Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Review Article

Turk J Elec Eng & Comp Sci
(2016) 24: 695 – 718
© TÜBİTAK
doi:10.3906/elk-1312-90

Comprehensive review of association estimators for the inference of gene networks

Zeyneb KURT[1], **Nizamettin AYDIN**[1,*], **Gökmen ALTAY**[2]
[1]Department of Computer Engineering, Yıldız Technical University, Esenler, İstanbul, Turkey
[2]Department of Biomedical Engineering, Bahçeşehir University, Beşiktaş, İstanbul, Turkey

Received: 11.12.2013 • **Accepted/Published Online:** 01.07.2014 • **Final Version:** 23.03.2016

Abstract: Gene network inference (GNI) algorithms allow us to explore the vast amount of interactions among the molecules in cells. In almost all GNI algorithms the main process is to estimate association scores among the variables of the dataset. However, there is no commonly accepted estimator to compute association scores for the current GNI algorithms. In this paper the association estimators that might be used in GNI applications are reviewed. The aim is to prepare a comprehensive and comparative review of all the important association estimators available in the literature. We performed this main aim by presenting, classifying, comparing, and discussing them to reveal which association estimator is more suitable for use in GNI applications by considering only the information available in the literature. Twenty-seven different estimators from various areas are investigated. The estimators were compared according to the GNI performances in the literature. The most promising association estimators for the GNI applications are suggested. As a result of the study, we identified eight promising methods for effective use in GNI. We expect this study to assist many researchers before using those estimators in their own GNI studies.

Key words: Association estimators, gene network inference (GNI) algorithms, classification of estimators, comparison of estimators

1. Introduction

Correlation between two random variables measures the dependency of those variables in terms of the degree of their association. The dependency between the variables is used in several application fields such as economics [1–3], signal processing [4–8], telecommunications [9,10], astrophysics [11,12], meteorology [13], statistics, and bioinformatics [14–21].

In this study, we aim to investigate all the important association estimators that might be used in gene network inference (GNI) algorithms. GNI algorithms play an important role in the bioinformatics field. They are widely used in the literature for illustrating the genome-wide associations among genes and gene-products such as proteins. For example, using GNI applications in pharmacological studies results in more reliable and cost effective products. In the literature, GNI techniques are mostly used in the following cases: finding the functions of relevant genes, regulating and regulated genes, drug targets, and biomarkers for the disease of interest, and so on.

GNI or reverse engineering of the gene networks elucidates the genome-wide interactions of genes by using gene expression values of the microarray datasets. The GNI process could be challenging due to the large-scale and noisy datasets. The association between gene pairs should be estimated efficiently and accurately, before

*Correspondence: naydin@yildiz.edu.tr

the gene network inference process. If this step is not correctly fulfilled then the ultimate inference process becomes erroneous for whichever GNI algorithm is used. Therefore, this is the most crucial process of any GNI algorithm. A block diagram regarding the usage of association estimators in GNI applications is given in Figure 1. In the first step, gene expression values are obtained from DNA microarray experiments from which the concentrations of mRNAs are quantified by appropriately converting spot color intensities into numerical values. Afterwards the expression ratios are normalized and a gene expression matrix is acquired. Each row in this matrix corresponds to one gene; each column corresponds to one sample. In order to obtain the interactions between gene pairs, this expression matrix is used. In the second step, direct mutual information (MI) estimators or correlation-based estimators compute the association score between all gene pairs. If the users prefer to use entropy-based estimators, they should perform the discretization process before the association estimation. The gene association matrix obtained at the end of the 2nd step is used by the GNI algorithm to infer the interaction net.

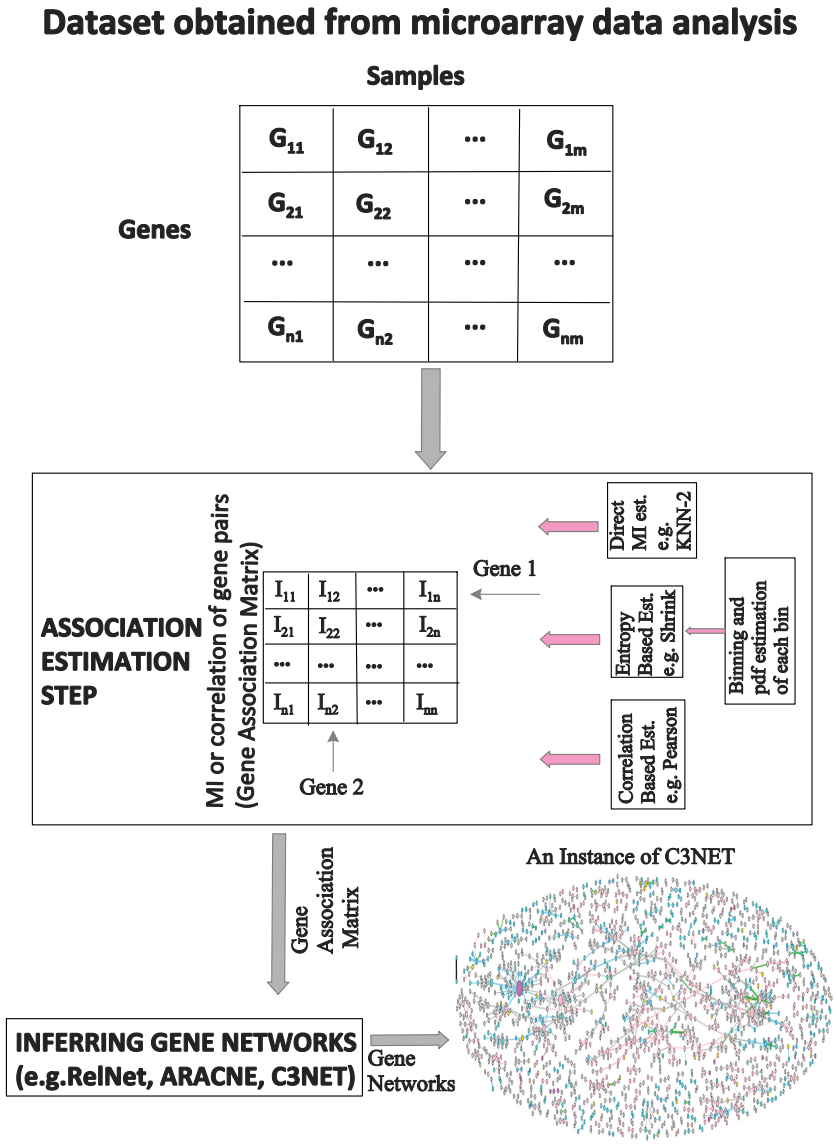

Figure 1. The block diagram of the GNI applications

In this study, rather than focusing on GNI algorithms, we investigate 27 different association estimators that can be used in GNI algorithms as the main process. Therefore, we reviewed correlation-based, entropy-based, and direct MI score estimators in the context whether they are used in genomics datasets or not. To the best of our knowledge, this is the most comprehensive review on this subject.

There are a few studies that review and classify the estimators [14,15,22–24]. The existing comparison studies do not include a large number of estimators. Among them, only a few studies make comparisons by using the gene expression datasets obtained from microarray data analysis [14,15]. At the end of the study, we will determine the most suitable and the best performing estimators that can be used in any of the available GNI algorithms regarding the current literature. Several studies including the estimators [1–24] are examined for this goal, not only in the genomics field but also in the several fields such as economics [1–3], signal processing [4–8], telecommunications [9,10], astrophysics [11,12], and meteorology [13], as denoted previously. Since we will determine the most promising estimators out of this study, any researchers who want to use the selected estimators in their GNI algorithm, need to reassess the inference performance before replacing their estimators. This is because although we propose some estimators that are promising according to the literature, they might have various performances in various datasets. This means that our suggestion is only based on the current literature information, which is certainly not complete. Basically, in this study, we are trying to make researchers aware that there might be better estimators for their GNI algorithms. Estimators will be presented and explained in detail in the Appendices of this review without the need for looking at the original references. Some explanatory examples are also provided in the Appendices.

Estimators can be evaluated by classifying according to several points of view. In this study two different classification strategies are followed. The first strategy classifies the estimators according to being parametric or not. The second classification strategy is based on whether the discretization process is required or not. Detailed explanations of the classification strategies will be given in Section 2.

At the end of this study, the most promising estimators are determined for using in GNI algorithms. The chosen estimators according to comparisons and discussions are as follows: least-squares mutual information (LSMI), K-nearest neighborhood (KNN) direct MI estimator-2, HHG (Heller, Heller, Gorfine), Miller–Madow, Spearman, Chao–Shen, B-spline estimators, and n-th order partial Pearson correlation coefficient.

The organization of the paper is as follows: estimators are classified in Section 2; comparisons of the previous studies and discussions of the estimators are given in Section 3; finally the conclusion is given in Section 4. The estimators are described in detail and examples for some of them are given in the Appendices.

2. Classification of the estimators

The reviewed estimators can be classified with respect to several aspects such as linearity/nonlinearity or being parametric/nonparametric. There are a limited number of studies in the literature that classify the estimators [22]. In [22], a small number of estimators are classified with respect to only being parametric or not, while far more estimators are classified with respect to several aspects in our study. Linear estimators assume that there is only a linear relationship between the variable pairs [14]. Among the reviewed estimators only Pearson correlation coefficient (PCC) and 1-st order and n-th order partial Pearson correlation coefficients (PPC^1 and PPC^n) satisfy this condition. The other estimators can also obtain the nonlinear relationship scores between the variables. Hence the number of the estimators is distributed as 3:24 according to being linear/nonlinear. Generally linear association estimators are insufficient to obtain the interaction scores between gene pairs. Hence, mostly nonlinear estimators are preferred despite their higher computational complexity instead of linear ones [14].

We classify the estimators according to two different strategies apart from linearity. Firstly we classify the estimators with respect to being parametric or nonparametric. Then we classify them according to requirement of the discretization process. The purpose of these classification strategies is to help readers to choose the most appropriate method for their dataset and conditions.

Parametric approaches assume that random variables or the relationship of the random variables have a particular distribution function [14,22]. If no information about the distribution of the dataset or distribution of the relationship is given, nonparametric approaches are preferable to parametric ones. For instance, PCC assumes that there is a linear relationship between the variables [14,15,17]. The abbreviations and the reference IDs of the reviewed association estimators are given in Table 1, while the classification according to being parametric or not is illustrated in Table 2. Abbreviations of the estimators, given in Table 1, are utilized in Table 2.

Table 1. Abbreviations and reference IDs of the reviewed estimators

Method	Abbreviation	Ref. ID	Method	Abbreviation	Ref. ID
Pearson Correlation Coefficient	PCC	1	An Analysis of Variance	ANOVA	15
Bayesian 1 (Jeffreys' prior)	Bayes1	2	Chao-Shen	CS	16
Bayesian 2 (Bayes-Laplace prior)	Bayes 2	3	B-spline	BS	17
Bayesian 3 (Schürmann-Grassberger, Perks prior)	Bayes 3	4	Kernel Density Estimator	KDE	18
Bayesian 4 (Minimax prior)	Bayes 4	5	K-Nearest Neighborhood (KNN) Entropy Estimator;	KNN Entr.	19
Edgeworth Expansion	Edgeworth	6	KNN direct MI Estimator-1	KNN MI-1	20
Least Squares Mutual Information	LSMI	7	KNN direct MI Estimator-2	KNN MI-2	21
First Order Partial Pearson Correlation Coefficient	(PPC1)	8	Best Upper Bound	BUB	22
n-th Order Partial Pearson Correlation Coefficient	(PPCn)	9	First Order Conditional Mutual Information	(CMI1)	23
Jackknife	Jackknife	10	Maximal Information Coefficient	MIC	24
Spearman Correlation Coefficient	SCC	11	Distance Correlation	dCor	25
Kendall Tau Correlation	Kendall	12	Heller, Heller, Gorfine	HHG	26
Maximum Likelihood (Empirical, Naive)	ML	13	James-Stein Shrinkage	Shrink	27
Miller Madow	MM	14			

The alternative classification is based on the requirement of the discretization process. Discretization is required when different distribution patterns exist in the dataset and they should be handled by separating the dataset into different cells. Moreover, discretization is required for the entropy-based approaches [23,25]. It is an extra operation accomplished before the association estimation. If the distribution does not change much through the dataset, this extra operation is not required.

Correlation-based methods such as PCC and Spearman correlation coefficient (SCC) do not need binning or discretization and directly obtain the correlation score [14]. Some other methods obtain the MI indirectly from entropy by discretizing the dataset [15,23], or by a more direct way without using entropy estimation [17]. Moreover, some methods exist that are aware of the partial correlation between the variables. Detailed explanations are given in Section 2.4.

Table 2. Classification of the estimators according to being parametric, nonparametric, and semiparametric

Classification of estimators		
Parametric	Nonparametric	Semiparametric
PCC Bayes 1 Bayes 2 Bayes 3 Bayes 4 Edgeworth LSMI PPC^1 PPC^n Jackknife*	SCC Kendall ML MM ANOVA CS BS KDE KNN Entr. KNN-MI-1 KNN-MI-2 BUB CMI1 MIC dCor HHG; Jackknife*	Shrink

*Jackknife can be parametric or nonparametric according to the chosen method in the leave-one-out technique.

Before the discussion of the comparison results, properties of the parametric, nonparametric, and semiparametric methods will be given in the following subsections. Furthermore, the alternative classification of the estimators according to being MI-based or correlation-based, i.e. according to discretization, is given in subsection 2.4.

Table 3. Alternative classification with respect to the binning process.

Correlation-based estimators		MI-based estimators		
		Entropy Estimators by Using Cell Probabilities		Direct MI estimators
Estimators which can only find Direct Dependency	Estimators which can also eliminate Inderect Dep.	Each cell consists of 1 sample	Each cell includes several samples	
PCC SCC Kendall Tau ANOVA dCor HHG	PPC^1 PPC^n	KDE KNN Entr. Edgeworth	ML MM Bayes 1 Bayes 2 Bayes 3 Bayes 4 CS Shrink BUB Jackknife CMI^1 BS MIC	KNN-MI-1 KNN-MI-2 LSMI

2.1. Properties of the parametric estimators

Parametric association and density estimators make assumptions about the underlying distribution of the random variable or the relationship and assume that the dataset or the relationship of the variables is from a known distribution function [14,22]. For instance, the Edgeworth expansion method assumes that the distribution of the dataset is Gaussian [22]. Parametric estimators may make too many assumptions, but they can define much more detail about the datasets. Parameters of the distribution can be obtained by using samples in the dataset. While searching the interaction between the random variables, several assumptions about the interaction between two random variables can be also made. The assumptions restrict the quantity of the relationship, but provide detailed information about the interaction. The approaches listed in the left column of Table 2 are parametric methods.

2.2. Properties of the nonparametric estimators

Nonparametric estimators are also known as distribution-free statistics. If anything else about the interested random variable is not known, nonparametric approaches should be used. The data do not have to belong to a known distribution family. Since there is less information about the data, nonparametric approaches are generally less powerful than the parametric ones.

In the nonparametric estimators, the relationships between two random variables are flexible. These relationships do not have to belong to a particular functional family. Hence we do not have any assumptions and restrictions about the relationships [14,15,22,23]. The approaches listed in the middle column of the Table 2 are nonparametric methods.

2.3. Properties of the semiparametric estimators

Semiparametric approaches possess some features of both parametric and nonparametric methods. The parametric segment of those approaches assumes that the distribution is known. The nonparametric part of those approaches does not have any assumption about the distribution or the relationship. Which segment of the semiparametric approaches will be more active can be arranged by a weighting parameter such as in the James–Stein shrinkage estimator [23]. Only one method, the James–Stein shrinkage estimator [23] among the reviewed 27 estimators belongs to the semiparametric estimators class.

2.4. An alternative classification of estimators

An alternative classification of the estimators with respect to whether being MI-based or correlation-based is given in this subsection. In other words, the alternative classification is done according to whether discretization (binning) is used or not used before the estimation process. A discretization operation is required when several cells of a dataset have different distributions and the entropy-based estimators are used.

Association metrics such as PCC and SCC are members of the correlation-based estimators class. They do not require separating the dataset into bins to compute the interaction scores of two random variables [14,24]. Correlation-based estimators mostly obtain only a linear relationship value between gene pairs from 2nd moments of the data by using all of the samples of each gene. Correlation-based estimators are also separated into two classes:

- Estimators that only determine the linear or nonlinear direct dependencies (see Table 3)
- Estimators that also eliminate the indirect dependencies (see Table 3)

In addition to the correlation-based estimators defined above, the association between random variables (e.g., genes) can be measured by MI. Measuring with the MI requires entropy estimations based on the cell (bin) frequencies mostly [14,15,25]. Hence, entropy estimator approaches can be a class of estimators that needs binning (discretization) of the dataset. Most of the entropy-based estimators depend on the histogram approach [15,23,25]. Discretization techniques and histogram-based approaches are mentioned in the Appendix Section A.2. Entropy estimators are divided into two groups according to the number of members in the cells after the binning process:

- Each cell can include several data samples (see Table 3),
- Each cell consists of only one sample; thus probability density over a sample is obtained, and then those are summed up (see Table 3).

In addition to correlation-based and entropy-based estimators, another class of estimators consists of methods estimating the MI directly. Therefore, MI-based estimators can be classified into two groups as entropy-based and direct MI estimators. This alternative subclassification is also included in Table 3. Abbreviations of the estimators, given in Table 1, are utilized in Table 3.

3. Comparison of the estimators and discussion

Association estimation between random variables can be useful in several applications such as GNI algorithms in bioinformatics research. In GNI applications, the interaction and relationship between the gene pairs should be efficiently obtained by correlation estimators as mentioned previously.

Comparisons of the reviewed 27 estimators are given in Table 4. The reference ID numbers of the estimators, given in Table 1, are utilized in Table 4 and numbers in the first column and the first row refer to these reference IDs. If a cell in Table 4 denoted by the i-th row and j-th column has a "$>$" sign, it means estimator i is better than estimator j in terms of estimation performance of the relationship. If the cell includes a "$<$" sign, it means the performance of the i-th estimator is worse than that of the j-th one.

Sign "$\approx$" means that the performances of the i-th and j-th estimators are similar and close to each other.

3.1. Analysis of ML, MM, PCC, SCC, shrinkage, and Schürmann–Grassberger estimators

Olsen et al. [14] and Simoes and Streib [15] compared the association estimators with respect to the inference performance in GNI applications. They also investigated the effect of the discretization techniques on estimation performance. The discretization process and methods are mentioned in the Appendix Section A.2. Olsen et al. denoted that the equal frequency technique performs better than the equal width technique. However, Simoes and Streib claimed that the equal width is better than the other one. In our opinion, separating the dataset into cells with similar frequencies appears to perform better when the distribution is unknown; hence the equal frequency is expected to perform better than the other one.

Olsen et al. also evaluated the performances of empirical (maximum likelihood-ML, naïve) [14,15,23,25], Miller–Madow (MM) estimator [26–29], and James–Stein shrinkage estimators, and PCC [14,30–36] and SCC [14,31,37–42] methods in terms of inference performance of GNI algorithms. They used accurate cellular networks (ARACNE [17]), context likelihood or relatedness (CLR), and maximum relevance/minimum redundancy network (MRNET) for GNI. In [14], 12 synthetic datasets were generated by the SynTReN [43]. The number of

Table 4. Comparison of the reviewed 27 estimators in the literature.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1							<	<	<		<	<	>*	>*			<	<	<		<		<	<	<	<	
2			>	<	<								>	>		<							<				<
3		<		<	<								>	>		<							<				<
4		>	>		<								$>*^4$,$<*^3$	$>*^4$,$<*^3$		<							<				$<*^4$,$>*^3$
5		>	>	>									>	>		<							<				<
6							<						$>*^6$							$>*^6$							
7	>					>												>			>						
8	>								<								>						>*2				
9	>							>									>						>				
10																						<					
11	>												>*	>*				>		>				<	<	<	
12	>																										
13	<*	<	<	$<*^4$,$>*^3$	<	$<*^6$					<*			<								<					$<*^4$,$>*^3$,>*
14	<*	<	<	$<*^4$,$>*^3$	<						<*		>									<					$<*^4$,$>*^3$,>*
15	>																										
16		>	>	>	>																						≈
17	>							<	<									$>*^5$				>	<				
18	>						<				<						$<*^5$			<		>		<	<	<	
19	>																			<	<						
20						$<*^6$					<							>	>		<			<	<	<	
21	>						<												>	>							
22										>			>	>			<	<									
23	>							<*2	<								>										
24	>										>							>		>					<	<	
25	>										>							>		>				>		<	
26	>										>							>		>				>	>		
27		>	>	$>*^4$,$<*^3$	>								$>*^4$,$<*^3$,<*	$>*^4$,$<*^3$,<*		≈											

The meaning of the signs in the Table is mentioned in detail at the Discussion section. Briefly they mean:
"*" denotes that the performance ranking is changing according to the noise existence in the dataset in the study of Olsen et al.
"*2" denotes that mostly (PPC1) > (CMI1) but in a few experiments ranks of the best second (PPC1) and the third (CMI1) estimators can be exchanged in the study of Çakır et al.
"*3" denotes the performance comparison according to the results of Simoes and Streib; it is compatible with "*" and it contradicts with "*4".
"*4" denotes the performance ranking of the study of Hausser and Strimmer. It contradicts with the results of "*" and "*3".
"*5" denotes that the performance ranking is changing with respect to the spline order in the study of Daub et al.
"*6" denotes that the ranking is changing according to whether the distribution close to normal or not. If it is close to normal, Edgeworth > NN and Edgeworth > ML.
NOTE: If the cell in the Table 4 which denoted by i-th row and j-th column has a ">" sign; it means estimator i is better than the estimator j in terms of estimation performance of the relationship. If the cell includes a "<" sign it means performance of i-th estimator is worse than j-th one. Sign "≈" means that the performances of i-th and j-th estimators are similar and close to each other.

genes takes value from the set {100, 200, 300}and the number of the samples takes value from the set {50, 100, 200, 300}. After that, a real yeast microarray dataset is used for the GNI process. They denoted that combination of MRNET and SCC and combination of CLR and PCC gave the best inference results in the experiments. The first combination is less biased because its performance is good without noise in the dataset. The second combination is less variant with additive noise and so it is more robust. Results of SCC are close to the results of the best algorithms for ARACNE and MRNET. The CLR inference method is less sensitive to the estimator. The best result is obtained by PCC for CLR. In the case of noisy data and with missing values in dataset, PCC and SCC give the highest F-score for all networks. PCC is better than SCC (and the rest of all) when missing values exist. SCC is better than PCC (and the rest of all) when data are complete but noisy [14]. Without the noise, ML and MM estimators gave the highest F-score value with equal frequency discretization for MRNET and ARACNE GNI algorithms. Because two correlation-based estimators (PCC and SCC) give the best results for the synthetic datasets in most of the scenarios, nonlinearity and nonmonotony of the relation of variables can be ignored. Taking into account the linear or monotone relationships is enough for an efficient estimation [14]. They also used Kendall's tau correlation coefficient [14,44–50] in their "minet" software implementation; however, they did not report any results about the Kendall's tau estimator in their study [14]. Finally it can be said that, when the dataset is noiseless, the performance ranking of the estimators from better to worse is: MM >ML >SCC >PCC >James–Stein shrinkage. When the dataset is noisy and complete, the ranking is: SCC >PCC >MM >ML >James–Stein shrinkage [14]. The sign "*" in Table 4 denotes that the performance ranking is changing according to the noise existence in the dataset.

Simoes and Streib also investigated the effects of MI estimation and data discretization on the inference of gene regulatory networks (GRN). They evaluated the inference performance by combining MM, ML, Schürmann–Grassberger (Bayesian 3), and James–Stein shrinkage estimators with three different discretization (equal frequency, equal width, global equal width) approaches [15]. They claimed that changing the discretization approach affects the result more efficiently than changing the estimators. They denoted that joining the MM estimator with the equal width (EW) and global equal width (GEW) discretizations achieves the best inference result. ML is the second best estimator and better than the Schürmann–Grassberger and shrinkage estimators. ML and MM are empirical-based estimators. Probability distribution of each bin is obtained from observed data samples. Moreover, MM is the only estimator that takes into account the bias. Unlike the ML and MM, the shrinkage and the Schürmann–Grassberger estimators try to enhance the estimation of probability distributions for each bin. In other words, the shrinkage and the Schürmann–Grassberger estimators aim to make efficient estimations about the distribution of data points. After that, they use these probabilities for the calculation of individual entropies, joint entropy, and finally for mutual information. Simoes and Streib claimed that MM outperforms the shrinkage estimator.

Finally, the experimental results of [15] illustrate that the performance ranking is: MM >ML >Schürmann–Grassberger (Bayes 3) >shrinkage. This result is consistent with the study of Olsen et al. with noiseless data. Simoes and Streib did not investigate the noisy case of the dataset. In the experiments they used synthetic datasets whose number of samples takes value from the set {50, 100, 200, 500, 1000}. Each one of the estimators in the study [15] assumes that the data has a single probability density. However, this is not the case for the real datasets.

3.2. Analysis of shrink, ML, MM, Bayes 1, Bayes 2, Bayes 3, Bayes 4, CS, and NSB

Hausser and Strimmer [23] proposed using a new type of James–Stein shrinkage estimator to obtain the estimation of the association between genes in the GNI algorithms. They wanted to see the effects of both undersampling and oversampling on the performance. They generated several datasets with different data generation and sampling scenarios before using the genomic datasets. Number of the samples takes value from the set {10, 30, 100, 300, 1000, 3000, 10,000}. At the end they used the proposed method to extract nonlinear gene association networks from a genomic dataset with 9 samples and 102 genes. They evaluated 9 different estimators by using artificial datasets. They claimed that the shrinkage method outperforms the other 8 methods (Jeffreys' prior (Bayesian 1) [51–56], Bayes–Laplace prior (Bayesian 2) [57–59], Schürmann–Grassberger (Perks' prior–Bayesian 3) [60-65], Minimax prior (Bayesian 4) [66-71], ML, MM, Chao–Shen (CS) [23,72–76], and NSB). All of the nine estimators are compared according to MSE and bias of the entropy values. With respect to these situations [23]:

- When the sample size is large, all nine estimators can achieve good estimations.
- NSB, CS, and shrinkage performed similarly and they are the most consistent and accurate ones, due to fact that they achieve low MSE values for all the scenarios and for all sample numbers. Those three best algorithms can be used for entropy estimation. In most of the scenarios the CS estimator is nearly unbiased.
- They also denoted that the computational complexity of the NSB method is larger than that of CS and shrinkage, i.e. it is the slowest one. In the simulations shrinkage was faster than NSB by 1000 times. Hence using CS or shrinkage is preferable. Implementation of the CS algorithm is easy and also it can produce good results in the GNI application in our opinion.
- ML and MM perform very badly even with large sample numbers.
- Bayesian estimators can achieve better or worse results than the ML; it depends on choosing of the prior and sampling size.
- Bayesian 2 and Bayesian 1 estimators are a bit better than the ML estimator, but they perform the worst for one of the scenarios.
- Bayesian 4 and Bayesian 3 estimators are better than most of the estimators (MM, ML, Bayesian 1, and Bayesian 2) in most scenarios.

Finally, in [23] the performance ranking from better to worse is: James–Stein shrinkage $\approx$ CS $\approx$ NSB >Bayes 4 >Bayes 3 >Bayes 1 >Bayes 2 >MM >ML. However, the results favoring shrinkage and Bayes 3 over the MM and ML methods, and the results favoring shrinkage over the Schürmann–Grassberger contradict the results of the studies of [14] and [15]. Olsen et al. [14] and Simoes and Streib [15] denoted that MM and ML methods are always better than the shrinkage estimator. Results of the study by Olsen et al. and Simoes and Streib are not the pure MI prediction performance. They are the F-scores obtained from the difference between the true genomic network and the network constructed by GNI algorithms. However, Hausser and Strimmer used only the simulated datasets for evaluating the MI estimation performances. Olsen et al. and Simoes and Streib used different datasets from that of Hausser and Strimmer. The contradiction between the results of the studies possibly arose from the above reasons. The sign "*3" in Table 4 denotes the performance comparison according to the results of Simoes and Streib; sign "*4" denotes the performance ranking of the study of Hausser and Strimmer. "*3" and "*4" contradict each other.

3.3. Analysis of ML, MM, jackknife, and BUB estimators

Paninski [25] investigated four different estimators in terms of several statistical points of views such as consistency of central limit theorem, bias, and variance. ML, MM, jackknife [77–81], and best upper bound (BUB) estimators were used for evaluation. All of the estimators try to minimize the variance and bias of the entropy estimation. He also investigated the confidence intervals for the situations $N << m$, $N >> m$ and $N \sim m$, where N is the number of samples and m is the number of cells. Paninski claimed that the MI estimation performance of the BUB approach is better than that of the ML, MM, and jackknife methods and it gives good results when the MM estimator fails, when $N \sim m$. However, this is not an appropriate case in MI estimation of gene pairs. This is illustrated in Table 4. Paninski proposed using the BUB estimator for analyzing neuroscientific data and denoted that this estimator is a version of bias correction of the MM estimator.

3.4. Analysis of BS, KDE, and BUB estimators

The B-spline (BS) estimator [16,25,82–87] and kernel density estimator (KDE) [88–91] are used in several studies of bioinformatics. Daub et al. [16] claimed that the performance of the BS approach changes with the spline order. They compared the BS with KDE and BUB estimators [16,92,93] by using two large-scale gene expression datasets for performance comparison of the MI estimation. The first dataset has 5345 genes and 300 samples; the second one includes 22,608 genes and 102 samples. The ranking of the performance is: BS (spline order = 3) >KDE >BS (spline order = 1) >BUB. The sign "*5" in the Table 4 denotes that the performance ranking changes with the spline order. Daub et al. also indicate that significance of the KDE does not depend on the bin number and was determined as similar to the significance of BS. The computational complexity of the KDE is O(10^4) times higher than the complexity of BS. Thus, the expensiveness of the KDE limits the size of utilizable data. The H_{BUB} entropy estimator produces intermediate significance between the result of binning and the result of the BS approach for higher bin numbers. For low bin numbers significance is relatively poor. Compared to KDE, BS functions produce similar significances. However, BS is less expensive than KDE. Daub et al. did not indicate the parameters of the KDE that they used in their comparisons. The optimal smoothing parameter of the KDE method is investigated in [17,94] by exploiting [95,96].

3.5. Analysis of BS, KNN Entr, KNN-MI$^{(1)}$, KNN-MI$^{(2)}$, PCC, and KDE estimators

Kraskov et al. [97] proposed two new approaches known as KNN-MI$^{(1)}$ and KNN-MI$^{(2)}$. These approaches directly estimate the MI by using the K-nearest neighborhood (KNN) method [98–105]. They compared those two approaches with the KNN entropy estimator [97,99,106–113], which estimates the MI indirectly from the entropies by using yeast expression dataset with 6000 genes and 300 samples. The bias caused by the separately estimation of H(X), H(Y), and H(X,Y) is decreased in the MI$^{(1)}$ and MI$^{(2)}$ methods. They claimed that the performance ranking is KNN-MI$^{(2)}$ >KNN-MI$^{(1)}$ >KNN-entropy estimator. Numata et al. also used the KNN-MI$^{(2)}$ method to compare it with the KNN-entropy estimator and PCC [99]. These three estimators were compared according to the MI estimation performance of the artificial datasets and in terms of the bias of the estimators. During the experiments, Numata et al. firstly used datasets having simple functional relationships, and then they used a metabolomic dataset with 43 samples and 181 random variables (standardized metabolite concentration ratios). They denoted that the direct MI estimator approach with k-NN proposed by Kraskov et al. gives the best estimation performance and the least bias among the three methods. Hence the ranking of the estimators is KNN-MI$^{(2)}$ >KNN-entropy >PCC [99].

Papana et al. [100] compared the KDE and KNN approaches by using simulated datasets on time series with several complexities. They denoted that the KNN estimator was more stable, less affected by the method-specific parameter, and computationally more effective because of using effective data structures for the searching of the neighbors. They denoted that, because KNN is not significantly corrupted with noise, it was better.

Lastly, the complexity of KNN is $O(N^3M^2)$, where N is the number of samples and M is the number of genes, while the worst-case complexity of BS is a bit less, $O(N^2M^2)$. Thus BS has a smaller computational complexity than the method favored above, KNN.

3.6. Analysis of PCC, PPC^1, PPC^n, BS, and CMI^1 estimators

We also reviewed the studies that eliminate the indirect dependencies by using conditional relationships. de la Fuente et al. used a higher order partial Pearson correlation (PPC) coefficient up to the third order [30]. Their purpose was not to infer the network by using PPC. They aimed to find interactions between the components reliably. After that, they used those interactions in network inference algorithms by using a dataset that has 781 genes. According to their studies, to construct a new edge in the graph, association value between the nodes, which are interacted by that edge, should be greater than a particular threshold value. They claimed that the results of second-order and third-order PPCs are similar and therefore using third-order PPC does not improve the obtained results significantly; using the second-order PPC is an optimal choice for reliable network inference. They used a high threshold value to eliminate the indirect interactions, i.e. the application is based on a high significance level. Thus, they obtained low false positive (FP) and high false negative (FN) rates as they expected [30]. Çakır et al. also evaluated the partial Pearson correlation estimator (zero-th order, first order, i.e. PPC^1, and n-th order, i.e. PPC^n), B-spline MI estimator, and first order conditional MI estimator (CMI^1, which uses b-spline estimator) by using two different real microarray datasets (E. coli and S. cerevisiae) and three variability approaches (enzymatic, intrinsic, and environmental variabilities) in their study [31]. They searched the metabolic network inference from the artificial metabolome data. The approaches PCC, PPC^1, and PPC^n [31, 114] are parametric methods because of the linearity assumption of the relationship; B-spline and CMI^1 [31,115–119] estimators are nonparametric and nonlinear methods. They claimed that their study is a relatively untouched area and there was no detailed study that examines the similarity measures on the metabolome data in the literature prior to their study. According to experiments [31]:

- PPC^n gave the best results except for the environmental variability case. In that case CMI^1 gave the best result.
- Unconditioned scores, i.e. PCC and MI, gave the worst results.
- They denoted that SCC gave identical results but it is applied to data rankings. However, they did not illustrate the results of SCC.
- Datasets used in the study do not have nonlinear relationships due to fact that the results of linear and nonlinear similarity measures are close to each other.
- Conditioning approaches remove the indirect links mostly; thus they improve the inference results.
- In most of the variability cases of the datasets PPC^1 and CMI^1 perform significantly better than the nonconditioned counterparts (PCC, BS) that provide more connectivity to the networks. PPC^1 is the

second and CMI^1 is the third best performing estimator in most experiments without pruning. CMI^1 is the second best estimator instead of PPC^1 in a few experiments.

- Çakır et al. denoted that they expanded the ARACNE (algorithm for the reconstruction of accurate cellular networks) method by using similarity scores different from MI. They used conditional similarity measures instead of an MI estimator (ARACNE uses KDE for MI estimation).

Hence, in most of the variability conditions for both of the datasets the performance ranking in [31] becomes PPC^n >PPC^1 >CMI^1 >BS >PCC. However, in a few experiments ranks of the best second (PPC^1) and the third (CMI^1) estimators may be changed. Sign "*2" denotes this case.

3.7. Analysis of MIC, PCC, SCC, KNN, KDE, dCor, and HHG estimators

Reshef et al. [24] proposed the maximal information coefficient (MIC) method. They compared the proposed method with PCC, SCC, KDE, and KNN estimators proposed by Kraskov et al. They claimed that MIC is the most stable and reliable estimator among them. According to the results of the experiments, the ranking is MIC >SCC >KNN >KDE >PCC. They performed experiments by using four different datasets: health, baseball, genomics, and large-scale human microbiota. They also generated datasets including several functional relationships (linear, quadratic, exponential, etc.) and different noise levels. They used those artificial datasets for comparison before employing the previously given four real datasets. There are several criticisms concerning the study [24]. Most of them claimed that distance correlation (dCor) and HHG (Heller, Heller, Gorfine) estimators outperform the MIC estimator.

The dCor estimator is proposed by Szekely et al [120]. dCor is used in several applications [121–125]. The proposed method measures and checks the dependency/independency of two given random variables. They did not use a genomics dataset; they used moderate-sized artificial datasets obtained from Monte Carlo simulation in their experiments. In some GNI algorithms, after obtaining the associations, independent genes can be eliminated. In our opinion, pruning of the independent genes from the gene interaction matrix can be achieved by using distance correlation metric.

Heller et al. [126] proposed a new test for checking the independency between random vectors. In the literature independency tests of variables are very few. Mostly dependency tests are used for assessing the association. The aim of [126] is to construct a consistent and multivariate independency test statistic, which is also known as HHG. The HHG is based on the pairwise difference of the random variables X and Y. They implemented simulations with small and moderate sample sizes to compare MIC with dCor [120] and HHG methods. They claimed that dCor and HHG approaches are more powerful than MIC. They also denoted that HHG outperforms the dCor metric. They did not use a genomics dataset in their experiments. In [24] it is claimed that equitability is more important than the power. However, Heller et al. denoted that, equitability does not help to detect relationships between the variables. Even if it helps, MIC provides equitability only for noiseless functions. However, a noise-free functional relationship is an unrealistic case. Thus MIC does not provide equitability for all associations. Furthermore, MIC can be used for only univariate datasets. However, dCor and HHG approaches can be used for multivariate datasets [120,126]. Furthermore, Tibshirani and Simon implemented simulations of MIC, PCC, and dCor [127]. They generated several variable pairs with different functional relationships for comparison of three methods. They claimed that dCor has more power than the MIC method; sometimes even PCC provides more power than MIC. They denoted that, if a method has low power, then its equitability property is not useful. dCor is more powerful than MIC and also it has less computational

complexity than MIC [127]. Finally, according to reported results in the literature, the ranking of the estimators can be given as HHG >dCor >MIC >SCC >KNN >KDE >PCC.

3.8. Analysis of ANOVA estimator for GNI

ANOVA is used in several studies [128–133] for bioinformatics. Küffner et al. [128] proposed using ANOVA for a specific GNI application, i.e. for gene regulatory network inference (GRNI). They claimed that ANOVA is a nonlinear and nonparametric metric and better than the PCC. However, their study is a specific GNI application, which only includes the relationship of the transcription factors (TF) and target genes (TG) and accepts only TFs as regulators in the network. The experiments involve five different large-scale datasets. Three of the datasets (artificial, E. coli, and S. cerevisiae) are taken from DREAM5 [134,135]; two of them (E. coli and S. cerevisiae) are taken from the M3D database [136]. In those five datasets, the number of genes takes value from {1643, 4511, 4297, 5950, 6572}and the number of chips takes value from {805, 805, 907, 536, 904}, respectively.

3.9. Analysis of PCC, Edgeworth, ML, KNN, KDE, and LSMI estimators

Edgeworth expansion is used in several fields [97,137–143]. Hulle proposed using Edgeworth expansion to estimate differential multivariate entropy [137]. He compared Edgeworth expansion and KNN for MI estimation, for both univariate and multivariate cases. The Edgeworth approach performs better than the K-NN direct MI estimator for the multivariate case while the true distribution is close to Gaussian. If the density model is not close to Gaussian, Edgeworth expansion estimations become worse. Hulle claimed that the computational complexity of the KNN is larger than the complexity of the Edgeworth approach. Moreover, Hulle denoted that the entropy estimation has a bias. However, by the standardized cumulants used in the MI estimation, gets decreased. In addition, Suzuki et al. compared the Edgeworth method with the Maximum Likelihood approach [144]. They denoted that Edgeworth performs well when the true distribution is close to the normal density. However, it performs worse when the distribution is not close to normal. Hence, the same relationship between Edgeworth estimation and closeness of the dataset to normal density is observed. Therefore, if the distribution is close to normal, the performance ranking becomes Edgeworth >KNN and Edgeworth >ML; otherwise the ranking is Edgeworth <KNN and Edgeworth <ML. Sign "*6" in Table 4 denotes that the ranking changes according to whether the distribution close to normal or not.

Suzuki et al. used the least squares mutual information (LSMI) estimator to measure the association between gene pairs [98]. LSMI is used in several application fields [144–147]. Suzuki et al. compared the proposed approach with the PCC, KNN, KDE, and Edgeworth methods. They claimed that the proposed LSMI approach performs better than those approaches. They also found that KNN performed better than KDE. However, there was a problem when choosing the parameter k appropriately in KNN. Suzuki et al. used two different microarray datasets in their applications. The first dataset includes 173 microarray data; the second one involves 300 microarray data. The missing values in both of the datasets are replaced by the average value of the expression values of their own dataset.

The LSMI has three advantages over the other methods. First, in the LSMI method, estimation of density is not necessary as opposed to the KDE method. Second, in the LSMI a model can be selected, while methods such as KNN do not allow selection of a model. Third, we do not need any assumption about the dataset. However, for instance in the Edgeworth method, the dataset distribution is assumed as close to Gaussian. Edgeworth performs well when the true distribution is close to the normal density; it performs worse when the

distribution is not close to normal. Finally, the performance ranking can be given as LSMI >PCC, LSMI >KDE, LSMI >KNN, and LSMI >Edgeworth. In this respect, LSMI should be a good choice to use for association estimation in GNI applications.

3.10. Overall analysis

In this subsection all of the previous analyses are considered to give the best performing estimators by specifying the used datasets and performance metrics. In most of the reviewed studies, firstly synthetic datasets were used for comparison. Then in some of those studies real datasets were also used for evaluation. In some of the studies the performance metric was chosen as the error between the actual MI and predicted MI values. However, in the other studies the performance metric was based on the difference between the actual gene network and the inferred gene network. These evaluation differences and dataset variability in the reviewed studies might cause a change in the ranking of the estimators. The sample size of the datasets might cause various performances for the same estimator [148]. Nevertheless, this ranking can give preliminary information about the methods. According to this information, the evaluation and comparison results in the literature are summarized as follows.

By using F-score metric of GNI and noiseless synthetic datasets generated by SynTReN, the ranking of the 5 estimators becomes MM >ML >SCC >PCC >shrinkage estimators. With noisy synthetic data it becomes SCC >PCC >MM >ML >shrinkage [14]. Genomics datasets tend to have noise. Hence SCC seems a promising method to use in GNI applications. In the study [15], which also uses synthetic networks, the ranking of four different MI-based estimators is MM >ML >Schürmann–Grassberger (Bayesian 3) >shrinkage. Area under the curve precision-recall (AUC-PR) was used as performance metric in [15]. The performance according to noise in the datasets was not evaluated [15]. Still, MM can be also accepted as another promising method for MI estimation [14,15,25]. In another study [23], 9 estimators were evaluated by generating synthetic datasets and using a MSE of the difference between the actual and predicted MI values. The differences in the ranking of the estimators might be caused by the fact that the metrics used in the studies of [14], [15], and [23] are different. The James–Stein shrinkage estimator was indicated as the best performer among nine estimators [23]. The shrinkage estimator was also used for a real genomics dataset at the end of the study and presented as a promising method for MI estimation [23]. The CS estimator can be considered another appropriate method for MI estimation, because it performs similar to NSB and its computational complexity is significantly less than that of NSB. Moreover, implementation of CS is much simpler than that of NSB [23]. Hence CS was one of the chosen promising estimators for GNI applications. In another study, the BUB estimator was compared with three other estimators in terms of several statistical views such as bias and variance by using neuroscientific data [25]. Paninski claimed that BUB performed better than ML and jackknife. BUB is a different form of MM and it is reported to give better results than MM when the sample size is nearly equal to the bin number [25]. However, this is not a general case for entropy-based GNI applications. In addition, in [16], BS was compared with BUB and KDE. It is stated that BS with any spline order performs better than the BUB estimator. Therefore, BUB did not seem a good choice. When the spline order is greater than 2, BS performs better than KDE. KDE requires an addition operation, copula transform, while BS does not need it. Moreover, it is also claimed that BS is 10^4 times less complex than KDE in [16]. Hence, BS seems a favorable choice for using in GNI applications. Because KNN-MI$^{(2)}$ does not involve bias and treats "the distribution over a sample and its k-th neighbor" more accurately, the direct MI estimator KNN-MI$^{(2)}$ performs better than entropy-based KNN and KNN-MI$^{(1)}$ estimators [97]. The performance metric in that study was systematic error of MI, i.e. the difference between the exact and predicted MI values. Results of this study illustrate that KNN-MI$^{(2)}$ was

a propitious method to use in GNI algorithms. Because KNN-MI$^{(2)}$ is favored in several studies [97,99,100], it might also be chosen. Indirect interaction elimination methods were also reviewed in this study. Higher order PPCs were claimed as better than their nonpartial counterparts [30,31]. In [31], it is reported that in most experiments PPC^n outperforms the PCC, PPC^1, CMI^1, and BS estimators. Hence PPC^n was another appropriate promising method to use. In [24], MIC was proposed to find association scores. It outperforms SCC, KNN, KDE, and PCC methods by using datasets with several functional relationships [24]. In many studies dCor and HHG estimators are claimed to outperform MIC significantly [120,126,127]. It is also reported that HHG outperforms dCor [126]. Hence, HHG seems another promising correlation estimator. In [98] LSMI was compared with PCC, KDE, KNN, and Edgeworth estimators by using two different microarray datasets. LSMI was reported to outperform those four methods and have three advantages over the other methods: LSMI does not require density estimation opposed to KDE. Model selection can be made in LSMI; however, KNN does not satisfy model selection. LSMI does not have any assumption or restriction about the dataset, but Edgeworth assumes that the distribution of the dataset is close to Gaussian. Hence, LSMI can be used efficiently in GNI applications.

It is worth mentioning that some of the reviewed estimators such as the shrinkage estimator [23], PCC, and higher order partial correlation coefficients [30,31] can be used in Bayesian network (BN) applications [149,150]. In [151], it is denoted that BNs can be inferred by using probabilistic classification or regression models, such as generalized linear regression, probabilistic neural networks, probabilistic decision trees, and dictionary methods. BNs can be efficiently used to extract the directed edges in the GRNs [152] and divided into two groups: static and dynamic. Dynamic BNs may assume that a Markovian process exists, i.e. each variable (gene) depends only on the variables of the previous time step while each sample corresponds to a particular time step [150]. However, we are not interested in inferring directional networks and the process does not have to be Markovian. Hence, BNs were not considered and outside the scope of this study. We examined the association estimators that can be used for inferring the undirected GNs such as ARACNE [17], RelNet [18], and C3NET [19]. Moreover, contrary to those GNI algorithms, BNs may not be used efficiently for large-scale datasets.

Furthermore, the estimators can be compared with respect to whether the copula transform is used or not. The copula transform uses the ranking values of the data and normalizes those values to make them between the range [0, 1].

4. Conclusion

In this study, correlation-based and MI-based estimators are reviewed. Although our goal was to determine the most efficient and suitable estimator to use in the GNI algorithms, we concluded that there is no best estimator identified in the literature. Nevertheless, we found some of them as the most promising estimators. As a result we also concluded that there is a need for a comprehensive comparative analysis study that evaluates all these estimators on the same datasets with various parameter conditions to be able to select the best estimator. We summarized the promising estimators below.

Considering all of the previous review analysis, the best performing estimators appear to be LSMI, HHG, and KNN-MI$^{(2)}$. As mentioned in subsection 3.9, LSMI has three advantages over the other three promising estimators (KDE, KNN, and Edgeworth). HHG gives better results than the estimator MIC, which is claimed to be better than SCC, KNN, KDE, and PCC methods. KNN-MI$^{(2)}$ estimates the MI in a more direct way than the other estimators. Hence it does not have the bias that is included by the other estimators.

Moreover, MM and SCC seem to be other promising methods. MM takes into account the bias. Hence it may estimate the MI more accurately than the other entropy-based methods. SCC is a correlation-based method. However, unlike the PCC method it does not have any assumption about the relationship. It can also handle nonlinear relationships. The CS estimator is also thought to be an appropriate method for using in GNI applications due to the fact that it performs similar to NSB, which is a more complicated approach than CS in terms of computational complexity. In fact implementation of the CS approach is simpler than that of most of the other approaches. Because it provides soft-binning and the borders of the bins are not sharp, BS seems a preferable method when the spline order is chosen appropriately. This situation increases the accuracy of the density estimation.

PPC^n is stated as the best performing method among the indirect dependency eliminator approaches [31]. Anyone who wants to eliminate indirect interactions may use PPC^n in his/her study. Hence, eight different methods are suggested to be used for GNI applications.

Briefly, the suggested estimators are SCC, MM, CS, BS, KNN-MI$^{(2)}$, PPC^n, HHG, and LSMI. The selected estimators are suggested to be used in GNI applications to investigate the effects of the estimators on GNI performance.

In addition, the reviewed estimators were classified according to whether they are parametric or nonparametric. Furthermore, MI-based estimators were compared according to the different discretization approaches. Classification strategies and their reasons were given in Section 2. Detailed explanations of the estimators and some explanatory examples for them are given in the Appendices.

To the best of our knowledge, this review is the most comprehensive review so far on this topic and we expect it to be an essential guide for researchers who work on related studies.

References

[1] Mattiussi V, Tumminello M, Iori G, Mantegna RN. Comparing correlation matrix estimators via Kullback–Leibler divergence. Available at Social Science Research Network (SSRN) 2011: 77-104.

[2] Rogers LCG, Zhou F. Estimating correlation from high, low, opening and closing prices. Annals of Applied Probability 2008; 18: 813-823.

[3] Lindskog F. Linear Correlation Estimation. RiskLab research papers, 11 Dec. 2000.

[4] Neemuchwala H, Hero AO. Image registration in high dimensional feature space. Proc. of SPIE Conference on Electronic Imaging, San Jose, Jan. 2005.

[5] Hero A, Ma B, Michel O, Gorman J. Applications of entropic spanning graphs. IEEE Signal Processing Magazine 2002; 19: 85-95.

[6] Póczos B, Kirshner S, Szepesvári C. REGO: Rank-based estimation of Renyi information using Euclidean graph optimization. Journal of Machine Learning Research - Proceedings 2010; 9: 605-612.

[7] Brunelli R, Messelodi S. Robust estimation of correlation with applications to computer vision. Pattern Recognition 1995; 28: 833-841.

[8] Shan C, Gong S, McOwan PW. Conditional mutual information based boosting for facial expression recognition. In British Machine Vision Conference (BMVC), Oxford, UK, Sep. 2005.

[9] Lee WCY. An extended correlation function of two random variables applied to mobile radio transmission. Bell Sys Tech Journal 1969; 48: 3423-3440.

[10] Barceló-Lladó JE, Pérez AM, Seco-Granados G. Enhanced correlation estimators for distributed source coding in large wireless sensor networks. IEEE Sensors Journal 2012; 12: 2799-2806.

[11] Kerscher M, Szapudi I, Szalay AS. A comparison of estimators for the two-point correlation function. The Astrophysical Journal 2000; 535: L13-L16.

[12] Habib E, Krajewski WF, Ciach GJ. Estimation of rainfall interstation correlation. Journal of Hydrometeorology 2001; 2: 621-629.

[13] Moon YI, Rajagopalan B, Lall U. Estimation of mutual information using kernel density estimators. Physical Review E 1995; 52: 2318-2321.

[14] Olsen C, Meyer PE, Bontempi G. On the impact of entropy estimation on transcriptional regulatory network inference based on mutual information. EURASIP Journal on Bioinformatics and Systems Biology 2009; 2009(308959).

[15] Simoes RM, Emmert-Streib F. Influence of statistical estimators of mutual information and data heterogeneity on the inference of gene regulatory networks. PLoS ONE 2011; 6(11).

[16] Daub CO, Steuer R, Selbig J, Kloska S. Estimating mutual information using B-spline functions - an improved similarity measure for analysing gene expression data. BMC Bioinformatics 2004; 5(118).

[17] Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera, RD, Califano A. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 2006; 7(Suppl 1): S7.

[18] Butte AJ, Kohane LS. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. Pacific Symposium on Biocomputing, Hawaii, vol. 5, pp. 418-429, 4–9 Jan. 2000.

[19] Altay G, Emmert-Streib F. Inferring the conservative causal core of gene regulatory networks. BMC Systems Biology 2010; 4(132).

[20] Altay G, Asim M, Markowetz F, Neal DE. Differential C3NET reveals disease networks of direct physical interactions. BMC Bioinformatics 2011; 12(296).

[21] Altay G, Emmert-Streib F. Revealing differences in gene network inference algorithms on the network-level by ensemble methods. Bioinformatics 2010; 26: 1738-1744.

[22] Walters-Williams J, Li Y. Estimation of mutual information: a survey. Lecture Notes in Computer Science Rough Sets and Knowledge Technology 2009; 5589: 389-396.

[23] Hausser J, Strimmer K. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. Journal of Machine Learning Research 2009; 10: 1469-1484.

[24] Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, Lander ES, Mitzenmacher M, Sabeti PC. Detecting novel associations in large data sets. Science 2011; 334(6062): 1518-1524.

[25] Paninski L. Estimation of entropy and mutual information. Neural Computation 2003; 15: 1191-1253.

[26] Vu VQ, Yu B, Kass RE. Coverage-adjusted entropy estimation. Statistics in Medicine 2007; 26: 4039-4060.

[27] Horvath S. Weighted Network Analysis: Applications in Genomics and Systems Biology, 1st Edition, Berlin, Germany: Springer, 2011.

[28] Meyer PE, Kontos K, Bontempi G. Biological network inference using redundancy analysis. Proceedings of the 1st International Conference on Bioinformatics Research and Development (BIRD'07) Lecture Notes in Computer Science; 12–14 March 2007; Berlin, Germany; 4414: 16-27.

[29] Federer A. Estimating networks using mutual information. Master Thesis, Swiss Federal Institute of Technology Zurich Department of Mathematics 2011.

[30] de la Fuente A, Bing N, Hoeschele I, Mendes P. Discovery of meaningful associations in genomic data using partial correlation coefficients. Bioinformatics 2004; 20: 3565-3574.

[31] Çakır T, Hendriks MMWB, Westerhuis JA, Smilde AK. Metabolic network discovery through reverse engineering of metabolome data. Metabolomics 2009; 5: 318-329.

[32] Cukierski WJ, Nandy K, Gudla P, Meaburn KJ, Misteli T, Foran DJ, Lockett SJ. Ranked retrieval of segmented nuclei for objective assessment of cancer gene repositioning. BMC Bioinformatics 2012; 13: 232.

[33] Zhang G, Su Z. Inferences from structural comparison: flexibility, secondary structure wobble and sequence alignment optimization. BMC Bioinformatics 2012; 13(Suppl 15): S12.

[34] Ray A, Lindahl E, Wallner B. Improved model quality assessment using ProQ2. BMC Bioinformatics 2012; 13: 224.

[35] Malovini A, Barbarini N, Bellazzi R, De Michelis F. Hierarchical Naive Bayes for genetic association studies. BMC Bioinformatics 2012; 13(Suppl 14): S6.

[36] Qiu P, Zhang L. Identification of markers associated with global changes in DNA methylation regulation in cancers. BMC Bioinformatics 2012; 13(Suppl 13): S7.

[37] Mahanta P, Ahmed HA, Bhattacharyya DK, Kalita JK. An effective method for network module extraction from microarray data. BMC Bioinformatics 2012; 13(Suppl 13): S4.

[38] Lin HC, Goldstein S, Mendelowitz L, Zhou S, Wetzel J, Schwartz DC, Pop M. AGORA: Assembly Guided by Optical Restriction Alignment. BMC Bioinformatics 2012; 13: 189.

[39] Wu C, Zhu J, Zhang X. Integrating gene expression and protein-protein interaction network to prioritize cancer-associated genes. BMC Bioinformatics 2012; 13: 182.

[40] Gonnet GH. Surprising results on phylogenetic tree building methods based on molecular sequences. BMC Bioinformatics 2012; 13: 148.

[41] Collingridge PW, Kelly S. MergeAlign: improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments. BMC Bioinformatics 2012; 13: 117.

[42] Ayadi W, Elloumi M, Hao JK. Pattern-driven neighborhood search for biclustering of microarray data. BMC Bioinformatics 2012; 13(Suppl 7):S11.

[43] Van den Bulcke T, Van Leemput K, Naudts B, Van Remortel P, Ma H, Verschoren A, De Moor B, Marchal K. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. BMC Bioinformatics 2006; 7(42).

[44] Kendall M. A new measure of rank correlation. Biometrika 1938; 30: 81-89.

[45] Datta S, Pihur V, Datta S. An adaptive optimal ensemble classifier via bagging and rank aggregation with applications to high dimensional data. BMC Bioinformatics 2012; 11: 427.

[46] Yu H, Kim T, Oh J, Ko I, Kim S, Han WS. Enabling multi-level relevance feedback on PubMed by integrating rank learning into DBMS. BMC Bioinformatics 2010; 11(Suppl 2): S6.

[47] Freilich S, Goldovsky L, Gottlieb A, Blanc E, Tsoka S, Ouzounis CA. Stratification of co-evolving genomic groups using ranked phylogenetic profiles. BMC Bioinformatics 2009; 10: 355.

[48] Pihur V, Datta S, Datta S. RankAggreg: an R package for weighted rank aggregation. BMC Bioinformatics 2009; 10: 62.

[49] Pyysalo S, Airola A, Heimonen J, Björne J, Ginter F, Salakoski T. Comparative analysis of five protein-protein interaction corpora. BMC Bioinformatics 2008; 9(Suppl 3): S6.

[50] Tien YJ, Lee YS, Wu HM, Chen CH. Methods for simultaneously identifying coherent local clusters with smooth global patterns in gene expression profiles. BMC Bioinformatics 2008; 9: 155.

[51] Casadei D. Estimating the selection efficiency. Journal of Instrumentation 2012; 7: 8021.

[52] Veitch J, Mandel I, Aylott B, Farr B, Raymond V, Rodriguez C, van der Sluys M, Kalogera V, Vecchio A. Estimating parameters of coalescing compact binaries with proposed advanced detector networks. Physical Review D 2012; 85(9).

[53] Molladavoudi S, Zainuddin H, Chan KT. Jensen-Shannon divergence and nonlinear quantum dynamics. Physics Letters A 2012; 376(26-27): 1955-1961.

[54] Reginatto M, Hall MJW. Quantum theory from the geometry of evolving probabilities. 31st International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering; AIP Conference Proceedings, Waterloo, Ontario, Canada, 9–16 July 2011; 1443: 96-103.

[55] Genoni MG, Olivares S, Brivio D, Cialdi S, Cipriani D, Santamato A, Vezzoli S, Paris MGA. Optical interferometry in the presence of large phase diffusion. Physical Review A 2012; 85: 043817-1-043817-5.

[56] Švihlík J, Fliegel K, Kukal J, Jerhotová E, Páta P, Vítek S, Koten P. Estimation of nonGaussian noise parameters in the wavelet domain using the moment-generating function. Journal of Electronic Imaging 2012; 21: 023025-15.

[57] Tuyl F, Gerlach R, Mengersen K. A comparison of Bayes–Laplace, Jeffreys, and other priors. The American Statistician 2008; 62: 40-44.

[58] Geisser S. On prior distributions for binary trials. The American Statistician 1984; 38: 244-247.

[59] Dempster AP. Upper and lower probability inferences based on a sample from a finite univariate population. Biometrika 1967; 54: 515-528.

[60] Perks W. Some observations on inverse probability including a new indifference rule. J Inst Actuaries 1947; 73: 285-334.

[61] Stambaugh RF. Predictive regressions. Journal of Financial Economics 1999; 54: 375-421.

[62] Bernardo JM. Reference posterior distributions for Bayesian inference. Journal of the Royal Statistical Society. Series B (Methodological) 1979; 41: 113-147.

[63] Kass RE, Wasserman L. The selection of prior distributions by formal rules. Journal of the American Statistical Association 1996; 91: 1343-1370.

[64] Good IJ. Maximum entropy for hypothesis formulation, especially for multidimensional contingency tables. The Annals of Mathematical Statistics 1963; 34: 911-934.

[65] Phillips PCB. To criticize the critics: an objective bayesian analysis of stochastic trends. Journal of Applied Econometrics 1991; 6: 333-364.

[66] Trybula S. Some problems of simultaneous minimax estimation. The Annals of Mathematical Statistics 1958; 29: 245-253.

[67] Fienberga SE, Holland PW. Simultaneous estimation of multinomial cell probabilities. Journal of the American Statistical Association 1973; 68: 683-691.

[68] Wu TD, Nevill-Manning CG, Brutlag DL. Minimal-risk scoring matrices for sequence analysis. Journal of Computational Biology 1999; 6: 219-235.

[69] Braess D, Dette H. The asymptotic minimax risk for the estimation of constrained binomial and multinomial probabilities. Sankhyā: The Indian Journal of Statistics (2003-2007) 2004; 66: 707-732.

[70] Wilming N, Betz T, Kietzmann TC, König P. Measures and limits of models of fixation selection. PLoS ONE 2011; 6(8).

[71] Allen DE, McAleer M, Powell RJ, Kumar-Singh A. A NonParametric and Entropy Based Analysis of the Relationship between the VIX and S&P 500. Social Science Research Network 2012 [Working Paper or Technical Report] (Unpublished).

[72] Jost L. Partitioning diversity into independent alpha and beta components. Ecology 2007; 88: 2427-2439.

[73] De Filippo C, Cavalieri D, Di Paola M, Ramazzotti M, Poullet JB, Massart S, Collini S, Pieraccini G, Lionetti P. Impact of diet in shaping gut microbiota revealed by a comparative study in children from Europe and rural Africa. Proceedings of the National Academy of Sciences of the United States of America (PNAS) 2010; 107: 14691-14696.

[74] Mao CX, Colwell RK. Estimation of species richness: mixture models, the role of rare species, and inferential challenges. Ecology 2005; 86: 1143-1153.

[75] Kéry M, Schmid H. Estimating species richness: calibrating a large avian monitoring programme. Journal of Applied Ecology 2006; 43: 101-110.

[76] Victor JD. Approaches to information-theoretic analysis of neural activity. Biological Theory 2006; 1: 302-316.

[77] Wieczorkowski R, Grzegorzewski P. Entropy estimators-improvements and comparisons. Communications in Statistics - Simulation and Computation 1999; 28: 541-567.

[78] Zahl S. Jackknifing an index of diversity. Ecology 1977; 58: 907-913.

[79] Moddemeijer R. A statistic to estimate the variance of the histogram-based mutual information estimator based on dependent pairs of observations. Signal Processing 1999; 75: 51-63.

[80] Wu EHC, Yu PLH, Li WK. A smoothed bootstrap test for independence based on mutual information. Computational Statistics & Data Analysis 2009; 53: 2524-2536.

[81] Schlogl A, Keinrath C, Scherer R, Furtscheller P. Information transfer of an EEG-based brain computer interface. First International IEEE EMBS Conference on Neural Engineering 2003; 20–22 March, pp. 641-644.

[82] Saha S, Dey KN, Dasgupta R, Ghose A, Mullick K. Missing value estimation in DNA microarrays using b-splines. Journal of Medical and Bioengineering 2013; 2: 88-92.

[83] Steuer R, Kurths J, Daub CO, Weise J, Selbig J. The mutual information: detecting and evaluating dependencies between variables. Bioinformatics 2002; S231-S240.

[84] Luan Y, Li H. Clustering of time-course gene expression data using a mixed-effects model with B-splines. Bioinformatics 2003; 19: 474-482.

[85] Li H, Sun Y, Zhan M. Analysis of gene coexpression by B-spline based CoD estimation. EURASIP Journal on Bioinformatics and Systems Biology 2007; 2007: 49478.

[86] Xu S. Time-course microarray data analysis. In: Principles of Statistical Genomics. Springer: New York, NY, USA, 2013. pp. 365-382.

[87] Storey JD, Xiao W, Leek JT, Tompkins RG, Davis RW. Significance analysis of time course microarray experiments. Proceedings of the National Academy of Sciences of the United States of America 2005; 102(35): 12837-12842.

[88] Chang DTH, Ou YY, Hung HG, Yang MH, Chen CY, Oyang YJ. Prediction of protein secondary structures with a novel kernel density estimator. BMC Research Notes 2008; 1:51.

[89] Chang DTH, Wang CC, Chen JW. Using a kernel density estimation based classifier to predict species-specific microRNA precursors. BMC Bioinformatics 2008; 9(Suppl 12): S2.

[90] Murakami Y, Mizuguchi K. Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. Bioinformatics 2010; 26: 1841-1848.

[91] Liao JG, Lin Y, Selvanayagam ZE, Shih WJ. A mixture model for estimating the local false discovery rate in DNA microarray analysis. Bioinformatics 2004; 20: 2694-2701.

[92] Hausser J. Improving Entropy Estimation and the Inference of Genetic Regulatory Networks. M.S. thesis. Department of Biosciences, National Institute of Applied Sciences of Lyon, Cedex, France; 2006.

[93] King BM, Tidor B. MIST: Maximum Information Spanning Trees for dimension reduction of biological data sets. Bioinformatics 2009; 25: 1165-1172.

[94] Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A. Technical Report: Parameter Estimation for the ARACNE Algorithm. Tech Rep nprot. 2006; 106-S2.

[95] Duin RPW. On the choice of smoothing parameter for Parzen estimators of probability density functions. IEEE T Comput 1976; C-25: 1175-1179.

[96] Koontz WLG, Fukunaga K. Asymptotic analysis of a nonparametric clustering technique. IEEE T Comput 1972; C-21: 967-974.

[97] Kraskov A, Stögbauer H, Grassberger P. Estimating mutual information. Phys Rev E 2004; 83(1).

[98] Suzuki T, Sugiyama M, Kanamori T, Sese J. Mutual information estimation reveals global associations between stimuli and biological processes. BMC Bioinformatics 2009; 10 (Suppl 1): S52.

[99] Numata J, Ebenhöh O, Knapp E-W. Measuring correlations in metabolomic networks with mutual information. Genome Inform 2008; 20: 112-122.

[100] Papana A, Kugiumtzis D. Evaluation of mutual information estimators on nonlinear dynamic systems. Nonlinear Phenomena in Complex Systems 2008; 11: 225-232.

[101] Pereda E, Quiroga RQ, Bhattacharya J. Nonlinear multivariate analysis of neurophysiological signals. Progress in Neurobiology 2005; 77: 1-37.

[102] Stögbauer H, Kraskov A, Astakhov SA, Grassberger P. Least-dependent-component analysis based on mutual information. Phys. Rev. E 2004; 70(6): 066123-1-066123-17.

[103] Rossi F, Lendasse A, François D, Wertz V, Verleysen M. Mutual information for the selection of relevant variables in spectrometric nonlinear modeling. Chemometrics and Intelligent Laboratory Systems 2006; 80: 215-226.

[104] Kraskov A, Stögbauer H, Andrzejak RG, Grassberger P. Hierarchical clustering using mutual information. EPL (Europhysics Letters) 2005; 70(2).

[105] Fukumizu K, Gretton A, Sun X, Schölkopf B. Kernel Measures of Conditional Dependence. In: Advances in Neural Information Processing Systems 20: Proceedings of the 2007 Conference, MIT Press, Cambridge, MA, USA, pp. 489-496.

[106] Singh H, Misra N, Hnizdo V, Fedorowicz A, Demchuk E. Nearest neighbor estimates of entropy. American J of Math and Manag Sciences 2003; 23: 301-321.

[107] Hnizdo V, Darian E, Fedorowicz, Demchuk E, Li S, Singh H. Nearest-neighbor nonparametric method for estimating the configurational entropy of complex molecules. J Comput Chem 2007; 28: 655-668.

[108] Numata J, Wan M, Knapp EW. Conformational entropy of biomolecules: beyond the quasi-harmonic approximation. Genome Informatics 2007; 18: 192-205.

[109] Kozachenko LF, Leonenko NN. Sample estimates of entropy of a random vector. Problems of Information Transmission 1987; 23: 95-101.

[110] Mnatsakanov RM, Misra N, Li S, Harner EJ. K_n-nearest neighbor estimators of entropy. Mathematical Methods of Statistics 2008; 17: 261-277.

[111] Li S, Mnatsakanov RM, Andrew ME. K-nearest neighbor based consistent entropy estimation for hyperspherical distributions. Entropy 2011; 13: 650-667.

[112] Rahvar ARA, Ardakani M. Boundary effect correction in k-nearest-neighbor estimation. Phys. Rev. E 2011; 83: 051121-1-051121-8.

[113] Mnatsakanov RM, Li S, Harner EJ. Estimation of multivariate shannon entropy using moments. Australian & New Zealand Journal of Statistics 2011; 53: 271-288.

[114] Schäfer J, Strimmer K. An empirical Bayes approach to inferring large-scale gene association networks. Bioinformatics (Oxford, England) 2005; 21: 754-764.

[115] Zhang X, Zhao XM, He K, Lu L, Cao Y, Liu J, Hao JK, Liu ZP, Chen L. Inferring gene regulatory networks from gene expression data by PC-algorithm based on conditional mutual information. Bioinformatics 2011; 28: 98-104.

[116] Meyer PE, Lafitte F, Bontempi G. *minet*: A R/Bioconductor package for inferring large transcriptional networks using mutual information. BMC Bioinformatics 2008; 9: 461.

[117] Kim DC, Wang X, Yang CR, Gao J. Learning biological network using mutual information and conditional independence. BMC Bioinformatics 2010; 11(Suppl 3):S9.

[118] Liang K-C, Wang X. Gene regulatory network reconstruction using conditional mutual information. EURASIP Journal on Bioinformatics and Systems Biology 2008; 2008: 253894.

[119] Chaitankar V, Ghosh P, Perkins EJ, Gong P, Zhang C. Time lagged information theoretic approaches to the reverse engineering of gene regulatory networks. BMC Bioinformatics 2010; 11: S19.

[120] Szekely G, Rizzo M. Brownian distance covariance. The Annals of Applied Statistics 2009; 3: 1236-1265.

[121] Szekely G, Rizzo M, Bakirov N. Measuring and testing independence by correlation of distances. The Annals of Statistics 2007; 35: 2769-2794.

[122] Sejdinovic D, Gretton A, Sriperumbudur B, Fukumizu K. Hypothesis testing using pairwise distances and associated kernels. Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.

[123] Sejdinovic D, Sriperumbudur B, Gretton A, Fukumizu K. Equivalence of distance-based and RKHS-based statistics in hypothesis testing, in review. [arXiv].

[124] Grothe O, Schmid F, Schnieders J, Segers J. Measuring association between random vectors. Submitted to Computational Statistics & Data Analysis, in review.

[125] Zhou Z. Measuring nonlinear dependence in time-series, a distance correlation approach. Journal of Time Series Analysis 2012; 33: 438-457.

[126] Heller R, Heller Y, Gorfine MA. A consistent multivariate test of association based on ranks of distances, Technical Report, June 1, 2012.

[127] Simon N, Tibshirani R. Comment on "Detecting Novel Associations in Large Data Sets". Science Dec, 2011

[128] Küffner R, Petri T, Tavakkolkhah P, Windhager L, Zimmer R. Inferring gene regulatory networks by ANOVA. Bioinformatics 2012; 28: 1376-1382.

[129] Bharathi A, Natarajan AM. Cancer classification of bioinformatics data using ANOVA. International Journal of Computer Theory and Engineering 2010; 2(3).

[130] Krawetz S. Bioinformatics for Systems Biology, 2nd ed., 2009, Springer.

[131] De Haan JR, Wehrens R, Bauerschmidt S, Piek E, van Schaik RC, Buydens LMC. Interpretation of ANOVA models for microarray data using PCA. Bioinformatics 2006; 23: 184-190.

[132] Nueda MJ, Conesa A, Westerhuis JA, Hoefsloot HCJ, Smilde AK, Talón M, Ferrer A. Discovering gene expression patterns in time course microarray experiments by ANOVA-SCA. Bioinformatics 2007; 23: 1792-1800.

[133] Smilde AK, Jansen JJ, Hoefsloot HCJ, Lamers R-JAN, van der Greef J, Timmerman ME. ANOVA-simultaneous component analysis (ASCA): a new tool for analyzing designed metabolomics data. Bioinformatics 2005; 21: 3043-3048.

[134] Stolovitzky G, Monroe D, Califano A. Dialogue on Reverse-Engineering Assessment and Methods: The DREAM of High-Throughput Pathway Inference, Annals of the New York Academy of Sciences 2007; 1115: 11-22.

[135] Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, Allison KR, The DREAM5 Consortium, Kellis M, Collins JJ, et al. Wisdom of crowds for robust gene network inference. Nature Methods 2012; 9: 796-804.

[136] Faith JJ, Driscoll ME, Fusaro VA, Cosgrove EJ, Hayete B, Juhn FS, Schneider SJ, Gardner TS. Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata. Nucleic Acids Res 2008; 36(Database issue): D866–D870.

[137] Van Hulle MM. Edgeworth approximation of multivariate differential entropy. Neural Computation 2005; 17: 1903-1910.

[138] Harremoes P. Maximum entropy and the Edgeworth expansion. IEEE Information Theory Workshop 2005, 29 Aug.-1 Sept. Rotorua, New Zealand.

[139] Zeng J, Xie L, Kruger U, Gao C. A nonGaussian regression algorithm based on mutual information maximization. Chemometrics and Intelligent Laboratory Systems 2012; 111: 1-19.

[140] Comon P. Independent component analysis, a new concept? Signal Processing 1994; 36: 287-314.

[141] Amari S, Cichocki A, Yang HH. A New Learning Algorithm for Blind Signal Separation. in Advances in Neural Information Processing Systems 1996; 8: 757-763, MIT Press.

[142] Huber P. Projection pursuit. The Annals of Statistics 1985; 13: 435-475.

[143] Hlaváčková-Schindler K, Paluš M, Vejmelka M, Bhattacharya J. Causality detection based on information-theoretic approaches in time series analysis. Physics Reports 2007; 441: 1-46.

[144] Suzuki T, Sugiyama M, Sese J, Kanamori T. Approximating mutual information by maximum likelihood density ratio estimation. The Journal of Machine Learning Research (JMLR): Workshop and Conference Proceedings 2008; 4: 5-20.

[145] Sugiyama M, Kanamori T, Suzuki T, Hido S, Sese J, Takeuchi I, Wang L. A Density-ratio framework for statistical data processing. IPSJ Transactions on Computer Vision and Applications 2009; 1: 183-208.

[146] Hido S, Tsuboi Y, Kashima H, Sugiyama M, Kanamori T. Statistical outlier detection using direct density ratio estimation. Knowledge and Information Systems 2011; 26: 309-336.

[147] Sugiyama M, Suzuki T. Least-Squares Independence Test. IEICE TRANSACTIONS on Information and Systems 2011; E94-D(6): 1333-1336.

[148] Altay G. Empirically determining the sample size for large-scale gene network inference algorithms. IET Systems Biology 2012; 6: 35-63.

[149] Scutari M. Learning Bayesian Networks with the bnlearn R Package. Journal of Statistical Software 2010; 35: 1-22.

[150] Lèbre S. Inferring dynamic genetic networks with low order independencies. Statistical Applications in Genetics and Molecular Biology 2009; 8: 1-39.

[151] Heckerman D. A tutorial on learning with Bayesian networks. Microsoft Research. Technical Report MSR-TR-95-06. Redmond, Washington, Mar. 1995.

[152] Perrin BE, Ralaivola L, Mazurie A, Bottani S, Mallet J, d'Alché-Buc F. Gene networks inference using dynamic Bayesian networks. Bioinformatics 2003; 19: ii138-ii148.

Appendices

NOTE: We illustrate numerical examples for some of the estimators.

A. Appendix. Estimators

In this Section the estimators are reviewed and described extensively so readers may not need to look at the original references. Comparisons and discussions about the estimators were given in the previous section. Reviewing and understanding of the estimators from one source is provided by this study. The order of the classification, given in Table 2, is followed for the explanation of the estimators. Firstly the parametric approaches, then the nonparametric approaches, and lastly the semiparametric methods will be mentioned. Nevertheless, before the descriptions of the estimators, the entropy and mutual information (MI) concepts, which are frequently cited in the estimator explanations, will be given in Section A.1. Furthermore, the discretization process needed in the most of the MI-based methods will be defined in subsection A.2. After that, parametric estimators are described in the subsections between A.3 and A.8, nonparametric estimators are defined in the subsections between A.9 and A.24, and semiparametric ones are given in the subsection A.25.

A.1. Entropy and mutual information (MI)

Entropy is an uncertainty metric about a random variable. It is mostly used in compression applications. It can be used in several application fields such as communications (text or image compression, analyzing sensor locations), natural language processing, signal analysis (segmentation, detection, image registration, and texture classification), statistical learning, chemistry, physics, etc. If the unpredictability is decreasing, entropy also decreases. Shannon entropy definition of a random variable, X, is given as

$$H(X) = -\sum_{x_i \in X} P(X = x_i) \log\left(P\left(X = x_i\right)\right) \tag{1}$$

The dataset should be separated into bins to use entropy estimators, as mentioned previously. The probability density of a bin corresponds to the expression $P(X = x_i)$. For instance, by using the probability densities of the bins, we obtain the entropy of X. There are several binning approaches (equal frequency, equal width, etc.) in the literature.

MI is a measure that illustrates the dependency of two random variables and it is obtained by

$$\begin{aligned} MI(X,Y) &= H(X) + H(Y) - H(X,Y) \\ &= -\sum_{x_i \in X} \sum_{y_j \in Y} P(X = x_i, Y = y_j) \log \frac{P(X=x_i, Y=y_j)}{P(X=x_i)P(Y=y_j)} \end{aligned} \tag{2}$$

where H denotes entropies.

The joint entropy $H(X,\ Y)$ is obtained as

$$H(X,Y) = -\sum_{x_i \in X} P(X = x_i, Y = y_j) \log\left(P(X = x_i, Y = y_j)\right) \tag{3}$$

A.2. Discretization techniques and histogram-based approaches

Association between gene pairs can be obtained by correlation-based estimators, entropy-based estimators, or direct MI estimators. Entropy estimators, which require calculation of the entropy, are different from the

correlation-based estimators (PCC, SCC, etc.). They first need the calculation of probability densities of different bins in the dataset or probability density over each data sample then summing up those densities to obtain overall density in the dataset. In this case, binning or discretization should be applied to the datasets in this case. The most commonly used discretization approaches are equal frequency and equal width techniques. They are explained in the following subsections. In addition, the histogram-based entropy estimators are defined in Section A.2.3.

A.2.1. Equal frequency

In this technique, each bin of the dataset has the same frequency value, i.e. each one of them should have the same number of samples, but widths of the bins may be different. Figure 2 illustrates an equal frequency binning example for a simple dataset including the following normalized gene expression values {0.16, 0.79, 0.31, 0.53, 0.19, 0.60, 0.26, 0.65, 0.69, 0.75}. Because 10 datapoints exist in the dataset, each bin should have 2 datapoints if 5 bins are assumed.

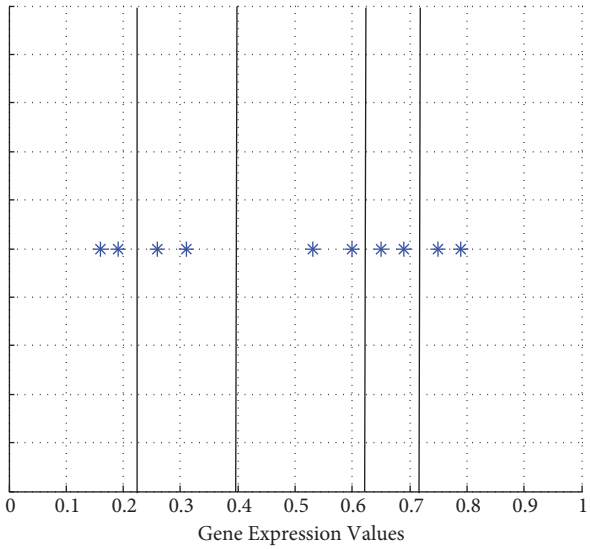

Figure 2. Equal frequency binning example

A.2.2. Equal width

In this technique, each bin has the same interval, but numbers of the data points in the bins may be different. An example is illustrated in Figure 3 for the same dataset used in Figure 2. In this example, the width of each bin is 0.2 units. The first three bins have two samples, the fourth bin has four samples, and the fifth bin has no sample.

In both binning techniques, number of bins is generally selected as $\sqrt{N}\sqrt{N}$, where N is the number of samples. Each bin should involve a particular number of data points [14 15].

A.2.3. Histogram-based approaches

MI estimation is based on the individual and joint entropies of the variables. In order to obtain entropies, the probability density of the dataset should be known. Histogram-based approaches (such as ML, MM, shrinkage estimator, and Schürmann–Grassberger estimators) use the number of the data samples in each bin. For instance

maximum likelihood or the empirical estimator obtains the probability density of the dataset by counting the number of the data points of each bin empirically.

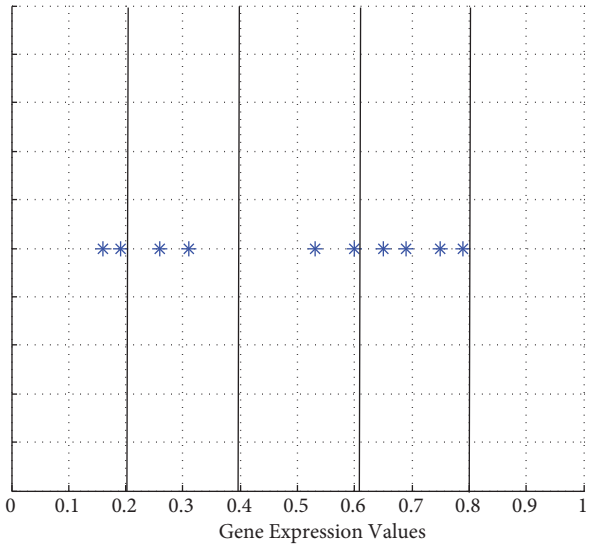

Figure 3. Equal width binning example

A.3. Pearson correlation coefficient

Pearson correlation coefficient (PCC) assumes that there is a linear relationship between two random variables. Because it calculates this linear association, it can be classified as a linear and parametric estimator. The relationship is denoted as

$$\rho = \frac{cov(X,Y)}{\sigma_x \sigma_y} \tag{4}$$

where $cov(X,Y)\mathrm{cov}(X,Y)$ means covariance of X and Y; $\sigma_x\sigma_x$ means standard deviation of variable X. Correlation value can be between [–1,1]. A correlation value of 0 does not mean that two variables must be independent. They may have a nonlinear relationship. Because it is a linear estimator, PCC fails to estimate the nonlinear relationship. This is illustrated by an example given in Figure 4.

In Figure 4 (a), a relationship close to the linearity is observed. Therefore correlation coefficient is found to be 0.98. However, because the relationship in Figure 4 (b) is far from linearity, the correlation coefficient for this relationship is closer to 0 (0.29).

Correlation and MI score are related if the joint distribution is normal. Entropy of the distribution of a multivariate Gaussian variable X:

$$H(X) = \frac{1}{2}\ln\left\{(2\pi e)^d(\sigma_x)\right\} \tag{5}$$

where σ_x is standard deviation of the variable X and d is dimension of X [14].

Similarly, $H(X,Y) = \frac{1}{2}\ln\left\{(2\pi e)^{2d}\det(\mathbf{C})\right\}$ where $\mathbf{C}$ is the covariance matrix.

From (5), MI between X and Y becomes [14]

$$MI(X,Y) = \frac{1}{2}\log\left(\frac{\sigma_x\sigma_y}{\det(\mathbf{C})}\right) = -\frac{1}{2}\left(1-\rho^2\right) \tag{6}$$

PCC can be estimated from the samples x_i and y_i of two variables (e.g., genes in genomics), X and Y:

$$\hat{\rho} = \frac{\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{d\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}\sqrt{d\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i\right)^2}} \tag{7}$$

where d is the dimension of each variable X.

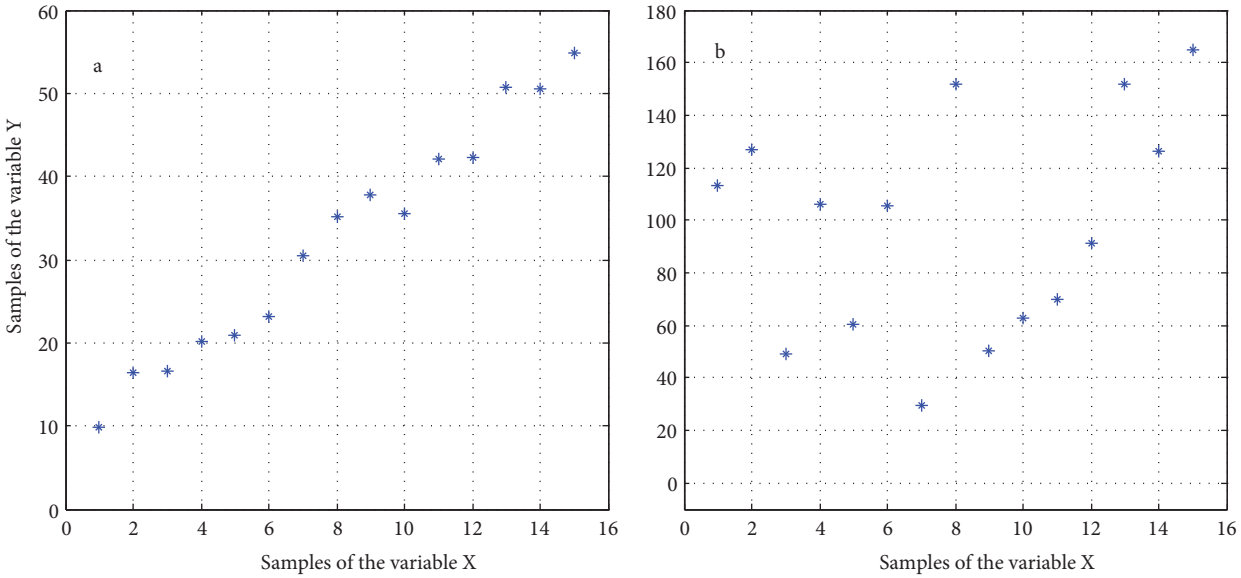

Figure 4. (a) When the relationship is close to the linearity (correlation coefficient is 0.98); (b) When the relationship is far from the linearity (corr. coeff. is 0.29)

A.4. Bayesian estimators

Hausser and Strimmer used four different Bayesian estimator variants (Jeffreys' prior [51–56], Bayes–Laplace estimator [57–59], Perks' estimator [60–65], minimax Bayesian estimators [66–71]) in their study [23]. They used a bioinformatics dataset as well as artificial datasets [23].

Bayesian estimators try to improve the estimation of ML, with the purpose of Bayesian estimators use Dirichlet distribution and the posterior distribution becomes also as Dirichlet distribution. Since the Bayesian methods make assumption about the distribution, they are parametric approaches. Those methods use parameters, a_1, a_2, ..., a_b as priors. Mean of the Dirichlet posterior distribution is

$$\hat{\theta}_k^{Bayes} = \frac{y_k + a_k}{N + A} \tag{8}$$

where $\hat{\theta}_k^{Bayes}$ is probability of each bin k according to Bayesian approach. When number of the cells (bins) is b, number of samples (observations) is N, observation count of each cell k is y_k and $A = \sum_{k=1}^{b} a_k$.

Entropy of the given dataset can be calculated from (9) for all of the Bayesian estimators:

$$\hat{H}^{Bayes} = -\sum_{k=1}^{b} \hat{\theta}_k^{Bayes} \log\left(\hat{\theta}_k^{Bayes}\right) \tag{9}$$

In the first Bayesian approach that is called as Jeffreys' prior, cell frequency prior is $a_k = {}^1/_2$ for $k = 1,...,b$.

In the second one, namely Bayes–Laplace approach, the only difference from Jeffreys' estimator is that, $a_k = 1$ for $k = 1,...,b$.

In the third Bayesian approach called Perks' Bayesian approach, which is also known as Schürmann–Grassberger estimator, the cell frequency prior is given as $a_k = {}^1/_b$ for $k = 1,...,b$..

In the last Bayesian approach (minimax approach), the cell frequency prior is $a_k = {}^{\sqrt{n}}/_b$ for $k = 1,...,b$.

All of the Bayesian estimators are parametric approaches. Discussions and comparisons including them are given in Section 3.

A.5. Edgeworth estimator

Hulle proposed using Edgeworth expansion to estimate the differential multivariate entropy [137]. In the study MI estimation by Edgeworth expansion is achieved as an application and data density model is assumed to be Gaussian; hence the Edgeworth estimator can be considered as a parametric method [137].

1-D Edgeworth expansion becomes popular in the context of independent component analysis (ICA). Edgeworth expansion of a d-dimensional density $p(v)$ up to order five can be denoted by (10) with the normal estimate, ϕ_p:

$$p(v) \approx \phi_p \left(1+\frac{1}{3!}\sum_{i,j,k} \kappa^{i,j,k} h_{ijk}(v)+\frac{1}{4!}\sum_{i,j,k,l} \kappa^{i,j,k,l} h_{ijkl}(v)+\frac{1}{72!}\sum_{i,j,k,l,p,q} \kappa^{i,j,k}\kappa^{l,p,q} h_{ijklpq}(v)\right) \tag{10}$$

where ϕ_p is a d-dimensional normal density that has the same mean and covariance parameters with the p; i, j, and k are input dimensions with $i,j,k \in \{1,\ldots,d\}$; h_{ijk} is Hermite polynomial, $\kappa^{i,j,k}$ is standardized cumulant with $\kappa^{i,j,k}=\frac{\kappa^{ijk}}{\sigma_i\sigma_j\sigma_k}$, where κ^{ijk} is the third central moment over the input dimensions i, j, and k. h_{ijkl} is i, j, k, and l-th Hermite polynomial over the input dimensions i, j, k, and l. $\kappa^{i,j,k,l}=\frac{\kappa^{ijkl}}{\sigma_i\sigma_j\sigma_k\sigma_l}$ where κ^{ijkl} is the fourth central moment over the input dimensions i, j, k, and l.

The differential entropy, $H(p)$, becomes

$$H(\phi_p)-\int \phi_p(v)\left(Z(v)+0.5Z(v)^2\right)dv = H(\phi_p)-\frac{1}{12}\left(\sum_{i=1}^{d}\left(\kappa^{i,i,i}\right)^2+3\sum_{i,j=1,i\neq j}^{d}\left(\kappa^{i,i,j}\right)^2+\frac{1}{6}\sum_{i,j,k=1,i\neq j\neq k}^{d}\left(\kappa^{i,j,k}\right)^2\right) \tag{11}$$

d-dimensional $H(\phi_p)$ entropy is calculated as

$$H(\phi_p) = 0.5\log|\Sigma| + \frac{d}{2}\log 2\pi + \frac{d}{2} \tag{12}$$

MI estimation for both 1-D and multivariate datasets is

$$\sum_i H(v_i) - H(v) \tag{13}$$

In genomics datasets $\mathbf{v}$ is a 2-D vector in the joint entropy case, $\mathbf{v} = v_1, v_2$ or $\mathbf{v} = x,\ y$. For the individual entropy $d = 1$ and $\mathbf{v}$ is 1-D. Firstly the individual entropies, $H(v_1)$ and $H(v_2)$ for $d = 1$, and then the joint

entropy $H(v_1,v_2)$ for $d=2$ should be obtained. After that, MI is calculated by (13). Maximum value of the dimension d should be 2. Hence the last term in (11) becomes invalid.

Discussions and comparisons including Edgeworth method are given in Section 3.

A.6. Least-squares mutual information (LSMI) estimator

The LSMI approach is actually a feature selection technique. Suzuki et al. used this feature selection approach to measure the association between the gene pairs [98]. They denoted that KDE is a naive approach for MI estimation and it is not effective for practical applications. The only advantage of this approach is that the bandwidth parameter, h, can be adjusted according to the dataset by cross validation. k-NN is an alternative approach for MI estimation. Its application is easier than that of the KDE approach. However, number of the neighbors, parameter k, could not be selected adaptively according to the dataset. Choosing an appropriate value for k is important but difficult. The LSMI does not deal with the challenges such as density estimation and determining the value of the parameter k. However, it estimates the MI by modeling the density ratio:

$$w(x,y)=\frac{p(x,y)}{p(x)\,p(y)} \tag{14}$$

MI definition, based on the squared loss, is

$$I_s(X,Y)=\iint\left(\frac{p(x,y)}{p(x)\,p(y)}-1\right)^2\times p(x)\,p(y)\,dxdy \tag{15}$$

The aim of the LSMI is to estimate the squared loss MI given in (15). During the MI estimation only density ratio given in (14) is used in the LSMI method. Squared loss MI calculation by using a density ratio $\hat{w}(x,y)$ is

$$\hat{I}_s(X,Y)=\frac{1}{n^2}\sum_{i,j=1}^{n}\left(\hat{w}(x_i,y_j)-1\right)^2 \tag{16}$$

LSMI method simply uses the solution of a linear equation system. Because of the linearity of the model, LSMI is considered a parametric estimator. The density ratio function can be modeled by the linear model given as

$$\hat{w}_a(x,y)=\alpha^T\varphi(x,y) \tag{17}$$

where parameters $\alpha=(\alpha_1\alpha_2,\ldots,\alpha_b)$ can be extracted from data samples and $\varphi(x,y)=(\varphi_1(x,y),\varphi_2(x,y),\ldots,\varphi_b(x,y))^T$ are basis functions. Each of the basis functions is a b-dimensional vector and the elements of these vectors are positive rational numbers. The basis functions are derived by using data samples, x_i and y_i. They could be any kernel function. An instance for these basis functions is given (22).

While searching for the parameters of the model, the cost function, $J(\alpha)$, is aimed to become minimum:

$$J(\alpha)=J_0(\alpha)-C=\frac{1}{2}\alpha^T H\alpha-h^T\alpha \tag{18}$$

where C is a constant.

In order to obtain the parameter α we should use (19) derived from (18):

$$\tilde{\alpha}=\left[\frac{1}{2}\alpha^T\hat{H}\alpha-\hat{h}^T\alpha+\lambda\alpha^T\alpha\right] \tag{19}$$

where $\hat{H}=\frac{1}{n^2}\sum_{i,j=1}^{n}\varphi(x_i,y_j)\,\varphi(x_i,y_j)^T$, $\hat{h}=\frac{1}{n}\sum_{i=1}^{n}\varphi(x_i,y_j)$ and the last element of the sum, $\lambda\alpha^T\alpha$, is the regularization term. $\mathbf{H}$ is a $b \times b$ matrix and $\mathbf{h}$ is a $b \times 1$ vector. Finally $\tilde{\alpha}$ is obtained by the linear model:

$$\tilde{\alpha}=\left(\hat{H}+\lambda I_b\right)^{-1}\hat{h} \tag{20}$$

where $\mathbf{I_b}$ is the identity matrix.

Efficiency and the performance of the LSMI depends on the choosing the basis functions, $\varphi(x,y)$, and selection of the regularization parameter, λ. Selection of the model parameters can be achieved by cross validation. Data samples are divided into K different subgroups for cross validation: $\{Z_k\}_{k=1}^{K}$. For a group k, the density ratio estimation, $\hat{w}_k(x,y)$, and the cost, $\hat{J}^{K-CV}$, is obtained by using the samples from rest of the groups, $\{Z_j\}_{j\neq k}$:

$$\hat{J}^{K-CV}=\frac{1}{K}\sum_{k=1}^{K}\hat{J}_r^{K-CV}. \tag{21}$$

Then the parameters α and the basis functions $\varphi(x,y)$ which minimize the cost function, are calculated.

Gaussian kernel can be used for basis functions $\varphi_l(x,y)$:

$$\varphi_l(x,y)=\exp\left(-\frac{\|x-u_l\|^2}{2\sigma^2}\right)\delta(y=v_l) \tag{22}$$

where the points $\{(u_l,v_l)\}_{l=1}^{b}$ are b different center points that are randomly selected from the points $\{(x_i,y_i)\}_{i=1}^{n}$. If $y=v_l$, the value of the indicator function $\delta(y=v_l)$ is 1. Otherwise its value is 0.

Suzuki et al., determine the number of the basis functions, b, as: $b=\min(100,\ N)$ in the experiments, where N is the number of the data samples [98].

Bandwidth parameter of the Gaussian kernel and the regularization parameter λ can be determined by grid search cross-validation. Discussions and comparisons including LSMI are given in Section 3.

A.7. First order partial Pearson correlation coefficient (PPC^1)

Correlation between random variables X and Y denotes the joint behavior of X and Y. Partial correlation denotes the joint behavior of X and Y when conditioning on the control variables Z_1, Z_2,..., Z_n. In other words, partial Pearson correlation coefficient (PPC) measures the strength of the interaction between two random variables under control of the variables Z_1, Z_2,..., Z_n.

Let us examine the first order PPC. The first order PPC between two random variables X and Y, conditioning on Z, depicts the correlation between the residuals of X and Y after they are regressed on the control variable Z. In other words, PPC of X and Y conditioning on Z is the correlation of the X's and Y's uncorrelated parts with Z. First order PPC can be obtained by using 0-th order correlation coefficients r_{xy}, r_{yz}, r_{xz}:

$$r_{xy|z}=\frac{r_{xy}-r_{xz}\times r_{yz}}{\sqrt{(1-r_{xz}^2)\times(1-r_{yz}^2)}} \tag{23}$$

Higher order partial correlations are expanded version of the first order partial correlation. They can be obtained from the previous-order PPCs iteratively. PPC enables us to eliminate indirect interactions in a

gene interaction matrix. For instance, if the $r_{xy|z}$ value is under a statistical threshold, then the value of the interaction between the genes X and Y is assigned 0. An example about the indirect connections is given in Section B of the Appendices.

A.8. *n*-th order partial Pearson correlation coefficient ($\mathbf{PPC^n}$)

Çakır et al. used a parametric and conditioned similarity score called as graphical Gaussian modeling (GGM) framework or n-th order partial Pearson correlation (PPC^n) [31]. GGM or PPC^n can be used for eliminating the indirect interactions in a gene interaction matrix. It shows the scores for all remaining variables concurrently. It is obtained by inversing the 0-th order Pearson correlation matrix and normalizing the inverse matrix to have diagonals -1. Normalization process is achieved by

$$\mathbf{\Pi}_{i,j} = -\frac{\omega_{i,j}}{\sqrt{\omega_{i,i} \times \omega_{j,j}}}, where \;\; \mathbf{\Omega = P^{-1}} = \omega_{i,j} \tag{24}$$

where $\mathbf{P}$ is 0-th order Pearson correlation matrix, Ω is inverse of $\mathbf{P}$, and $\mathbf{\Pi}$ is the n-th order partial Pearson correlation of the dataset. Discussions and comparisons including PPCn are given in Section 3.

A.9. Spearman correlation coefficient

Spearman rank correlation coefficient (SCC) is a special case of PCC. Data are converted to rankings before coefficient calculation. SCC can be calculated by replacing the terms x_i and y_i by their ranks:

$$\hat{\rho} = \frac{\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{d\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}\sqrt{d\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i\right)^2}} \tag{25}$$

SCC is able to detect not only the linear relationships, but also any kind of monotone relation without making any assumptions about the distributions of the variables. Therefore, SCC is a nonparametric method.

Two examples for two random variables (X, Y) are shown in Figure 5 (a) and (b). According to (a) PCC is low (0.24) and SCC is moderate (0.55). According to (b) PCC is moderate (–0.54) and SCC is low (–0.29).

A.10. Kendall tau correlation coefficient

Kendall tau (τ) rank correlation coefficient also measures the association between two random variables. This is a nonparametric metric that can be used for testing the statistical dependence of two random variables. Kendall rank correlation coefficient requires the ranks of the dataset similar to the SCC.

If the ranks of two pairs, (x_i,y_i) and (x_j, y_j) are compatible, then those two pairs are called concordant. In other words, if $x_i > x_j$ and $y_i > y_j$ or opposite, these two pairs are concordant. However, if the pairs satisfy the conditions: $x_i > x_j$ and $y_i < y_j$ or opposite, these pairs are called discordant. Lastly, if the pairs satisfy the equality of the ranks of the data samples, i.e. $x_i = x_j$ or $y_i = y_j$, the pair is neither concordant nor discordant. The Kendall τ coefficient is [44]

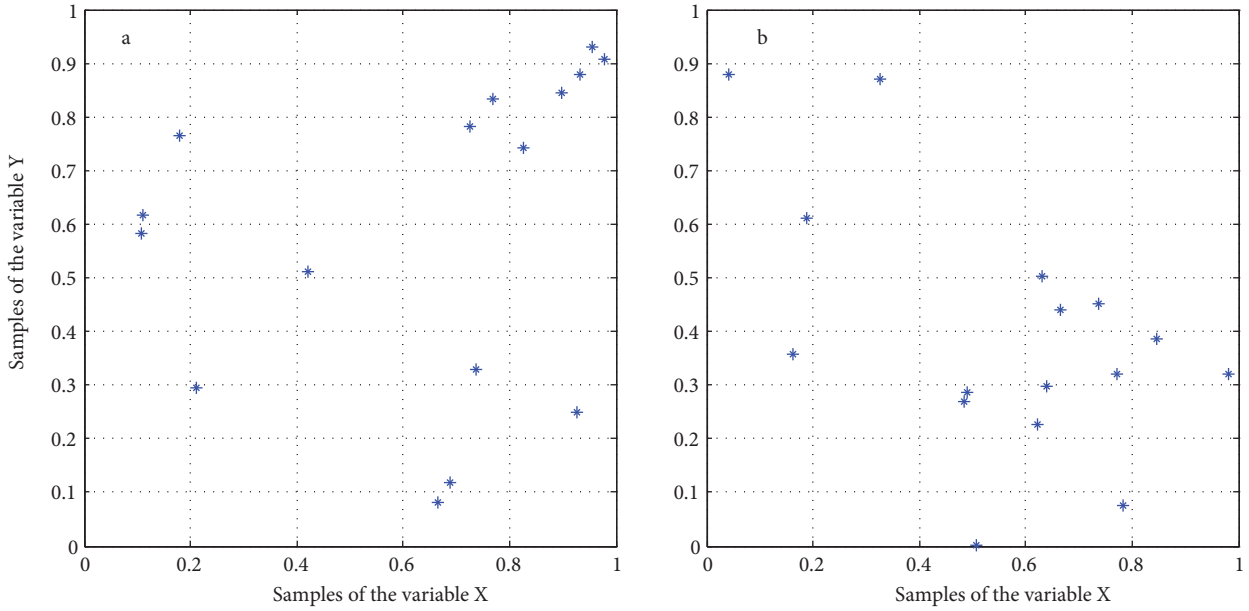

Figure 5. (a) Low PCC, moderate SCC example; (b) Moderate PCC, low SCC example

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{1}{2}N(N-1)} \tag{26}$$

where N is number of the data samples.

In Figure 5(a) while PCC is low (0.24), Kendall tau correlation coefficient is moderate (0.42). In Figure 5(b) while PCC is moderate (–0.54), Kendall tau correlation coefficient is low (–0.20).

A.11. Maximum likelihood (ML, empirical, naive) estimator

Most of the entropy-based estimators depended on the histogram approach. Empirical estimator is also based on the histogram approach. In the first step of the genomics applications, expression values of two genes obtained from the microarray data analysis are discretized into different intervals. Those are called bins.

Empirical approach estimates the entropy from the observed individual and joint frequencies for each bin. This is one of the simplest estimators. Walters-Williams and Li denoted that ML is a parametric estimator [22]. However, MI estimation by ML does not have any assumption about the distribution of the data or about the relationship between the random variables; hence ML should be considered a nonparametric estimator.

The empirical entropy H_{emp} is estimated from observed probability distribution. For a single random variable, it is given as

$$H_{emp} = -\sum_{k=1}^{b} \left(\frac{n_k}{N}\right) \log \left(\frac{n_k}{N}\right) \tag{27}$$

where N is the number of samples, b is number of bins; n_k is number of samples in the k-th bin.

H_{emp} gives the maximum-likelihood entropy estimator for a discrete random variable. Its drawback is that the true entropy H is underestimated as the number of bins increases. Increasing number of bins results

in undersampling of the cell frequencies [14,15,23,25]. Asymptotic bias of this estimator approaches

$$bias\,(H_{emp}) = -\frac{b-1}{2N}. \tag{28}$$

There are several applications which use the ML estimator [14,15,23,25]. An example of the MI calculation by ML estimator is given in Section C of Appendices. Discussions and comparisons including ML are given in Section 3.

A.12. Miller–Madow (MM) estimator

This estimator uses a constant factor that is proportional to the bin size and sample size to correct the estimation. It considers the undersampling bias. It decreases the bias, without increasing the variance [14,15,23,25].

Asymptotic bias of this ML estimator approaches $-\frac{b-1}{2N}$. Miller–Madow aims to remove this bias:

$$H_{mm} = H_{emp} + \frac{b-1}{2N} \tag{29}$$

Aim of an estimator is to minimize both the bias and the variance. However, their relationship is reciprocal. Variance becomes larger; bias becomes smaller when the complexity of an estimator increases.

A.13. An analysis of variance (ANOVA)

Analysis of variance (ANOVA) is a hypothesis testing application that uses the experimental data and says whether we accept or reject the null hypothesis. A two-way ANOVA association metric, denoted by the symbol η^2, is obtained by a two-way analysis of the variance. Two-way ANOVA models the measurements Y_{ijk} as responses to the factors A and B:

$$Y_{ijk}= \mu+\tau_k+\beta_j+\gamma_{jk}+\epsilon_{ijk} \tag{30}$$

where μ is the average response, τ_k is the effect of the k-th level of factor A, β_j is the effect of j-th level of factor B, γ_{jk} is the joint effect of the association between A and B, and ϵ_{ijk} represents the undefined error in the i-th sample.

ANOVA is constructed from different segments. Separating the variance sources is possible and then the hypothesis testing can be applied to each part. ANOVA is an auxiliary statistical test. It can be considered as a kind of structuring of the multidimensional models.

Separating the total sum of squares (SS) into its segments (factors) is the main idea of ANOVA. If we have two different factors the total SS becomes

$$SS_T=SS_{Factor1}+SS_{Factor2}+SS_{err} \tag{31}$$

There are particular assumptions of ANOVA in the statistical applications. Expected value of the error is assumed to be 0; variances of all errors are assumed identical and the errors are assumed as normally distributed.

Because of the assumptions, essentially ANOVA is a parametric method. However Küffner et al. proposed using a nonparametric and nonlinear version of two-way ANOVA in the gene regulatory network (GRN) inference algorithms. ANOVA is evaluated with respect to the inference performance of the GRNs. Küffner et al. aimed inferring GRN based on the nonlinear association between the regulators and their targets, i.e. they evaluated the associations between the transcription factor: target gene (TF:TG) by the metric η^2. Hence they aimed

to find the interactions between the pairs containing a TF and a TG. They denoted that the metric η^2 is not used in the gene network inference (GNI) and any other bioinformatics applications previously. A project called DREAM [134] exists that involves many GNI algorithms and provides comparisons between them. Küffner et al. used DREAM5 blind assessment to compare the proposed method with the other methods, PCC, MRNet, CLR, ARACNE. They claimed that the best performance was obtained by ANOVA method with real datasets in DREAM5 [135].

They use area under receiver-operator characteristic curve (AUROC) evaluation in the comparisons of the methods. The experiments involve five different datasets. Three of the datasets are taken from DREAM5; two of them are taken from M3D database [136]. All of the datasets are pre-processed and normalized previously. Potential TFs of each datasets are given to all of the GNI algorithms. Only the TFs are assumed as regulators in the networks, due to the fact that the interactions for other regulators are not included by the gold standards.

Basic gene expression levels can have very different values across the experiments. Hence, the expression values should be transformed into fold changes. Each measurement condition m has several replicates or measurements, m_i. Each of those measurement conditions is mapped to one or more control conditions. Then fold changes are obtained. This process is given in detail in Section D of the Appendices.

Each of the conditions m is a collection of different gene, drug and environmental perturbations. Gene perturbations of the control conditions should be less than that of the measurement condition m. We could not have control conditions for a condition m if it has a low perturbation level and if there cannot be any less perturbation combination possibility. Thus, some of the experiments do not allow having control conditions.

In this application of ANOVA, factor C corresponds to the effect of differential expression among the $k \in [1..q]$ different experimental conditions and factor G corresponds whether the expression profiles of the genes $j \in [g,t]$ change (one TF t and one TG g is taken into account at a time). Hence in the application there are q different conditions, in which values of k is: $k = [1,..,q]$; and there are p $(p = 2)$ different gene types $(j \in [g,t])$. In a condition there can be N different replicate, i denotes the index of the replicates in a condition $(i = 1, .., N)$.

To obtain the correlation coefficient η^2, sum of squares (SS) terms, as in Eq. (31), are used in the application. SS is a summation of the variances of different factors. It can also be thought as an unadjusted metric of the distribution.

$$SS_T = SS_C + SS_G + SS_{CG} + SS_{err} \tag{32}$$

where SS_C is the SS of the experimental conditions; SS_G is the SS of the genes (t or g); SS_{err} is the SS of the replicates in a condition.

Furthermore, for $x \in [C, G, CG, err, T]$, the variance V_x can be obtained by dividing the SS_x with a degree of freedom parameter df_x. df_x is the number of the interested data points minus 1. For instance, in a matrix of q conditions, the degrees of freedom for factor C are given by $df_C = q - 1$, while the total degrees of freedom are $df_T = M - 1$.

F-value is obtained by dividing the *effect variance* with the *error variance*. For example, the significance of the expression across the conditions can be found by $F_C = V_C / V_{err}$.

Interactions between the TF:TG pairs by two-way ANOVA is:

$$\eta^2_+ = \frac{SS_C}{SS_T}, \quad F_{\eta+} = \frac{V_C}{V_T} \tag{33}$$

Let us examine the calculation of the SSs. We firstly need a data matrix which includes the fold change values.

The size of the matrix is $M = Npq$. Each element of the matrix is f_{ijk}, which denotes the fold change (see Section 4) of the i-th replicate of the k-th experimental condition of the gene j (j can be a TF or a TG). Number of the replicates is N. Thus $i = 1,..,N$. Number of gene types is p. Thus $j = 1..p$ ($p = 2$, TF and TG). Number of the experimental conditions q, hence $k = 1,..,q$.

$$SS_C = x_{.j.} - x_{...} \tag{34}$$

$$SS_T = x_{ijk} - x_{...} \tag{35}$$

where $x_{...} = \frac{1}{M}\left(\sum_{i=1}^{N}\sum_{j=1}^{p}\sum_{k=1}^{q} f_{ijk}\right)^2$; $x_{ijk} = \sum_{i=1}^{N}\sum_{j=1}^{p}\sum_{k=1}^{q} f_{ijk}^2$ and $x_{.j.} = \frac{1}{Nq}\sum_{j=1}^{p}\left(\sum_{i=1}^{N}\sum_{k=1}^{q} f_{ijk}\right)^2$

Furthermore, the variances can be obtained by dividing the sum of squares by their degrees of freedom, df_x, as mentioned above.

Unlike the PCC, η^2 can not directly result in the negative correlations. Küffner et al. proposed to find a new measure (η_-^2) by inverting the sign of the TF fold changes. The maximum of the η_+^2 and η_-^2 is taken as the measure of η^2. Statistical significance can be evaluated by the $F_{\eta+}$ given in (33).

A possible interaction between TF and TG can be constructed if TG shows a response of the over-expression or knock-out of TFs. In a situation like this, during the calculation of η^2, the weight of the conditions that includes the gene perturbations affecting the TF is increased. Therefore, a weighting system can be used to construct efficient interactions of TF:TG. The weighting parameter is user-defined and denoted as w_{gp}. Moreover, the authors of [128] claimed that, the methods in the literature generally can measure only the global dependency; these methods could not measure the dependencies that are valid for only a subset of the conditions (local conditions). η^2 can also detect the local correlations in the datasets [128].

A.14. Chao-Shen Estimator

This estimator combines two different approaches, Horvitz-Thompson estimator and Good-Turing correction of ML estimator. ML entropy estimator was given by (27). Empirical probability of the k-th bin is $\hat{\theta}_k^{emp} = \frac{n_k}{N}$, where the number of the samples in the k-th bin is denoted by n_k and number of the all samples is N. From (27) the empirical entropy estimation equals to:

$$H_{emp} = -\sum_{k=1}^{b} \hat{\theta}_k^{emp} \log\left(\hat{\theta}_k^{emp}\right) \tag{36}$$

Good-Turing correction of the empirical (ML) estimator for the probability of the k-th bin is:

$$\hat{\theta}_k^{GT} = \left(1 - \frac{m_1}{N}\right)\hat{\theta}_k^{emp} \tag{37}$$

where m_1 is the number of bins with observation count equals to 1, i.e. the number of the bins with $n_k = 1$. According to Horvitz-Thompson estimator, the resulting entropy estimation is:

$$\hat{H}^{CS} = -\sum_{k=1}^{b} \frac{\hat{\theta}_k^{GT} \log\left(\hat{\theta}_k^{GT}\right)}{\left(1 - \left(1 - \hat{\theta}_k^{GT}\right)^n\right)} \tag{38}$$

A.15. B-spline (BS) estimator

Because there are no assumptions about the data, BS is a nonparametric method. BS is a kind of spline functions, called as *basis spline* alternatively. Spline is a piecewise-defined polynomial function. The connection points of the splines are known as knots. A spline function with a particular degree, smoothness and domain partition can be defined by a linear combination of B-splines of the same degree and smoothness. The control points determine the general shape of the curve and they are used in the definition of continuous polynomial piece-wised functions.

To generate B-splines, firstly a knot vector should be determined. The knot vector determines how and where the control points change the curve. The number of knots is equal to the addition of the number of control points and curve degree. The knot vector separates the parametric space into knot spans. Ranking of the knot vectors should not decrease. Sequential knots can be equal to each other. In this case, length of the knot span becomes zero. The positions of the knots impact the transformation of the parameter space to curve space. Knots usually do not help us for modeling. Establishing the knot vectors by taking into account the variation in the control points is possible.

Daub et al. proposed BS to achieve the numerical estimation of the mutual information for continuous data. In [16], the continuous gene expression data is discretized by dividing the data into bins. In the classical binning approaches, each data point belongs to only one bin. Because of the several noises, the data points near the border of the bins might shift to adjacent bins. The result of the binning can affect the resulting MI. Daub et al. proposed a generalization to classical binning. The data points can belong to more than one bin concurrently. Indicator function used in classical binning (Eq. (39)) is generalized and expressed by the polynomial BS functions as in (41).

$$\theta_i\left(x_u\right) = \begin{cases} 1, & if\ x_u \in a_i \\ 0, & otherwise \end{cases} \tag{39}$$

This indicator function is used for calculating the empirical probabilities of the bins in binning process as:

$$\hat{p}\left(a_i\right) = \frac{1}{N}\sum_{u=1}^{N}\theta_i\left(x_u\right). \tag{40}$$

Each data point is assigned to more than one bin-i and BS functions assign a weight, $\tilde{B}_{i,k}\tilde{B_{i,k}}$, for each of them. Consequently the probability of the i-th bin, $p(a_i)$, becomes:

$$\hat{p}\left(a_i\right) = \frac{1}{N}\sum_{u=1}^{N}\tilde{B}_{i,k}(x_u). \tag{41}$$

Pseudo code of proposed algorithm [16]:

Inputs: x_u, y_u where $u = 1,\ldots,N$; k: spline order of the BS functions; a_i and b_j bins $i = 1\ldots M_x$ and $j = 1\ldots M_y$

Output: MI(X,Y), the mutual information of X and Y

Steps:

1. Calculate the marginal entropy for variable X:

   a. Determine $\tilde{B}_{i,k}(x) = B_{i,k}(z)\tilde{B_{i,k}}\left(x\right) = B_{i,k}(z)$ with $z = (x - x_{\min})\frac{M_x - k + 1}{x_{\max} - x_{\min}} + 1$.

b. Determine M_x weighting coefficients for each x_u from $\tilde{B_{i,k}}(x_u)\tilde{B}_{i,k}(x_u)$

c. Calculate $p(a_i)$ for each bin from all x_u data points by Eq. (41).

d. Calculate the entropy $H(X) = -\sum_{i=1}^{b} \hat{p}\,(a_i) \log\left(\hat{p}\,(a_i)\right)$.

2. Calculate joint entropy of the variables X and Y:

a. Apply steps 1(a) and (b) to both X and Y separately.

b. Obtain joint probabilities p(a_i,b_j) for all M_x x M_y bins

$$p\,(a_i, b_j) = \frac{1}{N} \sum_{u=1}^{N} \tilde{B}_{i,k}\,(x_u) \times \tilde{B}_{j,k}\,(y_u)\,. \tag{42}$$

c. Calculate the joint entropy $H(X,Y)$.

3. Calculate MI$(X,Y) = H(X) + H(Y) - H(X,Y)$.

Data points are assigned to several bins concurrently, with weights defined by BS functions in [16]. Explanation about the order of the BS function and the BS function adaptation of the study [16] is given in detail in Section E of the Appendices.

Daub et al. searched the influence of the spline order (k) on the estimation of the MI. They achieved the experiments from $k=1$ to $k=5$; k is incremented by 1. The largest improvement is obtained by changing the k from 1 (simple binning) to 2. When $k>3$ there is not any significant improvement. They claimed that choice of the number of bins does not affect the resulting MI, as long as it is chosen within a reasonable range [16]. They also searched the influence of the number of bins (M in that study) on the estimation of the MI. They chose the minimum M value with respect to k value ($M \geq k+1$). They incremented M by 1 up to 10. They did not explain why they terminated the examining the M value at 10. It seems that they chose this value arbitrarily. Besides, 10 is a very small number of bins for using in the real genomics datasets.

A.16. Kernel density estimator (KDE)

Using KDE with microarray datasets was proposed in the MI estimation step of the ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) gene network inference algorithm. In the first step, they estimate the MI by using Gaussian Kernel estimator [17]. The probability density function (pdf) of the gene samples can be estimated by Kernel Density Estimators (KDEs), for each gene pair and individually for each gene. After the calculation of individual probability densities for each gene and joint probability densities for gene pairs, MI between the gene pairs can be obtained by (46).

KDEs estimate the function $f(x)$ by using the observations $x_1, x_2, \ldots x_M$, which were chosen from the density function $f(x)$. KDE is defined as:

$$\hat{f}_h(x) = \frac{1}{M} \sum_{i=1}^{M} K_h\,(x - x_i) = \frac{1}{Mh} \sum_{i=1}^{M} K\left(\frac{x - x_i}{h}\right) \tag{43}$$

where $K_h(.)$ is scaled kernel function; $K(.)$ is kernel function, M is number of the observations, h is smoothing parameter or kernel bandwidth. Gaussian kernel is one of the most used kernels. It provides the mathematical convenience. Bandwidth parameter h should provide condition that $h > 0$. Scaled kernel function $K_h(.)$ is defined as: $K_h(x) = {}^{1}/_{h} \times K({}^{x}/_{h})\, K_h(x) = \frac{1}{h} K({}^{x}/_{h})$.

KDEs are similar to histograms. In histogram approach, after putting a bar with a particular height for each data point in the observation dataset, those bars for particular bin widths are summed up to find the distribution of the samples. KDE with Gaussian kernel fits a normal function with a particular variance for each sample, then those Gaussians are summed up to achieve resulting estimator of the searched density. KDEs converge to the actual density faster than the histogram approach for continuous random variables. Furthermore, the resulting density function with KDE is much smoother than the histogram approach.

Individual density estimation of genes X and Y is denoted as $\hat{f}(x)\hat{f}(x)$ and $\hat{f}(y)$ and joint pdf $\hat{f}(x)\hat{f}(x,y)$ can be obtained as:

$$\hat{f}_h(x) = \frac{1}{M} \frac{1}{\sqrt{2\pi}h} \sum_i \exp\left(-\frac{(x-x_i)^2}{2h^2}\right). \tag{44}$$

Joint pdf of genes X and Y is estimated as:

$$f(\vec{z}) = \frac{1}{M} \frac{1}{2\pi h^2} \sum_i \exp\left(-\frac{(\vec{z}-\vec{z}_i)^2}{2h^2}\right) = \frac{1}{M} \frac{1}{2\pi h^2} \sum_i \exp\left(-\frac{(x-x_i)^2+(y-y_i)^2}{2h^2}\right) \tag{45}$$

where $\vec{z}_i \equiv \{x_i, y_i\}$, $i = 1 \ldots M \vec{z}_i \equiv \{x_i, y_i\}$, $i = 1...M$ is a 2-dimensional observation vector, and the kernel function is normal density function.

Finally, estimation of the MI between the genes X and Y is:

$$MI(x,y) = \frac{1}{M} \sum_i \log\left(\frac{f(x_i, y_i)}{f(x_i)\, f(y_i)}\right). \tag{46}$$

Because (44) is evaluated for each i-th and j-th sample of gene X (x_i and x_j), for $i = 1..M$ and $j = 1..M$, the complexity of calculating MI value of one gene pair is $O(M^2)$, where M is the number of the samples. There are $N \times N$ potential gene pairs, where N is the number of the genes in the dataset. Hence total complexity to obtain the MI matrix becomes $O(M^2 N^2)$.

Determining the optimal value of the parameter h is given in Section F of the Appendices.

MI estimation of ARACNE algorithm consists of two main steps:

1. Statistical relationships between the genes g_i and g_j can be defined by the candidate interactions with the estimation of MI, $I(g_i g_j) \equiv I_{ij}$. MIs were eliminated by using a threshold, I_0. This step suffers from false positives because of genes that have indirect relationships may be highly co-regulated without involving a nondegradable association [17].

2. In the second step most of the indirect candidate interactions are eliminated by data processing inequality (DPI). They claimed that DPI is not used in the reverse engineering of genetic networks previously [17].

They expressed that introducing DPI improves the resulting network. Margolin et al. compared those networks by using synthetic network datasets. They used *precision* and *recall* as performance metrics [17]:

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{Precision} = \frac{TP}{TP+FP} \tag{47}$$

where TP is true positive, FN is false negative and FP is false positive [17].

Performance of the network inference algorithms is evaluated by Precision-Recall Curves (PRCs). In detail discussions and comparisons including KDE, are given in Section 3.

A.17. K-nearest neighborhood entropy estimator

Binning process is not required for K-Nearest Neighborbood (KNN) entropy estimator. Firstly we need to find the individual entropies $H(X)$, $H(Y)$ and the joint entropy $H(X,Y)$, and then we can find the mutual information by:

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{48}$$

$$P_k(\in) = \frac{(N-1)!}{1!\,(k-1)!\,(N-k-1)!}\frac{dp_i(\in)}{d\in}p_i^{k-1} \times (1-p_i)^{N-k-1} = k\begin{pmatrix} N-1 \\ k \end{pmatrix}\frac{dp_i(\in)}{d\in}p_i^{k-1}(1-p_i)^{N-k-1}$$

Let us firstly investigate the $k = 1$ case, which corresponds to the first nearest neighborhood estimator. Kozachenko and Leonenko proposed a nonparametric entropy estimator, which depends on the nearest neighbor of the data samples [109]. $\in(i)/2$ is the distance between x_i and its k-th nearest neighbor. The probability distribution of the distance between the datapoint x_i and its k-th nearest neighbor $P_k(\in)$ is given in (49). Thus, there should be $k-1$ datapoints, whose distance to the datapoint x_i is smaller than $\in(i)/2$ at the dimension x. Similarly there should be $N-k-1$ datapoints, whose distance to the datapoint x_i is larger than $\in/2$ at the dimension x. $p_i(\in)$ denotes the sphere-shaped mass of the $\in$ that is centered at the datapoint x_i. By the trinomial formula $P_k(\in)$ becomes as:

(49)

where N is number of the samples. Trinomial formula can be given as: $(a+b+c)^m = \sum_{i,j,k}\begin{pmatrix} m \\ i,j,k \end{pmatrix}a^i b^j c^k = \sum_{i,j,k}\left(\frac{m!}{i!j!k!}\right)a^i b^j c^k$ where i,j,k are the all possible nonnegative indices that satisfy $i+j+k=m$.

From (49), expected value of the $\log p_i(\in)$ becomes as:

$$E[\log p_i(\in)] = \int_0^\infty P_k(\in)\log p_i(\in)\,d\in = k\begin{pmatrix} N-1 \\ k \end{pmatrix}\int_0^1 p^{k-1}(1-p)^{N-k-1}\log p\,dp = \psi(k) - \psi(N) \tag{50}$$

where $\psi(\cdot)$ is the digamma function. $E[\log p_i(\in)]$ is obtained for all of the N-1 datapoints except x_i. Assume a d-dimensional sphere S with the radius $\in/2$ and with the center x_i in a d-dimensional Euclidean space. The volume of the sphere is:

$$V_\in = c_d \in^d = \frac{\pi^{d/2}(\in/2)^d}{\Gamma\left(1+\frac{d}{2}\right)} \tag{51}$$

where $c_d = \pi^{d/2} \Big/ \left(2^d \Gamma\left(1 + {}^{d}/_{2}\right)\right)$ is the d-dimensional unit sphere's volume [97, 109]. From (51), we obtain:

$$p_i(\epsilon) \approx V_{\epsilon}\mu(x_i) = c_d \epsilon^d \mu(x_i) \tag{52}$$

where $\mu(x_i)$ is the constant density for the sample x_i. From (50) and (52), we get:

$$\log \mu(x_i) \approx \psi(k) - \psi(N) - dE(\log \epsilon) - \log c_d. \tag{53}$$

First combining the equations $H(X) = -\int \mu(x) \log \mu(x) dx$, (52), and (53) result in marginal entropy $H(X)$:

$$H(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N} \sum_{i=1}^{N} \log \epsilon(i) \tag{54}$$

where $\epsilon(i)/2$ is the distance between x_i and its k-th nearest neighbor and $\psi(\cdot)$ is the digamma function, as denoted previously. Thus $\psi(x) = \Gamma(x)^{-1} \frac{d\Gamma(x)}{dx}$. This function can be obtained by a recursive form: $\psi(x+1) = \psi(x) + {}^{1}/_{x}$ and $\psi(1) = -\gamma$. γ is the Euler constant which equals to: 0.5772156... Note that the digamma function can be used as: $\psi(x) = \log x - \frac{1}{2x}$ for larger sample numbers (N).

Marginal entropy $H(Y)$ can be obtained by (54) similarly. To calculate MI, firstly individual entropies of X and Y can be obtained by (54). In order to find joint entropy $H(X,Y)$, first the random variables x_i and y_i are assumed as one point $z_i = (x_i, y_i)$. Then k-th nearest point to the point z_i is searched. The distance between the point z_i and its k-th neighbor in the (x, y) space is denoted by $\epsilon(i)/2$ anymore. Furthermore dimension should be multiply by 2 because of the fact that, $dz = dx + dy$. In the genomics applications it is 1+1=2 for gene expression datasets. The joint entropy $H(X,Y)$ can also be obtained by (54) by the appropriate replacements. Finally, the mutual information MI(X,Y) is obtained by (48).

Direct MI estimation by KNN version 1 and 2 proposed in the study of [97], will be given in the following subsection.

A.18. K-nearest neighborhood (KNN) direct MI estimator-1 and KNN direct MI estimator-2

Kraskov et al. proposed to obtain MI(X,Y) directly by considering the individual and joint entropies simultaneously [97]. They proposed two different approaches for this goal. $MI^{(1)}(X,Y)$ denotes the first approach, $MI^{(2)}(X,Y)$ denotes the second one. In the first approach, the individual distributions of the variables X and Y should be found. Therefore $n_x(i)$ and $n_y(i)$ values are obtained by checking the points $x_j \leq x_i \pm \epsilon(i)/2$ and $y_j \leq y_i \pm \epsilon(i)/2$ where $\epsilon(i)/2$ is the k-th nearest neighbor to the (x_i, y_i) pair. Actually $\epsilon(i)$ is taken as $\max\{\epsilon_x(i), \epsilon_y(i)\}$ where $\epsilon_x(i)$ is the distance at dimension x and $\epsilon_y(i)$ is the distance at dimension y. Thus, $\epsilon(i)/2$ becomes such as the distance between x_i and its $[n_x(i)+1]$-th neighbor, not a fixed k-th neighbor. In this case at the end, (54) becomes:

$$H(X) = -\frac{1}{N} \sum_{i=1}^{N} \psi[n_x(i) + 1] + \psi(N) + \log c_{d_X} + \frac{d_X}{N} \sum_{i=1}^{N} \log \epsilon(i) \tag{55}$$

Similarly, $n_y(i)$ is acquired by counting the points which satisfy the condition $y \leq y_i \pm \epsilon(i)/2$ as mentioned above. Again we do not use a fixed k-th neighbor for the subspace y, (55) is also used for $H(Y)$ calculation

with appropriate arrangements. From (48), MI(X,Y) directly becomes:

$$MI^{(1)}(X,Y) = \psi(k) - \frac{1}{N}\sum_{i=1}^{N}\{\psi[n_x(i)+1] + \psi[n_y(i)+1]\} + \psi(N) \tag{56}$$

where $\psi(\cdot)$ is the digamma function, as explained previously [97].

An example is given in Figure 6. In this example $n_x(i)$ is 7 and $n_y(i)$ is 6 for $k=1$. Because $\in(i)$ is $\max\{\in_x(i), \in_y(i)\}$, distance $\in(i)$ equals to $\in_y(i)$, and is larger than $\in_x(i)$. Hence, $n_x(i)$ is not correct for the subspace x. So, $n_x(i)$ should be corrected by using the second approach [97]. Figure 7 illustrates the new situation. Shaded rectangulars in Figure 7 should not include any points in the second method. If they include some points, it means that the true entropy value $H(Y)$ is damaged.

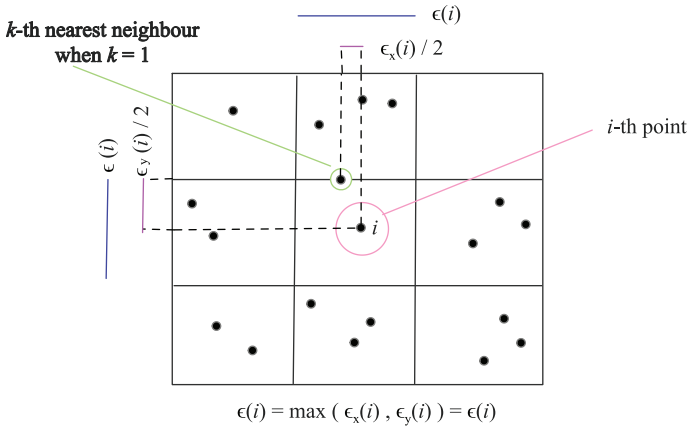

Figure 6. Finding $n_x(i)$ and $n_y(i)$ by using KNN-MI$^{(1)}$ method

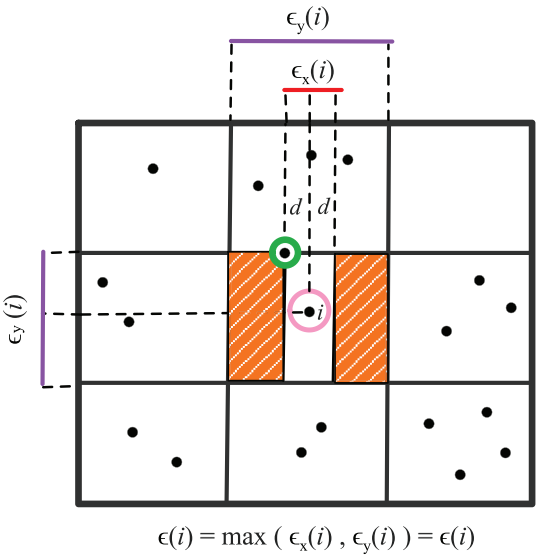

Figure 7. Shaded rectangles are excluded by the second KNN method

In the second approach, both axes are not evaluated with respect to the distance $\in(i)/2$; they are evaluated according to the distances $\in_x/2$ and $\in_y/2$ separately. $n_x(i)$ and $n_y(i)$ values are obtained by checking the points $x_j \leq x_i \pm {}^{\in_x(i)}/_2$ and $y_j \leq y_i \pm {}^{\in_y(i)}/_2$.

In the second approach, the probability distribution of the distance between the point z_i and its k-th nearest neighbor, $P_k(\in)$, is replaced by a 2-D distribution, $P_k(\in_x, \in_y)$. Furthermore, we do not consider a d-dimensional sphere, S, with the radius $\in/2$, and with the center z_i, in a d-dimensional Euclidean space any more. We should assume a rectangular area with a center point $(x_i,\ y_i)$ and with the size of $\in_x \times \in_y$. The mass of this rectangular is denoted by q_i. Expected value of the $\log q_i$ is:

$$E[\log q_i] = \int_0^{\infty}\int P_k(\in_x, \in_y) \log q_i(\in_x, \in_y)\, d\in_x\, d\in_y = \psi(k) - 1/k - \psi(N). \tag{57}$$

Finally we directly obtain the $MI^{(2)}(X,Y)$ by the second approach [97]:

$$MI^{(2)}(X,Y) = \psi(k) - \frac{1}{k} - \frac{1}{N}\sum_{i=1}^{N}\{\psi[n_x(i)+1] + \psi[n_y(i)+1]\} + \psi(N) \tag{58}$$

Kraskov et al. denoted that because the bias caused by the separately estimating of the entropies $H(X)$, $H(Y)$ and $H(X,Y)$ is decreased by not calculating them, the direct MI estimations with KNN version 1 and 2 show better estimation performance and give less bias than the KNN entropy estimator given in the previous subsection. Moreover direct MI estimation with KNN version 2 gives better results than the version 1. In version 1, it is assumed that $\in(i) = \max\{\in_x(i), \in_y(i)\}$. However evaluating both of the axes x and y for the same distance $\in(i)$, most probably causes the erroneous determination of the neighbors' distribution. If the axes x and y are evaluated separately according to the distances $\in_x(i)$ and $\in_y(i)$ respectively, the possible errors in the determination of the distribution of the neighbors is prevented. Therefore, version 2 is a better estimator than the version 1.

Although Suzuki et al. stated that in KNN method the determination of the parameter k is a problem, Kraskov et al. denoted that the parameter k in the KNN entropy estimator, is chosen such as the parameter h in the KDE. If parameter h, in the KDE, is chosen as small, then the statistical error becomes larger. Because of that, generally the parameter h is taken as the $1/2$ or $1/3$ of the total width of the dataset's distribution. Similarly, choosing the parameter k as a large value decreases the statistical error. In the study of Kraskov et al. they proposed that the value of k should be approximately: $\sqrt{k/N} \approx 0.4$ [97]. Therefore, in [99] k was chosen as 6 for sample size N is 40.

A.19. Best upper bound estimator (BUB)

Paninski evaluated ML, MM, Jackknife, and BUB estimators in terms of several statistical point of views such as consistency of central limit theorem, bias and variance. He also examined the confidence intervals for the cases $N << m$, $N >> m$ and $N \tilde{} m$, where N is number of the samples and m is number of the cells. BUB estimator is denoted as a version of the bias correction of MM estimator. Paninski denoted that, BUB estimator gives good results when Miller-Madow estimator fails, when $N \tilde{} m$ [25].

Histogram order statistics, h_j, are used in the BUB estimator:

$$h_j = \sum_{i=1}^{b} 1(n_i = j) \tag{59}$$

where n_i is the number of samples in the i-th bin.

Entropy estimation of a dataset by using H_{BUB}:

$$\hat{H}_{BUB}=\sum_{j=0}^{N} a_{j,N}\times h_j \hat{H}_{BUB} = \sum_{j=0}^{N} a_{j,N} \times h_j \tag{60}$$

where

$$a_{j,N} = -\frac{j}{N}\log\left(\frac{j}{N}\right) + \left(\frac{1-\frac{j}{N}}{2N}\right) a_{j,N}= -\frac{j}{N}\log\left(\frac{j}{N}\right) + \left(\frac{1-\frac{j}{N}}{2N}\right). \tag{61}$$

A.20. First order conditional mutual information (CMI^1)

First order conditional mutual information (CMI^1), is used in the several applications [31,115–119]. We can eliminate the indirect interactions between two variables, which have a nonlinear relationship by nonlinear conditional similarity measures such as CMI^1 as in the study [31]. 0-th order MI estimator was BS method in [31]. Eliminating the linear indirect relationship by PPC^1 and PPC^n are mentioned in the subsections A.7 and A.8.

For a gene pair $(X,\ Y)$, for each remaining gene, Z, a CMI score is obtained by:

$$CMI(X,Y|Z) = H(X,Z) + H(Y,Z) - H(Z) + H(X,Y,Z) \tag{62}$$

The minimum scores among them is selected as CMI^1 score of a pair $(X,\ Y)$. Nonlinear conditioning similarity measure with a higher order requires high computational power. Therefore it was not used in [31].

A.21. Maximal information coefficient (MIC)

Reshef et al. proposed a new metric for defining the association between two random variables. This method is called as Maximal Information Coefficient (MIC). It is a member of the maximal information-based nonparametric exploration (MINE) metrics family. The algorithm proposed in [24] stands on the idea that: if two variables are associated, then a grid can be obtained on the scatterplot of these two variables. While drawing scatterplot of two random variables, values of the first variable are taken as the abscissa values; values of the other random variable are taken as the ordinate values of the resulting points on the scatterplot. Thus the resulting scatter point shows us the interaction of two random variables. Size of the scatterplot is n-by-n, while the number of the samples is n. The grid divides the data to show the relationship between those two variables [24].

The algorithm searches all of the grids, with different dimensions, until reaching a maximal grid resolution. They try to find the maximum MI value between integer pairs x and y when any x-by-y grid applied to the data. Then MI values are normalized and assigned between interval [0,1] to provide a fair comparison between several grid sizes. Finally the characteristic matrix, $\mathbf{M} = (m_{x,y})$ is obtained. Cell $m_{x,y}$ denotes the largest mutual information value obtained by any x-by-y grid. The maximum value in the matrix $\mathbf{M}$ is taken as the statistic MIC.

For a grid G, I_G is the MI of the probability distribution that is defined for the boxes of G. Probability of a box has a ratio with the number of the samples in the box. Cell $m_{x,y}$ becomes as $\max\{I_G\}/\log(\min\{x,y\})$. I_G is only based on the rank of the data. MIC is a symmetric matrix, therefore MIC$(X,\ Y)$ = MIC$(Y,\ X)$.

Let us assume that we have a rank ordered pairs set, D. In order to calculate association between two random variables, we generate a grid on the scatterplot of those two variables. In order to obtain an x-by-y

grid, we divide the x-values of D into x bins and divide the y-values of D into y bins. MIC of a set D, which belongs to two random variables, is obtained by:

$$MIC(D) = \max_{xy<B(n)} \{\mathbf{M}(D)_{x,y}\} \tag{63}$$

where n is number of the samples and $B(n)$ is the maximum possible grid size. In the study $B(n) = n^{0.6}$ is chosen by default [24]. Discussions and comparisons including this method are given in Section 3.

A.22. Distance correlation (dCor) estimator

Szekely et al. proposed a new metric which measures and checks the dependency/independency of the random variables [121]. They call the method as "Distance correlation (dCor)". This metric is similar to standard correlation definition. However, there is an important difference between the dCor and standard correlation. If the standard correlation value between two random variables, X and Y, $\rho(X,Y)\rho(X,Y)$is zero, the random variables X and Y do not have to be independent, for instance they might have a nonlinear relationship. However if the distance correlation (dCor) value of those variables, $\Re(X,Y)R(X,Y)$, is zero; those two variables are absolutely independent. Furthermore, standard correlation value of the random variables is between the interval [-1, +1], i.e. $-1 \leq \rho \leq 1 - 1 \leq \rho \leq 1$; but distance correlation value is between the interval [0, 1], i.e. $0 \leq \Re \leq 10 \leq R \leq 1$. Note that $\Re(X,Y) \leq |\rho(X,Y)|\, R(X,Y) \leq |\rho(X,Y)|$ [121].

Distance covariance metric is similar to the standard covariance. It does not have to be between the interval [0, 1] or interval [-1, +1].

At this point it might be useful to remind the standard correlation and covariance expressions before examining the distance correlation and distance covariance. Correlation is joint expected value of two random variables, i.e. E[xy]. It checks whether a linear relationship between two variables exists or not. Correlation expression is given as:

$$Corr(X,Y) = \frac{E\left[(X - E\left[X\right])\left(Y - E\left[Y\right]\right)\right]}{\sigma_X \sigma_Y}. \tag{64}$$

where σ_X and σ_Y are standard deviations of X and Y respectively.

Covariance: Covariance is the expected value of scattering of two random variables from their own expected values. Covariance expression is given as:

$$Cov(X,Y) = E\left[(X - E\left[X\right])\left(Y - E\left[Y\right]\right)\right] \tag{65}$$

In [121] it is claimed that, distance covariance is a more reliable metric than nonmonotone dependency types. Distance correlation is also an efficient metric that measures the dependency. Hence, it can be used in the gene network inference applications to obtain estimation of interaction between the gene pairs. Distance statistics for the observations $(X,Y) = \{(X_k, Y_k) : k = 1, ..., n\}$ $(X,Y) = \{(X_k, Y_k) : k = 1, \ldots, n\}$, is given as:

$$a_{kl} = |X_k - X_l|_p; \quad \bar{a}_{k\cdot} = \frac{1}{n}\sum_{l=1}^{n} a_{kl}; \quad \bar{a}_{\cdot l} = \frac{1}{n}\sum_{k=1}^{n} a_{kl}; \quad \bar{a}_{\cdot\cdot} = \frac{1}{n^2}\sum_{k,l=1}^{n} a_{kl}; \quad A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot} \tag{66}$$

where n is number of the data samples and p is dimension of each X_k and Y_k. For gene expression values, p is 1. B_{kl} values of the random variable Y is obtained similarly. Finally empirical distance covariance equation

is given as:

$$\nu_n^2(X,Y) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}B_{kl} \tag{67}$$

Individual variances of the variables can be given as:

$$\nu_n^2(X) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}^2, \quad \nu_n^2(Y) = \frac{1}{n^2}\sum_{k,l=1}^{n} B_{kl}^2 \tag{68}$$

Finally empirical distance correlation is:

$$\Re_n^2(X,Y) = \begin{cases} \frac{\nu_n^2(X,Y)}{\sqrt{\nu_n^2(X)\nu_n^2(Y)}}, & \nu_n^2(X)\nu_n^2(Y) > 0 \\ 0 & , \ \nu_n^2(X)\nu_n^2(Y) = 0 \end{cases} \tag{69}$$

Distance correlations are used for each gene pair to obtain gene interaction graph. In [121], authors used the simulated X and Y samples for experiments. Genomics or microarray datasets were not used in [121].

Calculation of the distance correlation is simple, as it measures the dependency accurately. It can even detect the dependency/independency of the nonlinear or nonmonotone relationships. Pruning of the independent genes from the gene interaction matrix can be achieved by using distance correlation metric.

A.23. Heller, Heller, Gorfine (HHG) estimator

Heller et al. proposed a new test for checking independency between random vectors. In the literature independency tests of variables are very few. Mostly dependency tests are used for assessing the association. Their aim was to construct a consistent and multivariate independency test statistic. They call the test statistic as HHG. The null hypothesis in [126] is independency of two multivariate random variables. Discussions and comparisons including this method are given in Section 3.

The joint distribution of X and Y can be denoted by a region, which is centered at the point $(x_0,\ y_0)$ and has radii R_x and R_y around the x_0 and y_0. The center point and the radii at the both dimensions cannot be known exactly. For consistent and accurate test statistics $x_0,\ y_0,\ R_x$ and R_y should be chosen appropriately.

The proposed metric is based on the pairwise difference of the random variables X and Y, $\{d_X(x_i,x_j) : i,j \in \{1,\ldots,N\}\}, \{d_Y(y_i,y_j) : i,j \in \{1,\ldots,N\}\}$. The metric is a function of those pairwise distances. $d_X(.,\cdot)d_X(.,.)$ and $d_Y(.,.)d_Y(.,.)$ are calculated by a norm distance metric.

An indicator function I is defined as: if the expression inside the function is true, value of the function is 1; otherwise value is 0. While number of the samples is N, cross-classification of the observations of the random variables is summarized in the Table 5, for instance calculation of A_{11} is:

$$A_{11} = \sum_{k=1}^{N} I\left\{d\left(x_0,x_k\right) \le R_x\right\} I\left\{d\left(y_0,y_k\right) \le R_y\right\} \text{ where } k \in \{1,\ldots,N\}. \tag{70}$$

Tables 5 and 6 are adapted from [126] without changing. For defining the test statistics as consistent and accurate $x_0,\ y_0,\ R_x$ and R_y should be chosen appropriately as mentioned above. In order to choose the most appropriate values for those parameters, Heller et al., used each $(x_i,\ y_i)$ instead of $(x_0,\ y_0)$ and calculated the

$R_x = \mathrm{d}(x_i,\, x_j)$ for each $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, N\}$. Similarly R_y, for each i-th and j-th observations, is obtained. Thus for the rest N-2 samples, $k \in \{1, \ldots, N\}$ $(k \in \{1, \ldots, N\}$ where $k \neq i$ and $k \neq j)$, the A_{11}, A_{12}, A_{21} and A_{22} values are calculated [126] as given in Table 6. $A_1 \cdot = A_{11} + A_{12}$ and so on.

Table 5. The cross-classification of $I\{d(x_0,x_k) \leq R_x\}$ $I\{d(x_0, x_k) \leq R_x\}$ and $I\{d(y_0,y_k) \leq R_y\}$ $I\{d(y_0, y_k) \leq R_y\}$ [34]

	$d(y_0, \cdot) \leq R_y$	$d(y_0, \cdot) > R_y$	
$d(x_0, \cdot) \leq R_x$	A_{11}	A_{12}	$A_1 \cdot$
$d(x_0, \cdot) > R_x$	A_{21}	A_{22}	$A_2 \cdot$
	$A \cdot_1$	$A \cdot_2$	N

Table 6. The cross-classification of $I\{d(x_i, x_k) \leq d(x_i, x_j)\}$ and $I\{d(y_i, y_k) \leq d(y_i, y_j)\}$ [34]

	$d(y_i, \cdot) \leq d(y_i, y_j)$	$d(y_i, \cdot) > d(y_i, y_j)$	
$d(x_i, \cdot) \leq d(x_i, x_j)$	A_{11}	A_{12}	$A_1 \cdot$
$d(x_i, \cdot) > d(x_i, x_j)$	A_{21}	A_{22}	$A_2 \cdot$
	$A \cdot_1$	$A \cdot_2$	N

Checking the independence can be achieved by Pearson's chi-square test or likelihood ratio test. Heller et al used Pearson's chi-square test to evaluate the test statistics, which are obtained from A_{11}, A_{12}, A_{21} and A_{22} values for each selection of i and j. Pearson's chi-square test is given as:

$$T = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} S(i,j) \tag{71}$$

where

$$S(i,j) = \frac{(N-2)\left\{A_{12}(i,j)\, A_{21}(i,j) - A_{11}(i,j)\, A_{22}(i,j)\right\}^2}{A_{1\cdot}(i,j)\, A_{2\cdot}(i,j)\, A_{\cdot 1}(i,j)\, A_{\cdot 2}(i,j)}. \tag{72}$$

When the value of $S(i,j)$ gets bigger, the dependency between variables X and Y increases. Furthermore if $S(i,j)$ is large and $d(x_i,x_j)\, d(x_i, x_j)$ and $d(y_0,y_k)\, d(y_i, y_j)$ are small, random variables X and Y are dependent in the region around x_i and y_i with the radii $d(x_i, x_j)$ and $d(y_i, y_j)$.

In order to reduce computational complexity from N^3 to $N^2 \log N$ while choosing the parameters x_0, y_0, R_x and R_y, for a fixed i-th sample, they sorted the rest samples according to their distance to i-th sample in X. Hence j-th observation corresponds to j-th nearest sample to the sample i in X. They defined the rank of the distance from i-th sample in Y as $\pi(1)$, $\pi(2)$, ..., $\pi(N\text{-}1)$. j-th observation corresponds the $\pi(j)$-th nearest sample to the sample i in Y. $\pi(1)$, $\pi(2)$, ..., $\pi(N\text{-}1)$ values are a permutation of the array 1,..., N-1. The values in Table 6 are obtained by using π(j) and inv(j), where inv(j) is number of inversions of j in the permuted array π. In other words, inv(j) is the number of indices $m \in \{1, \ldots, j-1\}$ that satisfy $\pi(m) \in \{\pi(j)+1, \ldots, N-1\}$.

Thus A_{12} and A_{22} becomes $A_{12}(i,\, j) =$ inv(j) and $A_{22}(i,\, j) = N$ – $\pi(j)$ – inv(j).

Since $A_1 \cdot (i,\, j) = j$ – 1, A_{11} and A_{21} becomes $A_{11}(i,\, j) = j$ – 1– inv(j) and $A_{21}(i,\, j) = \pi(j)$+ inv(j) – j – 1.

A.24. Jackknife estimator

In this estimator approach, a data sample at each time point is left out and the entropy of the rest is calculated. Entropy estimation of the interested random variable when the sample i is left out, is denoted by $\hat{H}_{(i)}$. Average entropy estimation of the gene X when each of the samples is left out at each time point respectively, is obtained by $\hat{H}_{(.)} = \frac{1}{N}\left(\sum_{i=1}^{N} \hat{H}_{(i)}\right)$. In this case, bias estimation of the Jackknife estimator becomes $Bias_{JK} = (N-1)\left(\hat{H}_{(.)} - \hat{H}\right)$, where $\hat{H}$ can be the maximum likelihood estimation of the entropy or any other estimation of the entropy. It can be obtained by the several biased estimators. Hence the entropy value of the gene X can be obtained by Jackknife estimator [26,77–81]:

$$\hat{H}_{JK} = \hat{H} - Bias_{JK} = \hat{H} - (N-1)\left(\hat{H}_{(.)} - \hat{H}\right) = N\hat{H} - (N-1)\,\hat{H}_{(.)}. \tag{73}$$

Briefly, entropy estimator $\hat{H}$ in (73) can be any one of the estimators. If it is a parametric estimator, the resulting estimator becomes parametric; if it is chosen from the nonparametric estimators, the resulting Jackknife estimator becomes a nonparametric estimator. Therefore, essentially the Jackknife estimator can belong to both parametric and nonparametric classes.

A.25. James-Stein shrinkage estimator

The authors of [23] proposed semiparametric James-Stein shrinkage estimator. This method provides the common usage of two different approaches. Determination of which one of those approaches is more appropriate is done by a weighting parameter. The first approach is ML estimation of the mean value and it is unbiased. Since any assumption about the distribution and the relationship does not exist, this part of the James-Stein shrinkage estimator is nonparametric. The target mean of the second approach is $t_k = {}^{1}/_{b}$, where b is number of the bins. The second part of the shrinkage estimator assumes that all of the bins (cells) have the same frequency; hence this part is a parametric estimator. The first part is with high variance and low bias; the second one is with high bias and low variance. The aim of an ordinary estimator is keeping both of the bias and variance at minimum. The combination of those approaches can be denoted b

$$\hat{\theta}_k^{Shrink} = \lambda t_k + (1-\lambda)\,\hat{\theta}_k^{ML} \tag{74}$$

where $\hat{\theta}_k^{Shrink}$ is the probability of each bin according to the shrinkage approach, λ is the weighting parameter of two approaches [14, 15, 23]. Optimal value of the weighting parameter, λ, can be obtained by:

$$\hat{\lambda}^* = \frac{\sum_{k=1}^{b} \mathrm{Var}\left(\hat{\theta}_k^{ML}\right)}{\sum_{k=1}^{b}\left(t_k - \hat{\theta}_k^{ML}\right)^2} = \frac{\sum_{k=1}^{b} \hat{\theta}_k^{ML}\left(1-\hat{\theta}_k^{ML}\right)}{(N-1)\sum_{k=1}^{b}\left(t_k - \hat{\theta}_k^{ML}\right)^2} \tag{75}$$

where $\mathrm{Var}\left(\hat{\theta}_k^{ML}\right) = \frac{\hat{\theta}_k^{ML}\left(1-\hat{\theta}_k^{ML}\right)}{N-1}$ which is the unbiased estimator of the mean value.

Finally the entropy estimation becomes:

$$\hat{H}^{Shrink} = -\sum_{k=1}^{b} \hat{\theta}_k^{Shrink} \log\left(\hat{\theta}_k^{Shrink}\right) \tag{76}$$

B. Appendix. An Example to Understand the Partial Correlation

Assume that there are true direct interactions between variables X and Z; and between Y and Z. However direct interaction between X and Y does not exist (see Figure 8). Although there is not a direct interaction between X and Y, there is a high correlation between those two variables. Because it denotes the correlation of the X's and Y's uncorrelated parts with Z, association of X and Y exists via Z. Thus, the PPC of X and Y conditioning on Z might be below the significance (threshold) value.

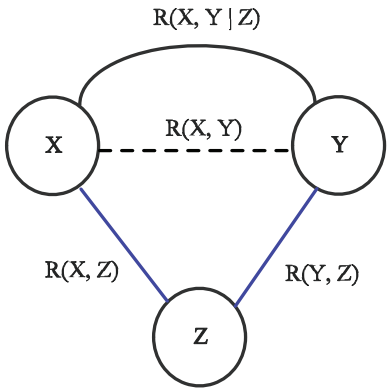

Figure 8. Variables X and Y is interacted via variable Z

While we are testing the interaction between the variables X and Z, we calculate the first order PPC by conditioning on the all variables, except X and Z, respectively and we check whether the PPC values below or above the significance threshold value. Finally we saw that all of the PPC values are above the threshold value. Thus, the interaction between X and Z is protected.

Interaction between variables Y and Z is checked as mentioned above. At the end, the edge between variables Y and Z is also protected because of the same reasons.

Let us examine the relation between the variables X and Y. The correlation between them is very high because both of them directly interacted with variable Z. The first order PPC between X and Y while conditioning on the variable Z is below the significance value. Hence the edge between X and Y is removed from the graph. After that, PPC between X and Y is no longer examined by conditioning on other variables. As soon as the PPC value becomes below the threshold value, we stop our investigation about the relationship of X and Y.

After all of the correlations are examined as described above, the edge between the variables that have nonsignificant partial correlations conditioning on any other variable/variables, are removed from the graph. Hence a graph, which is assumed to have only direct interactions, is acquired at the end. The resulting graph is undirected because correlation is a symmetric metric.

The possible PPC[1] relationships of the genes X and Y are shown as in Figure 9.

C. Appendix. An Example of the ML Estimator

Values of the random variables X and Y in the Figure 10 are:

x=[0.73; 0.11; 0.90; 0.98; 0.74; 0.11; 0.67; 0.83; 0.77; 0.93; 0.96; 0.21; 0.69; 0.93; 0.18; 0.42]

y=[0.78; 0.62; 0.85; 0.91; 0.33; 0.58; 0.08; 0.74; 0.83; 0.88; 0.93; 0.30; 0.12; 0.25; 0.77; 0.51];

Discretization of the dataset for both of the random variables is achieved by equal frequency. Assume that bin number is $b = \sqrt{N} = 4$, when number of samples is N =16.

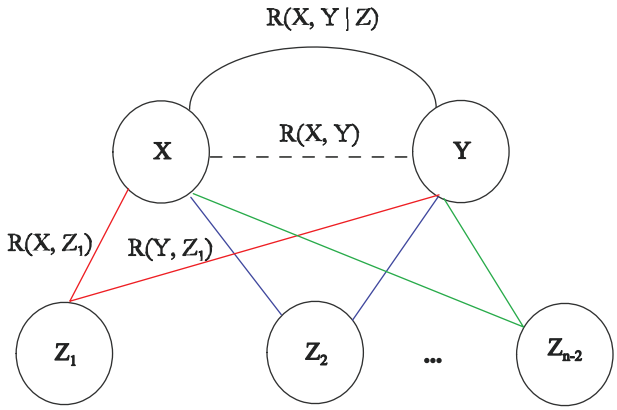

Figure 9. Illustration of possible conditional relationship (PPC[1]) of X and Y

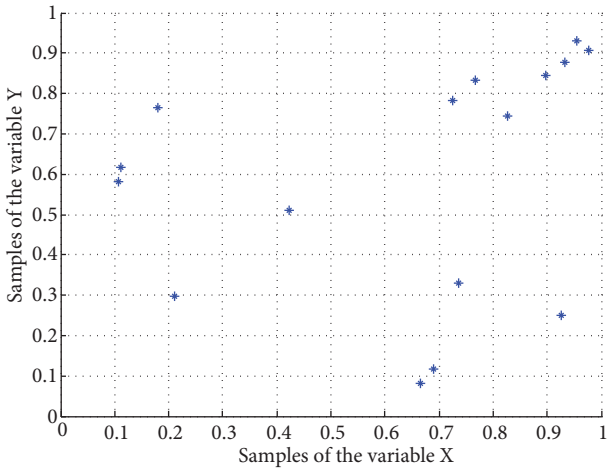

Figure 10. An example distribution (identical to the Fig 5a)

Firstly, the individual entropies of the random variables should be obtained by (27). For the random variable X, there are 4 different bins and in the equal frequency case the probability of each bin becomes 4/16.

$$\begin{aligned} H_{emp}(X) &= -\sum_{k=1}^{b}\left(\tfrac{n_k}{N}\right)\log\left(\tfrac{n_k}{N}\right) = -\tfrac{4}{16}\times\log\left(\tfrac{4}{16}\right) - \tfrac{4}{16}\times\log\left(\tfrac{4}{16}\right) - \tfrac{4}{16}\times\log\left(\tfrac{4}{16}\right) - \tfrac{4}{16}\times\log\left(\tfrac{4}{16}\right) \\ &= 0.6021 \end{aligned}$$

Similarly the individual entropy of the random variable Y, $H_{emp}(Y)$, becomes 0.6021.

The joint entropy of the variables X and Y is:

$$H_{emp}(X,Y) = -\sum_{k=1,l=1}^{b} p(bin\_x_k, bin\_y_l)\log p(bin\_x_k, bin\_y_l) \tag{77}$$

where $p(bin\_x_k, bin\_y_l)$ denotes the joint probability of the X samples belonging to k-th bin of the X and Y samples belonging to l-th bin of the Y.

$$p(bin\_x_k, bin\_y_l) = \sum_{i=1}^{N}\frac{\delta\left(x_i \in bin\_x_k\right)\times\delta\left(y_i \in bin\_y_l\right)}{N} \tag{78}$$

where $\delta\left(expression\right)$ is the indicator function. The value is 1 if the expression is true, otherwise it is 0.

$p(bin\_x_1, bin\_y_1) = {}^{1}/_{16}$ => there is only one index i that satisfies $x_i \in$ $1st\ bin\ of\ X$ $(i.e.\ bin\_x_1)$ and $y_i \in$ $1st\ bin\ of\ Y$ $(i.e.\ bin\_y_1)$ at the same time.

$p(bin\_x_1, bin\_y_2) = {}^{2}/_{16}$ => there are two indices i which satisfy $x_i \in$ $1st\ bin\ of\ X$ $(i.e.\ bin\_x_1)$ and $y_i \in$ $2nd\ bin\ of\ Y$ $(i.e.\ bin\_y_2)$ at the same time.

The process continues similarly for all possible $4 \times 4 = 16$ bin pairs of the variables X and Y:

$p(bin\_x_4, bin\_y_4) = {}^{3}/_{16}$ there are three indices i which satisfy $x_i \in$ $4th\ bin\ of\ X$ $(i.e.\ bin\_x_4)$ and $y_i \in$ $4th\ bin\ of\ Y$ $(i.e.\ bin\_y_4)$ at the same time.

From the Eq. (77):

$$H_{emp}(X,Y) = -\sum_{k=1,l=1}^{b} p(bin\_x_k, bin\_y_l) \log p(bin\_x_k, bin\_y_l)$$
$$= -7 \times \tfrac{1}{16} \times \log\left(\tfrac{1}{16}\right) - 3 \times \tfrac{2}{16} \times \log\left(\tfrac{2}{16}\right) - \tfrac{3}{16} \times \log\left(\tfrac{3}{16}\right) = 1.0018$$

$$MI_{emp}(X,Y) = H_{emp}(X) + H_{emp}(X) - H_{emp}(X,Y) = 0.6021 + 0.6021\text{-}1.0018 = 0.2024$$

Finally we obtain the MI between the variables X and Y by using the empirical estimator.

D. Appendix. Fold Change Calculation in the ANOVA Correlation Estimator

Each of the conditions m is a collection of different gene, drug and environmental perturbations. Gene perturbations of the control conditions should be less than that of the measurement condition m. We could not have control conditions for a condition m if it has a low perturbation level and if there cannot be any less perturbation combination possibility. Thus, some of the experiments do not allow controlling conditions [128].

m_1, m_2, m_3 are the replicates of the condition m in Figure 11. Perturbations of the control conditions a and b are less than the perturbation of the condition m. Average values of the control conditions, $\bar{m}^a$ and $\bar{m}^b$ are obtained by:

$$\overline{m^a} = \frac{1}{|m^a|}\sum_k m_k^a \text{ and } \overline{m^b} = \frac{1}{|m^b|}\sum_k m_k^b \tag{79}$$

Fold changes for the replicates of m, (m_1, m_2, m_3) are obtained as:

$$f_i^a = m_i - \overline{m^a} \text{ and } f_i^b = m_i - \overline{m^b} \tag{80}$$

Six different fold changes are computed for the example in Figure 11. One fold change is calculated for each of the m_i replicate and for each of the conditions (a and b).

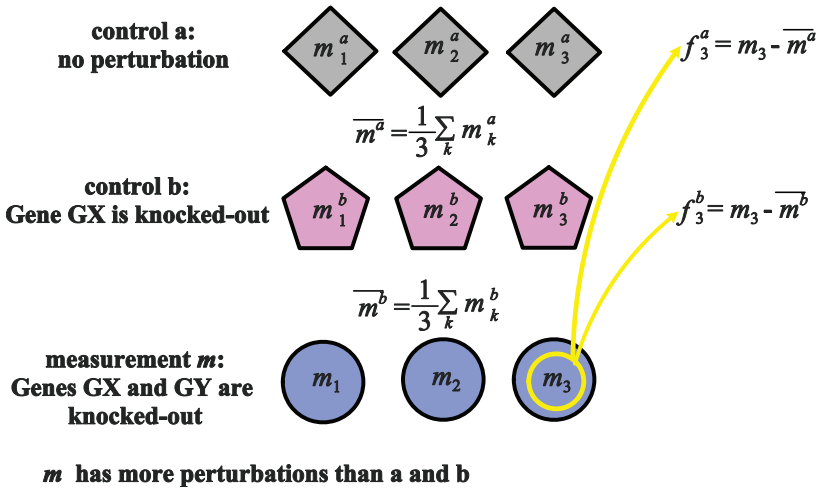

Figure 11. Transformation of the expression values into the fold changes

DREAM5 E. Coli dataset involves 805 chip measurements of 487 different measurement conditions. Among them Küffner et al., decided controls for 599 chips that correspond to 379 conditions. They obtained 935 fold changes from those 599 chips.

Possible associations between the pair TF:TG is ranked according to a score, s. This score can be any one of the dependency metrics such as PCC, SCC or MI, etc. They proposed the usage of score η^2, as denoted in the subsection A.13.

E. Appendix. Order of B-splines, BS adaptation of the study, and an example

Order of a curve depicts the number of close control points that impact any point on the curve. The curve is illustrated by a polynomial with the degree equal to the order of the curve minus 1. "Number of the control points" should be greater than or equals to the "curve's order".

A BS of order k is a parametric curve:

S: $[t_{k+1}, t_{m-k}]$ ->R when given m knot values, t_i, with $t_1 \leq t_2 \leq \cdots \leq t_m$

BS is linear combination of basis B-splines $B_{i,k}$ of degree k:

$$S(z) = \sum_{i=1}^{m-k-1} P_i B_{i,k}(z), z \in [t_{k+1}, t_{m-k}] \tag{81}$$

Basis B-splines of degree k, $B_{i,k}$ can be specified by Cox-de Boor recursion formula:

$$B_{i,1}(z) = \begin{cases} 1, & \text{if} \quad t_i \leq z \leq t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{82}$$

$$B_{i,k}(z) = B_{i,k-1}(z)\frac{z - t_i}{t_{i+k-1} - t_i} + B_{i+1,k-1}(z)\frac{t_{i+k} - z}{t_{i+k} - t_{i+1}} \tag{83}$$

For a BS of degree k, with m knot points, there are $m - k$-1 control points and basis B-splines.

BS adaptation of the study [16]:

Positions of the knots specify the shape of the basis functions. Definition of knot vector is mentioned above. Knot vector depends on the application. In [16] knot vector is defined by:

$$t_i = \begin{cases} 0, & \text{if} \quad i < k \\ i-k+1, & \text{if} \quad k \le i \le M-1 \\ M-1-k+2, & \text{if} \quad i > M-1 \end{cases} \tag{84}$$

where M is the number of bins, k is the order of the spline.

Furthermore k tells that, one datapoint belongs to k different bins simultaneously. Also size of the knot vector is $M+k$.

Knot vector defined above does not exist in other references than [16]. However in the implemented code, it is defined as:

$$t_i = i-1, \text{ where } i = 1, \ldots, M+k \tag{85}$$

An example: Estimation with BS binning:

Firstly t_i values should be found by using (85). There are $M+k$ values of t, thus $t_1=0, t_2=1, t_3=2, t_4=3, t_5=4$, while $M=3$, $k=2$.

For random variable X, according to $z=(x-x_{min})\frac{M_x-k+1}{x_{max}-x_{min}}+1$:

$$z=(x-x_{min})\frac{M_x-k+1}{x_{max}-x_{min}}+1=(x-0)\frac{3-2+1}{1.0-0.0}+1=2x+1$$

Thus $B_{i,k}(z)=B_{i,k}(2x+1)$, where $B_{i,k}$ represents the membership of the data point to the bin i when the degree of BS is k.

Indicator function is illustrated by degree of k B-splines, $B_{i,k}$. Hence $B_{i,k}$ is calculated according to (82) and (83):

$$x=0.0 \rightarrow z=2\times 0.0+1 \rightarrow B_{1,1}(1)=0; B_{2,1}(1)=1; B_{3,1}(1)=0; B_{4,1}(1)=0;$$

$$B_{1,2}(1)=B_{1,1}(1)\frac{1-t_1}{t_2-t_1}+B_{2,1}(1)\frac{t_3-1}{t_3-t_2}=1$$

$$B_{2,2}(1)=B_{2,1}(1)\frac{1-t_2}{t_3-t_2}+B_{3,1}(1)\frac{t_4-1}{t_4-t_3}=0$$

$$B_{3,2}(1)=B_{3,1}(1)\frac{1-t_3}{t_4-t_3}+B_{4,1}(1)\frac{t_5-1}{t_5-t_4}=0$$

$$x=0.2 \rightarrow z=1.4 \rightarrow B_{1,1}(1.4)=0; B_{2,1}(1.4)=1; B_{3,1}(1.4)=0; B_{4,1}(1.4)=0;$$

$$B_{1,2}(1.4)=B_{1,1}(1.4)\frac{1.4-t_1}{t_2-t_1}+B_{2,1}(1.4)\frac{t_3-1.4}{t_3-t_2}=0.6$$

$$B_{2,2}(1.4)=B_{2,1}(1.4)\frac{1.4-t_2}{t_3-t_2}+B_{3,1}(1.4)\frac{t_4-1.4}{t_4-t_3}=0.4$$

$$B_{3,2}(1.4)=B_{3,1}(1.4)\frac{1.4-t_3}{t_4-t_3}+B_{4,1}(1.4)\frac{t_5-1.4}{t_5-t_4}=0$$

$$x=0.4 \rightarrow z=1.8 \rightarrow B_{1,1}(1.8)=0; B_{2,1}(1.8)=1; B_{3,1}(1.8)=0; B_{4,1}(1.8)=0;$$

$$B_{1,2}(1.8)=B_{1,1}(1.8)\frac{1.8-t_1}{t_2-t_1}+B_{2,1}(1.8)\frac{t_3-1.8}{t_3-t_2}=0.2$$

$$B_{2,2}(1.8)=B_{2,1}(1.8)\frac{1.8-t_2}{t_3-t_2}+B_{3,1}(1.8)\frac{t_4-1.8}{t_4-t_3}=0.8$$

$$B_{3,2}(1.8)=B_{3,1}(1.8)\frac{1.8-t_3}{t_4-t_3}+B_{4,1}(1.8)\frac{t_5-1.8}{t_5-t_4}=0$$

$$x=0.6\rightarrow z=2.2\rightarrow B_{1,1}(2.2)=0;B_{2,1}(2.2)=0;B_{3,1}(2.2)=1;B_{4,1}(2.2)=0;$$

$$B_{1,2}(2.2)=B_{1,1}(2.2)\frac{2.2-t_1}{t_2-t_1}+B_{2,1}(2.2)\frac{t_3-2.2}{t_3-t_2}=0$$

$$B_{2,2}(2.2)=B_{2,1}(2.2)\frac{2.2-t_2}{t_3-t_2}+B_{3,1}(2.2)\frac{t_4-2.2}{t_4-t_3}=0.8$$

$$B_{3,2}(2.2)=B_{3,1}(2.2)\frac{2.2-t_3}{t_4-t_3}+B_{4,1}(2.2)\frac{t_5-2.2}{t_5-t_4}=0.2$$

$$x=0.8\rightarrow z=2.6\rightarrow B_{1,1}(2.6)=0;B_{2,1}(2.6)=0;B_{3,1}(2.6)=1;B_{4,1}(2.6)=0;$$

$$B_{1,2}(2.6)=B_{1,1}(2.6)\frac{2.6-t_1}{t_2-t_1}+B_{2,1}(2.6)\frac{t_3-2.6}{t_3-t_2}=0$$

$$B_{2,2}(2.6)=B_{2,1}(2.6)\frac{2.6-t_2}{t_3-t_2}+B_{3,1}(2.6)\frac{t_4-2.6}{t_4-t_3}=0.4$$

$$B_{3,2}(2.6)=B_{3,1}(2.6)\frac{2.6-t_3}{t_4-t_3}+B_{4,1}(2.6)\frac{t_5-2.6}{t_5-t_4}=0.6$$

$$x=1.0\rightarrow z=3\rightarrow B_{1,1}(3)=0;B_{2,1}(3)=0;B_{3,1}(3)=0;B_{4,1}(3)=1;$$

$$B_{1,2}(3)=B_{1,1}(3)\frac{3-t_1}{t_2-t_1}+B_{2,1}(3)\frac{t_3-3}{t_3-t_2}=0$$

$$B_{2,2}(3)=B_{2,1}(3)\frac{3-t_2}{t_3-t_2}+B_{3,1}(3)\frac{t_4-3}{t_4-t_3}=0$$

$$B_{3,2}(3)=B_{3,1}(3)\frac{3-t_3}{t_4-t_3}+B_{4,1}(3)\frac{t_5-3}{t_5-t_4}=1$$

According to Eq. (41):

$$\hat{p}(a_1)=\frac{\tilde{B}_{1,2}(0)+\tilde{B}_{1,2}(0.2)+\tilde{B}_{1,2}(0.4)+\tilde{B}_{1,2}(0.6)+\tilde{B}_{1,2}(0.8)+\tilde{B}_{1,2}(1)}{N}=\frac{1+0.6+0.2+0+0+0}{6}=\frac{1.8}{6}=0.3$$

$$\hat{p}(a_2)=\frac{\tilde{B}_{2,2}(0)+\tilde{B}_{2,2}(0.2)+\tilde{B}_{2,2}(0.4)+\tilde{B}_{2,2}(0.6)+\tilde{B}_{2,2}(0.8)+\tilde{B}_{2,2}(1)}{N}=\frac{0+0.4+0.8+0.8+0.4+0}{6}=\frac{2.4}{6}=0.4$$

$$\hat{p}(a_3)=\frac{\tilde{B}_{3,2}(0)+\tilde{B}_{3,2}(0.2)+\tilde{B}_{3,2}(0.4)+\tilde{B}_{3,2}(0.6)+\tilde{B}_{3,2}(0.8)+\tilde{B}_{3,2}(1)}{N}=\frac{0+0+0+0.2+0.6+1}{6}=\frac{1.8}{6}=0.3$$

$H(x)$ and $H(y)$:

$H(x) = H(y) =$ -0.3 log2(0.3) – 0.4 log2(0.4) – 0.3 log2(0.3) $=$ 1.57.

Joint probabilities $p(a_i, b_j)$ for all M_x x M_y bins is obtained by Eq. (42), $p(a_i b_j) = \frac{1}{N} \sum_{u=1}^{N} \tilde{B}_{i,k}(x_u) \times \tilde{B}_{j,k}(y_u)$:

$$\begin{aligned} p\left(a_1,b_1\right) &= \frac{\tilde{B}_{1,2}\left(0.0\right) \times \tilde{B}_{1,2}\left(0.8\right) + \tilde{B}_{1,2}\left(0.2\right) \times \tilde{B}_{1,2}\left(1.0\right) + \tilde{B}_{1,2}\left(0.4\right) \times \tilde{B}_{1,2}\left(0.6\right)}{6} \\ &+ \frac{\tilde{B}_{1,2}\left(0.6\right) \times \tilde{B}_{1,2}\left(0.4\right) + \tilde{B}_{1,2}\left(0.8\right) \times \tilde{B}_{1,2}\left(0.0\right) + \tilde{B}_{1,2}\left(1.0\right) \times \tilde{B}_{1,2}\left(0.2\right)}{6} \\ &= \frac{1 \times 0 + 0.6 \times 0 + 0.2 \times 0 + 0 \times 0.2 + 0 \times 1 + 0 \times 0.6}{6} = 0 \end{aligned}$$

Similarly, the joint probabilities are $p\left(a_1,b_1\right) = p\left(a_3,b_3\right) = 0$; $p\left(a_2,b_2\right) = \frac{1.28}{6} = 0.213$;

$$p\left(a_1,b_3\right) = p\left(a_3,b_1\right) = \frac{1.24}{6} = 0.207;$$

$$p\left(a_1,b_2\right) = p\left(a_2,b_1\right) = p\left(a_2,b_3\right) = p\left(a_3,b_2\right) = \frac{0.56}{6} = 0.093;$$

The joint entropy is: $H(x, y)$ = -0.213 log2(0.213) – 2 x 0.207 log2(0.207) – 4 x 0.093 log2(0.093) = 2.7.

Finally the MI score is:

$$MI\left(X,Y\right) = H\left(X\right) + H\left(Y\right) - H\left(X,Y\right) = 1.57 + 1.57 - 2.7 = 0.44$$

NOTE: The results obtained above for the same dataset are same with the results of the code given in [16].

F. Appendix. Determining h Value in the Kernel Density Estimator

Margolin et al. normalized the dataset by copula transform. Thus, variables became between 0 and 1. And also, expression data is ranked. Bandwidth parameter h is the smoothing parameter at the same time. It determines the amount of smoothness of the estimated density. Also it strongly affects the accuracy of the estimator. However ranking of the MI estimations weakly depends on the value of kernel widths, h. MI ranking is stable even MI itself is not certain. In [17] value of h was tried to be optimized for a good estimation by using a small part of dataset. However it is assigned for overall data. They investigated how to determine the optimal value of h in their technical reports [94].

To specify an optimal h value, Margolin et al. benefitted from the study of Duin [95]. In [95] the number of the samples and standard deviation of the samples are used to obtain h value. Thus the approach is based on the dataset. Purpose of that approach is maximizing the posterior probability. According to Bayes Decision Theory, the posterior probability is given as:

$$p\left(h \left| \bar{X}, \bar{Y} \right.\right) \approx p\left(\bar{X}, \bar{Y} \left| h \right.\right) \times p(h) \tag{86}$$

The prior $p(h)$ is estimated weakly as: $p\left(h\right) = \frac{1}{\pi(1+h^2)}$

The likelihood $p\left(\bar{X}, \bar{Y} \left| h \right.\right)$ reaches maximum value at $h = 0$:

$$p\left(\bar{X}, \bar{Y} \mid h\right) \approx \prod_{i=1}^{M} f_h\left(x_i, y_i\right) \tag{87}$$

Cross-validation approach uses $f_{h,i}(x,y)f_{h,i}(x,y)$ instead of $f_h(x,y)$ of Eq. (87). Former is the leave-one-out version of the latter [94]:

$$f_{h,i}(x,y) = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq i}} K_h\left(x - x_j, y - y_j\right) \tag{88}$$

where $K_h(.)$ is the scaled kernel function [94].

Duin denoted that Koontz and Fukunaga proposed calculating h by using the standard deviation of the observation data and number of samples M [96]:

$$h = \left(\frac{1}{2} tr(\hat{\mathbf{C}})\right)^{1/2} \times M^{-\alpha/2} \tag{89}$$

$h = \left(\frac{1}{2} tr(\hat{C})\right)^{1/2} \times M^{-\alpha/2}$ where $\hat{\mathbf{C}}$ is the estimated covariance matrix of the observation dataset and $0<\alpha<0.5$. According to the implemented code of [17], Eq (89) becomes Eq. (90) in ARACNE's estimator. The parameter α is chosen as $1/3$.

$$h = std\_dev\_of\_a\_subset\_of\_data \times \left(\frac{6M}{4}\right)^{-\alpha/2} \tag{90}$$

After obtaining the optimal h value for a particular number of samples, extrapolation of the h value according to number of samples is made. Sample size is started from 100, incremented by 20, until it reaches 360. They calculated optimal h value by maximizing the posterior probability given by Eq. (86). For each sample size three subsets were chosen randomly to obtain optimal h value. The average of h values of those three subsets is taken. Then the scale of h with respect to sample size is assumed as power-law [94]:

$$\hat{h} = \alpha \times (M)^{\beta} \tag{91}$$

$\hat{h} = \alpha \times M^{\beta}$ where M is the sample size at that moment. The regression parameters α and β are calculated by taking logarithm of Eq. (91) and fitting a linear regression model. The resulting model is:

$$h = 0.525 \times (M)^{-0.24}. \tag{92}$$