On Scoring Maximal Ancestral Graphs with the Max-Min Hill Climbing Algorithm

Konstantinos Tsirlis, Vincenzo Lagani, Sofia Triantafillou, Ioannis Tsamardinos

Abstract

We consider the problem of causal structure learning in presence of latent confounders. We propose a hybrid method, MAG Max-Min Hill-Climbing (M3HC) that takes as input a data set of continuous variables, assumed to follow a multivariate Gaussian distribution, and outputs the best fitting maximal ancestral graph. M3HC builds upon a previously proposed method, namely GSMAG, by introducing a constraint-based first phase that greatly reduces the space of structures to investigate. We show on simulated data that the proposed algorithm greatly improves on GSMAG, and compares positively against FCI and cFCI, two well known constraint-based approaches for causal-network reconstruction in presence of latent confounders.

Keywords

Score-based causal structure learning · Maximal Ancestral Graphs · Max-Min Hill Climbing

1 Introduction

Learning causal networks from observational data has been the subject of intense research for at least the last three decades [12]. A notable result obtained so far is that the joint probability distribution of the data can be linked to (a family of) possible causal structures by assuming faithfulness and the causal Markov condition. Directed Acyclic Graphs (DAGs) are one of the simplest types of causal graphs. In DAGs nodes denote the measured variables while edges indicate direct causal relationship in the form $A \rightarrow B$. Several algorithms have been proposed for learning DAGs, most of them falling under two categories, i.e., score-based and constraint-based methods. The first ones attempt to identify the causal structure that optimize a criterion of fitness on the data; the latter use a graphical criterion (d-separation) to constraint the causal structure according to the conditional (in)dependencies that hold in the data. Hybrid methods also exist, as for example the Max-Min Hill-Climbing algorithm (MMHC, [15]). MMHC first utilizes tests of conditional independence to restrict the search space and afterwards uses greedy search to identify the structure that optimizes a scoring function.

Regardless of the approach, usually more than one DAG can represent equally well the same set of relations in a distribution, forming a Markov equivalence class. Complete Partially Directed Acyclic Graphs (CPDAGs) represent Markov equivalence classes by introducing indirect edges, which cannot be oriented on the basis of the information contained in the data. DAGs' main drawback is the inability of representing latent variables, i.e., they assume causal sufficiency, an assumption that rarely holds in real world scenarios. Maximal Ancestral Graphs (MAGs) [10] are an extension of DAGs that can represent the presence of latent confounders. MAGs' causal semantics are more complicated than DAGs': directed edges denote causal ancestry, but the relation is not necessarily direct, and bi-directed edges denote the existence of confounding. The most attractive properties of MAGs is that they are closed under marginalization, and every non-adjacency in the graph corresponds to a conditional independence in the probability distribution. Partial Ancestral Graphs
(PAGs) represent the Markov equivalence classes of MAGs by using a special notation (o) for edge endpoints that cannot be determined.

Until recently, only constraint-based methods existed for learning causal structures in presence of confounding. These algorithms use an extension of d-separation, namely m-separation, that holds also in presence of hidden variables. FCI [12,10], is the first asymptotically correct constraint-based algorithm for MAGs (PAGs). This algorithm works in two stages. The first one attempts to identify the undirected skeleton of the graph, while the subsequent orientation phase uses a set of rules for orienting the edges. In practice, the orientation phase is prone to error propagation [11]. Conservative FCI [9] is an extension of FCI that performs additional conditional independence tests during the orientation phase and output only robust orientations that are consistent with all tested conditional independencies.

Triantafillou and Tsamardinos [14] proposed a score-based algorithm able to deal with confounding, namely the Greedy Search for MAGs (GSMAG). This algorithm builds upon a score function for mixed graphs [7], coupled with a simple greedy search upon the space of all possible MAGs. The Greedy Fast Causal Inference (GFCI, [8]) uses a different strategy, where a first approximation of the causal graph is obtained with a score-based method that ignores latent variables and then FCI orientation rules are used for identifying possible confounding.

In this paper we move along the lines of Triantafillou and Tsamardinos [14], by adapting the MMHC algorithm to be able to incorporate latent variables. The resulting algorithm, M3HC (MMHC for MAGs) is contrasted against GSMAG and against both FCI and cFCI on simulated data. The results of the experiments show that M3HC consistently outperforms or is on par with GSMAG and often provides improved performances over (c)FCI.

2 The M3HC algorithm

In this section we present the M3HC algorithm. Let \( D \) be a dataset containing \( N \) i.i.d. samples measured over \( V \) continuous variables, jointly following a multivariate Gaussian distribution \( \mathcal{N}(0, \Sigma) \) with positive definite covariance matrix \( \Sigma \). Our objective is to identify a MAG \( G \) that belongs to the Markov equivalent class of MAGs representing the data at hand. According to Richardson [10], the Bayesian Information Score (BIC) is an asymptotically correct criterion for scoring MAGs:

\[
BIC(\hat{\Sigma}, G) = -2 \ln(l_p(\hat{\Sigma}|G)) + \ln(N)(2V + E),
\]

where \( l_p(\hat{\Sigma}|G) \) is \( G \) likelihood, \( \hat{\Sigma} \) is a maximum likelihood estimate of the covariance matrix \( \Sigma \) [3], while \( V \) and \( E \) are the number of vertices and edges in \( G \), respectively.

Nowzohour and co-authors [6] demonstrated that \( l_p(\hat{\Sigma}|G) \) can be decomposed according to the c-components of \( G \): \( l_p(\hat{\Sigma}|G) = \sum_k s_k \), where each \( s_k \) correspond to a c-component and can be computed with a closed formula [6]. The c-components of a graph \( G \) correspond to the connected components of its bi-directed part, i.e., the graph stemming from \( G \) after the removal of all directed edges.

Algorithm [1] shows M3HC operation. The first phase of the algorithm ("skeleton identification") identifies the set \( P \) of variable pairs that are allowed to be directly connected by an edge in the final graph. In other words, the final MAG \( G \) can have an edge between variables \( \{X,Y\} \) only if \( \{X,Y\} \in P \). In the second phase the
Input: Data set \(D\) over variables \(V\), significance threshold \(\alpha\), maximum conditioning set \(\max K\).

Output: Set of admissible variable pairs \(P\).

\(P \leftarrow \emptyset;\)

foreach \(V_i \in V\) do
    \(PC_i \leftarrow \text{mmpc}(D, V_i, \alpha, \max K);\)
end

doi

foreach pair \((V_i, V_j) \in V\) do
    if \(V_i \in PC_i \text{ and } V_j \in PC_i\) then
        \(P \leftarrow P \cup \{V_i, V_j\};\)
end
end

\(sc \leftarrow curScore;\)

Algorithm 2: The mmpcSkeleton function

The empty graph is first set as initial solution and its score as current score. The algorithm then proceed in identifying for each pair of variables \((X, Y) \in P\) the action that improves the score the most. Possible actions include adding and deleting an edge, and changing the edge endpoints. As soon as an action improves upon the current score, the corresponding MAG and score are set as current solution and current score, respectively. The procedure terminates when no action on any admissible pair produces an improvement over the current score, and the current solution is returned. For consistency with (c)FCI we refer to the second part of \(M^3\)HC as “orientation phase”, even though its operation is not limited to edges’ orientation.

The logic behind the \(M^3\)HC algorithm is the same as its predecessor MMHC. The skeleton phase is fast because it runs local algorithms (MMPc) that quickly prune the search space and don’t operate at the level of the whole network. The scoring phase orients the edges by taking in account the information from all available data, and becomes fast because the first phase has largely pruned the space of possible MAGs.

The core of \(M^3\)HCs’ first phase is the \text{mmpcSkeleton} function, represented in Algorithm 2. This function applies the Max-Min Parent Children (MMPc) algorithm in turn on each variable. MMPc is a constraint-based, feature selection method that attempts to identify the variables that are adjacent to the target variable in the DAG representing the data, that is, the variables that are either the parents or the children of the target variable. The set of variable pairs \(P\) is given by all pairs of variables that are contained in each other’s Parent-Children set.

The second (orientation) phase of \(M^3\)HC is based upon the \text{scoreContribution} and \text{updateScore} functions. Both of these functions are defined and discussed in details in Triantafillou and Tsamardinos work [13]. In short, the \text{scoreContribution} function computes the score for each single c-component, while \text{updateScore} updates at the same time all scores and the c-components after a change in the graph.

3 Experimental protocol

We performed a large set of experiments on simulated data for contrasting our proposed \(M^3\)HC method against GSMAG and (c)FCI.

3.1 Comparison of \(M^3\)HC with GSMAG

We first compared \(M^3\)HC and GSMAG. Fifty random DAGs, equipped with Conditional Linear Gaussian parameterization, were generated for each variable size \([10, 20, 50, 70]\) and for each maximum number of parents for each node in \([3, 5]\) for a total number of 400 different random DAGs. 10% of the variables were chosen uniformly at random as latent variables, and we assumed their values not to be observed in the subsequent analyses. For each DAG we simulated one dataset with a varying sample size \([100, 1000, 10000, 100000]\) producing a total number of 1600 different synthetic datasets. The 4 batches of 400 different random DAGs were the same for all the different sample sizes. In each iteration we generated a DAG, we generated a data set and we applied both algorithms on it. Their output (a MAG converted to a PAG) was compared with the Ground Truth PAG (GTPAG) with the latent variables marginalized.

3.2 Comparison of \(M^3\)HC with (c)FCI

In the second phase of the experiments we compared \(M^3\)HC with FCI and cFCI. We followed the same schema used for contrasting \(M^3\)HC and GSMAG, but we extended the numbers of variables in each DAG to \([10, 20, ..., 50, 100, 200, 400, 600, 800]\) (only 5 random DAGs were generated with 800 variables to keep the computational time reasonably low) and we marginalize in turn a uniformly randomly chosen percentage (10% and 30%) of the variables each time. Furthermore, we also marginalize 50% of the variables for all datasets with 10000 samples, to explore what would happen in such an extreme (but not at all unrealistic) case. The total number of different synthetic datasets produced for this comparison is 6380.
3.3 M³HC on the Insurance and the full Munin network

We further applied M³HC on the Insurance (27 nodes) and the full Munin (1041 nodes) networks for demonstration purposes. Both networks were equipped with a conditional linear Gaussian parameterization to generate one dataset with 100000 samples. For the Insurance network we hide as latent the “RiskAversion”, the “DrivSkill” and “OtherCarCost” variables. These nodes were chosen manually as prototypical information that would not be easy to measure in a real-world scenario. For the Munin network 10% latent variables were chosen uniformly at random.

3.4 Experimental set-up

Both M³HC and (c)FCI requires the specification of two different hyper-parameters: a threshold of statistical independence $\alpha$ and the maximum number of variables to condition upon $\text{max}K$. The significance thresholds were set to 0.05 for sample size 100 and 1000 and we lowered it by 2 orders of magnitude for each subsequent sample size. The maximum conditioning set was held fixed at 10 for all tests of conditional independence, which we believe is high enough given the sparseness of the DAGs we simulate.

For M³HC and GSMAG we used a heuristic for escaping local maxima; specifically, we equipped both algorithms with a TABU list [5]. The TABU list keeps the best 100 candidate MAGs that best fit the data according to the scoring function, and if there is no improvement in the score after 50 iterations the algorithms terminate and the top scored MAG is returned as the final output MAG.

Finally, we used 6 different performance metrics to compare the algorithms output. To note, the output of M³HC and GSMAG were transformed in their corresponding PAGs, so that to be comparable with the output of (c)FCI.

- Structural Hamming Distance (SHD) [15] directly compares the structure of the learned and the original networks and its use is fully oriented toward discovery, rather than inference. Tsamardinos et. al. [13] defined the Structural Hamming Distance between two PDAGs as the number of the following operators required to make the PDAGs match: add or delete an undirected edge, and add, remove, or reverse the orientation of an edge. Triantafillou et. al. [11] adapted it to PAGs by adding the necessary additional penalties when adding or deleting a bi-directed edge.

- Running time is the time required for a single iteration to finish, where “iteration” corresponds to a single algorithm applied on a single simulated dataset. For M³HC we further record the running time required for both its first and second phase.

- Precision and Recall are defined as in Tillman et. al. (2011a) [13]. Precision is the number of correct edges with correct orientation (or lack of orientation, when undirected) in the predicted PAG divided by the number of edges in the predicted PAG, while Recall is defined as the number of correct edges with correct orientation (or lack of orientation, when undirected) in the predicted PAG divided by the number of correct edges in the ground truth PAG.

- We further dissect the difference between the predicted and ground truth PAG by computing the number of different undirected edges and the number of different endpoints between the two.

To find whether the differences in the performance metrics were statistically significant, we conducted standard two-tail paired $t$-tests between M³HC and all other algorithms in turn.

4 Results

4.1 M³HC vs GSMAG

M³HC outperforms, often in a statistically significant way, GSMAG in the vast majority of the comparisons and across all metric performance. Figure 4 shows the results for sample size 10000 and 3 maximum parents; results for all other combinations of sample size and number of maximum parents largely follow the same trends (see Supplementary Material).

In general, the difference between M³HC and GSMAG increases along with the number of nodes in the network. Also, performance tend to be worse for both algorithms when a maximum of 5 parents are allowed for each node in the network (i.e., for denser structures). M³HC running time increases approximately linearly with the number of nodes, and its variance remains stable. In contrast, GSMAG running time varies quite widely in large networks, and its increase is clearly super-linear. Interestingly, for 5 maximum parents both algorithms have worse performance in terms of endpoint orientation as we increased the sample size from 100 to 100000.

4.2 M³HC vs (c)FCI

Figure 2 and 3 shows the results for sample size 1000, 10% latent variables and 3 maximum parents and sam-
Comparison between \( \text{M}^3 \text{HC} \) (blue) and GSMAG (red) for sample size 10000, maximum parents 3 and 10% latent variables. Each panel reports the variation of a specific metric with respect to the number of variables, from top-left in clockwise order: Structural Hamming Distance (SHD), computation time, precision, difference in endpoints, difference in edges, recall. The lower the SHD and the computational time the better the performance, and vice versa for the other metrics. Statistically significant differences are marked with triangles (solid marker: \( \text{M}^3 \text{HC} \) has better performance. Empty marker: GSMAG outperforms \( \text{M}^3 \text{HC} \)). One, two or three markers indicate significance at level 0.05, 0.01 and 0.001, respectively.

Table 1 \( \text{M}^3 \text{HC} \) performance metrics in the full Munin network with generated data and 10% latent variables

<table>
<thead>
<tr>
<th>Metric</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHD</td>
<td>1243</td>
<td>615</td>
</tr>
<tr>
<td>Running Time (sec)</td>
<td>69160</td>
<td>62060</td>
</tr>
<tr>
<td>Precision</td>
<td>0.7437</td>
<td>0.0563</td>
</tr>
<tr>
<td>Recall</td>
<td>0.6374</td>
<td>0.1388</td>
</tr>
</tbody>
</table>

sent the differences between the true PAG and the one reconstructed by \( \text{M}^3 \text{HC} \). Black represents correctly retrieved and oriented edges, blue represents one wrong endpoint, yellow represents both endpoints wrong and red represents a missing edge. Overall, on this network \( \text{M}^3 \text{HC} \) achieves 0.42 precision and 0.32 recall. However, the yellow and blue edges amount to the 42.7% of all edges (29 out of 58), while the red edges reach only the 25% (17 edges). This means that the majority of \( \text{M}^3 \text{HC} \) errors occur in the orientation, as also evident in Figures 1-3, where the number of different endpoints is usually larger than the number of different edges. Particularly, all \( \text{M}^3 \text{HC} \) mis-orientation in the Insurance network happen for the circle endpoint of the ground-truth PAG. This means that \( \text{M}^3 \text{HC} \) predictions for endpoints that were either directed (\( \rightarrow \)) or undirected (\( \sim \)) in the ground-truth PAG were always correct.

Table 1 reports the average performances of \( \text{M}^3 \text{HC} \) over three different runs for the Munin network (1041 nodes, 100000 samples, 10% latent variables randomly
Fig. 2 Comparison between M³HC (blue), cFCI (red) and FCI (green) for sample size 1000, 10% latent variables and maximum parents 3. Each panel reports the variation of a specific metric with respect to the number of variables, from top-left in clockwise order: Structural Hamming Distance (SHD), computation time, precision, difference in endpoints, difference in edges, recall. The lower the SHD and the computational time the better the performance, and vice versa for the other metrics. Pairwise statistically significant differences are marked with triangles. Red color: pairwise comparison between M³HC and cFCI. Green color: pairwise comparison between M³HC and FCI. Solid marker: M³HC has better performance. Empty marker: (c)FCI outperforms M³HC. One, two and three markers indicate significance at level 0.05, 0.01 and 0.001, respectively.

Fig. 3 Comparison between M³HC (blue), cFCI (red) and FCI (green) for sample size 1000, 30% latent variables and maximum parents 3. Details as in Figure 2

chosen each time). Notably, M³HC achieves quite high values of precision and recall, 0.74 and 0.64, respectively.

Table 2 reports the internal time spent in the first (identification) and second (orientation) phase of M³HC. To note, the percentage of the total time spent in the first phase varies widely, from 37% to 86%.

5 Discussion

In this work we introduce M³HC, a score-based algorithm for continuous data able to deal with latent confounders. We contrast the proposed method against a similar algorithm, namely GSMAG, and two prototypi-
Fig. 4 Differences between the GTPAG and the $M^3$HC-reconstructed PAG of the Insurance Network. Black: Correct edge and correct both the endpoints, blue: one wrong endpoint, yellow: two wrong endpoints, red: missing edge.

Table 2 $M^3$HC internal running times over three repetitions in the full Munin network. At each repetition 10% of the nodes were randomly selected to be latent variables.

<table>
<thead>
<tr>
<th>Repetition</th>
<th>Identification phase</th>
<th>Orientation phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18260 (49%)</td>
<td>18270 (51%)</td>
</tr>
<tr>
<td>2</td>
<td>120700 (86%)</td>
<td>19400 (14%)</td>
</tr>
<tr>
<td>3</td>
<td>11070 (37%)</td>
<td>18040 (63%)</td>
</tr>
</tbody>
</table>

The results of our experiments show that $M^3$HC dominates GSMAG in all comparison, both in terms of computational time and predictive performances. The number of possible different MAGs is combinatorial with the number of variables. In contrast to $M^3$HC, GSMAG do not impose any restrictions in the MAG space, and so the running times of GSMAG become computationally intractable really fast when the number of nodes in the network increases. Restricting the space of possible MAGs has an evident beneficial effect also on $M^3$HC’s predictive performances, which are consistently better than GSMAG’s performances. Only for small sample sizes GSMAG has better recall. This might be due to low statistical power in the identification phase of $M^3$HC, where the tests of conditional independence used by MMPC may miss some of the actual dependencies.

In all our experiments we coupled both $M^3$HC and GSMAG with a TABU heuristic for avoiding local maxima. We observed that adding a TABU list actually increases the performance of the $M^3$HC algorithm (results not shown), while this increase does not hold for GSMAG [14]. We hypothesize that the much larger search space of GSMAG and the corresponding higher number of local maxima nullify the beneficial effect of the TABU heuristic.

$M^3$HC usually outperforms (c)FCI in terms of the capability of reconstructing the correct network. Particularly for low samples size, or in general for cases where the number of variables is higher than the number of samples, $M^3$HC seem to have better performance than (c)FCI. This is probably due to the low statistical power of the conditioning independence tests upon which FCI and (c)FCI base their whole operation. Moreover the precision of $M^3$HC is increasingly better than the pre-
cision of (c)FCI as we increase the number of latent variables.

In contrast, (c)FCI is usually quite faster than $M^3$HC. However, we did not make any effort to optimize the current $M^3$HC implementation, being more interested at this point in $M^3$HC reconstruction capabilities than computational speed.

Furthermore, applying $M^3$HC on the Munin network produces quite good results in term of both precision and recall, which is surprising for such a large (1041 variables) structure. Interestingly, much of $M^3$HC computational time is spent in the first identification phase; this is in contrast with the results obtained on the randomly generated DAGs, where at most 5% the computational time is spent in the first phase.

We plan to expand the present work in several directions. First, we plan to implement a number of algorithmic solutions for increasing $M^3$HC computational speed, including possibly incorporating block-coordinate descent (BCD) method for obtaining more efficient and precise maximum likelihood estimates [4]. This improvement would also resolve some of the numerical instabilities that lead to higher variation in the $M^3$HC computational time for cases where the number of variables is higher than the numbers of samples. Other lines of work will include trying different scoring criteria, initialization strategies for $M^3$HC scoring phase, and adapting the algorithm for dealing with both experimental and observational data.

Finally, we also plan to enlarge the scope of our experimentation, in order to include several other algorithms, as for example GFCI, and additional causal networks from different domains, so that to better characterize and contrast $M^3$HC capabilities and performance.

5.1 Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Acknowledgements The authors would like to thank Professor Mathias Drton and Y. Samuel Wang from the Department of Statistics of the University of Washington (USA) for their useful comments regarding RICF algorithm and Gorgos Borboudakis from the Department of Computer Science of the University of Crete (Greece) for his useful comments regarding MMPC and MMHC. Also Professor George E. Frouda- kis from the Department of Chemistry of the University of Crete (Greece) for providing the infrastructure to run some of the experiments. This work was funded by the European Research Council (ERC) and is part of the CAUSALPATH - Next Generation Causal Analysis project, No 617393.

References

Supplementary material for the paper: On Scoring Maximal Ancestral Graphs with the Max-Min Hill Climbing Algorithm

Konstantinos Tsirlis, Vincenzo Lagani, Sofia Triantafillou, Ioannis Tsamardinos
Fig. 1 Comparison between M^3HC (blue) and GSMAG (red) for sample size 100, maximum parents 3 and 10% latent variables. Each panel reports a specific metric, in clockwise order: Structural Hamming Distance (SHD), computation time, precision, difference in endpoints, difference in edges, recall. Statistically significant differences are marked with inverse triangles (solid marker: M^3HC has better performances. Empty marker: GSMAG outperforms M^3HC). The number of markers indicate significance at level 0.05, 0.01 and 0.001, respectively.

Fig. 2 Comparison between M^3HC (blue) and GSMAG (red) for sample size 100, maximum parents 5 and 10% latent variables. Details as in Figure 1
**Fig. 3** Comparison between M³HC (blue) and GSMAG (red) for sample size 1000, maximum parents 3 and 10% latent variables. Details as in Figure 1.

**Fig. 4** Comparison between M³HC (blue) and GSMAG (red) for sample size 1000, maximum parents 5 and 10% latent variables. Details as in Figure 1.
Fig. 5 Comparison between $M^3$HC (blue) and GSMAG (red) for sample size 10000, maximum parents 3 and 10% latent variables. Details as in Figure 1.

Fig. 6 Comparison between $M^3$HC (blue) and GSMAG (red) for sample size 10000, maximum parents 5 and 10% latent variables. Details as in Figure 1.
Fig. 7 Comparison between M^3HC (blue) and GSMAG (red) for sample size 100000, maximum parents 3 and 10% latent variables. Details as in Figure 1.

Fig. 8 Comparison between M^3HC (blue) and GSMAG (red) for sample size 100000, maximum parents 5 and 10% latent variables. Details as in Figure 1.
**Fig. 9** Comparison between $M^3$HC (blue), cFCI (red) and FCI (green) for sample size 100, 10% latent variables and maximum parents 3. Each panel reports a specific metric, in clockwise order: Structural Hamming Distance (SHD), computation time, precision, difference in endpoints, difference in edges, recall. Pairwise statistically significant differences are marked with inverse triangles (Red color: pairwise comparison between $M^3$HC and cFCI. Green color: pairwise comparison between $M^3$HC and FCI. Solid marker: $M^3$HC has better performances. Empty marker: (c)FCI outperforms $M^3$HC). The number of markers indicate significance at level 0.05, 0.01 and 0.001, respectively.

**Fig. 10** Comparison between $M^3$HC (blue), cFCI (red) and FCI (green) for sample size 100, 10% latent variables and maximum parents 5. Details as in Figure 9
Fig. 11 Comparison between $M^3$HC (blue), cFCI (red) and FCI (green) for sample size 100, 30% latent variables and maximum parents 3. Details as in Figure 9

Fig. 12 Comparison between $M^3$HC (blue), cFCI (red) and FCI (green) for sample size 100, 30% latent variables and maximum parents 5. Details as in Figure 9
Fig. 13 Comparison between $\text{M}^3\text{HC}$ (blue), cFCI (red) and FCI (green) for sample size 1000, 10% latent variables and maximum parents 3. Details as in Figure 9

Fig. 14 Comparison between $\text{M}^3\text{HC}$ (blue), cFCI (red) and FCI (green) for sample size 1000, 10% latent variables and maximum parents 5. Details as in Figure 9
Fig. 15 Comparison between $\text{M}^3\text{HC}$ (blue), cFCI (red) and FCI (green) for sample size 1000, 30% latent variables and maximum parents 3. Details as in Figure 9.

Fig. 16 Comparison between $\text{M}^3\text{HC}$ (blue), cFCI (red) and FCI (green) for sample size 1000, 30% latent variables and maximum parents 5. Details as in Figure 9.
Fig. 17 Comparison between \( M^3 \text{HC} \) (blue), cFCI (red) and FCI (green) for sample size 10000, 10% latent variables and maximum parents 3. Details as in Figure 9

Fig. 18 Comparison between \( M^3 \text{HC} \) (blue), cFCI (red) and FCI (green) for sample size 10000, 10% latent variables and maximum parents 5. Details as in Figure 9
Fig. 19 Comparison between M$^3$HC (blue), cFCI (red) and FCI (green) for sample size 10000, 30% latent variables and maximum parents 3. Details as in Figure 9

Fig. 20 Comparison between M$^3$HC (blue), cFCI (red) and FCI (green) for sample size 10000, 30% latent variables and maximum parents 5. Details as in Figure 9
Fig. 21 Comparison between $M^3HC$ (blue), cFCI (red) and FCI (green) for sample size 10000, 50% latent variables and maximum parents 3. Details as in Figure 9

Fig. 22 Comparison between $M^3HC$ (blue), cFCI (red) and FCI (green) for sample size 10000, 50% latent variables and maximum parents 5. Details as in Figure 9