A model for gene deregulation detection using expression data

Thomas Picchetti, Julien Chiquet, Mohamed Elati, Pierre Neuvial, Rémy Nicolle, Etienne Birmelé

METHODOLOGY

A model for gene deregulation detection using expression data

Thomas Picchetti[1], Julien Chiquet[2], Mohamed Elati[3], Pierre Neuvial[2], Rémy Nicolle[3,4] and Etienne Birmelé[1*]

*Correspondence: etienne.birmele@parisdescartes.fr
[1]Laboratoire MAP5, Université Paris Descartes and CNRS, Sorbonne Paris Cité, 45 rue des Saints-Pères, 75270 Paris Cedex 06, France
Full list of author information is available at the end of the article

Abstract

In tumoral cells, gene regulation mechanisms are severely altered. Genes that do not react normally to their regulators' activity can provide explanations for the tumoral behavior, and be characteristic of cancer subtypes. We thus propose a statistical methodology to identify the misregulated genes given a reference network and gene expression data.

Our model is based on a regulatory process in which all genes are allowed to be deregulated. We derive an EM algorithm where the hidden variables correspond to the status (under/over/normally expressed) of the genes and where the E-step is solved thanks to a message passing algorithm. Our procedure provides posterior probabilities of deregulation in a given sample for each gene. We assess the performance of our method by numerical experiments on simulations and on a bladder cancer data set.

Keywords: regulatory network; belief propagation; EM algorithm; deregulation; inference

Background

Various mechanisms affect gene expression in tumoral cells, including copy number alterations, mutations, modifications in the regulation network between the genes. A simple strategy to identify genes affected by these phenomena is to perform differential expression analysis. Results can then be extended to the scale of pathways using enrichment analysis [1] or functional class scoring [2]. However, such a strategy is blind to small variations in gene expression, especially as multiple testing correction applies. Moreover, it does not take interdependence between genes into account and can mark an expression change as abnormal when actually it is induced by a change in the regulators' activity. To overcome these drawbacks, an alternative strategy is to identify the affected genes by pointing important changes in the gene regulatory network (GRN) of the tumoral cell. Such an approach furthermore corresponds to the modelisation of phenomena altering regulation, as for instance mutations in regulatory regions [3].

The first step towards this is to procure a GRN. It can be obtained from curated databases or, in order to obtain tissue or condition-specific networks, reconstructed from expression data. In the latter case, the inference can be done by relying either on discrete or continuous models. In the discrete framework, gene expression profiles are discretized into binary or ternary valued variables (underexpressed/normal/overexpressed). The regulation structure is then given by a list

of truth tables [4]. This approach allows in particular to take coregulation into account, that is to require the activity of a whole set of co-activators or co-inhibitors to activate or inhibit the target [5, 6]. In the continuous case, inference can be done in a regression framework, where the expression of each target gene is explained by all its potential regulator genes. An edge is drawn between two genes if the corresponding regression coefficient is significantly different from zero, which can be deciphered by performing variable selection in the regression model. A popular choice for this task is to rely on sparsity-inducing penalties like the Lasso and its by-products [7, 8]. In particular, some variants allow to account for co-regulation by favoring predefined groups of regulators acting together in a sign-coherent way [9]. Other forms of penalties encourage a predefined hierarchy between the predictors [10], *i.e.* the regulator genes in the case at hand.

To unravel deregulated genes by means of GRN, a first possibility is to infer several networks independently (one for each tissue) and to compare them. However, due to the noisy nature of transcriptomic data and the large number of features compared to the sample size, most of the differences found in the networks inferred independently may not be linked with underlying biological processes. Methods have therefore been developed to infer several networks jointly to share similarities between the different tissues and penalize the presence of an edge in only one of them. Such methods exist for both time series [11] or steady-state [12] data.

A second possibility is to assess the adequacy of gene expression in tumoral cell to a reference GRN, in order to exhibit the more striking discrepancies – *i.e.* the regulations which are not fulfilled by the data –. In this perspective, [13] use an heuristic in a Boolean framework to update the regulatory structure by minimizing the discrepancies between the reference GRN and a new data set. A similar approach is depicted in [14] to predict the discrepancies and the unobserved genes of the network. More methods analyzing the coherence between known signaling pathways and gene data sets can be found in the review [15]. Still, they focus on checking the validity of the network rather than highlighting genes with an abnormal behavior.

At the pathway level rather than the gene level, it is possible to look for sample-specific regulation abnormalities by using SPIA [16]. PARADIGM [17] generalizes SPIA on heterogeneous data (DNA copies, mRNA and protein data). Moreover, it determines a score of activity for each gene of a pathway for each sample of the data set, and the use of hidden variables allows to compute this score even if some of the genes of the pathway are not measured. The method is however not network-wide in the sense that each gene has a deregulation score by pathway it belongs to, and pathways are treated independently. Moreover, as the pathways are extracted from curated databases, the regulations taken into account are not tissue-specific.

The aim of this paper is to develop a methodology to provide a network-wide deregulation score for each gene and each sample by taking the whole regulation network into account. For this purpose, we introduce a model based on a regulatory process in which genes are allowed to be deregulated, *i.e.* not respond to their regulators as expected. An EM strategy is proposed for parameter inference, where the hidden variables correspond to the status (under/over/normally expressed) of the genes. The E-step is solved thanks to a message passing algorithm. At the end

of the day, the procedure provides *posterior* probabilities of deregulation in a given sample for each target gene. We assess the performance of our method for detecting deregulations on simulated data. We also illustrate its interest on a bladder cancer data set, where we study the deregulations according to two reference GRN obtained by two state-of-the-art network inference procedures on a consensus expression data set.

Methods

The model

Our model draws inspiration from LICORN [5], a model originally developed for network inference purposes. LICORN considers a regulation structure in which genes are either regulators (transcription factors – TFs) or target genes. The expressions are discretized and each gene g is characterized by a ternary value $S_g \in \{-1, 0, +1\}$ encoding its expression status – under-, normally, or over-expressed. The regulation of each target gene g is governed by a set of co-activators $A(g)$ and co-inhibitors $I(g)$ among the TFs. Those sets are endowed with some "collective status" described by variables S_g^A and S_g^I, assuming that regulation works in a cooperative way: hence, the collective state of a set of regulators is over- (resp. under-) represented if and only if all elements in the set share the same status. Finally, the status S_g of the target gene g is deduced from S_g^A and S_g^I by following Truth Table 1.

In order to detect deregulated target genes given a regulatory network and gene expression profiles, we apply two major modifications to the LICORN model: first, we avoid discretization of the data by considering all the ternary variables introduced so far as hidden random variables. The expression X_g of a gene g is assumed to follow a normal distribution with parameters that depend on the hidden status, *i.e.*, $X_g|S_g = s \sim \mathcal{N}(\mu_s, \sigma_s)$. Second, we introduce for each gene an indicator variable D_g for deregulation, such that $D_g = 1$ with probability ϵ. Renaming the result of the truth table by S_g^R, the final status of the target is then deduced from the values of D_g and S_g^R:

$$\begin{cases} S_g = S_g^R & \text{if } D_g = 0, \\ \forall s \neq S_g^R, \mathbb{P}(S_g = s) = \frac{1}{2} & \text{if } D_g = 1. \end{cases}$$

For completeness, we must specify the distribution of the hidden states S_g for each TFs: we assume independent multinomial distributions with parameters $\boldsymbol{\alpha} = (\alpha_-, \alpha_0, \alpha_+)$.

The model is summarized for one target gene in Figure 1. For the sake of conciseness, the vector $\boldsymbol{\theta}$ entails all parameters of the models, that is, the means and standard deviations of the Gaussians, the vector $\boldsymbol{\alpha}$ of proportions and the deregulation rate ϵ. The data set contains n samples, r TFs and t target genes. We denote by $\mathbf{Z}$ the $n \times (r + 5t)$ matrix of all hidden states and by $\mathbf{X}$ the $n \times (r + t)$ matrix of all expression variables.

Note that the dependencies among variables are acyclic, implying that the likelihood can be decomposed in a product.

$$p(X,Z|\theta) = \prod p(S_j|\alpha) \times \prod p(S_i^A|S_j\ldots) \times \prod p(S_i^I|S_j\ldots) \times \prod p(S_i^R|S_i^I,S_i^A) \\ \times \prod p(D_i|\epsilon) \times \prod p(S_i|S_i^R,D_i) \times \prod p(X_k|S_k,\mu,\sigma)$$

For sake of readability, the indices of the products are omitted in the above formula. However, it should be clear when the product runs over target genes, regulator genes or all of them.

Estimation algorithm

As usual with latent variable models, the likelihood is intractable as the number of potential states of the hidden variables grows exponentially with the number of variables. Therefore, we adopt an EM-like strategy [18] by iterating the following steps, starting from an initial guess $\boldsymbol{\theta}^0$ of the model parameters:

E-step: Fix $\boldsymbol{\theta}$ and compute the conditional probability distribution of the hidden variables, given the observed expression values: $q(\mathbf{Z}) = \mathbb{P}(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})$

M-step: Fix q and find $\boldsymbol{\theta}$ that maximizes $\sum q(\mathbf{Z}) \log \mathbb{P}(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})$

Step E. The first issue at stake in the E-step is to deal with the number of potential states for the hidden variables of all the genes. Fortunately, we only need their marginal distributions in the M step, as will be shown in the corresponding section. Still, we need a way to compute these marginals without having to compute the joint distribution first.

To handle this issue, we rely on Belief Propagation [19] – a.k.a *message-passing algorithm* – to perform the E step, since the probability distribution arising from our model is easily represented as a factor graph. Indeed, consider a set of discrete values for all variables S_g^A, S_g^I, S_g^R and D_g. Conditionally on $\mathbf{X}$, the probability for the discrete variables to match the given value is proportional to the product of the following factors:

1. α_{S_g} for each regulator gene $g \in R$;
2. ϵ if $D_g = 1$, and $\frac{1-\epsilon}{2}$ if $D_g = 0$, for each target gene $g \in T$;
3. $\frac{1}{\sigma} \exp \frac{-(X_g-\mu)^2}{2\sigma^2}$ for each gene $g \in G$ (regulator or target), where μ and σ are the mean expression and standard deviation associated to state S_g;
4. a factor equal to one if S_g^A correctly represents the collective state of g's activators, and zero otherwise;
5. a factor equal to one if S_g^I correctly represents the collective state of g's inhibitors, and zero otherwise;
6. a factor equal to one if S_g^R is the entry in Table 1 corresponding to S_g^A and S_g^I, and zero otherwise;
7. a factor equal to one if either $D_g = 0$ and $S_g = S_g^R$ or $D_g = 1$ and $S_g \neq S_g^R$, and zero otherwise.

This factorization translates into the factor graph depicted in Figure 2 (a graph whose nodes are the variables and the above factors, each factor being connected to the variables it depends on). We use the *SumProduct* Belief Propagation algorithm,

implemented in the Dimple library [20] to compute approximated marginals of every hidden variable, given the regulation network, the parameter set, and the expression values. In the case where multiple samples are given, this can be done separately for each one since the samples are considered as independent.

Step M. In this step we keep the probability distribution q fixed and look for the parameters $\boldsymbol{\theta}$ that maximize

$$\sum_{\mathbf{Z}} q(\mathbf{Z}) \log \mathbb{P}(\mathbf{X}, \mathbf{Z}|\theta)$$

Since $\mathbb{P}(\mathbf{X}, \mathbf{Z}|\theta)$ is a product of simple factors, its logarithm is the sum of these factors. Also, note that boolean factors (4-7) can be omitted since they have no effect on the sum: whenever $q(Z) \neq 0$, these factors must be equal to 1 hence the logarithm is 0.

Calling G the set of genes, $R \subset G$ the set of regulators and $T \subset G$ the set of target genes, we are left to maximize the sum over all samples of

$$\begin{aligned}
&\sum_{g \in R} \sum_{Z} q(Z) \log \alpha_{S_g} \\
&+ \sum_{g \in T} \sum_{Z} q(Z) \left(D_g \log \epsilon + (1 - D_g) \log \frac{1-\epsilon}{2} \right) \\
&\qquad + \sum_{g \in G} \sum_{Z} q(Z) \left(\frac{-(X_g - \mu_{S_g})^2}{2\sigma_{S_g}^2} - \log \sigma_{S_g} \right)
\end{aligned}$$

These three terms depend on separate parameters and can be maximized separately. Moreover, we only require the marginals of variables S_g and D_g for this task, and not the full distribution q. Denoting by I the set of samples, it is straightforward to show that the former sum is maximized for the following parameters:

$$\alpha_- \propto \sum_{i \in I} \sum_{g \in R} q(S_{i,g} = -1), \ \alpha_0 \propto \sum_{i \in I} \sum_{g \in R} q(S_{i,g} = 0), \ \alpha_+ \propto \sum_{i \in I} \sum_{g \in R} q(S_{i,g} = +1),$$

$$\epsilon \propto \sum_{i \in I} \sum_{g \in T} q(D_{i,g} = 1), \qquad (1 - \epsilon) \propto \sum_{i \in I} \sum_{g \in T} q(D_{i,g} = 0),$$

$$\mu_s = \frac{\sum_i \sum_g q(S_{i,g} = s) X_{i,g}}{\sum_i \sum_g q(S_{i,g} = s)}, \qquad \sigma_s^2 = \frac{\sum_i \sum_g q(S_{i,g} = s)(\mu_s - X_i)^2}{\sum_i \sum_g q(S_{i,g} = s)}$$

Complexity analysis

Step M only involves computing a few sums of size [number of genes]×[number of samples] and is not time-consuming. Step E performs for each sample a fixed number of passes of Belief Propagation in the factor graph. Each pass consists in updating every node with information from its neighbors. The complexity of updating a factor grows exponentially with its degree, therefore it is important to limit the number of variables of each factor. It is done by replacing the factors corresponding to the

types (4) and (5) in Figure 2 by tree-like structures with many factors having 3 variables each.

With this approach the graph has approximately $N = 2E + G$ nodes, where E is the number of regulator-target edges in the regulation network, and G the number of genes. A personal computer performs a few million node updates per second, thus step E will run in t seconds if $N \times$[number of passes]$\times$[number of samples] is not much greater than t millions.

Regulatory network inference from expression data

To apply our methodology to real data, we use two different inference methods.

LICORN. The first one, named hLICORN, corresponds to the LICORN model and is available in the CoRegNet Bioconductor package [6]. In a first step, it efficiently searches the discretized gene expression matrix for sets of co-activators and co-repressors by frequent items search techniques and locally selects combinations of co-repressors and co-activators as candidate subnetworks. In a second step, it determines for each gene the best sets among those candidates by running a regression. hLICORN was shown to be suitable for cooperative regulation detection [5, 6].

Cooperative-Lasso + Stability Selection. The second inference procedure applies in a continuous setup. It consists in two steps: first, a selection step performed with a sparse procedure; and second, a resampling step whose purpose is to stabilize the selection for more robustness in the reconstructed network. Here are some details.

Step 1: selection. For each target gene, a sparse penalized regression method is used to select the set of relevant co-activators and co-inhibitors among all possible transcription factors. When no special structure is assumed in the network, this task can be performed with the Lasso penalty, as it was successfully applied for network inference in [8]. Here, however, we are looking for sets of regulators that work group-wise, either as co-activators or co-inhibitors. To favor such a structure, we build on the penalty proposed in [12, 9] that encourages selection of predefined groups of variables sharing the same sign (thus being either co-activators or co-inhibitors). This regularization scheme is known as the "cooperative-Lasso". It was originally designed to work with a set of groups that form a partition over the set of regulators. Here, we extend this method to a structure that defines a hierarchy (or tree) on the set of regulators R . We denote by $\mathcal{H} = \{\mathcal{H}_1, \ldots, \mathcal{H}_K\}$ this structure, with $\mathcal{H}_k$ the kth (non-empty) node of the hierarchy.

Technically, the optimization problem solved for selecting regulators of gene g is the following penalized regression problem

$$\hat{\boldsymbol{\beta}}^{(g)} = \underset{\boldsymbol{\beta}^{(g)} \in \mathbb{R}^{|R|}}{\arg\min} \frac{1}{2} \left\| \mathbf{X}_g - \mathbf{X}_R \boldsymbol{\beta}^{(g)} \right\|^2 + \lambda \sum_{k=1}^{K} \left\| \left(\boldsymbol{\beta}^{(g)}_{\mathcal{H}_k} \right)^+ \right\|_2 + \left\| \left(\boldsymbol{\beta}^{(g)}_{\mathcal{H}_k} \right)^- \right\|_2 ,$$

with $\mathbf{X}_g$ the expression profile of gene g and $\mathbf{X}_R$ the expression profiles of the regulators. The parameter $\lambda > 0$ tunes the amount of regularization, and thus the number of regulators associated with gene g; $\mathbf{v}^+$ and $\mathbf{v}^-$ are the positive, respectively the

negative elements of a vector $\mathbf{v}$, and $\mathbf{v}_{\mathcal{H}_k}$ the restriction of $\mathbf{v}$ to the elements in node $\mathcal{H}_k$ of the hierarchy. Hence, this penalty favors selection of sign-coherent groups of variables, like $(\boldsymbol{\beta}^{(g)}_{\mathcal{H}_k})^+$, standing for the estimated co-activators of gene g in node $\mathcal{H}_k$ of the hierarchy, or $(\boldsymbol{\beta}^{(g)}_{\mathcal{H}_k})^-$, the corresponding co-inhibitors.

Step2: Stabilization. We fit a sparse model as described above for each target gene, regressing on the same set of regulators R. The hierarchy $\mathcal{H}$ that we used is obtained by performing hierarchical clustering with average linkage on a distance based upon the correlation between expression profiles. We use the same λ for each gene, which is chosen large enough in order to select at least one set of regulators for all target genes. To select the final edges in the network, we rely on the stability selection procedure of [21], which was successfully applied to the reconstruction of robust regulatory networks in the case of a simple Lasso penalty [7], and is known to be less sensitive than selecting one λ per gene (*e.g.* by cross-validation). This technique consists in refitting the regression model on many subsamples obtained by drawing randomly $n/2$ observations from the original data set. We replicate 10,000 times this operation and obtain a estimated probability of selection for each edge. We fix the threshold in order to select a number of edges similar to LICORN, which corresponds to edges with a probability of selection greater than 0.65.

Results and Discussion

Classification performances on simulated data sets

In our experiments, the score $q(D_{i,g} = 1)$ is used to determine if gene g is deregulated or not in sample i. Performances are evaluated with Precision-Recall (PR) curves, which are known to be more informative than ROC curves or accuracy [22] when considering classification problem with very imbalanced data sets.

We generate expression data sets according to the model described earlier and feed them to the EM algorithm to evaluate its performance. To study the impact of each parameter, we try several values of this parameter while all others remain fixed to their default value. Ten data sets are generated and processed in each setting, resulting in 10 PR curves. We thus obtain clouds of curves, measuring both the variability for a given parameter set and the influence of the varying parameter.

We unsurprisingly note that σ has dramatic effect (see Figure 3). As a rule of thumb to distinguish two states from one another, the associated standard deviations must be smaller than the difference between their mean expressions.

Meanwhile, large values of ϵ mechanically result in better PR: the more the deregulated genes, the more the true positives among all positives (Figure 4).

On the contrary, all other parameter have little effect on the performance and we thus postpone the associated PR curves to the Additional File 1. Those parameters are μ, α, the number of passes in the Belief Propagation algorithm (as long as it is greater than five), the number of genes and the sample size (as long as their product is of several hundreds).

Managing the False Discovery Rate

Consider couples (i, g) whose deregulation score $q(D_{i,g} = 1) = s$: this score being a *posterior* probability, the expected proportion of true (respectively false) positives is s (respectively $1 - s$). Similarly, if K pairs pass the threshold, the expected number

of true positives among them is the sum of their scores, denoted by S. The false discovery rate (FDR) may be estimated by $(K - S)/K$. In practice, aiming for a particular FDR, one can start with a threshold of 1 and lower it gradually: as more pairs get selected, the ratio $(K-S)/K$ gradually increases. All one has to do is stop when it reaches the intended FDR. The concordance between the intended FDR and the actual proportion of false positives is illustrated on simulated data sets in the Additional File 1.

Tests on real data

We applied our method to the bladder cancer data set available in the `R`-package CoRegNet [6]. Expression data from patients with different status was pooled to infer gene co-regulatory networks with two independent procedures, namely *hLICORN* and the hierarchical *Cooperative-Lasso*. The inferred networks reflect the regulation trends over the whole set of 184 samples. Our EM algorithm is then run using the same expression data, but since samples are now treated individually, the results reflect how each sample violates the regulatory rules generally followed by the others.

On real data, the true deregulation status is unreachable. Hence, we match our result with Copy Number Alteration (CNA) data collected from the same samples, in order to support that our method correctly identifies deregulated gene-sample pairs. We do not expect CNAs to precisely coincide with failures of the regulation network, so we do not hope to detect exactly those pairs that present a CNA. However, the number of gene copies influences the expression independently from expression of the TFs [23]. We therefore expect to observe a link between CNA and gene deregulations.

To this end, we use CNA data provided by the CoRegNet package, associating to each gene-sample pair a copy number state: 0 for the diploid state (two copies), 1 for a copy number gain, -1 for a copy number loss, and 2 for a copy number amplification. Figure 5 compares the distribution of the perturbation scores across copy number states by representing, for each copy number class, the empirical cumulative distribution function of the perturbation scores. For each value s of the perturbation score in abscissa, the ordinate is the proportion of gene-sample pairs with a score greater than s. The fact that the curve corresponding to the diploid state is above all the other curves indicates that gene-sample pairs having a CNA are given a higher perturbation score diploid gene-sample pairs by our deregulation model. Although the difference seems slight, it is highly significant given the large number of scores, as indicated by the p-value of the Student test for the pairwise differences between the diploid state and each of the other altered states. As expected, the scores of the "amplification" state 2 are also higher than the scores of "gain" state 1.

Conclusion

In the present article, we develop a statistical model for gene expression based on a hidden regulatory structure. Given a reference GRN, it allows to determine which genes are misregulated in a sample, meaning an expression which does not matches the network given the expression of its regulators. Numerical experiments validate

the algorithmic procedure: when applied to bladder cancer data with known CNA, the deregulation score is higher in samples in which genes have an altered number of copies.

We believe that our methodology will be useful to understand which regulation mechanisms are altered in different cancer subtypes. Indeed, the results of our methodology are sample-specific. However, characterizing the deregulations which are common to most of the individuals suffering a given cancer subtype is a promising perspective.

The integration of CNA to the methodology, as already done in the context of differential expression [24], will also be considered in future work, as it would allow a better power for detecting genes suffering misregulation due to a copy alteration.

Availability of supporting data
The EM algorithm described in this article is available as a Java archive at http://www.math-info.univ-paris5.fr/∼ebirmele/
Bladder cancer data and hLicorn are available through the CoRegNet Bioconductor package.

Abbreviations
CNA: Copy Number Alteration GRN: Gene Regulatory Network PR curve: Precision-Recall ROC curve: Receiver Operating Characteristic curve TF: Transcription factor

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

Author's contributions
The work presented here was carried out in collaboration between all authors. ME and EB conceived the study. TP and EB designed it and wrote the manuscript. JC, PN and RN brought their expertise on inference and statistical interpretation on the real data. All authors provided valuable advises in developing the proposed method and modifying the manuscript. All authors read and approved the final manuscript.

Acknowledgements
The authors would like to thank François Radvanyi for helpful discussions.

Declarations
This work was partially supported the CNRS (CREPE, PEPS BMI). Publication charges were funded by CHIST-ERA grant (AdaLab, ANR 14-CHR2-0001-01).

Author details
[1]Laboratoire MAP5, Université Paris Descartes and CNRS, Sorbonne Paris Cité, 45 rue des Saints-Pères, 75270 Paris Cedex 06, France. [2]Laboratoire de Mathématiques et Modélisation d'Evry (LaMME), Université d'Evry-Val-d'Essonne/UMR CNRS 8071/ENSIIE/USC INRA, Evry, France. [3]institute of Systems and Synthetic Biology (iSSB), CNRS, University of Evry, France. [4]Institut Curie, PSL Research University, UMR 144 75248 Cedex 05, France, CNRS 75248 Paris Cedex 05, France.

References
1. Khatri, P., Draghici, S., Ostermeier, G.C., Krawetz, S.A.: Profiling gene expression using onto-express. Genomics **79**(2), 266–270 (2002). doi:10.1006/geno.2002.6698
2. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., Mesirov, J.P.: Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences **102**(43), 15545–15550 (2005). doi:10.1073/pnas.0506580102. http://www.pnas.org/content/102/43/15545.full.pdf
3. Melton, C., Reuter, J.A., Spacek, D.V., Snyder, M.: Recurrent somatic mutations in regulatory regions of human cancer genomes. Nature Genetics (2015)
4. Elati, M., Rouveirol, C.: Unsupervised Learning for Gene Regulation Network Inference from Expression Data: A Review, pp. 955–978. John Wiley and Sons, Inc., ??? (2011). doi:10.1002/9780470892107.ch41. http://dx.doi.org/10.1002/9780470892107.ch41
5. Elati, M., Neuvial, P., Bolotin-Fukuhara, M., Barillot, E., Radvanyi, F., Rouveirol, C.: Licorn: learning cooperative regulation networks from gene expression data. Bioinformatics **23**(18), 2407–2414 (2007). doi:10.1093/bioinformatics/btm352. http://bioinformatics.oxfordjournals.org/content/23/18/2407.full.pdf+html
6. Nicolle, R., Radvanyi, F., Elati, M.: Coregnet: reconstruction and integrated analysis of co-regulatory networks. Bioinformatics (2015)
7. Haury, A.-C., Mordelet, F., Vera-Licona, P., Vert, J.-P.: Tigress: Trustful inference of gene regulation using stability selection. BMC Systems Biology **6**(1), 145 (2012). doi:10.1186/1752-0509-6-145
8. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. Ann. Statist. **34**(3), 1436–1462 (2006). doi:10.1214/009053606000000281

9. Chiquet, J., Grandvalet, Y., Charbonnier, C., *et al.*: Sparsity with sign-coherent groups of variables via the cooperative-lasso. The Annals of Applied Statistics **6**(2), 795–830 (2012)
10. Jenatton, R., Audibert, J.-Y., Bach, F.: Structured variable selection with sparsity-inducing norms. The Journal of Machine Learning Research **12**, 2777–2824 (2011)
11. Kojima, K., Imoto, S., Yamaguchi, R., Fujita, A., Yamauchi, M., Gotoh, N., Miyano, S.: Identifying regulational alterations in gene regulatory networks by state space representation of vector autoregressive models and variational annealing. BMC Genomics **13**(Suppl 1), 6 (2012). doi:10.1186/1471-2164-13-S1-S6
12. Chiquet, J., Grandvalet, Y., Ambroise, C.: Inferring multiple graphical structures. Statistics and Computing **21**(4), 537–553 (2011)
13. Karlebach, G., Shamir, R.: Constructing logical models of gene regulatory networks by integrating transcription factor–dna interactions with expression data: An entropy-based approach. Journal of Computational Biology **19**(1), 30–41 (2012)
14. Guziolowski, C., Bourde, A., Moreews, F., Siegel, A.: Bioquali cytoscape plugin: analysing the global consistency of regulatory networks. BMC Genomics **10**(1), 244 (2009). doi:10.1186/1471-2164-10-244
15. Samaga, R., Klamt, S.: Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. Cell Communication and Signaling **11**(1), 43 (2013). doi:10.1186/1478-811X-11-43
16. Tarca, A.L., Draghici, S., Khatri, P., Hassan, S.S., Mittal, P., Kim, J.-s., Kim, C.J., Kusanovic, J.P., Romero, R.: A novel signaling pathway impact analysis. Bioinformatics **25**(1), 75–82 (2009). doi:10.1093/bioinformatics/btn577. http://bioinformatics.oxfordjournals.org/content/25/1/75.full.pdf+html
17. Vaske, C.J., Benz, S.C., Sanborn, J.Z., Earl, D., Szeto, C., Zhu, J., Haussler, D., Stuart, J.M.: Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm. Bioinformatics **26**(12), 237–245 (2010)
18. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B **39**(1), 1–38 (1977)
19. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Exploring artificial intelligence in the new millennium, pp. 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003). Chap. Understanding Belief Propagation and Its Generalizations. http://dl.acm.org/citation.cfm?id=779343.779352
20. Hershey, S., Bernstein, J., Bradley, B., Schweitzer, A., Stein, N., Weber, T., Vigoda, B.: Accelerating inference: towards a full language, compiler and hardware stack. CoRR **abs/1212.2991** (2012)
21. Meinshausen, N., Bühlmann, P.: Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **72**(4), 417–473 (2010)
22. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240 (2006). ACM
23. Pollack, J.R., Sørlie, T., Perou, C.M., Rees, C.A., Jeffrey, S.S., Lonning, P.E., Tibshirani, R., Botstein, D., Børresen-Dale, A.-L., Brown, P.O.: Microarray analysis reveals a major direct role of dna copy number alteration in the transcriptional program of human breast tumors. Proceedings of the National Academy of Sciences **99**(20), 12963–12968 (2002). doi:10.1073/pnas.162471999. http://www.pnas.org/content/99/20/12963.full.pdf
24. Salari, K., Tibshirani, R., Pollack, J.R.: Dr-integrator: a new analytic tool for integrating dna copy number and gene expression data. Bioinformatics **26**(3), 414–416 (2010). doi:10.1093/bioinformatics/btp702. http://bioinformatics.oxfordjournals.org/content/26/3/414.full.pdf+html

Figures

Tables

Table 1 LICORN truth table. How the target gene behaves (unless it is deregulated) according to its co-activators' state A and co-inhibitors' state I.

I \ A	-	0	+
-	0	+	+
0	-	0	+
+	-	-	-

Additional Files

Additional_File_1.pdf

File containing PR curves for varying α, μ, the number of genes/samples and the number of belief propagation iterations. It also contains figures illustrating the FDR estimation on simulated data.

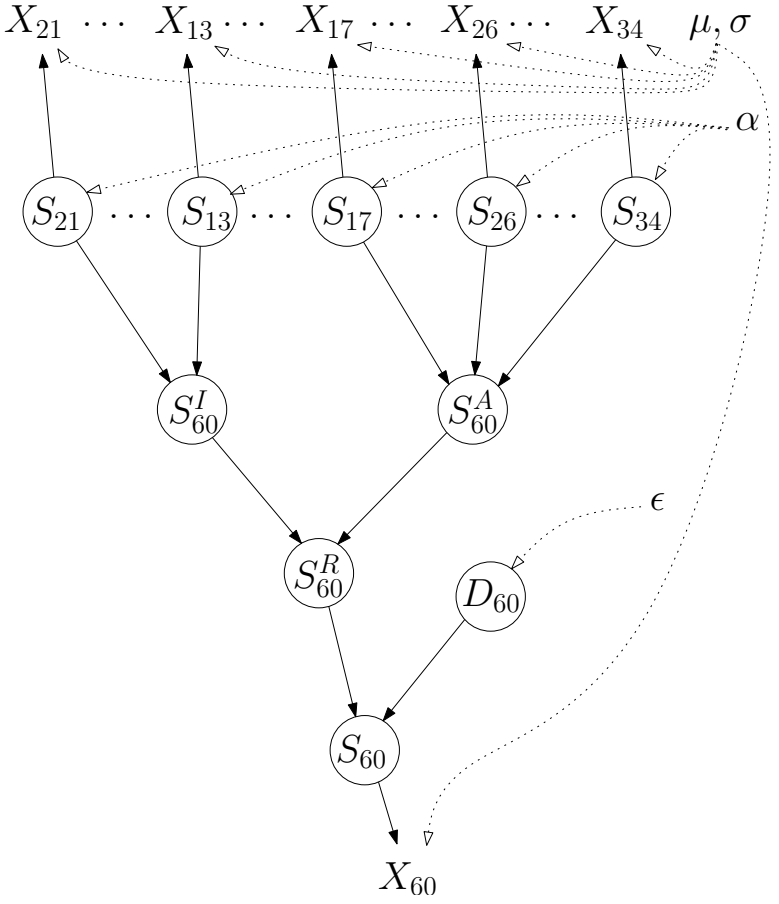

Figure 1 The model for one target gene regulated by two co-inhibitors and three co-activators. The circled variables are hidden. A dashed edge indicates that the distribution of the variable depends on the corresponding parameter.

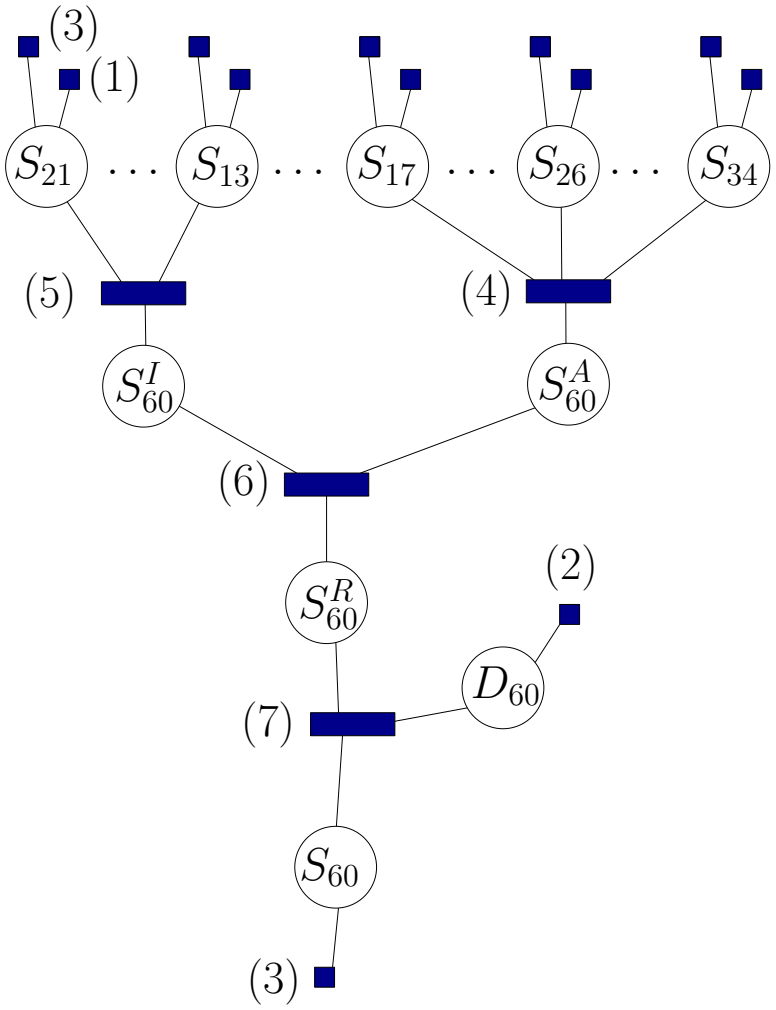

Figure 2 A partial view of the factor graph. The factor graph corresponding to Figure 1. The squares correspond to the factors, and are numbered according to the text. The algorithm oteratively updates the distribution of the circled variables.

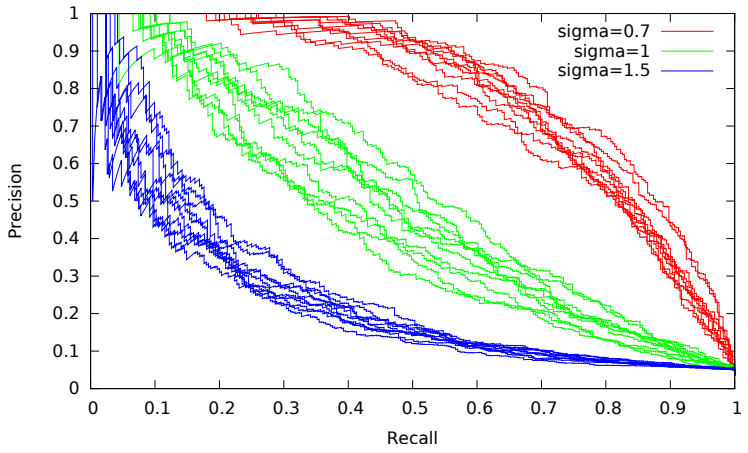

Figure 3 Influence of σ. PR curves for simulations with varying σ, with means $(\mu_-, \mu_0, \mu_+) = (-1, 0, 1)$. Ten simulations are run for each value

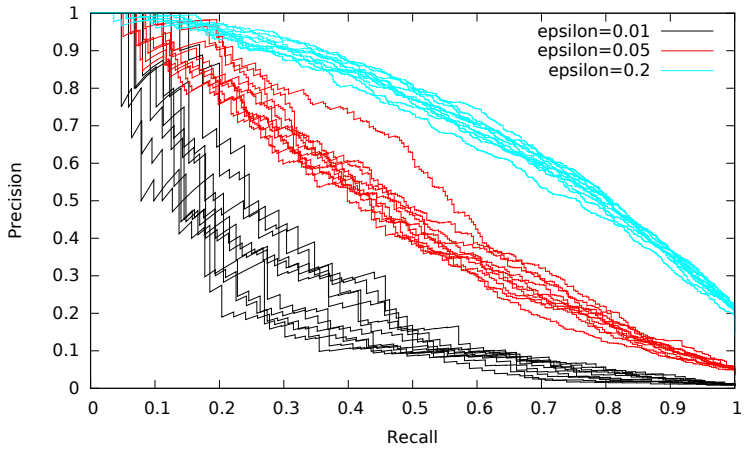

Figure 4 Influence of ϵ. PR curves for simulations with varying ϵ. Ten simulations are run for each value

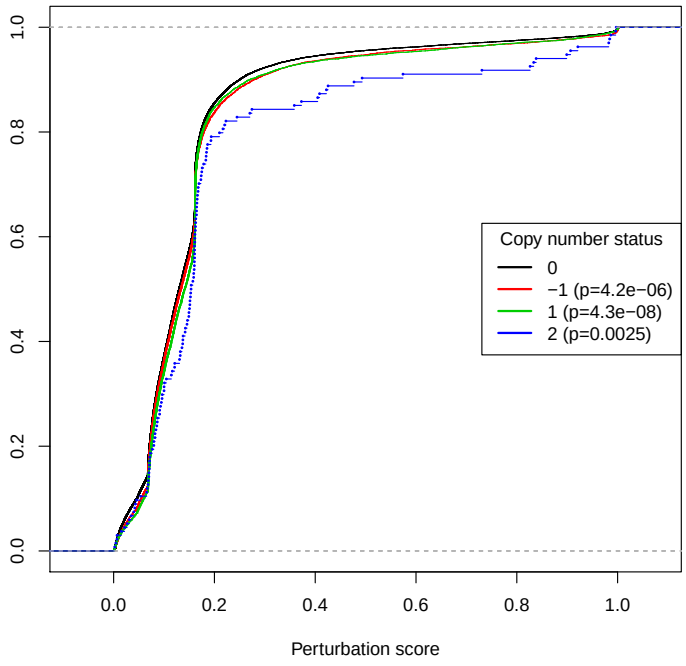

Figure 5 Empirical cumulative distribution of scores, by Copy-Number status. Student's test is used to compare every altered state with the normal.