BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 15 2014, pages 2142–2149
doi:10.1093/bioinformatics/btu182

Structural bioinformatics

Advance Access publication April 11, 2014

A comprehensive comparison of association estimators for gene network inference algorithms

Zeyneb Kurt[1], Nizamettin Aydin[1] and Gökmen Altay[2,*]

[1]Department of Computer Engineering, Yildiz Technical University, Davutpasa, 34220 Esenler, Istanbul, Turkey and [2]Department of Biomedical Engineering, Bahcesehir University, 34349 Besiktas, Istanbul, Turkey

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Gene network inference (GNI) algorithms enable the researchers to explore the interactions among the genes and gene products by revealing these interactions. The principal process of the GNI algorithms is to obtain the association scores among genes. Although there are several association estimators used in different applications, there is no commonly accepted estimator as the best one for the GNI applications. In this study, 27 different interaction estimators were reviewed and 14 most promising ones among them were evaluated by using three popular GNI algorithms with two synthetic and two real biological datasets belonging to *Escherichia coli* bacteria and *Saccharomyces cerevisiae* yeast. Influences of the Copula Transform (CT) pre-processing operation on the performance of the interaction estimators are also observed. This study is expected to assist many researchers while studying with GNI applications.

Results: B-spline, Pearson-based Gaussian and Spearman-based Gaussian association score estimators outperform the others for all datasets in terms of the performance and runtime. In addition to this, it is observed that, when the CT operation is used, inference performances of the estimators mostly increase, especially for two synthetic datasets. Detailed evaluations and discussions are given in the experimental results.

Contact: gokmen.altay@bahcesehir.edu.tr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on October 7, 2013; revised on February 15, 2014; accepted on March 11, 2014

1 INTRODUCTION

Gene network inference (GNI) algorithms are frequently used in the bioinformatics studies such as detecting the activator genes of the genetic diseases, determining the functions of the regulating and regulated genes, obtaining the drug targets of the medical cures, etc. The most important process of the GNI applications is to obtain the interaction scores among the cell molecules such as genes or proteins. In the GNI applications, the association score estimators generally use the gene expression datasets obtained from microarray data analysis to determine the interaction scores among each gene pair (Altay and Emmert-Streib, 2010a, b; Altay *et al.*, 2011; Olsen *et al.*, 2009). In this study, two synthetic and two real biological gene expression datasets are used for evaluation. Because the biological datasets involve different kinds of noise (caused by experimental processes and computational errors of microarray data analysis operation), it is difficult to obtain interaction scores among the cell molecules. If those scores cannot be estimated accurately, inference performance of the application would not be good enough whichever GNI algorithm is used.

In this study, several GNI algorithms are used to evaluate different association score estimators objectively. Each estimator is separately evaluated with three GNI algorithms and four datasets. In this manner, association estimators that impact the inference performance positively are presented to the researchers interested in this field. Additionally, we tried to illustrate whether the inference performance is mostly influenced by the GNI algorithms or by the association estimators. It is observed that inference performance of the GNI applications are mostly impacted from the association estimators, as given in the experimental results. Relevance Network (RelNet; Butte and Kohane, 2000), Accurate Cellular Networks (ARACNE; Margolin *et al.*, 2006) and Conservative Causal Core Networks (C3NET; Altay and Emmert-Streib, 2010a) are used as GNI algorithms. Because these are some of the most commonly used ones, they were selected for the comparison. Explanations of these GNI algorithms are given in the Section 4 of the Supplementary Material.

The purpose of this study is not to investigate the GNI algorithms, but it is to examine the association estimators that are the most important process of the GNI algorithms. Hence, 27 estimators from several association estimator studies, which are frequently cited, are reviewed from the literature. Among them, 14 most outstanding estimators are chosen in (Z.Kurt *et al.*, unpublished) to compare in this study. In-depth explanations are given in (Z.Kurt *et al.*, unpublished). This study differs from the previous study in several aspects. First of all, we examined almost all available association estimators in the previous review article to decide the most promising ones for using in GNI applications rather than performing common experiments. However, current study extends the preliminary review by involving several experiments, common datasets and evaluation measures. In the review article, the estimators were evaluated only by depending on the declared comparisons in the literature. However, this study uses several GNI algorithms to evaluate the inference performances of the estimators. In addition to our unpublished review article, a small subset of the estimators was compared in a previous narrow-scoped study (Kurt *et al.*, 2013) for only one GNI algorithm (C3NET) by using only synthetic datasets, while this study involves evaluation of

*To whom correspondence should be addressed.

14 estimators for three different GNI algorithms by using also real biological datasets.

1.1 Background

There are few studies that evaluate and review the association estimators (Hausser and Strimmer, 2009; Olsen *et al*., 2009; Simoes and Emmert-Streib, 2011). However, datasets and performance evaluation metrics used in these articles are generally different from each other.

Olsen *et al*. (2009) favored Spearman Correlation Coefficient (SCC) and Pearson Correlation Coefficient (PCC) for the noisy synthetic datasets. They also indicated that Miller–Madow (MM) outperforms the other estimators with noiseless datasets. Real biological genomics datasets tend to have noise; *hence, SCC is a good choice to use in the GNI applications. We also selected the second best estimator PCC that could obtain only the linear relationships*. Additionally, *Pearson-based Gaussian (PBG) and Spearman-based Gaussian (SBG) estimators*, which assume the joint distribution of the gene pairs as Gaussian function and based on PCC and SCC, respectively, *are also used in our study*. The PBG and the SBG obtain mutual information (MI) scores from the correlation scores of the PCC and the SCC with normality assumption.

Simoes and Emmert-Streib (2011) compared four MI-based estimators by using several synthetic datasets. The MM is observed as the best performing one in their study. Their performance evaluation metric and the datasets are different from the ones in (Olsen *et al*., 2009). *The MM is selected in our study because of its success in both studies of Olsen et al. (2009) and Simoes and Emmert-Streib (2011)*.

Hausser and Strimmer (2009) evaluated nine estimators by using small synthetic datasets created by using simple functional relationships. Shrinkage, Chao-Shen (CS) and Nemenman-Shafee-Bialek (NSB) estimators are obtained as the best performing ones. Computational complexity of NSB is indicated as significantly larger than that of the others. The Shrinkage estimator is not selected owing to the fact that it is the worst performing one in both studies of Olsen *et al*. (2009) and Simoes and Emmert-Streib (2011). *The CS is selected because it has simple implementation and less complexity*.

Paninski (2003) compared Best Upper Bound (BUB), Jackknife, Maximum Likelihood (ML) and MM estimators. It is reported that the BUB outperforms ML and Jackknife. The BUB performs better than MM when the sample size is close to the number of cells used in the discretization process. However, this condition is not applicable to MI-based GNI studies. *MM is given as a promising method in also (Paninski, 2003); therefore, we include it in our study*.

Daub *et al*. (2004) proposed and compared B-spline (BS) estimator, with BUB and Kernel Density Estimator (KDE) methods. The BS with any spline order outperforms the BUB. The BS with spline order <3 performs worse than the KDE. *Hence, the BS and the KDE are also selected for evaluation*.

The direct MI score estimator called as K-Nearest Neighborhood-2 (KNN-MI $^{(2)}$) was compared with several MI score estimators (Kraskov *et al*., 2004). It is claimed that, KNN-MI $^{(2)}$ can obtain MI score directly without using entropy. It does not include the entropy estimation error and bias differently from other MI-based techniques and can make more accurate estimations than the other methods (Numata *et al*., 2008; Papana and Kugiumtzis, 2008). *Therefore, the KNN-MI $^{(2)}$ is also selected for evaluation*.

The techniques that can eliminate the indirect relations were also reviewed. It is claimed that, higher order partial correlation and partial MI coefficients outperform their non-partial counterparts (Çakır *et al*., 2009; de la Fuente *et al*., 2004). Çakır *et al*. (2009) denoted that the n-th order partial PCC (PPCN) outperformed the PCC, first order partial PCC, BS and first order conditional BS estimators. *Hence, we decided to use the PPCN*.

Reshef *et al*. (2011) proposed Maximal Information Coefficient (MIC) and compared it with PCC, SCC, KNN and KDE by using synthetic datasets with several simple functional relationships. It is claimed that the MIC performs better than the others. However, in some studies, a method called HHG (Heller, Heller, Gorfine) is claimed to be better than the MIC (Heller *et al*., 2013). *Because it is more reliable and its implementation is easier than the MIC, the HHG is selected for evaluation*.

Suzuki *et al*. (2009) proposed the Least Square MI (LSMI) method. They reported that, LSMI outperforms PCC, KNN, KDE and Edgeworth estimators in several point of views. *Hence, we included the LSMI in our study*. However, we get the worst performance with the LSMI when we use the first synthetic dataset, which is the smallest one. Additionally, its runtime is one of the largest ones even for this small dataset. Therefore, the LSMI was excluded from this study and not evaluated for other datasets. Discussions of this case are given in the Section 3.

Finally, the chosen estimators for this study are PCC, SCC, PBG, SBG, PPCN, HHG, KDE, KNN-MI $^{(2)}$, MM with Equal Frequency (EF) discretization approach (MMEF), MM with Equal Width (EW) discretization approach (MMEW), CS with EF (CSEF), CS with EW (CSEW), BS with spline order 2 (BS2) and BS with spline order 3 (BS3).

1.2 Preliminaries

This study involves the most comprehensive review in this field by handling almost all available estimator comparisons and discussions in the literature. Some of the reviewed studies used only synthetic datasets with simple functional relationships rather than using gene expression datasets. Unlike the other studies, this study examines 14 estimators and compares their performances by using three different GNI algorithms and gene expression datasets. Two synthetic and two real biological gene expression datasets are used. The synthetic datasets involve artificial noise to simulate the real biological ones. Additionally, the estimators are compared with respect to whether Copula Transform (CT) is used or not. In the literature, there is no study that investigates the effects of the CT on the inference performance. Association estimators have more impacts on the performance than the GNI algorithms. We cannot say that only one estimator is always the best performing one. Comparisons of the estimators are performed by using F-score and Precision metrics. It is observed that using CT generally increases the performance of the estimators, especially for the synthetic datasets. In a few cases, CT rarely causes the decrease in performance for all of the datasets. Furthermore, increasing the sample size also increases the performance of the estimators.

At the end of the experiments, BS2, BS3, KDE, PBG and SBG estimators are the most promising ones for all datasets, in terms of both F-score and Precision metrics. When the runtimes of the estimators are also considered for comparison, the KDE is observed as an unsuitable method because of its large runtime. We conclude that BS2, BS3, PBG and SBG estimators are the most preferable approaches in terms of both performance and runtime. Additionally, optimal number of cells (bins) used during the discretization process of some MI-based estimators is also investigated. To the best of our knowledge, there is no study that investigates the optimal cell number. Association estimators and discretization techniques are briefly explained in Section 3 of the Supplementary Material. Section 2 of this article includes Materials and Methods. Experimental results of the estimators are given in Section 3. The conclusion and discussions are given in the last section.

2 MATERIALS AND METHODS

In this study, comparison and evaluation of the association estimators are performed according to utilization of CT by using RelNet, ARACNE and C3NET with two synthetic and one real biological *Escherichia coli* datasets and one real biological *Saccharomyces cerevisiae* yeast dataset. The synthetic datasets are generated by using simulator SynTReN (Van den Bulcke *et al*., 2006). The true network of the first three datasets belongs to a subnet of *E.coli* bacteria, which is a real biological gene subnetwork. True network of the last dataset is a subnetwork of *S.cerevisiae* yeast. In the first synthetic dataset there are 100 genes and 100 samples, while the second one includes 100 genes and 1000 samples. The real biological *E.coli* dataset involves 1146 genes and 524 samples, while the real biological *S.cerevisiae* yeast dataset involves 2760 genes and 247 samples.

The CT uses the ranking values of the data samples, instead of the original values. The CT normalizes the ranking values to be between $(-0.5, +0.5]$.

Proposed system is preceded as follows. First, from the gene expression datasets, the interaction scores of the gene pairs are obtained by using association estimators, resulting in an association matrix for each estimator. Then, the association matrices are given to three GNI algorithms to find out the inference performances of the estimators. Each GNI algorithm eliminates the statistically non-significant interactions (edges) in the final network in its own way. The RelNet is basis of other two GNI algorithms. In the RelNet, first a threshold value (I_0) is determined. Then, the edges with weights smaller than this value are eliminated from the net. The ARACNE involves two steps. First, it eliminates the edges by using I_0 value. Then, it uses data processing inequality approach to eliminate the candidate indirect interactions. The C3NET also eliminates the edges with respect to I_0. Then, it protects the maximum scored interaction and deletes other edges of each gene from the net.

The F-score used in the evaluations is given as follows:

$$F = \frac{2pr}{p + r} \tag{1}$$

where p and r denote *precision* and *recall*, respectively. They are given by the following equation:

$$p = \frac{TP}{TP + FP} \text{ and } r = \frac{TP}{TP + FN} \tag{2}$$

where TP is true positive, FP is false positive and FN is false negative. TPs denote the number of the edges that are inferred by the GNI algorithm and also actually exist in the true net. FPs are the edges inferred by the GNI algorithm but does not actually exist in the true net. FN edges actually exist in the true net but cannot be inferred by the GNI algorithm.

Using only truly and falsely inferred edges (TPs and FPs) is not sufficient to measure the performance fairly. Missing edges (FNs) in the final network should also be involved into the evaluation process. Because the F-score considers all kinds of edges, it is a more reliable measure for completely known networks (mostly the synthetic ones). Because of the missing information, the large number of FNs may misdirect the researchers for the incomplete networks (mostly for the real biological ones). Thus, precision is a more reliable measure for this kind of nets.

Moreover, in this study, optimal bin (cell) number of the discretization process for the estimators that require discretization was examined by using the C3NET and two synthetic datasets. The optimal number of bins can be taken as $\sqrt{N}$, where N is the sample size. Further details are given in Section 5 of Supplementary Material.

3 RESULTS

In this section, detailed evaluation and discussion of the association estimators are given according to utilization of CT. Firstly, the results of the experiments in the case of using CT are reported for each dataset separately. Then, the experimental results in the case of not using CT are given briefly.

3.1 Evaluation of the association estimators for the first synthetic dataset by using CT

The first dataset involves 100 genes and 100 samples and its true network belongs to a subnet of *E.coli* bacteria. For this dataset, barplot of F-score values of the estimators according to three GNI algorithms is given in Figure 1. In Figure 1 RN, AR and C3 abbreviations for RelNet, ARACNE and C3NET, respectively.

For the first dataset, according to the F-score values in the case of using ARACNE GNI algorithm, six association estimators are more prominent than the others with their higher F-score values. The PBG, SBG and KDE are observed as the best performing estimators. The BS2, BS3 and HHG estimators follow them with a slight difference. The worst performances are obtained with the PPCN, KNN and LSMI estimators.

In the case of using the C3NET GNI algorithm, the same six association estimators are the most promising ones. F-score values of the best performed six estimators (BS2, BS3, HHG, KDE, PBG, SBG) are close to each other. Among them, the KDE and BS3 slightly perform better than the others. Also in this case, the worst performing estimators are the PPCN, KNN and LSMI.

In the case of using the RelNet GNI algorithm, F-score results of the estimators are similar and significantly less than that of the other GNI algorithms. Because the RelNet cannot eliminate the indirect dependencies, its FP is much greater than that of the ARACNE and the C3NET. Hence, its F-score value is much less than that of the ARACNE and the C3NET. In this case, any comparison among the estimators might not be accurate enough.

Barplot of precision values of the estimators according to the three GNI algorithms is given in the Figure 2.

According to the Precision metric, in the case of using the ARACNE, six methods (PBG, SBG, KDE, BS2, BS3 and HHG) are the most promising ones. Those are the same six estimators, which are also the best performing ones with respect to F-score metric. The CSEF and the CSEW slightly follow the six best estimators. Although the CS follows these six estimators according to F-score results given in Figure 1, its performance

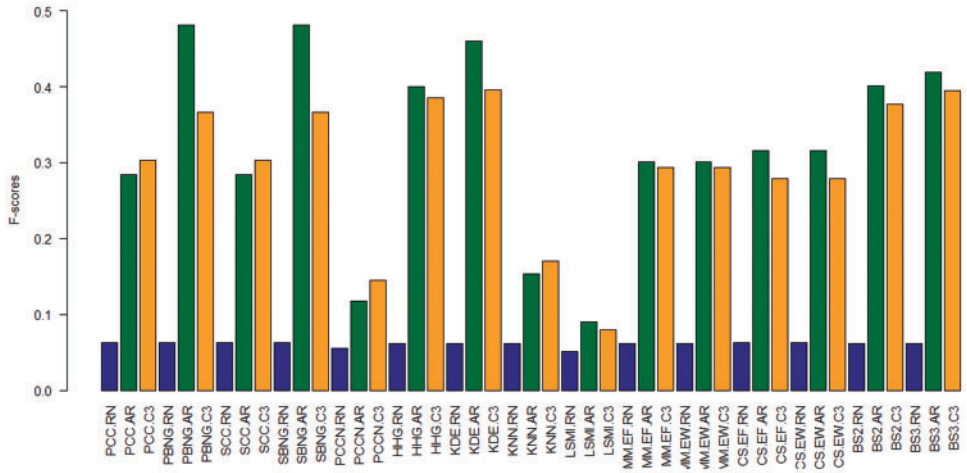

Fig. 1. Evaluation of the estimators for the first synthetic dataset by using CT

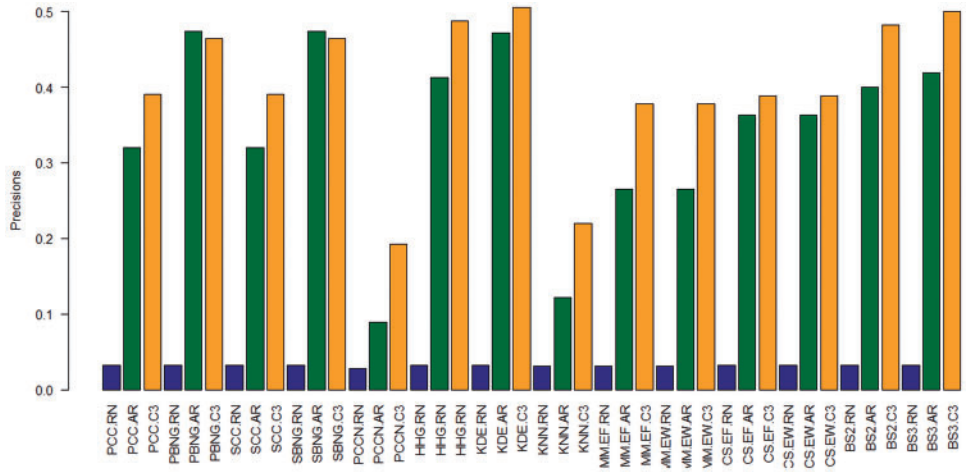

Fig. 2. Precision scores of the estimators for the first synthetic dataset by using CT

is not close to the best estimators in terms of F-score. The PPCN, KNN and LSMI estimators are the worst performing ones in terms of the Precision metric.

In the case of using the C3NET, comparison of the estimators in terms of the Precision metric is also given in Figure 2. The KDE, HHG, BS2, BS3, PBG and SBG estimators appear to be the most promising ones, as in the case of using F-score. The PCC and the SCC follow these six estimators. The PPCN, KNN and LSMI are the worst performing estimators.

In the case of using RelNet, the performances of all the estimators are close to each other and much less than the other GNI algorithms according to both metrics. Hence, any evaluation among the estimators by using RelNet would not be appropriate.

The BS2, BS3, HHG, KDE, PBG and SBG estimators are the commonly best performing ones according to both metrics for the first dataset. In the same experimental setup, the worst performing estimators are the PPCN, KNN and LSMI according to both metrics and also the worst result is obtained for the LSMI.

In the LSMI, two parameters, λ and σ, should be chosen optimally. Using grid search is suggested for the optimal parameter selection (Suzuki *et al.*, 2009). However, grid search for the LSMI method is computationally expensive. For instance, runtime of this method for the first dataset with 100 genes and 100 samples is >10 h in the R. The other datasets include more samples than the first dataset. Hence, running time of the LSMI would get larger with these datasets. Because it is the worst performing estimator, LSMI is excluded from the study and not implemented in C++.

3.2 Evaluation of the association estimators for the second synthetic dataset by using CT

The second dataset involves 100 genes and 1000 samples. True network of this dataset also belongs to a subnet of *E.coli* bacteria. Barplot of F-score values of the estimators for three GNI algorithms is given in Figure 3.

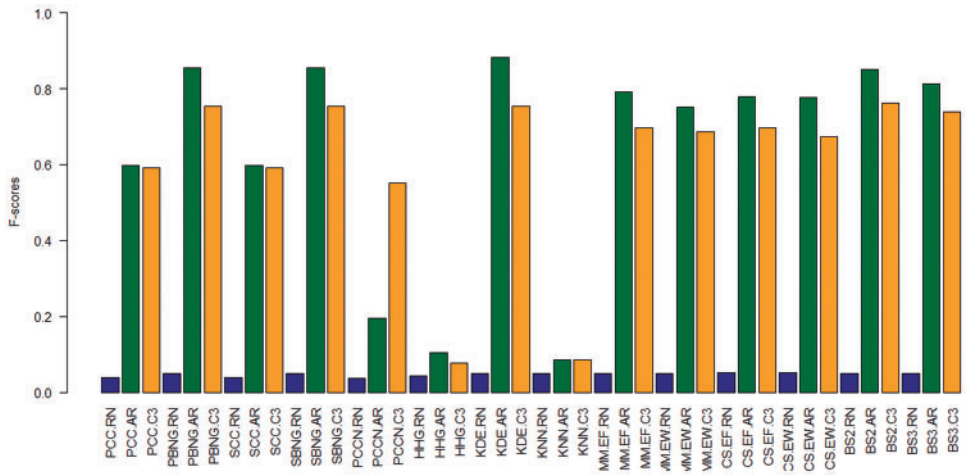

Fig. 3. Evaluation of the estimators for the second synthetic dataset by using CT

In the case of using the ARACNE algorithm, the BS2, BS3, KDE, PBG and SBG are observed as the most efficient estimators. The MM and CS estimators with both discretization techniques exhibit close performance to the best performing five estimators. The HHG exhibits significantly less performance, when the sample size is increased. The HHG, PPCN and KNN are the worst performing estimators in terms of F-score.

In the case of using the C3NET algorithm, five methods (BS2, BS3, KDE, PBG and SBG) exhibiting the best performances by using ARACNE also exhibit the best performances. Moreover, the MM and CS estimators with both discretization approaches perform similar to the best performing ones. The HHG, KNN and PPCN gave the worst F-score results.

The best performing five methods and the worst performing three methods are the same for both ARACNE and C3NET algorithms. The CS and MM exhibit higher performance and their F-scores can reach the performance of the best estimators, when the sample size is increased.

When the RelNet is used, all estimators perform close to each other and show significantly less performance than that of the other GNI algorithms, as in the case of using the first dataset. Therefore, we did not evaluate and discuss the estimators in the case of using the RelNet for also second dataset. The barplot of precision values of the estimators for three GNI algorithms is given in Figure 4.

According to precision metric, by using ARACNE the BS2, BS3, KDE, PBG and SBG are the best performing ones. The MM and CS with both discretization techniques show close performance to the best performing five estimators given above. These observations are the same as the results obtained with F-score.

By using the C3NET, the precision scores of the same five estimators (BS2, BS3, KDE, PBG and SBG) are larger than that of the others. Precisions of those five estimators are close to each other. The MM and CS also follow them with close precision values. Experimental results obtained by using F-score and precision metrics are the same as the results obtained by using the C3NET. Again, all of the estimators perform close to each other and show significantly less performance than that of the other GNI algorithms by using RelNet for both F-score and precision. Therefore, we did not evaluate and discuss the estimators for the RelNet.

Comparison results according to the precision metric favoring the BS2, BS3, KDE, PBG and SBG methods are compatible with the results obtained for the F-score metric for this dataset. The BS2, BS3, KDE, PBG and SBG estimators gave the highest and similar performance to each other with respect to both the F-score and Precision metrics. In addition to those five estimators, the HHG was one of the best performing estimators when the first dataset is used. Because increasing the sample size means increasing the information about the genes, the estimators exhibit higher performance and the interactions among the gene pairs could be extracted more accurately while the sample size is increased. Exceptionally, performance of HHG is reduced with increasing sample size. Hence, the HHG is not appropriate for larger datasets. Besides, performances of the CS and MM could reach to the performances of the best performing ones with increasing sample size. With smaller dataset, the CS and MM also follow the most promising estimators. However, their performances get closer to that of the most efficient estimators with larger dataset. In (Altay, 2012), it is already illustrated that, when the sample size of the dataset is increased up to a particular value, performances of the estimators also increase.

Finally, from the discussions in Sections 3.1 and 3.2 it is possible to infer that the best performing estimators are the same methods for both performance evaluation metrics and for both datasets. These are BS2, BS3, KDE, PBG and SBG estimators. The MM and CS estimators with both discretization techniques follow these estimators closely. F-score values obtained by using the RelNet, ARACNE and C3NET, R code runtimes, serial and parallel C++ code runtimes of the estimators by using CT for the first and second datasets respectively are presented in the Supplementary Tables S1 and S2. When we consider the runtimes together with the F-scores, BS2, BS3, PBG, SBG, (also CS) are the most promising ones. Although C++ codes run much faster than the R implementations, even C++ code runtime of KDE is significantly greater than the others. Therefore, it is not suggested to us in GNI applications.

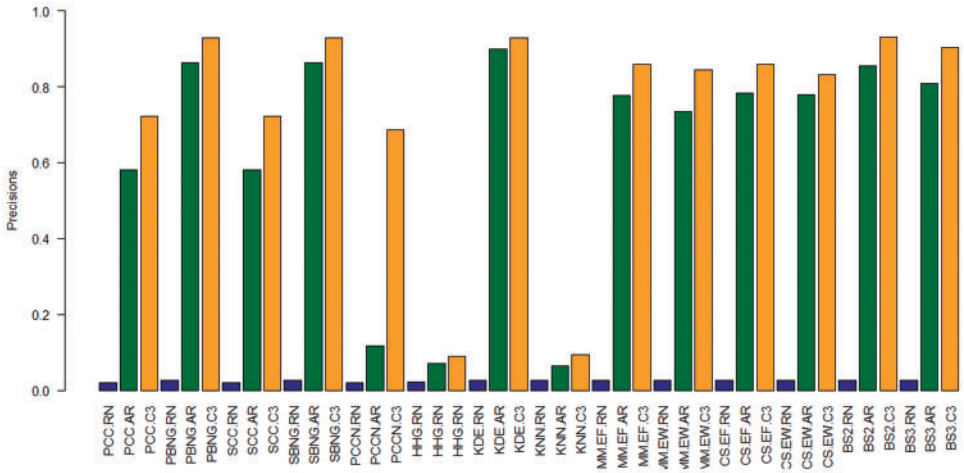

Fig. 4. Precision scores of the estimators for the second synthetic dataset by using CT

3.3 Evaluation of the association estimators for the real *E.coli* dataset by using CT

The real biological *E.coli* dataset is taken from the software package of Context Likelihood or Relatedness algorithm (Faith *et al.*, 2007). Reference network of the dataset is assembled mostly from the RegulonDB database, which provides information on regulation in *E.coli* (Gama-Castro *et al.*, 2008). The dataset involves 524 microarray chips and 1146 genes. There are 152 transcription factors (TF) among the genes. A total of 3091 interactions exist between the TFs and the regulated genes.

The association estimators are evaluated by using precision score given in Equation (2). The precision and F-score results of 14 estimators for three different GNI algorithms (RelNet, ARACNE and C3NET) are given in Supplementary Table S3. Although the F-score is commonly used to compare the estimators, true network of the dataset is not complete and only the known interactions are used. Therefore, instead of the F-score metric, the precision metric is chosen to evaluate the estimators for the real biological datasets. Incomplete knowledge of the true network causes the number of TP edges to be low. Hence, the F-score values also become much lower than the precision measure. Therefore, F-score might not provide a healthy comparison. Nonetheless F-score results of the methods are also given in Supplementary Table S3. Furthermore, from the Sections 3.1 and 3.2, *the precision metric is observed as an appropriate choice because its results are directly related and compatible with the results of the F-score metric for the synthetic datasets*. Precision barplot of the estimators for the real biological *E.coli* dataset is illustrated in the Supplementary Figure S1.

When the ARACNE is used, the CSEF and CSEW are distinguished from other estimators with a little difference. The BS2, BS3, MMEF, MMEW and KDE closely follow CS. Furthermore, PCC, SCC, PBG and SBG are the other methods that become prominent and perform close to the former ones. All the estimators except HHG, PPCN and KNN are observed as promising estimators.

When the C3NET is used, CSEF and CSEW are the most promising ones again. BS2, BS3, MMEF and MMEW follow the CS closely. Hence, CS, BS and MM become prominent at overall glance. The PCC, SCC, PBG, SBG and KDE seem similar and closely follow the former ones. Precision scores for the RelNet are increased, but still could not reach the other GNI algorithms for the first biological dataset. By using the RelNet, all the estimators except the HHG and PPCN perform almost equally. Because the RelNet could not eliminate indirect edges efficiently, its FP rates are higher than that of the others.

3.4 Evaluation of the association estimators for the real *S.cerevisiae* yeast dataset by using CT

Saccharomyces cerevisiae yeast dataset is taken from the Many Microbe Microarrays Database (M3D; Faith *et al.*, 2008). True network of the dataset is obtained from YeastNet v.2 database (Lee *et al.*, 2007). The dataset involves 247 samples and 2760 genes. There are 21 837 interactions between the gene pairs. The ratio of the edge numbers to the size of the *E.coli* truenet matrix is larger than that of the yeast, i.e. truenet of the *E.coli* is more complete than the yeast network. Hence, the inference performances obtained for the *E.coli* is greater than the performances obtained for the yeast dataset. We tried to run the HHG for evaluation but the runtime of the HHG took more than 8 days, and so we had to stop it. Therefore, we could not report the relevant results.

When ARACNE is used, KDE is distinguished from other estimators with a little difference. BS, CS, PCC, SCC, PBG and SBG are the other methods that become prominent and perform close to KDE. PPCN and KNN are the worst performing ones.

When C3NET is used, BS2, CS, PCC, SCC, PBG and SBG are the most promising ones. KDE follows them closely. Hence, BS, CS, KDE, PCC, SCC, PBG and SBG estimators become prominent at overall glance.

When the evaluation of the estimators is discussed for both synthetic and real biological datasets, the BS2, BS3, KDE, PBG and SBG methods stand out commonly. The results favoring the CSEF and CSEW together with the mentioned five estimators according to the precision metric for the first real dataset may

mislead the readers. Because it should be taken into account that F-score results of the CS could not be the highest one for the other datasets, still it is one of the best performing estimators. Apart from these, researchers should consider the runtimes of the estimators to decide between the BS, KDE, PBG and SBG estimators (*runtimes are given in Supplementary Tables S1–S4 for the datasets, respectively*). The KDE is not suggested to use because its runtime is significantly greater than the others. Because runtimes of the PBG and SBG are the smallest ones, they are preferable to use in the GNI applications. Moreover, runtime of the BS seems as acceptable. Therefore, the most preferable estimators are the BS, PBG and SBG according to the both performance and runtime.

Furthermore, the PPCN is observed as the worst one because when it eliminates the indirect edges, it also eliminates much of the TP edges incorrectly. The HHG also performs worse than the others when the sample size is increased.

Supplementary Table S5 involves the evaluation results of the estimators for all datasets more succinctly and clearly.

In addition to above discussions, according to the F-score metric, among the GNI algorithms the ARACNE performs a bit better than C3NET, and they significantly perform better than the RelNet. However, when the precision metric is considered, the C3NET performs a bit better than the ARACNE, and they perform better than RelNet significantly. Epsilon parameter is an important parameter affecting the performance of the ARACNE. In this study, epsilon was chosen to be 0. Furthermore, from the experiments and the figures it is observed that the estimators have more influence on the inference performance than the GNI algorithms. Readers should be aware of that the main goal of this study is to compare the estimators, not to compare GNI algorithms. Different GNI algorithms are used for comparing the estimators fairly according to different situations.

3.5 Evaluation of the estimators without using CT

In this section, general evaluation of the estimators is given in the case of not using CT. Supplementary Figures S3–S8 are used to illustrate the effects of the CT on estimators and give the comparison of the estimators for both cases (with and without the CT). The meanings of the bar colors and the abbreviations of the bar captions in the Supplementary Figures are given in the Section 2 of Supplementary Material.

Supplementary Figure S3 illustrates the F-score results of the first dataset with and without using CT together. Supplementary Figure S4 shows the precision results of the same dataset for both cases. Results according to the F-score and precision metrics are observed as compatible with each other. From the Figures, it is obvious that performances of the BS3, KDE and PBG, which were denoted as three of the best performing estimators with respect to both performance metrics, exhibit significantly better performance when the CT is used. Some of the other estimators (e.g. PCC, HHG) also significantly perform better by using CT. Hence, the CT is observed as a factor that generally increases the inference performances of estimators.

Supplementary Figures S5 and S6 illustrate the F-score and precision results of the second dataset, respectively, in the cases of using and not using CT. Using CT generally causes an increase in the inference performances of the estimators, also for this dataset. The performance increment for the second dataset is more significant for some of the estimators, while it is less significant for the others. Significant increments are observed for BS3, KDE and PBG estimators, which are among the best performing ones. These observations are the same and compatible with the observations for the first dataset.

Precision results of the first and second real biological datasets in the case of using and not using CT together are given in Supplementary Figures S7 and S8, respectively. The reason for using precision metric instead of F-score was given in the previous subsections. The results of precision and F-score are already observed as compatible and similar for the synthetic datasets. Hence, we can use precision for the evaluation. Although, the CT operation does not change the performance for some estimators, it generally causes an increase in the performance of the estimators for the real biological datasets. However, performance increments of the real datasets are less than that of the synthetic datasets. Moreover, the CT interestingly decreases the performance of a few estimators (PCC and PBG) for the real datasets. Nevertheless, CT generally increases the performances of the estimators for also these datasets. Finally, it is clearly observed that the CT is a factor that generally increases the performance of the estimators for all datasets. Hence, the positive effects of the CT should be taken into account by the researchers.

3.6 Runtime analysis of the association estimators

Estimators are implemented in both R and C++ languages. Implementing and parallelizing the estimators in C++ affect their runtimes, as given in Supplementary Tables S1 and S2. The PCC, SCC, PBG, SBG and PPCN methods were implemented by only using R's self-functions. Hence, they were already optimized and their runtimes were already expected as being smaller than the others. Implementing the other estimators in C++ significantly decreased their runtimes. Parallelized codes were run by using three cores of a quad-core machine. Runtimes of the HHG, KDE and KNN are reduced almost by half for the parallel implementation for all datasets. However, runtimes could not be reduced by parallel running for the BS, CS and MM with especially small sample size. Relative parallel runtimes of these estimators decreased slowly with increasing sample size. In the parallel implementation, data matrices are divided into partitions and sent to cores for calculation. Then, the result of each core is gathered into the main core, and finally result matrix is obtained. The parallelization does not decrease runtimes linearly with respect to core numbers. Runtimes will become less, if a machine with more cores can be used or if sample size of the dataset increases. Lastly, even parallel C++ code runtime of the KDE (i.e. one of the best performing methods) is significantly greater than the others. Thus, it is not suggested to use.

4 DISCUSSION

In this study, 14 association estimators and three GNI algorithms were evaluated by using two synthetic and two real biological datasets. Moreover, influence of using the CT on the inference performance of the estimators is also examined.

Changing the association estimator has a greater effect on the inference performances than changing the GNI algorithm. There is no single estimator that can be denoted as the best one. It is observed that the best performing estimators are common and compatible according to both the F-score and precision metrics, for all datasets. Analyses are firstly performed separately for each dataset, and then discussions are made commonly for all datasets. In this sense, for two synthetic datasets, *with respect to both F-score and precision metrics* by using CT, the most promising methods are the BS2, BS3, KDE, PBG and SBG. Precision score is used to evaluate the estimators when the real biological datasets are used. The BS2, BS3, CS, KDE, MM, PBG and SBG among MI-based approaches and PCC and SCC among correlation-based ones exhibit the best precision scores. The BS, CS and MM become slightly more prominent than the others for the first real biological dataset, while the BS, CS, KDE, PCC, PBG, SCC and SBG are the most promising ones for the second real biological dataset.

When the estimators are commonly evaluated with respect to all datasets, the *BS2, BS3, KDE, PBG and SBG* are observed as the best performing estimators. However, the runtime of the KDE is large. In the applications in which the runtime is more important than the precision of the estimators, the BS, PBG and SBG estimators should be preferred. The PPCN is commonly the worst one because it eliminates much of the TP edges incorrectly. The KNN is another commonly worst performing estimator. Moreover, the HHG surprisingly exhibits significantly less performance with increasing sample size. It can perform well for only the smallest dataset, but its performance become worse with other datasets.

Other observations obtained from the experiments are that inference performance gets better when the number of the samples in the dataset becomes larger. Furthermore, using the CT generally increases the inference performance of the estimators. Generally, the EF discretization technique results in better inference performances than the EW technique when the CT is not used. When the CT is used, EF and EW show similar performances, as expected.

Readers should keep in mind that the aim of this study is to compare and evaluate the association estimators, not to compare the GNI algorithms. However, some results about the GNI algorithms can be clearly seen from the figures in the experimental results. For example, the RelNet is the worst performing algorithm because it cannot eliminate the indirect interactions. Moreover, the ARACNE and the C3NET perform similar, while the C3NET gives significantly better precision scores than the ARACNE.

Furthermore, optimal average sample number in each bin during the discretization process is also examined. $\sqrt{N}$ is observed as a proper choice, where N is the sample size. Details are given in Section 5 of Supplementary Material. Lastly, implementing estimators in C++ results in less runtimes. Because the PCC, SCC, PBG, SBG and PPCN were implemented by using R's custom functions, they were not implemented in C++. Parallelized C++ codes make the HHG, KDE and KNN methods almost twice faster than their serial C++ codes. The parallel C++ code runtimes of BS, CS and MM are greater than their serial counterparts. Parallel runtimes could be reduced by using a dataset with more samples or a computer with more cores. Finally, we expect this comprehensive study is expected to assist many researchers studying in this field.

Funding:

Conflict of Interest: none declared.

REFERENCES

Altay,G. and Emmert-Streib,F. (2010a) Inferring the conservative causal core of gene regulatory networks. *BMC Syst. Biol.*, **4**, 132.

Altay,G. and Emmert-Streib,F. (2010b) Revealing differences in gene network inference algorithms on the network-level by ensemble methods. *Bioinformatics*, **26**, 1738–1744.

Altay,G. *et al.* (2011) Differential C3NET reveals disease networks of direct physical interactions. *BMC Bioinformatics*, **12**, 296.

Altay,G. (2012) Empirically determining the sample size for large-scale gene network inference algorithms. *IET Syst. Biol.*, **6**, 35–63.

Butte,A.J. and Kohane,L.S. (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac. Symp. Biocomput.*, **2000**, 418–429.

Çakır,T. *et al.* (2009) Metabolic network discovery through reverse engineering of metabolome data. *Metabolomics*, **5**, 318–329.

Daub,C.O. *et al.* (2004) Estimating mutual information using B-spline functions-an improved similarity measure for analysing gene expression data. *BMC Bioinformatics*, **5**, 118.

de la Fuente,A. *et al.* (2004) Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, **20**, 3565–3574.

Faith,J.J. *et al.* (2007) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, **5**, e8.

Faith,J.J. *et al.* (2008) Many microbe microarrays database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Res.*, **36** (**Suppl. 1**), D866–D870.

Gama-Castro,S. *et al.* (2008) RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nuclic Acids Res.*, **36** (**Suppl. 1**), D120–D124.

Hausser,J. and Strimmer,K. (2009) Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *J. Mach. Learn. Res.*, **10**, 1469–1484.

Heller,R. *et al.* (2013) A consistent multivariate test of association based on ranks of distances. *Biometrika*, **100**, 503–510.

Kraskov,A. *et al.* (2004) Estimating mutual information. *Phys. Rev. E*, **83**, 019903.

Kurt,Z. *et al.* (2013) Influence of the copula transform on the association estimators for gene network inference. In: *International Conference on Applied Informatics for Health and Life Sciences (AIHLS)*. Istanbul, Turkey, Sep 9–11, pp. 67–72.

Lee,I. *et al.* (2007) An improved, bias-reduced probabilistic functional gene network of baker's yeast, *Saccharomyces cerevisiae*. *PLoS One*, **2**, e988.

Margolin,A.A. *et al.* (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7** (**Suppl. 1**), S7.

Numata,J. *et al.* (2008) Measuring correlations in metabolomic networks with mutual information. *Genome Inform.*, **20**, 112–122.

Olsen,C. *et al.* (2009) On the impact of entropy estimation on transcriptional regulatory network inference based on mutual information. *EURASIP J. Bioinform. Syst. Biol.*, **2009**, 308959.

Paninski,L. (2003) Estimation of entropy and mutual information. *Neural Comput.*, **15**, 1191–1253.

Papana,A. and Kugiumtzis,D. (2008) Evaluation of mutual information estimators on nonlinear dynamic systems. *Nonlinear Phenom. Complex Syst.*, **11**, 225–232.

Reshef,D.N. *et al.* (2011) Detecting novel associations in large data sets. *Science*, **334**, 1518–1524.

Simoes,R.M. and Emmert-Streib,F. (2011) Influence of statistical estimators of mutual information and data heterogeneity on the inference of gene regulatory networks. *PLoS One*, **6**, e29279.

Suzuki,T. *et al.* (2009) Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, **10** (**Suppl. 1**), S52.

Van den Bulcke,T. *et al.* (2006) SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, **7**, 43.