Inference algorithms for gene networks: a statistical mechanics analysis

A Braunstein\textsuperscript{1,2}, A Pagnani\textsuperscript{1}, M Weigt\textsuperscript{1} and R Zecchina\textsuperscript{2}

\textsuperscript{1} Institute for Scientific Interchange, Viale S Severo 65, I-10133 Torino, Italy
\textsuperscript{2} Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy

E-mail: alfredo.braunstein@polito.it, pagnani@isi.it, weigt@isi.it and riccardo.zecchina@polito.it

Received 4 October 2008
Accepted 10 November 2008
Published 3 December 2008

Abstract. The inference of gene regulatory networks from high throughput gene expression data is one of the major challenges in systems biology. This paper aims at analysing and comparing two different algorithmic approaches. The first approach uses pairwise correlations between regulated and regulating genes; the second one uses message-passing techniques for inferring activating and inhibiting regulatory interactions. The performance of these two algorithms can be analysed theoretically on well-defined test sets, using tools from the statistical physics of disordered systems like the replica method. We find that the second algorithm outperforms the first one since it takes into account collective effects of multiple regulators.

Keywords: cavity and replica method, regulatory networks (theory), message-passing algorithms, genomic and proteomic networks
1. Introduction

According to the central dogma of molecular biology, there is a directed information flux from genes via mRNA to proteins whose amino acid sequence is coded in the genes. This simplified view leads, however, to a paradoxical situation: all cells in a multicellular organism carry the same genetic information, i.e., the same DNA sequence, but they differ greatly in their protein content which is responsible for the difference in functionality of, e.g., neurons, liver cells, skin cells etc. The crucial process which allows for cell differentiation is gene regulation [1], as sketched in figure 1: the transcription of a gene to mRNA is achieved by RNA polymerase (POL) or, in higher organisms, a protein complex containing polymerase. The binding of this polymerase to the DNA in the so-called promoter site depends, however, on the presence or absence of other proteins, the transcription factors, which have binding sites close to the promoter site. These transcription factors are themselves proteins, so they are coded for in genes in other regions of the DNA. We say that there is a gene regulatory interaction from genes 1 and 2 to gene 0. Since the expression of genes 1 and 2 may be regulated by further transcription factors, the set of all such interaction forms a complex gene regulatory network (GRN).
If we knew the gene regulatory network of an organism, it would be interesting to infer the dynamical behaviour and stationary points of these networks to understand the process of cell development and differentiation. In addition, this knowledge would allow us to numerically predict the outcome of genetic interventions, and eventually to identify new drug targets for medical treatment. It is, however, a very complicated, time-consuming and cost-intensive task to experimentally determine the detailed structure of such networks using techniques like gene knockout, chromatin immuno-precipitation (ChIP) etc. In fact, genome-wide networks are only known for simple organisms like \textit{E. coli} \cite{2} and baker’s yeast \cite{3,4}, whereas for higher organisms only a few functional modules are known; see e.g. \cite{5,6}.

On the other hand, today it is relatively easy to monitor the gene expression on a genome-wide scale; the most important experimental tools here are microarrays (DNA chips) which measure the simultaneous abundance of tens of thousands of mRNAs, i.e. of the products of the transcription process, which later on get translated into proteins. It seems therefore an enormously interesting task to infer gene regulatory interactions from expression data using bio-informatics means. Even if such an approach cannot be expected to replace a direct experimental determination of such interactions, it may well serve as a guide for designing efficient and focused experiments.

In the literature, various inference algorithms are discussed. A first class deals with relevance networks, measuring the correlation or mutual information of the expression of gene pairs \cite{7,8}. This approach obviously may include also indirect interactions, since also second or third neighbours in a regulatory network may be correlated. This is taken into account in the algorithm ARACNe \cite{9,10}, which uses information-theoretic arguments to prune all those links which can be explained by indirect interactions. In Bayesian and dynamic Bayesian networks \cite{11,12,13,14}, also the directed structure of causal dependences between gene expression levels is inferred: a global scoring function is assigned to arbitrary networks, and the highest scoring network is considered to be the best possible inferred network. Unfortunately, finding this network is an NP-hard task, it can reasonably accessed only using heuristic tools (greedy algorithms with restarts, simulated annealing and so on) which are likely to get trapped in local extrema of the scoring function. A somewhat similar approach is that of probabilistic Boolean networks \cite{15,16} which...
Inference algorithms for gene networks

exhaustively scan all gene pairs or triplets in order to see to what extent they determine
the expression level of an output gene. Even if it is polynomial, this algorithm is restricted
to relatively few genes and regulatory functions with at most three inputs.

The inference of gene regulatory networks is unfortunately plagued by serious
problems based on the quantity and the quality of available data. Some of the most
relevant limitations are listed below.

(i) The number $M$ of available expression patterns is in general considerably smaller
than the number $N$ of measured genes.

(ii) The available information is, in general, incomplete. The expression for some relevant
genes may not be recorded, and external conditions corresponding, e.g., to nutrient,
mineral, and thermal conditions are not given. Microarrays measure the abundance of
mRNA, whereas gene regulation works via the binding of the corresponding proteins
(transcription factors) to the regulatory regions on the DNA. Protein and RNA
concentrations are, however, not in a simple one-to-one correspondence.

(iii) The data are noisy. This relates to biological noise due to the stochastic nature of
underlying molecular processes, as well as the considerable experimental noise existing
in current high throughput techniques.

(iv) Microarrays do not measure the expression profiles of single cells, but of bunches
of similar cells. This averaging procedure may hide the precise character of the
regulatory processes taking place in the single cells.

(v) Non-transcriptional control mechanisms (chromatin remodelling, small RNAs etc)
cannot be taken into account in expression based algorithms.

(vi) A last problem is a conceptual one: searching for dependences between the expression
levels of genes does not automatically imply their direct physical interaction in
transcriptional regulation. To give an example, strongly co-regulated genes may
appear as good predictors of each other, without having any direct interaction.

The listed points obviously lead to a limited predictability of even the most sophisticated
algorithms. It is therefore very important to test various algorithms on the basis of well-
described data sets containing some or all of the aforementioned problems: only a critical
discussion on artificial data sets allows for a sensible interpretation of the outcomes when
the algorithm is run on biological data.

In this paper, we present a recently introduced statistical physics motivated inference
algorithm built on a message-passing technique which, in a sense, is equivalent to the
Bethe–Peierls approximation of statistical physics [17,18]. We further use statistical
physics tools, more precisely the replica method developed in the theory of disordered
systems and spin glasses, to characterize the performance of this algorithm on artificially
generated data, and to compare it to relevance or ARACNe networks. We will show that
the message-passing technique is able to take into account also collective effects in the
common action of various transcription factors, and therefore it outperforms local pair
correlation based algorithms.

The structure of the paper is as follows. In section 2, we formalize the network
inference problem. In section 3, we review two types of inference algorithms, namely
pair correlation based methods and a message-passing technique. Then we introduce
a generator for artificial data and perform first numerical tests. Section 5 contains a
Inference algorithms for gene networks

Figure 2. Schematic representation of transcriptional repression (left) and activation (right). Repression works via the competitive binding of POL and the TF to overlapping binding sites, whereas activation uses a short-range attraction between POL and TF when attached to nearby binding sites.

... statistical mechanics approach to the performance of both algorithms, and a careful analysis of their behaviour under various (artificial data) conditions. At the end, we conclude this work, and discuss extensions of the message-passing techniques which are necessary for a successful application to biological data.

2. The network inference problem

Let us first formalize the network inference problem. Given are stationary points of the regulatory network, or gene expression profiles. (Note: temporary expression data can be learnt with a modification of the algorithm presented. Here we concentrate fully on static data.) They can be written in the following form:

\[ x^\mu_i, \quad i = 0, \ldots, N, \quad \mu = 1, \ldots, M \] (1)

with \( i \) enumerating the genes considered and \( \mu \) the different expression patterns (profiles). In this work, we assume the expression values to be binarized; \( x^\mu_i = -1 \) signifies that gene \( i \) has very low expression in patterns \( \mu \), whereas a high expression is denoted by \( x^\mu_i = 1 \). Note that we use an Ising spin notation instead of a more standard Boolean one, but this choice will render the technical presentation more accessible. Such a binary representation seems to be oversimplified since gene expression is characterized by mRNA concentrations, but it is applied successfully in various biological examples [5, 6, 16]. It is expected to work fairly well since gene regulatory functions show frequently a switch-like behaviour [19]. The simplest examples, repression and activation by a single TF, are illustrated in figure 2.

At this level, the inference problem can be considered to be factorized over the output variables: we can first reconstruct the regulators for variable 0, then for variable 1 etc until we reach variable \( N \). In our presentation we concentrate therefore on one of these subproblems: without loss of generality, we are interested in the question of to what extent and in what way gene 0 is regulated by the genes \( 1, \ldots, N \). We can distinguish two subsequent subproblems:

- **Topology**: which genes do influence gene 0, and which other genes do not influence gene 0. This results of this question are directed links from the regulating genes to the regulated gene 0.
• **Functionality:** how do these genes influence the expression of gene 0? Here we want to restrict this question to the simplest types of interaction, classifying the input genes 1, . . . , N as activators, repressors or irrelevant for gene 0.

Note that the factorized approach may be oversimplified if experimental noise is considered. Here we are, however, interested not in giving a biologically realistic modelling, but in performing a theoretical analysis of the proposed message-passing algorithm. We therefore keep this assumption since it allows a more elegant and clear presentation of the results.

As mentioned, we aim at classifying the directed interactions from gene i to gene 0 by means of a ternary variable,

\[
J_i = \begin{cases} 
-1 & \text{if gene } i \text{ represses the expression of gene } 0 \\
0 & \text{if gene } i \text{ does not regulate gene } 0 \\
1 & \text{if gene } i \text{ activates the expression of gene } 0.
\end{cases}
\] (2)

Again, this classification is oversimplified with respect to biological reality. First, it does not include the fact that there may be strong and weak activators or inhibitors; all of them are just characterized by one single variable. Second, there may be interactions where a gene changes its activating or inhibiting character in the presence of other genes. To give an example, the XOR function of two variables cannot be represented within the aforementioned classification.

It would be possible to overcome this simplification by introducing more complicated variables. On the other hand, this would increase the number and complexity of the model parameters, and consequently the risk of overfitting. The task of extracting a good predictor from few noisy data requires the model to depend on as few parameters as possible.

To infer this interaction vector, we have also to set up a minimal functional model for the joint interaction of all relevant input genes. Having classified the input genes using this ternary variable, we can sum up the total influence of an expression pattern \(\vec{x} = (x_1, \ldots, x_N)\) on gene 0. If it overcomes some threshold \(\theta\) (which, for clarity of notation, we set to zero in the following), the expression of gene 0 is expected to be enhanced; below this threshold the expression level of gene 0 is repressed. At the Boolean level, we can therefore model the regulatory interaction by means of the threshold function

\[
x_0 = \text{sgn} \left( \sum_{i=1}^{N} J_i x_i \right).
\] (3)

Note that, slightly more than one decade ago, this function attracted considerable attention in statistical physics in the context of the theory of artificial neural networks [20, 21]; it describes a diluted perceptron. In fact, the technical approach presented below consists of an adaptation of the well-known Gardner calculus of perceptron learning [22] to the gene regulatory problem.

Using this model, we are looking for classification vectors \(\vec{J} = (J_1, \ldots, J_N)\) which are as compatible as possible with the input data \(\vec{x}^\mu, \mu = 1, \ldots, M\). We therefore introduce a
Hamiltonian
\[ \mathcal{H}(\vec{J}) = \sum_{\mu=1}^{M} \Theta \left( -x_0^\mu \sum_{i=1}^{N} J_i x_i^m \right) \] (4)

which counts the number of patterns whose output contradicts the model (3). \( \Theta \) symbolizes the step function, which jumps from zero for negative arguments to one for non-negative arguments. One aim with the algorithm will be to minimize this Hamiltonian.

To get reasonable results, we will also implement another biologically motivated constraint. Gene regulatory networks are diluted, i.e. the number of transcription factors for one gene is generally restricted to a small number. In the best known regulatory network, that of baker’s yeast, one finds evidence that the number of genes regulating commonly one gene is distributed according to a narrow exponential distribution, which can be explained with the physical constraints for the positioning of transcription factor binding sites. To take into account this diluted character, we will also control the number
\[ N_{\text{eff}}(\vec{J}) = \sum_{i=1}^{N} (1 - \delta_{J_i,0}) \] (5)
of effectively present, i.e., non-zero, links via a conjugated external field. This \textit{a priori} bias towards diluted graphs helps again to avoid overfitting of the data, since it reduces the entropy of the search space.

This idea is closely related to methods of regression and graphical model selection using \( \ell_1 \)-parameter regularization discussed in the machine-learning literature. The original idea was developed by Tibshirany for the case of sparse regression in linear models [23], but it can be extended to more general convex optimization problems [24]–[27]. In all these cases, the optimization of the cost function is restricted by the \( \ell_1 \)-norm \( \sum_i |J_i| \) of the model parameters \( J_i \). The cusp-like structure of this constraint in \( J_i = 0 \) forces some of the parameters to be exactly zero. In the presence of sufficient data this method can be shown to asymptotically reproduce the correct topological structure of the underlying model [28]. In principle, one could use any \( \ell_n \)-norm \( \sum_i |J_i|^n \) for regularizing parameters. However, only for \( 0 \leq n \leq 1 \) do some of the optimal parameters assume values equal to zero; for \( 1 \leq n \) the constraint is convex, and thus it can be added to the cost function without destroying its convexity. Only the \( \ell_1 \)-norm fulfills both conditions.

Our case is, however, different from these examples. The \textit{discrete} nature of the model parameters renders the model \textit{a priori} NP-hard, and convex optimization cannot be applied. Furthermore, we are interested in ensembles of possible networks rather than a single network as predicted by convex optimization.

3. Inference algorithms

In this section, we present two classes of inference algorithms. The first one is based on pairwise correlations between the output variable \( x_0 \) and inputs \( x_i \). This correlation can be measured in different ways; two of the most important measures are the connected correlation coefficient (or its normalized version, the Pearson correlation coefficient), and the mutual pairwise information of the two variables. Inference algorithms based on
this measure are widespread; in their simplest version they result in the so-called co-expression or relevance network. Such networks normally contain many more links than the actual regulatory interaction between transcription factors and regulated genes; also second neighbours may still be considerably correlated. This is taken into account in a very clever way in the algorithm ARACNE which eliminates all those links which, from an information-theoretic point of view, can be explained via a two-step interaction with an intermediate variable. This elimination step strongly increases the precision of the algorithm since it reduces the number of false positives, but it also decreases its sensitivity, cancelling many weak pair interactions.

The second method is the central one of our paper. It is based on a message-passing algorithm (or more precisely a belief-propagation algorithm) which characterizes the statistical properties of the set of all potential coupling vectors, weighted with respect to Hamiltonian (4) and the number of relevant links (5). We will see that this method is able to—at least partially—recognise the collective effects relating different inputs in the data-generating rule (20). It therefore goes beyond simple two-point correlations, at the cost of being related to a more specific model of gene regulation.

Both algorithms will output a ranking of all potential input variables \(1, \ldots, N\) according to the significance measure considered, going from the most significant genes to the most insignificant ones. The method itself has no intrinsic criterion for where to cut this list. It is, however, possible to use suitable data randomization procedures which give an indication of at which level the significance value of an input may be explained by a random input–output relation (more precisely one may assign a \(p\)-value giving the probability that the significance value or a larger one comes from a random input–output relation).

3.1. Relevance networks

Let us first explain pair correlation based methods, like relevance networks [7, 8]. They are based on measuring the correlation between the output variable \(0\) and each input variable \(i \in \{1, \ldots, N\}\). The central quantities hereby are the joint distribution \(p(x_0, x_i)\) of the two variables, as estimated from the data via

\[
p(x_0, x_i) \simeq \frac{1}{M} \sum_{\mu=1}^{M} \delta(x_0, x^\mu_0) \delta(x_i, x^\mu_i)
\]

and its marginals

\[
p(x_0) = \sum_{x_i} p(x_0, x_i) \simeq \frac{1}{M} \sum_{\mu=1}^{M} \delta(x_0, x^\mu_0)
\]

\[
p(x_i) = \sum_{x_0} p(x_0, x_i) \simeq \frac{1}{M} \sum_{\mu=1}^{M} \delta(x_i, x^\mu_i).
\]

The central question is that of to what extent the two variables are correlated, i.e. to what extent the joint distribution is different from the factorized distribution \(p(x_0)p(x_i)\). This can be measured either via the connected correlation coefficient,

\[
C_{0i} = \sum_{x_0, x_1} [p(x_0, x_i) - p(x_0)p(x_i)] x_0x_i,
\]

doi:10.1088/1742-5468/2008/12/P12001
or via the mutual information,

\[ I_{0i} = \sum_{x_0, x_i} p(x_0, x_i) \log \frac{p(x_0, x_i)}{p(x_0)p(x_i)}. \]  \hspace{1cm} (9)

Note that equation (8) is equivalent to the standard definition of the connected correlation, but the present formulation provides evidence of the distance measure between the joint and the factorized probabilities. The mutual information is also known as the Kullback–Leibler divergence between \( p(x_0, x_i) \) and \( p(x_0)p(x_i) \). It equals zero if and only if both are equal, i.e. if \( x_0 \) and \( x_i \) are statistically independent, and it is positive otherwise.

These numbers (or, in the case of the correlation coefficient, its absolute value) can be ordered and give thus a ranking of the inputs according to their statistical correlation with the output. One can expect the most significant correlations to come from direct interactions, whereas weak correlations are a sign of a non-existent link in the data-generating system (model or experiment).

Note that this method is not able to detect the directed character of gene regulatory interactions, since the very definition in equation (8) is symmetric against exchange of variables 0 and \( i \). This is a major drawback of correlation based methods, and can be eliminated using the algorithm proposed in section 3.2.

3.2. Belief propagation for network inference

In this section we present our message-passing algorithm for network inference. It has two major differences from the correlation based method presented before. On one hand, it is model based (cf equation (3)) and therefore may fail in situations where this model is not sufficiently well adapted. On the other hand, this model based approach allows us to take into account the joint action of all potential regulator variables. Therefore it may well go beyond the simple two-point correlations used in section 3.1.

The basic step is the introduction of a global weight function

\[ W(\vec{J}) = \exp\{-\beta \mathcal{H}(\vec{J}) - h N_{\text{eff}}(\vec{J})\} \]  \hspace{1cm} (10)

weighing all candidate coupling vectors \( \vec{J} \) (i.e., all potential regulatory networks) with respect to their energy \( \mathcal{H}(\vec{J}) \) and the number \( N_{\text{eff}}(\vec{J}) \) of non-zero coupling components; see equations (4) and (5) for the definitions. This weight depends on two not yet specified parameters: the formal inverse temperature \( \beta \) controls the energy. In the limit \( \beta \to \infty \) the weight concentrates in the global minima of \( \mathcal{H} \). The formal external field \( h \) controls, on the other hand, the density of non-zero entries in \( \vec{J} \). Due to the biologically reasonable requirement of diluted \( \vec{J} \), we expect this field to assume high values. There is no obvious strategy for fixing these two parameters, but we will describe a reasonable heuristic strategy.

Note that the data \( \vec{x} \) represent the quenched disorder, whereas the \( N \) components of the vector \( \vec{J} \) are the (ternary) degrees of freedom. The primary algorithmic aim here is to determine the marginal single-site distributions

\[ P_i(J_i) \propto \sum_{\{J_j \in \{0, \pm 1\}; j \neq i\}} W(\vec{J}) \]  \hspace{1cm} (11)

\( \text{doi:10.1088/1742-5468/2008/12/P12001} \)
where the normalization constant in this equation is, as usual, given by the partition function. A direct calculation of these marginals is a computationally intractable task, since equation (11) contains the sum over \( N - 1 \) ternary degrees of freedom, i.e. we have to sum up \( 3^{N-1} \) terms. It is therefore necessary to find some—possibly heuristic—tools which render this task efficiently solvable.

Here we use the idea of message-passing (MP) techniques. These were initially created for problems with sparse constraints, i.e., variables have to be contained in just a few constraints, and constraints have to contain few variables. Typical examples for such problems are random 3-SAT and colourings or vertex covers on finite-connectivity random graphs. Here we are in the opposite situation: each variable \( J_i \) is contained in \( M \) constraints given by the expression patterns \( x_\mu \), \( \mu = 1, \ldots, M \), and each constraint contains all \( N \) variables. Recently the applicability of BP has, however, been verified in a number of cases ranging from information theory to perceptron learning [29]–[31]. In [17,18] we have adopted these techniques to the specific needs of our network inference problem.

The BP equations can be easily written down. Each constraint sends a message to each variable, and each variable sends a message to each constraint:

\[
P_{i \to \mu}(J_i) \propto e^{-h |J_i|} \prod_{\nu \neq \mu} \rho_{\nu \to i}(J_i)
\]

\[
\rho_{\mu \to i}(J_i) \propto \sum_{\{J_j, j \neq i\}} \exp \left\{ -\beta \Theta \left[ -x_0^\mu \sum_k J_k x_k^\mu \right] \right\} \prod_{j \neq i} P_{j \to \mu}(J_j).
\]

The messages have an intuitive interpretation. The variable sends the information to the constraint: 'if you were not present, I would have the statistical properties given by \( P_{i \to \mu}(J_i) \)'. The constraint, on the other hand, sends a message to the variable: 'I have seen the behaviour of the other variables; please behave according to \( \rho_{\mu \to i}(J_i) \) in order to satisfy me'. Note that there are \( 2MN \) of these messages, which form a self-consistently closed set of equations. The true marginal distribution can then be estimated from the messages sent by all constraints,

\[
P_i(J_i) \propto e^{-h |J_i|} \prod_a \rho_{a \to i}(J_i).
\]

From a computational point of view, we still have not gained anything. The second of equations (12) still contains the exponential sum over all but one \( J_j \). This problem can be solved using the following observation: the dependence on the \( J_i \) enters via \( \sum_k J_k x_k^\mu \) and via the factorized distribution \( \prod_{j \neq i} P_{j \to \mu}(J_j) \). For \( N \gg 1 \) we can apply the central limit theorem and replace the sum over the \( N - 1 \) couplings by a single Gaussian integral; for finite \( N \) it will be used as a computationally efficient approximation. We replace the second line of equations (12) by

\[
\rho_{\mu \to i}(J_i) \propto \int_{-\infty}^{\infty} \frac{dh}{\sqrt{2\pi}\Delta_{\mu \to i}} \exp \left\{ -\frac{(h - h_{\mu \to i})^2}{2\Delta^2_{\mu \to i}} - \beta \Theta \left[ -x_0^\mu (J_i x_i^\mu + h) \right] \right\}
\]

\[\text{doi:10.1088/1742-5468/2008/12/P12001}\]
with
\[ h_{\mu \rightarrow i} = \sum_{j \neq i} x_j^\mu \langle J_j \rangle_{j \rightarrow \mu} \]
\[ \Delta^2_{\mu \rightarrow i} = \sum_{j \neq i} x_j^{2\mu} \left( \langle J_j^2 \rangle_{j \rightarrow \mu} - \langle J_j \rangle_{j \rightarrow \mu}^2 \right) \]

where \( \langle \cdot \rangle_{j \rightarrow \mu} \) denotes the average over \( P_{j \rightarrow \mu}(J_j) \). Due to the step-like shape of the Heaviside function, the right-hand site in equation (14) can be expressed via the sum of two error functions, which again leads to a faster numerical implementation.

Note that, in contrast to what happens in most combinatorial optimization problems (and, in particular, also in Bayesian and dynamic Bayesian network learning), we do not have the aim of constructing one single instance of a ‘good’ coupling vector \( \vec{J} \), because this vector may be quite different from the original data-generating vector. We are instead interested in characterizing the ensemble of all sets of vectors. More precisely, the marginal probability \( P_i(J_i) \) gives us a confidence measure for the hypothesis that the output variable \( x_0 \) actually depends on \( x_i \): it measures the fraction of all suitable coupling vectors \( \vec{J} \) (as weighted by equation (10)) where the entry \( i \) takes the value \( J_i \). We can therefore base a global ranking of all potential couplings on the probability \( (1 - P_i(J_i = 0)) \) of having a non-zero coupling (or alternatively on the average coupling \( \langle J_i \rangle \)).

Having calculated the messages and the marginal distributions, we may determine the energy of the average coupling vector
\[ E = \sum_{\mu=1}^{M} \Theta \left( -x_0^\mu \sum_{i=1}^{N} \langle J_i \rangle_i x_i^m \right) \]
and the Bethe entropy
\[ S = \sum_{\mu=1}^{M} S_\mu - (N - 1) \sum_{i=1}^{N} S_i \]
characterizing the number of ‘good’ coupling vectors. In the last expression, the site entropy \( S_i \) is given by
\[ S_i = - \sum_{J_i=-1,0,1} P_i(J_i) \ln P_i(J_i), \]
and the pattern entropy
\[ S_\mu = - \sum_{\vec{J}} P_\mu(\vec{J}) \ln P_\mu(\vec{J}) \]
\[ P_\mu(\vec{J}) = \exp \left\{ -\beta \Theta \left[ -x_0^\mu \sum_k J_k x_k^\mu \right] \right\} \prod_i P_{i-\mu}(J_i) \]
can be calculated in analogy to \( \rho_{\mu \rightarrow i} \) via a Gaussian approximation of the sum over \( \vec{J} \).

So far, we have not fixed the parameters \( \beta \) and \( h \) controlling the energy and the dilution of the inferred vectors. A simple heuristic strategy is the following: fix the number \( N_{\text{eff}}(\vec{J}) \) externally to some desired value, and start at high temperature and
initial $h$ given by $N_{\text{eff}} = 2 \cosh h/(1 + 2 \cosh h)$. Update the messages starting from some random initialization, and calculate the energy of the average coupling vector and the Bethe entropy. Now decrease the temperature gradually towards the zero entropy (or zero energy), adapting $h$ such that $N_{\text{eff}}(\bar{J})$ remains close to the desired value. This strategy can be repeated for various values of $N_{\text{eff}}$. In practical applications, an optimal $N_{\text{eff}}$ can also be determined by first dividing the data set into a training set and a test set, calculating the marginal probabilities on the training set, and minimizing the corresponding prediction error of the average coupling vector on the test set. In the case of artificial data coming from a known coupling vector, we can train on all data and compare directly with the correct vector.

4. Tests on artificial data

Before coming to a theoretical analysis of the algorithmic performance, we make a numerical test to get a first impression of the predictive power of the two algorithms.

4.1. The data generator

In order to check and compare the performance of the algorithms presented below, we have to introduce a data generator. It allows us to create data sets with well-controlled properties, and the knowledge about the data-generating rule allows us to compare the results obtained with the inference algorithm to those obtained with the true rule.

In this work, we concentrate on data generated using a simple threshold function with couplings $J_i^0$,

$$x_0^\mu = \text{sgn} \left( \sum_{i=1}^{N} J_i^0 x_i^\mu + \eta^\mu \right)$$  \hspace{1cm} (20)

for all $\mu = 1, \ldots, M$. The inputs $\vec{x}^\mu = (x_1^\mu, \ldots, x_N^\mu)$ have independently distributed binary entries. As compared to the ternary classification into activators, repressors and non-interacting genes, this rule has two major differences.

- **Heterogeneity:** the couplings $J_i$ may take values which are different from 0, ±1. This allows us to introduce regulatory variables of different interaction strengths, i.e. in particular weak and strong activators or repressors. In this work, we will study in particular the case where $J_i \in \{0, \pm 1, \pm 2\}$. The couplings themselves will be drawn randomly and independently with respect to a distribution

$$\rho(J_i^0) = (1 - k_1 - k_2)\delta(J_i^0) + \frac{k_1}{2} \delta(J_i^0 + 1) + \delta(J_i^0 - 1) + \frac{k_2}{2} \delta(J_i^0 + 2) + \delta(J_i^0 - 2).$$  \hspace{1cm} (21)

Generalizations will be straightforward, but they do not alter at all the conclusions of this work. Note that real gene regulatory networks are sparse, i.e. we have to work in the regime $k_{1,2} \ll 1$. In the statistical mechanics calculations presented below, we will consider the thermodynamic limit $N \to \infty$. At a first glance it seems to be reasonable to assume the usual scaling of finite-connectivity graphs, $k_{1,2} = \mathcal{O}(N^{-1})$, for the coupling probabilities. For technical reasons we will, however, concentrate...
on the scaling $k_{1,2} = \mathcal{O}(1) \ll 1$. This choice will be justified by comparison with numerical data.

- **Noise:** in biological data, there are various kinds of noise. The first one is biological noise resulting from the stochastic nature of cellular processes. The second and more annoying one is experimental noise which is strongly present in most high throughput data. In addition, biological data are normally incomplete in the sense that not all existing gene expression levels are measured, that nutrient or mineral concentrations, temperature and other external factors are not recorded etc. In our data generator, we have included a simple additive noise term $\eta^\mu$ which we assume to be independent for various patterns $\mu$. More precisely we assume it to be Gaussian with

$$
\eta^\mu = 0 \quad \eta^\mu \eta^\nu = \gamma N \delta_{\mu,\nu}
$$

(22)

for all $\mu, \nu \in \{1, \ldots, M\}$. Note that the scaling of the variance with $\sqrt{N}$ makes the noise of the same order of magnitude as the signal $\sum J^0_{i,\mu} x_i$. For the special case $\gamma = k_1 + 4k_2$, the noise and signal have exactly the same statistical properties.

Note that this data generation rule is more complex than the inference rule (3). This will result in non-feasible data, i.e., a non-zero ‘energy’ $\mathcal{H}(\vec{J})$ even for the best inferred couplings $\vec{J}$. The best possible result would be $J_i = \text{sgn} J^0_i$ (here with $\text{sgn}(0) = 0$) since this would correctly assign the attributes activating, repressing, irrelevant to the couplings. Due to the presence of noise as well as due to the finite amount of data available, such a perfect prediction will be impossible. We therefore introduce the following notation:

$$
\begin{align*}
J^0_i = 0 & \quad J_i = 0 & \text{true negative (TN)} \\
J^0_i \neq 0 & \quad J_i = 0 & \text{false negative (FN)} \\
J^0_i = 0 & \quad J_i \neq 0 & \text{false positive (FP)} \\
J^0_i \neq 0 & \quad J_i \neq 0 & \text{true positive (TP)}
\end{align*}
$$

(23)

It can be refined taking into account also the relative sign of the original and the inferred couplings; cf the above discussion about topology and functionality. One of the aims is to predict a fraction of all couplings with high precision, i.e. to have an as high as possible number of TP with a low number of FP. The quality measure that we use will be the confrontation of the recall, or sensitivity,

$$
\text{RC} = \frac{N_{TP}}{N_{TP} + N_{FN}}
$$

(24)

and of the precision, or specificity,

$$
\text{PR} = \frac{N_{TP}}{N_{TP} + N_{FP}}.
$$

(25)

The recall describes the fraction of all existing non-zero couplings which are predicted by the algorithm, whereas the precision tells us what fraction of all predicted links are actually present in the data generator.
4.1.1. Feasible data. As a first test, we have run BP on data sets with $k_2 = \gamma = 0$. In this case, also the couplings of the data generator function assume only the three values $J_i^0 \in \{0, \pm 1\}$, and noise is absent. This means that the data set is feasible; the model (3) on which we base our inference is able to reproduce the data without errors. We therefore work automatically at zero formal temperature, and thus at zero energy.

The difficulty of the test problem comes from the fact that we are using an extremely sparse coupling vector: only 3 out of $N = 600$ input variables are coupled to the output via a non-zero coupling $J_i^0$. We also present relatively small data sets with $M \leq 70$. Is BP able to recognize the three important input variables?

First, we have done experiments without the exterior diluting field $h$. The results are given in figure 3, where the quality of the prediction is measured via the overlap

$$q = \frac{1}{|J_0^0|^2} \sum_{i=1}^{N} \sum_{J_i} J_i J_i^0 P_i(J_i).$$

This overlap takes for all considered numbers $M$ of presented patterns only very small values; the algorithm is not able to recognize the relevant inputs. The situation changes drastically if we add the diluting field $h$. We find, at some $h$-dependent point, a discontinuous jump from a very bad BP solution with small overlap to a perfectly polarized solution with $q = 1$. The algorithm has perfectly recognized the relevant three inputs, and thus perfectly reconstructed the data-generating rule! The number of patterns needed decreases with $h$, and can be as small as $M = 35$ for large fields.

For even larger fields, the algorithm runs into problems: the dilution becomes as important for the algorithm as the energy constraints, and the BP solution polarizes completely to the vector $J_i \equiv 0$. Best performance is obtained for fields just below this point.
4.1.2. Unfeasible data. The success of BP for feasible data sets encourages us to go ahead to more involved cases, i.e., cases with heterogeneities in the coupling strength ($k_1, k_2 \neq 0$) and possibly with noise ($\gamma \neq 0$). Here we give a first example of a data generator with $N = 600$ inputs, but only 30 of them have a non-zero coupling to the output—15 of them with an absolute value equal to 2, and 15 with absolute value 1 ($k_1 = k_2 = 0.025$). We generate $M = 300$ patterns and apply both inference algorithms. The resulting rankings can be cut at arbitrary points, such that we can measure a recall versus precision curve as given in figure 4.

We find that both algorithms start at precision 1 for the highest ranking inputs. The algorithm based on mutual information includes, however, the first false positive after 8 true positives, whereas BP is able to find the first 14 true positives before including the first erroneous link. Also later on the performance of BP remains well above that of the MI. Note that most of the initially recognized links are strong activators or repressors; only one of the first 14 true positives in BP is an activator with $J^0_i = 1$. Stronger signals are obviously detected first.

Although they are not perfect, it is to be noted that both algorithms perform much better than a random ranking which would fluctuate around an average precision of only $PR = k_1 + k_2 = 0.05$, as marked by the blue line in the figure.

5. A statistical analysis of the algorithmic performance

Can we explain this behaviour analytically? Can we understand the discontinuous transition from very bad to perfect inference in the case of a feasible data set? Can we understand the relative behaviour of BP compared to the different pair correlation based methods?

doi:10.1088/1742-5468/2008/12/P12001
In the following we will first calculate exactly the precision versus recall curve for the correlation coefficient and the mutual information, and then we go to an analysis of the BP performance via a statistical mechanics analysis of the Gibbs weight given in equation (10). In both cases, we are mainly interested in the thermodynamic limit \( N \to \infty \). In this limit, we also consider \( M \) to diverge, with \( \alpha = M/N \) being asymptotically constant. The probabilities \( k_1 \) and \( k_2 \) for non-trivial couplings in the data generator are considered to be finite, i.e. non-trivial couplings constitute a finite fraction of all entries of \( \vec{J}^0 \).

5.1. Pair correlation and mutual information

Let us recall the estimate of the joint distribution of the values of variable 0 and variable \( i \) as calculated from the data. Using equation (6), and plugging in equation (20) for the value of the outputs \( x^0_0 \), we get

\[
p(x_0, x_i) = \frac{1}{M} \sum_{\mu=1}^{M} \delta(x_0, \text{sgn}[\vec{J}^0 \cdot \vec{x}^\mu + \eta^\mu]) \delta(x_i, x^\mu_i).
\]

Due to the random nature of the input patterns, this quantity itself is a random variable with some distribution \( P(p(x_0, x_i)) \). Since the different patterns are statistically independent, the joint probability results as a sum over \( M = \alpha N \) independently and identically distributed variables. According to the central limit theorem, \( P(p(x_0, x_i)) \) is thus Gaussian. Its mean can be calculated as

\[
\overline{p(x_0, x_i)} = \frac{1}{2N} \sum_{x_1, \ldots, x_N} \delta(x_0, \text{sgn}[\vec{J}^0 \cdot \vec{x}^1 + \eta^1]) \delta(x_i, x^1_i)
= \frac{1}{2N} \sum_{\{x_j : j \neq i\}} \delta \left( x_0, \text{sgn} \left[ J^0_{x_i} + \sum_{j \neq i} J^0_{x_j} x^1_j + \eta^1 \right] \right)
\]

where we have used that the input variables are independent and unbiased, and that the mean of all \( M \) contributions to the sum is identical. For calculating this sum, we use again the central limit theorem; the sum \( y = \sum_{j \neq i} J^0_{x_j} x^1_j \) is Gaussian distributed with mean zero and second moment

\[
\overline{y^2} = (k_1 + 4k_2 + \gamma) N + \mathcal{O}(1) =: \Delta_y^2.
\]

We thus get

\[
\overline{p(x_0, x_i)} = \frac{1}{2} \int_{-\infty}^{\infty} \frac{dy}{\sqrt{2\pi} \Delta_y} e^{-y^2/(2\Delta_y^2)} \delta \left( x_0, \text{sgn}[J^0_{x_i} x^1_i + y] \right)
= \frac{1}{2} H \left( -\frac{J^0_{x_0 x_i}}{\sqrt{(k_1 + 4k_2 + \gamma) N}} \right)
\]

with \( H(x) = \int_{-\infty}^{\infty} \frac{dy}{\sqrt{2\pi}} e^{-y^2/2} \). We expand this function around zero, and find

\[
\overline{p(x_0, x_i)} = \frac{1}{4} + \frac{J^0_{x_0 x_i}}{2\sqrt{2\pi(k_1 + 4k_2 + \gamma) N}} + \mathcal{O}(N^{-1}).
\]
The calculation of the variance is now straightforward. We have
\[ p(x_0, x_i)^2 = \frac{M - 1}{M} p(x_0, x_i)^2 + \frac{1}{M} p(x_0, x_i), \] (32)
and consequently we find
\[ [\Delta p(x_0, x_i)]^2 = \frac{1}{M} p(x_0, x_i)(1 - p(x_0, x_i)) = \frac{3}{16\alpha N} + \mathcal{O}(N^{-3/2}). \] (33)
Finally we need the correlation for different value pairs \((x_0, x_i) \neq (x'_0, x'_i)\). Following the same strategy as before, we find
\[ \Delta p(x_0, x_i) \Delta p(x'_0, x'_i) = -\frac{1}{16\alpha N} + \mathcal{O}(N^{-3/2}). \] (34)
An aside is necessary here. The covariance matrix of all four values of \(p(x_0, x_i)\) is degenerate, with a zero eigenvector \((1, 1, 1, 1)\). It is, however, clear that all four quantities cannot have a simple joint Gaussian distribution—given three, the fourth is determined by normalization, \(\sum_{x_0, x_i} p(x_0, x_i) = 1\). The zero eigenvector thus corresponds to the forbidden normalization breaking fluctuations.

5.1.1. Pair correlations. Let us, however, continue with the pair correlations between the output variable \(x_0\) and a potential regulator \(x_i\). We write
\[ C_{0i} = \sum_{x_0, x_i} p(x_0, x_i) \, x_0 x_i \]
\[ = p(1, 1) + p(-1, -1) - p(1, -1) - p(-1, 1) \]
\[ = 2p(1, 1) + 2p(-1, -1) - 1 \] (35)
where, in the last line, we have exploited the normalization of \(p(x_0, x_1)\). Due to imperfect sampling in a finite data set, this is again a Gaussian random variable. Using equation (31) we find its mean to be
\[ \bar{C}_{0i} = \frac{2 J^0}{\sqrt{2\pi(k_1 + 4k_2 + \gamma)}} + \mathcal{O}(N^{-1}), \] (36)
whereas its variance can be calculated directly from equations (33) and (34),
\[ (\Delta C_{0i})^2 = \frac{1}{\alpha N} + \mathcal{O}(N^{-3/2}). \] (37)
Both mean and variance scale as \(\mathcal{O}(N^{-1/2})\); introducing thus \(\tilde{C}_{0i} = \sqrt{N}C_{0i}\) we find its probability distribution
\[ P(\tilde{C}_{0i} | J^0) = \sqrt{\frac{2\pi}{\alpha}} \exp \left\{ -\frac{\alpha}{2} \left( \tilde{C}_{0i} - \frac{2 J^0}{\sqrt{2\pi(k_1 + 4k_2 + \gamma)}} \right)^2 \right\} \] (38)
conditioned to the value of the coupling \(J^0\) between variables 0 and \(i\) in the data generator (20). At this point, one recognizes the potential and also the limitations of network inference via pair correlations: the mean value of the correlations of different
variables is well separated by a term proportional to the coupling strength, and in particular strong activators and inhibitors should be distinguishable from irrelevant variables. On the other hand, the variance of distribution (38) is of the same order of magnitude as the mean, i.e. the distributions for different values $J^0_i$ overlap, and errors in the network inference are unavoidable. The only way to suppress errors is to augment the amount of data; the variance of the measured correlation values around its expectation decays as $1/\sqrt{\alpha}$.

The measured pair correlations, or more precisely their absolute values, serve as a ranking for the inclusion of different variables into the reconstructed network. Let us therefore imagine that we introduce a cut-off $\theta > 0$ for the absolute value of the rescaled correlations $C^0_i$, i.e., for all $i$ with $|C^0_i| > \theta$ we introduce a coupling; for all inputs with a smaller absolute correlation value we assume the coupling to be absent. The fraction of included couplings with given original $J^0_i$ is then found to be

$$P(|\tilde{C}^0_i| > \theta \mid J^0_i) = \int_{-\infty}^{-\theta} d\tilde{C}^0_i P(\tilde{C}^0_i \mid J^0_i) + \int_{\theta}^{\infty} d\tilde{C}^0_i P(\tilde{C}^0_i \mid J^0_i)$$

$$= H\left(\sqrt{\alpha} \left[ \theta - \frac{2J^0_i}{\sqrt{2\pi(k_1 + 4k_2 + \gamma)}} \right] \right)$$

$$+ H\left(\sqrt{\alpha} \left[ \theta + \frac{2J^0_i}{\sqrt{2\pi(k_1 + 4k_2 + \gamma)}} \right] \right).$$

(39)

For $J^0_i \neq 0$, the corresponding included couplings are true positives. For $J^0_i = 0$, they are false positives. We thus can write out the recall and the precision of the inferred network, depending on the cut-off parameter $\theta$:

$$RC(\theta) = \frac{k_1 P(|\tilde{C}^0_i| > \theta \mid \pm 1) + k_2 P(|\tilde{C}^0_i| > \theta \mid \pm 2)}{k_1 + k_2}$$

$$PR(\theta) = \frac{k_1 P(|\tilde{C}^0_i| > \theta \mid \pm 1) + k_2 P(|\tilde{C}^0_i| > \theta \mid \pm 2)}{(1 - k_1 - k_2)P(|\tilde{C}^0_i| > \theta \mid 0) + k_1 P(|\tilde{C}^0_i| > \theta \mid \pm 1) + k_2 P(|\tilde{C}^0_i| > \theta \mid \pm 2)}.$$  \hspace{1cm} (40)

Changing the value of $\theta$ from high to small values, we increase the recall from zero to one, at the cost of a decreasing precision. More details will be given below, in the comparison to the results from the message-passing based inference algorithm.

Note that we did not require the correct determination of the sign of the coupling, a generalization to this case is, however, straightforward. For not too small $\alpha$, it practically does not lead to recognizable differences, i.e. the waste majority all true positives is recognized due to the correct sign of the measured correlation.

5.1.2. Mutual information. To what extent does the network reconstruction change if we go from correlations to the pairwise mutual information as given in equation (9)? The latter has a non-linear dependence on the pair correlations $p(x_0, x_1)$, and consequently it does not have a Gaussian distribution. We may, however, use the fact that variables are only weakly correlated, and write

$$p(x_0, x_1) = \frac{1}{4} + \frac{z(x_0, x_1)}{\sqrt{N}}.$$  \hspace{1cm} (41)

\textbf{doi:10.1088/1742-5468/2008/12/P12001}
The newly introduced function \( z(x_0, x_1) \) contains all non-trivial correlations between \( x_0 \) and \( x_1 \), and it is of \( \mathcal{O}(N^0) \). Note that at this point we consider equation (41) to be an exact definition of \( z(x_0, x_i) \), i.e. the latter contains all the lower order corrections in \( N \) and not only the dominant term. Due to the normalization of \( p \) it fulfills \( \sum_{x_0, x_i} z(x_0, x_i) = 0 \). We subsequently write for the two marginal single-variable distributions of \( p(x_0, x_i) \)

\[
p(x_0) = \frac{1}{2} + \frac{z(x_0)}{\sqrt{N}} \quad p(x_i) = \frac{1}{2} + \frac{z(x_i)}{\sqrt{N}}. \tag{42} \]

Plugging this into the definition of the mutual information, we find

\[
I_{0i} = \sum_{x_0, x_i} \left[ \frac{1}{4} + \frac{z(x_0, x_i)}{\sqrt{N}} \right] \left[ \log \left( 1 + \frac{4 z(x_0, x_i)}{\sqrt{N}} \right) - \log \left( 1 + \frac{2 z(x_0)}{\sqrt{N}} \right) \right] - \log \left( 1 + \frac{2 z(x_i)}{\sqrt{N}} \right). \tag{43} \]

Expanding the logarithm up to second order in the \( z \), and using the normalization to eliminate all linear terms, we find after some straightforward algebra

\[
I_{0i} = \frac{2}{N} [z(1, 1) + z(-1, -1)]^2 + \mathcal{O} (N^{-(3/2)})
= \frac{C_{01}^2}{2} + \mathcal{O} (N^{-(3/2)}). \tag{44} \]

We thus conclude that, at least under the scaling limits considered in this work, the correlation coefficient and the mutual information carry the same information. Network reconstruction algorithms based on either of these two quantities thus lead to the same results.

### 5.2. Characterizing the set of coupling vectors

In section 5.1.2, we have analytically characterized the performance of a network inference algorithm based on pairwise correlation measures. Is a similar comparison possible for the belief-propagation algorithm introduced in section 3.2? As already mentioned, the algorithm is based on the measure

\[
P(J) = \frac{1}{Z(\beta, h)} \exp\{ -\beta \mathcal{H}(J) - h N_{\text{eff}}(J) \} \tag{45} \]

which assigns a weight to every coupling vector \( J \). This weight depends on its performance on the data set \( \{ \vec{x}^\mu \} \) as measured by the Hamiltonian \( \mathcal{H}(J) \) defined in equation (4), and on the effective number \( N_{\text{eff}}(J) \) of relevant inputs \( J_i \neq 0 \) as defined in equation (5). A positive inverse temperature \( \beta \) introduces a bias toward \( J \) with few errors on the data set; a positive \( h \) favours diluted coupling vectors. The aim of using the algorithm was to determine the marginal distributions \( P_i(J_i) \) for single entries of the vector \( J \), and use the probability \( P_i(J_i \neq 0) \) that such a component becomes non-zero for ranking the input variables according to their relevance for the output.
Note that in this way we do not work with a single reconstructed network, which may or may not be similar to the data generator, but with a probability measure over all feasible coupling vectors.

To characterize the performance of this ranking procedure we have to calculate first the partition function

\[
Z(\beta, h) = \sum_{\vec{J} \in \{0, \pm 1\}^N} \exp\{-\beta \mathcal{H}(\vec{J}) - h N_{\text{eff}}(\vec{J})\}. \quad (46)
\]

From this we can calculate the average energy

\[
\langle \mathcal{H} \rangle = \varepsilon N = -\frac{\partial}{\partial \beta} \log Z(\beta, h) \quad (47)
\]

and the average number of non-zero couplings

\[
\langle N_{\text{eff}} \rangle = n_{\text{eff}} N = -\frac{\partial}{\partial h} \log Z(\beta, h). \quad (48)
\]

The entropy characterizing the number of statistically dominant coupling vectors is now given by

\[
S = s N = \log(Z) + \beta \langle \mathcal{H} \rangle + h \langle N_{\text{eff}} \rangle. \quad (49)
\]

Besides these standard thermodynamic quantities, we also need the distributions \( P(J \neq 0 | \vec{J}^0) \), i.e., the probability that a single coupling component is different from zero, given the corresponding coupling in the data generator. This distribution will allow us to identify true and false positives following a strategy similar to that of section 4. The calculation necessary to obtain these thermodynamic quantities is a modified Gardner calculation as introduced in the theory of formal neuronal networks. This calculation follows a path beautifully reviewed in the textbook by Engel and van den Broeck [21]; we therefore skip all technical details and present only the major steps.

A first observation is that the free energy \(-\beta^{-1} \log Z(\beta, h)\) depends explicitly on the realization of the input data, i.e. on \( \vec{x}^\mu \), the noise \( \eta^\mu \) and the coupling vector \( \vec{J}^0 \) of the data generator. It is consequently a random variable which is, however, expected to be self-averaging: in the thermodynamic limit, its distribution is expected to become more and more concentrated around its average, and relative fluctuations are expected to disappear. We can therefore replace the actual free energy by its mean value \(-\beta^{-1} \log Z(\beta, h)\), where the overline denotes the joint average over \( \vec{x}^\mu \), \( \eta^\mu \) and \( \vec{J}^0 \). Technically this can be achieved using the replica trick: we write the logarithm as

\[
\log Z = \lim_{n \to 0} \frac{Z^n - 1}{n}, \quad (50)
\]
calculate the right-hand side first for positive integer \( n \), introduce an analytical continuation to real \( n \) and perform the limit. The averaging over \( Z^n \) for integer \( n \) can be performed easily since the \( n \)th power can be interpreted as an \( n \)-fold identical replication of the original system.
We thus have to calculate the following:

\[
\mathcal{Z} = 2^{-M N} \sum_{x_i^n} \prod_{\mu=1}^{N} \left[ \exp \left\{ \int \rho(J_0^\mu) e^{-\beta \sum_{i=1}^{N} J_0^\mu x_i^\mu} \right\} \right] \times \exp \left\{ -\beta \sum_{a=1}^{a} \sum_{\mu=1}^{M} \Theta \left( -\left[ \sum_{i=1}^{N} J_0^\mu x_i^\mu + \eta^a \right] \left[ \sum_{i=1}^{N} J_0^\mu x_i^\mu \right] \right) \right\}.
\]

(51)

Here, \( a = 1, \ldots, n \) enumerates the \( n \) replicas of the system, whereas \( a = 0 \) stands for the data generator. Following the path explained in [21], we can extract the interesting quantities under the assumption of replica symmetry:

\[
\varepsilon = 2\alpha \int D\hat{y} H \left[ y \sqrt{\frac{r^2}{k^0 q - r^2}} \right] \frac{e^{-\beta H[ -y \sqrt{q/(n_{\text{eff}} - q)]} - (e^{-\beta} - 1) H[ -y \sqrt{q/(n_{\text{eff}} - q)]}}{1 + (e^{-\beta} - 1) H[ -y \sqrt{q/(n_{\text{eff}} - q)]}}
\]

\[
s = -n_{\text{eff}} \hat{k} - r \hat{r} + \frac{1}{2} q \hat{q} + \int D\hat{x} \sum_{\hat{J}^0} \rho(\hat{J}^0) \log[1 + 2 e^{\hat{k} + \hat{q}/2} \cosh\{ \hat{r} \hat{J}^0 + x \sqrt{q]\}] + 2\alpha \int D\hat{x} H \left[ y \sqrt{\frac{r^2}{k^0 q - r^2}} \right] \log \left( 1 + (e^{-\beta} - 1) H[ -y \sqrt{q/(n_{\text{eff}} - q)]} \right)
\]

\[
+ \beta \frac{e^{-\beta H[ -y \sqrt{q/(n_{\text{eff}} - q)]}}}{1 + (e^{-\beta} - 1) H[ -y \sqrt{q/(n_{\text{eff}} - q)]}}
\]

(52)

with the Gaussian normal distribution \( D\hat{y} = dy/\sqrt{2\pi} e^{-y^2/2} \), \( k^0 = k_1 + 4k_2 + \gamma \) and the order parameters being determined via the self-consistent saddle-point equations

\[
n_{\text{eff}} = \frac{1}{N} \langle \hat{q}_a \cdot \hat{q} \rangle = \int D\hat{x} \sum_{\hat{J}^0} \rho(\hat{J}^0) \frac{2 e^{\hat{k} + \hat{q}/2} \cosh\{ \hat{r} \hat{J}^0 + x \sqrt{q}\}}{1 + 2 e^{\hat{k} + \hat{q}/2} \cosh\{ \hat{r} \hat{J}^0 + x \sqrt{q}\}}
\]

\[
q = \frac{1}{N} \langle \hat{q}_a \cdot \hat{q} \rangle = \int D\hat{x} \sum_{\hat{J}^0} \rho(\hat{J}^0) \left[ \frac{2 e^{\hat{k} + \hat{q}/2} \sinh\{ \hat{r} \hat{J}^0 + x \sqrt{q}\}}{1 + 2 e^{\hat{k} + \hat{q}/2} \cosh\{ \hat{r} \hat{J}^0 + x \sqrt{q}\}} \right]^2
\]

\[
r = \frac{1}{N} \langle \hat{J}_a \cdot \hat{J} \rangle = \int D\hat{x} \sum_{\hat{J}^0} \rho(\hat{J}^0) \hat{J}_a \hat{J} \frac{2 e^{\hat{k} + \hat{q}/2} \sinh\{ \hat{r} \hat{J}^0 + x \sqrt{q}\}}{1 + 2 e^{\hat{k} + \hat{q}/2} \cosh\{ \hat{r} \hat{J}^0 + x \sqrt{q}\}}
\]

\[
\hat{q} = -\frac{\alpha k^0 r}{\pi (k^0 q - r)^{3/2}} \int dy \exp \left\{ -\frac{k^0 q y^2}{2(k^0 q - r^2)} \right\} \times \log \left( 1 + (e^{-\beta} - 1) H[ -y \sqrt{q/(n_{\text{eff}} - q)]} \right)
\]

\[
\times \left[ \frac{e^{-\beta H[ -y \sqrt{q/(n_{\text{eff}} - q)]}}}{1 + (e^{-\beta} - 1) H[ -y \sqrt{q/(n_{\text{eff}} - q)]}} \right]
\]

\[
\hat{r} = -\frac{\alpha k^0 r}{\pi (k^0 q - r)^{3/2}} \int dy \exp \left\{ -\frac{k^0 q y^2}{2(k^0 q - r^2)} \right\} \log \left( 1 + (e^{-\beta} - 1) H[ -y \sqrt{q/(n_{\text{eff}} - q)]} \right).
\]

(53)

The parameter \( n_{\text{eff}} \) measures the self-overlap of a typical coupling vector, and thus the desired fraction of non-zero components, \( q \) measures the overlap between two different
coupling vectors, \( r \) the overlap between a typical coupling vector and the coupling of the original data-generating vector \( \vec{J}^0 \), and their conjugate parameters. All of these parameters still depend on \( \beta \) and \( h \). As explained in the algorithmic section, we fix them by requiring \( n_{\text{eff}} \) to be in some small given interval, and by increasing the temperature such that either the energy or the entropy goes to zero, focusing in this way to the ground states at given \( n_{\text{eff}} \).

At this point, we are also able to express the probability \( P(J \neq 0|\vec{J}^0) \) that a link is present in an inferred coupling vector, depending on its value in the data generator. We find that it takes the value

\[
P(J \neq 0|\vec{J}^0) = \frac{2e^{\hat{\phi}/2} \cosh\{\hat{r}\vec{J}^0 + x\sqrt{q}\}}{1 + 2e^{\hat{\phi}/2} \cosh\{\hat{r}\vec{J}^0 + x\sqrt{q}\}}
\]

with \( x \) being a random variable drawn from a Gaussian normal distribution. This random number represents the heterogeneity of the different inputs having the same value of the coupling in the data generator, which persists due to the imperfect sampling at finite \( \alpha \). Accepting all those inputs having a value of \( P(J \neq 0|\vec{J}^0) \) larger than an arbitrary threshold \( \phi \in (0, 1) \), we find the fraction of accepted couplings to equal

\[
P\left[P(J \neq 0|\vec{J}^0) > \phi\right] = H \left[\frac{1}{\sqrt{q}} \left(-\sqrt{q} \vec{J}^0 + \arccosh\frac{2e^{\hat{\phi}/2}(1-\phi)}{\phi}\right)\right] + H \left[\frac{1}{\sqrt{q}} \left(\sqrt{q} \vec{J}^0 + \arccosh\frac{2e^{\hat{\phi}/2}(1-\phi)}{\phi}\right)\right].
\]

By strict analogy to equations (40), we can determine the recall and the precision of the inferred network, depending on the cut-off parameter \( \phi \):

\[
\text{RC}(\phi) = \frac{k_1P[J \neq 0| \pm 1] > \phi + k_2P[J \neq 0| \pm 2] > \phi]}{k_1 + k_2}
\]

\[
\text{PR}(\phi) = \frac{k_1P[J \neq 0| \pm 1] > \phi + k_2P[J \neq 0| \pm 2] > \phi]}{(1-k_1-k_2)P[J \neq 0|0] > \phi + k_1P[J \neq 0| \pm 1] > \phi + k_2P[J \neq 0| \pm 2] > \phi]}.
\]

Again, for a very strict acceptance criterion \( \phi \) slightly below 1, we start with a small recall, but high precision. Approaching zero with the cut-off, the recall becomes better and better, but the precision goes down.

5.3. The algorithmic performance

5.3.1. Feasible data and perfect reconstruction. In section 4.1.1 we have seen that, in the case of a feasible data set, a discontinuous jump from a region of bad network inference to one of perfect network inference appears. This jump can be easily understood on the basis of the phase diagram displayed in figure 5. If we have a relatively small number of input patterns (\( \alpha < 1.287 \) in the parameter choice of the figure), there are a large variety of \( \vec{J} \) vectors perfectly reproducing the data set. The point is, however, that almost all of them are very dense, i.e., they have a large \( n_{\text{eff}} \). Without an external field favouring diluted couplings, the algorithm automatically concentrates on the maximum entropy point at high \( n_{\text{eff}} \) (which we found numerically to be always at \( n_{\text{eff}} \geq 0.5 \)), and the quality of
The existence of a first-order transition underlines the necessity to introduce the diluting field \( h \). Without the field, one would have to work at \( \alpha > 1.287 \), which, in the case shown in section 4.1.1, would correspond to more than 770 patterns. With the use of the field, we have perfect generalization already with slightly more than 32 patterns. This increase in predictive power is impressive, remembering in particular that the diluting field was biologically motivated by the fact that gene regulatory networks are sparse.

5.3.2. Unfeasible data and partial reconstruction. As already discussed before, the feasible situation is very unrealistic, and therefore only a first (but important) test case. In the following, we are going to investigate the behaviour for heterogeneous and possibly noisy data generators.

In this section, we therefore consider data coming from a generator with couplings \( J_i^0 \in \{0, \pm1, \pm2\} \), but try to infer them on the basis of our ternary classification to

\[ \text{Figure 5. Phase diagram for a feasible data set with } k_1 = 0.005, k_2 = 0, \gamma = 0. \]

The red line separates the phase space region with errorless solutions from a region where the ground state energy is positive. The always existing perfect solution \( \vec{J} = \vec{J}_0 \) is given by the green line.
activators, repressors and irrelevant inputs. The more complex nature of the data generator leads to the fact that no perfect zero-energy solution exists any more, and we have to go to finite energies (or finite temperatures) in the algorithm. As explained before, the parameters are fixed such that we work at various preselected dilutions, and the temperature is chosen such that we are in a corresponding ground state, i.e., we work at zero entropy.

More precisely, throughout the rest of the section we concentrate on the parameter selection \( k_1 = k_2 = 0.025 \), i.e., 2.5% of all relevant links are weak activators or repressors, and the other 2.5% are strong couplings. The remaining 95% are irrelevant inputs.

Let us first concentrate on the errorless case, \( \gamma = 0 \), and investigate the influence of the number \( \alpha \) of presented patterns per input bit. In figure 6 we have given the results for the precision versus recall curves at various dilutions. If the density of the inferred network is set too high (\( n_{\text{eff}} = 0.1 \) in the figure), the performance is not very good, but it increases if we further dilute the inferred network. The best performance is obtained if the dilution of the inferred network is slightly stronger than that of the data generator. The reason for this finding is very clear: in this way the algorithm keeps the strongest good signals, but cuts many of the strongest false positive signals—obviously at the cost of introducing some false negatives. A stronger diluting field would cut out more true positives which become thereby false negatives; a weaker field would include more false positives which before were true negatives. A more formal reason can be seen by looking at equation (54): in this equation the effective signal due to a coupling \( J^0 \) is given by \( \hat{r} J^0 \), and it is distorted by the noise \( x \sqrt{\hat{q}} \), with \( x \) being normally distributed. The best inference is possible if diverse signals are maximally separable, i.e., if the noise-to-signal ratio \( \sqrt{\hat{q}}/\hat{r} \) is minimal. In figure 7, we plot this ratio for various values of \( \alpha \), as a function

---

**Figure 6.** Theoretical results for the performance of BP versus pair correlation based methods like ARACNE, for various values of \( \alpha = M/N \).
Figure 7. Inverse signal-to-noise ratio: best inference at given data set size \( \alpha \) is possible at the minimum of the curve.

Figure 8. Theoretical versus numerical results. The numerical results are produced at \( N = 600 \), and averaged over 500 realizations of the disorder.

of the effective dilution \( n_{\text{eff}} \). The optimal value is found at \( n_{\text{eff}} < n_{\text{eff}}^0 \); we also see that the position of the minimum moves toward \( n_{\text{eff}}^0 \) for growing \( \alpha \).

In figure 6, we have also included the corresponding curves for algorithms like ARACNe based on pairwise correlations—the message-passing approach takes into account collective effects and therefore clearly outperforms the local pair correlation methods.

In figure 8 we have compared the analytical findings with the performance of BP on single samples of size \( N = 600 \), averaged over 500 samples. Even if the algorithm includes a Gaussian approximation compared to the exact BP equations, the coincidence of the two approaches is very good.
Figure 9. Increase in performance with $\alpha$, i.e. with the number of presented patterns.

Figure 10. Prediction error as a function of the number $M = \alpha N$ of presented patterns, for both algorithms, based on pairwise mutual information (red line) and belief propagation (green line), measured at constant dilution $n_{\text{eff}} = 0.04$.

Another striking finding of figure 6 is the strong improvement of the predictive quality with increasing data sets: for $\alpha = 0.1$, the precision starts to decrease immediately with increasing recall, whereas it stays practically equal to 1 for higher values of $\alpha$; cf also figure 9. This observation can be quantified better by measuring the prediction error

$$\Omega = 1 - \int_0^1 \text{PR} \, d\text{RC}$$

which gives the area over the precision versus recall curve. The prediction error is zero for a perfect predictor, and goes to $1 - k_1 - k_2 = 0.95$ for a completely random predictor.
In figure 10 we have plotted the prediction error as a function of the size of the data set as given by $\alpha$: the curves for both algorithm types start obviously at $\Omega = 0.95$ for empty data sets ($\alpha = 0$); without data there cannot be any prediction. The error then decreases exponentially with growing $\alpha$, but we see that BP behaves much better than the curves corresponding to the mutual information.

To complete our analysis, we have also included errors in the data generator, fixing all other parameters to $\alpha = 0.5$ and $n_{\text{eff}} = 0.04$. Other parameter values lead to qualitatively equivalent results. In figures 11 and 12 we have plotted the precision versus recall curves.

**Figure 11.** Precision versus recall of BP for various noise strengths $\gamma = 0.0, 0.025, 0.05, 0.075, 0.1, 0.125$ (curves from right to left), ranging from no noise to equal signal and noise strengths.

**Figure 12.** Prediction error of BP as a function of the noise strength $\gamma$, for various values of $\alpha$. 

In figure 10 we have plotted the prediction error as a function of the size of the data set as given by $\alpha$: the curves for both algorithm types start obviously at $\Omega = 0.95$ for empty data sets ($\alpha = 0$); without data there cannot be any prediction. The error then decreases exponentially with growing $\alpha$, but we see that BP behaves much better than the curves corresponding to the mutual information.

To complete our analysis, we have also included errors in the data generator, fixing all other parameters to $\alpha = 0.5$ and $n_{\text{eff}} = 0.04$. Other parameter values lead to qualitatively equivalent results. In figures 11 and 12 we have plotted the precision versus recall curves.
and the prediction error for various values $\gamma$ of the noise strength. The predictive power of the algorithms obviously decreases with increasing noise level. Note, however, that even for $\gamma = k_1 + 4k_2$, where the statistical properties of the signal and of the noise are equivalent, a non-trivial prediction is possible.

### 6. Conclusion and outlook

In this paper, we have analysed the performance of two approaches to the identification of gene regulatory interactions. Whereas the first method is based on a ranking of all gene pairs according to the measured correlations between their expression levels, the second method tries to include collective effects of activators and repressors coming from the joint action of multiple regulators. This second method, developed by the same authors in an earlier publication, uses message-passing techniques (belief propagation) which can be understood as an algorithmic reinterpretation of the cavity method in spin-glass physics.

Using a simple data generator, we achieved a theoretical analysis of the algorithmic performance. In the simpler case of pairwise correlations, this analysis is based on a straightforward application of the central limit theorem. The more involved case of the second algorithm requires the use of other spin-glass techniques, more precisely of the replica method, in order to perform a generalized Gardner calculation in the space of all candidate gene networks. We found that algorithms taking into account collective effects beyond pair correlations work better. The intuitive reason for this finding is that the best joint prediction is not necessarily obtained for the set of the best individual predictors. These results are true even if the data generator is more complex than the inference algorithm. We tested explicitly the cases of heterogeneous regulation strengths and noisy data.

There are some interesting directions to be followed. First of all, these algorithms are designed to be applied to real biological data. As already seen in [17], BP can also be used to extract sparse and predictive information from gene expression data. Still there is space for substantial improvements via integration of biological knowledge which goes beyond a simple diluting field. One possibility is, e.g., the integration of sequence information on putative transcription factor binding sites via inhomogeneous diluting fields.

One drawback for both the data generator considered and the derivation of the BP algorithm is that we neglected correlations between inputs. These correlations exist, however, in biological data, since transcription factors may also be co-regulated. An interesting approach in this direction was recently presented by Kabashima [32], and can be extended to our model. It would be interesting to see under which conditions correlations really change the behaviour of the algorithms.

A last point is the application of the algorithm to other problems. In fact it gives more generally a sparse Bayesian classifier. Bioinformatical applications of such classifiers goes far beyond the inference of gene networks. One example that we are currently analysing is the classification of expression patterns for cancer tissues according to the diagnosis (cancer versus healthy tissues, different cancer subtypes), and the detection of predictive but sparse gene signatures for the different diagnoses [30,33].

### Acknowledgments

We are grateful to Yoshiyuki Kabashima, Michele Leone and Francesca Tria for interesting and helpful discussions.
Inference algorithms for gene networks

References

[1] Alberts B, Johnson A, Lewis J, Raff M, Bray D, Hopkin K, Roberts K and Walter P, 2003 Essential Cell Biology (New York: Garland Science)
[2] Shen-Orr S S, Milo R, Mangan Sh and Alon U, 2002 Nat. Genet. 31 64
[3] Guelzim N, Bottani S, Bourgini P and Kepes F, 2002 Nat. Genet. 31 60
[4] Milo R, Shen-Orr S S, Itzkovitz S, Kashtan N, Chklovskii D and Alon U, 2002 Science 25 824
[5] Bolouri H and Davidson E, 2002 Dev. Biol. 246 2
Davidson E et al, 2002 Science 295 1669
[6] Albert R and Othmer H G, 2003 J. Theor. Biol. 223 1
[7] Butte A J and Kohane I S, 1999 Full Symposium, American Medical Informatics Association ed N Lorenzi (Washington, DC: Hanley and Belfus)
[8] Butte A J, Tamayo P, Slonin D, Golub T R and Kohane I S, 2000 Proc. Nat. Acad. Sci. 97 12182
[9] Margolin A A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla-Favera R and Califano A, 2006 BMC Bioinformatics 7 S7
[10] Basso K, Margolin A, Stolovitzky G, Klein U, Dalla-Favera R and Califano A, 2005 Nat. Genet. 37 382
[11] Murphy K and Mian S, 1999 Technical Report University of California, Berkeley
[12] Friedman N, Linial M, Nachman I and Pe’er D, 2000 J. Comput. Biol. 7 601
[13] Jing Yu, Smith V A, Wang P P, Hartemink A J and Jarvis E D, 2004 Bioinformatics 20 3594
[14] Hartemink A J, 2005 Nature 23 554
[15] Shmulevich I, Dougherty E R and Zhang W, 2002 Proc. IEEE 90 1778
[16] Hashimoto R F, Kim S, Shmulevich I, Zhang W, Bittner M L and Dougherty E R, 2004 Bioinformatics 20 1241
[17] Baillet-Bechet M, Braunstein A, Pagnani A, Weigt M and Zecchina R, 2008 Preprint
[18] Braunstein A, Pagnani A, Weigt M and Zecchina R, 2008 J. Phys.: Conf. Ser. 95 012016
[19] Buchler N, Gerland U and Hwa T, 2003 Proc. Nat. Acad. Sci. 100 5130
[20] Hertz J, Krogh A and Palmer R G, 1991 Introduction to the Theory of Neural Computation (Redwood, CA: Addison-Wesley)
[21] Engel A and van den Broeck C, 2001 Statistical Mechanics of Learning (New York: Cambridge University Press)
[22] Gardner E, 1987 Europhys. Lett. 4 481
Gardner E, 1988 J. Phys. A: Math. Gen. 21 257
[23] Tilgherany R, 1996 J. R. Stat. Soc. B 58 267
[24] Ravikumar P, Wainwright M J and Lafferty J D, 2008 Technical Report Dep. of Statistics, UC Berkeley
[25] Banerjee O, El Ghaoui L, d’Aspremont A and Natsoulis G, 2006 ACM Int. Conf. Proc. Series (Pittsburgh, PA) vol 148, p 89
[26] Lee S I, Ganapathi V and Koller D, 2007 NIPS 2006: Advances in Neural Information Processing Systems ed B Schölkopf, J Platt and T Hofmann (Cambridge, MA: MIT Press)
[27] Schmidt M, Niculescu-Mizil A and Murphy K, 2007 Proc. 22nd AAAI Conf. on Artificial Intelligence American Association of Artificial Intelligence
[28] Meinhausen N and Buehlmann P, 2006 Ann. Stat. 34 1436
[29] Kabashima Y, 2003 J. Phys. A: Math. Gen. 36 11111
[30] Uda S and Kabashima Y, 2005 J. Phys. Soc. Japan 74 2233
[31] Braunstein A and Zecchina R, 2006 Phys. Rev. Lett. 96 030201
[32] Kabashima Y, 2008 J. Phys.: Conf. Ser. 95 012001
[33] Tria F, Pagnani A and Weigt M, 2008 submitted

doi:10.1088/1742-5468/2008/12/P12001