Data-driven Gene Regulatory Network Inference based on Classification Algorithms

Sergio Peignier, Pauline Schmitt, Federica Calevro

Data-driven Gene Regulatory Network Inference based on Classification Algorithms

Sergio Peignier , Pauline Schmitt , and Federica Calevro

Univ Lyon, INSA Lyon, INRA, *BF2I, UMR0203, F-69621*, Villeurbanne, France
{*name.lastname*}*@insa-lyon.fr*

Abstract

Different paradigms of gene regulatory network inference have been proposed so far in the literature. The data-driven family is an important inference paradigm, that aims at scoring potential regulatory links between transcription factors and target genes, analyzing gene expression datasets. Three major approaches have been proposed to score such links relying on correlation measures, mutual information metrics, and regression algorithms. In this paper we present a new family of data-driven inference approaches, inspired on the regression based family, and based on classification algorithms. This paper advocates for the use of this paradigm as a new promising approach to infer gene regulatory networks. Indeed, the implementation and test of five new inference methods based on well-known classification algorithms shows that such an approach exhibits good quality results when compared to well-established paradigms.

Keywords: Bioinformatics, Gene Regulatory Network Inference, Classification

1 Introduction

Gene Regulatory Networks (GRNs) describe the complex interactions between specialized genes such as transcription factors (TFs) and their target genes (TGs). Such interactions mediate to a large extent the regulation of the expression of genes, and the adaptation of biological systems to different conditions. Understanding such networks, and analysing their organization and their dynamics, are therefore important steps towards the comprehension of complex mechanisms that shape living organisms. The development of high-throughput technologies has motivated the creation of computational methods to reverse-engineer the underlying GRNs. Inferring GRNs from high-throughput data is a challenging problem, and different methods tackling

this have been proposed so far in the system biology literature [1]. Among the major categories of methods, the so-called *data-driven* approaches are among the most popular techniques due to their simplicity, their computational efficiency and their accuracy [1]. These methods aim at scoring each possible regulatory link, by estimating the *dependency* between genes from experimental high-throughput data. Depending on the method used to infer the link between genes, data-driven methods have been classified in three major families [1], gathering respectively methods based on i) correlation metrics, ii) mutual information, and iii) feature importance scoring based on regression algorithms. In practice, scoring the dependency between TFs and TGs using feature importance, could similarly be achieved using classification algorithms, however this paradigm has been understudied in the literature.

In order to assess the effectiveness of this new paradigm, we present in this paper a computational framework including 5 data-driven inference methods, based on classification algorithms. The proposed methods have been assessed and compared to state-of-the-art approaches using the benchmark datasets and the evaluation procedure described in [2]. Our comparative analysis revealed that the proposed methods allow to obtain better results than state-of-the-art techniques, and thus, methods based on this paradigm are interesting tools for the analysts. We also included 8 well-known preprocessing techniques that were also tested to study their impact on the inference quality. For the sake of reproducibility, our framework as well as the result tables are available online[1].

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 describes more formally the GRN inference problem. Section 4 introduces major preprocessing techniques that were tested. Section 5 introduces our data-driven GRN inference paradigm, based on classification algorithms. Section 6 and 7 describe respectively the experimental assessment protocol and the results. We conclude with a summary and some perspectives in Section 8.

2 State-of-the-art

2.1 Gene regulatory network inference methods

Gene regulatory network inference techniques from gene expression matrix can be classified in three major families, namely *Model-Based*, *Data-Driven* and *Multi-Network* approaches. This section reviews the founding principles of these families, as well as previous works on GRN inference based on classification algorithms.

Model-Based These approaches rely on a predetermined GRN model, based on specific hypothesis and parameters. In this context, inferring a GRN, consists in fitting the parameters of the model with respect to experimental data. Once fitted, the GRN model can be used to simulate and study the biological system *in-silico*. Two major families of model-based approaches have been reported in [1], namely *Probabilistic Models* and *Dynamical Models*. The former family mainly includes Bayesian networks and Gaussian Graphical Models. While the latter one has been used to study time series of gene expression data, incorporating modeling techniques that can integrate temporal considerations, such as Dynamic Bayesian Networks, Ordinal Differential Equations, Boolean Networks, Probabilistic Boolean Networks and Neural Networks.

[1] https://gitlab.com/speignier/classifiedgrni

We refer the reader to [3] for a thorough survey of model-based GRN inference and modeling methods.

Data-Driven methods These methods aim at scoring each possible regulatory link between TFs and TGs, by estimating their level of *dependency*, using experimental data. According to [1] data-driven methods are very popular techniques due to their simplicity, their speed and their accuracy. These algorithms have been classified in three major families in [1], depending on the method used to estimate the dependency between genes. 1) The first family of methods is based on the assumption that a TG and a TF regulating its expression should exhibit correlated gene expressions, and thus rely on correlation scores to infer regulatory links (e.g., [4]). 2) The second family of methods relies on information theory scores such as Mutual Information, to capture more complex relationships between TFs and TGs, which cannot be apprehended by linear correlations (e.g., [5]). 3) The last family reported in [1] is based on feature importance scores assigned by regression algorithms that are trained to predict the expressions of a TG from those of TFs (e.g., [6, 7]).

Multi-Network methods Unlike the previous families, the methods belonging to this family tackle the GRN inference problem by combining heterogeneous data sources such as gene expression data, TF binding site motifs, or Chromatin Immuno-Precipitation data. For instance a recent method called SCENIC [8], refines the output of the GENIE3 data-driven method [6], using cis-regulatory TF binding site motif analysis. Another recent method, called PANDA [9, 10] reconstructs GRNs by integrating heterogeneous sources of data, using a message passing approach.

Classification applied to GRN inference So far, some studies have used classification algorithms to infer GRNs in a *supervised* way. For instance, Support Vector Machine (SVM) classifiers (e.g., [11, 12]), have been used to reverse-engineer GRNs. In these previous works, the GRN inference problem is seen as a binary classification task: given a TG and a TF, the task consists in classifying whether their interaction is true or not. Such methods depend to a large extent on their training datasets, and even if an increasing number of datasets describing regulatory interactions is available, unsupervised approaches remain necessary [13]. The paradigm presented in this paper differs from these previous approaches, in the sense that it is not a supervised approach requiring training datasets. Indeed, in our framework classification algorithms are simply used to score the dependency between TFs and TGs from gene expression data, using an approach analogous to the regression-based one.

3 Problem Statement

Gene expression dataset Let $X \in \mathbb{R}^{I \times J}$ be a gene expression matrix (e.g., derived from RNAseq or Microarray experiments), such that $X_{i,j}$ represents the level of expression of *gene* i in *condition* j. Moreover let $X_{i,\cdot}$ denote the vector of the *gene* i expression level across all conditions, and let $X_{\cdot,j}$ be the vector of the expression level of all genes for *condition* j. Let J and I denote respectively the number of conditions (i.e., columns) and the number of genes (i.e., rows), considered in X. In most of the cases, tens of thousands of genes (I) are described in at most a few hundred conditions (J), and thus often $J \ll I$.

Gene regulatory network GRNs are traditionally represented as an oriented graph, which nodes represent genes and edges denote regulatory links between TFs and their TGs. More formally, let $TG = \{tg^1, \ldots, tg^I\}$ denote the set of I genes of a given organism, and let $TF \subset TG$ be the subset of genes corresponding to TFs. $G = \langle TG, E \rangle$ denotes a GRN. The set of nodes of G is simply the set of genes TG. And the set of oriented edges E represent the set of regulatory links, such that $(tf, tg) \in E$ indicates that the transcription factor $tf \in TF$ regulates the gene expression of target gene $tg \in TG$. Notice that a TF can also be the TG of another TF.

Data-Driven GRN inference Methods belonging to this family aim at using gene expression data to score all possible regulatory links, and then selecting the most promising edges only. Let $E^{full} = \{(tf, tg) \in TF \times TG \quad | \quad tf \neq tg\}$ be the set of all possible links between TFs and TGs (excluding self-loops). Let us consider a function $w : \mathbb{R}^{I \times J}, TF, TG \rightarrow \mathbb{R}$ such that $w(X, tf, tg)$ is the *dependency* score associated to the regulatory link $(tf, tg) \in E^{full}$, inferred from the gene expression dataset X. Then, a subset of regulatory links is usually chosen to define a putative GRN, by selecting for instance the k links with the highest scores. Therefore, the output of a data-driven inference algorithm strongly depends on the scoring function it relies on.

Data-Driven GRN inference is a particularly complex task, since it suffers from the problem known as the "curse of dimensionality" [14]. Indeed, the number of genes I and, consequently, the number of possible interactions between them, i.e., $|E|$, is often orders of magnitude larger than the number of experimental conditions J, which leads to a large number of possible GRNs that can explain the experimental gene expression data [14].

4 Preprocessing techniques

4.1 Standardization

As in many machine learning problems, an important preliminary step consists in standardizing the dataset. For instance in [15], the authors have shown the importance that standardization techniques, such as the well-known z-score transformation, may have on gene expression data analysis. In this work we tested three standardization methods based on z-scores, described more formally hereafter.

- **Z-score rows** replaces each entry of the gene expression matrix as follows $X_{i,j} \leftarrow \frac{X_{i,j} - \mu_i}{\sigma_i}$. Where $\mu_i = \sum_j X_{i,j}/J$ represents the average gene expression of *gene* i and $\sigma_i = \sqrt{\sum_j (X_{i,j} - \mu_i)^2/(J-1)}$ denotes its standard deviation. This standardization ensures that all genes have comparable levels of expression.
- **Z-score columns** replaces each entry of the gene expression matrix as follows: $X_{i,j} \leftarrow \frac{X_{i,j} - \mu_j}{\sigma_j}$. Where $\mu_j = \sum_i X_{i,j}/I$ denotes the average gene expression of genes in *condition* j and $\sigma_j = \sqrt{\sum_i (X_{i,j} - \mu_j)^2/(I-1)}$ represents its standard deviation. This standardization technique ensures that all conditions have comparable ranges of expression.
- **Polishing standardization** [16] applies iteratively the z-score standardization along columns and rows until convergence. In practice, a few iterations are usually sufficient to converge. Here we only ran 10 iterations, which revealed to be sufficient to reach convergence.

These standardization techniques have been implemented using Pandas Python 3.7 library [17].

4.2 Discretization

The discretization of gene expressions is another common preprocessing step for several GRN inference algorithms [18]. This procedure aims at partitioning continuous variables to discretized intervals. In this work we included five well-known discretization techniques for gene expression, studied in [19], and described formally hereafter.

- **EFD** (Equal Frequency Discretization) partitions, for each gene, the full range of observed gene expression values in K subsets of equal sizes using K-quantiles as boundaries. Specifically, let $X_{i,.}$ be the expression of *gene* i, EFD computes the K-quantiles of the $X_{i,.}$. Let Q_k denotes the k-th K-quantile of $X_{i,.}$ such that $Pr[x < Q_k] \leq k/K$, for $x \in X_{i,.}$. Then $\forall j \in \{1, \ldots, J\}$ $X_{i,j} \leftarrow k$ if $Q_{k-1} \leq X_{i,j} < Q_k$ for $k \in \{1, \ldots, K\}$.
- **EWD** (Equal Width Discretization) splits, for each gene, the full range of observed gene expression values in K bins with equal size. More formally let $X_{i,.}$ be the vector of expressions of *gene* i, and let $min(X_{i,.})$ and $max(X_{i,.})$ denote respectively the minimal and the maximal gene expression of *gene* i. EWD computes K equal width bins $\delta = (max(X_{i,.}) - min(X_{i,.}))/K$. Then $\forall j \in \{1, \ldots, J\}$ $X_{i,j} \leftarrow k$ if $\delta \times (k-1) \leq X_{i,j} < \delta \times k$ for $k \in \{1, \ldots, K\}$.
- **Rowkmeans** (Row Kmeans Discretization) applies the K-means clustering algorithm [20] to partition the gene expression values of *gene* i into K clusters, and then the values are discretized according to their cluster membership. More formally let $X_{i,j}$ be the expression of *gene* i in *condition* j, $\forall j \in \{1, \ldots, J\}$, the values $X_{i,j}$ are partitioned in K clusters. Let C_k denote the k-th cluster, and let μ_k be its centroid, we consider that clusters are sorted according to their centroid location, i.e., $\mu_1 < \mu_2, < \cdots < \mu_K$. Then $\forall j \in \{1, \ldots, J\}$ $X_{i,j} \leftarrow k$ if $X_{i,j} \in C_k$ for $k \in \{1, \ldots, K\}$
- **Cokmeans** (Column Kmeans Discretization) applies the Kmeans clustering [20] to partition the gene expression values of *condition* j into K clusters, and then the values are discretized according to their cluster membership, similarly as for Rowkmeans.
- **Bikmeans** (Bidirectional Kmeans Discretization) [18] computes the Cokmeans and the Rowkmeans discretizations for a given gene expression matrix, combining them in order to consider both genes and conditions at once. More formally, let $X_{i,j}$ denote the expression of *gene* i along *condition* j, let $X_{i,j}^{Cokmeans}$ and $X_{i,j}^{Rowkmeans}$ denote respectively the discretized values of $X_{i,j}$ using the Cokmeans and the Rowkmeans techniques, with a parameter $K+1$. Then the final Bikmeans discretized value of $X_{i,j}$ is simply $\lfloor \sqrt{X_{i,j}^{Rowkmeans} \times X_{i,j}^{Colkmeans}} \rfloor$, where $\lfloor x \rfloor = max\{z \in \mathbb{Z} \quad | \quad z \leq x\}$ denotes the floor function.

The three discretization techniques tested in this work were implemented using scikit-learn Python 3.7 library [21],

5 Classification-based GRN inference

5.1 Definition

Classification and regression oriented GRN inference methods rely on similar principles. Indeed, in both cases the core idea is to train an algorithm to predict the expression level of a TG, from the expressions of a set of TFs. Then, the contribution of each TF to the prediction task is computed, as a feature importance score. Finally such scores can be directly used as proxies to quantify the *dependency* between the TG and each TF. The difference between the regression and classification oriented approaches is that while a regression algorithm can be trained directly on continuous gene expression data, it is necessary to discretize the TG expressions to form classes to train a classification algorithm.

More formally, let $tg^{(i)}$ denote a TG, and let $X_{i,.}$ be its continuous gene expression vector. Moreover let TF be the set of TFs (different from $tg^{(i)}$), and let $X_{TF,.}$ denote their expression across all experimental conditions. Let $y^{(i)} = Disc(X_{i,.})$ denote the discretized vector of gene expressions of $tg^{(i)}$, where $Disc : \mathbb{R}^J \rightarrow \mathcal{Y}^J$ is simply a discretization function mapping each continuous element of a vector into a set $\mathcal{Y} = \{1, 2, \ldots, K\}$, where K denotes the number of classes (or bins). In practice any method presented in Section 4, can be used to discretize gene expressions. Let $\Phi_\theta : \mathbb{R}^{|TF|} \rightarrow \mathcal{Y}$ denote a classification model with parameters $\theta \in \Theta$, that aims at predicting, the discrete gene expressions $y_j^{(i)}$ of $tg^{(i)}$ along each *condition* j, from the TFs gene expressions $X_{TF,j}$ the same condition. In this context, each $tf \in TF$ is considered as a predictive feature. In practice the classifier Φ_θ is trained by finding a set of parameters θ that minimize some kind of average classification error, on the training dataset, i.e., $argmin_\theta \frac{\sum_j error(y_j^{(i)}, \Phi_\theta(X_{TF,j}))}{J}$.

Let us consider a function $\Gamma : \Theta, \mathbb{R}^{|TF| \times J}, \mathbb{N}^J \rightarrow \mathbb{R}^{|TF|}$ that computes $\gamma^{(i)} = \Gamma(\theta, X_{TF,.}, y^{(i)})$ the vector of *importance* of predictive features (i.e., each $tf \in TF$) relatively to the classification task achieved by Φ_θ, from θ the parameters of the classification model, the training dataset $X_{TF,.}$ and the target vector (i.e., discretize expressions of TG $tg^{(i)}$). Then, the *importance* of $tf^{(j)}$ in the prediction of $y^{(i)}$ is simply $\gamma_j^{(i)}$, and this measure can be used to approximate $w(X_{,}, tf^{(j)}, tg^{(i)})$ the *dependency* between $tf^{(j)}$ and $tg^{(i)}$, as defined in Section 3.

5.2 Practical implementation

Target gene expression discretization Since classification algorithms require discrete target variables, we discretized each TG expression vector into K discrete levels (classes) of expression, using the k-means algorithm. The number of classes K was determined using the well-known elbow method: For different values K ranging from 2 to 30, we ran k-means to cluster the gene expressions into K clusters, and then we computed the average sum of square distance (SSE) between the gene expressions and the mean expression of their corresponding clusters. Finally we applied the Kneedle algorithm [22], to locate the elbow in the plot representing the SSE for different number of clusters K. The elbow is a classical indicator that determines an appropriate number of clusters. Given the location of this indicator, as depicted in Figure 1, in this work the number of discrete levels of expression was set to $K = 5$.

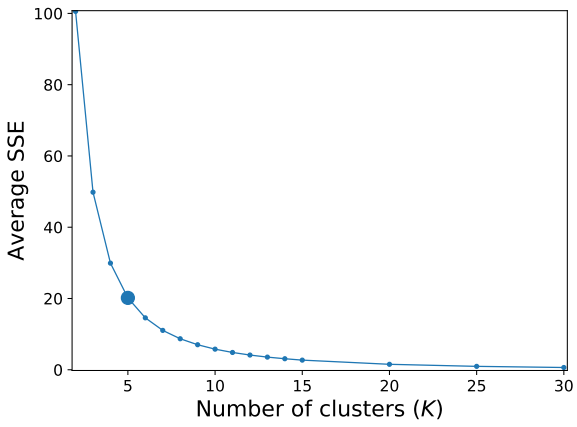

Fig. 1: Average sum of square distance (SSE) between gene expressions and clusters mean expression for different number of clusters K. The plot exhibits an elbow for $K = 5$ clusters.

Classification algorithms In this work we decided to test five well-known classification algorithms, which are implemented in the scikit-learn Python 3.7 library [21] (version 0.20.1). Four of these algorithms are ensemble methods based on decision trees, namely Random Forest (RF) [23], Extremely Randomized Trees (XRT) [24], AdaBoost (AB) [25] and Gradient Boosting (GB) [26]; and the last algorithm is the One-vs-All linear multi-class SVM classifier [27].

For a decision tree, the importance of a feature d is computed by summing the weighted impurity decrease (e.g., GINI, entropy, variance) for each split defined along d; then for a set of trees, the importance of d is simply the average importance over all trees [23]. In practice, the feature importance was obtained from the dedicated scikit-learn model's `feature_importances_` attributes.

In the case of the One-vs-All linear Support Vector Machine (SVM), a linear binary SVM is trained to build a hyperplane that separates each class (i.e., TG's level of expression) form other classes [27]. For a linear SVM, the absolute value of the coordinate of the hyperplane orthogonal vector, along a given feature d, is considered here as the importance of feature d; Since it indicates the participation of feature d, in the class separation, relatively to other features. For each feature (i.e., TF), its final importance is simply the average importance over all binary SVMs. In practice, the hyperplane orthogonal vector coordinates were obtained from the SVM scikit-learn classifier's `coef_` attribute.

6 Experimental Setup

6.1 Datasets

All DREAM5 benchmark datasets used in [2], contain a gene expression matrix, a list of putative TFs, and a gold standard GRN reporting known regulatory interaction between TFs and their TGs. Table 1 summarizes the major characteristics of this

dataset, namely, the number of genes (I), the number of experimental conditions (J), the number of TFs ($|TF|$); as well as the major caracteristics of the gold standard such as its number of TFs ($|TF_{gold}|$), TGs ($|TG_{gold}|$) and regulatory links ($|E_{gold}|$). Hereafter we describe briefly each benchmark dataset.

Tab. 1: Benchmark datasets summary

Dataset	Data	J	I	$\|TF\|$	$\|TF_{gold}\|$	$\|TG_{gold}\|$	$\|E_{gold}\|$
In silico	Simulated	805	1,643	195	178	1,565	4,012
S. aureus	Microarray	160	2,810	99	38	446	515
E. coli	Microarray	805	4,511	334	141	1,081	2,066
S. cerevisiae	Microarray	536	5,950	333	114	1,994	3,940

E. coli In order to define the gold standard GRN for *E. coli*, the authors of [2] collected a set of regulatory links with strong evidence, from the manually curated RegulonDB 6.8 database [28]. The gene expression matrix provided in [2], is a normalized Microarray dataset constituted of Affymetrix platform chips, downloaded from Gene Expression Omnibus (GEO)[2]. According to [2], the Microarray dataset was normalized and filtered using: Robust Multichip Averaging [29], background adjustment, quantile normalization, probeset median polishing and, finally, the normalized gene expression values were transformed into a logarithmic scale. Finally, the list of TFs provided in [2], is the union of the TFs list provided by RegulonDB 6.8 database [28], and a list that the authors derived from Gene Ontology (GO) annotations.

S. cerevisiae The *S. cerevisiae* gold standard GRN used in this paper was generated by MacIsaac *et al.* [30] by analyzing ChIP-chip datasets, and characterizing the presence of conserved TF binding sites motifs. The associated gene expression matrix provided in [2], is a Microarray dataset constituted of Affymetrix platform chips downloaded from the GEO website. This Microarray dataset underwent the same normalization and filtering steps described for the *E. coli* dataset. The list of potential TFs provided by [2], is the union of the comprehensive list presented in [31] and a set of TFs identified using GO terms.

S. aureus Unlike the previous datasets, no experimentally validated gold standard GRN was available for *S. aureus*. Thus, in [2], the authors used RegPrecise [32], a database of regulatory interactions in prokaryotes, as a gold standard proxy to study the quality of the inference algorithms. As well as for the previous benchmark datasets, the *S. aureus* gene expression matrix provided in [2], is a Microarray dataset constituted of Affymetrix platform chips downloaded from the GEO website. This Microarray dataset underwent the same normalization and filtering steps, that were described for the *E. coli* dataset. Finally a putative list of TFs was identified in [2], considering the GO annotation of *S. aureus* genes.

In silico Unlike the previous benchmark datasets, this one has been generated by an *in silico* GRN model, generated using the fourth version of the GeneNetWeaver [33] software. According to [2], this synthetic GRN shares the core structure of the RegulonDB *E. coli* GRN, and incorporates 10% of random regulatory links. Finally

[2] http://www.ncbi.nlm.nih.gov/geo

the gene expression data was generated numerically, using this *in silico* GRN as a simulator, and relying on a dedicated dynamical model based on Ordinary Differential Equations, based on a multiplicative regulatory interactions hypothesis.

6.2 Evaluation Procedure

6.2.1 General procedure

In order to assess quality of the inference methods presented in this paper, with respect to gold standard GRNs, we used the evaluation framework presented in [2]. In this framewoork, GRN inference is evaluated as a binary classification task, which aims at predicting the presence of true regulatory links.

Gold standard GRNs from real datasets, report lists of experimentally verified regulatory links between studied TFs and TGs. All these links are considered as *true interactions*, for the binary classification task. At first glance, one could think about considering all other possible links, that are not reported in the gold standards, as false interactions. Nevertheless, according to [2], gold standard links are only an incomplete subset of all the true regulatory interactions involved in the organism. Hence, all missing links should not be considered as false interactions. Indeed, one should avoid penalizing inference methods for detecting true interactions, that have not been experimentally verified yet. In order to choose a suitable set of false interactions, in [2] the authors decided to exclude any link involving a TF or a TG that has not been studied experimentally. And only the links between pairs of experimentally studied TF and TG, that have not been reported in the gold standards, are considered as *false interactions*, for the binary classification task.

More formally, let TG denote the set of genes of a given organism, and let $TF \subset TG$ denote the subset of genes corresponding to TFs. Moreover, let $G_{gold} = \langle TF_{gold} \cup TG_{gold}, E_{gold} \rangle$ be an oriented graph representing a gold standard GRN. Where $TF_{gold} \subseteq TF$ and $TG_{gold} \subseteq TG$ represent respectively the set of TFs and the set of TGs from the gold standard; While E_{gold} represents the set of true regulatory interaction, such that $\forall (tf_{gold}, tg_{gold}) \in E_{gold}, \quad tf_{gold} \in TF_{gold}$ and $tg_{gold} \in TG_{gold}$ denotes a true regulation of tg_{gold} by tf_{gold}. Let $E^{full}_{gold} = \{(e_1, e_2) \in TF_{gold} \times (TF_{gold} \cup TG_{gold}) \quad | \quad e_1 \neq e_2\}$ be the set of all possible links between TFs and other genes from the gold standard (excluding self-loops). For the evaluation, all edges in E_{gold} are considered as *true interactions* while edges in $E^{full}_{gold} \setminus E_{gold}$ are considered as *false interactions*. Notice that any interaction in $(TF \times TG) \setminus E^{full}_{gold}$ is not considered in the evaluation process.

6.3 Evaluation Measures

Evaluation measures As in [2] we assessed the methods using standard evaluation measures for binary classification, from the machine learning community, namely the Area Under the Receiver Operating Characteristic curve (AUROC) [34], and the Area Under the Precision Recall curve (AUPR) [35] values.

p-values In order to make sense of the evaluation results, we proceeded to compare the performance of the inference methods with those that could be achieved by random predictions. To do so we computed empirical p-values for AUROC and AUPR values, following the procedure described in [2], as described hereafter. For each gold standard, null empirical distributions of AUROC and AUPR values, termed respectively H_{AUROC} and H_{AUPR}, were computed for 25000 random GRNs. Such

random GRNs were built by assigning random scores to all possible regulatory links. Then, the probability density functions of the empirical distributions were modeled by fitting a beta probability density function, in order to extrapolate the probability distributions beyond the empirical values range. Then for x_{AUROC} and x_{AUPR}, an AUROC value and an AUPR value respectively, the corresponding p-values are defined as $p_{AUROC} = \Pr(x \geq x_{AUROC}|H_{AUROC})$ and $p_{AUPR} = \Pr(x \geq x_{AUPR}|H_{AUPR})$.

Quality scores Finally, quality scores for each method along each dataset were computed by taking the opposite of the decimal logarithm of the p-values: $score_{AUROC} = -log_{10}(p_{AUROC})$ and $score_{AUPR} = -log_{10}(p_{AUPR})$. Notice that higher scores are associated to high evaluation measures, i.e., measures are less likely to be obtained by a random scoring.

6.4 Experimental protocol

6.4.1 Comparison with DREAM5 competitors

Gene expression data preprocessing Several methods that competed in the DREAM5 challenge, as well as state-of-the-art approaches (e.g., [6, 7]), apply a simple z-score standardization to each row of the gene expression matrix, in order to center and scale the expression profile of each gene. In order to assess our method, we decided to standardize the benchmark gene expressions using the same preprocessing technique.

Parameter settings Each classification algorithm that has been used to score regulatory links relies on different meta parameters. One of the most important parameters of the methods based on sets of decision trees, is the number of such estimators that should be considered. Using many estimators, require more computational power, but tends to lead to better results. Here, we have set the number of estimators to 100 decision trees for each method. The remaining parameters were left to their default values, as established in their respective implementations on the 0.20.1 version of the scikit-learn Python 3.7 library [21].

DREAM5 participants results In [2] the 35 methods that participated in the DREAM5 challenge were categorized in 6 families, namely 1) **MI:** data-driven methods based on Mutual Information, 2) **Correlation:** data-driven methods based on correlation metrics, 3) **Regression:** data-driven methods based on regression methods, 4) **Bayesian:** model-based Bayesian networks, 5) **Meta:** Multi-network methods, 6) **Others:** Different methods including model-based methods based on Boolean networks, or Genetic Algorithms. Combining the results of all the participants, the authors of [2] inferred a single robust and high-quality GRN, which was termed **Community**. The performance measures obtained by each participant on each benchmark dataset, as defined in Section 6.2, have been made available by [2].

Each one of our inference methods was also executed on the benchmark gene expression datasets, and the inferred GRNs were evaluated against the corresponding gold standards networks, using the procedure presented in Section 6.2.

6.4.2 Impact of preprocessing techniques

In this paper we also assessed the impact of the preprocessing techniques presented in Section 4, on the performance of the methods proposed here. The three standardiza-

tion techniques do not need to be parameterized, while the number of bins (discrete classes) is the only parameter of the discretization techniques. In practice, we set the number of bins equal to five, since GRN inference methods tested in [18], performed better with this number of bins. Then, each inference method was tested on the preprocessed datasets, and the resulting GRNs were evaluated with respect to the gold standards. Control experiments were also conducted by running the inference methods on raw data directly. In order to evaluate the impact of the preprocessing steps, we computed the gains in terms of quality measures, with respect to the control experiment.

All experiments were executed on a 2 GHz Intel Core i7 CPU running MacOS Mojave 10.14.5.

7 Experimental Results

7.1 Comparison with DREAM5 competitors

In order to assess our methods, we have computed the AUROC and the AUPR metrics, as well as the corresponding p-values and scores as described in Section 6.2. Then, we proceeded to compare our results to those reported in [2], for the different families of methods. As depicted in table 2, classification based methods output in average the best results in terms of AUROC and AUPR metrics and scores, even surpassing the average results obtained by the community approach. Moreover, the promising results of classification based methods does not seem to depend on a specific dataset, since their results are among the best for the four benchmark datasets, and they outcompete the community GRN for the three real organisms, as shown in Figures 2 and 3.

As depicted in Figure 4, the different methods score differently depending on the benchmark dataset, and there is no ever-winning method among the classification based inference algorithms. For instance, the method based on SVM outputs the best results for *E. coli* and *S. cerevisiae* datasets both in terms of AUROC and AUPR, while it performs worse than the other four methods on the *In silico* dataset. An analogous phenomenon has also been observed in the case of the DREAM5 competitors, as reported in [2].

Tab. 2: Average values and scores for AUROC and AUPR metrics, and number of methods for each GRN inference family.

	AUROC		AUPR		Number of
	Values	Scores	Values	Scores	Methods
Classification	0.67	70.05	0.18	95.60	5
Community	0.64	35.62	0.13	47.94	1
Others	0.58	9.97	0.06	8.54	8
MI	0.60	9.53	0.09	9.09	5
Meta	0.60	7.69	0.09	13.77	5
Regression	0.59	7.17	0.09	18.70	8
Correlation	0.59	4.65	0.08	5.65	3
Bayesian	0.56	0.71	0.05	3.16	6

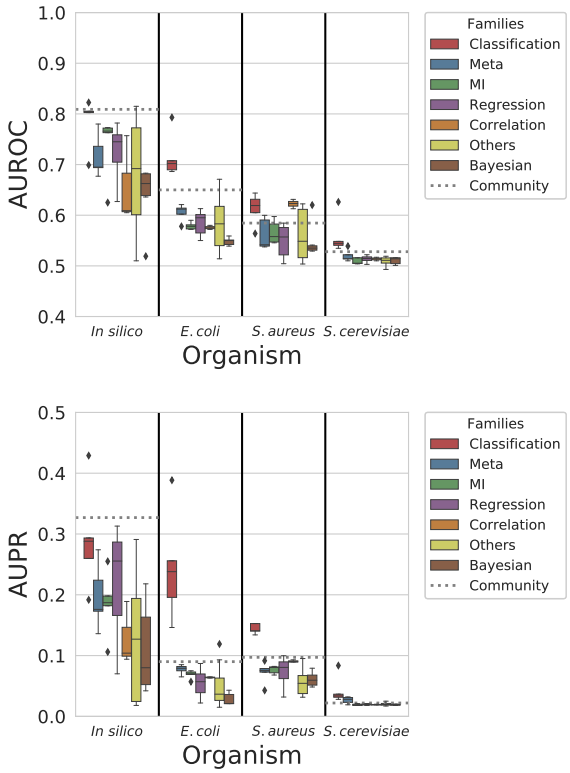

Fig. 2: Boxplots representing AUROC and AUPR values for each dataset and each family of GRN inference methods. Classification based approaches are among the best for all datasets, and they out-compete the community GRN for the three real organisms.

7.2 Impact of preprocessing techniques

As described in the Section 6.4.2, we have studied the impact of eight well-known preprocessing techniques, on the methods performance. The impact of these methods was evaluated by computing the gains in terms of AUROC and AUPR values and scores, with respect to control performance obtained on raw data. In this case, a negative (resp. positive) gain means that the results obtained on preprocessed data are worse (resp. better) than those obtained on raw data. The gains in AUROC range from -0.12 to 0.029, with an average gain equal to -0.013, and a standard deviation equal to 0.018. While the gains for AUPR range from -0.07 to 0.02, with an average gain equal to -0.019 and a standard deviation equal to 0.028. Therefore applying preprocessing techniques to these datasets tends to lead, in average, to a small degradation of the inference quality. Nevertheless, we note that these datasets already underwent a pipeline of sophisticated normalization and filtering steps, and consequently the impact of further preprocessings are likely to be less important. Average gains in AUROC and AUPR values, for each pair of inference method and preprocessing technique,

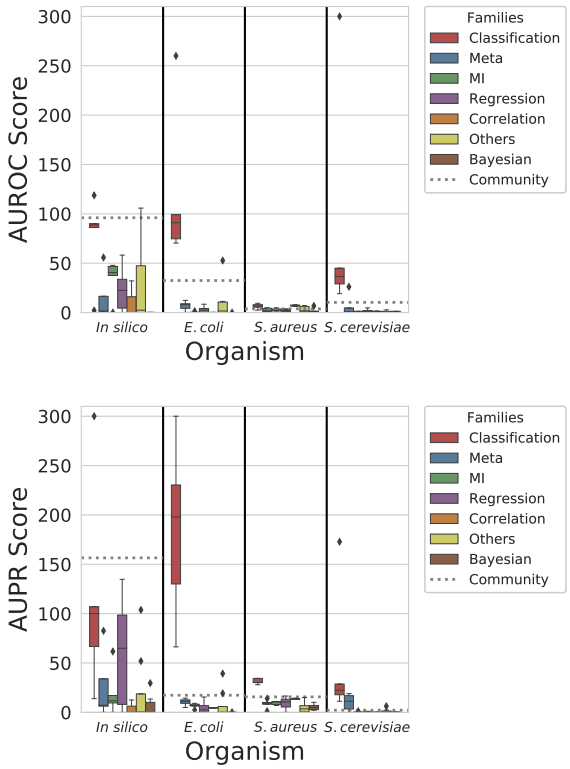

Fig. 3: Boxplots representing AUROC and AUPR scores for each dataset and each family of GRN inference methods. Classification based approaches are among the best for all datasets, and they out-compete the Community GRN for the three real organisms.

are depicted as cluster-maps in Figure 5. This Figure shows that the vectors of gain of the four decision tree based methods tend to cluster together, showing that these techniques behave similarly to similar changes in the dataset, while the SVM based approach tends to behave rather differently. This last method tends to perform better on datasets that underwent row z-score or polishing standardizations. While decision tree based methods, obtained consistently the highest gains, on datasets that were discretized using the EFD technique. This last result is consistent to the conclusions of some studies (e.g., [19, 36]) that affirm that discretizing continuous variables may lead some machine learning algorithms to produce more accurate models.

8 Conclusion

In this paper we presented a framework of data-driven GRN inference methods based on well-known classification algorithms, that includes five new GRN inference methods. These new techniques have been compared to well-established approaches on bench-

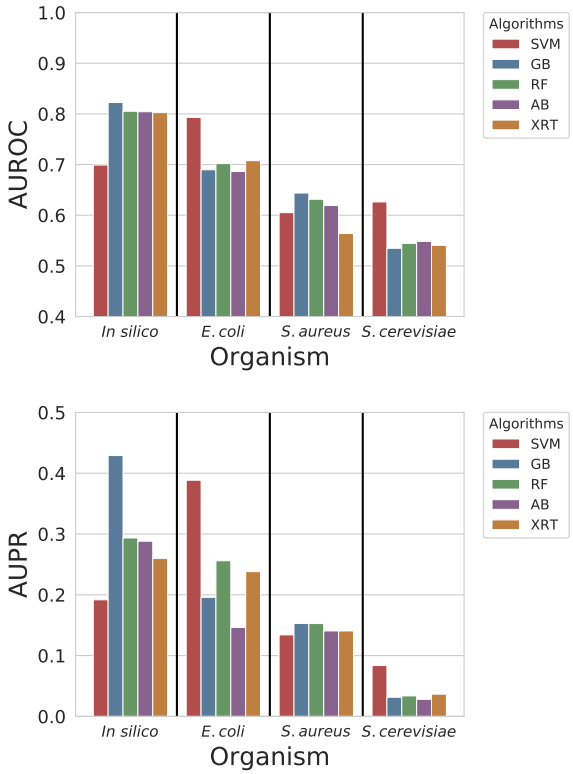

Fig. 4: Barplots representing the AUROC and the AUPR associated to each classification algorithm, for each benchmark dataset.

mark datasets, using the evaluation procedure described in [2]. The results showed that inference methods based on classification algorithms exhibit satisfactory results, outperforming other families on average. These promising results suggest that a data-driven GRN inference paradigm relying on classification methods is an interesting complementary tool for the community. Future work perspectives include i) Running further analysis on RNAseq datasets, more complex organisms (e.g., [37]) and running a parameter sensitivity analysis, ii) Incorporating and studying new classification based methods, iii) Combining the results of several classification based techniques in an ensemble learning manner, to get more robust results as suggested in [2].

References

[1] G. Sanguinetti and V. A. Huynh-Thu, "Gene regulatory network inference: an introductory survey," in *Gene Regulatory Networks*. Springer, 2019, pp. 1–23.

[2] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, D. Consortium, M. Kellis, J. J. Collins, and G. Stolovitzky, "Wis-

dom of crowds for robust gene network inference," *Nature methods*, vol. 9, no. 8, p. 796, 2012.

[3] L. E. Chai, S. K. Loh, S. T. Low, M. S. Mohamad, S. Deris, and Z. Zakaria, "A review on the computational approaches for gene regulatory network construction," *Computers in biology and medicine*, vol. 48, pp. 55–65, 2014.

[4] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Statistical applications in genetics and molecular biology*, vol. 4, no. 1, 2005.

[5] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, "Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles," *PLoS biology*, vol. 5, no. 1, p. e8, 2007.

[6] A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PloS one*, vol. 5, no. 9, p. e12776, 2010.

[7] A.-C. Haury, F. Mordelet, P. Vera-Licona, and J.-P. Vert, "Tigress: trustful inference of gene regulation using stability selection," *BMC systems biology*, vol. 6, no. 1, p. 145, 2012.

[8] S. Aibar, C. B. González-Blas, T. Moerman, H. Imrichova, G. Hulselmans, F. Rambow, J.-C. Marine, P. Geurts, J. Aerts, J. van den Oord *et al.*, "Scenic: single-cell regulatory network inference and clustering," *Nature methods*, vol. 14, no. 11, p. 1083, 2017.

[9] K. Glass, C. Huttenhower, J. Quackenbush, and G.-C. Yuan, "Passing messages between biological networks to refine predicted interactions," *PloS one*, vol. 8, no. 5, p. e64832, 2013.

[10] K. Glass, J. Quackenbush, and J. Kepner, "High performance computing of gene regulatory networks using a message-passing model," in *2015 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2015, pp. 1–6.

[11] F. Mordelet and J.-P. Vert, "Sirene: supervised inference of regulatory networks," *Bioinformatics*, vol. 24, no. 16, pp. i76–i82, 2008.

[12] Z. Gillani, M. S. H. Akash, M. M. Rahaman, and M. Chen, "Comparesvm: supervised, support vector machine (svm) inference of gene regularity networks," *BMC bioinformatics*, vol. 15, no. 1, p. 395, 2014.

[13] L. Cerulo, C. Elkan, and M. Ceccarelli, "Learning gene regulatory networks from only positive and unlabeled data," *BMC bioinformatics*, vol. 11, no. 1, p. 228, 2010.

[14] M. Banf and S. Y. Rhee, "Computational inference of gene regulatory networks: approaches, limitations and opportunities," *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, vol. 1860, no. 1, pp. 41–52, 2017.

[15] C. Cheadle, M. P. Vawter, W. J. Freed, and K. G. Becker, "Analysis of microarray data using z score transformation," *The Journal of molecular diagnostics*, vol. 5, no. 2, pp. 73–81, 2003.

[16] R. A. Olshen and B. Rajaratnam, "Successive normalization of rectangular arrays," *Annals of statistics*, vol. 38, no. 3, p. 1638, 2010.

[17] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed 20-06-2019]. [Online]. Available: http://www.scipy.org/

[18] Y. Li, L. Liu, X. Bai, H. Cai, W. Ji, D. Guo, and Y. Zhu, "Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks," *BMC bioinformatics*, vol. 11, no. 1, p. 520, 2010.

[19] S. Jung, Y. Bi, and R. V. Davuluri, "Evaluation of data discretization methods to derive platform independent isoform expression signatures for multi-class tumor subtyping," *BMC genomics*, vol. 16, no. 11, p. S3, 2015.

[20] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[22] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a" kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st international conference on distributed computing systems workshops*. IEEE, 2011, pp. 166–171.

[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[25] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[26] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] S. Gama-Castro, H. Salgado, M. Peralta-Gil, A. Santos-Zavaleta, L. Muniz-Rascado, H. Solano-Lira, V. Jimenez-Jacinto, V. Weiss, J. S. Garcia-Sotelo, A. Lopez-Fuentes *et al.*, "Regulondb version 7.0: transcriptional regulation of escherichia coli k-12 integrated within genetic sensory response units (gensor units)," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D98–D105, 2010.

[29] B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, vol. 19, no. 2, pp. 185–193, 2003.

[30] K. D. MacIsaac, T. Wang, D. B. Gordon, D. K. Gifford, G. D. Stormo, and E. Fraenkel, "An improved map of conserved regulatory sites for saccharomyces cerevisiae," *BMC bioinformatics*, vol. 7, no. 1, p. 113, 2006.

[31] C. Zhu, K. J. Byers, R. P. McCord, Z. Shi, M. F. Berger, D. E. Newburger, K. Saulrieta, Z. Smith, M. V. Shah, M. Radhakrishnan *et al.*, "High-resolution dna-binding specificity analysis of yeast transcription factors," *Genome research*, vol. 19, no. 4, pp. 556–566, 2009.

[32] P. S. Novichkov, O. N. Laikova, E. S. Novichkova, M. S. Gelfand, A. P. Arkin, I. Dubchak, and D. A. Rodionov, "Regprecise: a database of curated genomic inferences of transcriptional regulatory interactions in prokaryotes," *Nucleic acids research*, vol. 38, no. suppl_1, pp. D111–D118, 2009.

[33] T. Schaffter, D. Marbach, and D. Floreano, "Genenetweaver: in silico benchmark generation and performance profiling of network inference methods," *Bioinformatics*, vol. 27, no. 16, pp. 2263–2270, 2011.

[34] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[35] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[36] S. Kotsiantis and D. Kanellopoulos, "Discretization techniques: A recent survey," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 47–58, 2006.

[37] D. Potier, K. Davie, G. Hulselmans, M. N. Sanchez, L. Haagen, D. Koldere, A. Celik, P. Geurts, V. Christiaens, S. Aerts *et al.*, "Mapping gene regulatory networks in drosophila eye development by large-scale transcriptome perturbations and motif inference," *Cell reports*, vol. 9, no. 6, pp. 2290–2303, 2014.

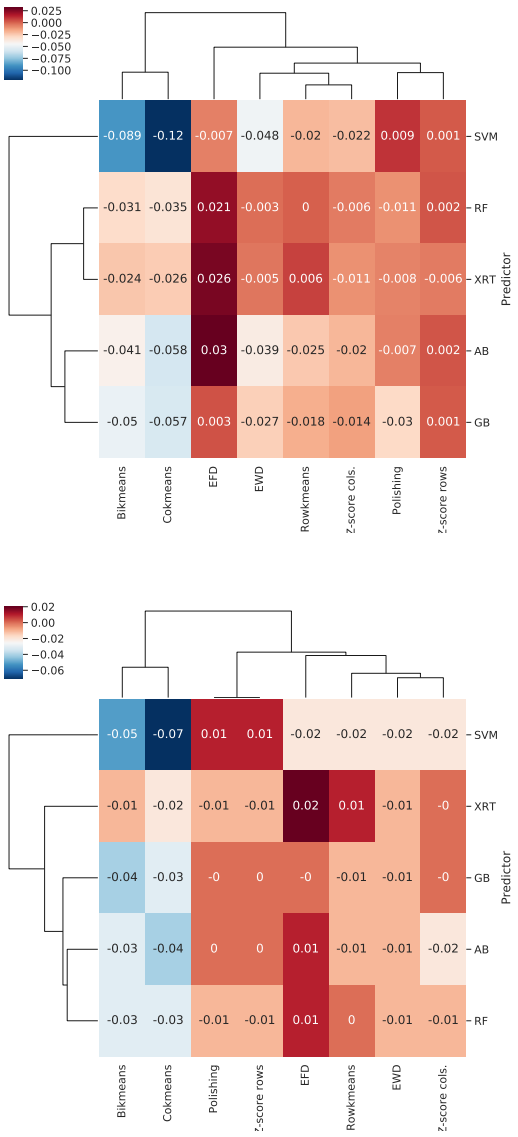

Fig. 5: Cluster-maps representing the average gain in AUROC and AUPR values for each combination of classifier (rows), and preprocessing (columns).