BRIEFINGS IN BIOINFORMATICS. VOL 15. NO 2. 195–211
Advance Access published on 21 May 2013

doi:10.1093/bib/bbt034

Supervised, semi-supervised and unsupervised inference of gene regulatory networks

Stefan R. Maetschke, Piyush B. Madhamshettiwar, Melissa J. Davis and Mark A. Ragan

Submitted: 19th January 2013; Received (in revised form): 15th April 2013

Abstract

Inference of gene regulatory network from expression data is a challenging task. Many methods have been developed to this purpose but a comprehensive evaluation that covers unsupervised, semi-supervised and supervised methods, and provides guidelines for their practical application, is lacking.

We performed an extensive evaluation of inference methods on simulated and experimental expression data. The results reveal low prediction accuracies for unsupervised techniques with the notable exception of the Z-SCORE method on knockout data. In all other cases, the supervised approach achieved the highest accuracies and even in a semi-supervised setting with small numbers of only positive samples, outperformed the unsupervised techniques.

Keywords: *gene regulatory networks; simulation; gene expression data; machine learning*

INTRODUCTION

Mapping the topology of gene regulatory networks is a central problem in systems biology. The regulatory architecture controlling gene expression also controls consequent cellular behavior such as development, differentiation, homeostasis and response to stimuli, while deregulation of these networks has been implicated in oncogenesis and tumor progression [1]. Experimental methods based, for example, on chromatin immunoprecepitation, DNaseI hypersensitivity or protein-binding assays are capable of determining the nature of gene regulation in a given system, but are time-consuming, expensive and require antibodies for each transcription factor [2]. Accurate computational methods to infer gene regulatory networks, particularly methods that leverage genome-scale experimental data, are urgently required not only to supplement empirical approaches but also, if possible, to explore these data in new more-integrative ways.

Many computational methods have been developed to infer regulatory networks from gene expression data, predominately using unsupervised techniques. Several comparisons have been made of network inference methods, but a comprehensive evaluation that covers unsupervised, semi-supervised and supervised methods is lacking, and many questions remain open. Here we address fundamental questions, including which methods are suitable for what kinds of experimental data types, and how many samples these methods require. In the following, we firstly review large-scale comparisons (more than five methods), before discussing evaluations focused on supervised and semi-supervised methods. Finally we discuss the remaining smaller comparisons with an application-specific focus.

Corresponding author. Mark Ragan. The University of Queensland, Institute for Molecular Bioscience and ARC Centre of Excellence in Bioinformatics, Brisbane, QLD 4072, Australia, Tel.: 61 7 3346 2616; Fax: 61 7 3346 2101; E-mail: m.ragan@uq.edu.au

Stefan Maetschke is a computer scientist, and his current research focus is on machine learning techniques and their application to the inference of gene regulatory and protein–protein interaction networks.

Piyush Madhamshettiwar is currently completing a PhD in bioinformatics. His work is focused on the inference and analysis of gene regulatory networks, and the development of integrative analytical workflows for cancer transcriptomics.

Melissa Davis is a computational cell biologist with research interests in genome-scale biological network and pathway analysis, and the integration of experimental data with knowledge-based modeling.

Mark Ragan has research interests in comparative mammalian and bacterial genomics, genome-scale bioinformatics and computational systems biology.

The most recent and largest comparison so far has been performed by Madhamshettiwar *et al.* [3]. They compared the prediction accuracy of eight unsupervised and one supervised method on 38 simulated data sets. The methods showed large differences in prediction accuracy but the supervised method was found to perform best, despite the parameters of the unsupervised methods having been optimized. Here we extend this study to 17 unsupervised methods and include a direct comparison with supervised and semi-supervised methods on a wide range of networks and experimental data types (knockout, knockdown and multifactorial).

Another comprehensive evaluation, limited to unsupervised methods, has been performed as part of the Dialogue for Reverse Engineering Assessments and Methods (DREAM), an annual open competition in network inference [4–8]. Results from DREAM highlight that network inference is a challenging problem. To quote Prill *et al.* [7], 'The vast majority of the teams' predictions were statistically equivalent to random guesses'. However, an important result of the DREAM competition is that under certain conditions, simple methods can perform well: '. . . the z-score prediction would have placed second, first, and first (tie) in the 10-node, 50-node, and 100-node subchallenges, respectively' [7].

Unsupervised methods rely on expression data only but tend to achieve lower prediction accuracies than supervised methods [3, 9, 10]. By contrast, supervised methods require information about known interactions for training, and this information is typically sparse. Semi-supervised methods reflect a compromise and can be trained with much fewer interaction data, but usually are not as accurate predictors as supervised methods. One of the few comparisons with supervised methods was performed by Mordelet and Vert [9]. They evaluated supervised inference of regulatory networks (SIRENE) in comparison to context likelihood of relatedness (CLR), algorithm for the reconstruction of accurate cellular networks (ARACNE), relevance networks (RN) and a Bayesian network (BN) on an *Escherichia coli* benchmark data set by Faith *et al.* [11] and found that the supervised method considerably outperformed the unsupervised techniques.

Cerulo *et al.* [10] compared supervised and semi-supervised support vector machines (SVMs) with two unsupervised methods and found the former superior. Our evaluation uses similar supervised and semi-supervised methods but includes many more unsupervised methods, distinguishes between experimental types and performs repeats, resulting in a more complete picture. A related evaluation by Schaffter *et al.* [12] compared six unsupervised methods on larger networks with 100, 200 and 500 nodes and simulated expression data. Again the z-score method was found to be one of the top performers in knockout experiments.

Several smaller evaluations have been performed but are largely restricted to four unsupervised methods (ARACNE, CLR, MRNET and RN) in comparisons with a novel approach on small data sets. The ARACNE method was introduced by Margolin *et al.* [13] and showed superior precision and recall when compared with RN and a BN algorithm on simulated networks. Meyer *et al.* [14] compared all four unsupervised inference algorithms on large yeast subnetworks (100 up to 1000 nodes) using simulated expression data, and Altay and Emmert-Streib [15] investigated the bias in the predictions of those algorithms. Faith *et al.* [11] evaluated CLR, ARACNE, RN and a linear regression model on *E. coli* interaction data from RegulonDB and found CLR to outperform the other methods. Lopes *et al.* [16] studied the prediction accuracy of ARACNE, MRNET, CLR and SFFS + MCE, a feature selection algorithm, on simulated networks and found the latter superior for networks with small node degree. Haynes and Brent [17] developed a synthetic regulatory network generator (GRENDEL) and measured the prediction accuracy of ARACNE, CLR, DBmcmc and Symmetric-N for various network sizes and different experimental types. Werhli *et al.* [18] compared RN, graphical Gaussian models (GGMs) and BNs on the Raf pathway, a small cellular signaling network with 11 proteins, and on simulated data. BNs and GGMs were found to outperform RN on observational data. Camacho *et al.* [19] compared regulatory strengths analysis, reverse engineering by multiple regression, partial correlations (PC) and dynamic BNs on a small simulated network with 10 genes, with different levels of noise. In the noise-free scenario, the PC method showed the highest accuracy. Finally, Cantone *et al.* [20] constructed a small, synthetic, *in vivo* network of five genes and measured time series and steady-state expression. In an evaluation of BANJO, ARACNE and two models based on ordinary differential equations, they found the latter two to achieve the highest accuracies. Bansal *et al.* [21] also evaluated BANJO, ARACNE and ordinary differential equations but on random networks and simulated expression data.

In the following sections, we first describe the different inference methods in detail, before evaluating their prediction accuracies on simulated and experimental gene expression data and regulatory networks of varying size. We continue with a discussion of the prediction results and conclude with guidelines for the use of the evaluated methods.

METHODS

We compared the prediction accuracy of unsupervised, semi-supervised and supervised network inference methods. Unsupervised methods do not use any data to adjust internal parameters. Supervised methods, on the other hand, exploit all given data to optimize parameters such as weights or thresholds. Semi-supervised methods use only part of the data for parameter optimization, for instance, a subset of known network interactions. Note that unsupervised methods would be rendered (at least) semi-supervised by optimizing their parameters on network data.

The inference methods we evaluate here aim to recreate the topology of a genetic regulatory network—that is, a network of gene-to-gene physical regulatory interactions, some of which, however, might be hidden by shortcuts [8]—based on expression data only. In this context, the accuracy of a method is assessed by the extent to which the network it infers is similar to the true regulatory network. Because many methods are not designed to infer self-interactions or interaction direction we disregard directed edges and self-interactions. Following other [9, 17, 22,] we quantify similarity by the Area under the Receiver Operator Characteristic curve (AUC)

$$AUC = \frac{1}{2}\sum_{k=1}^{n}(X_k - X_{k-1})(Y_k + Y_{k-1}) \qquad (1)$$

where X_k is the false-positive rate and Y_k is the true-positive rate for the k-th output in the ranked list of predicted edge weights. An AUC of 1.0 indicates perfect prediction, while an AUC of 0.5 indicates a performance no better than random predictions.

Note that in contrast to other measures such as F1 score, Matthews correlation, recall or precision [23], AUC does not require choice of a threshold to infer interactions from predicted weights; rather, it compares the predicted weights directly to the topology of the true network. In the Supplementary Material, we nonetheless report results based on F1 score and Matthews correlation.

To avoid discrepancies between the gene expression values generated by true regulatory networks and the actually known, partial networks, we performed evaluations on simulated, steady-state expression data, generated from subnetworks extracted from *E. coli* and *Saccharomyces cerevisiae* networks. This allowed us to assess the accuracy of an algorithm against a perfectly known true network [21]. We used *GeneNetWeaver* [12, 24] and *SynTReN* [25] to extract subnetworks and to simulate gene expression data.

GeneNetWeaver has been part of several evaluations, most prominently the DREAM challenges. The simulator extracts subnetworks from known interaction networks such as those of *E. coli* and *S. cerevisiae*, emulates transcription and translation, and uses a set of ordinary differential equations describing chemical kinetics to generate expression data for knockout, knockdown and multifactorial experiments.

To simulate knockout experiments, the expression value of each gene is in turn set to zero, whereas for knockdown experiments, the expression value is halved. In multifactorial experiments, the expression levels of a small number of genes are perturbed by a small random amount. In contrast to the DREAM challenge, we do not provide to the inference algorithms metadata such as which gene has been knocked out or knocked down. All unsupervised methods see only expression data, while supervised methods see expression data plus interaction data.

SynTReN is a similar but older simulator. Subgraphs are also extracted from *E. coli* and *S. cerevisiae* networks but it simulates only the transcription level and multifactorial experiments. However, *SynTReN* is faster than *GeneNetWeaver* and allows one to vary the sample number independently of the network size.

To enable a comprehensive and fair comparison, we evaluated the prediction accuracies of these inference methods on subnetworks with different numbers of nodes (10, . . . ,110) extracted from *E. coli* and *S. cerevisiae*, and used three experimental data types [(knockout, knockdown, multifactorial) with varying sample set sizes (10, . . . ,110)] simulated by GeneNetWeaver and SynTReN.

We performed no parameter optimization for unsupervised methods because this would require training data (known interactions) and render those methods supervised. For the supervised and semi-supervised methods, 5-fold cross-validation was applied and parameters were optimized on the training data only.

In addition to simulated data, we also evaluated all methods on two experimental data sets originating from the fifth DREAM systems biology challenge [8]. Specifically, we downloaded an *E. coli* network with 296 regulators, 4297 genes and the corresponding expression data with 487 samples, and an *S. cerevisiae* network with 183 regulators, 5667 genes and expression data with 321 samples. Both data sets are described in detail in the Supplementary Material of [8]. The following sections describe the inference methods in detail.

Unsupervised

This section describes the evaluated unsupervised methods. CLR, ARACNE, MRNET and MRNET-B are part of the R package 'minet' and were called with their default parameters [26], with the exception of ARACNE. With the default parameter $eps = 0.0$, ARACNE performed poorly and we used $eps = 0.2$ instead. Similarly, gene network inference with ensemble of trees (GENIE) [27], MINE [28] and partial correlation and information theory (PCIT) [29] were installed and evaluated with default parameters. All other methods were implemented according to their respective publications. SPEARMAN-C, EUCLID and SIGMOID are implementations of our own inference algorithms.

Correlation

Correlation-based network inference methods assume that correlated expression levels between two genes are indicative of a regulatory interaction. Correlation coefficients range from +1 to −1 and a positive correlation coefficient indicates an activating interaction, while a negative coefficient indicates an inhibitory interaction. The common correlation measure by Pearson is defined as

$$corr(X_i, X_j) = \frac{cov(X_i, X_j)}{\sigma(X_i) \cdot \sigma(X_j)} \tag{2}$$

where X_i and X_j are the expression levels of genes i and j, $cov(\cdot, \cdot)$ denotes the covariance, and $\sigma(\cdot)$ is the standard deviation. Pearson's correlation measure assumes normally distributed values, an assumption that does not necessarily hold for gene expression data. Therefore rank-based measures are frequently used, with the measures by Spearman and Kendall being the most common. Spearman's method is simply Pearson's correlation coefficient for the ranked expression values, and Kendall's τ coefficient is computed as

$$\tau(X_i, X_j) = \frac{con(X_i^r, X_j^r) - dis(X_i^r, X_j^r)}{\frac{1}{2}n(n-1)} \tag{3}$$

where X_i^r and X_j^r are the ranked expression profiles of genes i and j. $con(\cdot, \cdot)$ denotes the number of concordant and $dis(\cdot, \cdot)$ the number of disconcordant value pairs in X_i^r and X_j^r, with both profiles being of length n.

Because our evaluation of prediction accuracy does not distinguish between inhibiting and activating interactions, the predicted interaction weights are computed as the absolute value of the correlation coefficients

$$w_{ij} = |corr(X_i, X_j)|. \tag{4}$$

SPEARMAN-C

SPEARMAN-C is a modification of Spearman's correlation coefficient where we attempted to favor hub nodes, which have many, strong interactions. The correlation coefficient is corrected by multiplying it by the mean correlation of gene i with all other genes k, and the absolute value is taken as the interaction weight

$$w_{ij} = |corr(X_i, X_j) \cdot \frac{1}{n} \sum_{k}^{n} corr(X_i, X_k)| \tag{5}$$

where $corr(\cdot, \cdot)$ is Spearman's correlation coefficient.

Weighted gene co-expression network analysis

WGCNA [30] is a modification of correlation-based inference methods that amplifies high correlation coefficients by raising the absolute value to the power of β ('softpower').

$$w_{ij} = |corr(X_i, X_j)|^{\beta} \tag{6}$$

with $\beta \geq 1$. Because softpower is a nonlinear but monotonic transformation of the correlation coefficient, the prediction accuracy measured by AUC will be no different from that of the underlying correlation method itself. Consequently we show only results for correlation methods but not for the WGCNA modification, which would be identical.

Relevance networks

RN by Butte and Kohane [31] measure the mutual information (MI) between gene expression profiles to infer interactions. The MI I between discrete variables X_i and X_j is defined as

$$I(X_i, X_j) = \sum_{x_i \in X_i} \sum_{x_j \in X_j} p(x_i, x_j) \log\left(\frac{p(x_i, x_j)}{p(x_i)p(x_j)}\right) \tag{7}$$

where $p(x_i, x_j)$ is the joint probability distribution of X_i and X_j, and $p(x_i)$ and $p(x_j)$ are the marginal probabilities. X_i and X_j are required to be discrete variables. We used equal-width binning for discretization and empirical entropy estimation as described by Meyer *et al.* [26].

Context likelihood of relatedness

CLR [11] extends the RN by taking the background distribution of the MI values $I(X_i, X_j)$ into account. The most probable interactions are those that deviate most from the background distribution, and for each gene i, a maximum z-score z_i is calculated as

$$z_i = \max_j\left(0, \frac{I(X_i, X_j) - \mu_i}{\sigma_i}\right) \tag{8}$$

where μ_i and σ_i are the mean value and standard deviation, respectively, of the MI values $I(X_i, X_k)$, $k = 1, \ldots, n$. The interaction w_{ij} between two genes i and j is then defined as

$$w_{ij} = \sqrt{z_i^2 + z_j^2} \tag{9}$$

The background correction step aims to reduce the prediction of false interactions based on spurious correlations and indirect interactions.

Algorithm for the reconstruction of accurate cellular networks

ARACNE [13] is another modification of the RN that applies the Data Processing Inequality (DPI) to filter out indirect interactions. The DPI states that, if gene i interacts with gene j via gene k, then the following inequality holds:

$$I(X_i, X_j) \leq \min(I(X_i, X_j), I(X_j, X_k)) \tag{10}$$

ARACNE considers all possible triplets of genes (interaction triangles) and computes the MI values for each gene pair within the triplet. Interactions within an interaction triangle are assumed to be indirect and are therefore pruned if they violate the DPI beyond a specified tolerance threshold *eps*. We used a threshold of $eps = 0.2$ for our evaluations.

Partial correlation and information theory

PCIT [29] is similar to ARACNE. PCIT extracts all possible interaction triangles and applies the DPI to filter indirect interactions, but uses partial correlation coefficients instead of MI as interaction weights. The partial correlation coefficient $corr_{ij}^{\text{partial}}$ between two genes i and j within an interaction triangle (i, j, k) is defined as

$$corr_{ij}^{\text{partial}} = \frac{corr(X_i, X_j) - corr(X_i, X_k)corr(X_j, X_k)}{\sqrt{(1 - corr(X_i, X_k))^2(1 - corr(X_j, X_k))^2}} \tag{11}$$

where $corr(\cdot, \cdot)$ is Person's correlation coefficient. The partial correlation coefficient aims to eliminate the effect of the third gene k on the correlation of genes i and j.

MRNET

MRNET [14] uses the MI between expression profiles and a feature selection algorithm [minimum-redundancy–maximum-relevance (MRMR)] to infer interactions between genes. More precisely, the method places each gene in the role of a target gene j with all other genes V as its regulators. The MI between the target gene and the regulators is calculated and the MRMR method is applied to select the best subset of regulators. MRMR step-by-step builds a set S by selecting the genes i^{MRMR} with the largest MI value and the smallest redundancy based on the following definition:

$$i^{MRMR} = \operatorname*{argmax}_{i \in V \setminus S}(s_i) \tag{12}$$

with $s_i = u_i - r_i$. The relevance term $u_i = I(X_i, X_j)$ is thereby the MI between gene i and target j, and the redundancy term r_i is defined as

$$r_i = \frac{1}{|S|} \sum_{k \in S} I(X_i, X_k) \tag{13}$$

Interaction weights w_{ij} are finally computed as $w_{ij} = \max(s_i, s_j)$.

MRNET-B

MRNET-B is a modification of MRNET that replaces the forward selection strategy to identify the best subset of regulator genes by a backward selection strategy followed by a sequential replacement [32].

Gene network inference with ensemble of trees

GENIE is similar to MRNET in that it also lets each gene take on the role of a target regulated by the remaining genes and then uses a feature selection procedure to identify the best subset of regulator genes. In contrast to MRNET, Random Forests and Extra-Trees are used for regression and feature selection [27] rather than MI and MRMR.

SIGMOID

SIGMOID models the regulation of a gene by a linear combination with soft thresholding. The predicted expression value X'_{ik} of gene i at time point k is described by the sum over the weighted expression values X_{jk} of the remaining genes, constrained by a sigmoid function $\sigma(\cdot)$.

$$X'_{ik} = \sigma(\sum_{j \neq i}^{n} X_{jk} w_{ij} + b_i) \tag{14}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

The regulatory weights w_{ij} are determined by minimizing the following quadratic error function over the predicted expression values X'_{ik} and the observed values X_{ik}:

$$E(w, b) = \frac{1}{2} \sum_{i} \sum_{k} (X'_{ik} - X_{ik})^2 \tag{16}$$

Finally, the interaction weights w'_{ij} for the undirected network are computed by averaging over the forward and backward weights:

$$w'_{ij} = \frac{|w_{ij}| + |w_{ji}|}{2} \tag{17}$$

Mass–distance

MD by Yona *et al.* [33] is a similarity measure for expression profiles. It estimates the probability to observe a profile inside the volume delimited by the profiles. The smaller the volume, the more similar are the two profiles. Given two expression profiles X_i and X_j, the total probability mass of samples whose k-th feature is bounded between the expression values X_{ik} and X_{jk} is calculated as

$$\mathrm{MASS}_k(X_i, X_j) = \sum_{min(X_{ik}, X_{jk}) \leq x \leq max(X_{ik}, X_{jk})} freq(x) \tag{18}$$

where $freq(x)$ is the empirical frequency. The mass distance MD_{ij} is defined as the total volume of profiles bounded between the two expression profiles X_i and X_j and is estimated by the product over all coordinates k.

$$\mathrm{MD}_{ij} = \prod_{k}^{n} \mathrm{MASS}_k(X_i, X_j) \tag{19}$$

where n is the length of the expression profiles. Because the MD_{ij} is symmetric and positive, we interpret it directly as an interaction weight w_{ij}.

Mutual rank

MR by Obayashi and Kinoshita [34] uses ranked Pearson's correlation as a measure to describe gene co-expression. For a gene i, first Pearson's correlation with all other genes k is computed and ranked. Then the rank achieved for gene j is taken as score to describe the similarity of the gene expression profiles X_i and X_j:

$$rank_{ij} = \underset{j}{\mathrm{rank}}(corr(X_i, X_k), \ \forall k \neq i) \tag{20}$$

with $corr(\cdot, \cdot)$ being Pearson's correlation coefficient. The final interaction weight w_{ij} is calculated as the geometric average of the ranked correlation between gene i and j and vice versa:

$$w_{ij} = \frac{rank_{ij} \cdot rank_{ji}}{2} \tag{21}$$

Maximal information nonparametric exploration

MINE is a class statistics by Reshef [28]. The maximal information coefficient (MIC) is part of this class and a novel measure to quantify nonlinear relationships. We computed the MIC for expression profiles X_i and X_j and interpreted the MIC score as an interaction weight

$$w_{ij} = \mathrm{MIC}(X_i, X_j) \tag{22}$$

EUCLID

EUCLID is a simple method that uses the euclidean distance between the normalized expression profiles X'_i and of two genes as interaction weights

$$w_{ij} = \sqrt{\sum_{k} (X'_{ik} - X'_{jk})^2} \tag{23}$$

where profiles are normalized by computing the absolute difference of expression values X_{ik} to the median expression in profile X_i

$$X'_{ik} = |X_{ik} - \mathrm{median}(X_i)| \tag{24}$$

Z-SCORE

Z-SCORE is a network inference strategy by Prill *et al.* [7] that takes advantage of knockout data. It assumes that a knockout affects directly interacting genes more strongly than others. The z-score z_{ij} describes the effect of a knockout of gene i in the k-th experiment on gene j as the normalized deviation of the expression level X_{jk} of gene j for experiment k from the average expression $\mu(X_j)$ of gene j:

$$z_{ij} = |\frac{X_{jk} - \mu(X_j)}{\sigma(X_j)}| \tag{25}$$

The original Z-SCORE method requires knowledge of the knockout experiment k and is therefore not directly applicable to data from multifactorial experiments. The method, however, can easily be generalized by assuming that the minimum expression value within a profile indicates the knockout experiment $[\min(X_j) = X_{jk}]$. Equation (25) then becomes

$$w_{ij} = |\frac{\min(X_j) - \mu(X_j)}{\sigma(X_j)}| \tag{26}$$

and the method can be applied to knockout, knockdown and multifactorial data. Note that z_{ij} is an asymmetric score and we therefore take the maximum of z_{ij} and z_{ji} to compute the final interaction weight w_{ij} as

$$w_{ij} = \max(z_{ij}, z_{ji}) \tag{27}$$

Supervised

A great variety of different supervised machine learning methods has been developed. We limit our evaluation to SVMs because they have been successfully applied to the inference of gene regulatory networks [9] and can easily be trained in a semi-supervised setting [10]. We used the SVM implementation *SVMLight* by Joachims [35] for all evaluations.

SVMs are trained by maximizing a constrained, quadratic optimization problem over Lagrange multipliers α:

$$\max_{\alpha} \; L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$
$$\text{subject to} \begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad \text{for } \forall i. \end{cases} \tag{28}$$

The labels y_i determine the class to which feature vector $\mathbf{x}_i$ belongs and C is the so-called *complexity* parameter that needs to be tuned for optimal prediction performance. Once the optimal Lagrange multipliers α are found, a feature vector can be classified by its signed distance $d(\mathbf{x})$ to the decision boundary, which is computed as

$$d(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i \, \mathbf{x}_i^T \mathbf{x} + b \tag{29}$$

The distance $d(\mathbf{x})$ can be interpreted as a confidence value. The larger the absolute distance, the more confident the prediction, and similar to a correlation value we interpret the distance as an interaction weight.

In contrast to unsupervised methods, e.g. correlation methods, the supervised approach does not directly operate on pairs of expression profiles but on feature vectors that can be constructed in various ways. We computed the outer product of two gene expression profiles X_i and X_j to construct feature vectors:

$$\mathbf{x} = X_i X_j^T \tag{30}$$

The outer product was chosen because it is commutative, and predicted interactions are therefore symmetric and undirected. A sample set for the training of the SVM is then composed of feature vectors $\mathbf{x}_i$ that are labeled $y_i = +1$ for gene pairs that interact and $y_i = -1$ for those that do not interact.

If all gene pairs are labeled, all network interactions would be known and prediction would be unnecessary. In practice and for evaluation purposes, training is therefore performed on a set of labeled samples, and predictions are generated for the samples of a test set. Figure 1 depicts the concept. All samples within the training set are labeled and all remaining gene pairs serve as test samples.

Note that the term 'sample' in the context of supervised learning refers to a feature vector derived from a pair of genes and their expression profiles, whereas a sample in an expression data set refers to the gene expression values for a single experiment, e.g. a gene knockout.

We evaluate the prediction accuracy of the supervised method by generating labeled feature vectors for all gene pairs (samples) of a network. This entire sample set is then divided in to five parts. Each of the

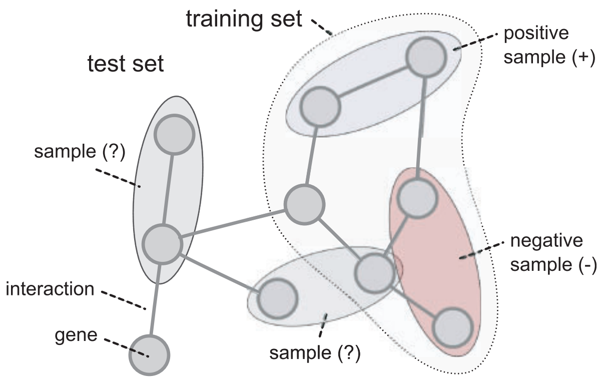

Figure 1: Extraction of samples for the training and test set from a gene interaction network.

parts is used as a test set and the remaining four parts serve as a training set. The total prediction accuracy is averaged over the prediction accuracies achieved during the five iterations (5-fold cross-validation).

Semi-supervised

Data describing regulatory networks are sparse and typically only a small fraction of the true interactions is known. The situation is even worse for negative data (non-interactions) because experimental validation largely aims to detect but not exclude interactions. The case that all samples within a training data set can be labeled as positive or negative is therefore rarely given for practical network inference problems, and supervised methods are limited to small training data sets, which negatively affects their performance.

Semi-supervised methods strive to take advantage of the unlabeled samples within a training set by taking the distribution of unlabeled samples into account, and can even be trained on positively labeled data only. Figure 2 shows the required labeling of data for the different approaches. Supervised methods require all samples within the training set to be labeled, while unsupervised methods require no labeling at all. Semi-supervised approaches can be distinguished into methods that need positive and negative samples and methods that operate on positive samples only.

The semi-supervised method used in this evaluation is based on the supervised SVM approach described above. The only difference is in the labeling of the training set. In the semi-supervised setting, only a portion of the training samples is labeled. To enable the SVM training, which requires all samples to be labeled, all unlabeled samples within the semi-supervised training data are relabeled as negatives [10]. This approach enables a direct comparison of the same prediction algorithm trained with fully or partially labeled data.

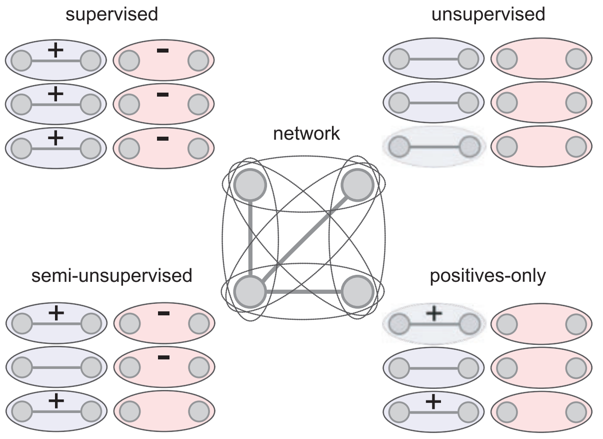

Figure 2: Original labeling of samples for supervised, unsupervised, semi-supervised and positives-only prediction methods. All the six samples within a sample set are generated by a four-node network with three interactions.

We assigned different percentages (10%, . . . ,100%) of true positive and negative or positive-only labels to the training set. The prediction performance of the different approaches was then evaluated by 5-fold cross-validation, with equal training/test set sizes for the supervised, semi-supervised, positives-only and unsupervised methods compared.

RESULTS

In the following, we first evaluate the prediction accuracy of unsupervised methods before comparing two selected unsupervised methods with supervised and semi-supervised approaches on simulated data. The last section compares unsupervised and supervised methods on experimental data.

Unsupervised methods

Figure 3 shows the prediction accuracies measured by AUC for all unsupervised methods for three different experimental types (knockout, knockdown and multifactorial) and the average AUC (all) over the three types. Networks with 10, 30, 50, 70, 90 and 110 nodes were extracted from *E. coli* and *S. cerevisiae* and expression data were simulated with GeneNetWeaver, with the number of samples (experiments) equal to the nodes of the network. Every evaluation was repeated 10 times, so each bar therefore represents an AUC averaged over 60 networks or 180 networks (all).

Most obvious are the large standard deviations in prediction accuracy across all methods and experimental types. For small networks, the accuracy of a method can easily vary between no better than guessing to close to perfect (see Supplementary Material). While most differences between methods are statistically significant (*P*-values <0.01 for Wilcoxon rank sum test with Bonferroni correction), differences are largely small and the ranking for most methods is therefore not stable and depends on the experimental data type, the source network, the subnetwork size and other factors (see Supplementary Material). However, a simple Pearson's correlation is consistently the second-best performer for all experimental types.

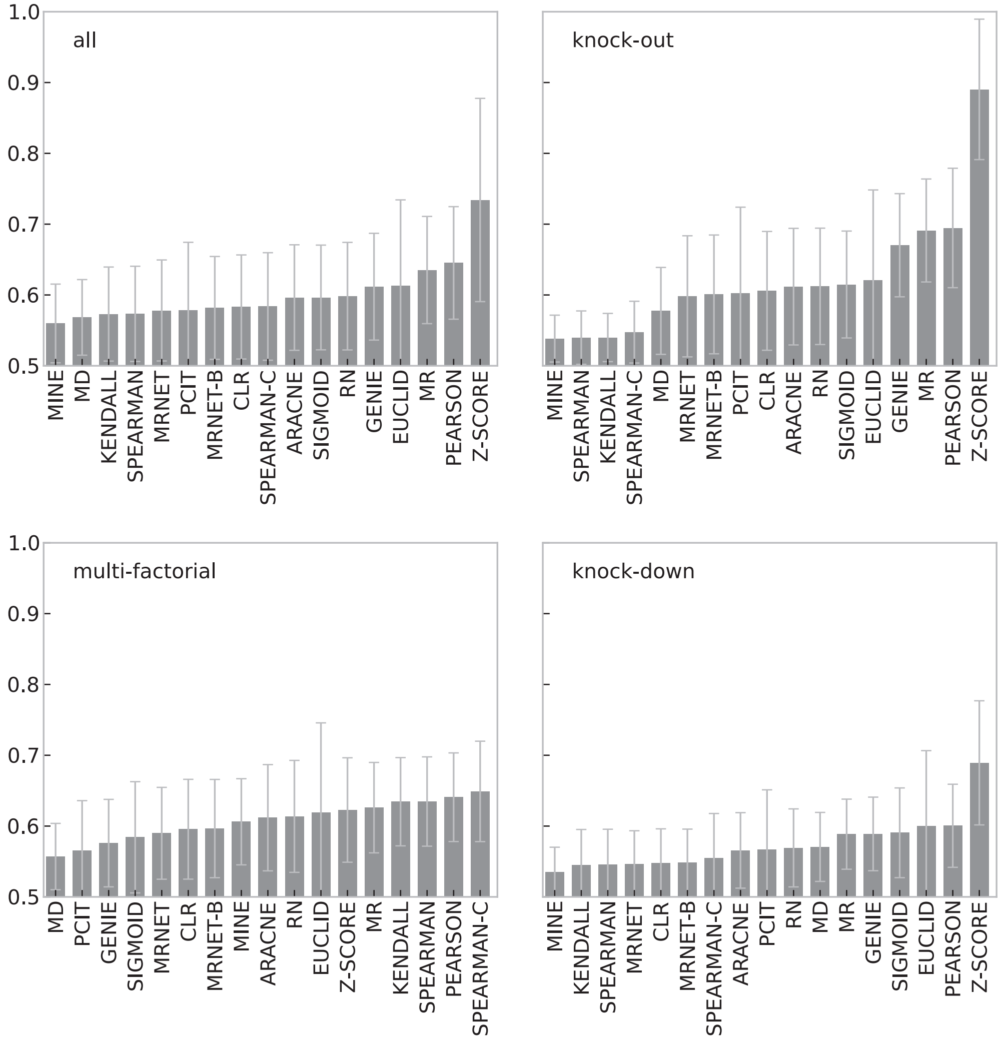

Figure 3: Prediction accuracy (AUC) of unsupervised methods on multifactorial, knockout, knockdown and averaged (all) data generated by GeneNetWeaver. Ten repeats over networks with 10,...,110 nodes, extracted from *E. coli* and *S. cerevisiae*. Error bars show standard deviation.

Interestingly, rank-based correlation methods (SPEARMAN, KENDALL) that are similar to Pearson correlation perform poorly on knockout and knockdown data but well for multifactorial experiments. Obviously, a seemingly minor change from a linear correlation (Pearson) to a rank-based correlation (Spearman) has a dramatic impact on the prediction performance in the case of knockout (and knockdown) data.

With the exception of the Z-SCORE method prediction, accuracies are low in general. Z-SCORE was specifically designed for knockout data and indeed clearly outperforms all other methods for this experimental type, despite its simplicity. It is the only unsupervised method that achieves a good prediction accuracy (AUC ≈ 0.9).

Network size

Figure 3 summarizes results averaged over networks. We also examined how the network size impacts the prediction performance of the various methods. The heat map in Figure 4 is based on the same data as Figure 3, but shows the prediction accuracies (AUC) of the inference methods on multifactorial data for networks with different numbers of nodes (see Supplementary Material for the related figures on knockout and knockdown data).

The rows in Figure 4 are ordered according to mean AUC and the ranking is therefore identical to that in the multifactorial bar graph in Figure 3. Top performers on average are the correlation methods by Pearson, Spearman and Kendall, with the corrected Spearman method (SPEARMAN-C) achieving the highest mean AUC. However, when focusing on networks of specific size, the best performance is achieved by the EUCLID method for small networks with 10 nodes. Other methods also show different behaviors with respect to network size. Correlation methods clearly achieve higher AUCs for large networks. Similar trends can be observed for MR, MINE, GENIE, MRNET, MRNET-B and CLR. In contrast, SIGMOID, PCIT and MD decrease in prediction accuracy for growing network sizes, while the performance of RN and ARACNE is seemingly unaffected by network size within the investigated size range.

Sample number

Apart from the size of the network, we also expected the number of samples to have an effect on the prediction accuracy of the inference algorithms. GeneNetWeaver generates gene expression profiles with the same number of samples as network nodes (genes). We therefore used SynTReN to vary network size and sample number independently. The heat map in Figure 5 shows prediction accuracy (AUC) averaged over all inference methods for different network sizes and sample numbers. SynTReN simulates expression data for multifactorial experiments only, and networks were extracted from *E. coli*. All experiments were repeated 10 times. The results show the expected trend of improving accuracy with increasing number of samples and decreasing size of network.

However, the absolute improvements in prediction accuracy are rather small with additional data, most likely because unsupervised methods can infer only simple network topologies reliably and small sample sets are sufficient for this purpose. For instance, networks with 50 nodes are predicted with an AUC of roughly 0.65, when 50 samples are available. Increasing the sample set size to 110 raises the prediction accuracy only to an AUC of around 0.67.

Supervised methods

Finally, we wanted to compare unsupervised with supervised and semi-supervised approaches. Because of the time-consuming training required for supervised methods, we limited our evaluation to networks with 30 nodes extracted from *E. coli* networks. Expression profiles were generated with GeneNet-Weaver, and each experiment was repeated 10 times.

Figure 6 shows the prediction accuracies (AUC) for supervised and semi-supervised methods for three different experimental types (knockout, knockdown and multifactorial) and the average AUC (all) data. For direct comparison, we included two

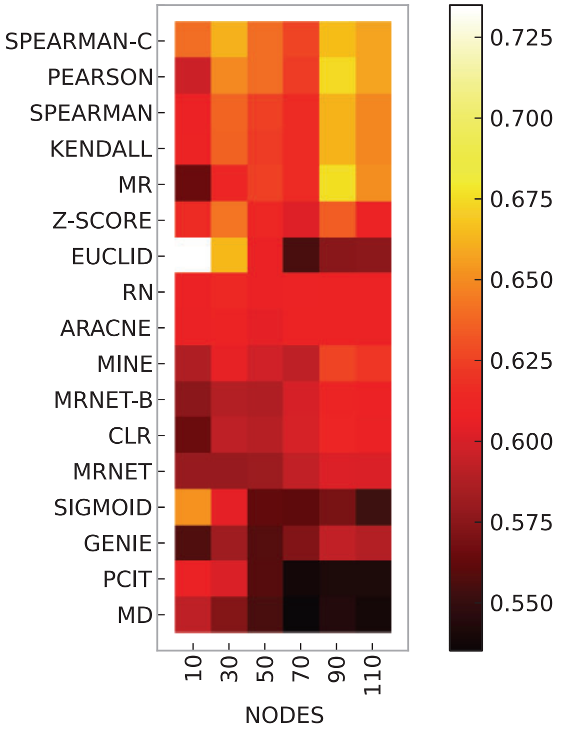

Figure 4: Prediction accuracy (AUC) of unsupervised methods on multifactorial data for different network sizes (nodes). Data generated by GeneNetWeaver and extracted from *E. coli* and *S. cerevisiae*.

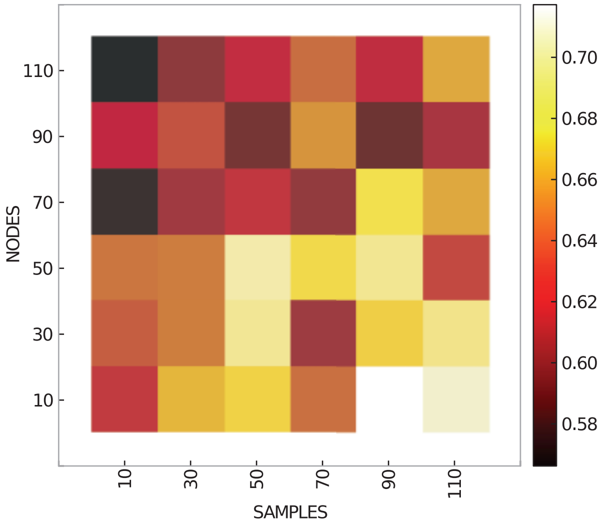

Figure 5: Prediction accuracy (AUC) averaged over all unsupervised methods on multifactorial for different network sizes (nodes) and sample numbers. Data generated by SynTReN and extracted from *E. coli*. Ten repeats.

unsupervised methods (Z-SCORE, SPEARMAN) in our evaluation of supervised methods. Supervised and semi-supervised methods are labeled 'SVM' followed by the percentage of labeled data (10, 30, 50, 70, 90, 100%). The suffix '+' indicates that only positive data were used and '±' indicates that positive and negative data were used. For instance, 'SVM-70±' describes an SVM trained on 70% of labeled data (positive and negative). All evaluations are 5-fold cross-validated and the complexity parameter *C* of the SVM was optimized via grid search (0.1 . . . 100) for each training fold.

The results show good prediction accuracies for supervised methods on all experimental types, with a slight advantage for knockout data. As expected, performance increases with the percentage of data labeled but there is little difference between labeling only positive data, and both positive and negative data. Apparently, supervised methods can be trained effectively even when only a portion of network interactions (positives) is known.

Even with as little as 10% of known interactions, semi-supervised methods still outperform unsupervised methods for multifactorial data. The Z-SCORE method is still the top-performing method on knockout data, but supervised methods are not far behind and considerably outperform Spearman correlation. For knockdown data, the Z-SCORE method loses its top rank, and semi-supervised methods perform better when at least 70% of the data are labeled.

To summarize, apart from the Z-SCORE method on knockout data, supervised and semi-supervised approaches considerably outperform unsupervised methods and achieve good prediction accuracies in general for networks of this size.

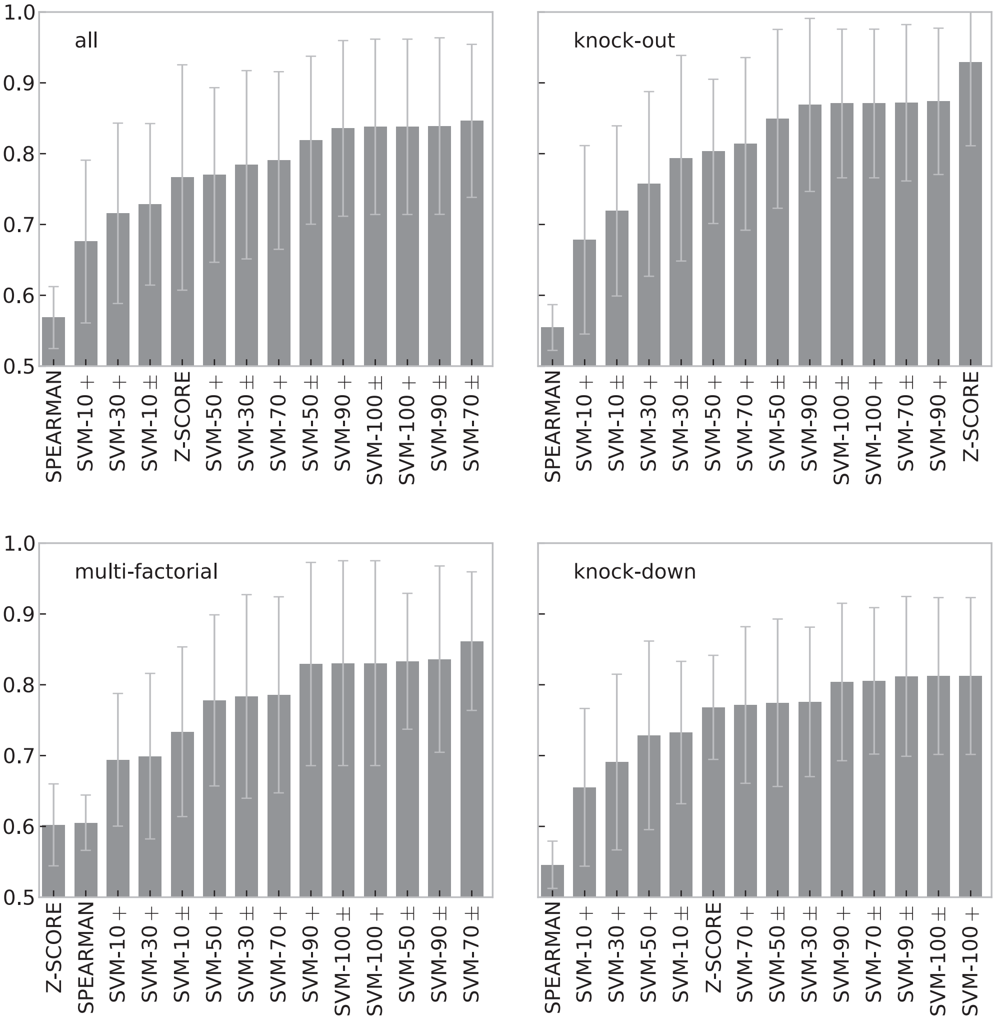

Figure 6: Prediction accuracy (AUC) of supervised methods on multifactorial, knockout, knockdown and averaged (all) data generated by GeneNetWeaver. Results are for 5-fold cross-validation and 10 repeats over networks with 30 nodes, extracted from *E. coli*. Error bars show standard deviation.

Experimental data

All results in the previous sections are based on simulated data. In this section, we analyze the performance of the inference methods on the experimental data described in Methods. For more-precise comparison, we applied the same methodology and used GeneNetWeaver to randomly extract subnetworks with 30 nodes from the experimental networks. Note, however, that GeneNetWeaver was not used to simulate gene expression—here we use empirical data [8] instead—and that the *E. coli* and *S. cerevisiae* source networks are different from those on which the simulation was based.

Figure 7 shows the prediction accuracy of the supervised and unsupervised methods for experimental networks and gene expression data of *E. coli* and *S. cerevisiae*. For better comparison, only the first 30 samples from the expression data sets were used. Results based on the complete expression data can be found in the Supplementary Material.

The results on the experimental data are in good agreement with the simulation results (Figures 3 and 6). The accuracy of the unsupervised methods remains low, while the supervised methods perform dramatically better. Even with only 10% of known interactions as a training source, the supervised methods outperform the unsupervised methods by a wide margin. The ranking of the unsupervised methods is inconsistent between data sets, but this is of little significance owing to their low accuracy and the large variances. Using the full expression data set (instead of only 30 samples) does not improve the

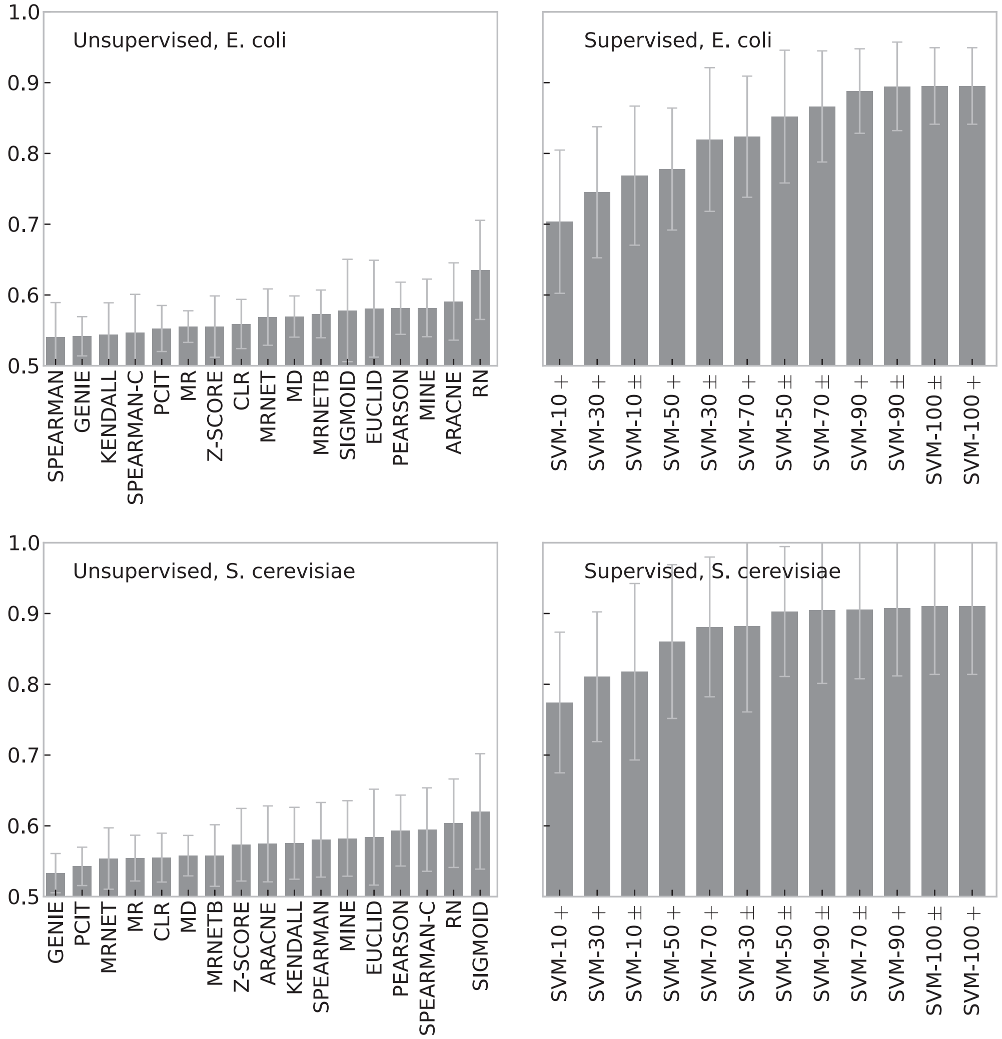

Figure 7: Prediction accuracy (AUC) of supervised and unsupervised methods for networks with 30 nodes extracted from *E. coli* and *S. cerevisiae*. The first 30 samples of the corresponding experimental expression data are used. AUCs are averaged over 10 repeats and error bars show standard deviation. Results for supervised methods are 5-fold cross-validated.

accuracy of the best-performing unsupervised methods (see Supplementary Material).

DISCUSSION

Directed interactions and self-interactions

Large-scale evaluations, including DREAM, measure prediction accuracy by comparing the topology of the inferred network with a known, true network. While this true network can contain directed interactions and loops, only a small subset of methods can infer direction or loops natively (e.g. GENIE [27]). GeneNetWeaver extracts directed edges and self-interactions if present, but to make our comparison direct and fair, we subsequently ignore them in computing AUC.

Unsupervised methods

Unsupervised methods are attractive because they do not require knowledge about the network to be inferred, and can be applied directly without a time-consuming training process. These are important practical advantages in comparison with supervised methods because reliable network data are often unavailable and training times for larger networks can become prohibitive. Parameter optimization would have removed these benefits and rendered unsupervised methods (semi-)supervised. On the other hand, as Madhamshettiwar *et al.* [3] have shown, parameter optimization can improve prediction accuracy.

Simulated data

While simulators such as GeneNetWeaver generate expression data that are in good agreement with biological measurements [6], they remain incomplete models, e.g. posttranscriptional regulation and chromatin states are missing, and an evaluation of inference methods on real data would clearly be preferable. However, currently known network structures, even for well-characterized organisms, are fragmentary and only partially correct representations of the interactions between genes [4]. Consequently, there is an unknown but probably large discrepancy between the expression data measured and the observed part of the actual network that generates them, rendering assessment of inference methods on observed gene regulatory networks and their expression values difficult. We therefore have largely focused our evaluation on *in silico* benchmarks, but methods that fail for simulated data are unlikely to succeed in the inference of real biological networks [21].

Linear SVMs

Another limitation of our study is the focus on linear SVMs for the evaluation of supervised and semi-supervised methods. We preferred linear SVMs over more-powerful nonlinear methods for two reasons. Firstly, linear SVMs are considerably faster to train and have fewer parameters to optimize than nonlinear SVMs—a significant advantage in a comprehensive study. Secondly, identifying a complex system with many variables (interaction weights) from a small number of samples calls for a simple predictor. We also tried to evaluate transductive SVMs [37] but found them time-consuming to train, and they achieved accuracies considerably lower than the semi-supervised SVMs (data not shown). We therefore did not perform a full evaluation and do not report results for transductive SVMs.

Feature vectors

We construct feature vectors by computing the outer product of the expression profiles of two genes. Cerulo *et al.* [10] constructed feature vectors by concatenating the two expression profiles. The outer product results in larger feature vectors (N^2 versus $2N$) but is independent of the order of the gene pair. The training set is therefore half the size compared with the concatenation approach [$n(n-1)$] and we achieved higher prediction accuracies with the linear SVM. Cerulo *et al.* [10], however, used nonlinear SVMs (RBF) that might achieve the same or better accuracies on concatenated feature vectors but are more time-consuming to train and require two parameters (C, γ) to be optimized. It therefore remains an open question, which method is preferable.

SIRENE by Mordelet and Vert [9] takes a different approach, with SVMs trained on feature vectors derived from single profiles. However, it requires knowledge about the transcription factors amongst the genes, and cannot predict interactions between target genes. Because each transcription factor is assigned a separate SVM, feature vectors are of length N and the training set has only n samples, the individual SVMs can be trained efficiently, but training time is multiplied by the number of transcription factors.

Unbalanced data sets

Gene regulatory networks tend to be sparse, with the number of positive samples (interactions) typically much smaller than the number of negative samples (non-interactions). Consequently data sets for the training of supervised methods are heavily unbalanced, and this could have a negative impact on the prediction accuracy of the classifier. We therefore tried to weight positive and negative samples inversely to their ratio, but did not observe any improvements in prediction accuracy (data not shown). All evaluations in this article were therefore performed with equally weighted ($w = 1$) samples.

Effect of sample number

We studied the effect of sample number on the prediction accuracy using an over-determined system (more samples than genes), and found only a marginal improvement for larger sample numbers. For an under-determined system (fewer samples than genes), increasing the number of samples is likely to be more beneficial. However, large networks cannot be inferred reliably (partly owing to a lack of data) and we therefore focused our evaluation on small networks, for which sufficient data (number of samples matching the number of genes) are often available in practice.

Network inference

The evaluation results reveal large variations in prediction accuracies across all methods. Nonlinear methods such as MINE do not perform better than linear Pearson's correlation, and in general, we find that complex methods are no more accurate than simple methods. The Z-SCORE method and Pearson correlation are the two best-performing unsupervised methods.

A detailed analysis revealed that unsupervised approaches work well for simple network topologies (e.g. star topology) and networks with exclusively activating or inhibiting interactions, but fail for more complex cases (see Supplementary Material). Mixed regulatory interactions constitute a fundamental problem for unsupervised network inference as depicted in Figure 8.

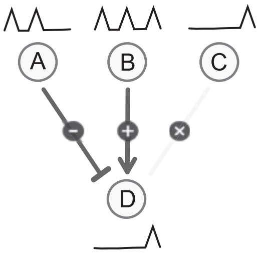

Figure 8: Gene A inhibits gene D, and gene B activates gene D. The resulting expression profile of gene D is, however, most similar to that of gene C, which does not regulate gene D.

Let gene A inhibit gene D but let gene B activate the same gene D. Given the expression profiles of genes A and B as shown in Figure 8, and assuming identical interaction weights but with opposite signs, the profile for gene D, resulting from a linear combination, is most similar to that of gene C and different from A or B. Consequently, the most appropriate but erroneous conclusion is to infer a regulatory relationship between C and D. Without any further information (e.g. knockouts, existing interactions) any method that infers interactions from the similarity of expression profiles alone is prone to fail in this common case. Schaffter *et al.* [12] identify other common network motifs and the methods that tend to infer them incorrectly, and Krishnan *et al.* [38] show that networks of a certain complexity cannot be reverse-engineered from expression data alone.

Experimental data

We evaluated all inference methods on simulated and experimental data, and found the results to be in good agreement. Specifically, the supervised methods consistently achieved much higher accuracies than unsupervised methods on both simulated and experimental data, and prediction accuracies for simulated and experimental data were similar. Marbach *et al.* [8] report substantially lower accuracies (of unsupervised methods) for *S. cerevisiae* and attribute this to the increased regulatory complexity and prevalence of posttranscriptional regulation in eukaryotes and/or to the lower coverage of the *S. cerevisiae* network.

In our evaluations, the accuracies for *S. cerevisiae* were only slightly lower than for *E. coli*, but note that Marbach *et al.* [8] infer the complete network with >5000 nodes, while we randomly extract small subnetworks with 10–110 nodes and calculate an average accuracy. Any part of the *S. cerevisiae* network with low coverage would affect the average accuracy less than it would the overall accuracy. It may be unnecessary to invoke regulatory complexity, especially considering the high accuracies of the supervised methods.

Marbach *et al.* [8] furthermore suggest that incorporating additional information such as transcription-factor binding and chromatin modification data or promoter sequences might improve the accuracy of prediction. We have performed evaluations with added transcription-factor binding data but could not improve prediction accuracies (data not shown).

Applications of inference methods

The inference of regulatory interactions from expression data has the potential to capture important associations between key genes regulating transcription in various biological conditions. Concerns about the accuracy of inference methods have led to the development of approaches that include GRN inference as one part of a broader computational framework. For example, He *et al.* [39] used modified correlation network inference to identify key genes involved in the immune suppressor function of human regulatory T cells. These authors combined two correlation-based approaches [36, 40] capable of identifying patterns of correlation across time-series microarray expression data. They then assessed the quality of their inferred networks by the extent to which interacting proteins shared biological process annotations, and selected key functional hubs from the inferred network using a metric computed from the results of literature mining and expression differences. They found that 6 of the top 10 hubs so ranked were already known to be involved in the suppressor function of regulatory T cells, and went on to experimentally characterize the novel role of the hub gene plasminogen activator urokinase (PLAU) in this suppressor function.

Similarly, Della Gatta *et al.* [41] applied the inference method ARACNE to reverse-engineer a genome-scale GRN in leukemia. To define the oncogenic regulatory circuits controlled by homeobox transcription factors TLX1 and TLX3, these investigators then used ChIP-chip data for these transcription factors along with expression data to identify direct targets of the TLX genes in the inferred network, requiring target genes to be differentially expressed between tumors that express TLX1 and TLX3. They then selected the most highly connected hub, RUNX1, as a putative master regulator of the TLX1 and 3 transcriptional programs, and validated the predictions of their network model with further ChIP-chip analysis of RUNX1.

These examples illustrate how GRN inference can productively be combined with downstream analysis: a GRN is initially inferred from experimental data, features of interest are identified based on network topology and interesting features are then ranked by applying additional criteria derived from experimental evidence, functional annotation or literature. Both He *et al.* [39] and Della Gatta *et al.* [41] used GRNs to identify highly connected nodes as features of interest, and then either rank these hubs, or refine the networks, by use of additional evidence. Interesting hubs prioritized in this way then form the basis of hypotheses that can be subjected to experimental investigation. Such hybrid approaches leverage the power of GRN inference methods, while controlling for variability in prediction performance by using additional criteria to select network features of interest.

CONCLUSION

Perhaps the most important observation from this evaluation is the large variance in prediction accuracies across all methods. In agreement with Haynes and Brent [17], we find that a large number of repeats on networks of varying size is required for reliable estimates of the prediction accuracy of a method. Evaluations on single data sets—especially on real data—are unsuitable to establish differences in the prediction accuracy of inference methods.

On average, unsupervised methods achieve low prediction accuracies, with the notable exception of the Z-SCORE method, and are considerably outperformed by supervised and semi-supervised methods. Simple correlation methods such as Pearson correlation are as accurate as much more complex methods, yet much faster and parameterless. Unsupervised methods are appropriate for the inference only of simple networks that are entirely composed of inhibitory or activating interactions but not both.

The Z-SCORE method achieved the best prediction accuracy of all methods on knockout data. However, experimental knockouts cannot be performed systematically, or at all, in many important biological systems. The method also fails when a gene is regulated by an or-junction of two genes.

On multifactorial data, the supervised and semi-supervised methods achieved the highest accuracies; even with as few as 10% of known interactions, the semi-supervised methods still outperformed all unsupervised approaches. There was little difference in prediction accuracy for semi-supervised methods trained on positively labeled data only, compared with training on positive and negative samples. Apparently semi-supervised methods can effectively be trained on partial interaction data, and non-interaction data are not essential.

These results have important implications for the application of network inference methods in systems biology. Even the best methods are accurate only for small networks of relatively simple topology, which means that large-scale or genome-scale regulatory network inference from expression data alone is currently not feasible. If inference methods are to be applied to data of the scale generated by modern microarray platforms, a feature selection step is usually required to reduce the size of the inference problem; attempts to apply network inference to such large-scale datasets may be premature, and consideration should be given to focusing the biological question to use smaller-scale higher-quality experimental data.

Our analysis also indicates that certain kinds of biological data are more amenable for accurate network inference than others. Most microarray datasets are most similar to our multifactorial data sets, which yielded poorly inferred networks with unsupervised methods. Increasing the number of samples in the experiment (a common strategy to improve inference) does not in fact generate the hoped-for improvements. More useful are knockout data, which our simulations show contain more useful information, and support higher-quality inference. Biologists who wish to gain insight into regulatory architecture should consider these limitations when designing experiments.

To summarize, small networks ($\leq$50 nodes) can be inferred with high accuracy (AUC $\approx$ 0.9) even with small numbers of samples using supervised techniques or the Z-SCORE method. However, even with the best-performing methods, large variations in prediction accuracy remain, and predictions may be limited to undirected networks without self-interactions.

SUPPLEMENTARY DATA

Supplementary data are available online at http://bib.oxfordjournals.org/

Key points

- Prediction accuracies strongly depend on the complexity of the network topology.
- Supervised methods generally achieve higher prediction accuracies than unsupervised methods.
- Unsupervised inference methods are accurate only for networks with simple topologies.
- An exception is the unsupervised Z-SCORE method that shows the highest accuracy on simulated knockout data.
- Knockout data are more informative for network inference than knockdown or multifactorial data.

FUNDING

Australian Research Council (DP110103384 and CE0348221).

References

1. Pe'er D, Hacohen N. Principles and strategies for developing network models in cancer. *Cell* 2011;**144**: 864–73.
2. Elnitski L, Jin V, Farnham P, *et al*. Locating mammalian transcription factor binding sites: a survey of computational and experimental techniques. *Genome Res* 2006;**16**(12): 1455–64.
3. Madhamshettiwar P, Maetschke S, Davis M, *et al*. Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. *Genome Med* 2012;**4**(5):41.
4. Stolovitzky G, Monroe D, Califano A. Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference. *Ann N Y Acad Sci* 2007;**1115**:1–22.
5. Stolovitzky G, Prill R, Califano A. Lessons from the DREAM2 challenges. *Ann N Y Acad Sci* 2009;**1158**: 159–95.
6. Marbach D, Prill R, Schaffter T, *et al*. Revealing strengths and weaknesses of methods for gene network inference. *Proc Natl Acad Sci USA* 2010;**107**(14):6286–91.
7. Prill RJ, Marbach D, Saez-Rodriguez J, *et al*. Towards a rigorous assessment of systems biology models: The DREAM3 challenges. *PLoS One* 2010;**5**(2):e9202.
8. Marbach D, Costello J, Küffner R, *et al*. Wisdom of crowds for robust gene network inference. *Nat Methods* 2012;**9**(8): 796–804.
9. Mordelet F, Vert JP. SIRENE: supervised inference of regulatory networks. *Bioinformatics* 2008;**24**:i76–82.
10. Cerulo L, Elkan C, Ceccarelli M. Learning gene regulatory networks from only positive and unlabeled data. *BMC Bioinformatics* 2010;**11**:228.
11. Faith JJ, Hayete B, Thaden JT, *et al*. Large-scale mapping and validation of *escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol* 2007;**5**(1):e8.
12. Schaffter T, Marbach D, Floreano D. GeneNetWeaver: *in silico* benchmark generation and performance profiling of network inference methods. *Bioinformatics* 2011;**27**(16): 2263–70.
13. Margolin A, Nemenman I, Basso K, *et al*. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 2006;**7(Suppl 1)**:S7.

14. Meyer P, Kontos K, Lafitte F, Bontempi G. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J Bioinform Syst Biol* 2007;**2007**:79879.
15. Altay G, Emmert-Streib F. Revealing differences in gene network inference algorithms on the network level by ensemble methods. *Bioinformatics* 2010;**26**(14):1738–44.
16. Lopes FM, Martins DC, Cesar RM. Comparative study of GRNS inference methods based on feature selection by mutual information. In: *IEEE International Workshop on Genomic Signal Processing and Statistics*. Guadalajara, JA, Mexico, 2009.
17. Haynes B, Brent M. Benchmarking regulatory network reconstruction with GRENDEL. *Bioinformatics* 2009;**25**(6): 801–7.
18. Werhli A, Grzegorczyk M, Husmeier D. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and Bayesian networks. *Bioinformatics* 2006;**22**(20): 2523–31.
19. Camacho D, Vera Licona P, Mendes P, Laubenbacher R. Comparison of reverse-engineering methods using an *in silico* network. *Ann N Y Acad Sci* 2007;**1115**:73–89.
20. Cantone I, Marucci L, Iorio F, *et al*. A yeast synthetic network for *in vivo* assessment of reverse-engineering and modeling approaches. *Cell* 2009;**137**:172–81.
21. Bansal M, Belcastro V, Ambesi-Impiombato A, di Bernardo D. How to infer gene networks from expression profiles. *Mol Syst Biol* 2007;**122**:78.
22. Husmeier D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics* 2003;**19**(17): 2271–82.
23. Baldi P, Brunak S, Chauvin Y, *et al*. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 2000;**16**(5):412–24.
24. Marbach D, Schaffter T, Mattiussi C, *et al*. Generating realistic *in silico* gene networks for performance assessment of reverse engineering methods. *J Comput Biol* 2009;**16**(2): 229–39.
25. Van den Bulcke T, Van Leemput K, Naudts B, *et al*. SynTReN: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics* 2006;**7**:43.
26. Meyer P, Lafitte F, Bontempi G. minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics* 2008;**9**:461.
27. Huynh-Thu V, Irrthum A, Wehenkel L, *et al*. Inferring regulatory networks from expression data using tree-based methods. *PLoS One* 2010;**5**(9):e12776.
28. Reshef DN. Detecting novel associations in large data sets. *Science* 2011;**334**:1518–24.
29. Reverter A, Chan EK. Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. *Bioinformatics* 2008;**24**:2491–7.
30. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* 2008;**9**:559.
31. Butte AJ, Kohane IS. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput* 2000;**1**:418–29.
32. Patrick M, Daniel M, Sushmita R, *et al*. Information-theoretic inference of gene networks using backward elimination. In: *The 2010 International Conference on Bioinformatics and Computational Biology* 2010.
33. Yona G, Dirks W, Rahman S, Lin DM. Effective similarity measures for expression profiles. *Bioinformatics* 2006;**22**(13): 1616–22.
34. Obayashi T, Kinoshita K. Rank of correlation coefficient as a comparable measure for biological significance of gene expression. *DNA Res* 2009;**16**:249–60.
35. Joachims T. Making large-scale SVM learning practical. In: Schölkopf B, Burges C, Smola A (eds). *Advances in Kernel Methods - Support Vector Learning*. Cambridge, MA: MIT Press, 1999;169–84.
36. He F, Zeng AP. In search of functional association from time-series microarray data based on the change trend and level of gene expression. *BMC Bioinformatics* 2006;**7**:69.
37. Joachims T. Retrospective on transductive inference for text classification using support vector machines. In: *Proceedings of the International Conference on Machine Learning (ICML)*. Montreal, Quebec, 2009.
38. Krishnan A, Giuliani A, Tomita M. Indeterminacy of reverse engineering of gene regulatory networks: the curse of gene elasticity. *PloS One* 2007;**2**(6):e562.
39. He F, Chen H, Probst-Kepper M, *et al*. PLAU inferred from a correlation network is critical for suppressor function of regulatory T cells. *Mol Syst Biol* 2012;**8**:624.
40. Qian J, Dolled-Filhart M, Lin J, *et al*. Beyond synexpression relationships: local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. *J Mol Biol* 2001;**314**(5):1053–66.
41. Della Gatta G, Palomero T, Perez-Garcia A, *et al*. Reverse engineering of TLX oncogenic transcriptional networks identifies RUNX1 as tumor suppressor in T-ALL. *Nat Med* 2012;**18**(3):436–40.