq j_2$ (but only if $j_2 \neq \emptyset$).

Finally, after all potential inconsistencies have been removed, we aggregate the confidence factors for edge endpoints as follows (since a given edge endpoint can be supported by several assertions with different confidence factors): $cf(a) = max\{\ cf_i \mid a : cf_i\ \}$. (Assertions with confidence factors below a given threshold, e.g. α=0.05, are automatically discarded.)

Note that the rule that orients edges for avoiding the formation of new colliders (rule R1 in [4], or rule G(ii) in [6]):

$$X *\!\!-\!\!> Y *\!\!-\!\!* Z \ \Rightarrow\ X *\!\!-\!\!> Y -\!\!-\!\!> Z$$

is a special case of our propagation of assertions (phase 5).

Also note that we do not apply the acyclicity rule (R2 from [4], or G(i) from [6]), since genetic networks are potentially cyclic. However, dealing with both cycles and latent variables is an open research problem, so we do not aim at completeness in the presence of cycles [5].

3 EVALUATION

We evaluated QFCI on synthetic datasets similar to the extreme conditions encountered in real-life microarray datasets, such as the Garber lung cancer data [3]. The main problem is related to the low power of conditional independence tests for such small sample sizes (*N*=73), for which it is generally impossible to discriminate between true but *weak* dependencies and nonzero fluctuations of correlations of otherwise independent variables. Thus, although we cannot expect to obtain a perfect model of the data, we would still like to recover at least the stronger dependencies in the data, while minimizing the number of wrong edges.

In the following we compare QFCI with the original FCI algorithm from the TETRAD IV distribution [8] and WinMine [9] on datasets of size 73 (the same as in [3]) sampled from synthetic linear models generated from Erdos-Renyi random graphs with 850 nodes and 2000 edges (corresponding to a biologically plausible average degree of 4.7).

Since expression levels in microarray measurements are continuous variables, we have chosen to employ CI tests based on the Fisher z transform of partial correlations. Strictly speaking, these tests are only correct if the variables are jointly normal, but they are still often useful for non-normal distributions as well. The alternative of discretizing the variables seems worse, especially in view of the small sample sizes, as it would further reduce the power of the tests. (The majority of scoring-based implementations use discretization of the continuous variables.)

The following Table presents the results of FCI, QFCI and WinMine using their default parameters (FCI and QFCI have a single parameter, α=0.05),[8] except for WinMine, which was run in the 'acyclic' mode. The Table contains the averages and standard deviations of the following *edge counts* corresponding to 5 runs of the algorithms on 5 different Erdos-Renyi random graphs with 850 nodes (variables) and approximately 2000 edges:

MEAN (STD) 5 networks	FCI (Tetrad)	QFCI & Tetrad	QFCI	QFCI & WinMine	WinMine
Original edges (c+m)	1995.6 (3.3)		1995.6 (3.3)		1995.6 (3.3)
Total induced edges (e)	828.4 (13.7)		824.2 (16.8)		624.6 (23.9)
Correct skeleton (c)	707 (25.6)	685.4 (27.2)	710.2 (30.6)	422.2 (16.6)	473.4 (11.6)
c/e	85.4% (1.9)		86.2% (1.8)		75.8% (0.5)
Compatible orientation (o)	423.8 (28.2)	405 (29.5)	663.6 (20.6)	221.8 (6.5)	267.4 (14.3)
o/c	**59.9%** (1.1)		**93.4%** (0.7)		**56.5%** (1.2)
-->	115.4 (17.4)		57.6 (9.1)		267.4 (14.3)
o->	134 (12.6)		242 (21.8)		
o-o	174.4 (14.9)		364 (13.4)		
incompabible orientation (i)	283.2 (29.5)	35.2 (13.3)	46.6 (15.1)	9.6 (4.3)	206 (15.6)
i/c	**40.1%** (1.2)		**6.6%** (0.5)		**43.5%** (1.3)
-->	35.8 (10.3)		6.4 (4.8)		206 (15.6)
o->	60.6 (6.0)		21 (8.2)		
<->	181.6 (21.1)		19.2 (5.0)		
---	5.2 (1.5)				
Wrong skeleton (w)	120.8 (19.3)	87.6 (17.3)	114 (19.5)	10 (4.1)	151.2 (26.5)
w/e	14.6% (1.4)		13.8% (1.2)		24.2% (1.1)
Missing edges (m)	1288.6 (25.1)		1285.4 (30)		1522.2 (13.5)
m/(c+m)	64.6% (0.5)		64.4% (0.5)		76.3% (0.5)

- *e*: the total number of induced edges

[7] of course, only in the case of unreliable CI tests.

[8] We have performed experiments on many more random graphs with various parameter settings and obtained similar results for all of these.

- *c*: the number of edges which are *correct* if we disregard their orientation ("correct skeleton")
- *w*: the number of *wrong* edges induced by the algorithms (induced edges which do not occur in the original model)
- *m*: the number of edges from the original model that are *missing* from the induced graph
- *o*: edges having an orientation *compatible* with the original edge (for example, o–> is compatible with —>, but not with <—)
- *i*: the number of edges which appear in the original graph, but only with an *incompatible* orientation.

The columns marked "QFCI & Tetrad" ("QFCI & WinMine") contain the overlaps between QFCI and Tetrad (respectively between QFCI and WinMine) in the corresponding categories.

Given the very small sample size, the large number of missing edges is not surprising.[9] Nor is the fact that FCI and QFCI induce more or less the same "skeleton" (i.e. undirected graph[10]). However, even with respect to the skeleton, QFCI and the original FCI perform significantly better than WinMine (the proportion of wrong edges induced by WinMine is already larger than that of QFCI and FCI, while the number of correct edges is much lower 473 < 710).

The large number of missing edges is not a fatal drawback in our bioinformatics application, in which almost nothing is known about the circuitry of the genes involved. The discovery of even only the strongest dependencies is thus still a significant step forward.

The most important improvement of QFCI w.r.t. the other algorithms is obtained in the orientation of edges. Indeed, while only 59.9% of the correct FCI edges had also correct (or at least compatible) orientations, 93.4% of the edges induced by QFCI had a compatible orientation. WinMine orients edges almost at random.

The above Table also presents a more detailed breakdown of the correctly and wrongly oriented edges. Not very surprisingly, about half of the compatible edges remain completely unoriented (o–o). Our results confirm the observation of Spirtes et al. [6] that FCI's error rate is significantly higher w.r.t. edge orientations than regarding the skeleton (of course, FCI was not designed to cope with such small sample sizes). Thus, overall, QFCI seems to fare better than existing algorithms, especially w.r.t. edge orientation.

Note that the precise nature of our biological problem seems to favour a constraint-based approach over a scoring-based algorithm based on model selection. Indeed, a scoring-based method will keep adding edges until no further improvement of the score can be achieved. But this does not mean that the given data *necessarily* implies all the edges induced, nor that their orientations reflect the true causal directions of influence. On the other hand, a constraint-based algorithm like QFCI will return only edges that have a high probability of occurring in all alternative models of the given data. Such an approach may miss many weak dependencies (which cannot be discriminated from noise with such small sample sizes), but the edges returned tend to be correct. Furthermore, directed edges can be interpreted as causal influences.

4 RELATED WORK AND CONCLUSIONS

The approach in [7] also deals with improving the edge orientation phase of CI-based algorithms. The main difference w.r.t. our approach is that [7] uses a *context-dependent* relative scoring function for determining colliders (i.e. one that scores a collider based on the orientations of *all the other* edges in the graph), which only works well with a reasonably *complete* model and in the absence of *latent variables*. (This is why the algorithm in [7] reorients all edges anew after each single edge addition – this wouldn't have been necessary if edge orientations would not be very sensitive to the incompleteness of the intermediate models.) On the other hand, while [7] attempts to obtain *complete statistical* models, we are aiming at determining the directions of influences that are fully determined by the data (and thus admit a *causal* interpretation) *independently* of the other orientations (to avoid the propagation of errors) in highly *incomplete* models with potentially many latent variables. (Incompleteness is due to the few and noisy samples and to the much stronger requirement of identifying the *original* edges and their orientations rather than just a statistically equivalent model of the data. This is – in non-monotonic logic terminology – a "skeptical" approach to recognizing features of the model, which requires, among others, *context-independent* collider recognition.)

The BNPowerConstructor [1] also employs quantitative information related to the independence tests in a constraint-based algorithm. However, extending it to deal with latent variables or small samples seems extremely difficult, if not inherently impossible.

We have also applied QFCI on the Garber lung cancer dataset [3] and obtained very encouraging results from a biological viewpoint. The Supplemental Figure online at http://www.ai.ici.ro/ecai04/SupplFig.pdf depicts the neighbours (up to depth 5) of the discrete variable associated to the '*small cell*' lung cancer subtype. It is particularly striking that QFCI finds 'small cell' connected to a single gene of unknown function, INSM1 (insulinoma-associated 1), which is known to serve as a marker for lung tumours of neuroendocrine differentiation. Moreover, all of the genes hand-picked by human experts in Garber et al. [3] in their discussion of the 'small cell' subtype are very close neighbors in our network: 7B2 (SGNE1), glutaminyl cyclase (QPCT), L-myc (Hs.92137) and the neuronal differentiation marker achaete-scute homolog (IMAGE:1416420), while several others appearing in our network have unknown functions and should be further investigated. The network obtained could represent a good starting point for elucidating the details of the genetic networks involved in lung carcinoma by enabling much better targeted experiments.

REFERENCES

1. Cheng J, Bell D, Liu W. Learning Bayesian networks from data: an efficient approach based on information theory. Proc. CIKM-97, 325.
2. Madigan D, York J. Bayesian graphical models for discrete data. Internat. Statist. Rev 63, 215-232, 1995.
3. Garber M.E. et al. Diversity of gene expression in adenocarcinoma of the lung. PNAS 98, 13784-9, Nov 20, 2001.
4. Pearl J. Causality: Models, Reasoning, and Inference, CUP 2000.
5. Richardson T, Spirtes P. Automated discovery of linear feedback models, Technical Report CMU-75-Phil.
6. Spirtes P, Glymour C, Scheines R. Causation, Prediction, and Search, MIT Press, 2001.
7. Steck H. On the use of skeletons when learning in bayesian networks. Proc. UAI-2000, 558-65.
8. TETRAD IV. http://www.phil.cmu.edu/projects/tetrad/tetrad4.html
9. WinMine. http://research.microsoft.com/~dmax/winmine/tooldoc.htm

[9] A more in-depth analysis revealed that these tend to be edges $x \rightarrow y$ corresponding to parents x with small influence on y in the given dataset.

[10] Refining the skeleton inference is not the main concern of this paper.