Greedy Fast Causal Interference (GFCI) Algorithm for Discrete Variables

This document provides a brief overview of the GFCI algorithm, focusing on a version of GFCI that works with discrete variables, which is called GFCI-discrete (GFCId).

Purpose

GFCId is an algorithm that takes as input a dataset of discrete variables and has two phases. The first phase greedily searches over selected causal Bayesian network (CBN) structures[1] (models), and outputs the highest scoring model it finds under the assumption that there are no unmeasured confounders and selection bias. This output is then input into a slight modification of the Fast Causal Inference (FCI) algorithm, which post-processes the output to produce a representation of a set of models that may include unmeasured confounders. The model that GFCId returns serves as a data-supported hypothesis about causal relationships that exist among the variables in the dataset. Such models are intended to help scientists form hypotheses and guide the design of experiments to investigate these hypotheses.

Methodological Approach

The first phase of GFCI runs FGES, which is an optimized and parallelized version [Ramsey, 2015] of an algorithm developed by Meek [Meek, 1997] called the Greedy Equivalence Search (GES). GES outputs a set of models that do not contain unmeasured confounders or selection bias. The algorithm was further developed and studied by Chickering [Chickering, 2002]. GES is a Bayesian algorithm that heuristically searches the space of CBNs and returns the model with highest score it finds. In particular, GES starts its search with the empty graph. It then performs a forward stepping search in which edges are added between nodes in order to increase the Bayesian score. This process continues until no single edge addition increases the score. Finally, it performs a backward stepping search that removes edges until no single edge removal can increase the score. FGES uses the BDeu[2] scoring measure, which is described in detail in [Heckerman, 1995].

The second phase of GFCI uses the output of FGS as input to a slight modification of the Fast Causal Inference (FCI) algorithm, which outputs a representation of a set of models that may contain unmeasured confounders. The FCI algorithm was developed by Spirtes, Glymour, and Scheines [Spirtes et al. 1993] and has two phases. FCI starts from an undirected graph that contains a superset of the adjacencies in its final output (in the case of GFCI the adjacencies in the output of the FGES phase), and then searches for conditional independence relations between pairs of variables that are adjacent; if it finds a conditioning set that makes a pair of adjacent variables independent, it removes the adjacency. After the adjacency phase, FCI uses properties of the conditioning sets that led to the removal of edges in order to orient as much as possible the remaining adjacencies. In the GFCI version of FCI, it also supplements the

---

[1] A CBN structure is a directed acyclic graph in which nodes represent variables and arcs represent direct causation among the nodes, where the meaning of *direct* is relative to the nodes in the CBN. For further information about CBNs, see [Spirtes, 2010; Lagani, 2016; Pearl 2016].

[2] BDeu stands for **B**ayesian **D**irichlet likelihood **e**quivalence and **u**niform. It is based on assuming a Dirichlet parameter prior probability and a multinomial likelihood. It uses Dirichlet parameter priors that guarantee that CBNs that represent the same dependence and independence relationships among the variables (by way of d-separation) are assigned the same score.

orientations found by using conditional independence tests with some (but not all) of the orientations that were found in the output of the FGES phase to orient edges in its output.

Input Data and Parameters

GFCId has the following requirements for data input:

- the (training) data are in a table in which columns represent variables, rows represent samples, and the value of each variable in a sample is discrete.
- the first row of the table lists the variable names, in order and unique; the data and variable names are separated by a delimiter (default: tab).
- no values for any samples are missing.

GFCId takes the following parameters, which modify the behavior of the algorithm:

- max-degree - The maximum degree of any node in the graph. Smaller values will reduce search time. The default is 100, which is very large.
- max-path-length - The maximum length for any discriminating path. Limiting path length will reduce search time. The default is -1 indicating unlimited path lengths.
- faithfulness-assumed - Using this flag indicates that (one edge) faithfulness should be assumed. Simulation results indicate that assuming faithfulness has little or no negative effect on precision-recall performance and leads to a marked decrease in runtime. The default is that faithfulness-assumed is set to true.
- structure-prior - For each node in a CBN, it provides the following prior probability that the node has a given set of parents:

$$\left(\frac{e}{v-1}\right)^{p} \cdot \left(1-\frac{e}{v-1}\right)^{v-p-1}$$

  where *v* is the number of variables in the CBN, *p* is the number of parents of a given child node in a particular CBN, and *e* is a global parameter that is approximately equal to the expected number of parents of the nodes in the CBN. By default we use e = 1. The structure prior of an entire network is equal to the product over the structure priors of each node in the network.
- sample-prior - a real valued parameter that specifies the parameter (probability) priors in the CBNs searched by GFCId. GFCId uses the BDeu scoring measure, for which the expectations of the prior probabilities are uniform. The sample-prior indicates how confident we are that these expectations are indeed uniform; the larger the sample-prior, the more confident we are. By default, sample-prior = 1, which reflects weak confidence that the probabilities in the data-generating CBN are uniform; such weak parameter priors allow the data to more strongly influence the CBN that is found by GFCId search.
- knowledge - the user may specify background knowledge by providing a file that describes variable precedence and required and/or forbidden edges in the CBN structure. By default, the algorithm assumes no prior knowledge about the CBN structure. The format of the prior knowledge file can be found at https://bd2kccd.github.io/docs/causal-cmd/ under Sample Prior Knowledge File.

- exclude-variables - the user may specify which variables to exclude from the dataset by using this switch to point to a file that contains the name of a variable in each row.
- thread - by default the algorithm will run in a parallel fashion using as many threads as are needed and available on the system. The user has the option to specify a smaller number of threads.
- alpha – the significance level of the independence tests performed by GFCI. The default is 0.01.

Output

GFCId outputs a set of CBNs represented by a graphical object called a **PAG** [Spirtes et al. 2000, Zhang 2008], which is a generalization of a pattern, and represents a set of indistinguishable CBNs which may contain unmeasured confounders. Each CBN represented by the output PAG entails (as closely as the algorithm could determine) the set of conditional independence relations judged to hold in the dataset. The appendix provides an introduction to PAGs and how to interpret them causally.

Algorithmic Assumptions

This section describes a sufficient set of assumptions for the application of GFCId to achieve the guarantees described in the next section. While the pattern output by GFCId may still include correct edges (and perhaps many correct edges) even if one or more of these assumptions are violated, there are no theoretical guarantees it will do so.

A sufficient set of conditions for recovering the causal structure of the data-generating process in the large sample limit (i.e., as the sample size grows without bound) is as follows: Assume that the causal process generating the data *D* given to GFCId is accurately modeled by a CBN containing only discrete variables, some of which may not be measured, which we call ***G***. Assume that each variable (node) in ***G*** is a function of its parents that is modeled by a multinomial probability distribution. These conditions are sufficient to recover the causal structure.

While the above procedure is simple, it includes several assumptions that may not be obvious. Key among them are the following:

- cases (samples) in the data *D* are independent and identically distributed.
- the causal Markov condition holds [Spirtes, 2010]. This condition states that a variable is independent of its non-effects, given its direct causes (parents). It expresses a form of local causality.
- the causal faithfulness condition holds with probability 1 [Spirtes, 2010]. This condition states that all the independence relationships among the measured variables are implied by the causal Markov condition.
- there is no selection bias. This means that the chance a case (sample) was selected from the population for inclusion in dataset *D* did not depend on the values of any of the measured variables in the data.
- there are no feedback cycles among the measured variables. Extensions to CBNs, such as causal Dynamic Bayesian Networks (DBNs) [Neapolitan, 2003], do allow feedback cycles, but they are not currently implemented in GFCId.

Structure Learning Performance Guarantees

If the assumptions in the previous section hold, then in the large sample limit, the CBN structure output by GFCId will contain an edge of one of four kinds between *X* and *Y* if and only if *X* and *Y* are not independent conditional on any subset of the other measured variables of less than or equal to a specified size. In addition, there is (1) an arc $X \rightarrow Y$ if and only if *X* directly or indirectly causes *Y*, and *Y* does not directly or indirectly cause *X*; (2) an edge $X \leftrightarrow Y$ if and only if *X* is not a direct or indirect cause of *Y* and *Y* is not a direct or indirect cause of *X* (which can only occur if there are latent confounders of *X* and some other variable or *Y* and some other variable; (3) an edge $X \text{ o}\!\!\rightarrow Y$ only if *Y* is not a direct or indirect cause of *X*, but *X* may or may not be an indirect cause of *Y*; (4) an edge *X* o–o *Y* indicates that *X* and *Y* are dependent no matter what subset of observed variables is conditioned on, but contains no orientation information (*X* may be a direct or indirect cause of *Y*, and *Y* may be an indirect cause of *X*, or there may be a latent common cause of *X* and *Y*.)

Practice Dataset

We used the CBN shown in Figure 1 and Table 1 to generate the simulated data shown in Table 2. The user may wish to apply GFCId (with an alpha value of 0.01, the sample prior set to 5, and the structure prior set to 1) to the dataset in Table 2 and verify that the CBN structure obtained is the one shown in Figure 1.

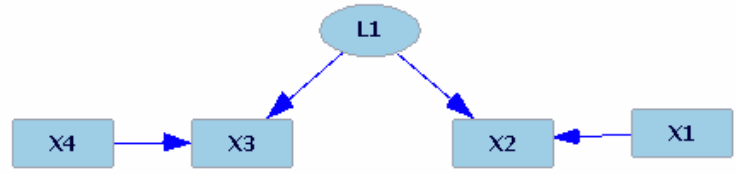

Figure 1. The CBN structure used to generated the practice dataset. X1-X4 are measured variables and L1 is a latent variable.

Table 1. Probabilities of the CBN used to generate the practice dataset. All variables are Boolean, represented by a value of 0 or 1.

$P(L1=0)$	0.4
$P(X1=0)$	0.6
$P(X4=0)$	0.5
$P(X2=0 \mid L1=0, X1=0)$	0.1
$P(X2=0 \mid L1=0, X1=1)$	0.3
$P(X2=0 \mid L1=1, X1=0)$	0.4
$P(X2=0 \mid L1=1, X1=1)$	0.8
$P(X3=0 \mid L1=0, X4=0)$	0.1
$P(X3=0 \mid L1=0, X4=1)$	0.4
$P(X3=0 \mid L1=1, X4=0)$	0.3

$P(X3=0 \mid L1=1, X4=1)$	0.8

Table 2. The practice dataset generated from the network structure and parameters in Figure 1 and Table 1. The *MULT* column indicates how many instances of a given row we provided to the GFCId algorithm.

X_1	X_2	X_3	X_4	*Mult*
0	0	0	0	151
0	0	0	1	370
0	0	1	0	320
0	0	1	1	140
0	1	0	0	489
0	1	0	1	1009
0	1	1	0	1308
0	1	1	1	782
1	0	0	0	284
1	0	0	1	665
1	0	1	0	624
1	0	1	1	309
1	1	0	0	106
1	1	0	1	233
1	1	1	0	425
1	1	1	1	285
			total:	7,500

Performance on Simulated Data

We evaluated the performance of GFCId on additional simulated data using the Tetrad data simulator (Tetrad, 2016). We first created a random CBN with a given number of nodes and edges, which we call CBN_{gen}. The directed acyclic graph of CBN_{gen} was generated by fixing the number of vertices at 100, and then randomly adding 200 edges, as long as the added edges did not create a cycle. We then randomly sampled the distribution defined by CBN_{gen} to generate a set of training data *D*. We provided that data to GFCId to obtain the PAG that it output, which we call P_{out}. We then derived the PAG of CBN_{gen}, which we call P_{gen}. Thus, both P_{gen} and P_{out} are PAGs. We compared P_{gen} with P_{out} to derive the following statistics:

AP = Adjacency Precision
AR = Adjacency Recall
AHP = Arrowhead precision
AHR = Arrowhead recall
TP = Tail precision
TR = Tail recall
SHD = Structural Hamming Distance
Time = Elapsed Time in Seconds

Two nodes are considered to be adjacent if they have any edge type between them. Adjacency recall is the fraction of pairs of variables adjacent in P_{gen} that are also adjacent in P_{out}. Adjacency precision is the fraction of pairs of variables adjacent in P_{out} that are also adjacent in P_{gen}. Arrowhead recall is the fraction of “>” that appear in P_{gen} that also appear in P_{out}, and arrowhead precision is the fraction of “>” in P_{out} that also appear in P_{gen}. Tail precision and tail recall are defined analogously for the “–“ endpoint type. The Structural Hamming Distance is the

number of edges that would need to be changed to transform P_{out} into P_{gen}. U is a statistic that emphasizes precision over recall, which would be the case for many applications: It is defined as adjacency precision + arrowhead precision + tail precision + 0.5 * (adjacency recall + arrowhead recall + tail recall). We recorded the CPU time in seconds that GFCId took to derive P_{out} when using a MacBook Pro with 16 gigabytes of memory and 4 processors. All of the statistics in each table are averages over 10 repeats of the process just described, where each repeat is generally a different randomly generated CBN. We evaluated the following three algorithms:

GFCId: Greedy Fast Causal Inference (discrete) using a BDeu score in stage 1 and the Chi Square test in stage 2
RFCI: Really Fast Causal Inference using a Chi Square test
FCI: Fast Causal Inference using a Chi Square test

For all three algorithms, alpha was set to 0.01. For GFCId, max-degree was set to 4, the sample prior was set to 1, and the structure prior was set to 1.

FCI and RFCI are algorithms with assumptions, inputs, and outputs similar to GFCId, but are completely constraint-based algorithms. In all 6 simulations that we ran, GFCId performed better overall than did FCI and RFCI. In the tables below, the algorithms are ordered by the value of their U statistic.

Tables 3-5 show results for a data-generating CBN with 100 measured variables, 15 latent variables, an average node degree of four, 200 edges, and variables that had either 2 or 3 categories.

Table 3. Sample size 100

Alg	AP	AR	AHP	AHR	TP	TR	SHD	Time	U
GFCId	0.93	0.14	0.52	0.008	0.27	0.003	0.336.50	0.10	0.30
RFCI	0.85	0.14	0.23	0.03	0.00	0.00	0.134.50	0.14	0.19
FCI	0.74	0.16	0.23	0.04	0.00	0.00	0.134.80	0.25	0.18

Table 4. Sample size 500

Alg	AP	AR	AHP	AHR	TP	TR	SHD	Time	U
GFCId	0.97	0.31	0.92	0.09	0.79	0.05	131.40	0.18	0.48
RFCI	0.95	0.35	0.39	0.21	0.31	0.02	115.50	0.41	0.32
FCI	0.92	0.36	0.39	0.23	0.29	0.01	113.50	0.68	0.32

Table 5. Sample size 1000

Alg	AP	AR	AHP	AHR	TP	TR	SHD	Time	U
GFCId	0.96	0.37	0.89	0.23	0.65	0.18	1082.30	3.17	0.48
RFCI	0.90	0.40	0.42	0.31	0.35	0.02	966.70	58.15	0.34
FCI	0.86	0.41	0.41	0.32	0.38	0.03	959.70	51.97	0.34

Tables 6-8 show results for a data-generating CBN with 1000 measured variables, 150 latent variables, an average node degree of four, 2000 edges, and variables that had either 2 or 3 categories.

Table 6. Sample size 100

Alg	AP	AR	AHP	AHR	TP	TR	SHD	Time	U
GFCId	0.94	0.09	0.62	0.003	0.22	0.0003	1479.50	0.81	0.30
RFCI	0.81	0.11	0.22	0.02	0.10	.0001	1436.30	8.66	0.20
FCI	0.54	0.13	0.19	0.04	0.22	0.0006	1447.00	10.91	0.17

Table 7. Sample size 500

Alg	AP	AR	AHP	AHR	TP	TR	SHD	Time	U
GFCId	0.96	0.27	0.91	0.10	0.70	0.06	1294.00	1.78	0.46
RFCI	0.87	0.31	0.38	0.19	0.35	0.01	1144.10	29.90	0.31
FCI	0.79	0.32	0.36	0.22	0.33	0.01	1129.0	27.58	0.29

Table 8. Sample size 1000

Alg	AP	AR	AHP	AHR	TP	TR	SHD	Time	U
GFCId	0.96	0.37	0.89	0.23	0.65	0.18	1082.30	3.17	0.48
RFCI	0.90	0.40	0.42	0.31	0.35	0.02	966.70	58.15	0.34
FCI	0.86	0.41	0.41	0.32	0.38	0.03	959.70	51.97	0.34

The results in the tables provide benchmarks that may be helpful in estimating the performance of GFCId when it is applied to real datasets. We emphasize, however, that the results obtained with such simulated data may be better than those obtained with real datasets.

References

Chickering DM. Optimal structure identification with greedy search. *Journal of Machine Learning Research* 3 (2002) 507-554.
http://www.jmlr.org/papers/volume3/chickering02b/chickering02b.pdf

Heckerman D, Geiger D, Chickering M. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning (1995) 197-243.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.5090&rep=rep1&type=pdf

Lagani V, Triantafillou S, Ball G, Tegner J, Tsamardinos I. Probabilistic computational causal discovery for systems biology. *Uncertainty in Biology* 17 (2016) 33-73.
http://www.mensxmachina.org/files/publications/Probabilistic%20Causal%20Discovery%20for%20Systems%20Biology_prePrint.pdf

Meek C. Causal inference and causal explanation with background knowledge. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (1995) 403-410.
https://arxiv.org/ftp/arxiv/papers/1302/1302.4972.pdf

Meek C. *Graphical Models: Selecting Causal and Statistical Models*. Ph.D. thesis, Carnegie Mellon University (1997).

Neapolitan RE. *Learning Bayesian Networks* (Pearson, 2003).

Pearl J, Glymour M, Jewell NP. *Causal Inference in Statistics – A Primer* (John Wiley & Sons, 2016). https://books.google.com/books?hl=en&lr=&id=IqCECwAAQBAJ&oi=fnd&pg=PT1&dq=Causal+Inference+in+Statistics&ots=NPpnh1N4lC&sig=-CyGyDAsTQP1vFstnAZh3dt-Ih8#v=onepage&q=Causal%20Inference%20in%20Statistics&f=false

Ramsey J. Scaling up Greedy Equivalence Search for continuous variables (2015). http://arxiv.org/ftp/arxiv/papers/1507/1507.07749.pdf

Spirtes P. Introduction to causal inference. *Journal of Machine Learning Research* 11 (2010) 1643-1662. http://jmlr.org/papers/volume11/spirtes10a/spirtes10a.pdf

Tetrad system (2016). http://www.phil.cmu.edu/tetrad/current.html

Appendix:
An Introduction to PAGs

Peter Spirtes

The output of the FCI[3] algorithm [Spirtes, 2001] is a partial ancestral graph (PAG), which is a graphical object that represents a set of causal Bayesian networks (CBNs) that cannot be distinguished by the algorithm.[4] Suppose we have a set of cases that were generated by random sampling from some CBN. Under the assumptions that FCI makes, in the large sample limit of the number of cases, the PAG returned by FCI is guaranteed to include the CBN that generated the data.

An example of a PAG is shown in Figure A2. This PAG represents the pair of CBNs in Figures A1a and A1b (where measured variables are boxes and unmeasured variables are ovals), as well as an infinite number of other CBNs that may have an arbitrarily large set of unmeasured confounders. Despite the fact that there are important differences between the CBNs in Figures A1a and A1b (e.g., there is an unmeasured confounder of X_1 and X_2 in Figure A1b but not in Figure A1a), they share a number of important features in common (e.g., in both CBNs, X_2 is a direct cause of X_6, there is no unmeasured confounder of X_2 and X_6, and X_6 is not a cause[5] of X_2). It can be shown that every CBN that a PAG represents shares certain features in common. The features that all CBNs represented by a PAG share in common can be read off of the output PAG according to the rules described next.

There are 4 kinds of edges that occur in a PAG: $A \rightarrow B$, A o$\rightarrow B$, A o–o B, and $A \leftrightarrow B$. The edges indicate what the CBNs represented by the PAG have in common. A description of the meaning of each edge in a PAG is given in Table A1.

---

[3] The results in this appendix also hold for the FCI+ [Claassen, 2013] and GFCI [Ogarrio, 2016] algorithms; for simplicity, we will just refer to the FCI algorithm in the remainder of the appendix. The RFCI algorithm [Colombo, 2012] outputs a slight modification of a PAG. The kind of PAG described here is actually a special case of a more general kind of PAG, where here we assume that there is no selection bias [Spirtes, 1999; Zhang, 2008].

[4] In the Gaussian and multinomial cases, the CBNs represented by a PAG cannot be distinguished by any algorithm without further assumptions.

[5] The word “cause” is used in this document to denote a cause that is either direct or indirect, relative to the measured variables. For example, in Figure A1a, X_1 is a direct cause of X_2 and an indirect cause of X_6.

Table A1: Types of edges in a PAG.

Edge type	Relationships that are present	Relationships that are absent
$A \rightarrow B$	*A* is a cause of *B*. It may be a direct or indirect cause that may include other measured variables. Also, there may be an unmeasured confounder of *A* and *B*.	*B* is not a cause of *A.*
$A \leftrightarrow B$	There is an unmeasured variable (call it *L*) that is a cause of *A* and *B*. There may be measured variables along the causal pathway from *L* to *A* or from *L* to *B*.	*A* is not a cause of *B*. *B* is not a cause of *A*.
A o$\rightarrow B$	Either *A* is a cause of *B*, or there is an unmeasured variable that is a cause of *A* and *B*, or both.	*B* is not a cause of *A.*
A o–o *B*	Exactly one of the following holds: (a) *A* is a cause of *B,* or (b) *B* is a cause of *A,* or (c) there is an unmeasured variable that is a cause of *A* and *B,* or (d) both a and c, or (e) both b and c.	

Table A1 is sufficient to understand the basic meaning of edge types in PAGs. Nonetheless, it can be helpful to know the following additional perspective on the information encoded by PAGs. Each edge has two endpoints, one on the *A* side, and one on the *B* side. For example $A \rightarrow B$ has a tail at the *A* end and an arrowhead at the *B* end. Altogether, there are three kinds of edge endpoints: a tail "–", an arrowhead ">", and a "o." Note that some kinds of combinations of endpoints never occur; for example, *A* o– *B* never occurs. As a mnemonic device, the basic meaning of each kind of edge can be derived from three simple rules that explain what the meaning of each kind of endpoint. A tail "–" at the *A* end of an edge between *A* and *B* means "*A* is a cause of *B*"; an arrowhead ">" at the *A* end of an edge between *A* and *B* means "*A* is *not* a cause of *B*"; and a circle "o" at the *A* end of an edge between *A* and *B* means "can't tell whether or not *A* is a cause of *B.*" For example $A \rightarrow B$ means that *A* is a cause of *B*, and that *B* is not a cause of *A* in all of the CBNs represented by the PAG.

The PAG in Figure A2 shows examples of each type of edge, and the CBNs in Figure A1 show some examples of what kinds of CBNs can be represented by that PAG.

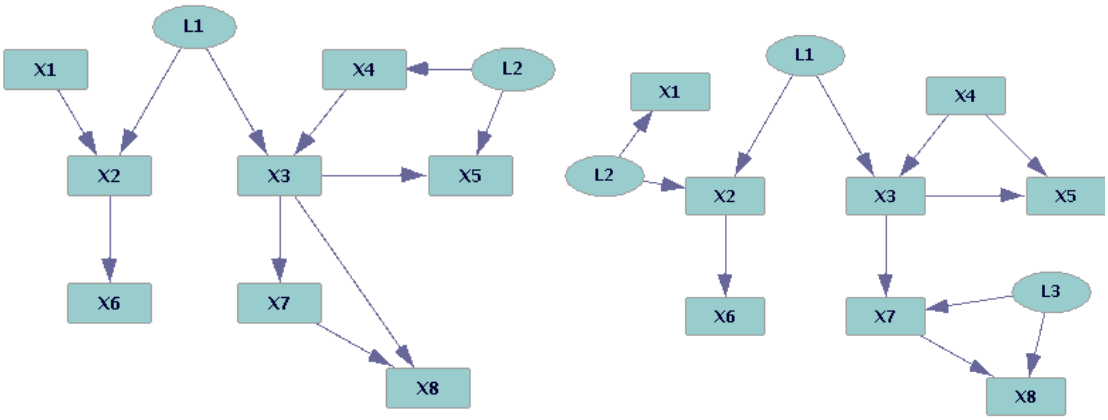

Figure A1. Two CBNs that FCI (as well as FCI+, GFCI, and RFCI) cannot distinguish.

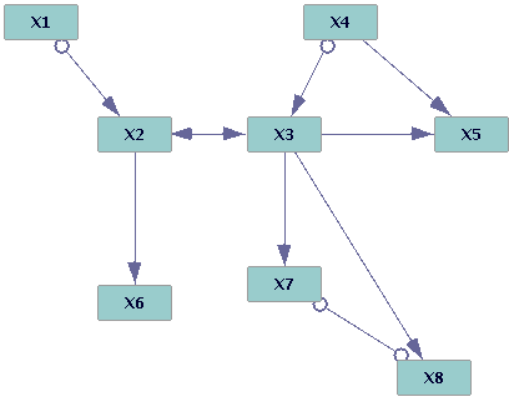

Figure A2. The PAG that represents the CBNs in both Figures A1a and A1b.

References

CCD-FGES (2016). Fast greedy search (FGES) algorithm for continuous variables. Available at: http://www.ccd.pitt.edu/wiki/index.php?title=Fast_Greedy_Search_(FGES)_Algorithm_for_Continuous_Variables

Claasen T, Mooij JM, Heskes T (2013). Learning sparse causal models is not NP-hard. *Uncertainty In Artificial Intelligence*, 29, 172-181.

Colombo D, Maathuis MH, Kalisch M, Richardson TS (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics,* 40, 294-321.

Lagani V, Triantafillou S, Ball G, Tegner J, Tsamardinos I (2016). Probabilistic computational causal discovery for systems biology. *Uncertainty in Biology*, 17, 33-73. http://www.mensxmachina.org/files/publications/Probabilistic%20Causal%20Discovery%20for%20Systems%20Biology_prePrint.pdf

Neapolitan RE (2003). *Learning Bayesian Networks* (Pearson).

Ogarrio JM, Spirtes P, Ramsey J (2016). A hybrid causal search algorithm for latent variable models. *JMLR Workshop and Conference Proceedings*, 52, 368-379. http://www.jmlr.org/proceedings/papers/v52/ogarrio16.pdf

Pearl J, Glymour M, Jewell NP (2016). *Causal Inference in Statistics – A Primer* (John Wiley & Sons). https://books.google.com/books?hl=en&lr=&id=IqCECwAAQBAJ&oi=fnd&pg=PT1&dq=Causal+Inference+in+Statistics&ots=NPpnh1N4lC&sig=-CyGyDAsTQP1vFstnAZh3dt-Ih8#v=onepage&q=Causal%20Inference%20in%20Statistics&f=false

Ramsey J (2015). Scaling up greedy equivalence search for continuous variables. http://arxiv.org/ftp/arxiv/papers/1507/1507.07749.pdf

Spirtes P, Glymour C, Scheines R (1993). Causation, Prediction, and Search (Springer). https://books.google.com/books/about/Causation_Prediction_and_Search.html?id=oUjxBwAAQBAJ&printsec=frontcover&source=kp_read_button&hl=en#v=onepage&q&f=false

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

Spirtes P, Richardson T, Meek C (1999). Causal discovery in the presence of latent variables and selection bias. In G Cooper & C Glymour (Eds.) *Computation, Causality, and Discovery*, pp. 211-252 (AAAI Press).

Spirtes P (2010). Introduction to causal inference. *Journal of Machine Learning Research,* 11, 1643-1662. http://www.jmlr.org/papers/volume11/spirtes10a/spirtes10a.pdf

Tetrad system (2016). http://www.phil.cmu.edu/tetrad/

Zhang J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias". *Artificial Intelligence*, *Volume 172, Issues 16–17,* 1873-1896.