

**GENE REGULATORY NETWORK INFERENCE USING
K-NEAREST-NEIGHBOR BASED MUTUAL INFORMATION
AND
THREE-NODE NETWORK CLASSIFICATION USING
DIMENSIONALITY REDUCTION AND MACHINE LEARNING**

by
Lior I. Shachaf

A dissertation submitted to The Johns Hopkins University in conformity
with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
March, 2022

© 2022 Lior I. Shachaf
All rights reserved

Abstract

- **Background:** A cell exhibits a variety of responses to internal and external cues. These responses are possible, in part, due to the presence of an elaborate gene regulatory network (GRN) in every single cell. In the past twenty years, many groups worked on reconstructing the topological structure of GRNs from large-scale gene expression data using a variety of inference algorithms. Insights gained about participating players in GRNs may ultimately lead to therapeutic benefits. Mutual information (MI) is a widely used metric within this inference/reconstruction pipeline as it can detect any correlation (linear and non-linear) between any number of variables (n-dimensions). However, the use of MI with continuous data (for example, normalized fluorescence intensity measurement of gene expression levels) is sensitive to data size, correlation strength and underlying distributions, and often requires laborious and, at times, ad hoc optimization.
- **Results:** In this work, we first show that estimating MI of a bi- and tri-variate Gaussian distribution using k-nearest neighbor (kNN) MI estimation results in significant error reduction as compared to commonly used methods based on fixed binning. Second, we demonstrate that implementing the MI-based kNN Kraskov-Stoögbauer-Grassberger (KSG) algorithm leads to a significant improvement in GRN reconstruction for popular inference algorithms, such as Context Likelihood of Relatedness (CLR). Third, through extensive in-silico benchmarking we show that a new inference algorithm CMIA (Conditional

Mutual Information Augmentation), inspired by CLR, in combination with the KSG-MI estimator, outperforms commonly used methods. Finally, we compare our three newly developed methods to classify three-node motifs: (i) MI and Z-score profiles, (ii) Dimensionality reduction by PCA and clustering using K-means, (iii) Supervised machine learning algorithms using MI input data. We show that at least 22 different 3-node motifs *in-silico* and 16 motifs on *E.coli* experimental data can be distinguished by using all 2d and 3d MI quantities and without any *a priori* knowledge of the regulator (source) genes.

- Conclusions: Using three canonical datasets containing 15 synthetic networks, the newly developed method for GRN reconstruction - which combines CMIA, and the KSG-MI estimator - achieves an improvement of 20-35% in precision-recall measures over the current gold standard in the field. Validated on *E. coli* gene expression data, our method for three-node motifs classification achieves more than 60% overall accuracy, with 9 network motifs reaching as high as 80-100% precision. This new methods will enable researchers to discover new gene interactions or choose gene candidates for experimental validations.

Thesis Readers

Dr. Jie Xiao (Primary Advisor)

Professor

Department of Biophysics and Biophysical Chemistry

Johns Hopkins School of Medicine

Dr. Patrick Cahan (Second reader)

Professor

Institute for Cell Engineering

Department of Biomedical Engineering,

Department of Molecular Biology and Genetics

Johns Hopkins School of Medicine

To Mimi, my partner in life, for your love and support throughout this long and strenuous journey.

To Sigal and Shani, I hope this infinitesimal small contribution to human knowledge will leave you a better world to live in.

To the rest of the Shachaf-Hornung and Asnes-Keller clans for their encouragement and unconditional love.

Acknowledgements

I would not have been able to submit my dissertation without the support of my current adviser, professor Jie Xiao. 18 months ago, in the middle of a pandemic, I didn't have an adviser, group or manuscript draft. Jie took me under her wing, so I would have an adviser and group to keep me on the right track and share my ideas and progress. I learned a lot from her leadership, human relationships and critical scientific thinking and writing. I will never be able to thank you enough.

To my former PI Elijah Roberts, I would like to thank his initial support, planting the seeds of my knowledge and allowing me to wander in research space.

I would like to thank professors Margaret Johnson and Brian Camley, who have been with me since my GBO, proposal seminar and advising committee, for your advice on achievable scientific aims, and additional technical meetings when I needed support.

I would like to thank professor Patrick Cahan, for joining my advising committee and helping me turn my unstructured scientific progress into manuscripts.

A special thank you to my friend and colleague Basilio, whom I have discussed science, research and taught me what I didn't know about molecular biology and biochemistry.

I would like to thank my past and present group colleagues for your support and friendship, and especially to Chris, Yuncong and Nico, for sharing our passion to supercoil based regulation.

Finally, I would like to thank my program director professor Greg Bowman for his advice, and the Biophysics faculty and staff for kindness and support and making the department feel like a family.

Contents

Abstract	ii
Dedication	iv
Acknowledgements	v
Contents	vi
List of Tables	x
List of Figures	xi
Chapter 1 Gene regulation network inference using k-nearest neighbor-	
based mutual information estimation	1
Background	1
Materials and Methods	4
Calculate mutual information of multiple variables	4
k-nearest-neighbor (kNN)	6
<i>in-silico</i> GRN Inference comparison:	7
Simulating gene expression data	8
Discretizing/density estimation	8
Mutual Information estimation	8
GRN inference algorithms	8
GRN performance evaluation	10

Results	11
Benchmark MI estimations of a Gaussian distribution	11
<i>in-silico</i> GRN Inference performance enhancement	12
<i>in-silico</i> GRN Inference performance comparison	14
<i>in-silico</i> GRN Inference performance of different organisms	16
Computational cost	18
Discussion	19
kNN-based MI estimator for data discretization/density estimation outperforms fixed-bin-based estimations	20
kNN-based MI estimator KSG in combination with CMIA achieves the highest accuracy but may subject to data stochasticity	21
Chapter 2 Classifying three-node network motifs of Transcription Factor (TF)-based regulation	24
Introduction	24
Materials and Methods	25
Datasets	25
Simulating gene expression data for three-node network motifs	25
DREAM 3 & 4 challenge datasets	26
<i>Escherichia Coli</i> experimental data	27
Calculating mutual information and Z-score	28
Principal Component Analysis (PCA)	28
Clustering algorithm	29
Machine Learning models	29
Performance evaluation	30
Results	31
Using unique Mutual Information profiles to identify network topologies	31
Simulated Two-genes motifs	31

Simulated Three-genes motifs	33
Results of simulated motifs vs. motifs extracted from DREAM3-4 networks	36
Using dimensionality reduction by Principal Components Analysis (PCA) to classify different three-node network motifs	38
PCA for simulated 3-node motifs	38
Network motif classification by Machine Learning (ML) algorithms	43
Identify network motifs on <i>Escherichia Coli</i> expression data	45
Discussion	47
Simulated motifs vs. motifs extracted from DREAM3-4 networks	48
Saturation of accuracy	48
Using more types of expression data	49
Z-score statistics	49
Dimensionality reduction	49
Classification by machine learning	50
Conclusions and general discussion	52
Gene regulation network inference using k-nearest neighbor-based mutual information estimation	52
Classifying three-node network motifs of Transcription Factor (TF)-based regulation	52
References	54
Appendix I Mutual Information overview	58
Introduction to Information Theory	58
Shannon's Entropy	58
Conditional and Joint Entropy	58
Mutual Information	59

Three-Dimensional Mutual Information	59
Interaction Information	59
Total Correlation	60
Uniqueness, Redundancy and Synergy	60
Formalism for Discrete Variables	60
Probability definitions	60
Entropy	61
Information	61
Appendix II Supplementary information for chapter on gene regula-	
tion network inference	62
A. Analytical solution for a multivariate Gaussian distribution	62
B. Miller-Madow correction to Shannon's entropy	62
C. Supplementary tables	63
D. Supplementary figures	67
Appendix III Supplementary information for chapter on network mo-	
tifs classification	75
A. Supplementary text	75
B. Supplementary figures	76

List of Tables

Table 1-1	Mutual Information formalism	5
Table 2-1	Simulated Motifs	27
Table 2-2	Machine Learning models and parameters	31
Table 2-3	Machine Learning classification accuracy on <i>in-silico</i> data . .	44
Table 2-4	Machine Learning classification accuracy on PCA data . . .	45
Table 2-5	Machine Learning precision and accuracy for different network motifs	46
Table 2-6	Characteristics of <i>in-silico</i> input data	48
Table II-1	Median AUPR values for different combinations of MI estimator and GRN inference algorithm for different network sizes.	63
Table II-2	Median AUPR values for different combinations of MI estimator and GRN inference algorithm for different organisms. . .	64
Table II-3	Characteristics of the 10 synthetic networks from DREAM3 and statistics of the different 3-node network motifs extracted.	66

List of Figures

Figure 1-1	Illustration of two methods to evaluate distribution	3
Figure 1-2	The different steps for evaluating GRN inference performance.	7
Figure 1-3	A schematic GRN inference example.	10
Figure 1-4	Percent error of different mutual information estimators for multivariate gaussian distribution.	13
Figure 1-5	AUPR values for different combinations of MI estimator (ML or KSG) and GRN inference algorithm (RL, CLR or CMIA).	15
Figure 1-6	AUPR difference of combinations of MI estimators and infer- ence algorithms relative to the gold standard [ML,CLR]. . .	15
Figure 1-7	Common 3-node network motifs	17
Figure 1-8	Performance comparison of GRN reconstruction for different <i>in-silico</i> networks modeled from <i>E. coli</i> & Yeast.	17
Figure 1-9	Computation time vs. different data sizes for a network of 50 genes.	19
Figure 1-10	Area Under Precision-Recall curve (AUPR) vs. different num- ber of bins or k-neighbors.	23
Figure 2-1	Illustration of common network motifs and topology used in GNW	26
Figure 2-2	Mutual Information profiles for Two-genes motifs	32
Figure 2-3	Mutual Information profiles for two-edge motifs	33

Figure 2-4	Mutual Information profiles of Cascade vs FFL motifs	35
Figure 2-5	Mutual Information profiles of FFL coherent vs incoherent motifs	36
Figure 2-6	Mutual Information profiles of Loop and Motif13	36
Figure 2-7	Mutual Information profiles of DREAM3 motifs	38
Figure 2-8	Mutual Information profiles of DREAM4 motifs	39
Figure 2-9	Principal Component Analysis and K-means clustering for Fan-in/out motifs	40
Figure 2-10	Principal Component Analysis and K-means clustering for Cascade motifs	41
Figure 2-11	Principal Component Analysis and K-means clustering for 16 motifs	42
Figure 2-12	Mutual Information profiles of <i>E. coli</i> motifs	46
Figure II-1	Two-way mutual information (MI2) & total correlation (TC) for multivariate gaussian dist. with varying bins and neighbors	67
Figure II-2	Boxplots of percent error for Interaction Information (II) of three different mutual information estimators	68
Figure II-3	Boxplots of percent error for Conditional Mutual Information (CMI) of three different mutual information estimators	69
Figure II-4	Boxplots of percent error for Three-way Mutual Information (MI3) of three different mutual information estimators	70
Figure II-5	Boxplots of percent error of Two-way Mutual Information calculated based on kNN methods.	71
Figure II-6	Boxplots of percent error of Total Correlation calculated based on kNN methods.	72
Figure II-7	AUPR difference relative to the gold standard combination [ML,CLR] for different Yeast networks from DREAM3 . . .	73

Figure II-8	AUPR difference relative to the gold standard combination [ML,CLR] for different networks of 100 genes from DREAM4	74
Figure III-1	Z-score profiles for Two-genes motifs	76
Figure III-2	MI profiles for two edge motifs with different repressing and inducing interactions	76
Figure III-3	Z-score profiles for all simulated three-node motifs	77

Chapter 1

Gene regulation network inference using k-nearest neighbor-based mutual information estimation

Background

Most cells in a multicellular organism contain the same genome, yet they can differentiate into different cell types and adapt to different environmental conditions [1]. These responses to internal and external cues are possible due to the presence of an elaborate gene regulatory network (GRN). A GRN is the genome’s “flowchart“ for various biological processes such as sensing, development, and metabolism, enabling the cell to follow specific instructions upon an internal or external stimulation. Understanding how genomic flowcharts are organized brings the potential to remediate dysfunctional ones [2] and design new ones for synthetic biology [3].

Advances in large-scale gene expression data collected from omic-level microarrays and RNA-seq experiments allow the construction of basic networks by clustering co-expressed genes using statistical correlation metrics such as covariance and threshold to determine the statistical significance [4]. Another common practice is to monitor the expression of multiple genes in response to perturbations and then infer the relationship between these genes [5]. Currently, there are several classes of methods

to infer GRNs from expression data, such as the Bayesian networks method, the statistical/information theory method, and ordinary differential equations (ODEs) (see excellent reviews [6–8]).

Originally introduced for communication systems by Shannon in the late 40s [9, 10], mutual information (MI) was quickly adopted by other disciplines as a statistical tool to evaluate the dependence between variables. Unlike the abovementioned traditional correlation methods like covariance, MI can detect linear and non-linear relationship between variables and can be applied to test the dependence between any number of variables (n-dimensions).

Over the last twenty years, researchers have implemented many methods employing MI to reconstruct GRNs, such as Relevance Networks [11]; ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks, [12]); and CLR (Context Likelihood of Relatedness, [13]). Using MI with two variables (i.e. genes) is straightforward, but due to the positive and symmetric nature of two-way MI [14], MI with only two variables cannot distinguish between direct and indirect regulation, coregulation, or logical gate-type interactions [15, 16]. To overcome these issues, a few groups have used different three-dimensional MI measures in inference algorithms [15, 17, 18] (for a comprehensive list of methods, see Mousavian et al. [19]). Importantly, in most methods using MI, continuous input (i.e., normalized fluorescence intensity data for gene expression) needs to be discretized first to build probability density functions (PDF). This practice is known to be sensitive to data size, correlation strength and underlying distributions [20].

In general, the simplest and most computationally inexpensive method to discretize continuous data is fixed (width) binning (FB) (Fig. 1-1A), where a histogram with a fixed number of bins (or bin width) determined by certain statistical rules is used to model the PDF. For finite data size, FB generally under- or over-estimates MI (Fig. S1A). Over the years, researchers developed different methods to mitigate bin

number sensitivity and to better estimate (or correct the bias in) MI, especially for data of small sizes. These methods correct either the entropies (Miller-Madow [21]) or the probability distribution by adaptive partitioning (AP) [22], k-Nearest Neighbor (kNN) [23] (Fig. 1-1B), kernel density estimator (KDE) [15] and/or B-spline functions, in which data points are divided into fractions between a predefined number of adjacent bins [24]. Unfortunately, all these methods make assumptions on the density distribution and require adjustment of parameters by the user for different scenarios except for kNN, which is shown to be accurate and robust across different values of k [20, 23]. However, kNN is rarely used due to the higher computational costs it entailed [25] or the limited improvement for two variables (2d) in downstream analysis.

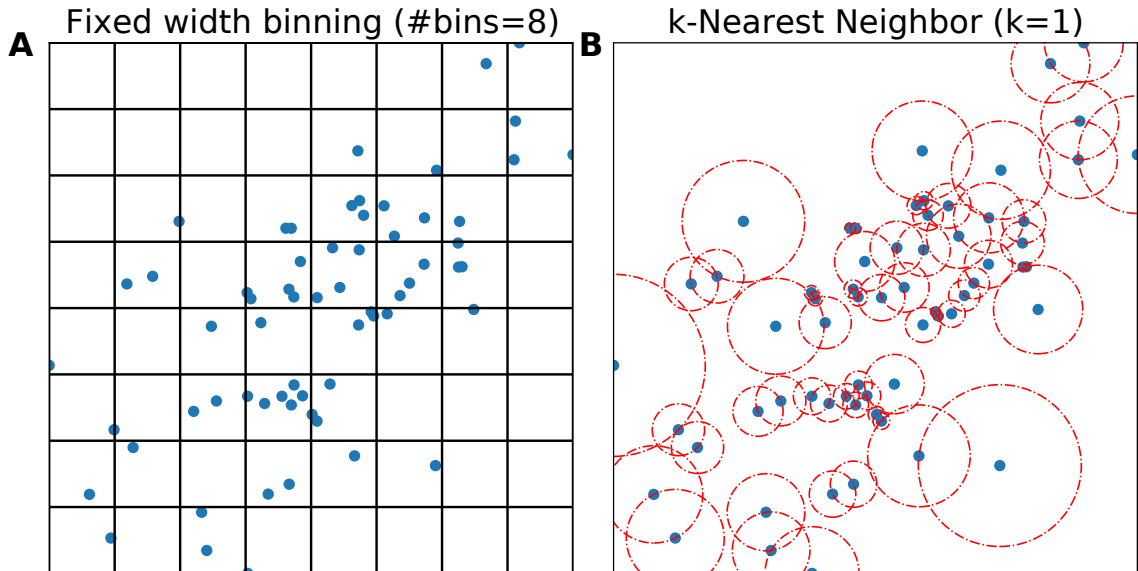


Figure 1-1. Illustration of two methods to evaluate distribution (A) Fixed width binning, and (B) k-Nearest-Neighbor ($k=1$). Data points are shown as blue circles, bin edges are shown in black, and distances to $k=1$ neighbor as the radius of dashed red circles.

The problem of accurately estimating the correlation between genes has only worsened in this new era of single cell transcriptome studies, as data is larger yet sparser, often with non-Gaussian distributions. In this work, we focus on two subjects:

(a) Improving MI estimation – we present an implementation of a three-way MI estimator based on kNN, which addresses large errors in estimating MI measures for three variables (3d). (b) Improving GRN inference – we present CMIA (Conditional Mutual Information Augmentation), a novel inference algorithm inspired by Synergy-Augmented CLR (SA-CLR) [18]. By testing various mutual information estimators against the ground truth solved from an analytical solution and comparing their performance using *in-silico* GRN benchmarking data, we find that kNN-based three-way MI estimator Kraskov-Stoögbauer-Grassberger (KSG) improves the performance of common GRN inference methods. Together with the inference algorithm, CMIA, it outperforms other commonly used GRN reconstruction methods in the field.


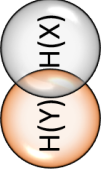
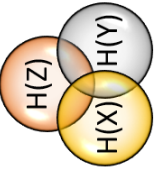





Materials and Methods

Calculate mutual information of multiple variables

In Table 1-1, we summarize the formalism for calculating MI (for detailed overview please see Appendix I). Shannon’s entropy is the basic building block of MI and represents the randomness of a variable: the more random it is, the more uniformly it is distributed, which gives a higher entropy. For our purposes, X , Y , or Z is a vector (x_1, x_2, \dots, x_n) , (y_1, y_2, \dots, y_n) or (z_1, z_2, \dots, z_n) representing a specific gene’s expression profile (data x , y or z) under different conditions/perturbations (n steady-states) or as a function of time (n time points). Two-way MI is defined as the shared (or redundant) information between the two variables X and Y (Table 1-1) and can be visualized by a Venn diagram (Table 1-1 right column).

While MI for two variables (genes or dimensions) is readily understood, for three variables or more, new measures arise including Total correlation (TC), Three-way MI (MI3), Interaction Information (II) and Conditional MI (CMI) (Table 1-1). Unfortunately, the term ‘three-way MI’ has been used loosely in the literature to refer to

Table 1-1. Mutual Information formalism

Term	Symbol	Formula	Venn diagram
Shannon's entropy of X	$H(X)$	$-\sum_x p(x) \log p(x)$	
Joint entropy of X & Y	$H(X, Y)$	$-\sum_x \sum_y p(x, y) \log p(x, y)$	
Joint entropy of X, Y & Z	$H(X, Y, Z)$	$-\sum_x \sum_y \sum_z p(x, y, z) \log p(x, y, z)$	
Two-way Mutual Information	$MI(X; Y)$	$H(X) + H(Y) - H(X, Y)$	
Total Correlation	$TC(X, Y, Z)$	$H(X) + H(Y) + H(Z) - H(X, Y, Z)$	
Three-way MI	$MI3((X, Y); Z)$	$TC - MI(X; Y)$	
Interaction Information	$II(X, Y, Z)$	$TC - MI(X; Y) - MI(X; Z) - MI(Y; Z)$	
Conditional MI	$CMI(X; Y Z)$	$TC - MI(X; Z) - MI(Y; Z)$	

all four of these measures, and because they represent distinct aspects of statistical dependence, in the context of GRN reconstruction, this can lead to different realizations. Unlike other MI quantities, Interaction-Information is hard to visualize using a Venn diagram, as it can have both positive and negative values. It is common to regard negative II as “Redundancy”, the shared information between all variables, and positive II as “Synergy”. Synergy can be interpreted as new information gained on the dependence between two variables $\{X,Y\}$ when considering the contribution of a third variable $\{Z\}$ on either $\{X\}$ or $\{Y\}$ v.s. without considering it, or mathematically: $II = CMI(X;Y|Z) - MI(X;Y)$.

To calculate the marginal and joint entropies of two variables (X and Y), we first need to know the probability of each data point. For discrete data, we can approach the underlying probability $p(x)$ by calculating the frequency ($f_x = N_x/N_{Tot}$) where N_x is the number of data points with value x , and N_{Tot} is the total sample size. For the continuous data case, the calculation is more complex. Although Shannon extended his theory for continuous data by replacing the summation with integrals [10], it is common practice in the field to discretize the data first so one can work with the discrete formalism (Table 1-1). The simplest discretization method is to use fixed (width) binning (FB) (Fig. 1-1A), but the optimal binning choice depends on the shape of the distribution and data size. For normally distributed data, the rule of thumb is to use the square-root of the data size as the number of bins.

k-nearest-neighbor (kNN)

Other than evaluating the probability densities to calculate mutual information, Kozachenko and Leonenko (KL) calculated the marginal and joint entropies (and the MI by summation) from the mean distance to the kth-nearest neighbor [26]. To minimize errors when combining entropies of different dimensions, Kraskov et al. calculate the MI directly [23]. KSG developed two algorithms, $I^{(1)}$ and $I^{(2)}$ (hereafter, KSG-1 and

KSG-2), to minimize errors when estimating MI compared to previous methods. We chose KSG-1 (defined below as MI_{KSG}) as it gives slightly smaller statistical error (dispersion). Note that although KSG-1 gives relatively larger systematic errors than KSG-2, these systematic errors do not change the ranking of the output values (from high to low), which is what we use in downstream analysis. An additional note is that using kNN can lead to negative values for mutual information, which contradicts Shannon’s theorem. Negative values are caused by statistic fluctuations when there is no correlation between variables. Therefore, in such a situation, we set negative values to zero (except for Interaction Information, where it is meaningful). To calculate MI using the KSG method, we use the following formulas:

$$MI_{KSG}(X; Y) = \Psi(k) - \langle \Psi(n_x + 1) + \Psi(n_y + 1) \rangle + \Psi(N) \tag{1.1}$$

$$TC_{KSG}(X; Y; Z) = \Psi(k) + 2 \cdot \Psi(N) - \langle \Psi(n_x) + \Psi(n_y) + \Psi(n_z) \rangle \tag{1.2}$$

Where $\Psi(x)$ is the digamma function, N is the number of data points, n_i is the number of points x_j whose distance from x_i is less than $\epsilon(i)/2$, and $\epsilon(i)/2$ is the distance from $u_i = (x_i, y_i, z_i)$ to its k th neighbor, as illustrated in Fig. 1(a) of [23].

***in-silico* GRN Inference comparison:**

MI calculations are used to infer interactions between genes to reconstruct the underlying GRN structure. To test the performance of different methods, we followed the methodology of the *in-silico* network inference challenges of the Dialogue for Reverse Engineering Assessments and Methods (DREAM) competitions DREAM3-4 [27] as depicted in Fig. 1-2.

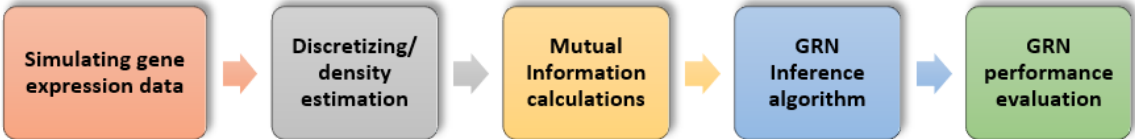


Figure 1-2. The different steps for evaluating GRN inference performance.

Simulating gene expression data

We used GeneNetWeaver [28] to generate steady-state and time-series gene expression datasets for realistic *in-silico* networks of sizes of 50, and 100 genes containing various experimental conditions (knockouts, knockdowns, multifactorial perturbation, etc.). GeneNetWeaver uses a thermodynamic model to quantify mRNA transcription and regulation with added molecular and experimental noise.

Discretizing/density estimation

To handle the continuous expression data, we chose either:

- (a) Density estimation by fixed bin. We used the common practice \sqrt{n} , where n = number of data points (in our case, different experimental conditions), as the number of bins.
- (b) Density estimation by k-Nearest Neighbor (kNN). Unless stated otherwise, we chose $k=3$ as a good compromise between precision and computation cost as discussed in [23].

Mutual Information estimation

Depending on our previous selection, we chose between several MI estimators:

- (a) For the fixed-bin discretizing method, we used either Shannon’s formula (also referred to as Maximum Likelihood, ML) or Miller-Madow (MM) estimator (Appendix B.).
- (b) For kNN we used either KL or KSG formulas for MI.

GRN inference algorithms

We used popular algorithms in the field that use either only two-way MI or both two- and three-way MI to infer undirected network structure by sorting predicted

interacting gene pairs from most probable to least probable. Each algorithm starts with a MI matrix containing calculation for all possible pairs (some use all possible triplets) and applies different rules to filter results and sort the gene pairs (see summary below). We used the same MI matrices for a fair comparison between the inference algorithms. The following algorithms were used in our comparison:

- (a) Relevance Network (RL) – Gene pairs are sorted according to their $MI(X;Y)$ value from highest to lowest, and a threshold applied to truncate non-significant results [11]. We didn't set a threshold to maximize AUPR (see below).
- (b) Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) – Same as RL with the addition of Data Processing Inequality (DPI), which means for every three genes MI is calculated for each pair and the pair with the lowest MI is removed if the difference is larger than some threshold [12]. In our implementation, we set the threshold to zero, so we always removed the lowest interacting pair (same implementation as *Minet* [29]). On the other extreme, where we kept all the pairs, ARACNE is the same as RL.
- (c) Context Likelihood of Relatedness (CLR) – Background correction is performed by calculating Z-score for the MI of each gene interacting with all other genes, and then gene pairs are sorted by their mutual Z-score [13]. We didn't use B-spline smoothing in the density estimation step in accordance with the implementation in the R-package *Minet* [29].
- (d) Synergy Augmented CLR (SA-CLR) – Same as CLR, with the difference that now the highest Interaction-Information term is added to MI prior to performing the background correction [18].
- (e) Conditional Mutual Information Augmentation (CMIA) – Similar to SA-CLR but we used conditional mutual information instead of interaction-information.

- (f) Luo et al. MI3 (hereafter CMI2rt) – Two regulators are assumed for each target gene, and for each target gene we searched for the best R1,R2 pair that maximizes: $CMI(T;R1|R2)+CMI(T;R2|R1)$ [15].

GRN performance evaluation

To evaluate the performance of common algorithms in the field, we used known (true) synthetic networks and counted the number of true and false positives (TP and FP respectively) predictions as well as true and false negative (TN and FN respectively) (Fig. 1-3). This allowed us to plot precision ($Precision = TP/(TP + FP)$) v. s. recall ($Recall = TP/(TP + FN)$) and calculate the area under precision-recall curve (AUPR). As biological networks are sparse on edges (interactions), AUPR is considered a better metric than AUROC (area under the receiver operating characteristic curve, which is the false positive rate $FPR = FP/(FP + TN)$ v.s. recall) as mentioned elsewhere [30].

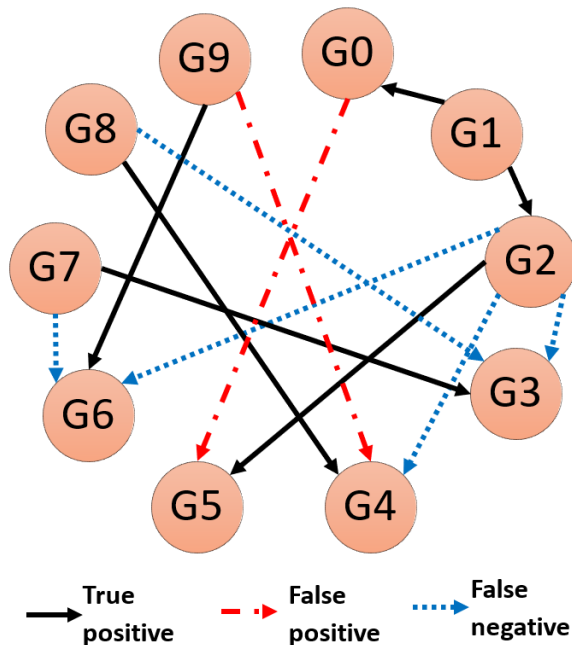


Figure 1-3. A schematic GRN inference example.

The true network contains 10 genes (a.k.a. nodes), and 11 interactions (or edges). The prediction algorithm correctly predicted 6 times (True positive), missed 5 interactions (False negative), and predicted 2 interactions that did not exist (False positive).

Results

Benchmark MI estimations of a Gaussian distribution

To evaluate the performance of different mutual information (MI) estimators on continuous data, we calculated their deviations from the true underlying value by defining a percent error:

$$percent_error = \frac{|Analytical_MI - Estimated_MI|}{Analytical_MI} \times 100\% \quad (1.3)$$

In most biologically relevant cases, one does not know what the true MI value is, because one does not know the probability distributions of the variables we are concerned with. Nevertheless, the true underlying value of MI of a few distributions such as Gaussian distribution can be analytically calculated. Therefore, to allow quantitative comparisons between different MI estimators, we used the analytical solution of Shannon’s entropy for a Gaussian distribution (see Appendix A.) to calculate the MI by entropy summation (Table 1-1). We then compared all methods of different data sizes (100, 1K, 10K, referring to the number of different conditions/perturbations/time points of individual genes) and different correlation strengths (0.3, 0.6, 0.9) between two or three variables (number of genes, 2d or 3d) drawn from a Gaussian distribution with a mean at zero and a variance of one (the absolute values of mean and variance are not important in the calculation as the final solution only contains correlation, see Appendix A.). For two-way MI (two variables, or 2d) (Fig. 1-4A), we compared the following MI estimators: (i) Maximum Likelihood (ML, given by Shannon, Table 1-1), (ii) Miller-Madow correction (MM, see Appendix B.), (iii) Kozachenko-Leonenko (KL) [26], and (iv) KSG. The first two methods use FB to discretize the continuous data, and in general the best number of bins changes depending on the data size and correlation between variables (Appendix Fig. II-1A). As *a priori* the correlation strength is unknown, for the number of bins we used the common practice \sqrt{N} , where N equals the number of data points, and the result was rounded down to align with

methods in the next section. The latter two methods both use kNN, and we found that any selection of k resulted in good alignment with the analytical solution (see Appendix Fig. II-1B). We chose the third nearest-neighbor ($k=3$) as recommended by Kraskov et al [23] because a k value of 3 resulted in a good trade-off between precision and computational cost. As shown in Fig. 1-4, in all cases the two kNN-based MI estimators performed well similarly and outperformed the fixed-binning methods judged by the percentage error.

While two-way MI estimators were studied extensively [23, 31], to our knowledge, no benchmark was done on MI with three or more variables. We repeated the same methodology described above but this time for the 3d Total Correlation (TC) (Fig. 1-4B, Appendix Fig. II-1C-D). Similar to the 2d case, kNN-based MI estimators KL3 and KSG3 outperformed the other methods. We also examined the other three-way MI quantities, three-way MI (MI3), Interaction Information (II), Conditional Mutual Information (CMI) (see Appendix Fig. II-2,II-3,II-4) and obtained similar results. We also explored whether a higher kNN value, for example $k=10$, further improved accuracy. We found that a higher k value ($k=10$) does not improve the accuracy dramatically compared to that in $k=3$ (Appendix Fig. II-5,II-6), but it did reduce the variance for small correlations ($r=0.3$).

***in-silico* GRN Inference performance enhancement**

Next, we aim to investigate whether the high precision of MI estimation based on kNN for bi- and tri-variate Gaussian distributions also translates to a high performance in inferring GRN structure compared to other MI estimation methods described above. To compare the performance of different MI estimators and inference algorithms, we used a total of 15 different synthetic networks: ten synthetic networks from the DREAM3 (Dialogue for Reverse Engineering Assessments and Methods) competition [27] with 50 and 100 genes, respectively, and five networks from DREAM4 with 100

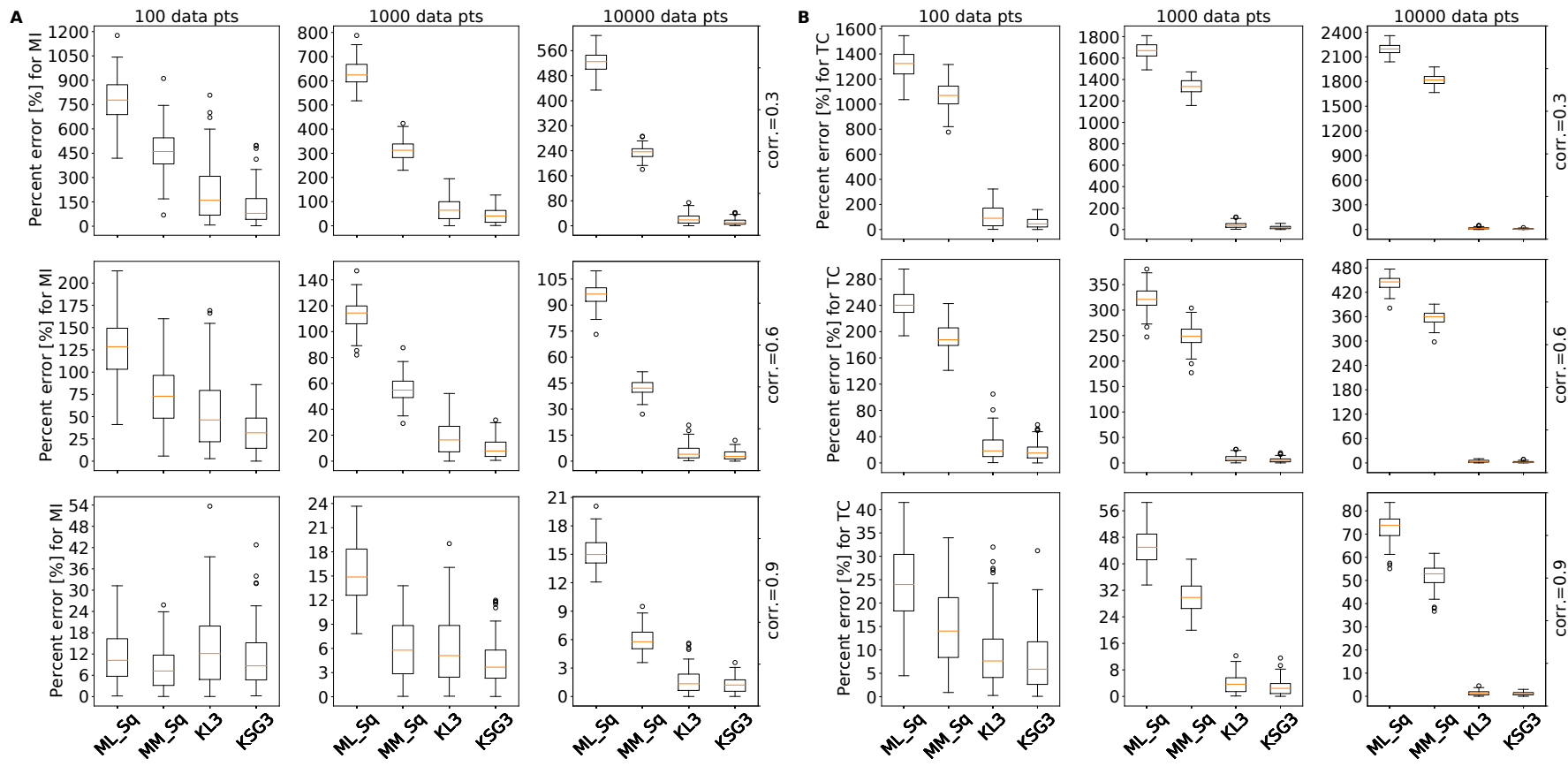


Figure 1-4. Percent error of different mutual information estimators for multivariate gaussian distribution.

Each boxplot represents 100 replicates, with columns representing sample size = {100,1K,10K}, and rows the correlation = {0.3,0.6,0.9}. (A) Percent error (y-axis) for two-way mutual information (MI2) was compared for 3 different methods: ML_Sq=Maximum Likelihood (Shannon’s MI) with fixed width binning (number of bins is determined by square-root), MM_Sq=Miller-Madow formula for MI with square-root for the number of bins, KSG3 =KSG formula for kNN-MI with k=3; (B) same methods compared for total correlation (TC).

genes. The networks were extracted from documented regulation databases of *E. coli* and *S. cerevisiae* [32]. We used the software GeneNetWeaver 3.1.2b [28] with default settings to generate simulated expression data for each network and performed ten replicates to include the variance in expression data due to experimental and stochastic molecular noise. Furthermore, to comply with the majority of available experimental data, we only used the simulated steady state data (Wild type, knockouts, dual-knockouts, knockdowns, multifactorial perturbation) accumulating to 170, 169 and 201 conditions in the 50 gene synthetic networks for *E. coli* 1, *E. coli* 2 and Yeast1/2/3 respectively, 341, 322 and 401 conditions in the DREAM3 100 gene synthetic networks for *E. coli* 1, *E. coli* 2 and Yeast1/2/3 respectively, and 393, 401 conditions in the DREAM4 100 gene networks. We then ran the expression data through our custom Python 3.8 code pipeline to calculate the area under precision-recall curve (AUPR) for each replicate. In Fig. 1-5 we show sorted boxplots of the AUPR values (y-axis) comparing six combinations of three inference algorithms (Relevance Networks, RL; Context-Likelihood-Relatedness, CLR; and our Conditional-Mutual-Information-Augmentation, CMIA) and two MI estimators (ML, fixed bin-based; KSG, kNN-based), for five networks with 50 genes (Fig. 1-5A), five networks of 100 genes from DREAM3 (Fig. 1-5B), and five networks of 100 genes from DREAM4 (Fig. 1-5C). In all cases, the kNN-based KSG as the MI estimator improves the performance of the inference algorithms. The improvement is more significant for CMIA, which uses three-way MI calculations, and corroborate the higher percent error we found when estimating TC (Fig. 1-4B).

***in-silico* GRN Inference performance comparison**

To verify whether the performance enhancement introduced by kNN-based MI estimators is general for other GRN inference algorithms, we further extended our benchmark to twenty-four different combinations of the four MI estimators (discrete bin-based ML

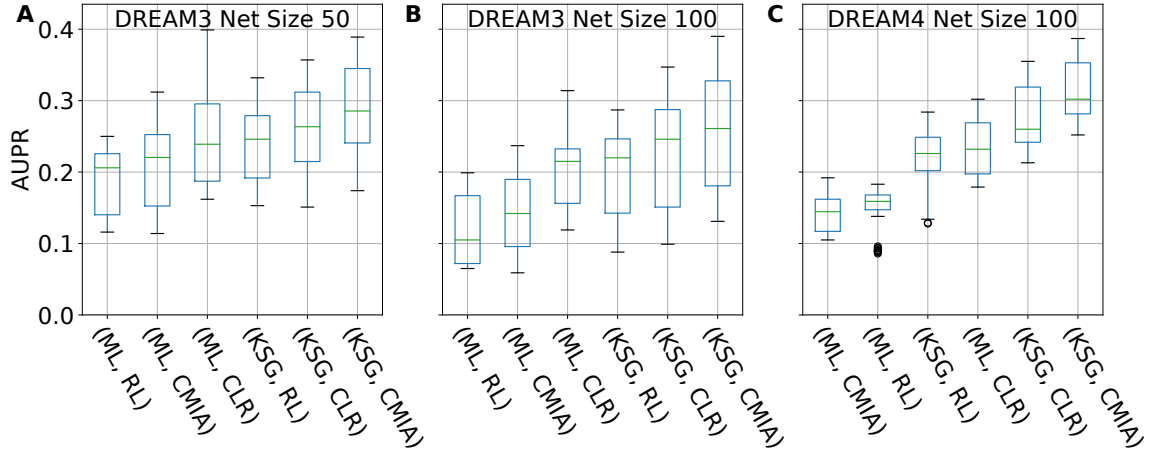


Figure 1-5. AUPR values for different combinations of MI estimator (ML or KSG) and GRN inference algorithm (RL, CLR or CMIA).

(A): Sorted boxplots showing networks of size 50 from DREAM3, (B): Networks of size 100 from DREAM3, (C): Networks of size 100 from DREAM4. For the different network sizes each boxplot represents 50 networks (5 different networks X 10 replicates).

and MM, and kNN-based KL, and KSG) with six inference algorithms described in the Methods section (RL, CLR, ARACNE, SA-CLR, CMIA, CMI2rt) and compared them to the field gold standard combination {ML, CLR} (Fig. 1-6).

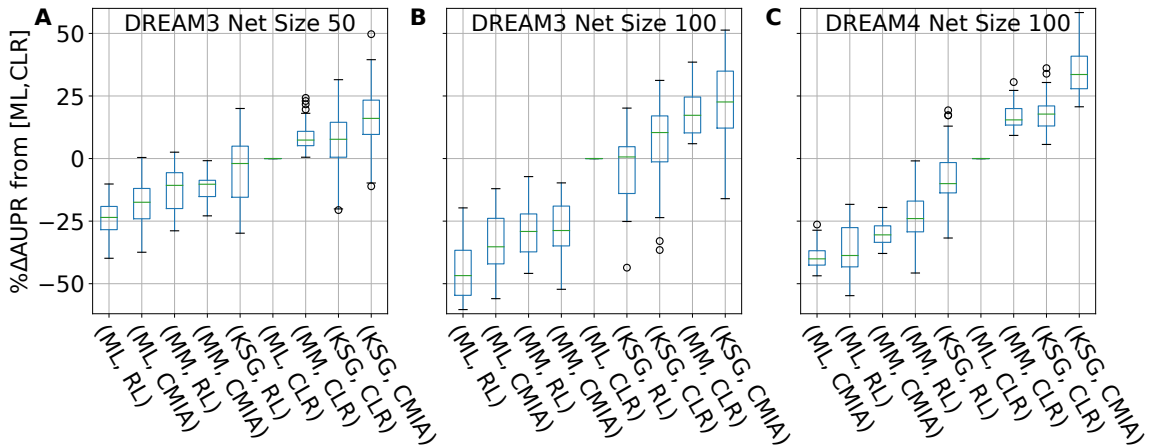


Figure 1-6. AUPR difference of combinations of MI estimators and inference algorithms relative to the gold standard [ML, CLR].

(A): Sorted boxplots showing comparison for Network size of 50 from DREAM3, (B): and size of 100 from DREAM3, (C): size of 100 from DREAM4. Each boxplot represents 50 networks (5 different networks X 10 replicates). A complete list of tested GRN inference algo & MI estimators can be found in Appendix Table II-1

To compare the performance differences quantitatively, we calculated the change in AUPR for each replicate relative to the field’s gold standard combination of CLR inference algorithm with ML for MI calculations. In Fig. 1-6 we show the top nine combinations, omitting ARACNE and CMI2rt among the inference algorithms, and KL from the MI estimators because of their poor performance. We also omitted SA-CLR due to its similarity to CLR and CMIA (see full data in Appendix Table II-1). The combination of {KSG,CMIA} gave the best median score in the combined networks inspected under each category. It showed a median improvement of 16% and 24% for networks of 50 and 100 genes from DREAM3, respectively (Fig. 1-6A, B), and 34% improvements for networks of 100 genes from DREAM4 (Fig. 1-6C). Furthermore, replacing the MI estimator from ML to KSG in the case of the gold standard {ML,CLR} can lead to significant improvement in GRN reconstruction performance, with median increase in AUPR of 8-18%.

***in-silico* GRN Inference performance of different organisms**

Next, we examined the performance of these different algorithms with regards to different biological organisms, as *E. coli* and *S. cerevisiae* have distinct distributions of different network motifs (Fig. 1-7), which may lead to different performance in network inference. For example, the fan-out motif, where one gene regulates two (or more) target genes, is more abundant in *E. coli*, while the cascade motif, where a gene regulates a second gene that in turn regulates a third gene, is more abundant in *S. cerevisiae* [7, 33]. In both cases, the three participating genes exhibit some degree of correlation, yet not all are directly connected. The 10 networks from DREAM3 were divided into four *E. coli* networks (Fig. 1-8A, C-F) and six *S. cerevisiae* networks (Fig. 1-8B, Appendix Fig. II-7).

For the combined *E. coli* networks (Fig. 1-8A), KSG greatly improved the performance of both RL and CMIA algorithms but showed only a modest 6% improvement

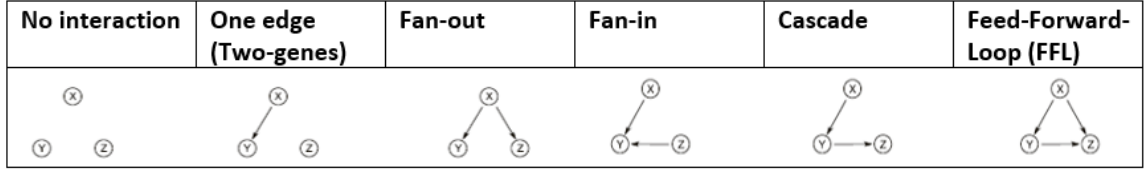


Figure 1-7. Common 3-node network motifs

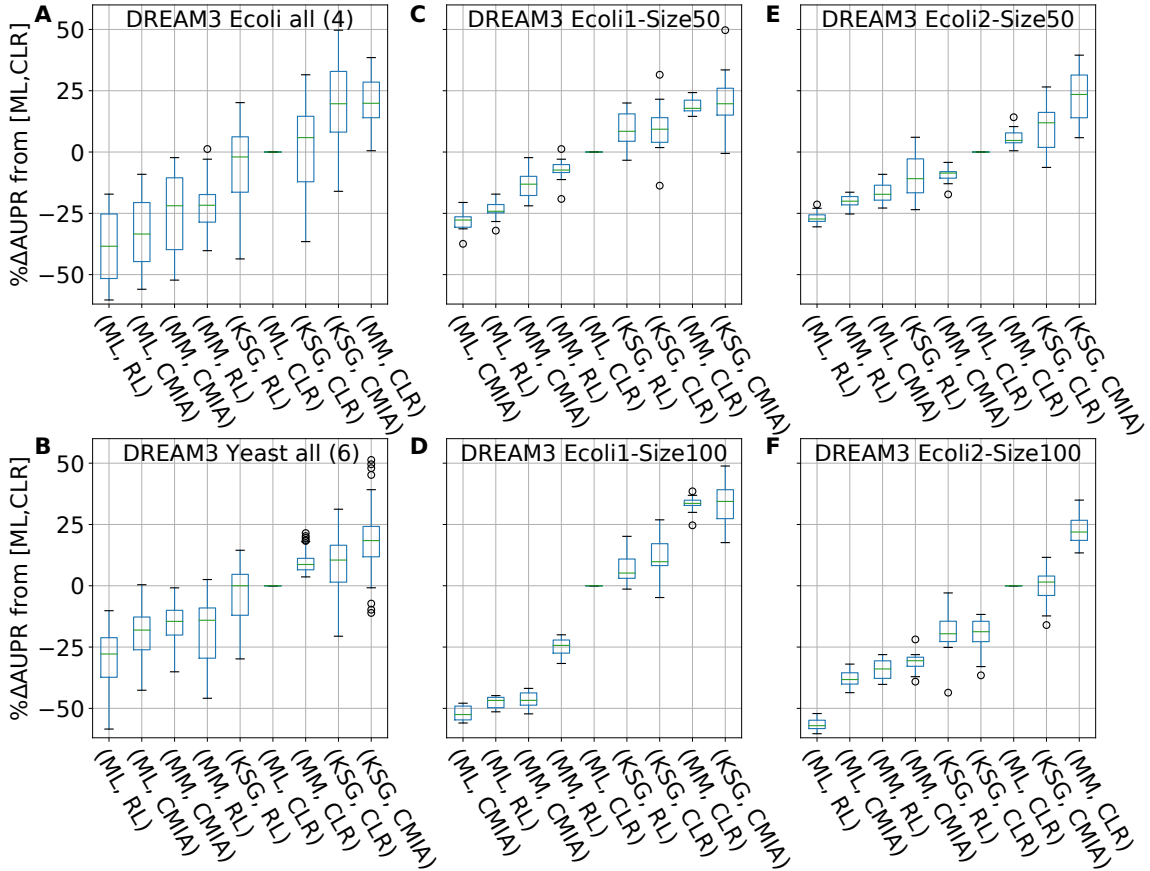


Figure 1-8. Performance comparison of GRN reconstruction for different *in-silico* networks modeled from *E. coli* & Yeast.

x-axis shows different combinations of [MI estimator, inference algo], y-axis shows percentage AUPR difference (increase or decrease) relative to the gold standard combination [ML, CLR]. (A): Sorted boxplots of the combined four *E. coli* networks from DREAM3. Each boxplot represents 40 networks (4 different networks X 10 replicates). (B) same as (A) but for the six Yeast networks. (C)-(F): Sorted boxplots of the 4 different *E. coli* networks from DREAM3. Each boxplot represents 10 replicates. A complete list of tested MI estimators & GRN inference algo can be found in Appendix Table II-2

in performance for CLR. For the combined *E. coli* networks, {KSG, CMA} achieved a median improvement of 20%, but was second best to {MM, CLR}, with a small

0.2% difference. The performance comparison of the individual *E. coli* networks (Fig. 1-8C-F) showed that {KSG,CMIA} was the best performer on three out of four networks. Furthermore, replacing ML with KSG when combined with CLR improved the performance by 10-15% except in the case of DREAM3 Ecoli2-Size100 (Fig. 1-8F). In the *S. cerevisiae* networks, again KSG improved all algorithms, and most significantly CMIA, and showed a median improvement of 18%. Several replicates did not show any performance improvement, indicating the significance of stochasticity even though all kinetic parameters for each network were identical. In summary, out of 24 combinations of MI estimators and inference algorithms, the combination {KSG,CMIA} yielded the best median score in 13 out of the 15 networks inspected (except networks DREAM3 Yeast1-Size50 & Ecoli2-Size100, Fig. 1-8C-F, Appendix Fig. II-7 and II-8). Therefore, we conclude that using kNN-based KSG to calculate MI improved the performances of the inference algorithms evaluated in most cases.

Computational cost

Computational cost is a major concern when applying kNN-based methods. We measured the time required to calculate all the two- and three-way interactions in a 50 gene network (1125 pairs and 19600 triplets, respectively, after taking symmetry into account) with different data size [100, 250, 500, 1000] for three MI estimation methods: FB-ML, kNN-KL and kNN-KSG. The code for the three estimators was written in Python 3.8, used built-in functions from Numpy v1.19 and Scipy v1.5, and was run on a single core of a desktop [Intel Xeon E5-1620 @ 3.6 GHz]. As seen in Fig. 1-9 FB-ML was the fastest, as histogram-type calculations have been optimized in Python over the years.

FB-ML was also insensitive to data size (in the tested range). While the python-based KSG implementation was most computationally heavy, the time was tractable (under 400 s even for the largest data size (1000) and 3d calculation). The speed

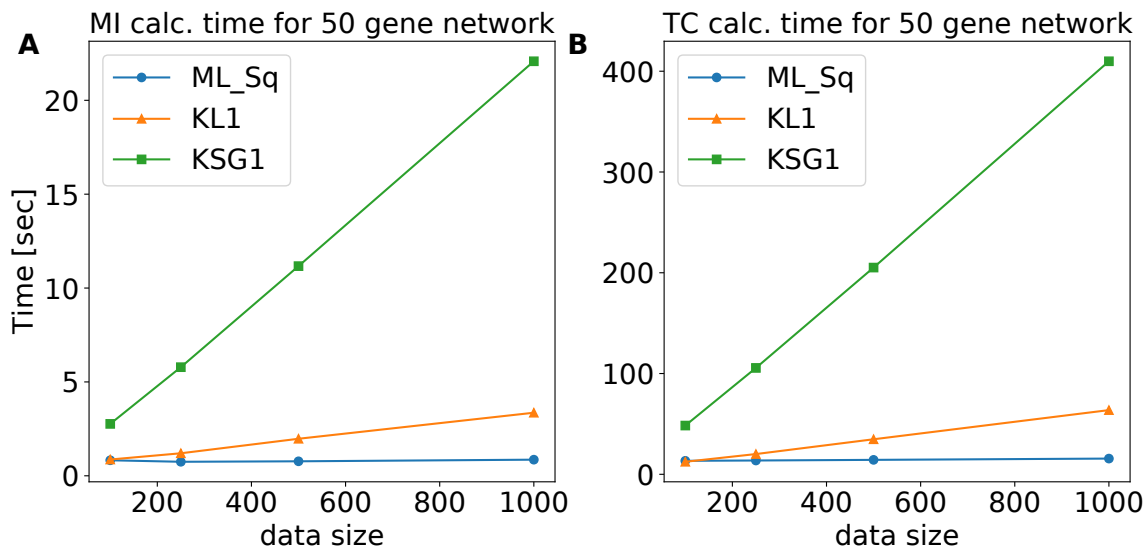


Figure 1-9. Computation time vs. different data sizes for a network of 50 genes. (A) The calculation is performed over 1125 pairs for data sizes of [100, 250, 500, 1000]. (B) The calculation is performed over 19600 triplets for data sizes as in the left panel

could be further boosted by rewriting the code in C/C++, similar to what was done by Meyer et al. [29] and Sales et al. [25]. Furthermore, the KD-Tree class of algorithms [34], which was in the main core of this work’s implementation, could greatly benefit from multiple cores or parallel processing. After building the initial tree, distance calculation between neighbors can proceed in parallel, offering 4-to-16 fold improvement in speed on a current personal computer, depending on the number of available cores.

Discussion

To date, a plethora of discretization methods, MI estimators, and inference algorithms exist in the literature to reconstruct GRNs. Some common methods are available in the R/Bioconductor package *Minet* [29] and in Julia language [35]. In fact, as different methods have certain advantages depending on the investigated scenario and constraints, it is advantageous to consider and compare the performance of different combinations of multiple methods [36].

kNN-based MI estimator for data discretization/density estimation outperforms fixed-bin-based estimations

Here, we demonstrate that the MI estimator KSG based on kNN yields smaller errors compared to other MI estimation methods using discretized fixed bins in the case of a bi- and tri-variate Gaussian distribution. KSG proves to be robust against different data sizes and correlations as well as the k parameter used, unlike FB methods where the parameter used (number of bins) has a large effect on accuracy of the MI estimator. In principle, one can achieve smaller errors using MI based on discretized bins by choosing a different bin number other than the rule of thumb \sqrt{N} , for correlations smaller than 0.9. However, *a priori* one does not know the correlation strength. In fact, estimating the correlation strength is what one tries to achieve when using MI. We also note that the gene expression profiles of different synthetic networks and real experimental systems could be better described by distributions other than Gaussian. Fortunately, the analytical solution to the mutual information of a few of these distributions can be calculated [37] and will be explored in future work.

Note that in this work we did not compare the performance of another frequently used binning method, adaptive partitioning, which is computationally faster than kNN for large data sets. In brief, adaptive partitioning is a general term referring to three methods that divide the data uniformly between the bins. The first method is equal frequency in which the bin size varies to allow for equal number of data points in each bin. The second method is equiprobable partitioning [22], in which data is ranked and partitioned in the middle, and Pearson chi-square test is used to determine the number of sub-partitions, where the significance level of the chi-square test can be tuned (1%-5%) according to the size of the data. This method works well for 1d data, but it has some ambiguity when implemented in higher dimensions in that data points must be ranked according to one of the axes (or more in $>2d$), and there are no appropriate rules to rank multidimensional data points. The third method is Bayesian

blocks [38], which uses a Bayesian statistics approach to attempt to find the optimal number of bins and their sizes by maximizing a fitness function that depends on those two parameters. While this is a seemingly promising approach, it is unclear how to implement such a method beyond 1D. Because of these reasons, we did not include this binning method in the comparison.

Another previously used method in the literature is KDE [15], but it is the most computationally costly and requires large data sets. It approximates the data distribution using a predefined known distribution (i.e., a Gaussian) with user-defined smoothing parameters. This practice can be problematic because in most cases the underlying data distribution is unknown, and experimental data is much sparser than required to achieve results similar to other, simpler methods, such as FB.

kNN-based MI estimator KSG in combination with CMIA achieves the highest accuracy but may be subject to data stochasticity

It is clear from Fig. 1-5 and 1-6 that the combination of kSG-based MI estimation and inference algorithm CMIA achieved the highest precision and recall when reconstructing an unknown network. Yet, this combination also showed a large variation in the performance enhancement. As shown in Fig. 1-6, 1-8A-B, we observed that when KSG was combined with CLR or CMIA, a few replicates did not show any performance improvement, or even had a decreased performance indicated by the negative $\% \Delta AUPR$ value, as indicated by the outliers and bottom whisker of the boxplot.

To investigate the source of this variation in the ensemble network plots we inspected different combinations of MI estimators, inference algorithms, data size used, and individual networks (Fig. 1-8C-F, Appendix Fig. II-7). We found that higher k values (up to $k = 15$) did not affect the variability in the AUPR results (Fig. 1-10).

However, MI calculation done by KSG exhibited large variations in performance when smaller data size was used as that in the case of 50 gene networks. For example, in Fig. 1-8C,E, KSG showed a performance enhancement in the range of $\sim 25\text{-}35\%$ for the three different inference algorithms, but the variability was reduced by half when ML instead of KSG was used. This was also shown in the large variance calculated for KSG for a Gaussian distribution (Appendix Fig. II-1D, left column). This observation indicates that KSG is more sensitive to stochasticity (intrinsic noise) when data size is smaller than a few hundred points. Our choice of algorithm KSG-1 over KSG-2 (see Materials and Methods) was intended to keep a low statistical error and thus, low variability. However, using total correlation and two-way mutual information to calculate other measures, such as interaction information (Table 1-1), can lead to higher errors as the systematic errors might not cancel out as we have demonstrated in this work. Additionally, when using KSG, we set negative values of total correlation and two-way mutual information to zero (due to statistical fluctuations at low correlation values) prior to calculating the other 3d MI quantities. This practice does not change the results for pairs or triplets with highly positive MI values, but in some cases could lead to increased errors as gene pairs with low MI would be ranked differently.

We note that two networks (DREAM3 Yeast1-Size50 & *E. coli*2-Size100) out of the 15 networks investigated showed no performance enhancement when using {KSG, CMIA} compared to the Gold Standard {ML, CLR} (Fig. 1-8F, Appendix Fig. II-7). It is unclear why the performance did not improve in these two cases based on the largely similar statistics of different motifs of the ten networks from DREAM3 (Appendix Table II-3). It could be due to a specific sub-structure of this network, but further analysis is needed.

Another important result we observed (Fig. 1-6, 1-8) is that the combination {MM, CLR} achieved higher AUPR for all replicates over {ML, CLR}. This is probably due to the size of the data used, as MM was developed to correct the bias in MI

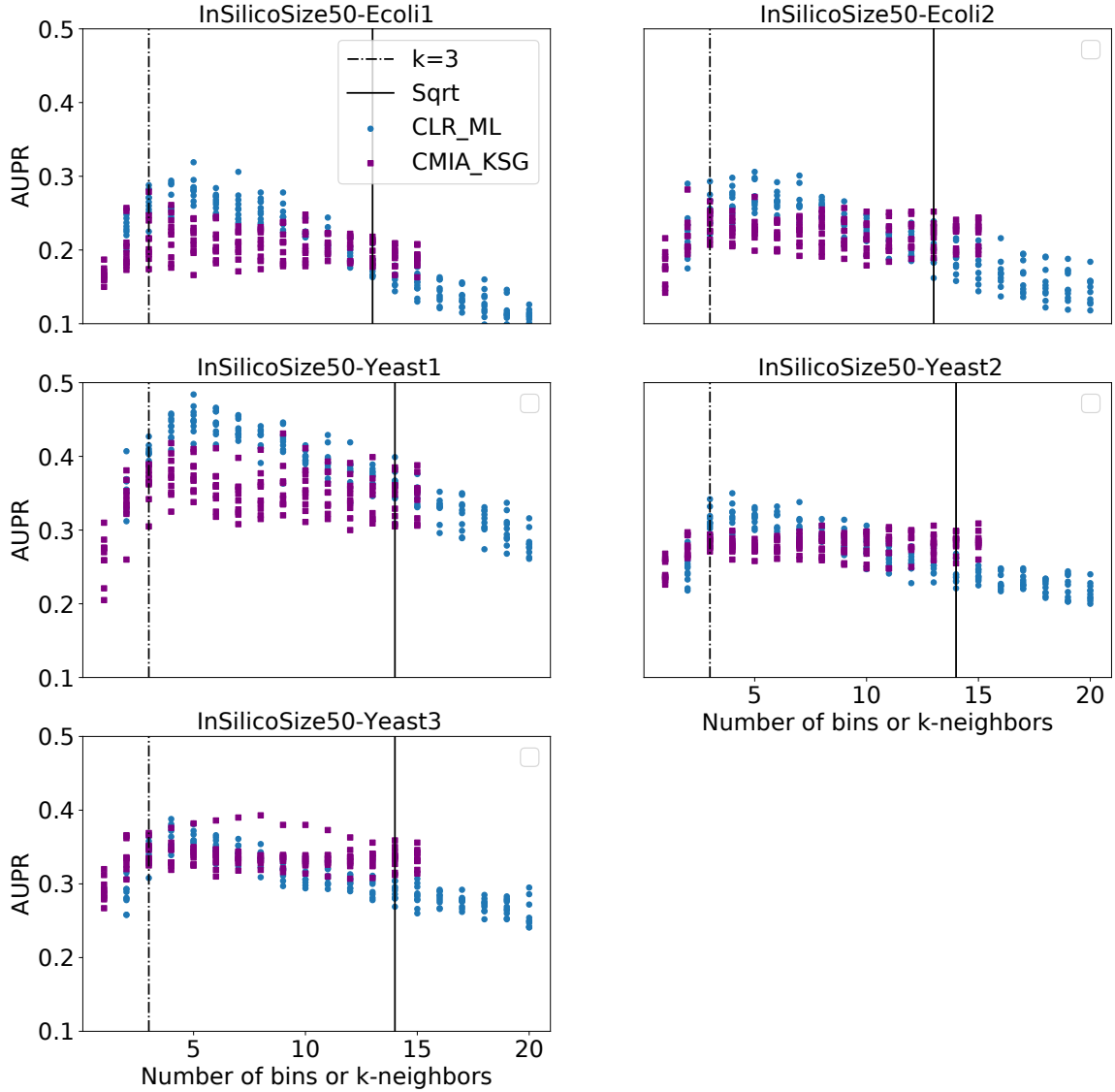


Figure 1-10. Area Under Precision-Recall curve (AUPR) vs. different number of bins or k-neighbors.

For the five 50 gene networks from DREAM3, with 10 replicates each, we calculated the AUPR for two inference algorithm and MI estimator $\{CLR,ML\}$ with blue dots and $\{CMIA,KSG\}$ with purple dots for different number of bins for ML, and different number of k-neighbors for KSG. The black dashed vertical line represents $k=3$ and the solid black line represents $\#bins = \text{floor}(\text{sqrt}(\text{data_pts}))$.

estimation for small data sets. We thus suggest using this combination as the new gold standard of the field when working with similar data sizes and when fixed-binning for data discretization is preferred.

Chapter 2

Classifying three-node network motifs of Transcription Factor (TF)-based regulation

Introduction

In addition to the canonical regulation based on transcription factors (TFs) [1], in recent decades, researchers hypothesize that there is a global gene regulatory network based on supercoiling (SC) [39]. To decouple the two types of regulation, we need to investigate whether they give rise to distinguishable “signatures” in the gene expression profiles of the regulated genes. The complex nature of gene regulation shown in nature [40] and our *in-silico* SC based transcription model [41, 42] requires a non-linear and high dimensional statistical toolset. Therefore, we expect that by analyzing the two- and three-way mutual information (MI), an advance dependency metric, between triplets of genes with an exhaustive simulation benchmark using *in-silico* models, we can find the similarities and differences between the two types of regulation, which may allow us to ultimately construct a topologically regulated gene network for *E. coli*. There are multiple methods to estimate two- and three-way MI (where three-way MI refers to Total Correlation (TC), Three-Way MI (MI3), Interaction Information (II), and Conditional MI (CMI)) for continuous variables (i.e. normalized gene expression) [43]. Following an extensive benchmark on two-

and three-dimensional MI estimators (as described in chapter 1), we have selected the k-Nearest-Neighbor (MI-KNN) developed by Kraskov-Stoögbauer-Grassberger (KSG) [23] to conduct the investigation based on its high accuracy and unsupervised robustness to correlation value and sample size. In this chapter, we are going to take a systems biology approach using network motifs [33], to qualitatively and quantitatively investigate various realizations of TF based regulation, with the final goal of correctly classifying different network motifs based on their gene expression profiles.

Materials and Methods

Datasets

In this study, we have used four datasets, three *in-silico*, and one real experimental dataset for *E. coli*.

Simulating gene expression data for three-node network motifs

Here we generated 100 replicates of simulated gene expression data for 56 three-components motifs with repressing and inducing interactions (see Table 2-1).

For each network motif:

1. Write network topology file (TSV) with 10 genes containing a single motif.
2. Generate new kinetic model in GeneNetWeaver (GNW) [28] for each replicate – this draws new propensities to the mRNA and protein production equations:

$$F_i^{RNA}(x, y) = \frac{dx_i}{dt} = m_i \cdot f_i(y) - \lambda_i^{RNA} \cdot x_i \quad (2.1)$$

$$F_i^{Prot}(x, y) = \frac{dy_i}{dt} = r_i \cdot x_i - \lambda_i^{Prot} \cdot y_i \quad (2.2)$$

Where m_i = maximum transcription rate, r_i = translation rate, λ_i = mRNA and protein degradation rate, x and y = mRNA and protein concentration. $f_i(y)$ = activation function of gene i .



Figure 2-1. Illustration of common network motifs and topology used in GNW

(A) Six common network motifs, from top-left (clockwise): No-interaction, Two-genes, Fan-out, Feedforward loop (FFL), Cascade, Fan-in, (B) Network topology example used in GNW, where the arrowheads represents the direction of regulation. Blue and red color represents inducing or repressing regulation, respectively.

3. Use perturbation file (same file for all motifs and replicates) – `random.uniform(-1,1)` => this specify the amount by which the basal transcription rate is perturbed for every gene => 1000 different perturbations
4. Generate datasets based on stochastic differential equation (SDE) for steady-state (S.S.) levels of multifactorial perturbations of the network.

DREAM 3 & 4 challenge datasets

The generated networks in the Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenge are inspired by real network structures discovered in *E. coli* and *S. cerevisiae* [32] and are widely used for benchmarking inference methods. We used 10 *in-silico* networks of size 50 and 100 from the DREAM-3 benchmark, where networks do not include self-interaction, bidirectional interaction between genes (only single edge) and feedback loops. And 5 networks of 100 genes from DREAM-4, where cycles and two-way interactions are allowed. We used the software GeneNetWeaver v3.1.2b [28] with default settings to simulate 10 replicates of each network steady-state gene expression data (wild-type, knockouts, knockdowns, multifactorial perturbations) based on SDE. From those networks we extracted most three-node motifs based on

Table 2-1. Simulated Motifs

Motif name	Interactions (edges)	Sign motifs	Inducing (I), Repressing (R)	Tot. motifs wPermute	Simulated
No Interaction	0	1	-	1	1
Two genes	1	2	I,R	12	2
Two genes	2	3	II,IR,RR	12	5
Fan-out	2	3	II,IR,RR	12	5
Fan-in	2	3	II,IR,RR	12	7
Cascade	2	4	II,IR,RI,RR	24	9
Coherent feedforward loop	3	4	III,IRR RIR,RRI	24	9
Incoherent feedforward loop	3	4	IIR,IRI, RII,RRR	24	4
Feedback loop	3	4	III,RRR,IIR,RRI	16	2
...
Motif13	6	16	Partial list: IIIRII, IIIRRR, IIRIII, IIRIRR, RRIRII, RRIRRR, RRRIII,RRRIRR	64	8
Co-reg*	0	11	II,IR,RR,III,IRR,RIR, RRI,IIR,IRI,RII,RRR	60	4

the true structure and removed triplets that are participating in two or more motifs simultaneously, which bias the expression profiles and potentially makes machine learning models training on the data less accurate.

***Escherichia Coli* experimental data**

We use the publicly available compendium of *E. coli* genomic steady-state expression data (<http://m3d.mssm.edu/>) containing 907 experimental conditions for 4297 genes, collected from multiple labs [44]. To evaluate the prediction of our method, we have extracted three-node motifs from the 3969 strongly evident regulatory interactions (after removing self-interaction) between 206 transcription factor (TF) and among 1642 genes, documented in RegulonDB v10.9 [45]. Furthermore, we removed triplets where a target gene was co-regulated by a source outside the triplet (a fourth gene) and added motifs with permuted axis (to increase sample size for low occurring motifs).

We finally obtained {18,29} samples for Cascade motifs, {25,69} for FFL, 220 for Fan-in, 7565 for Fan-out, and sampled 5000 for Two-genes and 100k for No-Interaction motif.

Calculating mutual information and Z-score

We calculated all 2d and 3d mutual information quantities (MI, TC, II, MI3, CMI) using KSG with k=3 (as described in chapter 1). As KSG calculates two-way MI and TC directly, we use these quantities to calculate II, MI3, and CMI (Table 1-1). A small caveat to using kNN based MI is that we can get small negative values due to statistical fluctuations when there is no (or very low) correlation between genes. We set negative MI and TC values to zero (by definition they are positive) prior to calculating II, MI3, and CMI, and afterwards set negative values to zero for MI3 and CMI if any exist. We calculate Z-score for each quantity following the implementation in Minet [29]

$$Zscore_X = \max\left(0, \frac{MI(X;Y) - Mean(MI_X)}{STD(MI_X)}\right) \quad (2.3)$$

$$Zscore_{XY} = \sqrt{(Zscore_X)^2 + (Zscore_Y)^2} \quad (2.4)$$

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [46] is a dimensionality reduction method, by which we transform our data to a new coordinate system which emphasizes variance between variables (dimensions). Where the first principal component contains the most variance the second contains variance not captured by the first component and so on. The new components are orthogonal to each other and thus are uncorrelated. This allows us to accomplish few objectives:

1. Visualizing high-dimensional space by using only 2-3 principal components dimensions to recognize clusters or patterns by the naked eye.

2. Classification (inference) – We can use the new principal components as input to a clustering algorithm, i.e. K-means which can assist in classification.
3. Data compression - as some variables are almost completely redundant, we can rank the new principal coordinates according to their contribution to the variance and omit the ones that contribute the least according to our accuracy requirements.

We use PCA, clustering and scaling functions in python from scikit-learn v0.24.1 [47].

Clustering algorithm

K-means [48] is an unsupervised clustering algorithm that gets the number of clusters as input (in our case, the number of network motifs investigated that we want to classify) and tries to separate the data to K groups of equal variance. It starts by choosing K random data points, named “centroids” and calculate the distance (some metric) from each point to each centroid. Next, it calculate the mean (position) for each cluster based on the points that are closest to that centroid and update the centroid position according to the newly calculated mean. It repeats this process iteratively until there are no more updates to the clusters centers (or a small number below a threshold) and within-cluster sum-of-squares is minimized. We use 10 random initializations to avoid converging to a local minimum.

$$\text{within-cluster sum-of-squares} = \sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \quad (2.5)$$

Machine Learning models

Conventional machine learning methods for inferring gene regulatory networks, that use only gene expression data as input, usually split the problem into multiple regression analysis problems. In this way, for a network of N genes, for each gene j , in the network, we consider it as a target and all other $N - 1$ genes as sources which

determine its expression profile by some unknown function $f(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_N)$. Each individual problem then become an optimization for some merit (loss/gain) function. Furthermore, this allows us to estimate the relationship between each source, i , and the target gene j , with some weight, w_{ij} . The algorithm then try to rank the weights for each target gene, and combine the ranking from the N regression problems to generate a global ranking for the interactions between genes [49]. We compare the classification performance using four commonly used Machine Learning (ML) algorithms: Support Vector Machine (SVM) [50], Multilayer Perceptron (MLP, also known as, artificial neural network) [51], Random Forest (RF) [52], and Gradient Boosted Trees (GBT) [53, 54]. We use machine learning functions in python from scikit-learn v0.24.1 [47] and scan a subset of available parameters for each method (Table 2-2) to find the best model in each machine learning type under our training data (60% of samples). As this significantly reduces our training data, we use 5-fold cross-validation on our training data, meaning we split the training data by 5 and train the model on 4/5 of it, saving the last 1/5 to test the model. We then repeat this process 4 more times and average the precision and recall results (see definitions below). We choose the best model from each machine learning family and proceed to the evaluation step. We use our evaluation data (20% of samples) to choose a single model out of the 4 family types. Finally, we use the remaining 20% testing data to confirm that our best model is indeed performing similarly to what we got on the training and evaluation data.

Performance evaluation

To evaluate the performance of the various machine learning models investigated, we use the known (true) network motif and count the number of true and false positives (TP and FP respectively) predictions as well as false negative (FN). This allowed us to calculate for each model the precision and recall for each individual motif and the

Table 2-2. Machine Learning models and parameters

Model	Parameter	Range	Description
SVM	Kernel	linear, rbf	Used for decision func.
SVM	C	0.1, 1, 10	Inverse regularization parameter. Simplicity and overfitting trade-off
MLP	Hidden layer size	10, 50, 100	Neurons in the hidden layer
MLP	Activation	relu, tanh, logistic	Function for the hidden layer
MLP	Learning rate	constant, adaptive, invscaling	Schedule for weight updates.
RF	# estimators	5, 50, 250	Number of trees in the forest
RF	Max depth	2, 4, 8, 16, 32, None	Tree size (# of splits). None = tree grows until split criteria reached
GBT	# estimators	5, 50, 250, 500	Boosting stages to perform
GBT	Max depth	1, 3, 5, 7, 9	Max depth limits the number of nodes in the tree.
GBT	Learning rate	0.01, 0.1, 1, 10, 100	Learning rate shrinks the contribution of each tree

total accuracy:

$$Precision(motif) = \frac{TP(motif)}{TP(motif) + FP(motif)} \quad (2.6)$$

$$Recall(motif) = \frac{TP(motif)}{TP(motif) + FN(motif)} \quad (2.7)$$

$$Accuracy = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1, & \text{if prediction is true} \\ 0, & \text{if prediction is false} \end{cases} \quad (2.8)$$

Where N is the total number of samples.

Results

Using unique Mutual Information profiles to identify network topologies

Simulated Two-genes motifs

We first examined the MI values of 100 replicates of seven two-genes motifs using a boxplot (Fig. 2-2). Where the x-axis represents each of the 2d and 3d mutual

information quantities (11 in total, see Table 1-1) and the y-axis is the MI value in units of nats. Each subplot shows the “MI profile” of a different motif (where we use the name convention, {motif name}-{direction}_{inducing/repressing edges}). It is clear from the MI values (y-axis) that we can distinguish which pair is interacting (XY , XZ or YZ), but as two-way MI is symmetric ($MI(X;Y) = MI(Y;X)$), we can not tell the direction of interaction ($X \rightarrow Y$ or $Y \rightarrow X$). Another observation is that positive interaction (the two edges are either all inducing or repressing) between two nodes gives a higher median value than one edge. Next, we looked at inducing (I) versus repressing (R) regulation (Fig. 2-2). In principle, due to the positive nature of all 2d and 3d MI quantities, except for Interaction Information (II), it is not possible to differentiate between inducing and repressing interaction, unlike correlation. It is interesting to observe that negative feedback (inducing + repressing interaction) gives the lowest median MI value among the two-genes-two-edges motifs and closer to single edge interaction.

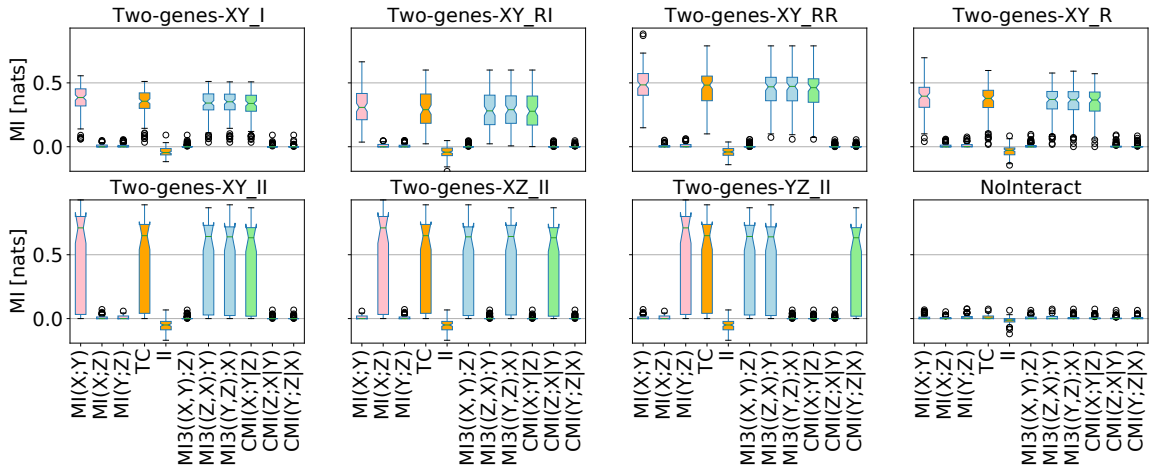


Figure 2-2. Mutual Information profiles for Two-genes motifs

Each subplot (boxplot) represents a unique network motif. We show 7 Two-genes motifs and the No-interaction motif, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the MI value in nats (information bits in e base).

Following the increased accuracy obtained by CLR [13] method in inferring gene regulatory networks by using Z-score statistics compared to other methods relying

solely on MI values, we applied this methodology (see Materials and Methods) and plot the Z-score profiles (SI-Fig. III-1). Except for the unique pattern of the profiles, for the Two-genes motifs this does not give us any additional information compared to the MI profiles based on MI values.

Simulated Three-genes motifs

There are 729 three-genes motifs (also includes two-genes motifs) if we account for inducing, repressing or no interaction among the genes and all possible permutations of X, Y & Z [55]. We chose to investigate a subset (Table 2-1) and present in Fig. 2-3 the MI profiles of all three-genes with two edges (omitting two-genes motifs shown previously) and only inducing interaction (for those motifs, repressing or mixed interactions effect MI values globally and does not alter the relationship between quantities, see SI-Fig. III-2).

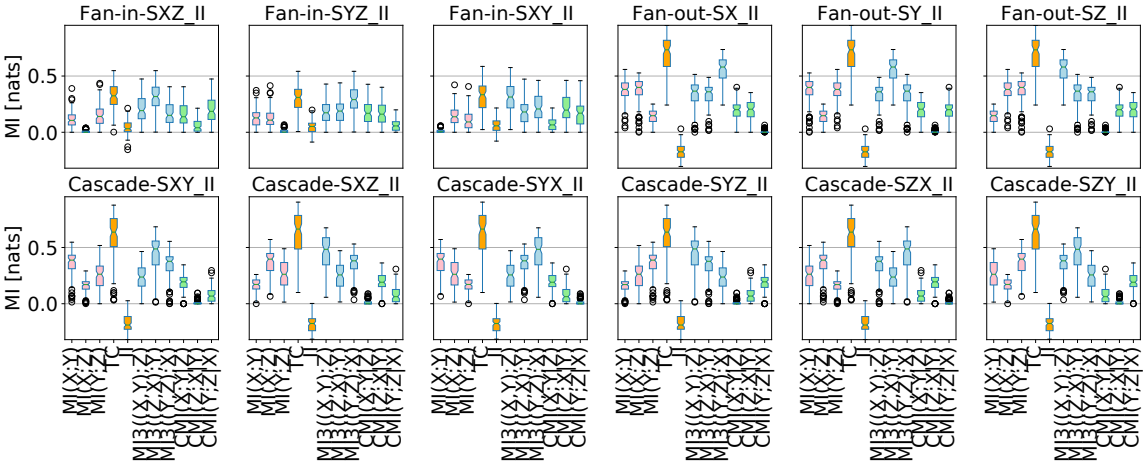


Figure 2-3. Mutual Information profiles for two-edge motifs

Each subplot (boxplot) represents a unique network motif. We show 12 motifs with two-edges, namely, Fan-in, Fan-out and Cascade, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the MI value in nats (information bits in e base).

It is clear from the MI profiles that all 12 motifs investigated can be distinguished from one another, including the direction of regulation (which genes are the source (regulator) and target (regulated)). A few notable examples are: (i) **Fan-out-SX**

versus Fan-in-SYZ, where we have the same edges (XY and XZ), but, while for Fan-out we measure two-way MI between the targets as both are control by the same source, for Fan-in, there is no two-way MI between the sources. This also leads to redundant II ($II < 0$) for Fan-out and synergistic II ($II > 0$) for Fan-in. Where synergy is the information gain when considering the effect of a third variable (gene) on the shared information between two variables v.s. the shared information between two variables without considering the third variable (see Appendix I). (ii) **Cascade-SXY versus Cascade-SXZ**, can be distinguished thanks to Data Processing Inequality (DPI) [14]. In the Cascade-SXY motif, DPI states that if X and Z are connected directly through an intermediate gene Y , then $MI(X; Z) < \min[MI(X; Y), MI(Y; Z)]$ as for a linear network structure, information can only decrease when passed through multiple nodes (genes).

For motifs with three edges, we found that each feedforward-loop (FFL) motif has a similar Cascade motif but with opposite direction for the leading interaction (Fig. 2-4) (here we consider only inducing interactions for FFL). The only meaningful difference, although small is in conditional MI (CMI) for the indirect interaction in the Cascade motif (Z -score ~ 0) versus the long interaction in the FFL motif (Z -score < 1) (see Z -score profiles for all simulated motifs in Appendix III-3). $CMI(Z; X|Y)$ means that given that we know variable Y , what is the two-way MI between Z & X ? In the Cascade-SXY case, if we know Y , there is no information to be gained by measuring $MI(X; Z)$, and so $CMI \sim 0$, while for FFL-SXY, $CMI > 0$, as knowing Y doesn't incorporate the information in the direct interaction $X - Z$.

Another issue with three-edge motifs is that unlike 1-2 edge motifs, FFL's MI profiles depends on the nature of interaction (repressing or inducing) (Fig. 2-5). Furthermore, we can not distinguish between a coherent FFL (where both long and short regulation path have the same sign) and incoherent FFL (where the long and short regulation path have opposite signs) and thus only the direction of the 2nd and

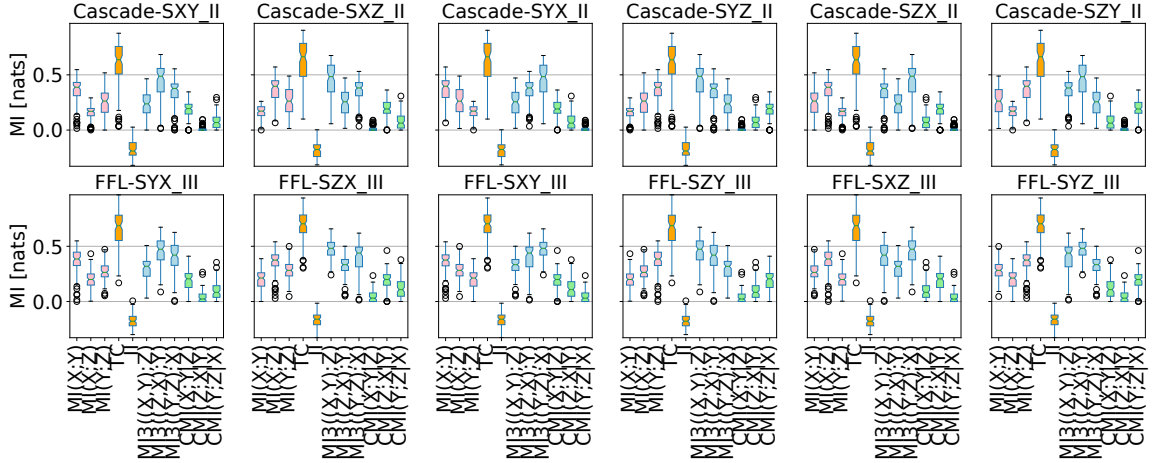


Figure 2-4. Mutual Information profiles of Cascade vs FFL motifs

Each subplot (boxplot) represents a unique network motif. We show 6 Cascade motifs (top row) with their equivalent FFL motifs (bottom row), where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the MI value in nats (information bits in e base).

3rd interactions is certain, allowing us to distinguish between three pairs of motifs $\{\{S\text{-XY}, S\text{-YX}\}, \{S\text{-XZ}, S\text{-ZX}\}, \{S\text{-YZ}, S\text{-ZY}\}$, where we use the name convention $S\text{-}\{\text{first source}\}\{\text{second source}\}$.

Finally, we investigated the MI profiles of two feedback loops (all inducing or repressing) Fig. 2-6, as well as eight realizations of Motif13 [33] which has 6 edges. Loop_III has a unique MI profile where II is highly redundant and all CMI are close to zero, but Loop_RRR, shows significant CMI and almost no redundancy. Intuitively, in the positive loop, looking at a single pair is enough to know the behavior of the other pairs but for the negative loop as shown by Elowitz et. al. [56], you need another gene to explain the oscillatory behavior. Motif13 poses similar complexity as FFL with different combinations of inducing/repressing interactions, as discussed above, and is beyond the scope of this semi-quantitative methodology.

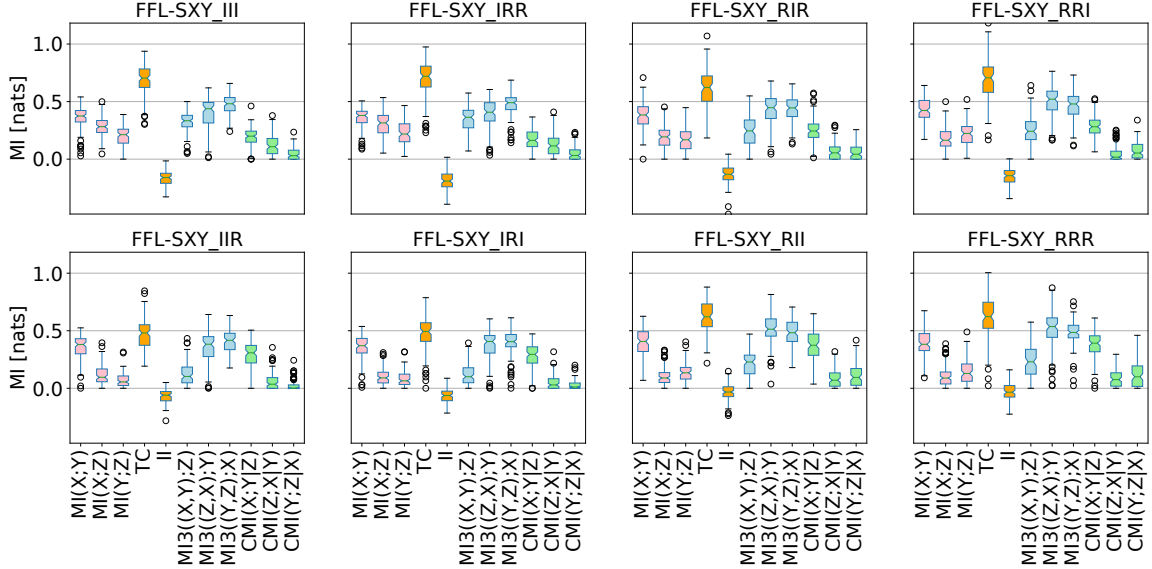


Figure 2-5. Mutual Information profiles of FFL coherent vs incoherent motifs
 Each subplot (boxplot) represents a unique network motif, for FFL-SXY with different inducing (I) and repressing (R) interactions. Top row are 4 coherent FFL, and bottom row are 4 incoherent FFL, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the MI value in nats (information bits in e base).

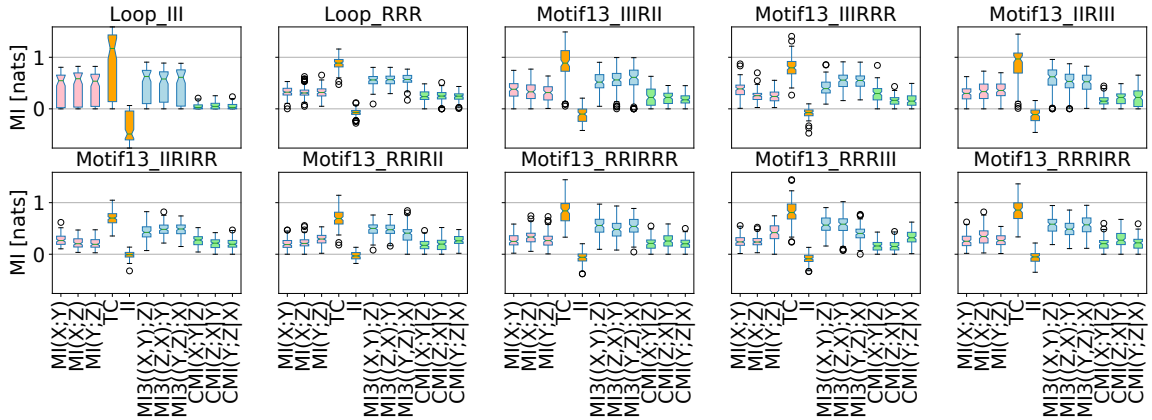


Figure 2-6. Mutual Information profiles of Loop and Motif13
 Each subplot (boxplot) represents a unique network motif. We show 2 feedback loop motifs and 8 Motif13 (6 edges), where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the MI value in nats (information bits in e base).

Results of simulated motifs vs. motifs extracted from DREAM3-4 networks

Next, we explored the MI profiles of various motifs extracted from 15 *in-silico* networks of size 50 and 100, used in the DREAM-3 & 4 benchmarks. In Fig. 2-7 we show the

mean MI values of the following DREAM-3 motifs: No-Interaction, two-genes, Fan-in, Fan-out, Cascade and FFL. In general, the MI values are $\sim \times 4$ smaller than the simulated motif examples, giving noisier MI profiles which reduces the signal-to-noise ratio. This makes it more difficult to divide each MI value into separable discrete levels as can be done in the simulated motifs examples. It is interesting to see how the profiles for Cascade has changed compared to our simulated motif toy-model. Now, the highest mean MI value is measured between the second direct interaction instead of the first direct interaction, as was seen in the previous section. FFL has also changed significantly in its MI profile, making the interaction from first source to second the strongest, and from first source to target (short regulation path) the weakest. This makes it almost impossible to determine the direction of regulation in Cascade and FFL motifs by solely inspecting their mean MI profiles.

For completeness, we added two- and three co-regulated motifs (XY, XZ, YZ, XYZ), which means that a fourth gene regulates two or three genes but it (the source) is not included in the MI triplet calculation. This gives rise to MI between the genes in a way similar to the MI measured between two-genes (co-reg- $XY / XZ / YZ$) or targets of a Fan-out motif (co-reg- XYZ), as can be seen in Fig. 2-7.

In Fig. 2-8 we show DREAM-4 extracted motifs. For this dataset we are missing 3 Cascade and 3 FFL motifs, as we discarded samples where a fourth gene regulated a member of the triplet extracted). As seen in the DREAM-3 dataset the MI values are smaller and noisier compared to our simulated motif dataset.

As distinguishing the motifs became more difficult, in the next section we examined whether a method like PCA can assist by emphasizing variance between samples and reducing the dimensionality of this high dimensional space (22 variables if we include both MI values and their Z-scores).

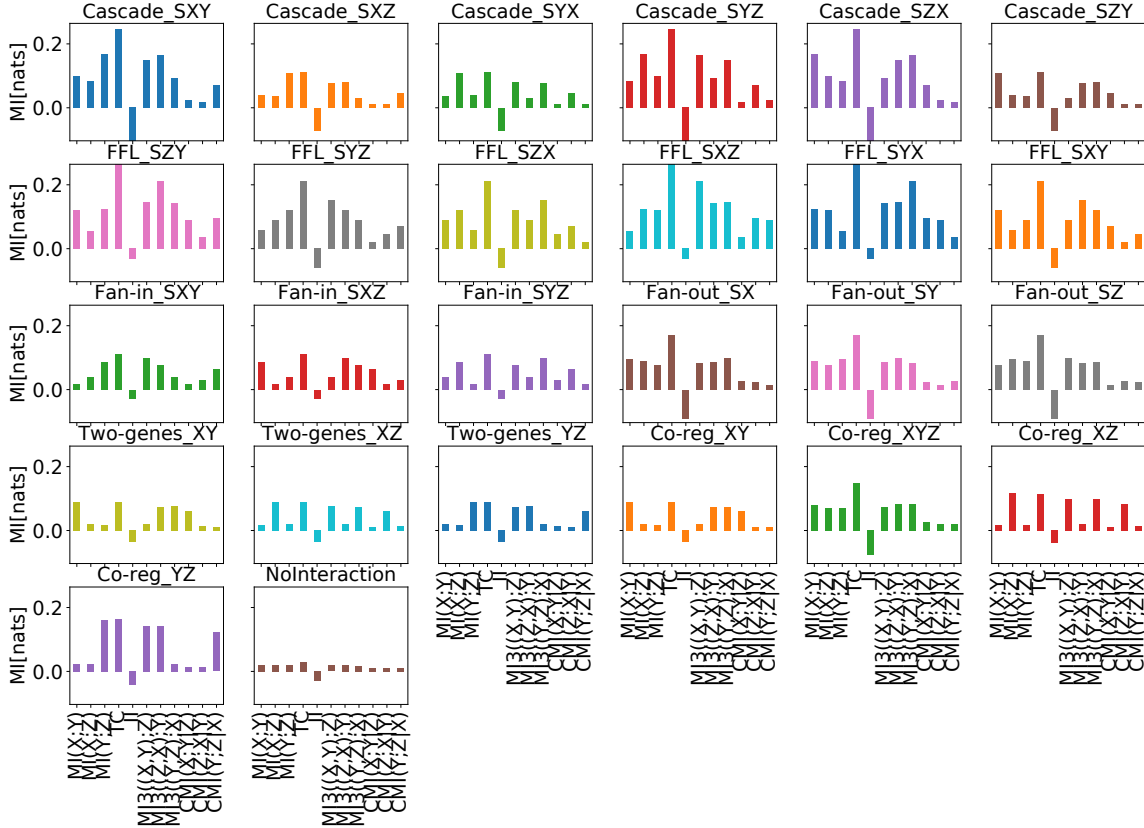


Figure 2-7. Mutual Information profiles of DREAM3 motifs

Each subplot represents a unique network motif. We show 26 motifs extracted from 10 DREAM-3 networks of 50 and 100 genes, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the mean MI value in nats (information bits in e base).

Using dimensionality reduction by Principal Components Analysis (PCA) to classify different three-node network motifs

PCA for simulated 3-node motifs

After we calculated all the two- and three-way MI measures (22 quantities in total, with their Z-score counterparts Fig. 2-3,2-4,III-3) we use Principal Component Analysis (PCA) [46] to reduce this high dimensional space to only 3-5 dimensions that capture together $\sim 85 - 99.5\%$ of the variance in the original data (Fig. 2-9B). With fewer dimension we can now plot the different principal components against each other and investigate whether we could distinguish qualitatively between different network motifs, such as 3 Fan-in vs. 3 Fan-out (Fig. 2-9A). By inspecting the two subplots

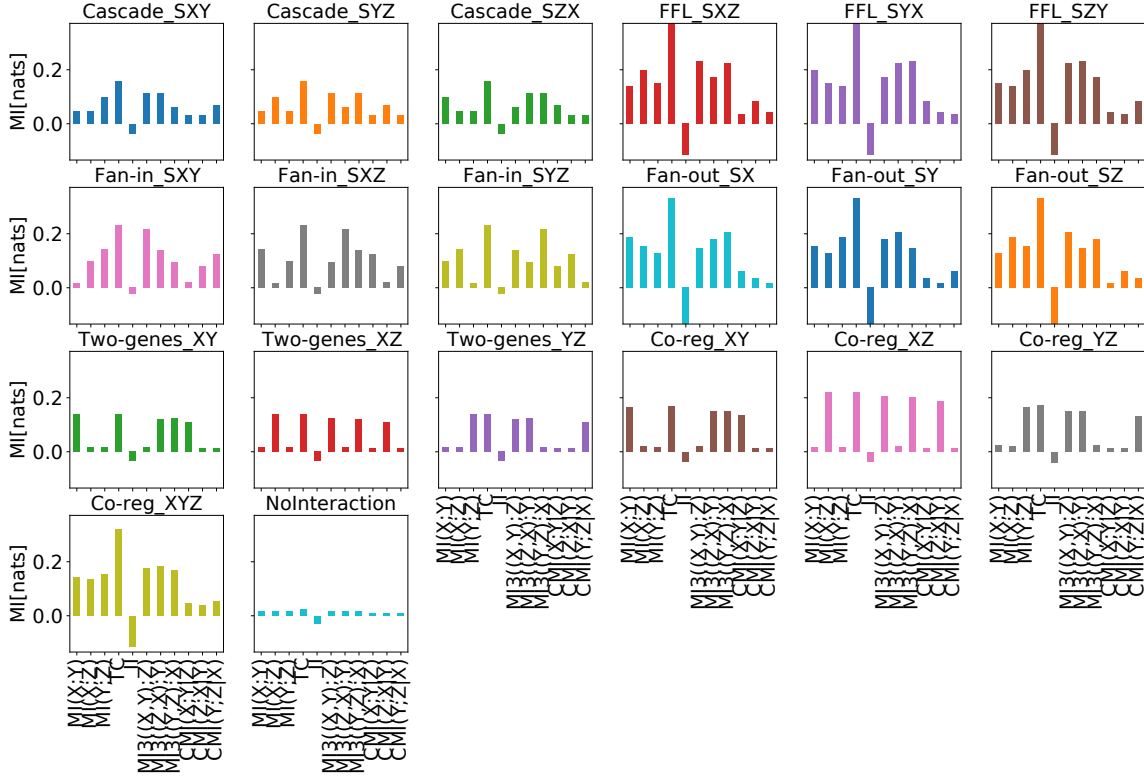


Figure 2-8. Mutual Information profiles of DREAM4 motifs

Each subplot represents a unique network motif. We show 20 motifs extracted from 5 DREAM-4 networks of 100 genes, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the mean MI value in nats (information bits in e base).

PC-1 vs PC-2 and PC-2 vs. PC-3, we can easily cluster by the naked eye the 6 motifs.

To translate the qualitative clustering visualized when using PCA to a prediction tool for classifying motifs for new data, we need to combine PCA with a clustering algorithm such as K-means (see Materials and Methods). We applied K-means to the 6 motifs (3xFan-in + 3xFan-out) and used $k=6$ and the first 4 principal components (PC-1,PC-2,PC-3,PC-4) as input data. K-means determined the location of the 6 centroids (big black circles in Fig. 2-9A overlaying the data points). We then assigned the correct label (motif) to each centroid (as this is an unsupervised method) and tested the cluster prediction for each data point to calculate the accuracy. We used the same data to “train” the K-means algorithm and test the predictions, and got an overall accuracy of 0.612.

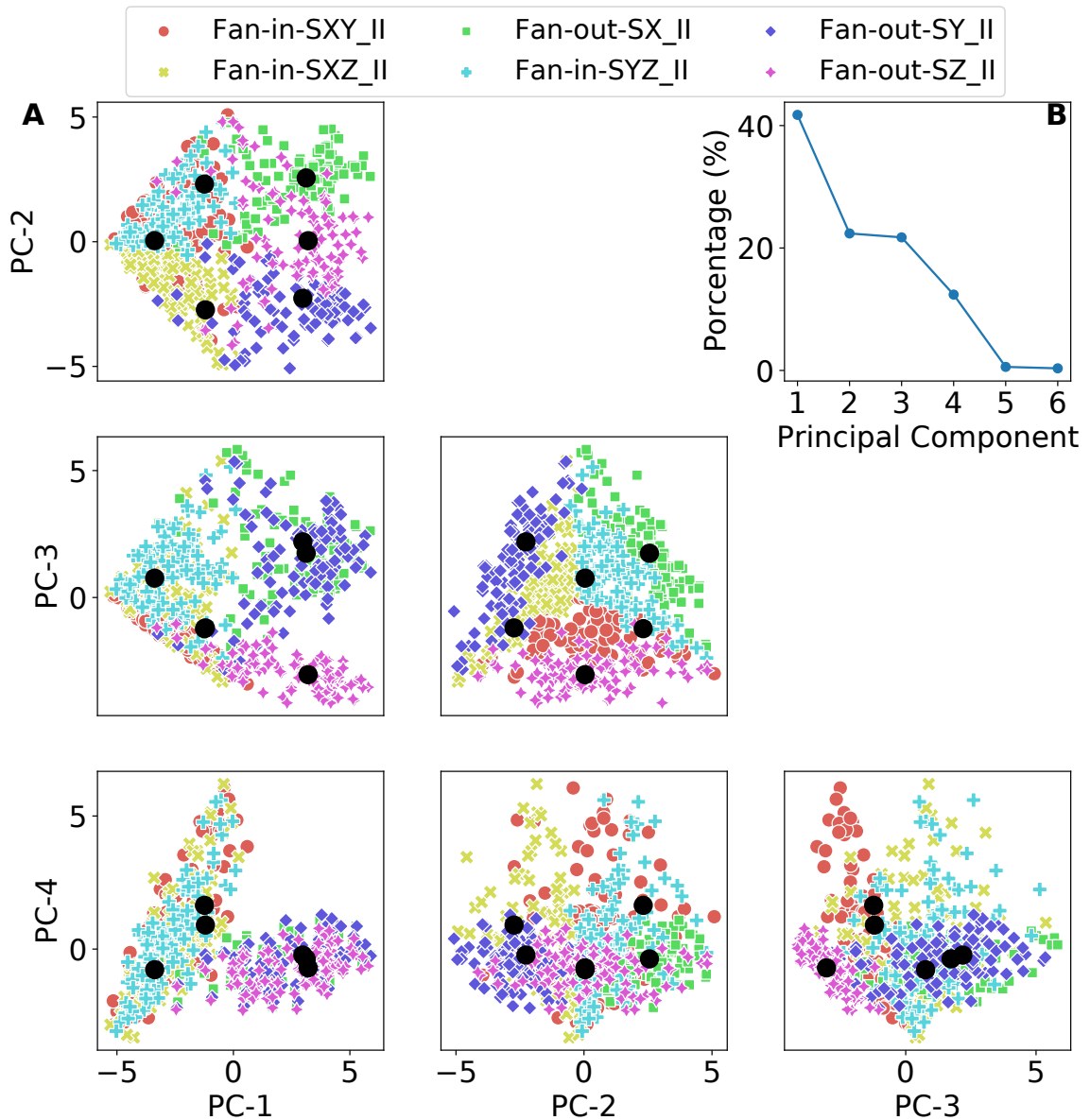


Figure 2-9. Principal Component Analysis and K-means clustering for Fan-in/out motifs (A) 2D plots of different principal components vs. each other for 6 network motifs (small colored markers), and the centroid location of 6 clusters (black markers) calculated by K-means, (B) PCA analysis showing that 4 principal components capture close to 99% of the variability in the data. y-axis is the percentage of variance explained by a principal component, x-axis is the six components used in the analysis.

We have repeated the same PCA and K-means methodology with the 6 Cascade motifs (Fig. 2-10), but it was significantly harder to manually assign the correct motif label to each cluster (big black circles in Fig. 2-10A), and the accuracy achieved was

only 0.465. K-means is known to work poorly on elongated or irregular shape data, and as data becomes more interwound the clustering accuracy decreases.

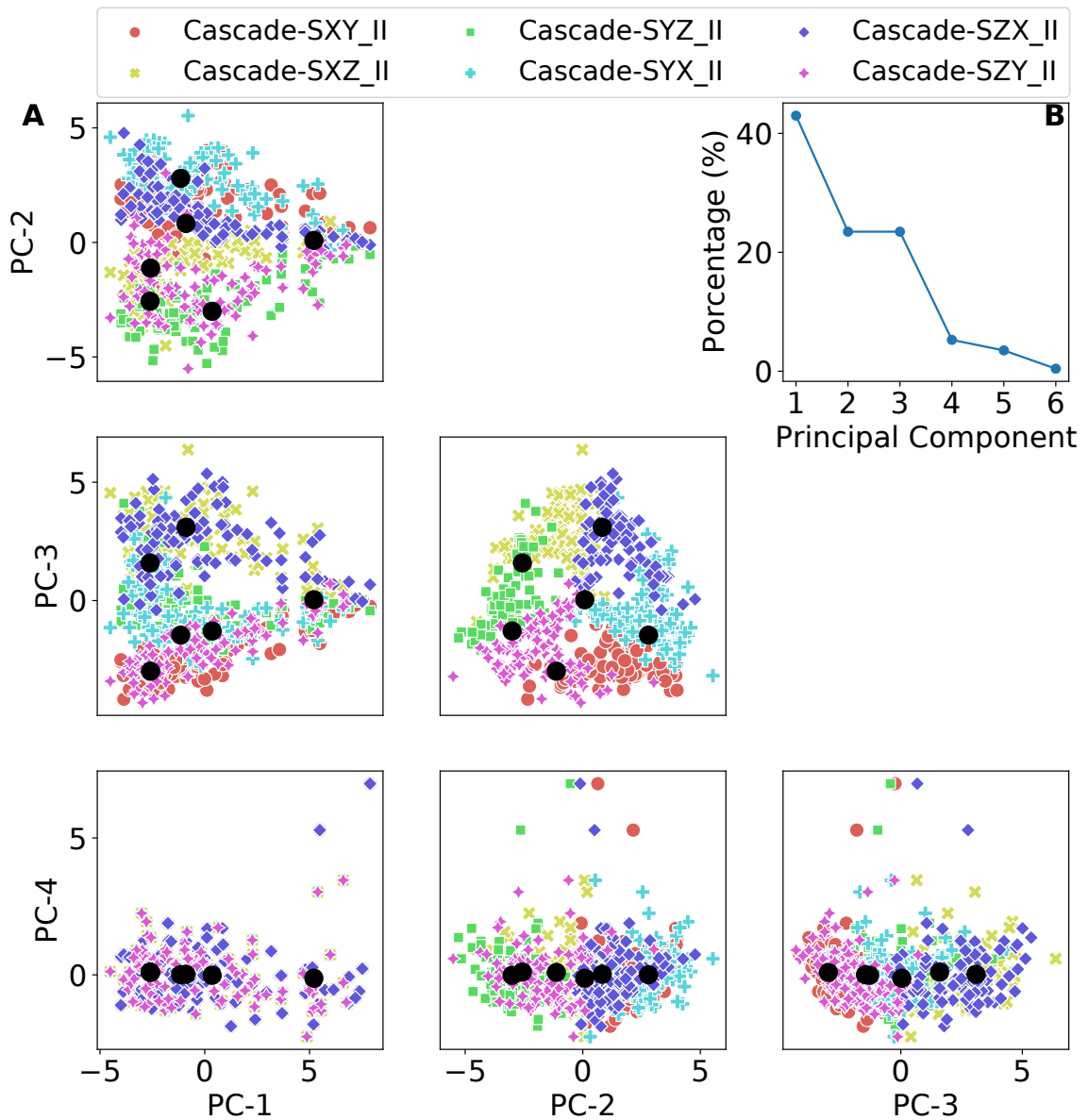


Figure 2-10. Principal Component Analysis and K-means clustering for Cascade motifs (A) 2D plots of different principal components vs. each other for 6 network motifs (small colored markers), and the centroid location of 6 clusters (black markers) calculated by K-means, (B) PCA analysis showing that 4 principal components capture close to 96% of the variability in the data. y-axis is the percentage of variance explained by a principal component, x-axis is the six components used in the analysis.

Furthermore, adding motifs beyond 6 makes it impractical to visually assign most

motifs to their corresponding centroids (Fig. 2-11). This led us to try a supervised machine learning approach to classify the various network motifs.

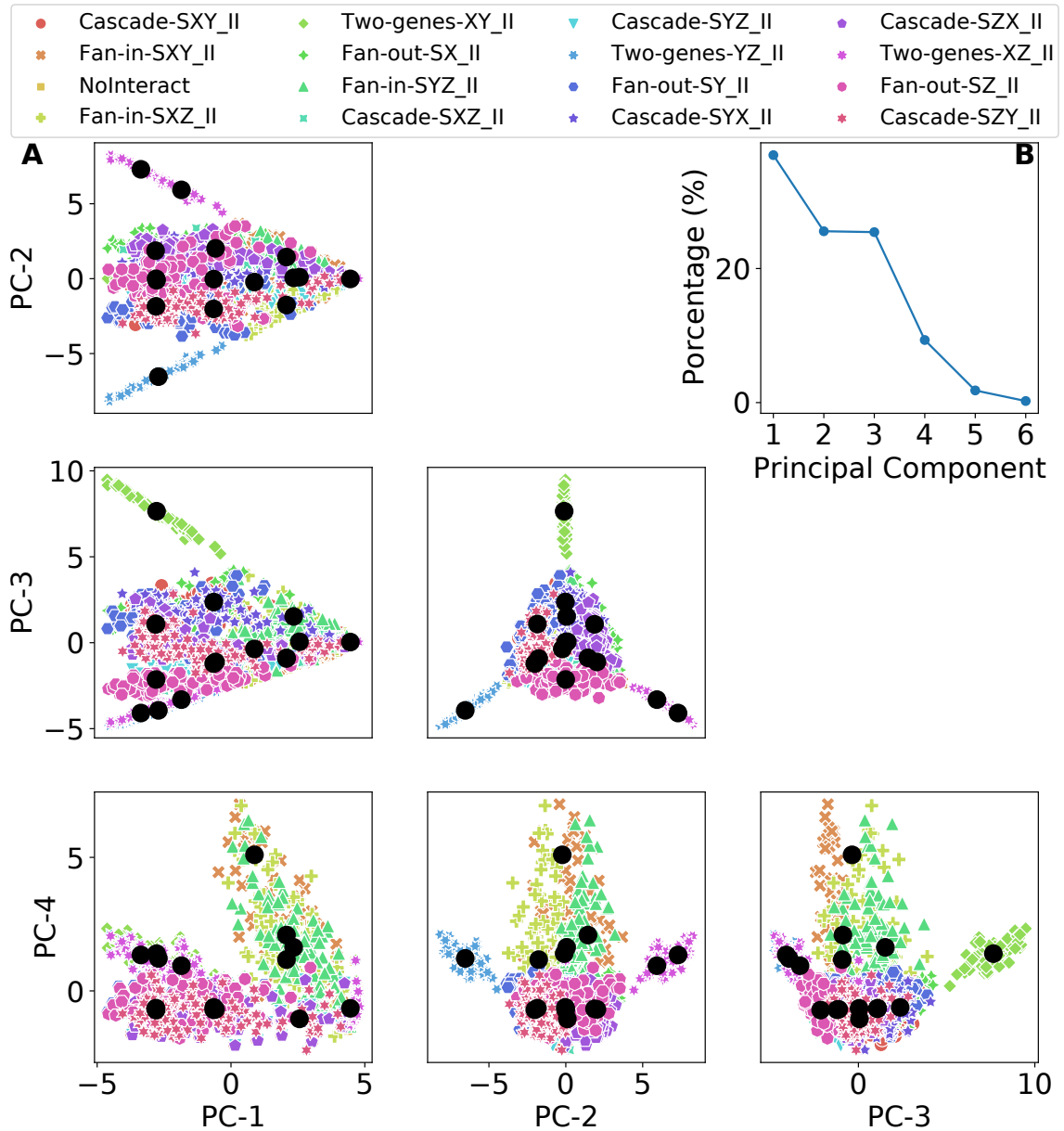


Figure 2-11. Principal Component Analysis and K-means clustering for 16 motifs (A) 2D plots of different principal components vs. each other for 16 network motifs (small colored markers), and the centroid location of 16 clusters (black markers) calculated by K-means, (B) PCA analysis showing that 4 principal components capture close to 98% of the variability in the data. y-axis is the percentage of variance explained by a principal component, x-axis is the six components used in the analysis.

Network motif classification by Machine Learning (ML) algorithms

We compared the performance of four types of machine learning algorithms, namely, SVM, MLP, RF and GBT, under various network motif subsets (No-Interaction, 3xTwo-genes, 3xFan-in, 3x-Fan-out, 3-6xCascade, 3-6xFFL) with different training data size (15,30,60 samples for each motif) for the three datasets (Simulated 3-node motifs, extracted motifs from DREAM-3 & 4). For each algorithm we performed parameter scan and cross-validation (see Materials and Methods) on the training data and picked the best model (algorithm with specific parameter). For our current purpose, the details of specific parameters are less important, and we only focus on the parameter set that yields the highest accuracy, but one might prefer a different parameter set that emphasizes i.e. model simplicity against overfitting or shallower layers or steps for faster computation. We then compared the accuracy of the 4 top models (one for each algorithm type) on the evaluation data and finally on the testing data. We summarize our findings in Table 2-3, where we recorded the accuracy of the best model. A representative example of the evaluation comparison together with precision and recall values for each motif can be found in the supplementary information section.

It is clear from the table, that the tree-based method GBT outperforms the other methods 2/3 of the times. Its main disadvantage is its significantly slower training, as it learns sequential (hours vs. seconds or minutes for the other models). Overall, the accuracy of predictions is very high ($\sim 89 - 98\%$) for a relatively small training set and 16 labels, except for the DREAM-3 dataset, where we only got $\sim 60\%$ accuracy. For our 3-node simulated motifs we had 100 replicates for each motif, but in the case of DREAM-3/4 *in-silico* networks, some motifs were missing or had relatively small number (30-50). As we sampled at least 100 realizations from each motif, this might introduce a bias. Yet, it is unclear why models that trained on DREAM-3 preformed

Table 2-3. Machine Learning classification accuracy on *in-silico* data

Motifs	Train samples	simulated 3 node data	DREAM3 extracted	DREAM4 extracted
16 (No FFL)	15	SVM & MLP: Accu.=0.912	GBT: Accu.=0.6	GBT: Accuracy=0.912 *missing 3 cascades
16 (No FFL)	30	SVM & MLP: Accu.=0.925	GBT: Accu.=0.59	GBT: Accuracy=0.906 *missing 3 cascades
16 (No FFL)	60	MLP Accu.=0.903	GBT: Accu.=0.594	GBT & RF: Accuracy=0.894 *missing 3 cascades
DREAM4 extracted (16 with 3xCascade and 3xFFL)	15	RF: Accu.=0.8	GBT: Accu.=0.602	GBT: Accuracy=0.911
DREAM4 extracted (16 with 3xCascade and 3xFFL)	30	RF: Accu.=0.975	GBT: Accu.=0.609	GBT: Accuracy=0.912

poorly compared to the same models that trained on DREAM-4 data, as both were simulated with the same software using default settings and have similar network characteristics (Appendix Table II-3). Further analysis will be needed to reconcile this discrepancy.

Next, we repeated the same methodology, but this time used only the first 5 principal components (which explains $\sim 99\%$ of the variance in the data) as input to the models instead of the 22 variables used previously. Interestingly all the models performed significantly poorer compared to the untransformed data (Table 2-4), with reduction of $\sim 25 - 30\%$ in accuracy.

We have also tried to normalize the principal components data between $[0,1]$, but got similar results to the unnormalized data.

Table 2-4. Machine Learning classification accuracy on PCA data

Motifs	Train samples	simulated 3 node data
DREAM4 extracted (16, no Co-reg)	15	SVM: Accuracy=0.612
DREAM4 extracted (16, no Co-reg)	30	GBT: Accuracy=0.681

Identify network motifs on *Escherichia Coli* expression data

To test our classification method on real experimental data, we have run our algorithm on a publicly available compendium of *E. coli* genomic expression data at steady-state [44] (see Materials & Methods). In Figure 2-12, we show mean Mi profiles (among 200 samples) of the various 3-node motifs extracted from *E. coli*. Qualitatively, there is larger background noise indicated by non-zero MI measure between pairs that are not correlated, i.e. the two source genes of a Fan-in motif, or two-genes pairs that are not directly connected. This makes the task of distinguishing the various motifs based on their MI profile more difficult and will probably lead to lower accuracy using machine learning classification models (see below). We did not perform any data curation other than choosing target genes that are not regulated by more than one motif based on documented interaction in RegulonDB [45].

Next, we randomly sampled the motifs (50 samples per motif) and further split samples into 30-10-10 for machine learning models training, evaluation and testing, respectively. In Table 2-5 we summarize the precision per motif and overall accuracy results for the testing data (160 samples that the model did not see before).

Where the parameters for each best model for each trained dataset are:

- *E. coli* - 'GBT': (max_depth=5, n_estimators=250)
- Simulated 3-node: 'RF': (max_depth=32, n_estimators=50)
- DREAM4 - 'GBT':(max_depth=7, n_estimators=500)

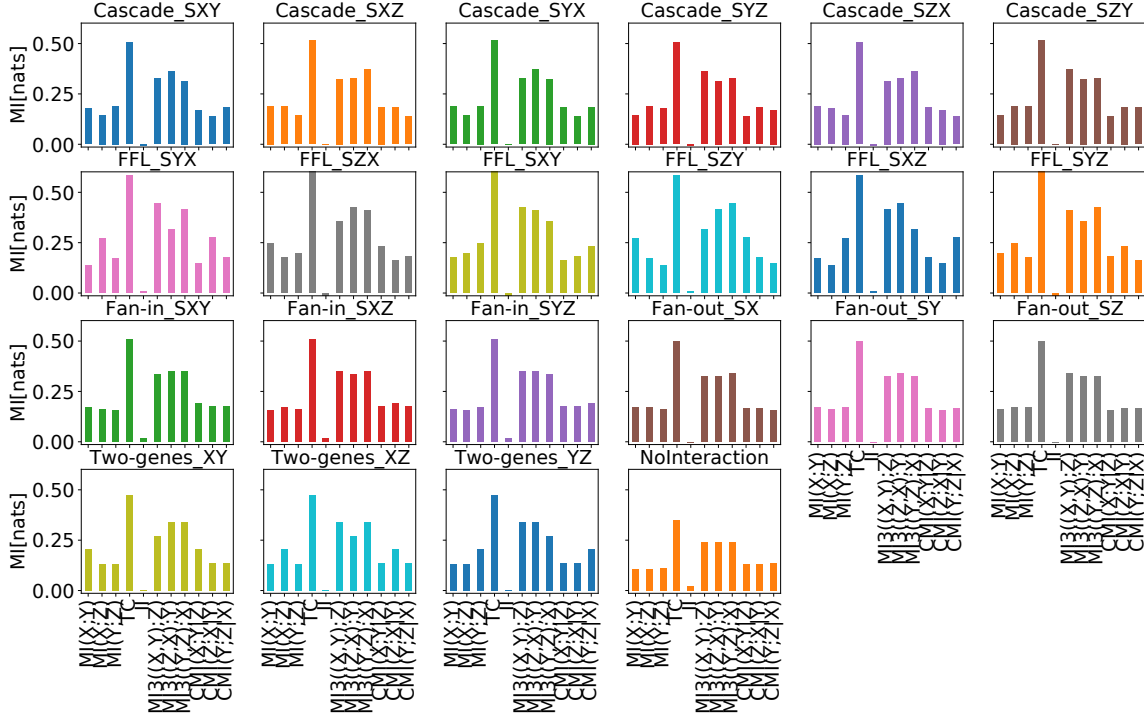


Figure 2-12. Mutual Information profiles of *E. coli* motifs

Each subplot represents a unique network motif. We show 22 motifs extracted from *E. coli* based on RegulonDB documented strong interactions, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the mean MI value in nats (information bits in *e* base).

Table 2-5. Machine Learning precision and accuracy for different network motifs

Dataset	E. coli		simulated	
	without Z-score	with Z-score	3 node	DREAM4
Best model	GBT	GBT	RF	GBT
3x Cascade	0.943-1	1	1	0.935-1
3x FFL	0.886-0.949	0.868-0.974	0.75-1	0.946-0.981
3x Fan-in	0.643-0.676	0.784-0.929	0.889-1	0.981-1
3x Fan-out	0.128-0.333	0.192-0.395	1	0.75-0.967
1x NoInteraction	-	0.16	0.833	0.673
3x Two-genes	0.139-0.184	0.122-0.297	1	0.774-0.9
Overall accuracy	0.577	0.631	0.975	0.912

Although the overall accuracy of classification in *E. coli* is significantly smaller than in the simulated 3-node or the DREAM-4 extracted motifs (0.631 v.s. 0.975 or 0.912,

respectively), there is a large variation in the classification precision of individual motifs. While Cascade, FFL and Fan-in show precision of $\sim 80-100\%$, which is in accordance with the precision obtained from *in-silico* data. Fan-out, No-interaction, and Two-genes motifs show precision of $\sim 10-40\%$, considerably lower than $\sim 70-100\%$ achieved with *in-silico* data. There could be a number of factors which contribute to this low precision. Qualitatively, this can be visualized by the mean MI profiles for individual motifs of the three datasets. For example, one factor could be experimental noise, as the two-genes motif in *in-silico* data (Fig. 2-2) shows close to zero MI levels for pairs that are not directly interacting (although we add experimental noise to the simulations), while in the real *E. coli* data (Fig. 2-12), there is a high background level and relatively small difference in MI between interacting and non-interacting genes (small signal-to-noise ratio). Another possible explanation is that the true network structure of *E. coli* is unknown, and our current view of existing interactions is certainly smaller than reality. We do not know for certain whether an extracted motif is “pure” or one of its component genes is being regulated by a source outside the triplet (a fourth gene). If this additional regulation is strong, it will bias the MI profile, making it more difficult to identify the correct motif.

Next, we checked whether using our models (SVM,MLP,RF,GBT) trained on DREAM-4 extracted motifs will improve the precision of classifying network motifs in real *E. coli* experimental data. To our surprise, all the models performed poorly, achieving accuracy in the range 0.044-0.078 on the same testing data used to test the models trained on real *E. coli* data.

Discussion

In general, majority of GRN inference algorithms do not focus on identifying small network motifs to build the global network from them, but rather rank pairwise interactions according to a single MI quantity (or statistics measure) with an arbitrary

threshold. This practice is known to suffer from systematic errors when reconstructing network motifs with steady-state gene expression data only [7].

Simulated motifs vs. motifs extracted from DREAM3-4 networks

As shown in Fig. 2-3,2-7,2-8, and Table 2-3 our MI profile method qualitatively enables classification of several 3-node motifs, but there are a few differences between the input data used in the simulated three-node motifs and the ones extracted from the 15 DREAM-3 & 4 networks, summarized in Table 2-6.

Table 2-6. Characteristics of *in-silico* input data

	Simulated 3 node motifs	Motifs extracted from DREAM networks
Network size	10	50-100
Replicates	100	10
Input data size (conditions)	1000	169-401
Steady-state- expression data type	Multifactorial perturbations	wildtype, knockout, dual-knockouts, knockdown, multifactorial
Multifactorial- perturbation distribution	Uniform[-1,1]	Gaussian[mean=0,std.=0.25]

Saturation of accuracy

For our 3-node simulated motifs we used 1000 perturbations for the gene expression data, as this is in the ballpark of publicly available *E. coli* data [44]. Yet, it is worth testing whether the accuracy saturates with a smaller data size, which will make our method more applicable to other organisms with less available data. Future work can calculate MI with 100, 250, 500 and 1000 perturbation respectively, and follow the pipeline outlined in this work.

Using more types of expression data

For our 3-node simulated motifs we only used multifactorial perturbation S.S. data in our study, omitting knockouts, knockdowns and time-series (T.S.), but this needs to be further explored as using more data types should improve the accuracy [28].

Z-score statistics

There are two ways to calculate Z-score, the standard score method,

$$Zscore_X = \left(\frac{MI(X;Y) - Mean(MI_X)}{STD(MI_X)} \right) \quad (2.9)$$

and the method implemented in Mrnet [29] and in Chapter 1,

$$Zscore_X = max \left(0, \frac{MI(X;Y) - Mean(MI_X)}{STD(MI_X)} \right) \quad (2.10)$$

We don't use the Z-score calculation directly, but rather through the likelihood estimate used by Faith et al. [13] (see Materials & Methods). For MI values larger than the mean the two methods give identical results, but Meyer et al. [29] method suppress the Z-score contribution from MI values smaller than the mean as it set them to zero. This gives slightly better overall results when reconstructing large GRN using Z-scores for positive MI values, as in CLR [13] and our method depicted in Chapter 1. Unfortunately, Interaction-Information can have both positive (synergy) or negative (redundancy) values, and using the later method results in bias when calculating Z-score for it. This means that motifs that have highly redundant interactions ($II < 0$) will show non-significant Z-score value for II . We need more analysis to determine the full extent of this bias on downstream processes.

Dimensionality reduction

Our PCA analysis shows that $\sim 98\%$ of the variance in the data can be captured by 4 principal components or $\sim 99.5\%$ with 5 components out of the 22 features (variables).

This shows high redundancy, and probably many variables can be omitted. To discover the most important real variables, we can repeat the PCA with different subsets of variables, until there is a considerable change in the explained variation per principal component.

There are two other common methods for visualizing high-dimensional space, namely, t-SNE [57] and UMAP [58]. Their main advantage over PCA is that they are not limited to only linear relationships and PCA's emphasize on inter-variable differences. While PCA's advantage is in its simplicity and the interpretability of the principal components. Comparing our results using t-SNE and UAMP is the subject of future work.

Classification by machine learning

To date, machine learning methods perform poorly (<6% precision) when inferring true two-way interactions using real *E. coli* experiemntal data [49, 59]. Here we are not trying to build a complete large network but rather classify small three-node motifs. We use mutual information quantities as a compact representation of the relationship between genes, instead of their full expression profiles under various conditions. Furthermore, we add the statistical significant of each 2d and 3d interaction by calculating their Z-score. For a network of 100 genes this allows for a significant reduction ($\sim \times 5$) in the number of variables (number of features, in ML language) for the machine learning algorithm and another $\times 3$ reduction in computation load as our observation unit is a triplet of genes vs. a single gene for conventional ML methods for GRN. This has two main advantages, first, a reduced model complexity which translates into faster training, second, this mitigate the problems arising when the number of features is larger than the number of observations which can lead to overfitting. Although we can further reduce the input data size by using five principal components (instead of 22) with minimal loss of variance information (<1% as shown

in the previous section), this had a large negative effect.

Surprisingly, using the principal components as input to train the machine learning models resulted in poor accuracy. This could be explained by the linear transformation done by PCA, which ignores the non-linear relationship between the various MI quantities. Perhaps, using the non-linear dimensionality reduction methods t-SNE or UAMP can lead to better performances when using reduced input data to train the models. But this will considerably increase computation costs in the data preparation step and might negate the benefit of training a machine learning model on reduced data.

We didn't test the widely used Linear Regression method, as it only gives a binary result, and for our purposes will require splitting the problem to multiple schemes of one-vs-all (as the number of motifs we want to classify) and combining the results to resolve any conflicts, which could be cumbersome. For MLP we only used a single hidden layer with different amount of neurons, and using more layers (deep learning) should improve accuracy, but further analysis is needed.

For the models trained on real *E. coli* data, we can try to improve the precision for Fan-out, Two-genes and No-interaction, by curating the input data. We can either use our MI profile method with different thresholds or alternatively, use our model to curate the existing database of documented motifs, as it might contain errors (missing interactions that can alter the motif identity). The idea is first to use the trained model to reclassify the poorly performing motifs and use only true-positives as inputs for second round of training. Luckily, there are thousands to millions of Fan-out, Two-genes and No-interaction motifs in *E. coli*, that we can filter down for a more curated training dataset.

Conclusions and general discussion

Gene regulation network inference using k-nearest neighbor-based mutual information estimation

We have shown that the kNN-based KSG MI estimator improves the performance of inference algorithms, especially ones that use three-way MI calculations. This result corroborates our observations in comparing MI calculations against the analytical solution of two-way MI of a bi-variate Gaussian distribution and the total correlation of a tri-variate Gaussian distribution. Furthermore, the combination of CMIA and KSG give the overall best performance, and hence should be preferred when precision and recall are more important than speed when reconstructing a GRN. Looking forward, the goal of complete reconstruction of GRNs may require new inference algorithms and probably MI in more than three dimensions.

Classifying three-node network motifs of Transcription Factor (TF)-based regulation

We have developed and compared three methods to classify three-node motifs:

1. MI and Z-score profiles
2. Dimensionality reduction by PCA and clustering using K-means
3. Supervised machine learning algorithms using MI input data

We have shown that at least 24 different 3-node motifs *in-silico* and 16 motifs on

E. coli experimental data can be distinguished by using all 2d and 3d MI quantities together of only steady-state expression data and without any *a priori* knowledge of the regulator (source) genes. This unprecedented resolution can assist in a more accurate large GRN reconstruction of any model organism by assembling the entire network from the bottom up and mitigate the problem of false positives in co-regulated genes with no direct interaction (Fan-out) or for indirect interaction (Cascade). It will be interesting to run our pipeline on the millions of triplets that has no documented interactions in *E. coli* or other model organisms, or even on triplets with existing documented interactions to check if the model can detect more regulators.

Our MI based method can also be used as a fast first-order approach, incorporating cases of ambiguity about the direction of regulation, such as the FFL motif, into separate entire network topologies. The ensemble of different network topologies can then be further tuned by adding more “expensive” experimental data (knockouts, time-series) or using it as input to other methods to rank the different network realizations from most probable to least probable.

Another possible application is to use all two- and three-way MI quantities (total of 22 with their Z-score counterparts) together with a Hidden Markov Model (HMM) to divide MI values to few discrete states which will allow us to generate a digital signature (or “barcode”) to most common 3-node network topologies.

References

1. Alberts, B. *et al.* *Molecular Biology of the Cell* (W.W. Norton & Company, Dec. 2007).
2. Cordero, D. *et al.* Large differences in global transcriptional regulatory programs of normal and tumor colon cells. *BMC Cancer* **14**, 1–13 (Sept. 2014).
3. Bashor, C. J. & Collins, J. J. Understanding Biological Regulation Through Synthetic Biology. *Annual review of biophysics* **47**, 399–423 (May 2018).
4. Eisen, M. B., Spellman, P. T., Brown, P. O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* **95** (1998).
5. Courcelle, J., Khodursky, A., Peter, B., Brown, P. O. & Hanawalt, P. C. Comparative Gene Expression Profiles Following UV Exposure in Wild-Type and SOS-Deficient *Escherichia coli*. *Genetics* **158**, 41–64 (May 2001).
6. Bansal, M., Belcastro, V., Ambesi-Impiombato, A. & Di Bernardo, D. How to infer gene networks from expression profiles. *Molecular Systems Biology* **3**, 1–10 (2007).
7. Marbach, D. *et al.* Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences of the United States of America* **107**, 6286–6291 (2010).
8. Maetschke, S. R., Madhamshettiwar, P. B., Davis, M. J. & Ragan, M. A. Supervised, semi-supervised and unsupervised inference of gene regulatory networks. *Briefings in Bioinformatics* **15**, 195–211. arXiv: [1301.1083](https://arxiv.org/abs/1301.1083) (2014).
9. Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal* **27**, 379–423 (July 1948).
10. Shannon, C. E. A Mathematical Theory of Communication. *Bell System Technical Journal* **27**, 623–656 (1948).
11. Butte, A. J. & Kohane, I. S. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* **426**, 418–429 (2000).
12. Margolin, A. A. *et al.* ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* **7**, 1–15. arXiv: [0410037](https://arxiv.org/abs/0410037) [q-bio] (2006).
13. Faith, J. J. *et al.* Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biology* **5**, 0054–0066 (2007).

14. Cover, T. M. & Thomas, J. A. *Elements of Information Theory* 1–748 (John Wiley and Sons, Apr. 2005).
15. Luo, W., Hankenson, K. D. & Woolf, P. J. Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information. *BMC Bioinformatics* **9**, 1–15 (2008).
16. Timme, N., Alford, W., Flecker, B. & Beggs, J. M. Synergy, redundancy, and multivariate information measures: An experimentalist’s perspective. *Journal of Computational Neuroscience* **36**, 119–140 (2014).
17. Liang, K.-C. & Wang, X. Gene Regulatory Network Reconstruction Using Conditional Mutual Information. *EURASIP Journal on Bioinformatics and Systems Biology* **2008**, 1–14 (2008).
18. Watkinson, J., Liang, K.-C., Wang, X., Zheng, T. & Anastassiou, D. Inference of regulatory gene interactions from expression data using three-way mutual information. *Annals of the New York Academy of Sciences* **1158**, 302–313 (2009).
19. Mousavian, Z., Kavousi, K. & Masoudi-Nejad, A. Information theory in systems biology. Part I: Gene regulatory and metabolic networks. *Seminars in Cell and Developmental Biology* **51**, 3–13 (2016).
20. Ross, B. C. Mutual information between discrete and continuous data sets. *PLoS ONE* **9** (2014).
21. Miller, G. A. & Madow, W. G. *On the maximum likelihood estimate of the Shannon-Weiner measure of information* (Operational Applications Laboratory, Air Force Cambridge Research Center ..., 1954).
22. Darbellay, G. A. & Vajda, I. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory* **45**, 1315–1321 (1999).
23. Kraskov, A., Stögbauer, H. & Grassberger, P. Estimating mutual information. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* **69**, 16 (2004).
24. Daub, C. O., Steuer, R., Selbig, J. & Kloska, S. Estimating mutual information using B-spline functions – an improved similarity measure for analysing gene expression data. **12**, 1–12 (2004).
25. Sales, G. & Romualdi, C. Parmigene-a parallel R package for mutual information estimation and gene network reconstruction. *Bioinformatics* **27**, 1876–1877 (2011).
26. Kozachenko, L. F. & Leonenko, N. N. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii* **23**, 9–16 (1987).
27. Prill, R. J. *et al.* Towards a rigorous assessment of systems biology models: The DREAM3 challenges. *PLoS ONE* **5** (2010).
28. Schaffter, T., Marbach, D. & Floreano, D. GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* **27**, 2263–2270 (2011).
29. Meyer, P. E., Lafitte, F. & Bontempi, G. Minet: A r/bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics* **9**, 1–10 (2008).

30. Murphy, K. P. *Machine learning: a probabilistic perspective (adaptive computation and machine learning series)* (2012).
31. Kurths, J., Daub, C. O., Weise, J., Selbig, J. & Steuer. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics* **18 Suppl 2**, S231–40 (2002).
32. Marbach, D., Schaffter, T., Mattiussi, C. & Floreano, D. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology* **16**, 229–239 (2009).
33. Alon, U. *An introduction to systems biology: Design principles of biological circuits* (2006).
34. *Scipy spatial algorithms*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html>. Accessed: 2021-10-28.
35. Chan, T. E., Stumpf, M. P. & Babbie, A. C. Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures. *Cell Systems* **5**, 251–267.e3 (2017).
36. Marbach, D. *et al.* Wisdom of crowds for robust gene network inference. *Nature Methods* **9**, 796–804 (2012).
37. Darbellay, G. A. & Vajda, I. Entropy expressions for multivariate continuous distributions. *IEEE Transactions on Information Theory* **46**, 709–712 (2000).
38. Scargle, J. D., Norris, J. P., Jackson, B. & Chiang, J. Studies in astronomical time series analysis. VI. Bayesian block representations. *Astrophysical Journal* **764**. arXiv: [1207.5578](https://arxiv.org/abs/1207.5578) (2013).
39. Dorman, C. J. & Dorman, M. J. DNA supercoiling is a fundamental regulatory principle in the control of bacterial gene expression. *Biophysical Reviews* **8**, 209–220 (2016).
40. Bintu, L. *et al.* Transcriptional regulation by the numbers: models. *Current Opinion in Genetics & Development* **15**, 116–124 (2 Apr. 2005).
41. Bohrer, C. H. & Roberts, E. A biophysical model of supercoiling dependent transcription predicts a structural aspect to gene regulation. *BMC Biophysics* **9**, 1–13 (2016).
42. Geng, Y. *et al.* A spatially resolved stochastic model reveals the role of supercoiling in transcription regulation. *bioRxiv*. eprint: <https://www.biorxiv.org/content/early/2021/12/29/2021.12.29.474406.full.pdf> (2021).
43. Mc Mahon, S. S. *et al.* Information theory and signal transduction systems: From molecular information processing to network inference. *Seminars in Cell and Developmental Biology* **35**, 98–108 (2014).
44. Faith, J. J. *et al.* Many Microbe Microarrays Database: Uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Research* **36**, 866–870 (2008).
45. Santos-Zavaleta, A. *et al.* RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in *E. coli* K-12. *Nucleic Acids Research* **47**, D212–D220 (Jan. 2019).
46. Abdi, H. & Williams, L. J. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 433–459 (2010).

47. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).
48. Lloyd, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**, 129–137 (Mar. 1982).
49. Huynh-Thu, V. A., Irrthum, A., Wehenkel, L. & Geurts, P. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE* **5**, 1–10 (2010).
50. Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B. & Rätsch, G. Support vector machines and kernels for computational biology. *PLoS Computational Biology* **4** (2008).
51. Lancashire, L. J., Lemetre, C. & Ball, G. R. An introduction to artificial neural networks in bioinformatics - Application to complex microarray and mass spectrometry datasets in cancer studies. *Briefings in Bioinformatics* **10**, 315–329 (2009).
52. Geurts, P., Irrthum, A. & Wehenkel, L. Supervised learning with decision tree-based methods in computational and systems biology. *Molecular BioSystems* **5**, 1593–1605 (2009).
53. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* **29**, 1189–1232 (2001).
54. Chen, T. & Guestrin, C. *XGBoost: A scalable tree boosting system* in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 13-17-August-2016* (2016).
55. Tyson, J. J. & Novák, B. Functional Motifs in Biochemical Reaction Networks. *Annual review of physical chemistry* **61**, 219 (May 2010).
56. Elowitz, M. B. & Leibler, S. A synthetic oscillatory network of transcriptional regulators, 335–338 (1999).
57. Van Der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9**, 2579–2605 (2008).
58. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv: [1802.03426](https://arxiv.org/abs/1802.03426) (2018).
59. Zhao, M., He, W., Tang, J., Zou, Q. & Guo, F. A comprehensive overview and critical evaluation of gene regulatory network inference technologies. *Briefings in Bioinformatics* **00**, 1–15 (2021).
60. McGill, W. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory* **4**, 93–111 (Sept. 1954).
61. Watanabe, S. Information Theoretical Analysis of Multivariate Correlation. *IBM Journal of Research and Development* **4**, 66–82 (Jan. 1960).

Appendix I

Mutual Information overview

Introduction to Information Theory

In present days, Information Theory is being widely used in many fields very far from what its originator Claude Shannon had probably conceived when he published his work in the late 40s as a complete theory for digital communication [9, 10]. Its main use outside the world of communication is to determine the "similarity" between different sets of variables and help determine if there is any correlation between them whether it is linear or non-linear. Before diving into its full mathematical representation, let us start with building our intuition about information theory using some simple terms and a Venn diagram.

Shannon's Entropy

We begin with a discrete random variable X (i.e. $X \equiv \{x_1, x_2, \dots, x_n\}$), this could be the outcomes of flipping a coin multiple times or rolling a dice. Shannon [9] defined the "uncertainty" of X as the Entropy of X , $H(X)$. Entropy is a non-negative quantity ($H(X) \geq 0$), it is maximal where all possible outcomes (more than one) have the same probability (in other words, a uniform probability distribution function). For example, flipping a fair coin or rolling a dice (the entropy is maximal but not equal for those two cases, and we will calculate it in a later section). On the other hand, if our variable X has only one possible value than there is no uncertainty and the entropy equals zero. In some places, entropy is also referred to as "self-information" (see below section on Mutual Information).

Conditional and Joint Entropy

Following the same logic, the joint entropy $H(X, Y)$, is defined as the uncertainty of the pair X , Y . And the conditional entropy $H(X|Y)$, is the uncertainty of X given Y . We can represent their relationship in the following formula (mathematical proofs can be found in [14]):

$$\begin{aligned} H(X, Y) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned} \tag{I.1}$$

We can extend the above relationship to 3 variables:

$$\begin{aligned} H(X, Y, Z) &= H(X) + H(Y, Z|X) \\ &= H(X) + H(Y|X) + H(Z|Y, X) \end{aligned} \tag{I.2}$$

For n variables (Chain rule for entropy) [14]:

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i | X_{i-1}, \dots, X_1) \quad (\text{I.3})$$

Mutual Information

We can now define the Mutual Information (MI) shared by X and Y as:

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned} \quad (\text{I.4})$$

This is also referred to as "information redundancy" or "Reduction of uncertainty" of X thanks to Y (or vice versa). MI is a symmetric ($I(X; Y) = I(Y; X)$) and non-negative ($I(X; Y) \geq 0$) quantity. It is zero only if X and Y are independent ($H(X, Y) = H(X) + H(Y)$). We also get that self-information equals the entropy ($I(X; X) = H(X)$). Summarizing the above, we get a range for MI: $0 \leq I(X; Y) \leq \max\{H(X), H(Y)\}$.

Three-Dimensional Mutual Information

In 1954, McGill [60] extended Shannon's work to the case of two sources $\{X_1, X_2\}$ and one receiver Y (or vice versa), by simply changing X in Eq. (I.4) with $\{X_1, X_2\}$:

$$\begin{aligned} I(X_1, X_2; Y) &= H(X_1, X_2) - H(X_1, X_2|Y) \\ &= H(X_1, X_2) + H(Y) - H(X_1, X_2, Y) \end{aligned} \quad (\text{I.5})$$

Naturally, this can be extended to n variables, where we can measure the mutual information between a group of $n - 1$ variables (treated as a single vector) and a target. This however, doesn't allow us to evaluate the individual gain (or loss) of information by each individual source, but this can be solved by comparing calculations with different number of sources.

Interaction Information

McGill also defined the Interaction-Information (II), which is a symmetric quantity:

$$\begin{aligned} II(X_1 \cdot X_2 \cdot Y) &= I(X_1; Y|X_2) - I(X_1; Y) \\ &= I(X_2; Y|X_1) - I(X_2; Y) \\ &= I(X_1; X_2|Y) - I(X_1; X_2) \end{aligned} \quad (\text{I.6})$$

Where the Conditional Mutual Information (CMI) of X_1 and X_2 given Y is defined by:

$$\begin{aligned} I(X_1; X_2|Y) &= H(X_1|Y) - H(X_1|X_2, Y) \\ &= -H(Y) + (H(X_1, Y) + H(X_2, Y)) - H(X_1, X_2, Y) \end{aligned} \quad (\text{I.7})$$

It is important to note, that the RHS of Eq.(I.7) represent only two out of many possible combinations of entropy terms. We can also write the interaction information as an expansion of entropy terms:

$$\begin{aligned} II(X_1 \cdot X_2 \cdot Y) &= -H(X_1) - H(X_2) - H(Y) \\ &\quad + H(X_1, X_2) + H(X_1, Y) + H(X_2, Y) \\ &\quad - H(X_1, X_2, Y) \end{aligned} \quad (\text{I.8})$$

We can now write the three-dimensional mutual information using the two-dimensional MI and the interaction-information:

$$I(X_1, X_2; Y) = I(X_1; Y) + I(X_2; Y) + II(X_1 \cdot X_2 \cdot Y) \quad (\text{I.9})$$

$$I(X_1, X_2; Y) = I(X_1; Y|X_2) + I(X_2; Y|X_1) - II(X_1 \cdot X_2 \cdot Y) \quad (\text{I.10})$$

We can plug Eq.(I.6) into the three-dimensional MI equation above Eq.(I.9), to get:

$$\begin{aligned} I(X_1, X_2; Y) &= I(X_2; Y) + I(X_1; Y|X_2) \\ &= I(X_1; Y) + I(X_2; Y|X_1) \\ &= I(X_1; Y) + I(X_2; Y) + I(X_1; X_2|Y) - I(X_1; X_2) \end{aligned} \quad (\text{I.11})$$

We can expand the MI to n sources (Chain rule for information [14]):

$$\begin{aligned} I(X_1, X_2, \dots, X_n; Y) &= H(X_1, X_2, \dots, X_n) - H(X_1, X_2, \dots, X_n|Y) \\ &= \sum_{i=1}^n I(X_i; Y|X_{i-1}, \dots, X_1) \end{aligned} \quad (\text{I.12})$$

Total Correlation

Total Correlation (TC) is another frequently used term (also referred sometime as redundancy or MI¹) that was first shown by McGill [60], but was coined and further developed by Watanabe in 1960 [61].

$$TC(X_1, X_2, \dots, X_n) = \sum_i H(X_i) - H(X_1, X_2, \dots, X_n) \quad (\text{I.13})$$

By adding and subtracting the same joint entropy terms, we can rewrite the TC using MI terms (see Appendix A of [16]):

$$TC(X_1, X_2, \dots, X_n) = I(X_1; X_2) + I(X_1, X_2; X_3) + \dots + I(X_1, \dots, X_{n-1}; X_n) \quad (\text{I.14})$$

Uniqueness, Redundancy and Synergy

"Redundancy" and "Synergy" are common terms in the field of information theory, yet they lack common definition, and so create a lot of confusion as different definitions exist. We can start discussing their meaning when looking into the relationship between three variables or more (i.e. two sources X_1, X_2 and a target Y). In the most intuitive way, we can define redundancy as the portion of information both X_1 and X_2 share in common about Y , and synergy as information we gain (or emerges) about Y from inspecting X_1 and X_2 together, rather than separately. Following the same line of thought "Uniqueness" can be viewed as the information only X_1 brings about Y or only what X_2 brings about Y . Using the terms we defined in the previous section, we can write: CMI = Uniqueness, II = Redundancy if $II < 0$, and Synergy if $II > 0$.

Formalism for Discrete Variables

Probability definitions

For variables X and Y , we can construct a space $X - Y$ where each point corresponds to each pair $\{x, y\}$. We can generate any ensemble $X - Y$ by assigning a joint probability $P(x, y)$. Where

$$\sum_X \sum_Y P(x, y) = 1 \quad (\text{I.15})$$

¹This confusion is mainly due to the fact that in $2D$ they are all expressed the same but for higher dimensions ($n > 2$) they are different

The probability distribution $P(x)$ (also called "marginal") can be defined in terms of $P(x, y)$ by

$$P(x) \equiv \sum_Y P(x, y) \quad (\text{I.16})$$

The conditional probability distribution $p(y|x)$ is define as

$$P(y|x) \equiv \frac{P(x, y)}{P(x)} \quad (\text{I.17})$$

For three variables, we can define the conditional probability distribution $P(x|y, z)$ as

$$P(x|y, z) \equiv \frac{P(x, y, z)}{P(p, z)} \quad (\text{I.18})$$

If $P(x|y, z)$ is independent of any pair y, z ($P(x|y, z) = P(x)$) than X is independent of YZ and we can write

$$P(x, y, z) = P(x)P(y, z) \quad (\text{I.19})$$

Entropy

Shannon's Entropy:

$$H(X) = - \sum_x p(x) \log p(x)$$

Joint Entropy:

$$H(X, Y) = - \sum_x \sum_y p(x, y) \log p(x, y)$$

Conditional Entropy:

$$H(X|Y = y) = - \sum_x p(x|y) \log p(x|y)$$

$$H(X|Y) = \sum_y p(y) H(X|Y = y) = \sum_y p(y) \sum_x p(x|y) \log \frac{1}{p(x|y)}$$

Information

Infomration provided by y_i about x_k is defined by:

$$I(x_k; y_i) \equiv \log \frac{P(x_k|y_i)}{P(x_k)} = \log \frac{P(x_k|y_i)P(y_i)}{P(x_k)P(y_i)} = \log \frac{P(x_k, y_i)}{P(x_k)P(y_i)} \quad (\text{I.20})$$

This can be positive or negative, depending on the probability of occuring together vs. separately.

Mutual Information (MI):

$$I(X; Y) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

Conditional Mutual Information (CMI):

$$I(X; Y|Z) = \sum_z p(z) \sum_x \sum_y p(x, y|z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)}$$

Appendix II

Supplementary information for chapter on gene regulation network inference

A. Analytical solution for a multivariate Gaussian distribution

Shannon [9] showed that the entropy term of a multivariate Gaussian distribution is given by:

$$H(X) = \frac{1}{2} \log[(2\pi e)^n |COV|] \quad (\text{II.1})$$

Where $|COV|$ represents the covariance matrix. For simplicity we set all the correlations between variables to be equal to ρ . As all MI quantities can be calculated by their entropy components, we have:

$$MI(X; Y) = H(X) + H(Y) - H(X, Y) = -\frac{1}{2} \log(1 - \rho^2) \quad (\text{II.2})$$

$$TC(X, Y, Z) = H(X) + H(Y) + H(Z) - H(X, Y, Z) = -\frac{1}{2} \log(1 - 3 \cdot \rho^2 + 2 \cdot \rho^3) \quad (\text{II.3})$$

B. Miller-Madow correction to Shannon's entropy

Due to the logarithmic nature of Shannon's entropy:

$$H^{Shan} = - \sum_x p(x) \log p(x) \quad (\text{II.4})$$

Under or overestimating $p(x)$ by the same value gives different errors on the entropy calculation, leading to bias (downwards). Miller and Madow proposed to correct the bias in Shannon's entropy by adding the asymptotic bias term [21]:

$$H^{MM} = H^{Shan} + \frac{\{non_empty_bins\} - 1}{2N} \quad (\text{II.5})$$

Where N is equal to the data size.

Two-way mutual information and higher dimension measures can be calculated by summation of entropies, for example in the case of two-way MI:

$$MI^{MM}(X;Y) = H^{MM}(X) + H^{MM}(Y) - H^{MM}(X,Y) \quad (\text{II.6})$$

C. Supplementary tables

Table II-1. Median AUPR values for different combinations of MI estimator and GRN inference algorithm for different network sizes.

Data set	Network Size	Infalگو	MIest	median_AUPR	AUPR_relative
DREAM3	50	ARACNE	KL	0.013	-95.7
DREAM3	50	ARACNE	KSG	0.136	-46.6
DREAM3	50	ARACNE	ML	0.068	-72.9
DREAM3	50	ARACNE	MM	0.088	-64.0
DREAM3	50	CLR	KL	0.092	-64.4
DREAM3	50	CLR	KSG	0.264	7.7
DREAM3	50	CLR	ML	0.239	0.0
DREAM3	50	CLR	MM	0.253	7.4
DREAM3	50	CMI2rt	KL	0.008	-97.1
DREAM3	50	CMI2rt	KSG	0.068	-72.6
DREAM3	50	CMI2rt	ML	0.010	-96.2
DREAM3	50	CMI2rt	MM	0.013	-95.0
DREAM3	50	CMIA	KL	0.092	-62.8
DREAM3	50	CMIA	KSG	0.285	16.0
DREAM3	50	CMIA	ML	0.221	-17.5
DREAM3	50	CMIA	MM	0.225	-10.3
DREAM3	50	RL	KL	0.021	-93.7
DREAM3	50	RL	KSG	0.246	-2.0
DREAM3	50	RL	ML	0.206	-23.5
DREAM3	50	RL	MM	0.232	-10.7
DREAM3	50	SA_CLR	KL	0.084	-65.0
DREAM3	50	SA_CLR	KSG	0.290	15.9
DREAM3	50	SA_CLR	ML	0.188	-36.8
DREAM3	50	SA_CLR	MM	0.189	-33.6
DREAM3	100	ARACNE	KL	0.018	-89.4
DREAM3	100	ARACNE	KSG	0.103	-50.5
DREAM3	100	ARACNE	ML	0.051	-75.2
DREAM3	100	ARACNE	MM	0.068	-64.5
DREAM3	100	CLR	KL	0.062	-71.6
DREAM3	100	CLR	KSG	0.246	10.4
DREAM3	100	CLR	ML	0.215	0.0
DREAM3	100	CLR	MM	0.231	17.3
DREAM3	100	CMI2rt	KL	0.014	-94.0
DREAM3	100	CMI2rt	KSG	0.051	-76.0
DREAM3	100	CMI2rt	ML	0.002	-99.2
DREAM3	100	CMI2rt	MM	0.004	-97.9
DREAM3	100	CMIA	KL	0.073	-66.5
DREAM3	100	CMIA	KSG	0.261	22.6

Continued on next page

Table II-1 – *Continued from previous page*

Data set	Network Size	Infalگو	MIest	median_AUPR	AUPR_relative
DREAM3	100	CMIA	ML	0.142	-35.3
DREAM3	100	CMIA	MM	0.157	-28.8
DREAM3	100	RL	KL	0.030	-84.6
DREAM3	100	RL	KSG	0.220	0.6
DREAM3	100	RL	ML	0.105	-46.8
DREAM3	100	RL	MM	0.138	-29.1
DREAM3	100	SA_CLR	KL	0.070	-66.2
DREAM3	100	SA_CLR	KSG	0.259	20.1
DREAM3	100	SA_CLR	ML	0.073	-57.0
DREAM3	100	SA_CLR	MM	0.077	-56.5
DREAM4	100	ARACNE	KL	0.002	-99.1
DREAM4	100	ARACNE	KSG	0.129	-43.1
DREAM4	100	ARACNE	ML	0.080	-68.3
DREAM4	100	ARACNE	MM	0.103	-58.5
DREAM4	100	CLR	KL	0.030	-87.9
DREAM4	100	CLR	KSG	0.260	17.8
DREAM4	100	CLR	ML	0.232	0.0
DREAM4	100	CLR	MM	0.262	15.5
DREAM4	100	CMI2rt	KL	0.001	-99.4
DREAM4	100	CMI2rt	KSG	0.086	-64.4
DREAM4	100	CMI2rt	ML	0.001	-99.5
DREAM4	100	CMI2rt	MM	0.002	-99.1
DREAM4	100	CMIA	KL	0.031	-87.3
DREAM4	100	CMIA	KSG	0.302	33.6
DREAM4	100	CMIA	ML	0.144	-40.1
DREAM4	100	CMIA	MM	0.169	-30.5
DREAM4	100	RL	KL	0.002	-99.0
DREAM4	100	RL	KSG	0.226	-10.0
DREAM4	100	RL	ML	0.159	-38.7
DREAM4	100	RL	MM	0.191	-24.0
DREAM4	100	SA_CLR	KL	0.031	-87.3
DREAM4	100	SA_CLR	KSG	0.301	30.2
DREAM4	100	SA_CLR	ML	0.076	-67.0
DREAM4	100	SA_CLR	MM	0.082	-65.2

Table II-2. Median AUPR values for different combinations of MI estimator and GRN inference algorithm for different organisms.

Organism	Infalگو	MIest	median_AUPR	AUPR_relative
Ecoli	ARACNE	KL	0.017	-88.8
Ecoli	ARACNE	KSG	0.077	-52.5
Ecoli	ARACNE	ML	0.033	-79.8
Ecoli	ARACNE	MM	0.054	-66.5
Ecoli	CLR	KL	0.043	-70.0
Ecoli	CLR	KSG	0.159	5.9
Ecoli	CLR	ML	0.168	0.0
Ecoli	CLR	MM	0.197	19.9
Ecoli	CMI2rt	KL	0.008	-94.3
Ecoli	CMI2rt	KSG	0.032	-79.6
Ecoli	CMI2rt	ML	0.003	-98.6

Continued on next page

Table II-2 – *Continued from previous page*

Organism	Infalgo	MIest	median_AUPR	AUPR_relative
Ecoli	CMI2rt	MM	0.003	-98.0
Ecoli	CMIA	KL	0.051	-66.5
Ecoli	CMIA	KSG	0.191	19.7
Ecoli	CMIA	ML	0.115	-33.4
Ecoli	CMIA	MM	0.133	-21.9
Ecoli	RL	KL	0.034	-76.5
Ecoli	RL	KSG	0.157	-2.0
Ecoli	RL	ML	0.096	-38.4
Ecoli	RL	MM	0.124	-21.7
Ecoli	SA_CLR	KL	0.055	-66.5
Ecoli	SA_CLR	KSG	0.193	19.7
Ecoli	SA_CLR	ML	0.081	-53.1
Ecoli	SA_CLR	MM	0.086	-51.5
Yeast	ARACNE	KL	0.016	-94.7
Yeast	ARACNE	KSG	0.154	-46.6
Yeast	ARACNE	ML	0.077	-72.2
Yeast	ARACNE	MM	0.101	-63.9
Yeast	CLR	KL	0.096	-63.6
Yeast	CLR	KSG	0.291	10.5
Yeast	CLR	ML	0.269	0.0
Yeast	CLR	MM	0.286	8.7
Yeast	CMI2rt	KL	0.012	-95.5
Yeast	CMI2rt	KSG	0.095	-68.5
Yeast	CMI2rt	ML	0.010	-96.2
Yeast	CMI2rt	MM	0.013	-95.0
Yeast	CMIA	KL	0.101	-63.0
Yeast	CMIA	KSG	0.328	18.4
Yeast	CMIA	ML	0.224	-18.1
Yeast	CMIA	MM	0.236	-14.5
Yeast	RL	KL	0.024	-92.4
Yeast	RL	KSG	0.252	0.0
Yeast	RL	ML	0.196	-27.8
Yeast	RL	MM	0.219	-14.1
Yeast	SA_CLR	KL	0.098	-64.1
Yeast	SA_CLR	KSG	0.323	18.0
Yeast	SA_CLR	ML	0.174	-37.9
Yeast	SA_CLR	MM	0.177	-36.5

Table II-3. Characteristics of the 10 synthetic networks from DREAM3 and statistics of the different 3-node network motifs extracted.

Network	SS data	Edges	Triplets	No Interaction	Two-genes	Fan-in	Cascade	Fan-out	FFL	Sum of 2 edges	Sum of 2&3 edges
InSilicoSize100-Ecoli1	341	125	161700	150051	11059	47	55	477	11	579	590
InSilicoSize100-Ecoli2	322	119	161700	150759	10228	24	51	630	8	705	713
InSilicoSize100-Yeast1	401	166	161700	146042	15113	75	212	193	65	480	545
InSilicoSize100-Yeast2	401	389	161700	127499	30631	627	1231	1361	351	3219	3570
InSilicoSize100-Yeast3	401	551	161700	115759	39003	1385	2052	2382	1119	5819	6938
InSilicoSize50-Ecoli1	170	62	19600	16936	2361	21	41	232	9	294	303
InSilicoSize50-Ecoli2	169	82	19600	16230	2816	47	20	475	12	542	554
InSilicoSize50-Yeast1	201	77	19600	16204	3126	43	103	94	30	240	270
InSilicoSize50-Yeast2	201	160	19600	13056	5536	241	306	333	128	880	1008
InSilicoSize50-Yeast3	201	173	19600	12629	5812	195	303	487	174	985	1159

D. Supplementary figures

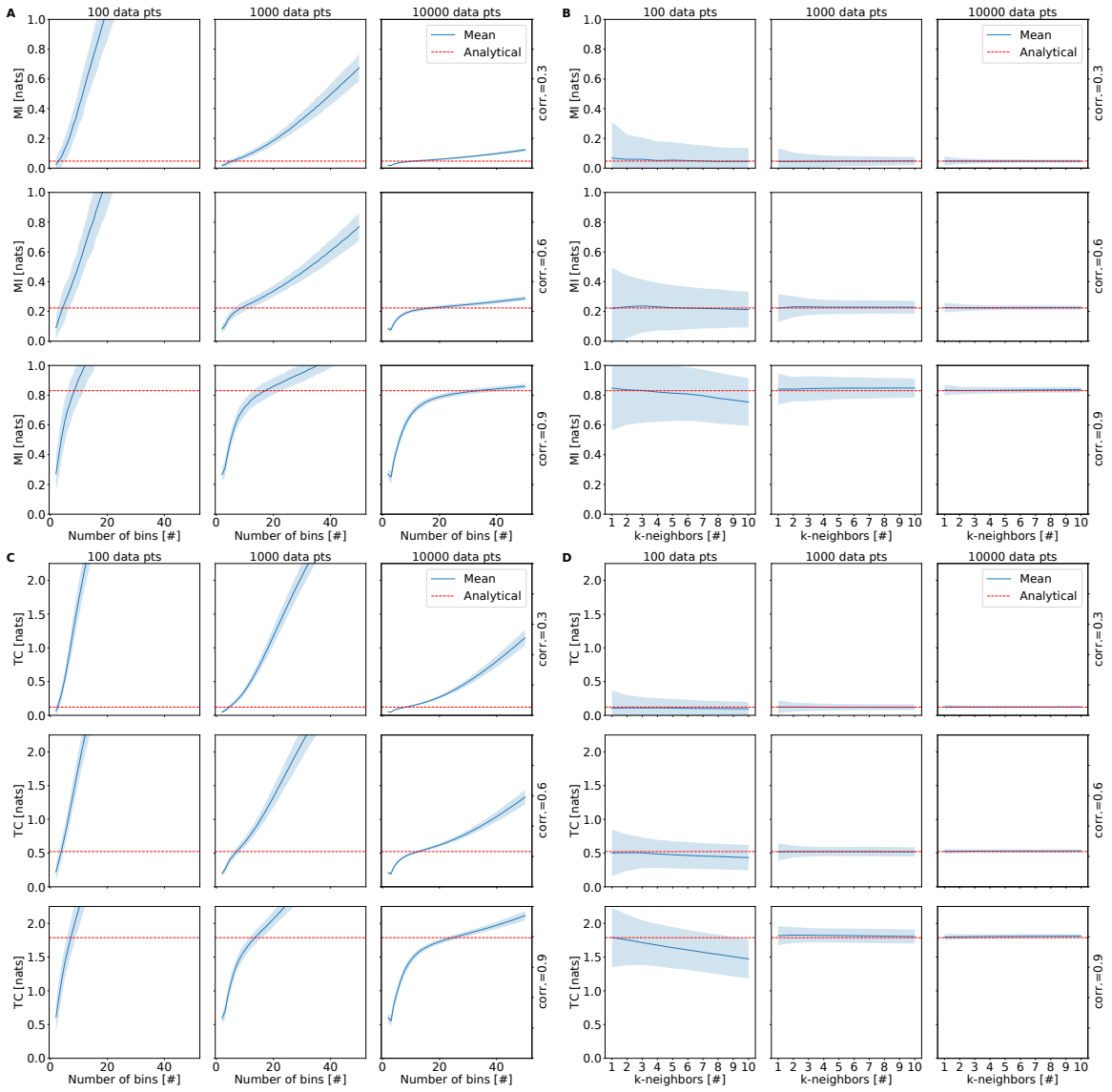


Figure II-1. Two-way mutual information (MI2) & total correlation (TC) for multivariate gaussian dist. with varying bins and neighbors

100 replicates for each sample size = {100,1K,10K}, correlation = {0.3,0.6,0.9}. (A) MI2 with natural log base calculated using Maximum Likelihood with fixed width binning (FB), where the shaded area represents mean +/- 2std. (B) MI2 based on KSG k-nearest-neighbor (KNN). (C) TC based on FB. (D) TC based on kNN

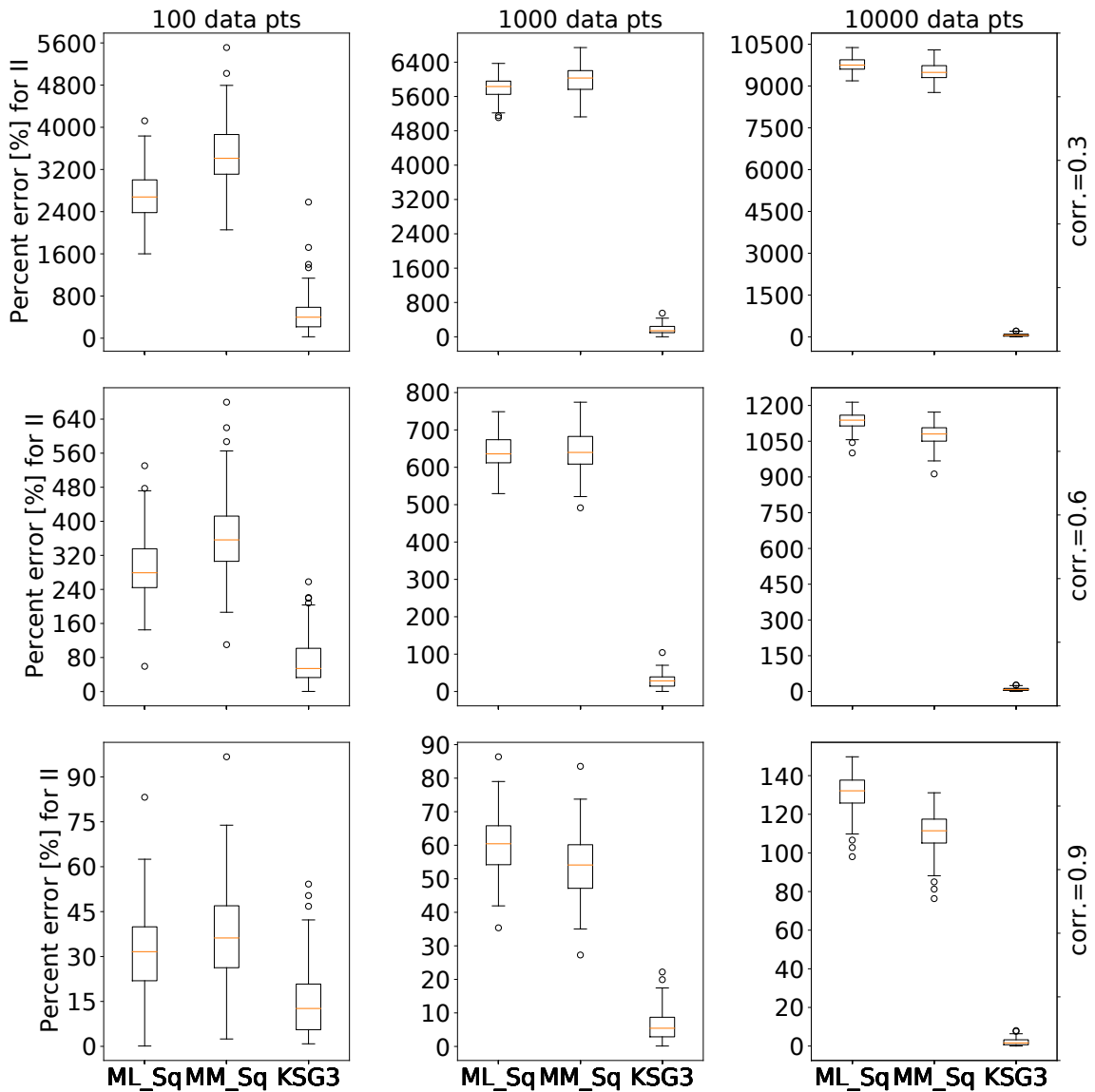


Figure II-2. Boxplots of percent error for Interaction Information (II) of three different mutual information estimators

With columns representing sample size = {100,1K,10K}, and rows the correlation = {0.3,0.6,0.9}. 9 subplots showing percent error for Interaction Information (II) for 3 different methods: ML_Sq=Shannon's MI with fixed width binning (number of bins is determined by square-root), MM_Sq=Miller-Madow formula for MI with square-root for the number of bins, kNN3=KSG formula for MI with k=3.

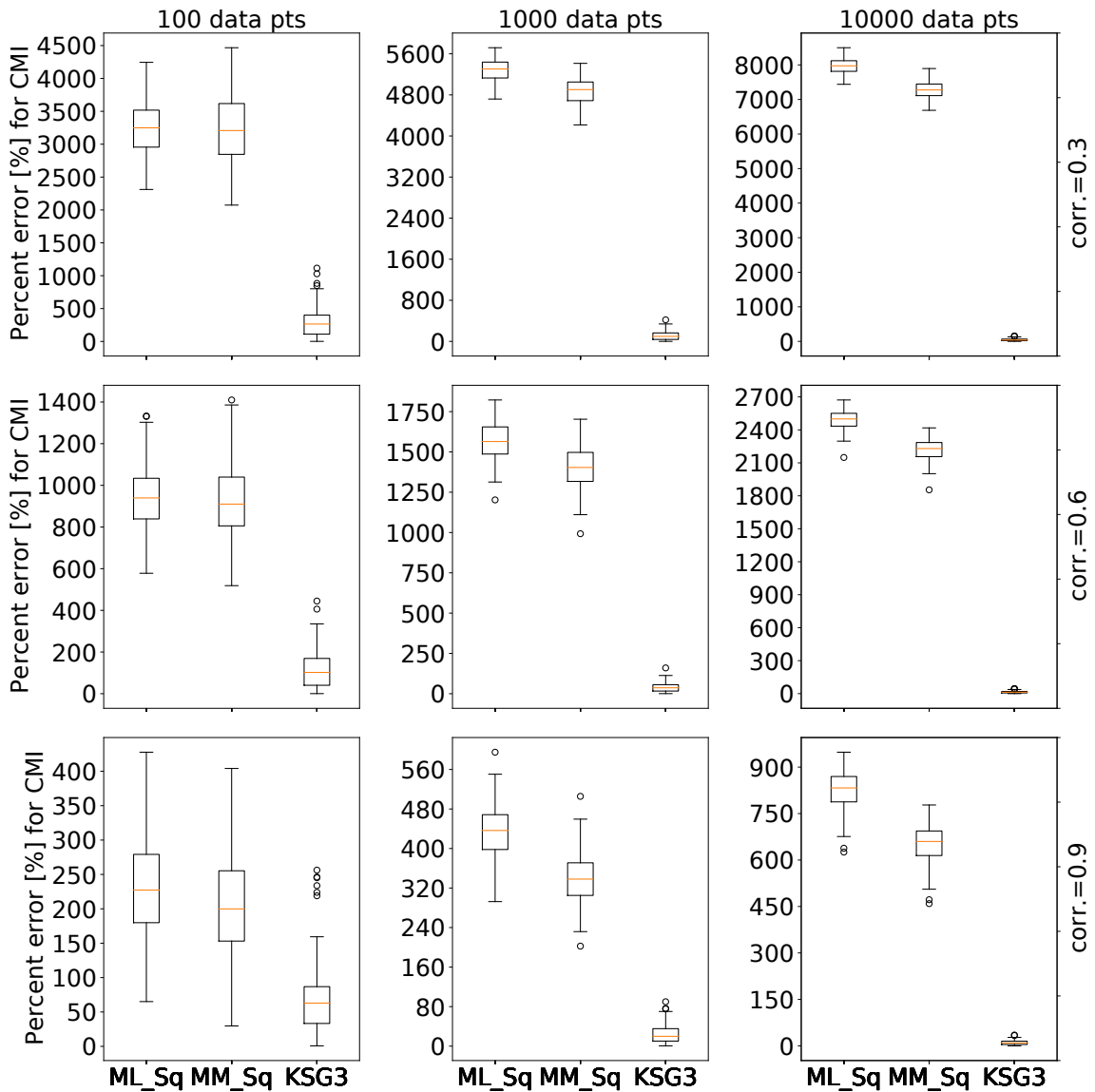


Figure II-3. Boxplots of percent error for Conditional Mutual Information (CMI) of three different mutual information estimators

With columns representing sample size = $\{100, 1K, 10K\}$, and rows the correlation = $\{0.3, 0.6, 0.9\}$. 9 subplots showing percent error for Conditional Mutual Information (CMI) for 3 different methods: ML_Sq=Shannon's MI with fixed width binning (number of bins is determined by square-root), MM_Sq=Miller-Madow formula for MI with square-root for the number of bins, kNN3=KSG formula for MI with $k=3$.

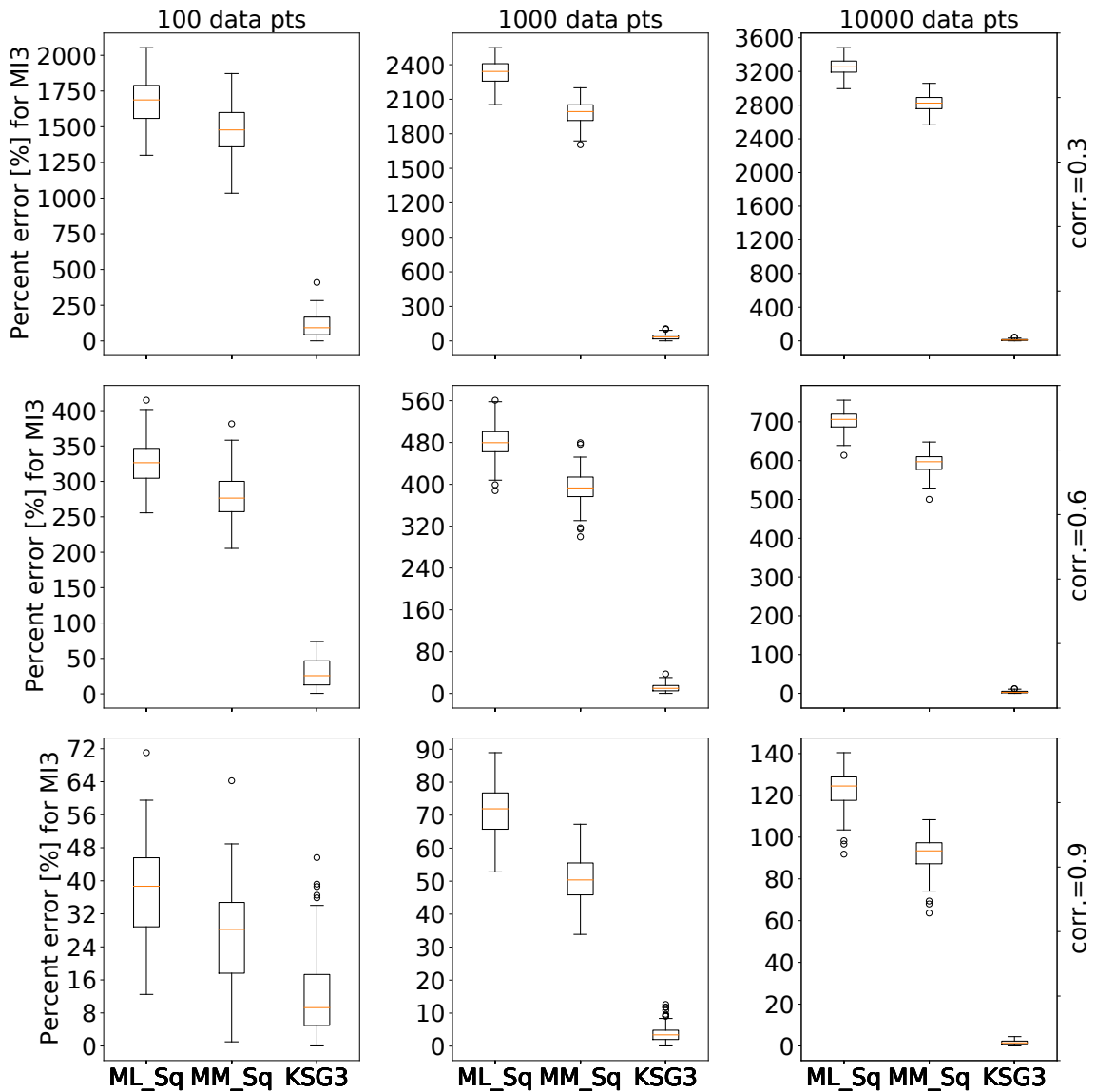


Figure II-4. Boxplots of percent error for Three-way Mutual Information (MI3) of three different mutual information estimators

With columns representing sample size = $\{100,1K,10K\}$, and rows the correlation = $\{0.3,0.6,0.9\}$. 9 subplots showing percent error for three-way mutual information (MI3) for 3 different methods: ML_Sq=Shannon's MI with fixed width binning (number of bins is determined by square-root), MM_Sq=Miller-Madow formula for MI with square-root for the number of bins, kNN3=KSG formula for MI with $k=3$.

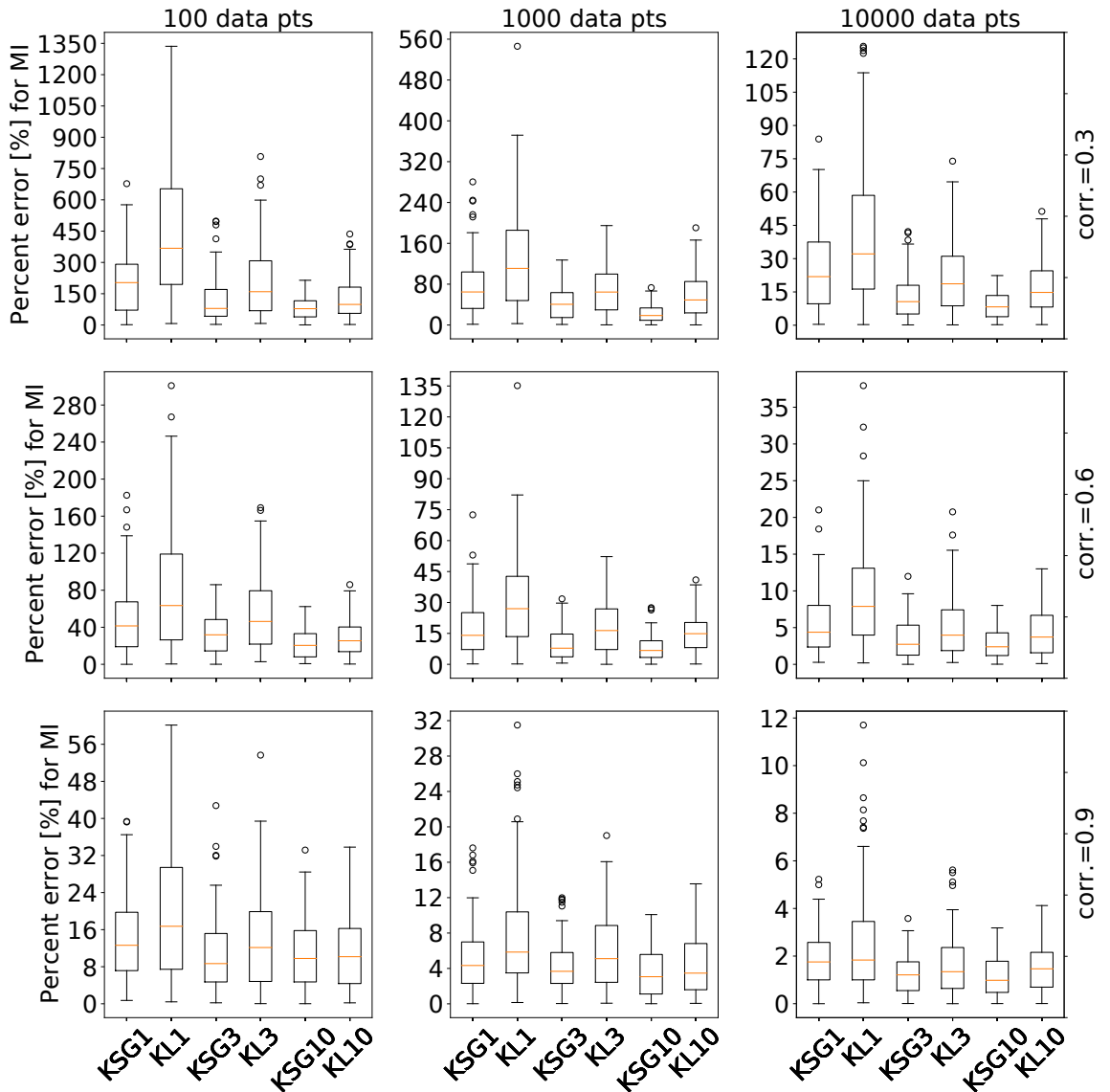


Figure II-5. Boxplots of percent error of Two-way Mutual Information calculated based on kNN methods.

100 replicates of bi-variate gaussian dist. With sample size = {100,1K,10K}, correlation = {0.3,0.6,0.9}. We compare KL and KSG methods for k=1,3,10.

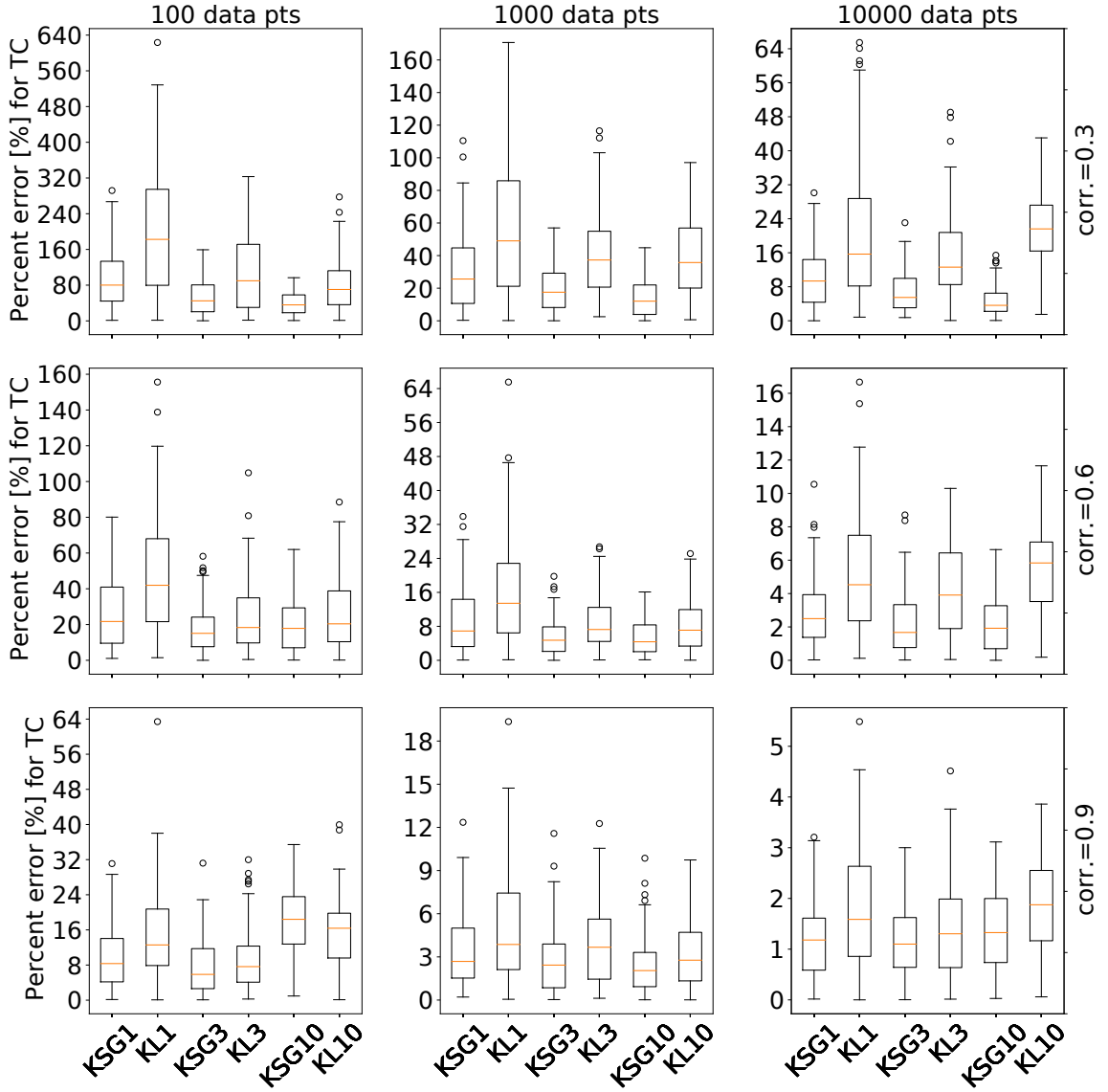


Figure II-6. Boxplots of percent error of Total Correlation calculated based on kNN methods.

100 replicates of tri-variate gaussian dist. With sample size = $\{100,1K,10K\}$, correlation = $\{0.3,0.6,0.9\}$. We compare KL and KSG methods for $k=1,3,10$.

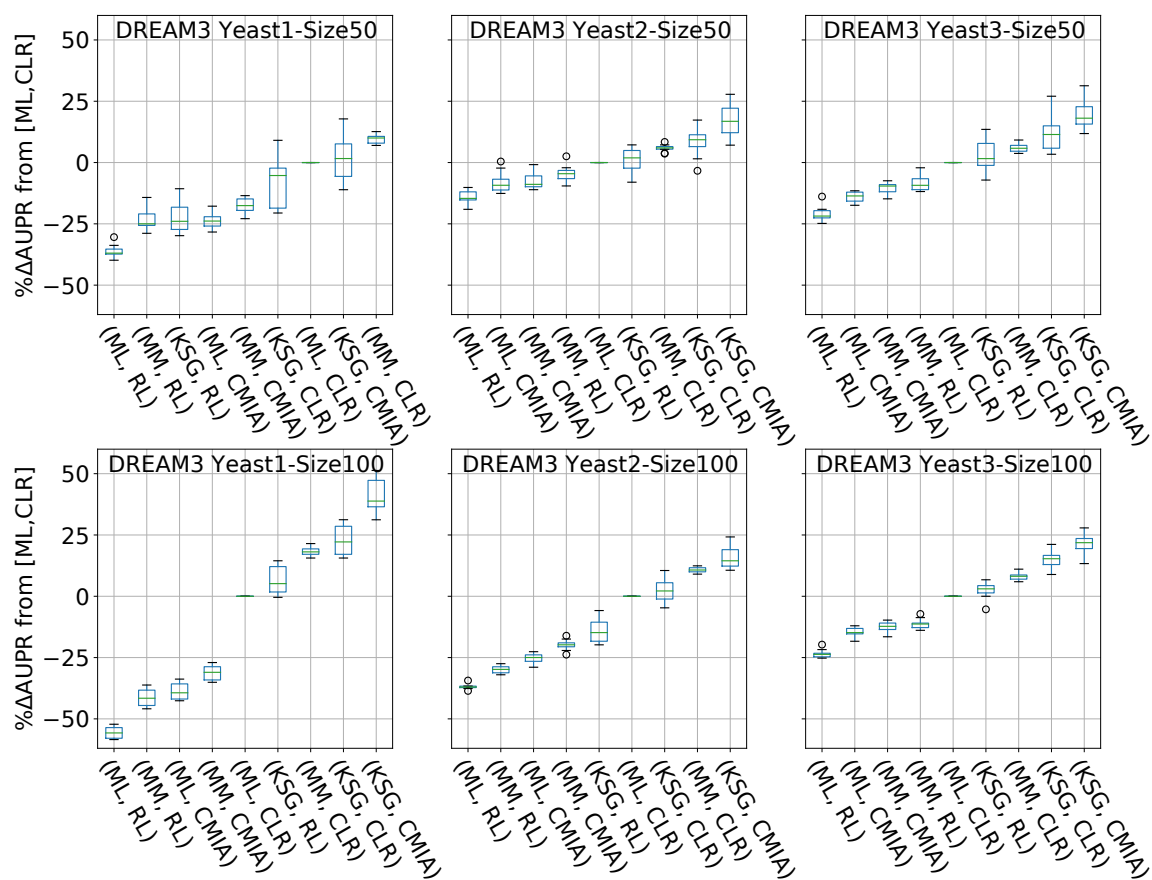


Figure II-7. AUPR difference relative to the gold standard combination [ML,CLR] for different Yeast networks from DREAM3

Sorted boxplots of percentage AUPR difference (increase or decrease) relative to the gold standard combination [ML,CLR] for different combinations of MI estimator and GRN inference algorithm for the 6 different Yeast networks from DREAM3. Each boxplot represents 10 replicates.

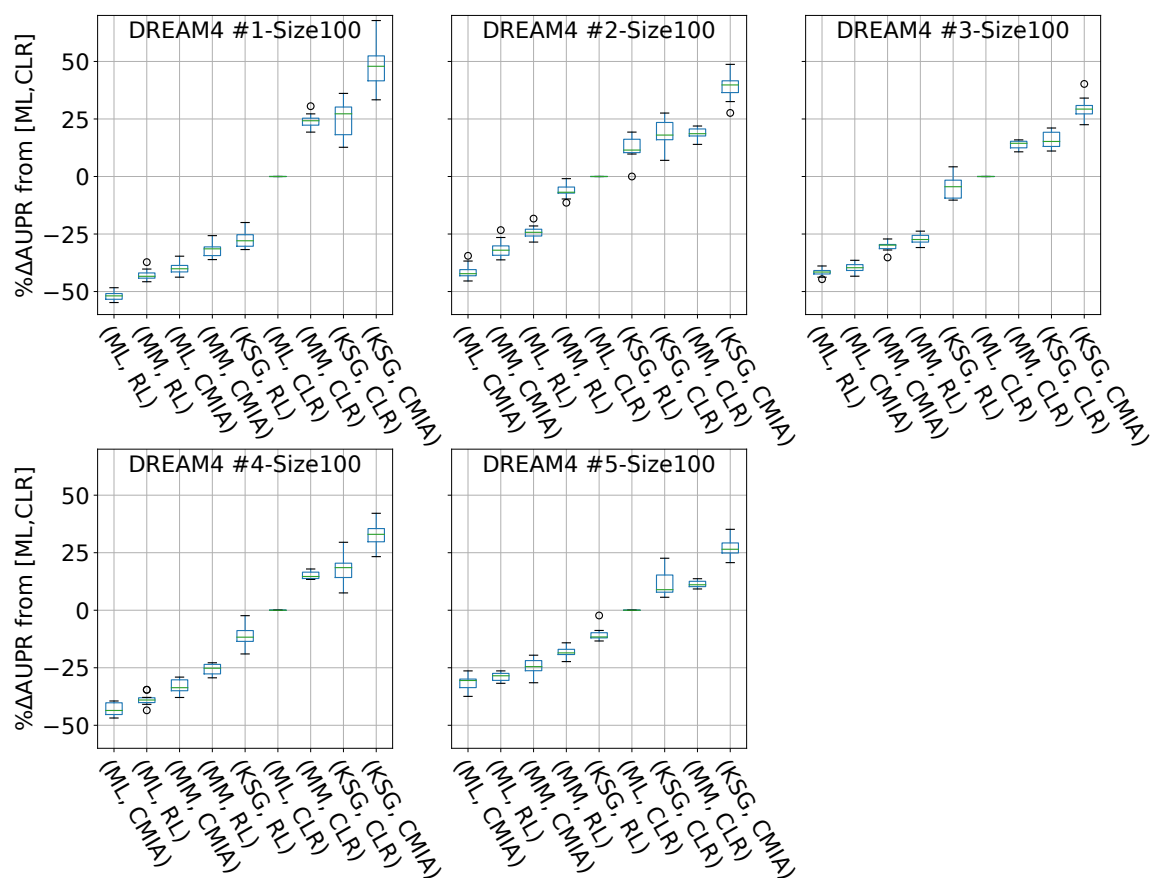


Figure II-8. AUPR difference relative to the gold standard combination [ML,CLR] for different networks of 100 genes from DREAM4
Sorted boxplots of percentage AUPR difference (increase or decrease) relative to the gold standard combination [ML,CLR] for different combinations of MI estimator and GRN inference algorithm for the 5 different networks of 100 genes from DREAM4. Each boxplot represents 10 replicates.

Appendix III

Supplementary information for chapter on network motifs classification

A. Supplementary text

Simulated 3-node motifs -16 motifs, no FFL

Labels = ['Cascade-SXY_II', 'Cascade-SXZ_II', 'Cascade-SYX_II', 'Cascade-SYZ_II', 'Cascade-SZX_II', 'Cascade-SZY_II', 'Fan-in-SXY_II', 'Fan-in-SXZ_II', 'Fan-in-SYZ_II', 'Fan-out-SX_II', 'Fan-out-SY_II', 'Fan-out-SZ_II', 'NoInteract', 'Two-genes-XY_II', 'Two-genes-XZ_II', 'Two-genes-YZ_II']

Models

'SVM': SVC(C=10, kernel='linear'), 'MLP': MLPClassifier(activation='tanh', learning_rate='invscaling'), 'RF': RandomForestClassifier(max_depth=8, n_estimators=50), 'GBT': GradientBoostingClassifier(max_depth=7, n_estimators=500)

Evaluate models on the validation set

SVM – Accuracy: 0.912 / Precision: [0.86666667 1. 0.95652174 0.95652174 1. 0.95454545 0.75 0.9047619 0.83333333 1. 1. 0.94444444 0.66666667 1. 0.92592593 1.] / Recall: [0.8125 0.82352941 0.95652174 1. 0.89473684 1. 0.85714286 0.73076923 0.9375 0.95238095 0.94117647 1. 1. 0.83333333 0.92592593 1.] / Latency: 7.9ms

MLP – Accuracy: 0.938 / Precision: [1. 0.9375 0.95652174 0.95652174 1. 1. 0.9047619 1. 0.83333333 1. 1. 0.94444444 0.64285714 1. 0.96428571 1.] / Recall: [0.875 0.88235294 0.95652174 1. 0.89473684 1. 0.9047619 0.88461538 0.9375 0.9047619 0.94117647 1. 1. 0.83333333 1. 1.] / Latency: 9.0ms

RF – Accuracy: 0.912 / Precision: [1. 0.83333333 0.88 0.86956522 1. 0.95454545 0.95 0.96 0.83333333 0.94736842 0.92307692 0.94117647 0.69230769 1. 0.96428571 0.9375] / Recall: [0.8125 0.88235294 0.95652174 0.90909091 0.89473684 1. 0.9047619 0.92307692 0.9375 0.85714286 0.70588235 0.94117647 1. 0.83333333 1. 1.] / Latency: 18.5ms

GBT – Accuracy: 0.888 / Precision: [0.83333333 1. 0.875 0.86956522 1. 1. 0.9 1. 0.72727273 1. 0.85714286 1. 0.57692308 0.8 0.96296296 1.] / Recall: [0.9375 0.88235294 0.91304348 0.90909091 0.89473684 1. 0.85714286 0.76923077 1. 0.85714286 0.70588235 1. 0.83333333 0.83333333 0.96296296 0.86666667] / Latency: 101.5ms

Evaluate best model on test set

MLP – Accuracy: 0.903 / Precision: [0.88888889 0.86956522 0.95 1. 1. 0.95454545 0.83333333 0.89473684 0.85714286 1. 1. 0.94736842 0.79310345 0.88235294 0.77272727 0.96] / Recall: [0.94117647 0.95238095 1. 0.9 0.73913043 0.95454545 0.95238095 0.73913043 0.81818182 0.82608696 0.94117647 1. 1. 0.75 1. 1.] / Latency: 7.6ms

B. Supplementary figures

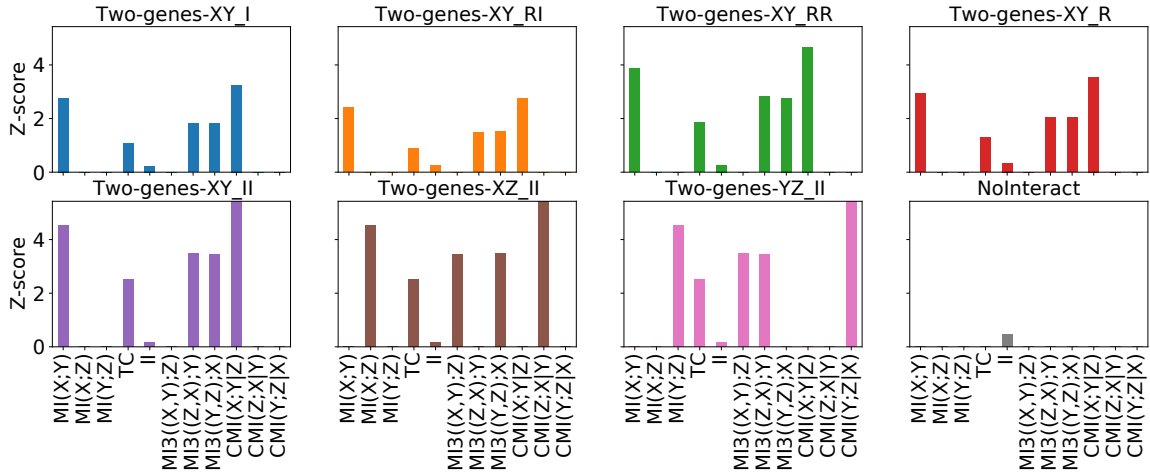


Figure III-1. Z-score profiles for Two-genes motifs

Each subplot represents a unique network motif. We show 7 Two-gene motifs and the No-interaction motif, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the Z-score value.

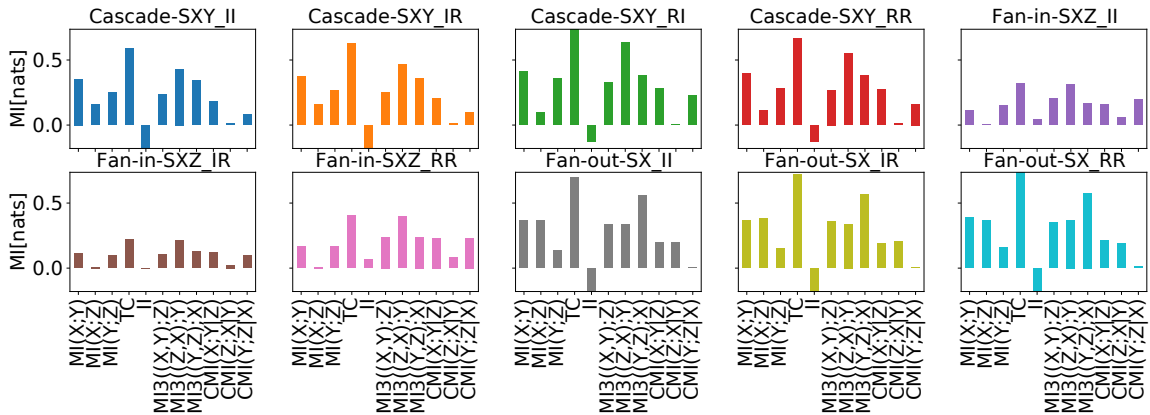


Figure III-2. MI profiles for two edge motifs with different repressing and inducing interactions

Each subplot represents a unique network motif. We show 10 motifs with two-edges, namely, Fan-in, Fan-out and Cascade with different combinations of repressing and inducing interactions, where, the x-axis shows all the 2d and 3d MI measures (11 in total), y-axis is the mean MI value in nats (information bits in e base).

