BMC Systems Biology

METHODOLOGY ARTICLE Open Access

ENNET: inferring large gene regulatory networks from expression data using gradient boosting

Janusz Sławek and Tomasz Arodź*

Abstract

Background: The regulation of gene expression by transcription factors is a key determinant of cellular phenotypes. Deciphering genome-wide networks that capture which transcription factors regulate which genes is one of the major efforts towards understanding and accurate modeling of living systems. However, reverse-engineering the network from gene expression profiles remains a challenge, because the data are noisy, high dimensional and sparse, and the regulation is often obscured by indirect connections.

Results: We introduce a gene regulatory network inference algorithm ENNET, which reverse-engineers networks of transcriptional regulation from a variety of expression profiles with a superior accuracy compared to the state-of-the-art methods. The proposed method relies on the boosting of regression stumps combined with a relative variable importance measure for the initial scoring of transcription factors with respect to each gene. Then, we propose a technique for using a distribution of the initial scores and information about knockouts to refine the predictions. We evaluated the proposed method on the DREAM3, DREAM4 and DREAM5 data sets and achieved higher accuracy than the winners of those competitions and other established methods.

Conclusions: Superior accuracy achieved on the three different benchmark data sets shows that ENNET is a top contender in the task of network inference. It is a versatile method that uses information about which gene was knocked-out in which experiment if it is available, but remains the top performer even without such information. ENNET is available for download from https://github.com/slawekj/ennet under the GNU GPLv3 license.

Keywords: Gene regulatory networks, Network inference, Ensemble learning, Boosting

Background

Regulation of gene expression is a key driver of adaptation of living systems to changes in the environment and to external stimuli. Abnormalities in this highly coordinated process underlie many pathologies. At the transcription level, the control of the amount of mRNA transcripts involves epigenetic factors such as DNA methylation and, in eukaryotes, chromatin remodeling. But the key role in both prokaryotes and eukaryotes is played by transcription factors (TF), that is, proteins that can bind to DNA in the regulatory regions of specific genes and act as repressors or inducers of their expression. Many interactions between transcription factors and genes they regulate have been discovered through traditional molecular biology experiments. With the introduction of high-throughput experimental techniques for measuring gene expression, such as DNA microarrays and RNA-Seq, the goal moved to reverse-engineering genome-wide gene regulatory networks (GRNs) [1]. Knowledge of GRNs can facilitate finding mechanistic hypotheses about differences between phenotypes and sources of pathologies, and can help in the drug discovery and bioengineering.

High throughput techniques allow for collecting genome-wide snapshots of gene expression across different experiments, such as diverse treatments or other perturbations to cells [2]. Analyzing these data to infer the regulatory network is one of the key challenges in the computational systems biology. The difficulty of this task arises from the nature of the data: they are typically

*Correspondence: tarodz@vcu.edu
Department of Computer Science, Virginia Commonwealth University, Richmond, Virginia

noisy, high dimensional, and sparse [3]. Moreover, discovering direct causal relationships between genes in the presence of multiple indirect ones is not a trivial task given the limited number of knockouts and other controlled experiments. Attempts to solve this problem are motivated from a variety of different perspectives. Most existing computational methods are examples of influence modeling, where the expression of a target transcript is modeled as a function of the expression levels of some selected transcription factors. Such a model does not aim to describe physical interactions between molecules, but instead uses inductive reasoning to find a network of dependencies that could explain the regularities observed among the expression data. In other words, it does not explain mechanistically how transcription factors interact with regulated genes, but indicate candidate interactions with a strong evidence in expression data. This knowledge is crucial to prioritize detailed studies of the mechanics of the transcriptional regulation.

One group of existing methods describes GRN as a system of ordinary differential equations. The rate of change in expression of a transcript is given by a function of the concentration levels of transcription factors that regulate it. Network inference includes two steps: a selection of a model and an estimation of its parameters. Popular models imply linear functions a priori [4-7]. Bayesian Best Subset Regression (BBSR) [8] has been proposed as a novel model selection approach, which uses Bayesian Information Criterion (BIC) to select an optimal model for each target gene. Another group of methods employ probabilistic graphical models that analyze multivariate joint probability distributions over the observations, usually with the use of Bayesian Networks (BN) [9-11], or Markov Networks (MN) [12]. Various heuristic search schemes have been proposed in order to find parameters of the model, such as greedy-hill climbing or the Markov Chain Monte Carlo approach [13]. However, because learning optimal Bayesian networks from expression data is computationally intensive, it remains impractical for genome-wide networks.

Other approaches are motivated from statistics and information theory. TwixTwir [14] uses double two-way t-test to score transcriptional regulations. The null-mutant z-score algorithm [15] scores interactions based on a z-score transformed knockout expression matrix. Various algorithms rely on estimating and analyzing cross-correlation and mutual information (MI) of gene expression in order to construct a GRN [16-20], including ANOVA η^2 method [21]. Improvements aimed at removing indirect edges from triples of genes have been proposed, including techniques such as the Data Processing Inequality in ARACNE [22,23], and the adaptive background correction in CLR [24]. Another method, NARROMI [25], eliminates redundant interactions from the MI matrix by applying ODE-based recursive optimization, which involves solving a standard linear programming model.

Recently, machine-learning theory has been used to formulate the network inference problem as a series of supervised gene selection procedures, where each gene in turn is designated as the target output. One example is MRNET [26], which applies the maximum relevance/minimum redundancy (MRMR) [27] principle to rank the set of transcription factors according to the difference between mutual information with the target transcript (maximum relevance) and the average mutual information with all the previously ranked transcription factors (minimum redundancy). GENIE3 [28] employs Random Forest algorithm to score important transcription factors, utilizing the embedded relative importance measure of input variables as a ranking criterion. TIGRESS [29] follows a similar approach but is based on the least angle regression (LARS). Recently, boosting [30,31] was also used to score the importance of transcription factors, in ADANET [32] and OKVAR-Boost [33] methods.

In this paper, we propose a method that combines gradient boosting with regression stumps, augmented with statistical re-estimation procedures for prioritizing a selected subset of edges based on results from the machine-learning models. We evaluated our method on the DREAM3, DREAM4 and DREAM5 network inference data sets, and achieved results that in all cases were better than the currently available methods.

Methods

The ENNET algorithm

Formulating the gene network inference problem

The proposed algorithm returns a directed graph of regulatory interactions between P genes in form of a weighted adjacency matrix $V \in \mathbb{R}^{P \times P}$, where $v_{i,j}$ represents regulation of gene j by gene i. As an input, it takes gene expression data from a set of experiments, together with the meta-data describing the conditions of the experiments, including which genes were knocked out. Usually, the raw expression data need to be pre-processed before any inference method could be applied to reverse-engineer a GRN. Pre-processing has a range of meanings, here it is regarded as a process of reducing variations or artifacts, which are not of the biological origin. It is especially important when the expression is measured with multiple high-density microarrays [34]. Concentration levels of transcripts must be adjusted and the entire distribution of adjusted values aligned with a normal distribution. Methods for normalization of expression data are outside of the scope of our work. The data we used were already normalized using RMA [34,35] by the DREAM challenge organizers. We further normalized the expression data to zero mean and unit standard deviation.

The network inference process relies heavily on the type of expression data provided as an input. Two main groups of expression profiles are: the one with known, and the one with unknown initial perturbation state of the expression of genes in the underlying network of regulatory interactions. For example, knockout and knockdown data are provided with the additional metadata, which describe which genes were initially perturbed in each experiment. On the other hand, multifactorial and time series data are usually expression profiles of an unknown initial state of genes. Wildtype, knockout, knockdown, and multifactorial data describe the expression of initially perturbed genes, which are however in a steady state at the time of measurement, whereas time series data describe the dynamics of the expression levels of initially perturbed genes. The types of data available in popular benchmark data sets are summarized in Table 1.

The variability of possible input scenarios poses a problem of representing and analyzing expression data. Here, we operate on an $N \times P$ expression matrix E, where $e_{i,j}$ is the expression value of the j-th gene in the i-th sample. Columns of matrix E correspond to genes, rows correspond to experiments. We also define a binary perturbation matrix K, where $k_{i,j}$ is a binary value corresponding to the j-th gene in the i-th sample, just like in the matrix E. If $k_{i,j}$ is equal to 1, it means that the j-th gene is known to be initially perturbed, for example knocked out, in the i-th experiment. Otherwise $k_{i,j}$ is equal to 0. If no information is available about knockouts, all values are set to 0.

Decomposing the inference problem into gene selection problems

We decompose the problem of inferring the network of regulatory interactions targeting all P genes into P independent subproblems. In each subproblem incoming edges from transcription factors to a single gene transcript are discovered. For the k-th decomposed subproblem we create a target expression vector Y_k and a feature expression matrix X_{-k}. Columns of the X_{-k} matrix constitute a set of possible transcription factors. Vector Y_k corresponds to the expression of the transcript, which is possibly regulated by transcription factors from X_{-k}. In a single gene selection problem we decide which TFs contribute to the target gene expression across all the valid experiments. Columns of X_{-k} correspond to all the possible TFs, but if a target gene k is also a transcription factor, it is excluded from X_{-k}. We do not consider a situation in which a transcription factor would have a regulatory interaction with itself. When building the target vector Y_k corresponding to the k-th target gene, $k \in \{1, ..., P\}$, we consider all the experiments valid except from the ones in which the k-th gene was initially perturbed, as specified in the perturbation matrix K. We reason that the expression value of the k-th gene in those experiments is not determined by its TFs, but by the external perturbation. Each row in the Y_k vector is aligned with a corresponding row in the X_{-k} matrix. In order to justify all the possible interactions we need to solve a gene selection problem for each target gene. For example, if a regulatory network consists of four genes ($P = 4$), we need to solve four gene selection problems. In the k-th problem, $k \in \{1, 2, 3, 4\}$, we find which TFs regulate the k-th target gene. In other words, we calculate the k-th column of the output adjacency matrix V.

Table 1 Different types of expression data provided in popular data sets

Data set	WT	KO	KD	MF	TS
DREAM3 size 100	●	●	●	○	●
DREAM4 size 100	●	●	●	○	●
DREAM4 size 100 MF	○	○	○	●	○
DREAM5⋆	●	●	●	●	●

Different types of expression data provided in popular data sets: WT- Wildtype, KO- Knockouts, KD- Knockdowns, MF- Multifactorial, TS- Time series, ● Available, ○ Unavailable. ⋆) Even though all the data types are available, they are all processed as MF.

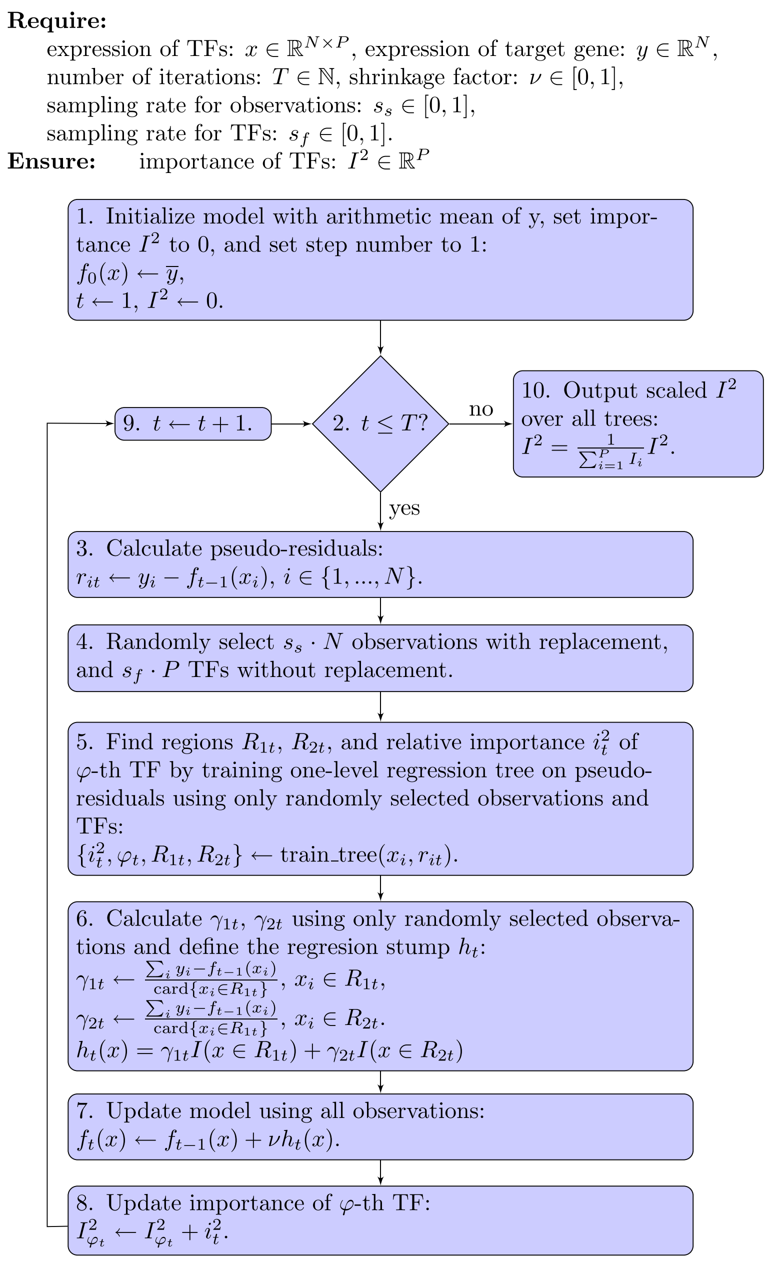
Solving the gene selection problems

Once the target gene expression vector Y_k and the TF expression matrix X_{-k} are created for each gene k, we solve each k-th gene selection problem independently, in the following way. We search for the subset of columns in X_{-k} that are related to the target vector Y_k by an unknown function f_k, as shown in Equation 1,

$$\forall k \in \{1, ..., P\}, \quad \exists f_k : Y_k = f_k(X_{-k}) + \epsilon_k, \qquad (1)$$

where ϵ_k is a random noise. A function f_k represents a pattern of regulatory interactions that drive the expression of the k-th gene. We want f_k to rely only on a small number of genes acting as transcription factors, those that are the true regulators of gene k. Essentially, this is a feature selection or a gene selection task [28,32,36,37], where the goal is to model the target response Y_k with an optimal small set of important predictor variables, i.e., a subset of columns of the X_{-k} matrix. A more relaxed objective of the gene selection is the variable ranking, where the relative relevance for all input columns of the X_{-k} matrix is obtained with respect to the target vector Y_k. The higher a specific column is in that ranking, the higher the confidence that a corresponding TF is in a regulatory interaction with the target gene k.

Our solution to the variable ranking involves ensemble learning. We use an iterative regression method, which in each iteration chooses one transcription factor based on an optimality criterion, and adds it to the non-linear regression ensemble. The main body of our method, presented in Figure 1, is based on Gradient Boosting Machine [38] with a squared error loss function. First, ENNET initializes f_0 to be an optimal constant model, without selecting any transcription factor. In other words, f_0 is initialized to an average of Y_k. At each next t-th step the algorithm creates an updated model f_t, by fitting a base learner h_t and adding it to the previous model f_{t-1}. The base learner is fitted to a sample of pseudo residuals, with respect to a sample of transcription factors, and thus is expected to reduce the error of the model. Pseudo-residuals are re-calculated at the beginning of each iteration with respect to the current approximation f_t. As a base learner, we use regression stumps, which select a single TF that best fits pseudo residuals. A regression stump $h_t(x)$ partitions the expression values x of a candidate TF into two disjoint regions R_{1t} and R_{2t}, where $R_{2t} = \mathbb{R} - R_{1t}$, and



Require:
expression of TFs: $x \in \mathbb{R}^{N \times P}$, expression of target gene: $y \in \mathbb{R}^N$,
number of iterations: $T \in \mathbb{N}$, shrinkage factor: $\nu \in [0, 1]$,
sampling rate for observations: $s_s \in [0, 1]$,
sampling rate for TFs: $s_f \in [0, 1]$.
Ensure: importance of TFs: $I^2 \in \mathbb{R}^P$

1. Initialize model with arithmetic mean of y, set importance I^2 to 0, and set step number to 1:
$f_0(x) \leftarrow \overline{y}$,
$t \leftarrow 1$, $I^2 \leftarrow 0$.

2. $t \leq T$?

no

10. Output scaled I^2 over all trees:
$I^2 = \frac{1}{\sum_{i=1}^{P} I_i} I^2$.

yes

3. Calculate pseudo-residuals:
$r_{it} \leftarrow y_i - f_{t-1}(x_i)$, $i \in \{1, ..., N\}$.

4. Randomly select $s_s \cdot N$ observations with replacement, and $s_f \cdot P$ TFs without replacement.

5. Find regions R_{1t}, R_{2t}, and relative importance i_t^2 of φ-th TF by training one-level regression tree on pseudo-residuals using only randomly selected observations and TFs:
$\{i_t^2, \varphi_t, R_{1t}, R_{2t}\} \leftarrow \text{train\_tree}(x_i, r_{it})$.

6. Calculate γ_{1t}, γ_{2t} using only randomly selected observations and define the regresion stump h_t:
$\gamma_{1t} \leftarrow \frac{\sum_i y_i - f_{t-1}(x_i)}{\text{card}\{x_i \in R_{1t}\}}$, $x_i \in R_{1t}$,
$\gamma_{2t} \leftarrow \frac{\sum_i y_i - f_{t-1}(x_i)}{\text{card}\{x_i \in R_{1t}\}}$, $x_i \in R_{2t}$.
$h_t(x) = \gamma_{1t} I(x \in R_{1t}) + \gamma_{2t} I(x \in R_{2t})$

7. Update model using all observations:
$f_t(x) \leftarrow f_{t-1}(x) + \nu h_t(x)$.

8. Update importance of φ-th TF:
$I_{\varphi_t}^2 \leftarrow I_{\varphi_t}^2 + i_t^2$.

9. $t \leftarrow t + 1$.

Figure 1 The flowchart of the ENNET algorithm. ENNET algorithm is a modification of a Gradient Boosting Machine algorithm, with a squared error loss function and a regression stump base learner. The algorithm calculates a vector of importance scores of transcription factors, which can possibly regulate a target gene. It is invoked *P* times in a problem of inferring a *P*-gene network, i.e., a *P*-column adjacency matrix *V*.

returns values γ_{1t} and γ_{2t}, respectively, for those regions, as shown in Equation 2,

$$h_t(x) = \gamma_{1t} I(x \in R_{1t}) + \gamma_{2t} I(x \in R_{2t}), \tag{2}$$

where I is the identity function returning the numerical 1 for the logical true, and the numerical 0 for the logical false. Regions R_{1t}, R_{2t} are induced such that the least-squares improvement criterion is maximized:

$$i^2(R_{1t}, R_{2t}) = \frac{w_{1t} w_{2t}}{w_{1t} + w_{2t}} (\gamma_{1t} - \gamma_{2t})^2, \tag{3}$$

where w_{1t}, w_{2t} are proportional to the number of observations in regions R_{1t}, R_{2t} respectively, and γ_{1t}, γ_{2t} are corresponding response means. That is, γ_{1t} is the average of the values from the vector of pseudo-residuals for those samples where an expression of the chosen TF falls into the region R_{1t}. The value of γ_{2t} is defined in an analogous way. The averages γ_{1t} and γ_{2t} are used as the regression output values for regions R_{1t} and R_{2t}, respectively, as shown in Equation 2. The criterion in Equation 3 is evaluated for each TF, and the transcription factor with the highest improvement is selected. In each t-th step, we only use a random portion of rows and columns of X_{-k}, sampled according to the observation sampling rate s_s, and the TF sampling rate s_f.

The procedure outlined above creates a non-linear regression model of the target gene expression based on the expression of transcription factors. However, in the network inference, we are interested not in the regression model as a whole, but only in the selected transcription factors. In each t-th step of the ENNET algorithm, only one TF is selected as the optimal predictor. The details of the regression model can be used to rank the selected TFs by their importance. Specifically, if a transcription factor φ_t is selected in an iteration t, an improvement i^2_t serves as an importance score $I^2_{\varphi_t}$ for that φ_t-th TF. If the same TF is selected multiple times at different iterations, its final importance score is a sum of the individual scores.

In the training of the regression model, the parameter ν, known as a shrinkage factor, is used to scale a contribution of each tree by a factor $\nu \in (0, 1)$ when it is added to the current approximation. In other words, ν controls the learning rate of the boosting procedure. Shrinkage techniques are also commonly used in neural networks. Smaller values of ν result in a larger training risk for the same number of iterations T. However, it has been found [38] that smaller values of ν reduce the test error, and require correspondingly larger values of T, which results in a higher computational overhead. There is a trade-off between these two parameters.

Refining the inferred network

Once the solutions of the independent gene selection problems are calculated, we compose the adjacency matrix V representing a graph of inferred regulatory interactions. Each of the solutions constitutes a single column-vector, therefore we obtain the adjacency matrix V by binding all the partial solutions column-wise. Then we apply a re-evaluation algorithm to achieve an improved final result. The first step does not require any additional data to operate other than the previously calculated adjacency matrix V. It exploits the variance of edge probabilities in the rows of V, i.e., edges outgoing from a single transcription factor, as a measure of the effect of transcriptional regulation. We score transcription factors based on their effects on multiple targets. We assume that the effect of transcriptional regulation on a directly regulated transcript is stronger than the one of the regulation on indirectly regulated transcripts, e.g. transcripts regulated through another transcription factor. Otherwise, knocking out a single gene in a strongly connected component in a network of regulatory interactions would cause the same rate of perturbation of the expression level of all the transcripts in that component. As a measure of that effect we use previously a calculated adjacency matrix V and multiply each row of V matrix by its variance σ_i^2. An updated adjacency matrix V^1 is given by Equation 4:

$$\forall (i,j) : v^1_{i,j} = \sigma_i^2 \cdot v_{i,j}, \tag{4}$$

where σ_i^2 is a variance in the i-th row of V. Note that V matrix is built column-wise, i.e., a single column of V contains the relative importance scores of all the transcription factors averaged over all the base learners with respect to a single target transcript. On the other hand, rows of V matrix are calculated independently in different subproblems of the proposed inference method. Each row of V contains relative importance scores with respect to a different target transcript. We reason that if a transcription factor regulates many target transcripts, e.g. a transcription factor is a hub node, the variance in a row of V corresponding to that transcription factor is elevated and therefore it indicates an important transcription factor.

The second step of refining the network requires knockout expression data. We reason that direct regulation of a transcript by a transcription factor would lead to a distinct signature in the expression data if the transcription factor was knocked out. A similar reasoning gave foundations for the null-mutant z-score method [15] of reverse-engineering GRNs. However, in the proposed method this step is only applied if knockout expression profiles are available. In this step we calculate an adjacency matrix V^2, which is an update to an already derived adjacency matrix V^1, as shown in Equation 5:

$$\forall (i,j) : v^2_{i,j} = \left| \frac{\overline{e_{\alpha(i),j}} - \overline{e_{\beta(i),j}}}{\sigma_j} \right| \cdot v^1_{i,j},$$
$$\alpha(i) = \{r : k_{r,i} \neq 0\}, \beta(i) = \{r : k_{r,i} = 0\}, \tag{5}$$

where $\overline{e_{\alpha(i),j}}$ is an average expression value of the j-th transcript in all the experiments $\alpha(i)$ in which the i-th gene was knocked-out, as defined by K matrix, $\overline{e_{\beta(i),j}}$ is the mean expression value for that transcript across all the other knockout experiments, $\beta(i)$, and σ_j is the standard deviation of the expression value of that transcript in all the knockout experiments. The $|\frac{\overline{e_{\alpha(i),j}}-\overline{e_{\beta(i),j}}}{\sigma_j}|$ coefficient shows how many standard deviations the typical expression of the j-th transcript was different from the average expression in the experiment in which its potential i-th transcription factor was knocked-out.

Performance evaluation

A considerable attention has been devoted in recent years to the problem of evaluating performance of the inference methods on adequate benchmarks [35,39]. The most popular benchmarks are derived from well-studied *in vivo* networks of model organisms, such as *E. coli* [40] and *S. cerevisiae* [41], as well as artificially simulated *in silico* networks [39,42-45]. The main disadvantage of *in vivo* benchmark networks is the fact that experimentally confirmed pathways can never be assumed complete, regardless of how well the model organism is studied. Such networks are assembled from known transcriptional interactions with strong experimental support. As a consequence, gold standard networks are expected to have few false positives. However, they contain only a subset of the true interactions, i.e., they are likely to contain many false negatives. For this reason, artificially simulated *in silico* networks are most commonly used to evaluate network inference methods. Simulators [39] mimic real biological systems in terms of topological properties observed in biological *in vivo* networks, such as modularity [46] and occurrences of network motifs [47]. They are also endowed with dynamical models of a transcriptional regulation, thanks to the use of non-linear differential equations and other approaches [42,48,49], and consider both transcription and translation processes in their dynamical models [48-50] using a thermodynamic approach. Expression data can be generated deterministically or stochastically and experimental noise, such as the one observed in microarrays, can be added [51].

Here, we used several popular benchmark GRNs to evaluate the accuracy of our proposed algorithm and compare it with the other inference methods. The data sets we used come from Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenges and are summarized in Table 1. We evaluated the accuracy of the methods using the Overall Score metric proposed by the authors of DREAM challenges [35], as shown in Equation 6:

$$\text{Overall Score} = -\frac{1}{2} \cdot \log_{10}(\overline{p}_{\text{aupr}} \cdot \overline{p}_{\text{auroc}}), \qquad (6)$$

where $\overline{p}_{\text{aupr}}$ and $\overline{p}_{\text{auroc}}$ are geometric means of p-values of networks constituting each DREAM challenge, relating to an area under the Precision-Recall curve (AUPR) and an area under the ROC curve (AUROC), respectively.

Results and discussion

We assessed the performance of the proposed inference algorithm on large, universally recognized benchmark networks of 100 and more genes, and compared it to the state-of-the-art methods. We summarize the results of running different inference methods in Figure 2. For a comparison we selected a range of established methods from literature: ARACNE, CLR, and MRNET as implemented in the *minet* R package [52], GENIE3 and C3NET as implemented by their respective authors, our previously reported method ADANET, and the top three performers in each of the three DREAM challenges as listed on the DREAM web site. Some of the methods were designed for use with knockout data, while others are developed with multifactorial data in mind, where no information is given about the nature of the perturbations. Therefore, depending on the nature of the particular DREAM data set, only the suitable group of methods is used for the comparison.

The accuracy of ENNET

DREAM3 [15,53,54] features *in silico* networks and expression data simulated using GeneNetWeaver software. Benchmark networks were derived as subnetworks of a system of regulatory interactions from known model organisms: *E. coli* and *S. cerevisiae*. In this study we focus on a DREAM3 size 100 subchallenge, as the largest of DREAM3 suite. The results of all the competing methods except those that are aimed at multifactorial problems are summarized in Table 2. ENNET and Yip et al. methods achieved the best Overall Scores for that subchallenge, as well as the best scores for all the individual networks. However, it is believed from the analysis of the later challenges [39] that Yip et al. method made a strong assumption on the Gaussian type of a measurement noise, which was used in DREAM3, but was no longer used in later DREAM challenges. For example, in DREAM4 challenge Yip et al. method was ranked 7th.

DREAM4 challenge [15,53,54] was posted one year after DREAM3 challenge. It features two large subchallenges: DREAM4 size 100, and DREAM4 size 100 multifactorial. For each subchallenge, the topology of the benchmark networks were derived from the transcriptional regulatory system of *E. coli* and *S. cerevisiae*. In DREAM4 size 100 subchallenge all the data types listed in Table 1 were available except multifactorial, therefore ADANET, GENIE3, CLR, C3NET, MRNET, and ARACNE methods were excluded from the comparison. The results of all the methods are summarized in Table 3. ENNET method

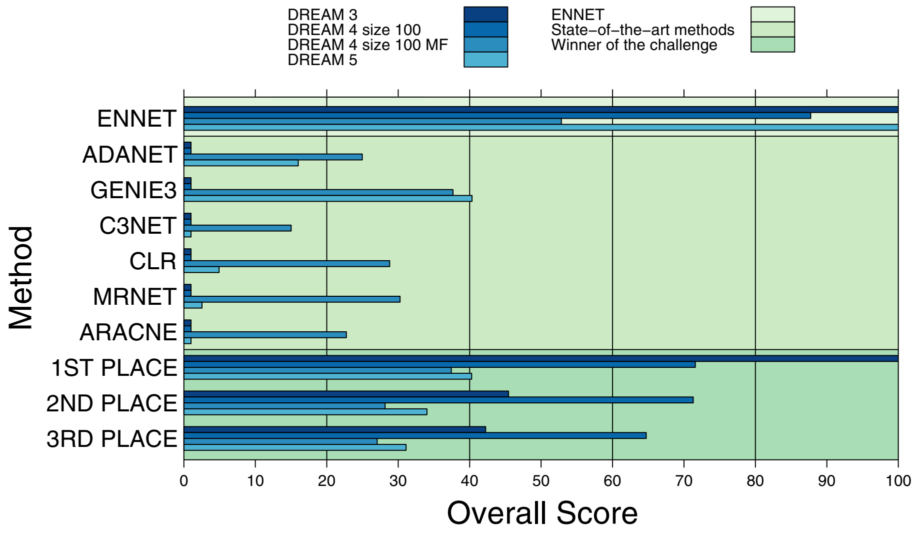


Figure 2 The Overall Score of GRN inference methods by data set. Results of the different inference methods on DREAM challenges. Results of the state-of-the-art methods were collected after running the algorithms with the default sets of parameters on pre-processed data. Results in the "Winner of the challenge" part of the figure correspond to the best methods participating in the challenge.

clearly outperformed all the others and achieved consistently high scores across all the benchmark networks. In the second DREAM4 large subchallenge, DREAM4 size 100 multifactorial, only multifactorial data were available, therefore all the methods were included in the comparison, and run as originally designed. The results of all the methods are summarized in Table 4. ENNET achieved the best Overall Score.

Three benchmark networks in DREAM5 [35] were different in size, and structured with respect to different model organisms. However, this time expression data of the only one network were simulated *in silico*, the two other sets of expression data were measured in real experiments *in vivo*. Like in all DREAM challenges, *in silico* expression data were simulated using an open-source GeneNetWeaver simulator [54]. However, DREAM5 was the first challenge where participants were asked to infer GRNs on a genomic scale, e.g. for thousands of target genes, and hundreds of known transcription factors. Gold standard networks were obtained from two sources: RegulonDB database [40], and Gene Ontology (GO) annotations [55]. The results of all the inference methods for DREAM5 expression data are summarized in Table 5. ENNET achieved the best score for the *in silico* network, and the best Overall Score, as well as the best individual AUROC scores for all the networks. Clearly all the participating methods achieved better scores for an *in silico* network than for either one of *in vivo* networks.

Table 2 Results of the different inference methods on DREAM3 networks, challenge size 100

Method	Network (AUPR/AUROC respectively)										Overall
	1		2		3		4		5		
	Experimental results										
ENNET	0.627	0.901	**0.865**	**0.963**	**0.568**	0.892	**0.522**	0.842	0.384	0.765	>**300**
	Winner of the challenge										
Yip et al.	**0.694**	**0.948**	0.806	0.960	0.493	**0.915**	0.469	**0.856**	**0.433**	**0.783**	>**300**
2nd	0.209	0.854	0.249	0.845	0.184	0.783	0.192	0.750	0.161	0.667	45.443
3nd	0.132	0.835	0.154	0.879	0.189	0.839	0.179	0.738	0.164	0.667	42.240

Results of the different inference methods on DREAM3 networks, challenge size 100. An area under the ROC curve (AUROC) and an area under the Precision-Recall curve (AUPR) are given for each network respectively. The overall Score for all the networks is given in the last column. The best results for each column are in bold. Numbers in the "Experimental results" part of the table were collected after running the algorithms with the default sets of parameters on pre-processed data. However, ADANET, GENIE3, CLR, C3NET, MRNET, and ARACNE methods, as they are originally defined, take a multifactorial matrix as an input, which is unavailable in this challenge. Therefore they were excluded from the comparison. Numbers in the "Winner of the challenge" part of the table correspond to the best methods participating in the challenge.

Table 3 Results of the different inference methods on DREAM4 networks, challenge size 100

Method	Network (AUPR/AUROC respectively)										Overall
	1		2		3		4		5		
					Experimental results						
ENNET	**0.604**	0.893	**0.456**	**0.856**	**0.421**	**0.865**	**0.506**	**0.878**	**0.264**	**0.828**	**87.738**
					Winner of the challenge						
Pinna et al.	0.536	**0.914**	0.377	0.801	0.390	0.833	0.349	0.842	0.213	0.759	71.589
2nd	0.512	0.908	0.396	0.797	0.380	0.829	0.372	0.844	0.178	0.763	71.297
3rd	0.490	0.870	0.327	0.773	0.326	0.844	0.400	0.827	0.159	0.758	64.715

Results of the different inference methods on DREAM4 networks, challenge size 100. An area under the ROC curve (AUROC) and an area under the Precision-Recall curve (AUPR) are given for each network respectively. The Overall Score for all the networks is given in the last column. The best results for each column are in bold. Numbers in the "Experimental results" part of the table were collected after running the algorithms with the default sets of parameters on pre-processed data. However, ADANET, GENIE3, CLR, C3NET, MRNET, and ARACNE methods, as they are originally defined, take a multifactorial matrix as an input, which is unavailable in this challenge. Therefore they were excluded from the comparison. Numbers in the "Winner of the challenge" part of the table correspond to the best methods participating in the challenge.

ENNET shows better *in vivo* results than the other methods in terms of an area under the the ROC curve. Still, predictions for *in vivo* expression profiles show a low overall accuracy. One of the reasons for a poor performance of the inference methods for such expression profiles is a fact that experimentally confirmed pathways, and consequently gold standards derived from them, cannot be assumed complete, regardless of how well is a model organism known. Additionally, there are regulators of gene expression other than transcription factors, such as miRNA, and siRNA. As shown in this study, *in silico* expression profiles provide enough information to confidently reverse-engineer their underlying structure, whereas *in vivo* data hide a much more complex system of regulatory interactions.

Computational complexity of ENNET

Computational complexity of ENNET depends mainly on the computational complexity of the regression stump base learner, which is used in the main loop of the algorithm. As shown in Figure 1, we call the regression stump algorithm T times for each k-th target gene, $k \in \{1, ..., P\}$. Given a sorted input, a regression stump is $O(PN)$ complex. We sort the expression matrix in an $O(PN \log N)$ time. All the other instructions in the main loop of ENNET are at most $O(N)$. The computational complexity of the whole method is thus $O(PN \log N + TP^2N + TPN)$. Because, in practice, the dominating part of the sum is TP^2N, we report a final computational complexity of ENNET as $O(TP^2N)$, and compare it to the other inference methods in Table 6. Note that the measure for

Table 4 Results of the different inference methods on DREAM4 networks, challenge size 100 multifactorial

Method	Network (AUPR/AUROC respectively)										Overall
	1		2		3		4		5		
					Experimental results						
ENNET	**0.184**	0.731	**0.261**	**0.807**	**0.289**	**0.813**	**0.291**	**0.822**	**0.286**	**0.829**	**52.839**
ADANET	0.149	0.664	0.094	0.605	0.191	0.703	0.172	0.712	0.182	0.694	24.970
GENIE3	0.158	**0.747**	0.154	0.726	0.232	0.777	0.210	0.795	0.204	0.792	37.669
C3NET	0.077	0.562	0.095	0.588	0.126	0.621	0.113	0.687	0.110	0.607	15.015
CLR	0.142	0.695	0.118	0.700	0.178	0.746	0.174	0.748	0.174	0.722	28.806
MRNET	0.138	0.679	0.128	0.698	0.204	0.755	0.178	0.748	0.187	0.725	30.259
ARACNE	0.123	0.606	0.102	0.603	0.192	0.686	0.159	0.713	0.166	0.659	22.744
					Winner of the challenge						
GENIE3	0.154	0.745	0.155	0.733	0.231	0.775	0.208	0.791	0.197	0.798	37.428
2nd	0.108	0.739	0.147	0.694	0.185	0.748	0.161	0.736	0.111	0.745	28.165
3rd	0.140	0.658	0.098	0.626	0.215	0.717	0.201	0.693	0.194	0.719	27.053

Results of the different inference methods on DREAM4 networks, challenge size 100 multifactorial. An area under the ROC curve (AUROC) and an area under the Precision-Recall curve (AUPR) are given for each network respectively. The Overall Score for all the networks is given in the last column. The best results for each column are in bold. Numbers in the "Experimental results" part of the table were collected after running the algorithms with the default sets of parameters on pre-processed data. Numbers in the "Winner of competition" part of the table correspond to the best methods participating in the challenge.

Table 5 Results of the different inference methods on DREAM5 networks

Method	Network (AUPR/AUROC respectively)						Overall
	1		3		4		
			Experimental results				
ENNET	**0.432**	**0.867**	0.069	**0.642**	0.021	**0.532**	>**300**
ADANET	0.261	0.725	0.083	0.596	0.021	0.517	16.006
GENIE3	0.291	0.814	**0.094**	0.619	0.021	0.517	40.335
C3NET	0.080	0.529	0.026	0.506	0.018	0.501	0.000
CLR	0.217	0.666	0.050	0.538	0.019	0.505	4.928
MRNET	0.194	0.668	0.041	0.525	0.018	0.501	2.534
ARACNE	0.099	0.545	0.029	0.512	0.017	0.500	0.000
			Winner of the challenge				
GENIE3	0.291	0.815	0.093	0.617	0.021	0.518	40.279
ANOVA η^2	0.245	0.780	0.119	0.671	**0.022**	0.519	34.023
TIGRESS	0.301	0.782	0.069	0.595	0.020	0.517	31.099

Results of the different inference methods on DREAM5 networks. An area under the ROC curve (AUROC) and an area under the Precision-Recall curve (AUPR) are given for each network respectively. The Overall Score for all the networks is given in the last column. The best results for each column are in bold. Numbers in the "Experimental results" part of the table were collected after running the algorithms with the default sets of parameters on pre-processed data. Numbers in the "Winner of the challenge" part of the table correspond to the best methods participating in the challenge.

the information-theoretic methods: CLR, MRNET, and ARACNE does not include a calculation of the mutual information matrix.

When implementing ENNET algorithm we took advantage of the fact that gene selection problems are independent of each other. Our implementation of the algorithm is able to calculate them in parallel if multiple processing units are available. User can choose from variety of parallel backends including multicore package for a single computer and parallelization based on Message Passing Interface for a cluster of computers. The biggest data we provided as input in our tests were *in vivo* expression profiles of *S. cerevisiae* from the DREAM 5 challenge. These are genome-wide expression profiles of 5950 genes (333 of them are known transcription factors) measured in 536 experiments. It took 113 minutes and 30 seconds to calculate the network on a standard desktop workstation with one Intel®Core™i7-870 processor with 4 cores and two threads per core (in total 8 logical processors) and 16 GB RAM. However, it took only 16 minutes and 40 seconds to calculate the same network on a machine with four AMD Opteron™6282 SE processors, each with 8 cores and two threads per core (in total 64 logical processors) and 256 GB RAM. All the data sets from the DREAM 3 and the DREAM 4 challenges were considerably smaller, up to 100 genes. It took less than one minute to calculate each of these networks on a desktop machine.

Table 6 The computational complexity of ENNET and the other GRN inference methods

Method	Complexity
ENNET	$O(TP^2N), T = 5000$
ADANET	$O(CTP^2N), C = 30, T = \lceil\sqrt{P}\rceil$
GENIE3	$O(TKPN \log N), T = 1000, K = \lceil\sqrt{P}\rceil$
C3NET	$O(P^2)$
CLR	$O(P^2)$
MRNET	$O(fP^2), f \in [1, P]$
ARACNE	$O(P^3)$

The computational complexity of ENNET and the other GRN inference methods with respect to the number of genes *P* and the number of samples *N*. The computational complexity of CLR, MRNET, and ARACNE is given without calculating the Mutual Information matrix.

Setting parameters of ENNET

The ENNET algorithm is controlled by four parameters: the two sampling rates s_s and s_f, the number of iterations T and the learning rate ν. The sampling rate of samples s_s and the sampling rate of transcription factors s_f govern the level of randomness when selecting, respectively, rows and columns of the expression matrix to fit a regression model. The default choice of the value of s_s is 1, i.e., we select with replacement a bootstrap sample of observations of the same size as an original training set at each iteration. Because some observations are selected more than once, around 0.37 of random training samples are out of bag in each iteration. It is more difficult to choose an optimal value of s_f, which governs how many transcription factors are used to fit each base learner. Setting this parameter to a low value forces ENNET to score transcription factors, even if their improvement criterion, as shown in Equation 2, would not have promoted them in a pure greedy search, i.e., $s_f = 1$. However, if a chance of selecting a true transcription factor as a feature is too low, ENNET will suffer from selecting random genes as true regulators.

Even though reverse-engineering of GRNs does not explicitly target a problem of predicting gene expression, we choose the values of sampling rates such that the squared-error loss of a prediction of the target gene expression as given by f_T (see Figure 1) is minimal. This is done without looking at the ground truth of regulatory connections. For each benchmark challenge we performed a grid search over $(s_s, s_f) \in \{0.1, 0.3, 0.5, 0.7, 1\} \times \{0.1, 0.3, 0.5, 0.7, 1\}$ with fixed $\nu = 0.001$, $T = 5000$. For each specific set of parameters we analyzed an average 5-fold cross-validated loss over all the observations (across all gene selection problems). We further analyze our approach with respect to one of the challenges: DREAM4 size 100, as shown in Figure 3. The minimal average loss was achieved for $s_s = 1$ and $s_f = 0.3$ (see Figure 3 A), which is consistent with the default parameters proposed for Random Forest algorithm [28]. We also compared the measure based on an average loss with the Overall Score as defined by Equation 6. The results were consistent across the two measures, i.e., a selection of parameters that gave a low average loss also led to the accurate network predictions (see Figure 3 B). An advantage of the average loss measure is a fact that the gold standard network is not used to tune parameters.

In Figure 4 we present a detailed analysis of the accuracy of the GRN inference across different networks of the DREAM4 size 100 challenge. Each point on both Figure 4 A and Figure 4 B is a result of running ENNET with different parameters: $(s_s, s_f) \in \{0.1, 0.3, 0.5, 0.7, 1\} \times \{0.1, 0.3, 0.5, 0.7, 1\}$ with fixed $\nu = 0.001$, $T = 5000$. The highlighted points are corresponding to $s_s = 1$, $s_f = 0.3$, $\nu = 0.001$, $T = 5000$. An area under the Precision-Recall curve and an area under the ROC curve are two different measures of the accuracy of an inferred network, which are well preserved across the five networks: for each separate network we observe that AUPR and AUROC decreases in a function of an average loss. As the Overall Score is closely related to AUPR and AUROC, the results shown in Figure 4 explain the shape of a surface shown in Figure 3.

As ENNET uses boosting, it needs a careful tuning of the number of iterations T and the learning rate ν. It has been shown [38] that parameters T and ν are closely coupled. Usually the best prediction results are achieved when ν is fixed to a small positive number, e.g. $\nu \leq 0.001$, and the optimal value of TY is found in a process of cross-validation. As described above, we reason that the choice of parameters, which gives a low average loss on a cross-validated test set, leads to an accurate network prediction. Therefore in Figure 5 we present how an average loss depends on $T \in \{1, ..., 5000\}$ for different values of $\nu \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, with fixed $s_s = 1$, $s_f = 0.3$. Each of the line shows how much ENNET overtrains the data for a given T and ν. Finally, the optimal choice of parameters for DREAM4 size 100 challenge is $s_s = 1$, $s_f = 0.3$, $T = 5000$, $\nu = 0.001$. Following the same practice, we used this default set of parameters: $s_s = 1$, $s_f = 0.3$, $T = 5000$, $\nu = 0.001$ to evaluate ENNET

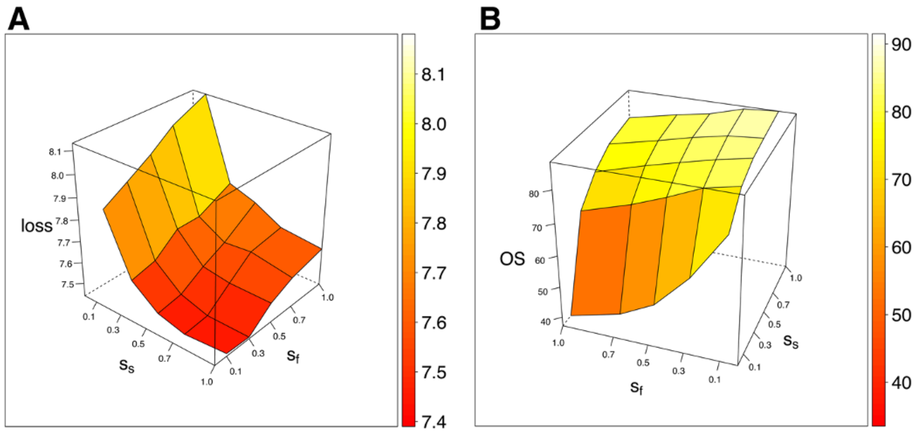


Figure 3 The analysis of the sampling rates s_s and s_f for DREAM 4 size 100 challenge: the Overall Score and a loss. The analysis of the sampling rates s_s and s_f for the DREAM 4 size 100 challenge. **A:** For each set of parameters $(s_s, s_f, M, \nu) \in \{0.1, 0.3, 0.5, 0.7, 1\} \times \{0.1, 0.3, 0.5, 0.7, 1\} \times \{5000\} \times \{0.001\}$ we analyzed an average 5-fold cross-validated loss over all the observations (across all gene selection problems) from all 5 networks. The minimal average loss was achieved for high values of $s_s = 1$ and low values of $s_f = 0.3$. **B:** We also compared the measure based on an average loss with the original Overall Score, as proposed by the authors of the DREAM challenge. The results were consistent across the two measures, i.e., the parameters that gave low average loss also led to accurate network predictions (a high Overall Score).

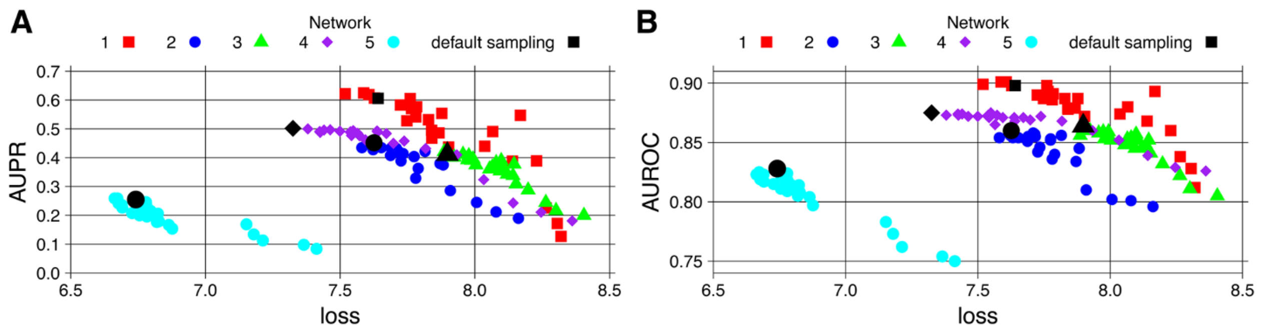


Figure 4 The analysis of the sampling rates s_s and s_f for DREAM 4 size 100 challenge: AUPR, AUROC, and a loss. The analysis of the sampling rates s_s and s_f for DREAM 4 size 100 challenge. **A:** For each set of parameters $(s_s, s_f, M, \nu) \in \{0.1, 0.3, 0.5, 0.7, 1\} \times \{0.1, 0.3, 0.5, 0.7, 1\} \times \{5000\} \times \{0.001\}$ we analyzed an area under the Precision-Recall curve (AUPR) in function of an average 5-fold cross-validated loss over all the observations (across all gene selection problems) from all 5 networks. For each network AUPR is decreasing in a function of a loss. For each network a point corresponding to the default set of parameters is highlighted, i.e., $(s_s, s_f, M, \nu) = (1, 0.3, 5000, 0.001)$. Usually, the default set of parameters gives the minimal loss (maximal AUPR). **B:** By analogy, different choices of parameters lead to a different area under the ROC curve (AUROC). The two measures are consistent with each other.

algorithm on all the benchmark networks using ground truth, i.e., for calculating the Overall Score and comparing it to the other algorithms.

Stability of ENNET

Because ENNET uses random sampling of samples and features at each iteration of the main loop, as shown in Figure 1, it may calculate two different networks for two different executions on the same expression data. With the default choice of parameters, i.e., $s_s = 1$, $s_f = 0.3$, $T = 5000$, $\nu = 0.001$, we expect numerous random resamplings, and therefore we need to know if a GRN calculated by ENNET is stable between different executions. We applied ENNET to the 5 networks that form DREAM 4 size 100 benchmark, repeating the inference calculations independently ten times for each network. Then, for each network, we calculated a Spearman's rank correlation between all pairs among the ten independent runs. The lowest correlation coefficient we obtained was $\rho > 0.975$, with p-value $< 2.2e-16$, indicating that the networks that result from independent runs are very similar. This proves that ENNET, despite being a randomized algorithm, finds a stable solution to the inference problem.

Conclusions

We have proposed the ENNET algorithm for reverse-engineering of Gene Regulatory Networks. ENNET uses a variety of types of expression data as an input, and shows robust performance across different benchmark networks. Moreover, it does not assume any specific model of a regulatory interaction and do not require

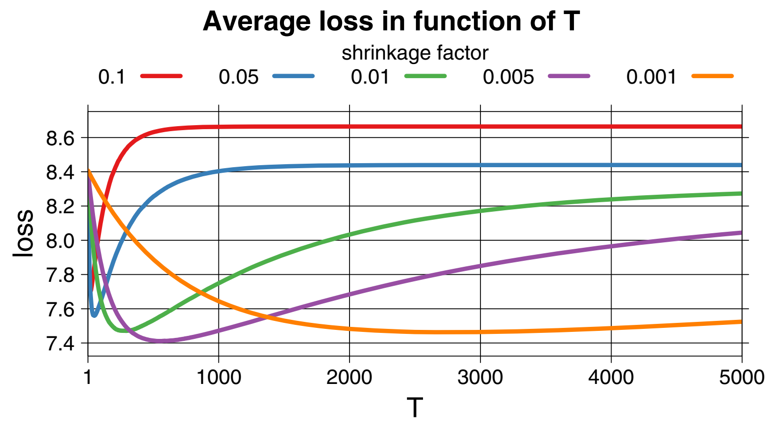


Figure 5 The analysis of the number of iterations T and the shrinkage ν for the DREAM 4 size 100 challenge. The analysis of the number of iterations T and the shrinkage factor ν for DREAM 4 size 100 challenge. These two parameters are closely coupled: the lower is the shrinkage parameter ν, the more iterations T are needed to train the model such that it achieves the minimal loss.

fine-tuning of its parameters, i.e., we define the default set of parameters, which promises accurate predictions for the future networks. Nevertheless, together with the algorithm, we propose a procedure of tuning parameters of ENNET towards minimizing empirical loss. Processing genome-scale expression profiles is feasible with ENNET: including up to a few hundred transcription factors, and up to a few thousand regulated genes. As shown in this study, the proposed method compares favorably to the state-of-the-art algorithms on the universally recognized benchmark data sets.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
JS and TA conceived the method and drafted the manuscript. JS implemented the method and ran the experiments. JS and TA read and approved the final manuscript.

Received: 24 June 2013 Accepted: 17 October 2013
Published: 22 October 2013

References

1. Someren E, Wessels L, Backer E, Reinders M: **Genetic network modeling.** *Pharmacogenomics* 2002, **3**(4):507–525.
2. Eisen M, Spellman P, Brown P, Botstein D: **Cluster analysis and display of genome-wide expression patterns.** *Proc Natl Acad Sci* 1998, **95**(25):14863.
3. Gardner T, Faith J: **Reverse-engineering transcription control networks.** *Phys Life Rev* 2005, **2:**65–88.
4. Chen T, He H, Church G, et al.: **Modeling gene expression with differential equations.** In *Pacific Symposium on Biocomputing, Volume 4*. Singapore: World Scientific Press; 1999:4.
5. D'haeseleer P, Wen X, Fuhrman S, Somogyi R, et al.: **Linear modeling of mRNA expression levels during CNS development and injury,** In *Pacific Symposium on Biocomputing, Volume 4*. Singapore: World Scientific Press; 1999:41–52.
6. Gardner TS, di Bernardo D, Lorenz D, Collins JJ: **Inferring genetic networks and identifying compound mode of action via expression profiling.** *Science* 2003, **301**(5629):102–105.
7. Yip K, Alexander R, Yan K, Gerstein M: **Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data.** *PLoS One* 2010, **5:**e8121.
8. Greenfield A, Hafemeister C, Bonneau R: **Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks.** *Bioinformatics* 2013, **29**(8):1060–1067.
9. Friedman N, Linial M, Nachman I, Pe'er D: **Using Bayesian networks to analyze expression data.** *J Comput Biol* 2000, **7**(3–4):601–620.
10. Perrin BE, Ralaivola L, Mazurie A, Bottani S, Mallet J, d'Alche Buc F: **Gene networks inference using dynamic Bayesian networks.** *Bioinformatics* 2003, **19**(suppl 2):ii138–ii148.
11. Yu J, Smith V, Wang P, Hartemink A, Jarvis E: **Advances to Bayesian, network inference for generating causal networks from observational biological data.** *Bioinformatics* 2004, **20**(18):3594–3603.
12. Segal E, Wang H, Koller D: **Discovering molecular pathways from protein interaction and gene expression data.** *Bioinformatics* 2003, **19**(suppl 1):i264–i272.
13. Neapolitan R: *Learning Bayesian Networks*. Upper Saddle River: Pearson Prentice Hall; 2004.
14. Qi J, Michoel T: **Context-specific transcriptional regulatory network inference from global gene expression maps using double two-way t-tests.** *Bioinformatics* 2012, **28**(18):2325–2332.
15. Prill R, Marbach D, Saez-Rodriguez J, Sorger P, Alexopoulos L, Xue X, Clarke N, Altan-Bonnet G, Stolovitzky G: **Towards a rigorous assessment of systems biology models: the DREAM3 challenges.** *PLoS One* 2010, **5**(2):e9202.
16. Bansal M, Belcastro V, Ambesi-Impiombato A, di Bernardo D: **How to infer gene networks from expression profiles.** *Mol Syst Biol* 2007, **3:**78.
17. Butte A, Kohane I: **Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements.** In *Pacific Symposium on Biocomputing, Volume 5*. Singapore: World Scientific Press; 2000:418–429.
18. Lee W, Tzou W: **Computational methods for discovering gene networks from expression data.** *Brief Bioinform* 2009, **10**(4):408–423.
19. Markowetz F, Spang R: **Inferring cellular networks–a review.** *BMC Bioinformatics* 2007, **8**(Suppl 6):S5.
20. Altay G, Emmert-Streib F: **Inferring the conservative causal core of gene regulatory networks.** *BMC Syst Biol* 2010, **4:**132.
21. Küffner R, Petri T, Tavakkolkhah P, Windhager L, Zimmer R: **Inferring gene regulatory networks by ANOVA.** *Bioinformatics* 2012, **28**(10):1376–1382.
22. Margolin A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera R, Califano A: **ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.** *BMC Bioinformatics* 2006, **7**(Suppl 1):S7.
23. Margolin A, Wang K, Lim W, Kustagi M, Nemenman I, Califano A: **Reverse engineering cellular networks.** *Nat Protoc* 2006, **1**(2):662–671.
24. Faith J, Hayete B, Thaden J, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins J, Gardner T: **Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles.** *PLoS Biol* 2007, **5:**e8.
25. Zhang X, Liu K, Liu ZP, Duval B, Richer JM, Zhao XM, Hao JK, Chen L: **NARROMI: a noise and redundancy reduction technique improves accuracy of gene regulatory network inference.** *Bioinformatics* 2013, **29:**106–113.
26. Meyer P, Kontos K, Lafitte F, Bontempi G: **Information-theoretic inference of large transcriptional regulatory networks.** *EURASIP J Bioinform Syst Biol* 2007, **2007:**8.
27. Ding C, Peng H: **Minimum redundancy feature selection from microarray gene expression data.** In *Computational Systems Bioinformatics Conference CSB2003*. Washington: IEEE; 2003:523–528.
28. Irrthum A, Wehenkel L, Geurts P, et al.: **Inferring regulatory networks from expression data using tree-based methods.** *PLoS One* 2010, **5**(9):e12776.
29. Haury AC, Mordelet F, Vera-Licona P, Vert JP: **TIGRESS: trustful inference of gene regulation using stability selection.** *BMC Syst Biol* 2012, **6:**145.
30. Freund Y, Schapire RE: **Experiments with a new boosting algorithm.** *International Conference on Machine Learning* 1996:148–156.
31. Freund Y, Schapire RE: **A decision-theoretic generalization of on-line learning and an application to boosting.** *J Comput Syst Sci* 1997, **55:**119–139.
32. Sławek J, Arodź T: **ADANET: inferring gene regulatory networks using ensemble classifiers.** In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*. New York: ACM; 2012:434–441.
33. Lim N, Şenbabaoğlu Y, Michailidis G, d'Alché Buc F: **OKVAR-Boost: a novel boosting algorithm to infer nonlinear dynamics and interactions in gene regulatory networks.** *Bioinformatics* 2013, **29**(11):1416–1423.
34. Bolstad B, Irizarry R, Åstrand M Speed, T: **A comparison of normalization methods for high density oligonucleotide array data based on variance and bias.** *Bioinformatics* 2003, **19**(2):185–193.
35. Marbach D, Costello J, Küffner R, Vega N, Prill R, Camacho D, Allison K, Kellis M, Collins J, Stolovitzky G, et al: **Wisdom of crowds for robust gene network inference.** *Nat Methods* 2012, **9**(8):797.
36. Theodoridis S, Koutroumbas K: *Pattern Recognition*. London: Elsevier/Academic Press; 2006.
37. Tuv E, Borisov A, Runger G, Torkkola K: **Feature selection with ensembles, artificial variables, and redundancy elimination.** *J Mach Learn Res* 2009, **10:**1341–1366.
38. Friedman JH: **Greedy function approximation: a gradient boosting machine.** *Ann Stat* 2001, **29**(5):1189–1232.
39. Schaffter T, Marbach D, Floreano D: **GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods.** *Bioinformatics* 2011, **27**(16):2263–2270.
40. Gama-Castro S, Salgado H, Peralta-Gil M, Santos-Zavaleta A, Muñiz-Rascado L, Solano-Lira H, Jimenez-Jacinto V, Weiss V, García-Sotelo

J, López-Fuentes A, et al.: **RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (gensor units).** *Nucleic Acids Res* 2011, **39**(suppl 1):D98–D105.
41. Kim S, Imoto S, Miyano S: **Inferring gene networks from time series microarray data using dynamic Bayesian networks.** *Brief Bioinform* 2003, **4**(3):228–235.
42. Di Camillo B, Toffolo G, Cobelli C: **A gene network simulator to assess reverse engineering algorithms.** *Ann N Y Acad Sci* 2009, **1158:**125–142.
43. Kremling A Fischer, S, Gadkar K, Doyle F, Sauter T, Bullinger E, Allgöwer F, Gilles E: **A benchmark for fethods in reverse engineering and model discrimination: problem formulation and solutions.** *Genome Res* 2004, **14**(9):1773–1785.
44. Mendes P, Sha W, Ye K: **Artificial gene networks for objective comparison of analysis algorithms.** *Bioinformatics* 2003, **19**(suppl 2):ii122–ii129.
45. Van den Bulcke T, Van Leemput K, Naudts B, Van Remortel P, Ma H, Verschoren A, De Moor B, Marchal K: **SynTReN a generator of synthetic gene expression data for design and analysis of structure learning algorithms.** *BMC Bioinformatics* 2006, **7:**43.
46. Ravasz E, Somera A, Mongru D, Oltvai Z, Barabási A: **Hierarchical organization of modularity in metabolic networks.** *Science* 2002, **297**(5586):1551–1555.
47. Shen-Orr S, Milo R, Mangan S, Alon U: **Network motifs in the transcriptional regulation network of Escherichia coli.** *Nat Genet* 2002, **31:**64–68.
48. Hache H, Wierling C, Lehrach H, Herwig R: **GeNGe: systematic generation of gene regulatory networks.** *Bioinformatics* 2009, **25**(9):1205–1207.
49. Roy S, Werner-Washburne M, Lane T: **A system for generating transcription regulatory networks with combinatorial control of transcription.** *Bioinformatics* 2008, **24**(10):1318–1320.
50. Haynes B, Brent M: **Benchmarking regulatory network reconstruction with GRENDEL.** *Bioinformatics* 2009, **25**(6):801–807.
51. Stolovitzky G, Kundaje A, Held G, Duggar K, Haudenschild C, Zhou D, Vasicek T, Smith K, Aderem A, Roach J: **Statistical analysis of MPSS, measurements: application to the study of LPS-activated macrophage gene expression.** *Proc Natl Acad Sci USA* 2005, **102**(5):1402–1407.
52. Meyer PE, Lafitte F, Bontempi G: **Minet: AR/Bioconductor package for inferring large transcriptional networks using mutual information.** *BMC Bioinformatics* 2008, **9:**461.
53. Marbach D, Prill R, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G: **Revealing strengths and weaknesses of methods for gene network inference.** *Proc Natl Acad Sci* 2010, **107**(14):6286–6291.
54. Marbach D, Schaffter T, Mattiussi C, Floreano D: **Generating realistic in silico gene networks for performance assessment of reverse engineering methods.** *J Comput Biol* 2009, **16**(2):229–239.
55. Ashburner M, Ball C, Blake J, Botstein D, Butler H, Cherry J, Davis A, Dolinski K, Dwight S, Eppig J, et al.: **Gene Ontology: tool for the unification of biology.** *Nat Genet* 2000, **25:**25.

doi:10.1186/1752-0509-7-106
Cite this article as: Sławek and Arodź: ENNET: inferring large gene regulatory networks from expression data using gradient boosting. *BMC Systems Biology* 2013 **7**:106.