



OPEN

## An order independent algorithm for inferring gene regulatory network using quantile value for conditional independence tests

Sayed Hadi Mahmoodi<sup>1</sup>, Rosa Aghdam<sup>1,2</sup>✉ & Changiz Eslahchi<sup>1,2</sup>✉

In recent years, due to the difficulty and inefficiency of experimental methods, numerous computational methods have been introduced for inferring the structure of Gene Regulatory Networks (GRNs). The Path Consistency (PC) algorithm is one of the popular methods to infer the structure of GRNs. However, this group of methods still has limitations and there is a potential for improvements in this field. For example, the PC-based algorithms are still sensitive to the ordering of nodes i.e. different node orders results in different network structures. The second is that the networks inferred by these methods are highly dependent on the threshold used for independence testing. Also, it is still a challenge to select the set of conditional genes in an optimal way, which affects the performance and computation complexity of the PC-based algorithm. We introduce a novel algorithm, namely Order Independent PC-based algorithm using Quantile value (OIPCQ), which improves the accuracy of the learning process of GRNs and solves the order dependency issue. The quantile-based thresholds are considered for different orders of CMI tests. For conditional gene selection, we consider the paths between genes with length equal or greater than 2 while other well-known PC-based methods only consider the paths of length 2. We applied OIPCQ on the various networks of the DREAM3 and DREAM4 in silico challenges. As a real-world case study, we used OIPCQ to reconstruct SOS DNA network obtained from *Escherichia coli* and GRN for acute myeloid leukemia based on the RNA sequencing data from The Cancer Genome Atlas. The results show that OIPCQ produces the same network structure for all the permutations of the genes and improves the resulted GRN through accurately quantifying the causal regulation strength in comparison with other well-known PC-based methods. According to the GRN constructed by OIPCQ, for acute myeloid leukemia, two regulators BCLAF1 and NRSF reported previously are significantly important. However, the highest degree nodes in this GRN are ZBTB7A and PU1 which play a significant role in cancer, especially in leukemia. OIPCQ is freely accessible at <https://github.com/haammim/OIPCQ-and-OIPCQ2>.

Identifying regulations between genes is an important issue for better understanding the biological processes<sup>1–4</sup>. It provides information on what genes of particular interest are over-expressed or under-expressed by different environmental conditions. Experimental methods for inference of Gene Regulatory Networks (GRN) are expensive, tedious, time-consuming and sometimes not reproducible. Recently, gene expression data is accessible through high-throughput sequencing technologies, which provides an insight on the regulatory mechanism<sup>1,3,5–7</sup>. In GRN, genes are denoted as nodes while the goal is to detect interactions between them, referred to as edges. Computational methods developed to reconstruct GRNs are generally categorized in either machine-learning-based or model-based methods<sup>8–35</sup>. In literature, Pearson correlation coefficients<sup>36,37</sup> and information theory<sup>5,16,19–27,29–32</sup> are widely used to measure the regulation strength between genes. Both information theory and Pearson correlation coefficient can infer large-scale networks, but Mutual Information (MI) has the capability to measure non-linear dependencies which is a suitable measure to distinguish the relation between genes<sup>36,38</sup>. The Path Consistency (PC) method and its improvements (PC-based methods) are used for inferring the structure of GRN. PC-based methods such as Fast Causal Inference (FCI), Really Fast Causal Inference (RFCI), PC Algorithm based on Conditional Mutual Information (PCA-CMI) and their modifications<sup>25,39–46</sup> have two common drawbacks. The first

<sup>1</sup>Department of Computer and Data Sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran. <sup>2</sup>School of Biological Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. ✉email: [rosa.aghdam@ipm.ir](mailto:rosa.aghdam@ipm.ir); [ch-eslahchi@sbu.ac.ir](mailto:ch-eslahchi@sbu.ac.ir)

is that these methods are not consistent for different sequential node orders<sup>47</sup>. The second is that the networks inferred by these methods are highly dependent on the threshold used for independence testing. Consensus Network (CN)<sup>16</sup>, introduced Sequential ORDERing (SORDER) algorithm to select a suitable sequential ordering of genes. It also improves the accuracy of the obtained results by taking the consensus of different networks. Zhang et al.<sup>19</sup> introduced Conditional Mutual Inclusive Information (CMI2), which improves the GRN skeleton by utilizing interventional probability and Kullback–Leibler (KL) divergence. One of the issues so far unresolved in the literature is the order-dependency restriction of the algorithms, which the current work aims to address. Also, in conditional-independent tests, the proper selection of a collection of nodes which contains the separator sets significantly influences the performance of constraint-based methods. In the proposed method, a strategy for an effective selection of nodes based on existing paths between any pair of genes is devised in order to improve the network results. Also, our method is an order independent algorithm to reconstruct GRNs from gene expression data to overcome restrictions of order-dependent algorithms. The rest of the paper is organized as follows: Section Preliminaries is related to the details of MI, CMI, CMI2, PCA-CMI, and CMI2NI algorithm. In section Results, the results of OIPCQ on the Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenges and the SOS DNA network in *Escherichia coli* were compared with the results of three state-of-the-art approaches including PCA-CMI, CN and CMI2NI. Finally, a case study was provided to evaluate the performance of OIPCQ for inferring a network of Acute Myeloid Leukemia (AML). The gene expression data is available in The Cancer Genome Atlas (TCGA) website at <http://cancergenome.nih.gov/>. The discussion and some possible further works are presented in Section Discussion. In section Methods, the drawbacks of PC-based algorithms and the details of OIPCQ Algorithm are presented.

## Preliminaries

**Mutual information and conditional mutual information.** Both MI and CMI are proven to be effective for inferring GRNs due to their capability to measure nonlinear dependencies between variables<sup>48</sup>. MI and CMI between the variables  $X$  and  $Y$ , given the vector of variables  $\mathbf{Z}$ , are defined as follows<sup>49,50</sup>:

$$MI(X, Y) = \int_{\mathcal{R}} \int_{\mathcal{R}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy, \quad (1)$$

$$CMI(X, Y|\mathbf{Z}) = \int_{\mathcal{R}^p} \int_{\mathcal{R}} \int_{\mathcal{R}} p(x, y, \mathbf{z}) \log \frac{p(x, y|\mathbf{z})}{p(x|\mathbf{z})p(y|\mathbf{z})} dx dy d\mathbf{z}, \quad (2)$$

where  $p$  is the dimension of vector  $\mathbf{Z}$  and  $p(x, y)$ ,  $p(x)$  and  $p(y)$  represent the joint distribution of  $X$  and  $Y$ , marginal distribution of  $X$ , marginal distribution of  $Y$ , respectively.  $p(x, y, \mathbf{z})$ ,  $p(x, y|\mathbf{z})$ ,  $p(x|\mathbf{z})$  and  $p(y|\mathbf{z})$  indicate joint distribution of  $X$ ,  $Y$  and  $\mathbf{Z}$ , the conditional density distribution of  $X$  and  $Y$  given  $\mathbf{Z}$ , the conditional density distribution of  $X$  given  $\mathbf{Z}$  and the conditional density distribution of  $Y$  given  $\mathbf{Z}$ , respectively. Under the assumption that gene expression data follows a Gaussian distribution, MI for two continuous variables  $X$  and  $Y$  can be calculated as:

$$MI(X, Y) = \frac{1}{2} \log \frac{\sigma_X^2 \sigma_Y^2}{\sigma_{XY}^2}, \quad (3)$$

where  $\sigma_X^2$ ,  $\sigma_Y^2$  and  $\sigma_{XY}$  indicate the variance of  $X$ , the variance of  $Y$  and the covariance between  $X$  and  $Y$ , respectively. When  $X$  and  $Y$  are independent, then  $MI(X, Y) = 0$ . Similarly,  $CMI(X, Y|\mathbf{Z})$  is defined as:

$$CMI(X, Y|\mathbf{Z}) = \frac{1}{2} \log \frac{|C(X, \mathbf{Z})||C(Y, \mathbf{Z})|}{|C(\mathbf{Z})||C(X, Y, \mathbf{Z})|}, \quad (4)$$

where  $C$  is the covariance matrix and  $|\cdot|$  is the determinant of matrix  $C$ . In which  $C(X, Y)$  and  $C(X, Y, \mathbf{Z})$  denote the covariance matrix of variables  $X$  and  $Y$  and variables  $X, Y$  and  $\mathbf{Z}$ , respectively. When  $X$  and  $Y$  are conditionally independent given  $\mathbf{Z}$ , then  $CMI(X, Y|\mathbf{Z}) = 0$ .

**Conditional mutual inclusive information (CMI2).** The CMI2 uses both KL divergence and interventional and is defined as:

$$\begin{aligned} CMI2(X, Y|\mathbf{Z}) &= \frac{DKL(P\|P_{X \rightarrow Y}) + DKL(P\|P_{Y \rightarrow X})}{2} \\ &= \sum_{x, y, z} p(x, y, z) \ln \frac{p(x, y, z)}{p(x, z) \sum_x p(y|z, x)p(x) + p(y, z) \sum_y p(x|z, y)p(y)}, \end{aligned} \quad (5)$$

where  $p(x, y, z)$  is the joint probability distribution of  $X$ ,  $Y$  and  $\mathbf{Z}$ ,  $P_{X \rightarrow Y} = P_{X \rightarrow Y}(X, Y, \mathbf{Z})$  and  $P_{Y \rightarrow X} = P_{Y \rightarrow X}(X, Y, \mathbf{Z})$  are the interventional probability distributions of  $X$ ,  $Y$  and  $\mathbf{Z}$  for removing edges  $X \rightarrow Y$  and  $Y \rightarrow X$ , respectively.  $DKL(P\|P_{X \rightarrow Y})$  and  $DKL(P\|P_{Y \rightarrow X})$  are KL divergences from  $P$  to  $P_{X \rightarrow Y}$ , and from  $P$  to  $P_{Y \rightarrow X}$ , respectively. Similar to CMI, the order of CMI2 is equal to the size of  $\mathbf{Z}$  ( $|\mathbf{Z}|$ ).

**PC algorithm based on conditional mutual information (PCA-CMI).** In PCA-CMI<sup>25</sup>, a network is initiated with a completely undirected graph. Then, through an iterative process, the skeleton gets updated as

Algorithm	Dataset		
	10 genes, 10 edges	50 genes, 77 edges	100 genes, 166 edges
PCA-CMI	0.03	0.03	0.05
CN <sub>Consensus</sub>	0.6	0.6	0.6
CN <sub>Ind.Test</sub>	(0.02, 0.05)	(0.02, 0.05)	(0.03, 0.05)
CMI2NI	0.03	0.04	0.06
OIPCQ <sub>θ1</sub>	0.05	0.02	0.03
OIPCQ <sub>θ2</sub>	0.01	0.05	0.05

**Table 1.** Threshold values for methods used on three sets of DREAM3-Yeast1-null-mutant dataset.

edges are removed based on the results of the independent tests between adjacent nodes. Finally, the algorithm makes the skeleton which is fully undirected.

Let  $S_i$  be a skeleton of  $i$ th order with  $i$  starting from  $-1$ . So,  $S_{-1}$  denotes a completely undirected graph from which the algorithm starts. For two adjacent nodes  $X$  and  $Y$  in  $S_{i-1}$ , a set  $V_{XY} = ADJ(X) \cap ADJ(Y)$  is defined where  $ADJ(X)$  being a set of adjacent vertices of  $X$  in  $S_{i-1}$ .  $CMI(X, Y|M)$  is calculated for each  $i$ -subset  $M$  of  $V_{XY}$ . For calculating  $MI(X, Y)$  and  $CMI(X, Y|M)$ , Eqs. (3) and (4) are used respectively. For removing the edge between two adjacent nodes  $X$  and  $Y$  in  $S_{i-1}$ ,  $CMI_{max}(X, Y|Z)$  as  $\max_M CMI(X, Y|M)$  and  $\theta$  as the threshold for independent test are considered. The edges for which  $CMI_{max}(X, Y|Z) < \theta$  are removed from  $S_{i-1}$ .

**CMI2NI: GRN inference method based on CMI2.** Given an expression dataset with  $n$  genes and  $m$  samples, CMI2NI infers its underlying GRN. In CMI2NI, after obtaining MI and CMI2 with Eqs. (3) and (5), the PCA-CMI algorithm was used to remove the (conditional) indirect edges from the complete graph. GRN inference is performed by removing those edges without strong causal regulations recursively until there is no change in the network topology. For more details of the CMI2NI algorithm, see<sup>19</sup>.

## Results

In this section, the performance of OIPCQ and OIPCQ2 are benchmarked against other well-known methods (PCA-CMI, CN and CMINI) using both simulated (DREAM project) and real data (SOS DNA and AML). The DREAM project is an in silico network challenge introduced in 2006. In this work, we used DREAM3 and DREAM4 datasets. DREAM3 contains three sub-challenges of size 10, 50 and 100 genes. Each sub-challenge contains five gold standard networks (Ecoli1, Ecoli2, Yeast1, Yeast2 and Yeast3) and for each, there are three gene expression sets (heterozygous knockdown, null-mutants (steady state) and trajectories (time courses)). Among these sets, Yeast1, which is a steady-state dataset, is used. DREAM4 contains three sub-challenges of size 10, 100 and 100-multifactorial, among which, we have used all five networks of the 100-multifactorial sub-challenge. The five gold standard networks of the 100-multifactorial sub-challenge have 100 genes and they have 176, 249, 195, 211 and 193 gold standard links, respectively. To benchmark the performance of OIPCQ and OIPCQ2 against well-known algorithms, True Positive (TP), False Positive (FP), True Positive Rate (TPR), Positive Predictive Value (PPV), False Positive Rate (FPR), False Discovery Rate (FDR), overall ACCuracy (ACC), F-measure and Matthews Correlation Coefficient (MCC) are calculated. They are defined as follows:

$$TPR = \frac{TP}{TP + FN}, FDR = \frac{FP}{FP + TP}, FPR = \frac{FP}{FP + TN}$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}, F - measure = 2 \frac{PPV \times TPR}{PPV + TPR},$$

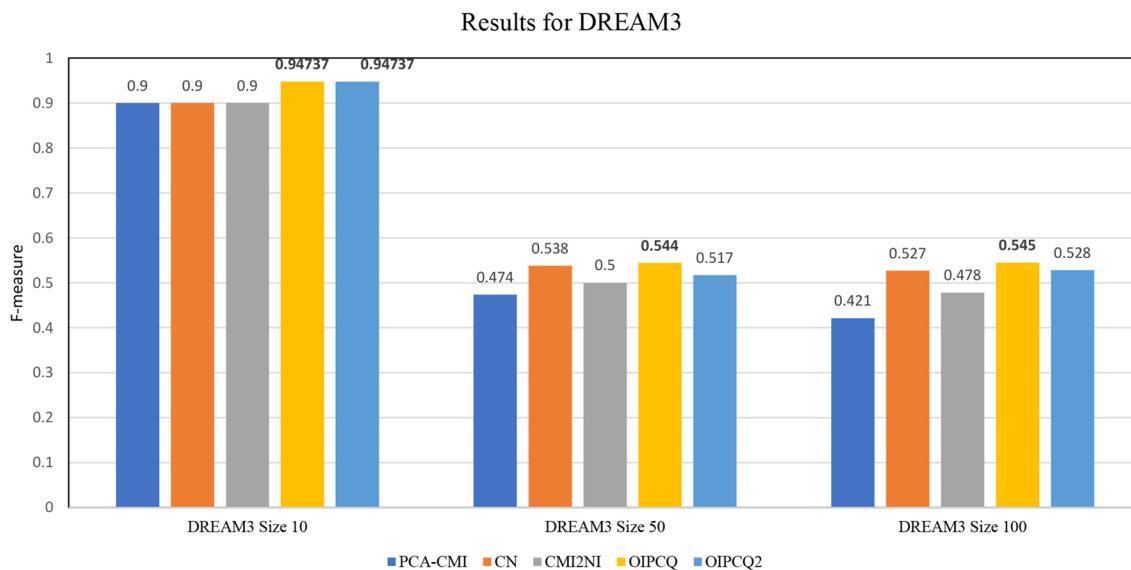
$$PPV = \frac{TP}{TP + FP}, MCC = \frac{TP \times TN - FP \times FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}.$$

**Results for DREAM3.** On the DREAM3 datasets, OIPCQ with two thresholds ( $\theta_1$  for MI and  $\theta_2$  for CMI) were implemented and compared with PCA-CMI, CN and CMI2NI. For OIPCQ and OIPCQ2 the same parameters ( $\theta_1$  and  $\theta_2$ ) are considered. For PCA-CMI and CMI2NI algorithms one parameter is considered. CN algorithm requires two thresholds, one for producing a consensus network (CN<sub>Consensus</sub>) and interval threshold for independent tests (CN<sub>Ind.Test</sub>). The selected thresholds for the mentioned methods are selected based on receiver operating characteristic (ROC) curve. These thresholds are shown in Table 1. The benchmark results for DREAM3 are summarized in Tables 2, S1 and S2 in Supplementary file. The F-measure values for mentioned algorithms are illustrated in Fig. 1. The results show that OIPCQ and OIPCQ2 consistently perform better than all other algorithms in terms of PPV, ACC, MCC and F-measure criteria. OIPCQ and OIPCQ2 algorithms were benchmarked with CMI2NI using DREAM3 datasets with sizes 10, 50 and 100. In all sizes, OIPCQ and OIPCQ2 had better performance compared to CMI2NI in terms of F-measure criteria. For size 10, FP was improved from 1 to 0 with no change in TP. For size 50, FP was decreased from 40 to 30 and TP was increased from 39 to 40. For size 100, FP was changed from 38 to 34 and TP was improved from 64 to 75.

We evaluated the performance of OIPCQ and OIPCQ2 algorithms in three orders (order 0, order 1 and order 2). The results suggest that a higher-order network has higher accuracy (ACC) and F-measure with a lower FPR

Algorithm	TP	FP	PPV	TPR	ACC	F-measure	FPR	FDR	MCC
PCA-CMI	9	1	0.9	0.9	0.95556	0.9	0.02857	0.1	0.87143
CN	9	1	0.9	0.9	0.95556	0.9	0.02857	0.1	0.87143
CMI2NI	9	1	0.9	0.9	0.95556	0.9	0.02857	0.1	0.87143
OIPCQ	9	<b>0</b>	<b>1</b>	0.9	<b>0.97778</b>	<b>0.94737</b>	<b>0</b>	<b>0</b>	<b>0.93541</b>
OIPCQ2	9	<b>0</b>	<b>1</b>	0.9	<b>0.97778</b>	<b>0.94737</b>	<b>0</b>	<b>0</b>	<b>0.93541</b>

**Table 2.** Results for DREAM3-size10-Yeast1. Best results are indicated in bold.



**Figure 1.** Comparison of F-measure values of OIPCQ and OIPCQ2 algorithms with other methods for learning DREAM3 Challenge with 10 genes, DREAM3 Challenge with 50 gene and DREAM3 Challenge with 100 genes.

	TP	FP	PPV	TPR	ACC	F-measure	FPR	FDR	MCC
Size10-Order 0	<b>9</b>	1	0.9	0.9	0.955556	0.9	0.028571	0.1	0.871429
Size10-Order 1	<b>9</b>	<b>0</b>	<b>1</b>	<b>0.9</b>	<b>0.977778</b>	<b>0.947368</b>	<b>0</b>	<b>0</b>	<b>0.935414</b>
Size10-Order 2	<b>9</b>	<b>0</b>	<b>1</b>	<b>0.9</b>	<b>0.977778</b>	<b>0.947368</b>	<b>0</b>	<b>0</b>	<b>0.935414</b>
Size50-Order 0	57	156	0.267606	0.74026	0.856327	0.393103	0.135889	0.732394	0.387026
Size50-Order 1	43	57	0.43	0.558442	0.925714	0.485876	0.049652	0.57	0.451002
Size50-Order 2	40	<b>30</b>	<b>0.571429</b>	<b>0.519481</b>	<b>0.945306</b>	<b>0.544218</b>	<b>0.026132</b>	<b>0.428571</b>	<b>0.515858</b>
Size100-Order 0	<b>98</b>	161	0.378378	0.590361	0.953737	0.461176	0.033654	0.621622	0.450085
Size100-Order 1	77	64	0.546099	0.463855	0.969091	0.501629	0.013378	0.453901	0.487513
Size100-Order 2	75	<b>34</b>	<b>0.688073</b>	<b>0.451807</b>	<b>0.974747</b>	<b>0.545455</b>	<b>0.007107</b>	<b>0.311927</b>	<b>0.545552</b>

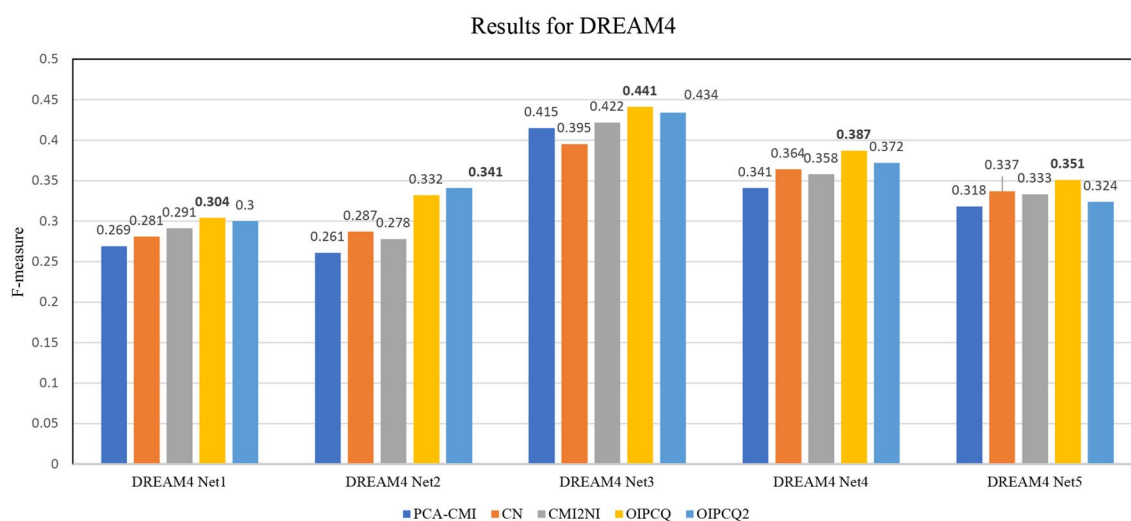
**Table 3.** Results for different orders of OIPCQ algorithm for DREAM3-Yeast1 of size10, 50 and 100. Best results are indicated in bold.

than that of a lower order network. This observation demonstrates that both OIPCQ and OIPCQ2 methods can construct the true network step by step, and are effective and efficient in inferring GRNs in order 2. The results of different orders of OIPCQ and OIPCQ2 algorithms for DREAM3 are shown in Tables 3 and 4, respectively.

**Results for DREAM4.** Similar to the DREAM3 case, for DREAM4, OIPCQ and OIPCQ2 with two thresholds ( $\theta_1$  for MI and  $\theta_2$  for CMI) were implemented and compared with PCA-CMI, CN and CMI2NI. The selected thresholds for the mentioned methods are shown in Table S3 in Supplementary file. The benchmark results for DREAM4 for five networks of the 100-multifactorial sub-challenge are summarized in Tables S4 through S8 and Fig. 2. Similar to DREAM3, the results show that OIPCQ and OIPCQ2 consistently perform better than all other algorithms in terms of PPV, ACC, MCC and F-measure metrics.

	TP	FP	PPV	TPR	ACC	F-measure	FPR	FDR	MCC
Size10-Order 0	<b>9</b>	1	0.9	0.9	0.955556	0.9	0.028571	0.1	0.871429
Size10-Order 1	<b>9</b>	<b>0</b>	<b>1</b>	<b>0.9</b>	<b>0.977778</b>	<b>0.947368</b>	<b>0</b>	<b>0</b>	<b>0.935414</b>
Size10-Order 2	<b>9</b>	<b>0</b>	<b>1</b>	<b>0.9</b>	<b>0.977778</b>	<b>0.947368</b>	<b>0</b>	<b>0</b>	<b>0.935414</b>
Size50-Order 0	57	156	0.267606	0.74026	0.856327	0.393103	0.135889	0.732394	0.387026
Size50-Order 1	40	52	0.434783	0.519481	0.927347	0.473373	0.045296	0.565217	0.436671
Size50-Order 2	39	<b>35</b>	<b>0.527027</b>	<b>0.506494</b>	<b>0.940408</b>	<b>0.516556</b>	<b>0.030488</b>	<b>0.472973</b>	<b>0.484925</b>
Size100-Order 0	<b>99</b>	166	0.373585	0.596386	0.952929	0.459397	0.034699	0.626415	0.449227
Size100-Order 1	80	64	0.555556	0.481928	0.969697	0.516129	0.013378	0.444444	0.501917
Size100-Order 2	75	<b>43</b>	<b>0.635593</b>	<b>0.451807</b>	<b>0.972929</b>	<b>0.528169</b>	<b>0.008988</b>	<b>0.364407</b>	<b>0.522602</b>

**Table 4.** Results for different orders of OIPCQ2 algorithm for DREAM3-Yeast1 of size10, 50 and 100. Best results are indicated in bold.

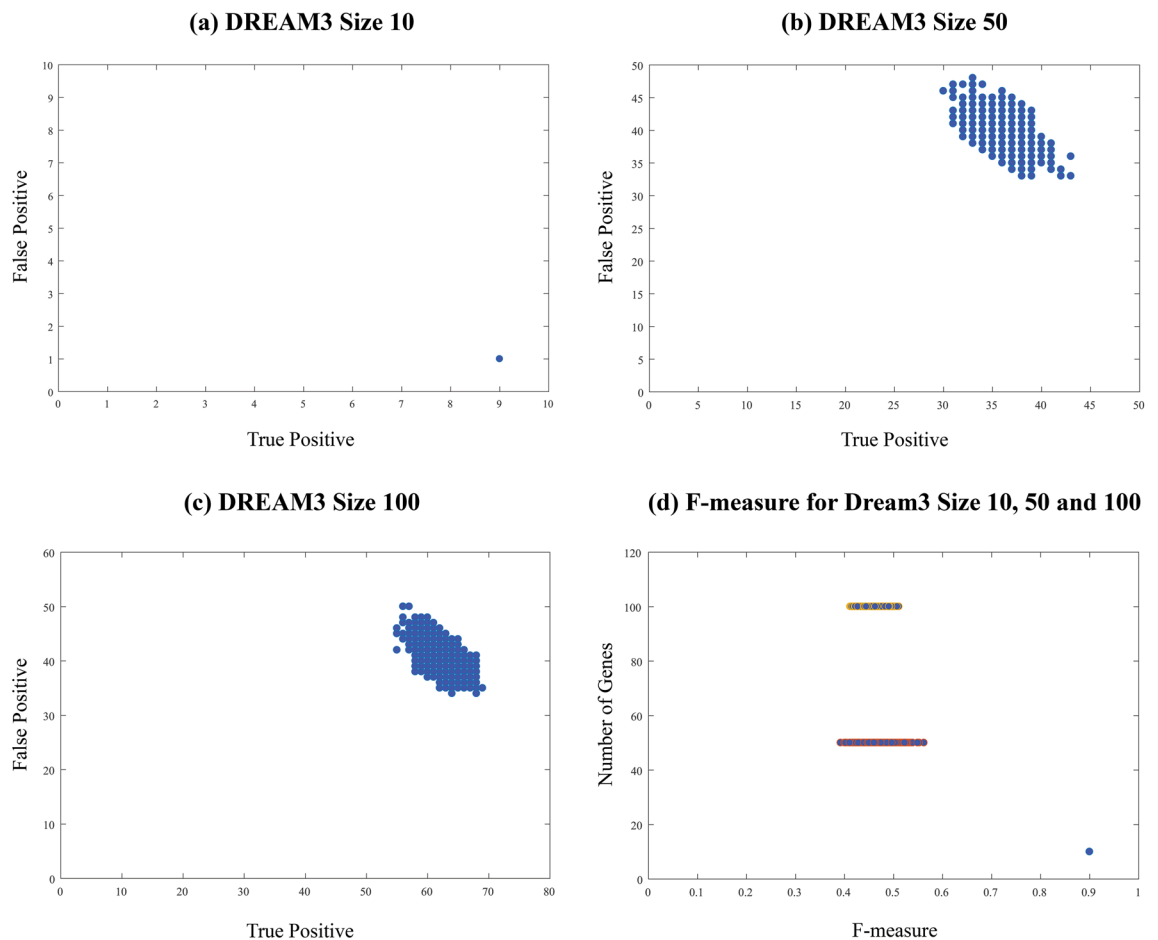


**Figure 2.** Comparison of F-measure values of OIPCQ and OIPCQ2 algorithms with other methods for learning DREAM4 challenge for five networks of the 100-multifactorial sub-challenge.

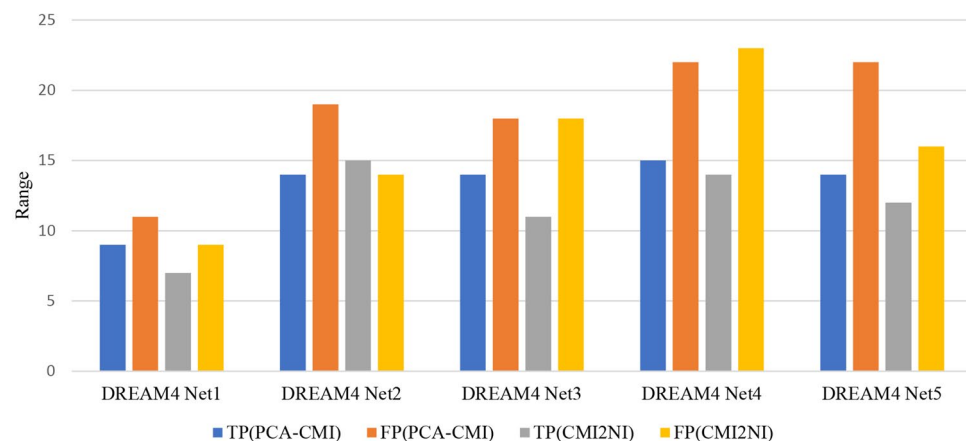
**Range of the variation of TP and FP values.** In the Materials and methods section, the order dependency of PC-based Algorithms is discussed. To illustrate the order dependency of PC-based algorithms, we implemented the PCA-CMI on the DREAM3 dataset with 10, 50 and 100 genes. For each of these sets, 1000 different gene order permutations were generated and tested. The threshold was set as  $\theta = 0.05$  for CMI tests. Figure 3 parts (a), (b) and (c) show the TP against FP for each randomly-generated permutation for DREAM3 dataset with 10, 50 and 100 genes, respectively. Figure 3 part (d) shows the F-measure values for these datasets. The study on different sequential node ordering resulted in different TP and FP. It is concluded that, by considering different sequential node ordering, the resulted networks are also different. Figure 3d illustrates that the larger the networks are, the more they are affected by the order of the input genes. The order dependency is less of a concern in networks with fewer variables (networks with less than 10 genes). Also, 1000 random sequences of genes for DREAM4 dataset are generated and the range of the variation of TP and FP are calculated. Figure 4 illustrates the range of values for DREAM4 datasets resulted by PCA-CMI and CMI2NI algorithms. It is concluded that, the range of variation of the values is significant and indicates the importance of using order-independent algorithm or selecting an appropriate order of genes as the algorithm input. The standard deviations for TPs and FPs are approximately equal to 2 and 3, respectively.

**Results for SOS-DNA and AML.** The efficiency of OIPCQ and OIPCQ2 algorithms are also tested on real datasets *E. coli*(SOS-DNA) and AML. The real network for SOS-DNA consists of a network with 9 genes and 24 edges. Thresholds used for the implemented algorithms are listed in Table S9 in Supplementary file. The benchmark results on the SOS-DNA dataset are presented in Fig. 5 and Table S10 in Supplementary file.

The results show that OIPCQ performs consistent with the benchmark algorithms. The AML network contains 81 genes, of which 65 are target and 16 are regulatory genes. RACER algorithm, presented in<sup>51</sup>, was developed to infer the GRN in AML dataset and is referred to as a Golden Standard for this dataset. Zhang's CMI2NI algorithm<sup>19</sup> produced a network with 549 edges, of which 113 are common with RACER. In order to compare the networks constructed by OIPCQ and OIPCQ2 with CMI2NI, we have selected thresholds such that the constructed network has equal edges to the CMI2NI's network. Hence, the produced network contains 549 edges from which 114 are in common with RACER. The results of OIPCQ and OIPCQ2 on AML dataset are



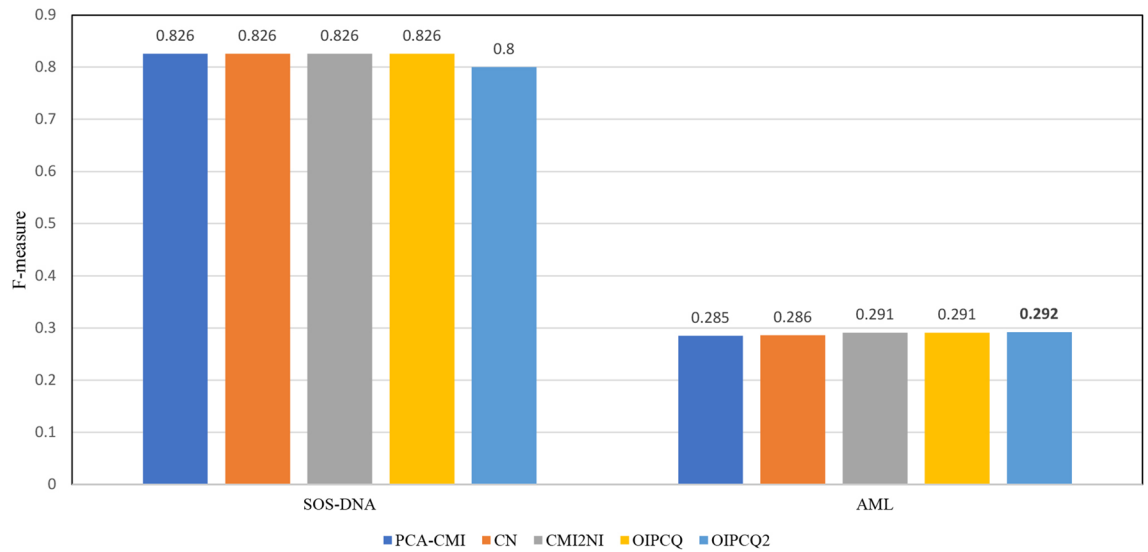
**Figure 3.** True Positive against False Positive for 1000 randomly-generated permutations for DREAM3 dataset with (a) 10 genes, (b) 50 genes, and (c) 100 genes resulted by PCA-CMI. The F-measure values for DREAM3 dataset with 10, 50 and 100 genes is represented in part (d).



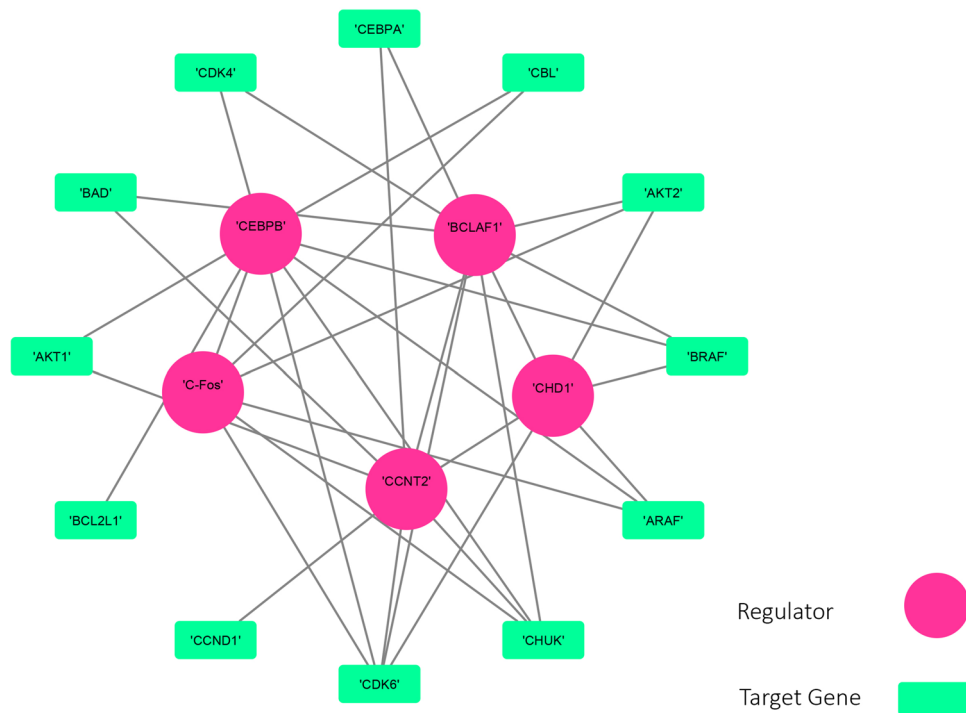
**Figure 4.** Range of the variation TP and FP values for 1000 randomly-generated permutation for DREAM3 and DREAM4 datasets resulted by PCA-CMI and CMI2NI algorithms.

summarized in Table S11 in the supplementary file. Figures S1 and Fig S2 generated by Cytoscape<sup>52</sup> show the GRN constructed by OIPCQ and OIPCQ2 on AML, respectively. The central nodes in these figures show the 16 regulators. In these networks, the highest degrees belonged to ZBTB7A and PU1 regulators with respective values of 53 and 47 (see Table S12). Figure 6 generated by Cytoscape<sup>52</sup> illustrates the resulted subnetwork by OIPCQ algorithm for the first 17 regulators and their targets of the AML dataset.

Results for SOS and AML dataset



**Figure 5.** Comparison of F-measure values of OIPCQ and OIPCQ2 algorithms with other methods for learning SOS-DNA and AML.



**Figure 6.** Subnetwork of GRN of AML that infer by OIPCQ.

These values are significantly higher compared to that of RACER network with 12 and 9 degrees. Previous studies<sup>53,54</sup> have shown that ZBTB7A functions as a transcriptional suppressor. ZBTB7A was also proven to play a critical role in AML as a transcription factor<sup>55</sup>. AML is also influenced by the slow decline of the transcript factor PU1<sup>56,57</sup>. In addition to the mentioned two regulators, BCLAF1 and NRSF are reported by Zhang et al.<sup>19</sup>, as significant regulators with significant role in cancer. In both of our networks, BCLAF1 and NRSF (Figs. S1 and Fig S2) had high degrees as well, which is consistent with the results of CMI2NI<sup>19</sup>. In order to verify these findings, the pathway enrichment was done separately for each of the four regulators ZBTB7A, PU1, BCLAF1 and NRSF, along with their target genes. The pathway enrichment was done in the cancer annotation system CaGe (<http://mgrc.kribb.re.kr/cage/>)<sup>58-61</sup>. The results of the pathway enrichments are presented in Tables S13 to S20, which include significant pathways. In these tables the obtained p-values related to the pathways correspond to

No.	Pathway (number of genes in pathways)	Regulator	Method	Genes overlapped	p value	q-value
1	CHRONIC MYELOID LEUKEMIA (73 genes)	BCLAF1	CMI2NI/OIPCQ/OIPCQ2	22/ <b>25/25</b>	2.90e-35/2.17e-41/2.17e-41	2.41e-32/1.80e-38/1.80e-38
2	ACUTE MYELOID LEUKEMIA (60 genes)	BCLAF1	CMI2NI/OIPCQ/OIPCQ2	18/ <b>20/20</b>	1.94e-28/3.03e-32/ 3.03e-32	5.38e-26/8.41e-30/ 8.41e-30
3	PATHWAYS IN CANCER ( 328 genes)	BCLAF1	CMI2NI/OIPCQ/OIPCQ2	28/ <b>31/31</b>	5.05e-30/2.41e-34/2.41e-34	2.10e-27/1.00e-31/1.00e-31
1	CHRONIC MYELOID LEUKEMIA (73 genes)	NRSF	CMI2NI/OIPCQ/OIPCQ2	18/ <b>26/26</b>	2.94e-29/8.90e-44/8.90e-44	2.45e-26/7.42e-41/ 7.42e-41
2	ACUTE MYELOID LEUKEMIA (60 genes)	NRSF	CMI2NI/OIPCQ/OIPCQ2	15/14/14	3.32e-24/6.78e-20/6.78e-20	1.38e-21/1.88e-17/1.88e-17
3	PATHWAYS IN CANCER (328 genes)	NRSF	CMI2NI/OIPCQ/OIPCQ2	22/ <b>28/28</b>	1.02e-23/5.47e-29/ 5.47e-29	2.83e-21/2.28e-26/2.28e-26

**Table 5.** Comparison of CMI2NI, OIPCQ and OIPCQ2 methods based on the three important pathways resulted by KEGG. Best results are indicated in bold. Among the significant pathways, three most significant ones were selected. The pathway enrichment was done separately for each of the four regulators ZBTB7A, PU1, BCLAF1 and NRSF. Column 1 indicates the name of pathways and number of genes in pathways. The name of regulators and algorithms are in columns 2 and 3, respectively. The overlap between resulted sets and genes in pathways are represented in columns 4. The related  $p$  values and  $q$ -values are collected in columns 5 and 6, respectively.

	PCA-CMI	CMI2NI	OIPCQ	OIPCQ2	OIPCQ <sub>q</sub>	OIPCQ2 <sub>q</sub>
DREAM3-size10	3.1476	3.3372	0	0	0	0
DREAM3-size50	9.7928	9.8719	3.1112	1.6733	4.5769	3.3982
DREAM3-size100	20.8978	21.6884	3.4808	0.9072	6.1006	5.1318
DREAM4-Net1	12.9749	12.7407	0.7739	0.8604	1.2504	1.1699
DREAM4-Net2	19.7166	20.9606	2.0103	5.0361	3.843	6.6629
DREAM4-Net3	19.1421	19.6161	1.7407	1.07	2.39	2.7508
DREAM4-Net4	19.7112	20.724	3.4289	2.5650	5.16	3.0038
DREAM4-Net5	21.9107	22.6012	1.6316	1.7157	2.4411	2.3825
SOS-DNA	3.881	3.927	0	0.3457	0	0.8439
AML	20.6249	18.0287	4.5840	3.9921	6.5335	6.1059

**Table 6.** Standard deviation for true positive based on different thresholds.

each of the four regulators and their target genes in the resulted networks from OIPCQ and OIPCQ2 algorithms are more meaningful than those of CMI2NI. In order to compare the importance of target genes of BCLAF1 and NRSF in OIPCQ and OIPCQ2 with CMI2NI, three most significant pathways were selected and compared (Table 5). The result of Table 5 indicate that the relationship between genes in our networks are more related to LEUKEMIA cancer in comparison with CMI2NI's network. In summary, based on the results obtained in this study, we can claim that the regulators ZBTB7A and PU1 beside to BCLAF1 and NRSF play a significant role in cancer, and especially in leukemia.

**Summary: all cases.** For threshold-dependent methods, TPR and FPR are used to generate the Receiver Operating Characteristic (ROC) curve. The area under the ROC Curve (AUC) is calculated to measure the performance of each method and benchmark them. To make a fair comparison, we referred to the algorithms' reference articles and used their suggested thresholds that have produced the best results based on F-measure values.

We also ran these algorithms on additional datasets. If a better threshold than what was suggested in the references was found, we included it in our paper for comparison. The rationale was to compare the best outcome of our algorithm with the best outcome of the benchmark algorithms in a fair manner.

To study and illustrate the dependency of the four algorithms (PCA-CMI, CMI2NI, OIPCQ and OIPCQ2) on the threshold for MI and CMI tests, the standard deviation of TP and FP based on different threshold values are calculated and shown in Tables 6 and 7, respectively. Among the four algorithms, the smallest standard deviation values for TP and FP are from OIPCQ and OIPCQ2 algorithms. For the aforementioned algorithms, the 1000 threshold values for MI and CMI tests are selected in the range (0,1) with the incremental step of 0.001. For each data, approximately 1000 different TP and FP are obtained based on different thresholds and standard deviation of them are calculated. To calculate the standard deviation of the results for OIPCQ and OIPCQ2, 1000 different threshold values for  $CMI(X, Y|Z)$  and  $CMI(X, Y|Z, W)$  are used. The standard deviations are calculated by considering a constant quantile of 70 and a constant value for the  $MI(X, Y)$ . In addition, for OIPCQ and OIPCQ2, the standard deviation of the TP and FP are also calculated based on different values for quantile  $CMI(X, Y|Z)$  and  $CMI(X, Y|Z, W)$ . For this purpose, the range of a quantile was (0.5, 0.9) with steps of 0.001 (400 steps) and fixed value of 0.05 for MI and CMI tests. The standard deviation of TP and FP based on different values for quantiles are shown in two last columns (OIPCQ<sub>q</sub> and OIPCQ2<sub>q</sub>) of Tables 6 and 7, respectively. In



	PCA-CMI	CMI2NI	OIPCQ	OIPCQ2	OIPCQ <sub>q</sub>	OIPCQ <sub>2,q</sub>
DREAM3-size10	1.1896	1.2965	0	0	0	0
DREAM3-size50	18.3173	24.27	9.0795	7.5631	10.551	8.1585
DREAM3-size100	44.3284	49.291	8.6476	6.4613	9.9375	7.8771
DREAM4-Net1	86.4018	91.0742	1.9737	1.3515	2.3155	2.0872
DREAM4-Net2	92.1005	95.4474	16.8821	31.3403	20.1067	33.2911
DREAM4-Net3	130.5841	140.8131	11.6187	8.1272	13.6606	10.914
DREAM4-Net4	128.34	137.1658	15.2308	10.1293	18.6175	13.1936
DREAM4-Net5	178.9869	189.3208	8.7945	7.2971	10.38	9.4407
SOS-DNA	1.0464	1.0947	0	0	0.1581	0.3616
AML	86.1101	77.3476	18.7795	16.3635	27.8585	25.7403

**Table 7.** Standard deviation for false positive based on different thresholds.

	PCA-CMI	CN	CMI2NI	OIPCQ	OIPCQ2
DREAM3-size10	0.1126	0.1072	0.1021	0.1097	<b>0.0943</b>
DREAM3-size50	0.0237	0.0243	0.0237	0.0231	<b>0.0211</b>
DREAM3-size100	0.0184	0.0196	0.02	<b>0.0177</b>	0.0182
DREAM4-Net1	0.014	0.1262	0.0122	<b>0.0098</b>	0.0104
DREAM4-Net2	0.0121	0.012	0.0125	<b>0.0111</b>	0.0125
DREAM4-Net3	0.0156	0.0142	0.0133	0.0132	<b>0.013</b>
DREAM4-Net4	0.0116	0.0109	0.0148	0.0127	<b>0.0099</b>
DREAM4-Net5	0.0131	0.137	0.0142	0.0145	<b>0.0119</b>
SOS-DNA	0.0379	0.0336	0.034	0.0363	<b>0.0326</b>
AML	0.0091	0.0096	0.0111	0.0116	<b>0.0073</b>

**Table 8.** Standard Deviation for F-measure based on removing 10% of the samples. Best results are indicated in bold.

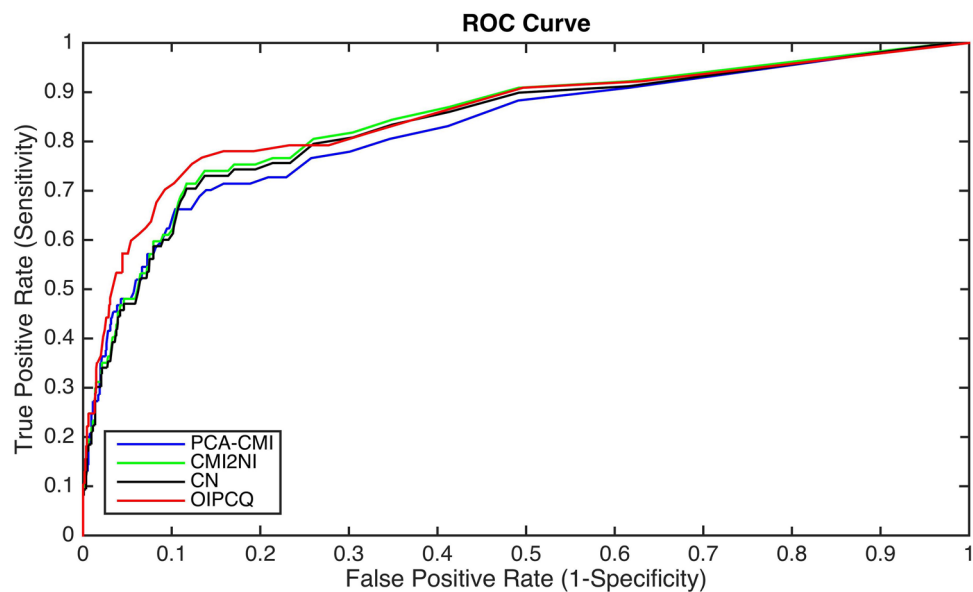
Dataset	FP	F-measure	FDR	MCC
DREAM3-size10-Yeast1	OIPCQ-OIPCQ2	OIPCQ-OIPCQ2	OIPCQ-OIPCQ2	OIPCQ-OIPCQ2
DREAM3-size50-Yeast1	CN	OIPCQ	CN	OIPCQ
DREAM3-size100-Yeast1	OIPCQ	OIPCQ	OIPCQ	OIPCQ
DREAM4-Net1	CN	OIPCQ	CN	OIPCQ
DREAM4-Net2	OIPCQ	OIPCQ	OIPCQ	OIPCQ
DREAM4-Net3	OIPCQ	OIPCQ	OIPCQ	OIPCQ
DREAM4-Net4	OIPCQ	OIPCQ	OIPCQ	OIPCQ
DREAM4-Net5	OIPCQ	OIPCQ	OIPCQ	OIPCQ
SOS-DNA	All algorithms	All algorithms	All algorithms	All algorithms
AML	OIPCQ	OIPCQ2	OIPCQ2	OIPCQ2

**Table 9.** Report Best algorithm for each data sets.

addition, to investigate the dependence of algorithms on sample values and evaluate the stability of the results by removing a percentage of samples, 10% of the samples are removed and new F-measure values are calculated. These steps are repeated 200 times and the standard deviations for resulted F-measure values are shown in Table 8. Results indicate that the algorithms are robust and have a relatively similar performance according to the standard deviation of F-measure values. In summary, by removing a small percentage of the samples, the results do not change significantly.

Table 9 shows the performance of algorithms on each dataset according to the important measures. For DREAM3-size50-Yeast1 and DREAM4-Net1 datasets CN algorithm superior to OIPCQ based on FP and FDR measures. Based on F-measure, which considers TP, FN and FP together, OIPCQ outperforms other algorithms in learning the GRN structure in all the tested data sets.

The OIPCQ and OIPCQ2 algorithms have three parameters:  $\theta_1$  (the threshold for MI test);  $\theta_2$  (the threshold for CMI test), and  $k$  as  $k$ th percentile of all  $CMI(X,Y|Z)$  values. In order to benchmark our algorithms against



**Figure 7.** ROC curves of different methods for DREAM3 challenge with 50 nodes. The red line is related to the ROC curve of OIPCQ algorithm with a AUC of 0.8458 value which has a larger value than other methods.

Method	PCA-CMI	CN	CMI2NI	OIPCQ
AUCD10	0.9642	0.9734	0.956	<b>0.9800</b>
AUCD50	0.8101	0.8315	0.834	<b>0.8458</b>
AUCD100	0.8419	0.8558	0.855	<b>0.8656</b>

**Table 10.** Comparison of different methods for learning DREAM3. Best results are indicated in bold. *AUCD10* AUC value for a 10-gene network in DREAM3, *AUCD50* AUC value for a 50-gene network in DREAM3, *AUCD100* AUC value for a 100-gene network in DREAM3.

Method	Net1	Net2	Net3	Net4	Net5
Team415	<b>0.75</b>	0.69	<b>0.76</b>	<b>0.77</b>	0.76
Team549	0.73	0.70	0.74	0.74	0.74
Team395	0.69	0.64	0.72	0.72	0.71
PCA-CMI	0.70	0.69	0.74	0.74	0.74
CN	<b>0.75</b>	<b>0.73</b>	<b>0.76</b>	0.70	0.76
OIPCQ	<b>0.75</b>	0.71	<b>0.76</b>	<b>0.77</b>	<b>0.77</b>

**Table 11.** Comparison of different methods for Learning DREAM4 Challenge. TeamName is the name of the team which registered for this challenge. The best performer for the relative item is noted in bold.

other methods, we set  $k = 70$  and  $\theta_1 = \theta_2 = \theta$ , and calculate AUC based on the  $\theta$  parameter. In our algorithms, by keeping the parameters constant and  $\theta_1$  and  $\theta_2$  equal, the performance of the algorithms declines. Despite the decline, they still outperform the benchmarked algorithms. Since the OIPCQ and OIPCQ2 algorithms are similar, we just report the result of OIPCQ algorithm. Results of AUC for DREAM3 of sizes 10, 50 and 100 are shown in Table 10. As an example, the ROC curves of different methods for the DREAM3 challenge with 50 nodes are shown in Fig. 7 which shows the better performance of the OIPCQ algorithm in comparison with the benchmarked methods (PCA-CMI, CMI2NI and CN).

Table 11 shows the results on DREAM4 data sets for different methods. The result of AUC values for OIPCQ algorithm is compared with that of PCA-CMI, CN and three best teams which participated on this challenge in <http://wiki.c2b2.columbia.edu/dream/results/DREAM4/>. From Table 11, we can find that the proposed method (OIPCQ) performs as good as the best method in DREAM4 challenge. In Networks 1,3 and 4, the results of the OIPCQ algorithm are similar to the best team (TEAM415). In Network 5, our algorithm has the best performance and in Networks 2, the CN algorithm among the challenge participants has the best result.

Method	PCA-CMI	CN	CMI2NI	OIPCQ
AUCSOS	0.79	0.791	<b>0.8</b>	<b>0.8</b>

**Table 12.** Comparison of AUC for real data sets (AUCSOS: AUC values for a SOS network with 9 genes). Best results are indicated in bold.

AUC values related to different algorithms for SOS is illustrated in Table 12. According to this table, the AUC values of CMI2NI and OIPCQ algorithms are larger than other those of methods.

According to Tables 10, 11 and 12, the AUC values of all algorithms are almost similar and AUC values of OIPCQ algorithm are larger than those of other methods.

## Discussion

Survival of living organisms depends on the interaction between thousands of genes. GRN are schematic representations of interactions among all gene pairs in a given cell. The functions and dynamics of various cells can be figured out through reconstructing the GRNs. In PC-based methods, the maximum of CMI values is used as a threshold for removing the network edges. Considering the distribution of the CMI values, the choice of the maximum value may not always be appropriate. For example, a single large value within a set of CMI values that are significantly lower, may result in high false positives. In our approach, the distribution of the CMI values is taken into account by choosing a certain quantile threshold. This quantile threshold is set based on the training process on DREAM3 dataset. This threshold is also applied to other datasets considered as independent data sets. In fact, this threshold can be adjusted for each dataset differently and better results can be obtained. In order to reduce the parameters and the computation time of the algorithm, this threshold is set based on the training process on DREAM3 dataset. In PC-based methods, the edges are removed in an iterative process until some criteria are met. In OIPCQ on the other hand, the edges are removed at the end of each order of algorithm, a threshold is determined and a number of edges are removed based on the selected threshold (Fig. 9).

The iterative process used in PC-based methods for removing edges from a network has two main drawbacks:

1-In each order of the PC-based algorithm and during each step in the iterative process, if an edge is removed in error, it will cause the error to propagate to the future steps at the same order of algorithm. In OIPCQ, on the other hand, the edges are removed at the end of each order of the algorithm. If an edge is removed in error at the end of each order, it will cause the error to propagate to the future orders of the algorithm and not the subsequent steps in each order of the algorithm. In other word, since in PC-based methods,  $U_{XY}$  and  $V_{XY}$  are updated in each iterative step (by removing edges in each iterative step, the size of  $U_{XY}$  and  $V_{XY}$  gets smaller), it is possible that the informative nodes in the separator sets are eliminated incorrectly. Subsequently, the test of independency considers the smallest set of vertices and therefore it is possible that an edge is retained by mistake (increasing FP). In OIPCQ, on the other hand,  $U_{XY}$  and  $V_{XY}$  are updated at the end of each order of the algorithm which mitigates this source of error.

2-The order of input variables has an impact on the final network constructed. In our simplified version of the algorithm, firstly, in each order set of neighbors of all adjacent nodes are determined. Then, the independency tests are performed and their test statistics are obtained as criteria for removing the edges. Such a process eliminates the chance for removing an edge in error that would cause an incorrect change in the set of neighbors. As a result, any order of inputs will result in the same network construction. The results we obtained on all tested datasets confirm that the number of FP's are reduced in our approach compared to the iterative approach. The drawback of the OIPCQ algorithm compared to PC-based algorithms is the running time of the algorithm. In fact, by removing edges in each iteration of PC-based methods the size of  $V_{XY}$  for the adjacent node  $X$  and  $Y$  can be decreased. So, the number of calculations for CMI tests is decreased. In the OIPCQ algorithm, first in each order of algorithm  $V_{XY}$  (for order 1) and  $U_{XY}$  (for orders greater than 1) are calculated and considered constant. As a result, more calculations are needed in the OIPCQ algorithm.

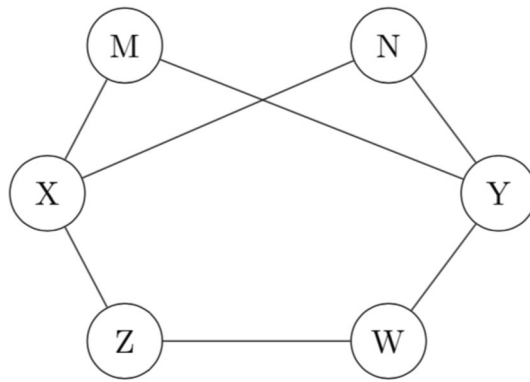
It can be concluded that OIPCQ and OIPCQ2 outperform other algorithms on simulated datasets. Also, the OIPCQ and OIPCQ2 results on the AML data shows more similarities with RACER compared to some popular inferring network methods. Similar results are concluded by using OIPCQ and OIPCQ2 algorithms, therefore, applying Eq. (5) instead of 4 for calculating CMI, the constructed GRN do not change the result significantly. Finally, the main advantage of the proposed method is that it is applicable to all PC-based methods.

## Methods

In this section, we discuss about drawbacks of PC-based algorithms. Following that, the details of the proposed OIPCQ algorithm for inferring the structure of GRN are introduced.

**Drawbacks of PC-based algorithms.** The first is that these methods are not robust for different sequential node orders. The second is that the results by these methods are highly dependent on the maximum value of CMI tests used for removing edges in each step of the algorithms. In addition, the proper selection of a collection of nodes which contains the separator sets significantly influences the performance of these methods.

*PC-based algorithms are order-dependent.* The network topology resulted from PC-based algorithms are dependent on the order of genes taken as input. In each order of PC-based algorithms, the edges are removed based on conditional mutual information tests which explained in section "PC Algorithm based on Conditional



**Figure 8.** Example of Paths of length 2 and greater than 2 between X and Y.

Mutual Information (PCA-CMI)”. Therefore, the adjacent of the vertices are updated as a result of sequential removing edges from the network. According to this method,  $V_{XY}$  depends on the initial order of the nodes and a different nodes order may eventually result in a different final network.

**Construction of separator sets.** In PCA-CMI, CMI2NI, and CN algorithms, the separator set is extracted from  $V_{XY}$ . So, these algorithms in each order only considers the paths of length 2 and ignores any existing connections with length greater than 2. One way of dealing with this constraint is to use  $U_{XY} = ADJ(X) \cup ADJ(Y)$  for order greater than one ( $i > 1$ ). For  $i > 1$ , by using  $U_{XY}$  instead of  $V_{XY}$ , the decision will be made by more information considering all the paths between X and Y. For example, in Fig. 8 by using  $V_{XY} = \{M, N\}$  only the  $CMI(X, Y|M, N)$  for order  $i = 2$  is calculated and only two paths of length 2,  $X - N - Y$  and  $X - M - Y$ , between X and Y are considered. By using  $U_{XY} = \{M, N, Z, W\}$ , we also considered the path of length 3,  $X - Z - W - Y$ , for checking the dependency between X and Y. The results show that, by considering more paths, OIPCQ helps to keep more reliable edges compared to other methods.

**Removing edges in PC-based algorithms is threshold-dependent.** In PC-based algorithms, the decision for removing edges from a network strictly depends on the value of  $CMI_{max}(X, Y|Z)$  and threshold  $\theta$  as the criterion for removing the edges. In fact, the edge  $XY$  is removed if for each Z in separator X and Y,  $CMI(X, Y|Z) < \theta$ . This method results in many FN. On the other hand, if the most  $CMI(X, Y|Z)$  are close to zero and only one of them is greater than  $\theta$ , PCA-CMI, CN and CMI2NI keep the edge  $XY$  in the network. Our investigation show that most of such edges are FP. So, considering the distribution of  $CMI(X, Y|Z)$ , quantile-based criterion for removing an edge is more effective and yield better results than using constant value as a threshold for removing edge. Our algorithm is trained by 70th percentile of all  $CMI(X, Y|Z)$  values in one dataset and this quantile is used for all datasets. In OIPCQ and OIPCQ2 algorithms, user sets the threshold.

**The OIPCQ algorithm.** The OIPCQ starts from a complete graph and iterates the following process to extract skeleton  $S_i$  from  $S_{i-1}$ .

**Step 0: Initialization:** Generate a complete network with number of nodes equal to the number of genes.

**Step 1: Calculate MI:** Compute MI values for each pair of genes.

**Step 2: Remove Edges:** Eliminate corresponding edges for which MI values are smaller than  $\theta_1$  ( $\theta_1$  denotes the threshold for MI test). The resulted network in this step is denoted by  $S_0$ .

**Step 3: Calculate CMI of order 1:** If there exists any edges between X and Y in  $S_0$ , find all genes Z which are adjacent to both X and Y, and then calculate their  $CMI(X, Y|Z)$  for Z belongs to  $V_{XY}$ . In this step, the paths of length 2 are considered between X and Y.

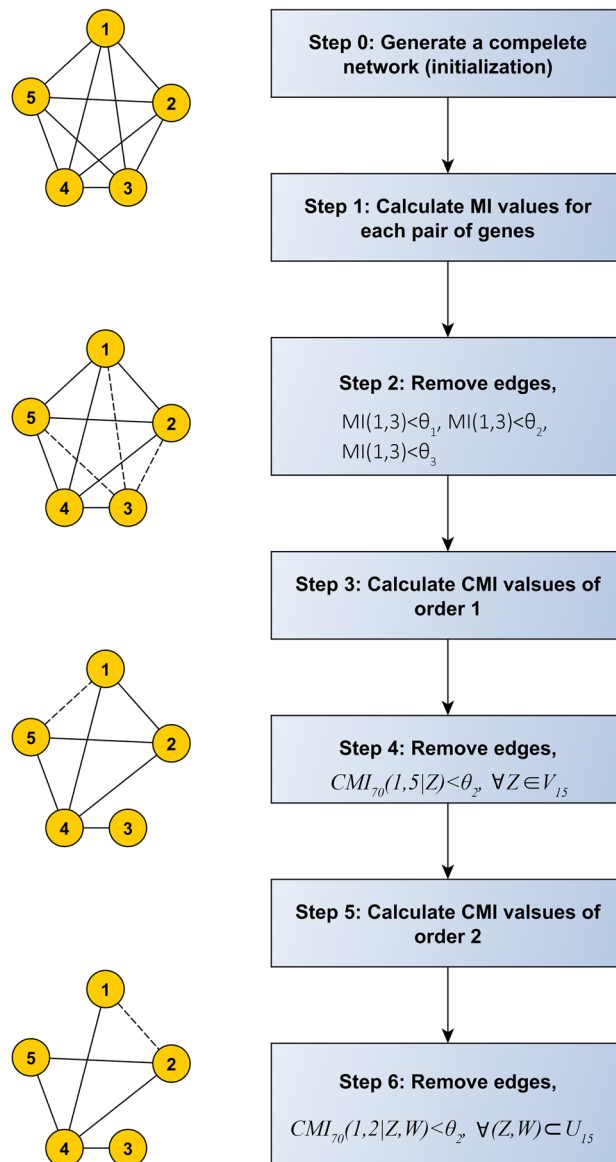
**Step 4: Remove Edges:** Define  $CMI_{70}(X, Y|Z)$  as all 70th percentile of all  $CMI(X, Y|Z)$  values. If  $CMI_{70}(X, Y|Z)$  is less than  $\theta_2$  ( $\theta_2$  is the threshold for CMI test), remove the edge between X and Y. The resulted network in this step is denoted by  $S_1$ .

**Step 5: Calculate CMI of order 2:** Do the steps above for X and Y and calculate  $CMI(X, Y|Z, W)$ , where Z and W belong to  $U_{XY}$ . In this step, the paths of length at least 2 are considered between X and Y.

**Step 6: Remove Edges:** Define  $CMI_{70}(X, Y|Z, W)$  as the 70th percentile of all  $CMI(X, Y|Z, W)$  values. If  $CMI_{70}(X, Y|Z, W)$  is less than  $\theta_2$ , remove the edge between X and Y.

An example of the OIPCQ algorithm is illustrated in Fig. 9 for network with 5 genes. The other version of OIPCQ, named OIPCQ2 was introduced using CMI2 for detecting dependency between genes. In both OIPCQ and OIPCQ2 algorithms, in each order of algorithms for each X and Y,  $U_{XY}$  and  $V_{XY}$  are defined and fixed. Then, at the end of each order the algorithm decides to remove edges based on threshold  $\theta_1$  and  $\theta_2$ . By using this method and fixing the  $U_{XY}$  and  $V_{XY}$  in each order of algorithms, the order dependency issue is solved and both OIPCQ and OIPCQ2 algorithms are order independent.

OIPCQ and OIPCQ2 algorithms compute  $MI(X, Y)$  in steps 1 and 2,  $CMI(X, Y|Z)$  in steps 3 and 4, and  $CMI(X, Y|Z, W)$  in steps 5 and 6.



**Figure 9.** An example of the OIPCQ algorithm. MI and CMI denote the mutual information and conditional mutual information.  $CMI_{70}(X, Y|Z, W)$  indicates the 70th percentile of the CMI values.  $V_{XY} = ADJ(X) \cap ADJ(Y)$  and  $U_{XY} = ADJ(X) \cup ADJ(Y)$ .

In PCA-CMI, CMI2NI, and CN algorithms, the separator set is extracted from  $V_{XY}$ . So, these algorithms in each order only consider the paths of length 2 and ignore any existing connections with lengths of greater than 2. One way of dealing with this constraint is to use  $U_{XY} = ADJ(X) \cup ADJ(Y)$  for orders greater than one ( $i > 1$ ). For  $i > 1$ , by using  $U_{XY}$  instead of  $V_{XY}$ , the decision will be made by more information considering all the paths between  $X$  and  $Y$ . In general, PC-based algorithms first consider a complete graph, then try to reduce the number of edges in the early steps to reach the desired network. In the early steps, such as  $CMI(X, Y|Z)$ , the computational time is less than the computational time for  $CMI(X, Y|Z, W)$ .

If  $CMI(X, Y|Z, W)$  has a low value for the 70th percentile of all  $Z$  and  $W$ ,  $CMI(X, Y|Z)$  also has a low value. These steps (first order one then second order) are performed to increase the speed of the algorithm and reduce the computational complexity.

Received: 24 June 2020; Accepted: 24 March 2021  
 Published online: 07 April 2021

## References

1. Marbach, D. *et al.* Revealing strengths and weaknesses of methods for gene network inference. *Proc. Natl. Acad. Sci.* **107**, 6286–6291 (2010).
2. Bansal, M., Belcastro, V., Ambesi-Impiombato, A. & Di Bernardo, D. How to infer gene networks from expression profiles. *Mol. Syst. Biol.* **3**, 78 (2007).
3. De Smet, R. & Marchal, K. Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.* **8**, 717 (2010).
4. Chai, L. E. *et al.* A review on the computational approaches for gene regulatory network construction. *Comput. Biol. Med.* **48**, 55–65 (2014).
5. Aghdam, R., Ganjali, M. & Eslahchi, C. Ipca-cmi: an algorithm for inferring gene regulatory networks based on a combination of pca-cmi and mit score. *PLoS ONE* **9**, e92600 (2014).
6. Hughes, T. R. *et al.* Functional discovery via a compendium of expression profiles. *Cell* **102**, 109–126 (2000).
7. Hurley, D. *et al.* Gene network inference and visualization tools for biologists: application to new human transcriptome datasets. *Nucleic Acids Res.* **40**, 2377–2398 (2011).
8. Friedman, N., Linial, M., Nachman, I. & Pe'er, D. Using Bayesian networks to analyze expression data. *J. Comput. Biol.* **7**, 601–620 (2000).
9. Marbach, D. *et al.* Wisdom of crowds for robust gene network inference. *Nat. Methods* **9**, 796 (2012).
10. Zhang, X. *et al.* Narromi: a noise and redundancy reduction technique improves accuracy of gene regulatory network inference. *Bioinformatics* **29**, 106–113 (2012).
11. Zhu, H., Rao, R. S. P., Zeng, T. & Chen, L. Reconstructing dynamic gene regulatory networks from sample-based transcriptional data. *Nucleic Acids Res.* **40**, 10657–10667 (2012).
12. Akutsu, T., Miyano, S. & Kuhara, S. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In *Bioinformatics'99*, 17–28 (World Scientific, 1999).
13. Zhu, D. & Li, H. Improved Bayesian network inference using relaxed gene ordering. *Int. J. Data Min. Bioinform.* **4**, 44–59 (2010).
14. Krämer, N., Schäfer, J. & Boulesteix, A.-L. Regularized estimation of large-scale gene association networks using graphical Gaussian models. *BMC Bioinform.* **10**, 384 (2009).
15. Menéndez, P., Kourmpetis, Y. A., ter Braak, C. J. & van Eeuwijk, F. A. Gene regulatory networks from multifactorial perturbations using graphical lasso: application to the dream4 challenge. *PLoS ONE* **5**, e14147 (2010).
16. Aghdam, R., Ganjali, M., Zhang, X. & Eslahchi, C. CN: a consensus algorithm for inferring gene regulatory networks using the Sorder algorithm and conditional mutual information test. *Mol. Biosyst.* **11**, 942–949 (2015).
17. Husmeier, D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics* **19**, 2271–2282 (2003).
18. Nair, A., Chetty, M. & Wangikar, P. P. Improving gene regulatory network inference using network topology information. *Mol. Biosyst.* **11**, 2449–2463 (2015).
19. Zhang, X., Zhao, J., Hao, J.-K., Zhao, X.-M. & Chen, L. Conditional mutual inclusive information enables accurate quantification of associations in gene regulatory networks. *Nucleic Acids Res.* **43**, e31–e31 (2015).
20. Chaitankar, V., Ghosh, P., Perkins, E. J., Gong, P. & Zhang, C. Time lagged information theoretic approaches to the reverse engineering of gene regulatory networks. *BMC Bioinformatics* **11**, S19 (2010).
21. Faith, J. J. *et al.* Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.* **5**, e8 (2007).
22. Luo, W., Hankenson, K. D. & Woolf, P. J. Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information. *BMC Bioinform.* **9**, 467 (2008).
23. Margolin, A. A. *et al.* ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics* **7**(1), 1–15 (2006).
24. Wang, K. *et al.* Genome-wide identification of post-translational modulators of transcription factor activity in human b cells. *Nat. Biotechnol.* **27**, 829 (2009).
25. Zhang, X. *et al.* Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics* **28**, 98–104 (2011).
26. Zhou, C., Zhang, S.-W. & Liu, F. An ensemble method for reconstructing gene regulatory network with jackknife resampling and arithmetic mean fusion. *Int. J. Data Min. Bioinform.* **12**, 328–342 (2015).
27. Villaverde, A., Ross, J. & Banga, J. Reverse engineering cellular networks with information theoretic methods. *Cells* **2**, 306–329 (2013).
28. López-Kleine, L., Leal, L. & López, C. Biostatistical approaches for the reconstruction of gene co-expression networks based on transcriptomic data. *Brief. Funct. Genomics* **12**, 457–467 (2013).
29. Basso, K. *et al.* Reverse engineering of regulatory networks in human b cells. *Nat. Genet.* **37**, 382 (2005).
30. Modi, S. R., Camacho, D. M., Kohanski, M. A., Walker, G. C. & Collins, J. J. Functional characterization of bacterial srnas using a network biology approach. *Proc. Natl. Acad. Sci.* **108**, 15522–15527 (2011).
31. Wang, J. *et al.* Reconstructing regulatory networks from the dynamic plasticity of gene expression by mutual information. *Nucleic Acids Res.* **41**, e97–e97 (2013).
32. Sumazin, P. *et al.* An extensive microRNA-mediated network of RNA–RNA interactions regulates established oncogenic pathways in glioblastoma. *Cell* **147**, 370–381 (2011).
33. Li, Z., Li, P., Krishnan, A. & Liu, J. Large-scale dynamic gene regulatory network inference combining differential equation models with local dynamic Bayesian network analysis. *Bioinformatics* **27**, 2686–2691 (2011).
34. Barabási, A.-L., Gulbahce, N. & Loscalzo, J. Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.* **12**, 56 (2011).
35. Rezaei Tabar, V., Zareifard, H., Salimi, S. & Plewczynski, D. Learning directed acyclic graphs by determination of candidate causes for discrete variables. *J. Stat. Comput. Simul.* **89**, 1957–1970 (2019).
36. De La Fuente, A., Bing, N., Hoeschele, I. & Mendes, P. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics* **20**, 3565–3574 (2004).
37. Saito, S., Hirokawa, T. & Horimoto, K. Discovery of chemical compound groups with common structures by a network analysis approach (affinity prediction method). *J. Chem. Inf. Model.* **51**, 61–68 (2010).
38. Meyer, P. E., Lafitte, F. & Bontempi, G. minet: Ar/bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinform.* **9**, 461 (2008).
39. Spirtes, P., Meek, C. & Richardson, T. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh conference on Uncertainty in Artificial Intelligence*, 499–506 (Morgan Kaufmann Publishers Inc., 1995).
40. Spirtes, P., Glymour, C. N. & Scheines, R. *Causation, Prediction, and Search* Vol. 81 (MIT Press, 2000).
41. Spirtes, P. An anytime algorithm for causal inference. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, 213–221 (Citeseer, 2001).
42. Zhang, J. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artif. Intell.* **172**, 1873–1896 (2008).

43. Colombo, D. *et al.* Learning high-dimensional directed acyclic graphs with latent and selection variables. *Ann. Stat.* **40**, 294–321 (2012).
44. Claassen, T., Mooij, J. & Heskes, T. Learning sparse causal models is not np-hard. arXiv preprint [arXiv:1309.6824](https://arxiv.org/abs/1309.6824) (2013).
45. Aghdam, R., Ganjali, M., Niloofar, P. & Eslahchi, C. Inferring gene regulatory networks by an order independent algorithm using incomplete data sets. *J. Appl. Stat.* **43**, 893–913 (2016).
46. Aghdam, R. *et al.* Inferring gene regulatory networks by pca-cmi using hill climbing algorithm based on mit score and Sorder method. *Int. J. Biomath.* **9**, 1650040 (2016).
47. Aghdam, R., Rezaei Tabar, V. & Pezeshk, H. Some node ordering methods for the k2 algorithm. *Comput. Intell.* **35**, 42–58 (2019).
48. Brunel, H. *et al.* Miss: a non-linear methodology based on mutual information for genetic association studies in both population and sib-pairs analysis. *Bioinformatics* **26**, 1811–1818 (2010).
49. Altay, G. & Emmert-Streib, F. Revealing differences in gene network inference algorithms on the network level by ensemble methods. *Bioinformatics* **26**, 1738–1744 (2010).
50. Darbellay, G. A. An estimator of the mutual information based on a criterion for conditional independence. *Comput. Stat. Data Anal.* **32**, 1–17 (1999).
51. Li, Y., Liang, M. & Zhang, Z. Regression analysis of combined gene expression regulation in acute myeloid leukemia. *PLoS Comput. Biol.* **10**, e1003908 (2014).
52. Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).
53. Liu, X.-S. *et al.* Zbtb7a acts as a tumor suppressor through the transcriptional repression of glycolysis. *Genes Dev.* **28**, 1917–1928 (2014).
54. Liu, X. *et al.* Somatic human zbtb7a zinc finger mutations promote cancer progression. *Oncogene* **35**, 3071 (2016).
55. Hartmann, L. *et al.* Zbtb7a mutations in acute myeloid leukaemia with t (8; 21) translocation. *Nat. Commun.* **7**, 11733 (2016).
56. Rosenbauer, F. *et al.* Acute myeloid leukemia induced by graded reduction of a lineage-specific transcription factor, pu. 1. *Nat. Genet.* **36**, 624 (2004).
57. Steidl, U. *et al.* Essential role of jun family transcription factors in pu. 1 knockdown-induced leukemic stem cells. *Nat. Genet.* **38**, 1269 (2006).
58. Park, Y.-K. *et al.* Cage: a web-based cancer gene annotation system for cancer genomics. *Genomics Inform.* **10**, 33 (2012).
59. Kanehisa, M. & Goto, S. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).
60. Kanehisa, M. Toward understanding the origin and evolution of cellular organisms. *Protein Sci.* **28**, 1947–1951 (2019).
61. Kanehisa, M., Furumichi, M., Sato, Y., Ishiguro-Watanabe, M., Tanabe, M. KEGG: integrating viruses and cellular organisms. *Nucleic Acids Res.* **49**(D1), D545–D551 (2021).

## Acknowledgements

Rosa Aghdam is a postdoctoral fellow at Faculty of Mathematical Sciences, Department of Computer Science, the Shahid Beheshti University under support of Iranian National Science Foundation (INSF97009558).

## Author contributions

R.A., S.H.M. and C.E. developed the method and performed computational and statistical analysis. R.A., S.H.M. and C.E. design and wrote the paper. R.A. and C.E. contributed in writing and editing the manuscript. All authors read and approved the final manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** is available for this paper at <https://doi.org/10.1038/s41598-021-87074-5>.

**Correspondence** and requests for materials should be addressed to R.A. or C.E.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021