BIOINFORMATICS

Vol. 24 ECCB 2008, pages i76–i82
doi:10.1093/bioinformatics/btn273

# SIRENE: supervised inference of regulatory networks

Fantine Mordelet[1,2,3,4,*] and Jean-Philippe Vert [1,2,3]

[1]Ecole des Mines de Paris, ParisTech, 35 rue Saint-Honoré, Fontainebleau F-77300, [2]Institut Curie, Paris F-75248, [3]INSERM, U900, Paris F-75248 and [4]CREST, INSEE, 3 av. Pierre Larousse, Malakoff, F-92240 France

**ABSTRACT**

**Motivation:** Living cells are the product of gene expression programs that involve the regulated transcription of thousands of genes. The elucidation of transcriptional regulatory networks is thus needed to understand the cell's working mechanism, and can for example, be useful for the discovery of novel therapeutic targets. Although several methods have been proposed to infer gene regulatory networks from gene expression data, a recent comparison on a large-scale benchmark experiment revealed that most current methods only predict a limited number of known regulations at a reasonable precision level.

**Results:** We propose SIRENE (Supervised Inference of Regulatory Networks), a new method for the inference of gene regulatory networks from a compendium of expression data. The method decomposes the problem of gene regulatory network inference into a large number of local binary classification problems, that focus on separating target genes from non-targets for each transcription factor. SIRENE is thus conceptually simple and computationally efficient. We test it on a benchmark experiment aimed at predicting regulations in *Escherichia coli*, and show that it retrieves of the order of 6 times more known regulations than other state-of-the-art inference methods.

**Availability:** All data and programs are freely available at http://cbio.ensmp.fr/sirene.

**Contact:** Fantine.Mordelet@ensmp.fr

## 1 INTRODUCTION

Elucidating the structure of gene regulatory networks is crucial to understand how transcription factors (TFs) regulate gene expression and allow an organism to regulate its metabolism and adapt itself to environmental changes. While high-throughput sequencing and other post-genomics technologies offer a wealth of information about individual genes, the experimental characterization of transcriptional *cis*-regulation at a genome scale remains a daunting challenge, even for well-studied model organisms. *In silico* methods that attempt to reconstruct such global gene regulatory networks from prior biological knowledge and available genomic and post-genomic data therefore constitute an interesting direction towards the elucidation of these networks.

Transcriptional *cis*-regulation directly influences the level of mRNA transcripts of regulated genes. Not surprisingly, many *in silico* methods have been proposed to reconstruct gene regulatory networks from gene expression data, produced at a fast rate by microarrays (Bansal *et al.*, 2007). Clustering gene expression profiles across different conditions identifies groups of genes with similar transcriptomic response, suggesting co-regulation within each group (Tavazoie *et al.*, 1999). Clustering methods are widely used, computationally efficient, but do not easily lead to the identification of regulators for a given set of genes. Some authors nonetheless have observed that identifying similarities, or more generally mutual information between the expression profiles of a TF and of a target gene is a good indicator of regulation (Butte *et al.*, 2000; Faith *et al.*, 2007). When time series of gene expression data are available, other reverse-engineering methodologies can be applied to capture the interactions governing the observed dynamics. Different mathematical formalisms have been proposed to model such dynamics, including Boolean networks (Akutsu *et al.*, 2000) or ordinary or stochastic partial differential equations (Bansal *et al.*, 2006; Chen *et al.*, 1999, 2005; di Bernardo *et al.*, 2005; Gardner *et al.*, 2003; Tegner *et al.*, 2003). Some authors have also attempted to detect causality relationships between gene expression data, be they time series or compendia of various experiments, using statistical methods such as Bayesian networks (Friedman *et al.*, 2000). These methods that estimate the regulatory network by fitting a dynamic or statistical model are often computationally and data demanding.

The comparison of these different approaches and of their capacity to accurately reconstruct large-scale regulatory networks has been hampered by the difficulty to assemble a realistic set of biologically validated regulatory relationships and use it as a benchmark to assess the performance of each method. Recently, Faith *et al.* (2007) compiled such a benchmark, by gathering all known transcriptional *cis*-regulation in *Escherichia coli* and collecting a compendium of several hundreds of gene expression profiling experiments. They compared several approaches, including Bayesian networks (Friedman *et al.*, 2000), ARACNe (Margolin *et al.*, 2006) and the context likelihood of relatedness (CLR) algorithm, a new method that extends the relevance networks class of algorithms (Butte *et al.*, 2000). They observed that CLR outperformed all other methods in prediction accuracy, and experimentally validated some predictions. CLR can therefore be considered as state-of-the-art among methods that use compendia of gene expression data for large-scale inference of regulatory networks.

In this article, we present SIRENE (Supervised Inference of Regulatory Networks), a new method to infer gene regulatory networks on a genome scale from a compendium of gene expression data. SIRENE differs fundamentally from other approaches in that it requires as inputs not only gene expression data, but also a list of known regulation relationships between TF and target genes. In machine-learning terminology, the method is *supervised* in the sense that it uses a partial knowledge of the information we want to predict in order to guide the inference engine for the prediction of new information. The necessity to input some known regulations is not a serious restriction in many applications, as many regulations have already been characterized in model organisms,

*To whom correspondence should be addressed.

and can be inferred by homology in newly sequenced genomes. Known regulations allow us to use a natural induction principle to predict new regulations: if a gene $A$ has an expression profile similar to a gene $B$ known to be regulated by a given TF, then gene $A$ is likely to be also regulated by the TF. The fact that genes with similar expression profiles are likely to be co-regulated has been used for a long time in the construction of groups of genes by unsupervised clustering of expression profiles. The novelty in our approach is to use this principle in a supervised classification paradigm. This inference paradigm has the advantage that no particular hypothesis is made regarding the relationship between the expression data of a TF and those of regulated genes. In fact, expression data for the TF are not even needed in our approach.

Many algorithms for supervised classification can be used to transform this inference principle into a working algorithm. We use in our experiments the support vector machine (SVM) algorithm, a state-of-the-art method for supervised classification. The idea to cast the problem of gene or protein networks inference as a supervised classification problem, using known interactions as inputs, has been recently proposed and investigated for the reconstruction of protein–protein interaction (PPI) and metabolic networks (Ben-Hur and Noble, 2005; Yamanishi *et al.*, 2004). Bleakley *et al.* (2007) proposed a simple method where a local model is estimated to predict the interacting partners of each protein in the network, and all local models are then combined together to predict edges throughout the network. They showed that this method gave important improvement in accuracy compared with more elaborated methods on both the PPI and metabolic networks. Here we adapt this strategy for the reconstruction of gene regulatory networks. For each TF, we estimate a local model to discriminate, based on their expression profiles, the genes regulated by the TF from others genes. All local models are then combined to rank candidate regulatory relationships between TFs and all genes in the genome. SIRENE is conceptually simple, easy to implement and computationally scalable to whole genomes because each local model only involves the training of a supervised classification algorithm on a few hundreds or thousands examples.

We test SIRENE on the benchmark experiment proposed by Faith *et al.* (2007), which aims at reconstructing known regulations within *Escherichia coli* genes from a compendium of gene expression data. On this benchmark, SIRENE strongly outperforms the best results reported by Faith *et al.* (2007), with the CLR algorithm. For example, at a 60% true positive rate (precision), CLR identifies 7.5% of all known regulatory relationships (recall), while SIRENE has a recall of 44.5% at the same precision level using expression profiles.

## 2 SYSTEM AND METHODS

### 2.1 SIRENE

SIRENE is a general method to infer new regulation relationships between known TF and all genes of an organism. It requires two types of data as inputs. First, each gene in the organism needs to be characterized by some data, in our case a vector of expression values in a compendium of expression profiles. Second, a list of known regulation relationships between known TF and some genes is needed. More precisely, for each TF, we need a list of genes known to be regulated by the TF, and if possible a list of genes known not to be regulated by it. Such lists can typically be constructed from publicly available databases of experimentally characterized regulations, e.g. RegulonDB for *E.coli* genes (Salgado *et al.*, 2006). While such databases usually do not contain information about the absence of regulations, we discuss in Section 2.3 below how we generate negative examples.

When such data are available, SIRENE splits the problem of regulatory network inference into many binary classification subproblems, one subproblem being associated with each TF. More precisely, for each TF, SIRENE trains a binary classifier to discriminate between genes known to be regulated and genes known not to be regulated by the TF, based on the data that characterize the genes (e.g. expression data). The rationale behind this approach is that, although we make no hypothesis regarding the relationship between the measured expression level of a TF and its targets, we assume that if two genes are regulated by the same TF then they are likely to exhibit similar expression patterns. In our implementation, we use an SVM to solve the binary classification problems (Section 2.2), but any other algorithm for supervised binary classification could in principle be used. Once trained, the model associated with a given TF is able to assign to each new gene, not used during training, a score that tends to be positive and large when it believes, based on the data that characterize the gene, that the gene is regulated by the TF. The final step is to combine all scores of the different models to rank the candidate TF–gene interactions in a unique list by decreasing score.

In summary, SIRENE decomposes the difficult problem of gene regulatory network inference into a large number of subproblems that attempt to estimate local models to characterize the genes regulated by each TF. A similar approach was proposed by Bleakley *et al.* (2007) to infer undirected graphs, and successfully tested on the reconstruction of metabolic and PPI networks. Here we are confronted with a slightly different problem, since the graph we wish to infer is directed and we just need to infer local models to predict genes regulated by any given TF.

### 2.2 SVM

In our implementation of SIRENE, we use an SVM to train predictors for each local model associated with a TF. SVM is a popular algorithm to solve general supervised binary classification problems, which is considered state-of-the-art in many applications and is available in many free and public implementations (Schölkopf *et al.*, 2004; Vapnik, 1998). The basic ingredient of an SVM is a kernel function $K(x,y)$ between any two genes $x$ and $y$, that can often be thought of as a measure of similarity between the genes. In our case, the similarity between genes is measured in terms of expression profiles. Given a set of $n$ genes $x_1,\ldots,x_n$ that belong to two classes, denoted arbitrarily $-1$ and $+1$, an SVM estimates a scoring function for any new gene $x$ of the form:

$$f(x)=\sum_{i=1}^{n}\alpha_i K(x_i,x).$$

The weights $\alpha_i$ in this expression are optimized by the SVM to enforce as much as possible large positive scores for genes in the class $+1$ and large negative scores for genes in the class $-1$ in the training set. A parameter, often called $C$, allows to control the possible overfitting to the training set. The scoring function $f(x)$ can then be used to rank genes with unknown class by decreasing score, from the most likely to belong to class $+1$ to the most likely to belong to class $-1$.

The kernel $K(x,y)$ defines the similarity measure used by the SVM to build the scoring function. In our experiments, we want to infer regulations from gene expression data. Each collection of gene expression data is a vector, so we simply use the common Gaussian radial basis function kernel between vectors $u$ and $v$:

$$K(u,v)=\exp\left(-\frac{||u-v||^2}{2\sigma^2}\right),$$

where $\sigma>0$ is the bandwidth parameter of the kernel.

Each SVM has therefore two parameters, $C$ and $\sigma$. In order to limit the risk of overfitting and positive bias in our performance evaluation that could result from an over-optimization of these parameters on the benchmark data, we simply fix them for all SVM to the unique values $C=+\infty$ and $\sigma=8$. The value $C=+\infty$ means that we train hard-margin SVM, which is always possible with a Gaussian kernel (Vapnik, 1998). The choice $\sigma=8$ was based on the observation that we use expression profiles for 445 microarrays scaled to zero mean and unit standard deviation, i.e. each gene is represented by a vector of dimension 445 and of length $\sqrt{445}\sim 21$. Hence the distance between two orthogonal profiles is of the order of $\sqrt{2}\times\sqrt{445}\sim 32$. We expect that a bandwidth of the order of $\sigma=8$, which puts two orthogonal profiles at about $4\sigma$ from each other, is a safe default choice. We performed preliminary experiments with different values of $C$ and $\sigma$, which did not result in any significant improvement or decrease of performance, suggesting that the behaviour of SIRENE is robust to variations in its parameters around these default values. All results below were obtained with this default parameter choice.

### 2.3 Choice of negative examples

SIRENE being a supervised inference algorithm, two sets of positive and negative training examples are needed for each SVM. Although regulations reported in databases such as RegulonDB can safely be taken as positive training examples, the choice of negative examples is more problematic for two reasons. First, few information is published and archived regarding the fact that a given TF is found not to regulate a given target gene. Hence there is no systematic source of negative examples for our problem. A natural choice is then to take TF–gene pairs not reported to have regulatory relationships in databases as negative examples, mixing both true negative and false negative. In that case, we are then confronted with the second problem which is that, once a hard-margin SVM is trained on positive and negative examples, it always predicts significantly negative scores on negative examples used during training. As a result it is not possible to use the SVM score on genes used during training if we want to find TF-pairs that were wrongly assigned to the negative class.

To overcome this issue, we propose the following scheme. Let us suppose we want to predict whether genes are regulated or not by a given TF. All genes known to be regulated by this TF form a set of positive examples, and no prediction is needed for them. The other genes are split in three subsets of roughly equal size. Then, in turn, each subset is taken apart, and an SVM is trained with all positive examples and all genes in the two other subsets as negative examples. The SIRENE score for the genes in the subset left apart is the SVM prediction score on these genes, which were not used during SVM training. Repeating this loop 3 times, we obtain the SIRENE score for all genes with no known regulation by the TF. This process is then repeated for all other TF one by one. The advantage of this procedure is that, even though there are false negatives in the training set of each SVM, the predictions on the genes not used during training can still be positive if some of these genes look similar to the positive training examples.

### 2.4 CLR

We compare the performance of SIRENE with CLR, a method for gene network reconstruction from gene expression data that was shown by Faith *et al.* (2007) to be state-of-the-art on a large-scale benchmark evaluation. CLR is an extension of the relevance networks class of algorithms (Butte *et al.*, 2000), which predicts regulations between TF and genes when important mutual information can be detected. In the case of CLR, an adaptive background correction step is added to the estimation of mutual information. For each gene, the statistical likelihood of the mutual information score is computed within its network context. Then, for each TF–target gene pair, the mutual information score is compared to the context likelihood of both the TF and the target gene, and turned into a $z$-score. Putative TF–gene interactions are then ranked by decreasing $z$-score.

### 2.5 Experimental protocol

In order to assess the performance of SIRENE as an inference engine, and compare it with other existing methods, we test it on a benchmark of a known regulatory network. However, SIRENE being a supervised method, we adopt a cross-validation procedure to make sure that its performance is measured on prediction not used during the model training step. Consequently we adopt the following 3-fold cross-validation strategy, coherent with the SIRENE protocol to make predictions explained in Section 2.3. Given a set of TF, a set of genes, and a set of known TF–gene regulations within these sets, we split randomly the set of genes in three parts, train the SVM for each TF on two of these subsets and evaluate their prediction quality on the third subset, i.e. on the regulations of those genes that were not used during training (Fig. 1). This process is repeated 3 times, testing successively on each subset, and the prediction qualities of all folds are averaged.

In this cross-validation procedure, a particular attention must be paid to the existence of transcription units and operons in *E.coli*. Indeed, a given TF typically regulates all genes within an operon, which moreover usually have very similar expression profiles. As a result, if genes within an operon are split between a training and

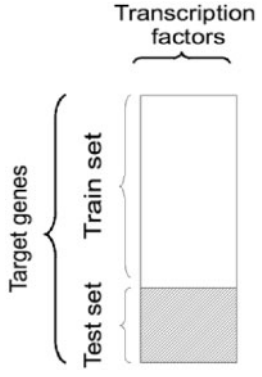


**Fig. 1.** Cross-validation for the transcriptional regulatory graph.

a test set, then the SVM prediction is likely to be correct simply because the SVM will predict that a test gene with a profile very similar to a training gene should be in the same class. In other words, the SVM can probably easily recognize operons and make correct predictions due to the presence of operons. However, we are interested here in the prediction of inference of regulations for new operons. To simulate this problem in our cross-validation setting, we make sure that all genes that belong to the same operon are in the same subset of genes, i.e. are always either in the training set or in the test set together. In our experiments below we report results both on a classical cross-validation setting and on this particular scheme that preserves the integrity of operons in the train/test splits.

The CLR algorithm is evaluated with the same protocol. However, since CLR is unsupervised, the training set is not used in each fold, and the final receiving operating characteristic (ROC) and precision/recall (PR) curves are equivalently obtained by computing the curves on all genes simultaneously.

To evaluate the quality of a prediction we rank all possible TF–gene regulation in the test set by decreasing score, and compute both the ROC curve and the PR curve. The ROC curve plots the recall, i.e. the percentage true interactions that have a score above a threshold, as a function of the false positive rate, i.e. the fraction of negative interactions that have a score above a threshold, when the threshold varies. The PR curve plots the precision, i.e. the percentage of true positive among the predictions above a threshold, as a function of recall, when the threshold varies. One ROC and PR curve is obtained in each fold of cross-validation, and these curves are averaged over the three folds to yield the final estimated ROC and PR curve.

## 3 DATA

We used in our experiments the expression and regulation data made publicly available by Faith *et al.* (2007) for *E.coli*, and downloaded from http://gardnerlab.bu.edu/netinfer_plos_2007/?page_id=5. The expression data consist of a compendium of 445 *E.coli* Affymetrix Antisense2 microarray expression profiles for 4345 genes. The microarrays were collected under different experimental conditions such as PH changes, growth phases, antibiotics, heat shock, different media, varying oxygen concentrations and numerous genetic perturbations. The expression data for each gene were normalized to zero mean and unit standard deviation. The regulation data consist of 3293 experimentally confirmed regulations between 154 TF and 1211 genes, extracted from the RegulonDB database (Salgado *et al.*, 2006).

We downloaded the list of 899 known operons in *E.coli* from RegulonDB. Each operon contains one or several genes, and each gene belongs to at most one operon. Genes not present in any of the RegulonDB were considered to form an operon by themselves, resulting in a total of 3360 operons for the 4345 genes. This operon information was used to create the folds in the cross-validation procedure, as explained in Section 2.5.

## 4 RESULTS

SIRENE was compared to CLR and other algorithms on the *E.coli* benchmark used by Faith *et al.* (2007) and described in the previous section. Figure 2 shows the ROC and PR curves of CLR and SIRENE. The two curves for the later, labeled SIRENE and SIRENE-Bias, are respectively obtained when we use the cross-validation protocol presented in Section 2.5 and when we use a classical cross-validation scheme where genes within a known operon can be split between training and test sets.

CLR scores were obtained directly from Faith *et al.* (2007). The PR curve of CLR is similar to that presented by Faith *et al.* (2007), confirming that we use the exact same benchmark. Both for ROC and PR, SIRENE performance curves are significantly above CLR. SIRENE-bias is itself much better than SIRENE, confirming the importance of the evaluation bias if operons are split artificially between training and test sets in the cross-validation procedure. In what follows, we restrict ourselves to the analysis of the results of SIRENE in the correct cross-validation protocol.

The PR curve is particularly relevant because the number of true regulations is very small compared to the total number of possible TF–gene pairs. We see that the recall obtained by SIRENE, i.e. the

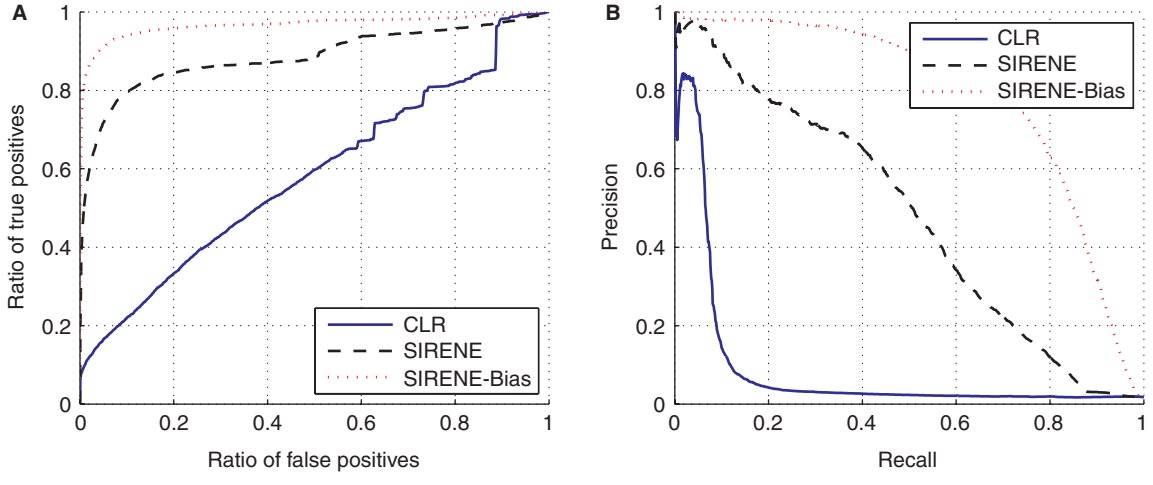


**Fig. 2.** Comparison of performance between CLR and SIRENE. (**A**) ROC curves and (**B**) precision/recall curves. The SIRENE curve corresponds to the SIRENE algorithm evaluated by 3-fold cross-validation, when genes within an operon are never split between the training and the test set. The SIRENE-bias curve is the same algorithm evaluated by classical 3-fold cross-validation, where genes are randomly assigned to training and test sets.

proportion of known regulations that are correctly predicted, is several times larger than the recall of CLR at all levels of precision. More precisely, Table 1 compares the recalls of SIRENE (bold figures), CLR and several other methods at 80 and 60% precision. The other methods reported are relevance network (Butte *et al.*, 2000), ARACNe (Margolin *et al.*, 2006), and a Bayesian network (Friedman *et al.*, 2000) implemented by Faith *et al.* (2007). The performance of these three methods was taken directly from Faith *et al.* (2007).

At 60% precision, SIRENE predicts 6 times more known regulations than CLR, which was the best among all methods tested on this benchmark by Faith *et al.* (2007). With 44.5% recall at this precision level, the performance of SIRENE allows one, in principle, to retrieve almost half of all known regulations.

The main conceptual difference between SIRENE and other methods is that SIRENE is a supervised method that requires known regulations to train its models. As an attempt to understand why the performance of SIRENE was better than that of other state-of-the-art unsupervised methods, we reasoned that TF with a large number of known regulated target genes could better take advantage of the supervised setting, and therefore that predictions for these TF should in general be better than predictions for TF with few known targets. To validate this hypothesis, we computed the ROC curve for SIRENE by cross-validation, restricted to the prediction of targets for each individual TF in turn. For each TF, we then computed the area under the ROC curve (AUC) as an indicator of how well the targets of each particular TF are predicted. We did this estimation for both CLR and SIRENE, and show in Figure 3 the distributions of AUC scores for all TF as a function of the number of known target genes in RegulonDB, for both CLR and SIRENE. As expected, the values for SIRENE tend to be larger than those for CLR. More importantly, we observe in the SIRENE plot a trend to have better AUC values for TF trained on more known targets. This trend is not present for CLR, which does not benefit from the knowledge of more or less targets for each TF. This result was expected and suggests that, as our knowledge expands and the number of known regulations continues to increase, so will the performance of supervised methods like SIRENE.

**Table 1.** Recall of different gene regulation prediction algorithm at different levels of precision (60 and 80%)

| Method | Recall at 60% (%) | Recall at 80% (%) |
|---|---|---|
| SIRENE | **44.5** | **17.6** |
| CLR | 7.5 | 5.5 |
| Relevance networks | 4.7 | 3.3 |
| ARACNe | 1 | 0 |
| Bayesian network | 1 | 0 |

The values for relevance network, ARACNe and Bayesian network were taken from Faith *et al.* (2007).

Having validated the relevance and performance of SIRENE on the regulonDB benchmark, we performed a global prediction of the *E.coli* regulatory network at 60% precision in order to predict new regulations in *E.coli*. More precisely, for each of the 154 TF with at least one known target in RegulonDB we computed the SIRENE score for all *E.coli* genes (4345 in total) that were not known targets, using the protocol described in Section 2.3. The RegulonDB database contained 3293 known TF-target regulations, so we assigned a score to the $4345 \times 154 - 3293 = 665837$ other candidate TF–gene pairs. From the cross-validation experiment we calibrated the level of SIRENE score threshold associated with various levels of precision. We selected all pairs with a score above a threshold of $-0.41$, corresponding to an estimated precision of 60%. At this threshold, 991 new regulations were predicted in addition to the 3293 known ones. Combining known and predicted regulations we obtained a regulatory network with 4284 edges involving 1688 genes.

In order to illustrate some predicted regulations, we focus now on the regulations of TF by other TF. Removing all non-TF genes of the predicted network, we obtain a graph with 131 TF and 349 interactions among them (TF with no interaction were removed). Among them, the *rpoD* gene, which codes for the RNA polymerase sigma factor, accounts alone for 85 regulations. In order to obtain a picture easier to visualize with the Cytoscape software (Shannon *et al.*, 2003), we removed rpoD from this graph,

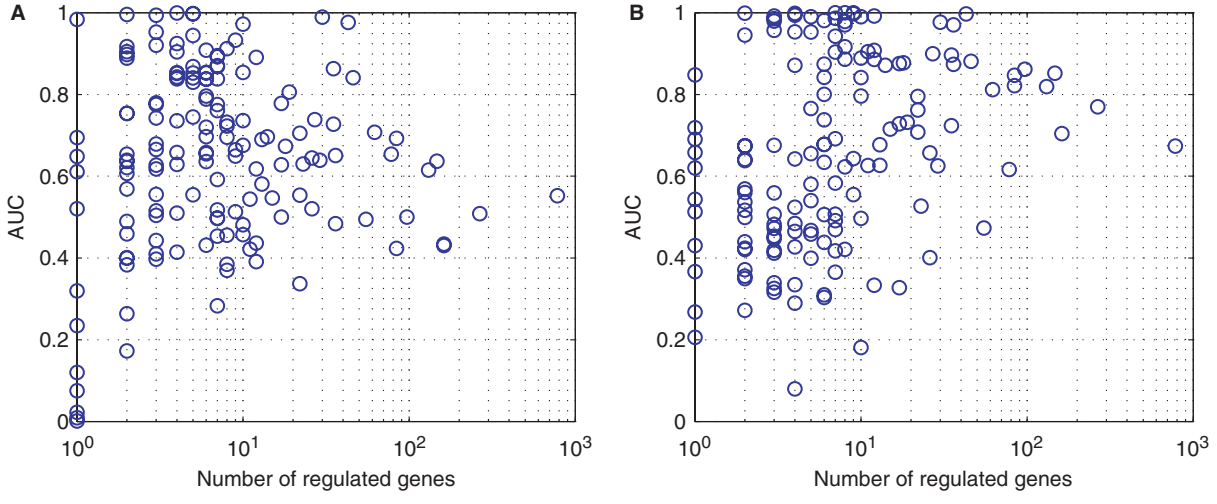


**Fig. 3.** AUC per TF as a function of the number of regulated genes. (**A**) CLR and (**B**) SIRENE.

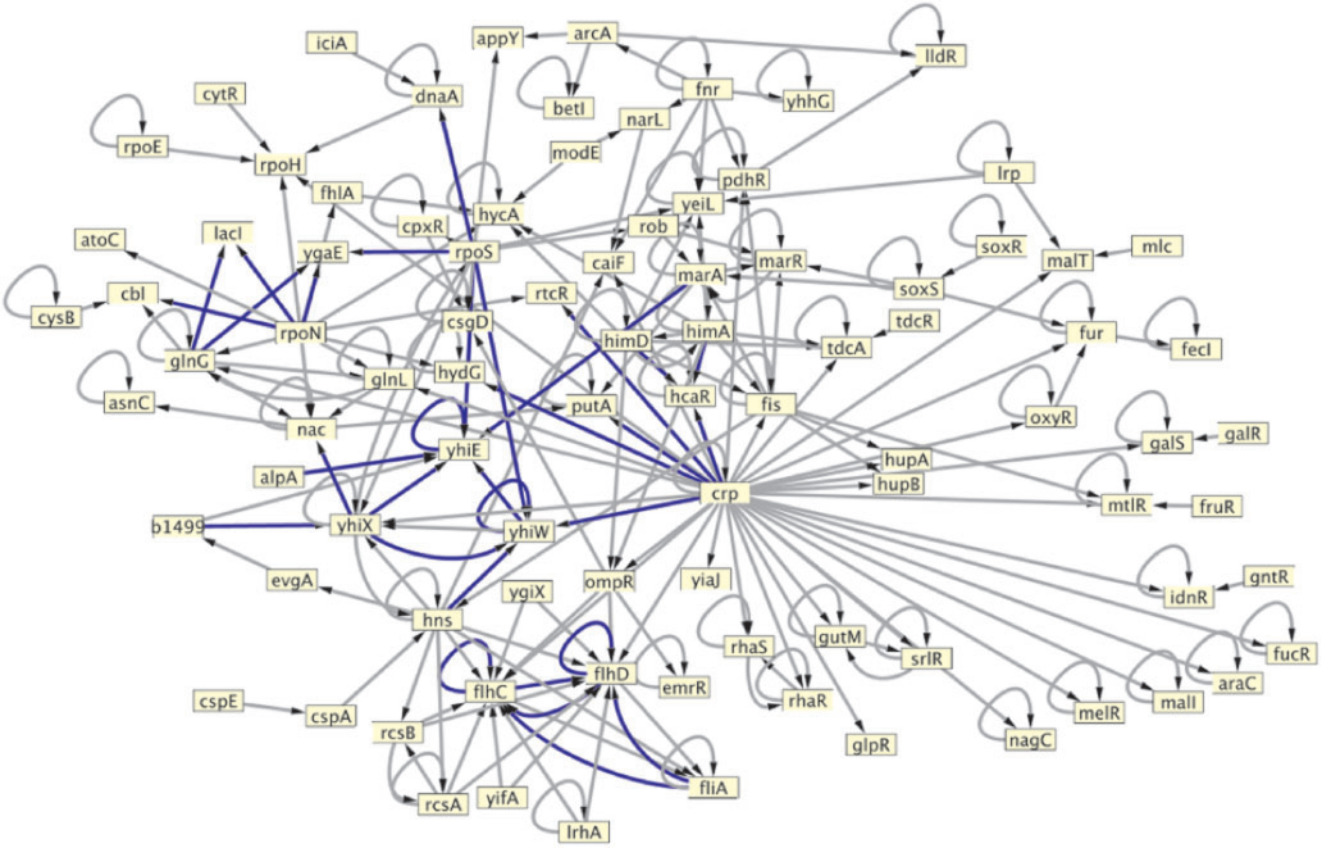


**Fig. 4.** Main connected component of the predicted regulatory network among TF of *E.coli*, at an estimated 60% precision level. For clarity purpose the rpoD gene was removed from this picture. Gray arrows indicate known regulations, blue arrows indicate new predicted interactions.

and only kept the main connected component which is shown in Figure 4. This core regulatory network involves 90 TF, and combines 196 known regulations among them with 32 predicted ones.

Most regulations in this densely connected region of the *E.coli* regulatory network have been investigated in detail, and it is not a surprise that the number of newly predicted regulations is limited. Still a quick survey of the literature can confirm some of these predictions. For example, four new regulators are predicted for *yhiW* (*crp,hns,rpoS,yhiX* and itself), which is itself predicted to regulate *yhiE*. Although these regulations were not present in the database used to train the model, they are confirmed by the literature. The GadW protein coded by *yhiW* is a regulator that participates in controlling several genes of the acid resistance system. It is indeed regulated by the proteins coded by *yhiX* and by the general proteins *crp,hns,rpoS* that control resistance to acidity through the gad system that utilizes two isoforms of glutamate decarboxylase encoded by gene regions *gadA* and *gadB* and a putative glutamate: aminobutyric acid antiporter encoded by *gadC* (Ma *et al.*, 2003; Tucker *et al.*, 2002; Waterman and Small, 2003). Another predicted regulation that was confirmed by a literature search is the dependence of *hcaR*, a TF involved in the oxidative stress response, by a functional CAP protein encoded by the *crp* gene (Turlin *et al.*, 2001). Although preliminary, these first validations confirm the relevance of the approach and may suggest further experimental validations for subsystems of interest.

## 5 DISCUSSION

We presented SIRENE, a new method for gene regulatory inference from a compendium of gene expression data. It is conceptually simple and computationally efficient, scaling to predictions at the level of whole genomes. Contrary to other methods for regulation inference based on the detection of similarity or causality between expression profiles of TF and their targets, we make no such hypothesis. Instead, we make the natural hypothesis that genes regulated by the same TF are likely to exhibit similar expression variations. Hence, the method is supervised, in the sense that it needs as input, besides expression data, a set of known TF–target pairs. We tested SIRENE on a benchmark experiment recently proposed to assess the performance of gene regulatory network inference methods on the reconstruction of *E.coli* regulations. At 60% precision, it predicts 6 times more regulations than CLR, a state-of-the-art method on this benchmark.

SIRENE is easy to implement and scales well to large-scale inference. Indeed, the main idea behind SIRENE is to decompose the network inference into a set of local binary classification problems, aimed at discriminating targets from non-targets of each

TF. Although we used an SVM as a basic algorithm to solve these local problems, any algorithm for pattern recognition may be used instead. Each local problem involves at the most a training set of a few thousands genes, easily manageable by most machine-learning algorithms. This strategy also paves the way to the use of other genomic data to predict regulation. Indeed, local models for gene classification often improve in performance when several data, such as phylogenetic or cell subcellular localization information is available, and SVM provide a convenient framework to practically perform this data integration (Bleakley *et al.*, 2007; Lanckriet *et al.*, 2004). Another interesting features of SIRENE is its ability to predict self-regulation, that other methods have generally difficulties to deal with.

An important limitation of SIRENE is its inability to predict targets of TF with no a priori known target. More generally, the performance of SIRENE tends to decrease when few targets are known. Thus, for example, it cannot be used to discover new transcription factors. An interesting direction of future research is therefore to extend the predictions to TF with no known target. A possible direction may be to combine the supervised approach with other non-supervised approaches in some meaningful way.

Finally, we note that the evaluation criteria used in the benchmark experiment through global precision/recall curve, although more relevant than the ROC curve, certainly remains to be improved. The fact that it is biased towards TF for which we know many regulations (the hubs) implies that the method is less likely to propose reliable new interactions for TF with few known neighbors. This is a great disadvantage in some applications where we are interested in 'orphan' TF.

## REFERENCES


Akutsu,T. *et al.* (2000) Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. *J. Comput. Biol.*, **7**, 331–343.

Bansal,M. *et al.* (2006) Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, **22**, 815–822.

Bansal,M. *et al.* (2007) How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, **3**, 78.

Ben-Hur,A. and Noble,W.S. (2005) Kernel methods for predicting protein-protein interactions. *Bioinformatics*, **21**(Suppl. 1), i38–i46.

Bleakley,K. *et al.* (2007) Supervised reconstruction of biological networks with local models. *Bioinformatics*, **23**, i57–i65.

Butte,A.J. *et al.* (2000) Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc. Natl Acad. Sci. USA*, **97**, 12182–12186.

Chen,T. *et al.* (1999) Modeling gene expression with differential equations. *Pac. Symp. Biocomput.*, 29–40.

Chen,K.-C. *et al.* (2005) A stochastic differential equation model for quantifying transcriptional regulatory network in *Saccharomyces cerevisiae*. *Bioinformatics*, **21**, 2883–2890.

di Bernardo,D. *et al.* (2005) Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat. Biotechnol.*, **23**, 377–383.

Faith,J.J. *et al.* (2007) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, **5**, e8.

Friedman,N. *et al.* (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**, 601–620.

Gardner,T.S. *et al.* (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, **301**, 102–105.

Lanckriet,G.R.G. *et al.* (2004) A statistical framework for genomic data fusion. *Bioinformatics*, **20**, 2626–2635.

Ma,Z. *et al.* (2003) GadE (YhiE) activates glutamate decarboxylase-dependent acid resistance in *Escherichia coli* K-12. *Mol. Microbiol.*, **49**, 1309–1320.

Margolin,A.A. *et al.* (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7** (Suppl. 1), S7.

Salgado,H. *et al.* (2006) Regulondb (version 5.0): *Escherichia coli* k-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res.*, **34**(Database issue), D394–D397.

Schölkopf,B. *et al.* (2004) *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA.

Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.

Tavazoie,S. *et al.* (1999) Systematic determination of genetic network architecture. *Nat. Genet.*, 22(3) 281–285

Tegner,J. *et al.* (2003) Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proc. Natl Acad. Sci. USA*, **100**, 5944–5949.

Tucker,D.L. *et al.* (2002) Gene expression profiling of the ph response in *Escherichia coli*. *J. Bacteriol.*, **184**, 6551–6558.

Turlin,E. *et al.* (2001) Regulation of the early steps of 3-phenylpropionate catabolism in *Escherichia coli*. *J. Mol. Microbiol. Biotechnol.*, **3**, 127–133.

Vapnik,V.N. (1998) *Statistical Learning Theory*. Wiley, New York.

Waterman,S.R. and Small,P.L.C. (2003) Transcriptional expression of *Escherichia coli* glutamate-dependent acid resistance genes gadA and gadBC in an hns rpoS mutant. *J. Bacteriol.*, **185**, 4644–4647.

Yamanishi,Y. *et al.* (2004) Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, **20**, i363–i370.