Published in IET Systems Biology
Received on 10th August 2009
Revised on 4th June 2010
doi: 10.1049/iet-syb.2009.0047

ISSN 1751-8849

CORE-Net: exploiting prior knowledge and preferential attachment to infer biological interaction networks

F. Montefusco[1,2] C. Cosentino[1] F. Amato[1]
[1]Department of Experimental and Clinical Medicine, School of Computer and Biomedical Engineering,
Università degli Studi Magna Græcia, v.le Europa, Campus Salvatore Venuta, 88100 Catanzaro, Italy
[2]College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, Northpart Road,
Exeter EX4 4QF, UK
E-mail: carlo.cosentino@unicz.it

Abstract: The problem of reverse engineering in the topology of functional interaction networks from time-course experimental data has received considerable attention in literature, due to the potential applications in the most diverse fields, comprising engineering, biology, economics and social sciences. The present work introduces a novel technique, CORE-Net, which addresses this problem focusing on the case of biological interaction networks. The method is based on the representation of the network in the form of a dynamical system and on an iterative convex optimisation procedure. A first advantage of the proposed approach is that it allows to exploit qualitative prior knowledge about the network interactions, of the same kind as typically available from biological literature and databases. A second novel contribution consists of exploiting the growth and preferential attachment mechanisms to improve the inference performances when dealing with networks which exhibit a scale-free topology. The technique is first assessed through numerical tests on in silico random networks, subsequently it is applied to reverse engineering a cell cycle regulatory subnetwork in *Saccharomyces cerevisiae* from experimental microarray data. These tests show that the combined exploitation of prior knowledge and preferential attachment significantly improves the predictions with respect to other approaches.

1 Introduction

The development of reverse engineering techniques to unravel the topology of biological interaction networks is a crucial and challenging problem. The ever increasing knowledge base in the field of molecular biology, combined with the achievements of biotechnology and the availability of huge experimental data sets, requires the deployment of systems engineering methodologies, able to formally describe such complexity through mathematical models and to use them to unveil the underpinning basic mechanisms.

The intense research effort in the last decade has spawned a large number of techniques dealing with the reverse engineering of biological systems and particularly with the problem of inferring the topology of gene regulatory networks (GRN). At present, the predominant methods in the literature are those based on statistics (e.g. Bayesian networks (BNs) [1]), information theory (e.g. mutual information [2]) and system identification theory (e.g. linear regression [3, 4], principal component regression [5], partial least square [6]); the reader is referred to [7] for an extensive review of the literature on the subject. Of course, each technique is better suited for a certain case, depending on the size and type of the network, of the data set and of the experiments. Nonetheless, comparative studies on both synthetic and real data sets show that these approaches cannot yet be considered mature and well assessed, because they have not yet reached the high levels of performance, reliability and standardisation required for being commonly adopted in biological studies [8, 9].

A common weakness of most of the aforementioned approaches (excluding Bayesian networks) can be certainly addressed to their limited capability in conjugating the theoretical treatment of the problem with the wide availability of information that can be drawn from biological journals and databases: in the majority of cases the network topology is assumed to be completely unknown, that is, the inference algorithm takes as inputs only a list of genes (which are the candidate nodes of the network) and their expression profiles. However, this scenario is very far from reality: on the one hand, several decades of research in molecular biology have provided us with an extensive knowledge base on the GRN of a number of organisms, which is now consolidating in the form of digital databases, endowed with tools for the creation of readily intelligible graphs. Therefore at present it is much more likely that the topology of the network to be inferred is partially known and one is interested in seeking novel gene functional interactions to expand the current knowledge. On the other hand, studies on the identifiability of biological models have clearly demonstrated that in the general case, even when the model structure is assumed to be known (which is a very optimistic hypothesis), the model parameters are not identifiable using just the experimental measurements without adding extra information or imposing constraints in the parameters space [10, 11, p. 19, 12]. An interesting approach based on dynamical system identification, proposed in [13], exploits mechanistic models, taking into account all possible uni- and bimolecular reactions, which, however, seems to be only suitable for small networks. It can exploit prior information, using the representation matrix model, if the reaction structure is known and one wants to estimate the reaction constants. In [14] the authors have developed a mixed-integer non-linear programming (MINLP)-based optimisation approach for constructing a coarse-grained model (CGM), by using qualitative information and biochemical information as constraints into the minimisation problem.

The present work fits in the framework of network reconstruction methods based on dynamical systems theory. This approach is particularly well suited when dealing with time-course experiments, especially when the number of experimental points is relatively low; indeed, the approaches based on statistical models require large data sets and/or assume the samples are independent, which is clearly not true when we consider the measurements of the expression of the same gene at two consecutive time points.

In [15] we have introduced a novel machinery to formulate the system identification problem as a convex optimisation procedure cast in the form of linear matrix inequalities (LMIs). The distinctive feature of this method is that, differently from other techniques based on dynamical systems, it enables to straightforwardly embed prior knowledge on the network topology into the inference process, thus significantly increasing the reconstruction performance.

This approach is receiving increasing attention: Julius *et al.* [16] added a stability constraint in the convex optimisation and used the l_1-norm in the cost function in order to obtain a sparse connectivity matrix, whereas August and Papachristodoulou [17] extended the convex optimisation approach by using polynomial models.

The present work provides novel contributions of both methodological and applicative nature: first of all, it improves the method devised in [15], adopting a strategy that enables to exploit the topological features of biological networks. Indeed, experimental evidences show that biological networks exhibit a scale-free topology [18–20], that entails a power-law distribution of the connectivity degree. A novel method, named CORE-Net (Convex Optimisation algorithm for Reverse Engineering biological interaction Networks), is proposed to exploit this information to improve the inference performances. The basic idea underlying our approach is that of using, within the reverse engineering process, the same mechanisms proposed by [20], Albert and Barabási [21], namely 'growth' and 'preferential attachment', which underpin the generation process of a scale-free network. These mechanisms are combined with the machinery presented in [15], which allows one to take into account individual known interactions from biological experiments, databases and literature.

The second contribution of the paper is represented by a comparison with other state-of-the-art reverse engineering techniques, based on dynamic Bayesian networks and mutual information theory, on both in silico and in vitro data sets. Such tests show that, when dealing with time-series data, the proposed technique outperforms the others.

Finally, the applicability of the devised technique is illustrated by tackling a real biological case study with in vitro experimental data (whereas in [15] only in silico data were considered), namely the reverse engineering of a regulatory subnetwork of the cell cycle in *Saccharomyces cerevisiae*.

The paper is structured as follows: Section 2 describes the network model. Section 3 illustrates the reverse engineering technique. The results obtained in the numerical tests and in the biological case study are reported in Section 4. Finally, conclusions are given in Section 5.

2 Methods

Since the output of reverse engineering algorithms is often given in terms of graphs (either in numerical or graphical format), it is worth briefly recalling some basic facts about the representation of networks through graphs and understanding how it is related to the class of mathematical models that is adopted in the sequel.

2.1 Network representation

Biological interaction networks can be conveniently described in a formal and concise way by means of 'graphs', where the nodes correspond to molecular species (e.g. genes, mRNA, transcription factors) and an edge (which is defined by a pair of nodes) represents a 'functional' interaction: an edge between gene A and gene B does not mean that there is a physical interaction, but rather that the product of the transcription of A can affect the expression of B. The graph is said to be 'directed', or a 'digraph', when the edges are directed: in this case the edge is defined by an 'ordered' pair of nodes. A further level of detail can be achieved by associating a sign (or a weight) to each edge, which represents the type (and the extent) of the effect that A has on B. In the latter case, we will say that the graph is 'signed' (or 'weighted').

Two nodes of the network are called adjacent if one 'directly' influences the expression of the other. Indeed, the network topology can create a non-direct influence (a path) between two nodes, for example, if A is a transcription factor for B, which inhibits the expression of C, then A can be considered, indirectly, an antagonist of C as well. The main goal of a topological inference algorithm, then, is to recover a map of the adjacent nodes. A compact way to define a weighted digraph of n nodes and m edges is in the form of a $n \times n$ connectivity matrix, having m non-zero coefficients.

2.2 Network model

The dynamical evolution of a biological network can be described, at least for small excursions of the involved quantities from the equilibrium point, by means of linear systems, made up of ordinary differential equations (ODEs) in the continuous-time case, or difference equations in the discrete-time case (see [3, 22] and references therein).

In this paper, we consider the continuous-time linear time-invariant model

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}u(t) \tag{1}$$

where $\boldsymbol{x}(\boldsymbol{t}) = (x_1(t), \ldots, x_n(t))^{\mathrm{T}} \in \mathbb{R}^n$, the state variables x_i, $i = 1, \ldots, n$, represent the levels of the different compounds present in the system (e.g. mRNA concentrations for gene expression levels), $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the dynamic matrix and $\boldsymbol{B} \in \mathbb{R}^{n \times 1}$ is a vector that determines the direct targets of external perturbations, $u(t) \in \mathbb{R}$ (e.g. drugs, overexpression or downregulation of specific genes), which are typically induced during in vitro experiments.

Note that the derivative (and therefore the evolution) of x_i at time t is directly influenced by the value $x_j(t)$ iff $A_{ij} \neq 0$. Moreover, the type (i.e. promoting or inhibiting) and extent of this influence can be associated to the sign and magnitude of the element $\boldsymbol{A}_{ij}$, respectively. In view of these considerations, if we look at the state variables as quantities associated to the nodes of a network, the matrix $\boldsymbol{A}$ can be considered as a compact numerical representation of the network topology. Therefore the topological reverse engineering problem can be rephrased as the problem of identifying the dynamical system (1). A possible criticism regards the use of a linear model, which are certainly inadequate to capture the complex non-linear dynamics of certain molecular reactions. This issue would be reasonable if we aimed to identifying a reliable model for describing and predicting the evolution of a biological system; in this case, instead, the goal is just that of capturing the qualitative functional relationships between the states of the system, which can be usually done by a first-order linear approximation. Indeed, a number of other approaches, which are based on linear dynamical models as well, can be found in the literature (e.g. [3, 23, 24]). The advantage of recasting the problem in the framework of linear systems identification relies on the existence of well-established and computationally appealing techniques, which can be tailored to the specific application.

3 Inference algorithm

3.1 Identification of the connectivity matrix

Given a set of experimental measurements, the basic step of the inference process consists in estimating the weighted connectivity matrix $\boldsymbol{A}$ and the exogenous perturbation vector $\boldsymbol{B}$ of system (1). In the framework of system identification, this problem is classically dealt with by means of a least-squares estimator (LSE) and a regression algorithm. Here, instead, we will formulate it as a convex optimisation problem in the form of LMIs.

Assume that $h + 1$ experimental observations, $x(k) \in \mathbb{R}^n$, $k = 0, \ldots, h$, are available, then we can recast the problem in the discrete-time domain as

$$\Xi := \begin{pmatrix} x(h) & \ldots & x(1) \end{pmatrix} = \Theta\Omega \tag{2}$$

where

$$\boldsymbol{\Theta} = [\hat{\boldsymbol{A}}\ \hat{\boldsymbol{B}}], \quad \boldsymbol{\Omega} := \begin{pmatrix} x(h-1) & \ldots & x(0) \\ u(h-1) & \ldots & u(0) \end{pmatrix}$$

The identification problem can be transformed into that of minimising the norm of $\boldsymbol{\Xi} - \boldsymbol{\theta\Omega}$, thus we state the following problem.

Problem 1: Given the sampled data set $x(k)$, $k = 0, \ldots, h$, and the associated matrices $\boldsymbol{\Xi}$, $\boldsymbol{\Omega}$, find

$$\begin{aligned} &\min_{\Theta} \varepsilon \\ &\text{s.t.} \quad (\Xi - \Theta\Omega)^{\mathrm{T}}(\Xi - \Theta\Omega) < \varepsilon \boldsymbol{I} \end{aligned} \tag{3}$$

Note that condition (3) is quadratic in the unknown matrix variable $\boldsymbol{\Theta}$. In order to obtain a linear optimisation

doi: 10.1049/iet-syb.2009.0047

problem, we turn it into the equivalent condition

$$\begin{pmatrix} -\varepsilon \boldsymbol{I} & (\boldsymbol{\Xi} - \boldsymbol{\Theta\Omega})^{\mathrm{T}} \\ (\boldsymbol{\Xi} - \boldsymbol{\Theta\Omega}) & -\boldsymbol{I} \end{pmatrix} < 0 \tag{4}$$

by applying the properties of Schur complements (see [25, p. 123]). Problem 1 with the inequality constraint in the form (4) is a generalised eigenvalue problem ([26, p. 10]), and can be easily solved through off-the-shelf efficient numerical algorithms, such as those implemented in the Matlab LMI Toolbox [27].

A noteworthy advantage of the proposed convex optimisation formulation is that the approach can be straightforwardly extended to the case when multiple experimental data sets are available for the same biological network. In this case, there are several matrix pairs $(\boldsymbol{\Xi}_k, \boldsymbol{\Omega}_k)$, one for each experiment: the problem can be formulated again as in (3), but using a number of constraints equal to the number of experiments, that is

$$\begin{aligned} &\min_{\Theta} \varepsilon \\ &\text{s.t.} \quad (\boldsymbol{\Xi}_k - \boldsymbol{\Theta\Omega}_k)^{\mathrm{T}}(\boldsymbol{\Xi}_k - \boldsymbol{\Theta\Omega}_k) < \varepsilon_k \boldsymbol{I}, \quad k = 1, \ldots, N_{\mathrm{e}} \end{aligned}$$

where N_{e} is the number of available experiments.

It is worth noticing that, since the system evolution is sampled, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ are not actually the estimates of $\boldsymbol{A}$ and $\boldsymbol{B}$ in (1), but rather of the corresponding matrices of the discrete-time system obtained through the zero-order-hold (ZOH) discretisation method ([28, p. 676]) with sampling time T_{s} from system (1), that is

$$\boldsymbol{x}(k+1) = \boldsymbol{A}_{\mathrm{d}}\boldsymbol{x}(k) + \boldsymbol{B}_{\mathrm{d}}u(k) \tag{5}$$

where $\boldsymbol{x}(k+1)$ is a shorthand notation for $\boldsymbol{x}(kT_{\mathrm{s}} + T_{\mathrm{s}})$, $\boldsymbol{x}(k)$ for $\boldsymbol{x}(kT_{\mathrm{s}})$, $u(k)$ for $u(kT_{\mathrm{s}})$, and

$$\boldsymbol{A}_{\mathrm{d}} = \mathrm{e}^{AT_{\mathrm{s}}}, \quad \boldsymbol{B}_{\mathrm{d}} = \left(\int_0^{T_{\mathrm{s}}} \mathrm{e}^{A\tau}\, \mathrm{d}\tau\right) B$$

In general, the sparsity patterns of A_{d} and B_{d} differ from those of $\boldsymbol{A}$ and $\boldsymbol{B}$. However, if the sampling time is suitably small, $(\boldsymbol{A})_{ij} = 0$ implies that $(\boldsymbol{A}_{\mathrm{d}})_{ij}$ exhibits a very low value, compared to the other elements on the same row and column, and the same applies for $\boldsymbol{B}_{\mathrm{d}}$ and $\boldsymbol{B}$ (see the Appendix for a detailed discussion). Therefore in order to reconstruct the original sparsity pattern of the continuous-time system's matrices, one can set to zero the elements of the estimated matrices whose values are below a certain threshold; this is the basic principle underpinning the edges selection strategy (see the next subsection).

Besides the LMI formulation, the problem is identical to the one tackled by classical linear regression, that is, finding the values of $n(n+1)$ parameters of a linear model that yield the best fitting of the observations in the least-squares sense. Hence, if the number of observations, $n(h+1)$, is greater or equal than the number of explanatory variables, that is $h \geq n$, the problem admits a unique globally optimal solution. In the other case, $h < n$, the interpolation problem is undetermined, thus there exist infinitely many values of the optimisation variables that equivalently fit the experimental measurements. In the latter case, several expedients can be adopted to solve the undetermination: first, it is crucial to exploit clustering techniques to reduce the number of nodes and smoothing techniques to increase the number of samples, in order to satisfy the constraint $h \geq n$. Furthermore, adopting a bottom-up reconstruction approach (i.e. starting with a blank network and increasingly adding new edges) may help in overcoming the dimensionality problem: in this case, indeed, the number of edges incident to each node (and therefore the number of explanatory variables) is iteratively increased and can be limited to satisfy the above constraint. Finally, the introduction of sign constraints on the optimisation variables, derived by available prior knowledge on the network topology, implies a significant reduction of the solution space.

In the light of the latter consideration, it is important to devise a method to take into account prior knowledge about the network topology within the optimisation procedure. Since each element of $\boldsymbol{A}$ can be interpreted as the weight of the edge between two nodes of the network, the goal can be achieved by forcing some of the optimisation variables to be zero and others to be strictly positive (or negative), introducing the additional inequality $\boldsymbol{A}_{ij} > 0$ (<0) to the set of LMIs. Similarly, we can impose a sign constraint on the ith element of the input vector, $\boldsymbol{b}_i$, if we a priori know the qualitative (i.e. promoting or repressing) effect of the perturbation on the ith node. An edge can be easily pruned from the network, instead, by setting to zero the corresponding entry in the matrix optimisation variable in the LMIs.

3.2 Edges selection strategy

So far we have devised a method to add/remove edges and to introduce constraints on the sign of the associated weights in the optimisation problem. The problem remains of how to devise an effective strategy to select the non-zero entries of the connectivity matrix.

The initialisation network for our algorithm has only self-loops on every node, that means the evolution of the ith state variable is always influenced by its current value. This yields a diagonal initialisation matrix, $\hat{\boldsymbol{A}}^{(0)}$. Subsequently, new edges are added step-by-step to the network according to the following heuristic iterative procedure:

(P1) A first matrix, $\bar{\boldsymbol{A}}$, is computed by solving Problem 1, without setting any optimisation variable to zero. The available prior information is taken into account at this point by adding the proper sign constraints on the corresponding entries of $\boldsymbol{A}$ before solving the optimisation problem, as explained in the previous subsection. Since it typically exhibits all non-zero entries, matrix $\bar{\boldsymbol{A}}$ is not representative of the network topology, but it is rather used to weigh the relative influence of each entry on the system's dynamics. Such information will be used to select the edges to be added to the network at each step. Each element of $\bar{\boldsymbol{A}}$ is normalised with respect to the values of other elements on the same row and column, which yields the matrix $\tilde{\boldsymbol{A}}$, whose elements are defined as

$$\tilde{\boldsymbol{A}}_{ij} = \frac{\bar{\boldsymbol{A}}_{ij}}{(\|\bar{\boldsymbol{A}}_{\star,j}\| \cdot \|\bar{\boldsymbol{A}}_{i,\star}\|)^{1/2}}$$

where $\bar{\boldsymbol{A}}_{ij}$ is the (i, j) entry of $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{A}}_{\star,j}$ ($\bar{\boldsymbol{A}}_{i,\star}$) denotes the jth column (ith row).

(P2) At the kth iteration, the edges ranking matrix $\tilde{\boldsymbol{G}}^{(k)}$ is computed.

$$\tilde{\boldsymbol{G}}_{ij}^{(k)} = \frac{|\tilde{\boldsymbol{A}}_{ij}|p_j^{(k)}}{\sum_{l=1}^{n} p_l^{(k)}|\tilde{\boldsymbol{A}}_{il}|} \tag{6}$$

where

$$p_j^{(k)} = \frac{K_j^{(k)}}{\sum_{l=1}^{n} K_l^{(k)}} \tag{7}$$

is the probability of inserting a new edge starting from node j and $K_l^{(k)}$ is the number of outgoing connections from the lth node at the kth iteration. The $\mu(k)$ edges with the largest scores in $\tilde{\boldsymbol{G}}^{(k)}$ are selected and added to the network; $\mu(\cdot)$ is chosen as a decreasing function of k, that is, $\mu(k) = \lceil n/k \rceil$; thus, the network grows rapidly at the beginning and is subsequently refined by adding few nodes at each iteration. The form of the function $p(\cdot)$ stems from the so-called preferential attachment (PA) mechanism, which states that in a growing network new edges preferentially start from popular nodes (those with the highest connectivity degree, i.e. the hubs).

(P3) The mask of non-zero elements of $\hat{\boldsymbol{A}}^{(k)}$ is defined by adding the entries selected at point (P2) to those selected until iteration $k-1$ (including those derived by a priori information), and the set of inequality constraints is updated accordingly; then Problem 1, with the additional constraints, is solved to compute $\hat{\boldsymbol{A}}^{(k)}$.

(P4) The residuals generated by the identified model are compared with the values obtained at previous iterations; if the norm of the vector of residuals has decreased, in the last two iterations, at least by a factor ϵ_r with respect to the value at the first iteration, then the procedure iterates from point (P2), otherwise it stops and returns the topology described by the spartsity pattern of $\hat{\boldsymbol{A}}^{(k-2)}$. The factor ϵ_r is inversely correlated with the number of edges inferred by the algorithm; on the other hand, using a smaller value of ϵ_r raises the probability of obtaining false positives. By conducting numerical tests for different values of ϵ_r, we have found that setting $\epsilon_r = 0.1$ usually yields a good balance between the various performance indices.

Concerning the input vector, we assume that the perturbation targets and the qualitative effects of the perturbation are known, thus the pattern (but not the values of the non-zero elements) of $\hat{\boldsymbol{B}}$ is preassigned at the initial step and the corresponding constraints are imposed in all the subsequent iterations.

Finally, it is worth noticing that, differently from what is done in [15], where the algorithm starts from a fully connected network and then the edges are subsequently pruned according to a maximum parsimony criterion, here the procedure iteratively increments the number of edges, thus helping to overcome the undetermination problem, as referred in 3.1, and to decrease the computational burden. On the other hand, in certain cases, the LMI-based iterative pruning approach in [15] can be preferable with respect to the incremental mechanism implemented by CORE-Net$_{\text{noPA}}$, as will be shown in the next section.

4 Results

A first statistical evaluation is performed by testing the algorithm against a set of randomly generated in silico networks. The applicability to a real problem is investigated by applying the method to experimental in vitro data in a biological case study. The results are compared with those obtained by other common inference algorithms.

The tests are evaluated by using two common statistical indices (see [29, p. 138])

- Sensitivity (Sn), defined as

$$\text{Sn} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

which is the fraction of actually existing interactions (TP: = true positives, FN: = false negatives) that the algorithm infers, also named 'Recall'.

- Positive predictive value (PPV)

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

which measures the reliability of the interactions (FP: = false positives) inferred by the algorithm, also named 'Precision'.

To compute these performance indexes, we do not consider the weight of the edge, but only its existence, so the network is considered as a directed graph.

4.1 Assessment with in silico data sets

In this subsection, the performance of the algorithm is assessed by applying it to a set of random networks, 20 of ten nodes and 20 of 20 nodes (see the Appendix for details about the in silico network generation). Different tests have been conducted on each network, using data sets of 20 samples and three levels of measurement noise, corresponding to σ equal to 0, 10 and 30% of the sample value, and assuming two different levels of prior knowledge (PK), 0 and 30% of PK.

4.1.1 Comparison with BANJO: Figs. 1 and 2 show the performances obtained on ten and 20 nodes networks, with and without PK.

The results are compared with those obtained by using a reverse engineering method based on BNs, a theoretical framework that allows one to take into account the a priori biological knowledge in an elegant and principled manner based on the probability theory. In particular, we used the software BANJO (BAyesian Network inference with Java Objects), a tool developed by Hartemink and co-workers [30], which performs structure inference for static and dynamic Bayesian networks (DBNs). In our case, BANJO is set to search for the 'Best' DBNs: for each network and test, a setting file has been defined, to which describes our data and how the inference has to be performed by BANJO.

Besides comparing different methods, in order to validate the inference capability of the algorithms, we also report the results obtained by a random choice of the edges, which is based on a binomial distribution: given any ordered pair of nodes, the existence of a directed edge between them is assumed true with probability p_r, false with probability $1 - p_r$. By varying the parameter p_r in [0, 1], the random inference algorithm produces the results described in terms of (PPV, Sn) by the solid curves in Figs. 1 and 2. Note that, when using the random inference algorithm without prior knowledge, there is no relationship between PPV and Sn, which implies that it is possible to achieve an arbitrary Sn value in [0, 1], whereas the PPV is constant, and depends only on the sparsity coefficient of the original network (PPV is equal to $(1 - \eta)$ for directed networks and $(1 - \eta^2)$ for undirected ones). When the prior knowledge is taken into account, instead, both PPV and Sn have to be considered, because they are no longer independent. The mathematical relationships between PPV, Sn and p_r are reported in Table 2 in [15].

Figs. 1 and 2 also provides a comparison of the results obtained with and without using the PA mechanism. The version of the algorithm without PA (named CORE-Net$_{\text{noPA}}$) is readily obtained by not taking into account the preferential attachment probabilities p_j^k: at each step the new edges to be added are selected only on the basis of the estimated weights, that is $\tilde{\boldsymbol{G}}^{(k)} = \tilde{\boldsymbol{A}}, \forall k$. These methods are also compared with the iterative pruning approach (which does not implement PA) proposed in [15] (denoted LMI-IP).

When no prior knowledge is exploited, the PA heuristic does just slightly improve the performances. The inference power of the algorithm is, of course, maximum in the ideal case of noise-free samples. The results are sensibly different from those obtained by the random reconstruction algorithm, which confirms that our technique actually possesses an inference capability, even though the

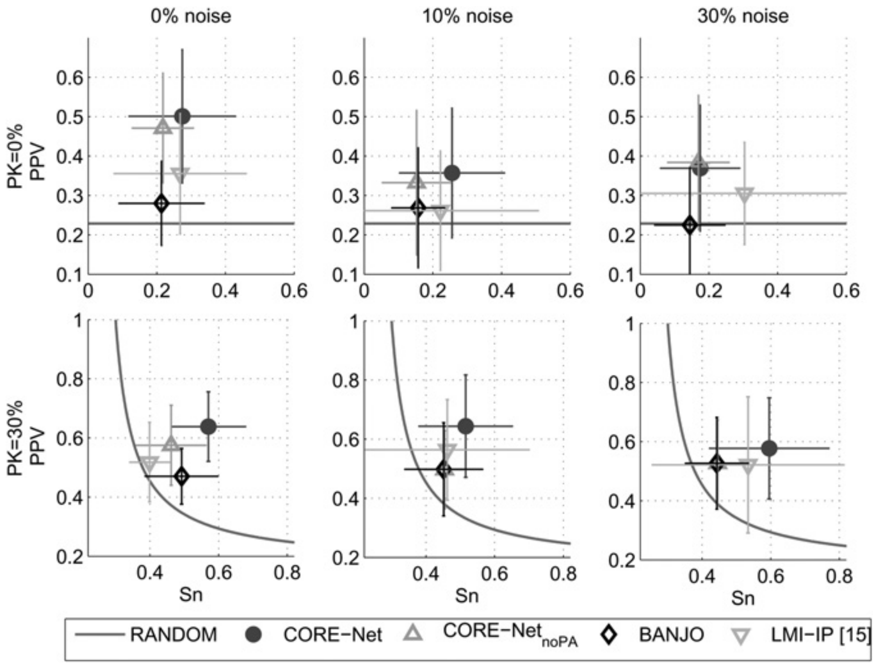

Figure 1 *Results averaged over a set of 20 random scale-free networks of ten nodes, using 20 data points*

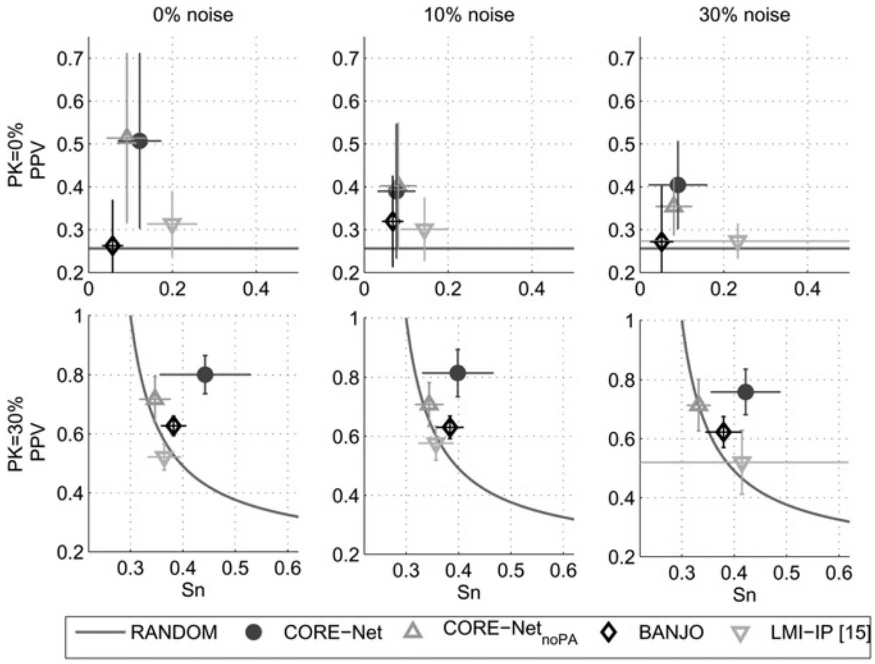

Figure 2 *Results averaged over a set of 20 random scale-free networks of 20 nodes, using 20 data points*

performance indices are lower in the presence of noise. Note also that the algorithm with PA mechanism always performs better than the others.

When a priori information is taken into account, instead, the results of CORE-Net exhibit a significant improvement with respect to other methods. In these experiments, we have assumed a prior knowledge of the 30% of the edges, corresponding to about six edges out of 20 and 29 edges out of 97 for the networks of ten and 20 nodes, respectively. CORE-Net$_{noPA}$, BANJO and LMI-IP all undergo a significant degradation when the number of nodes doubles; indeed, they get very close to the solid curve of the random inference algorithm, whereas CORE-Net remains fairly distant from it. Note also that the measurement noise does not seem to significantly degrade the performances of the various methods.

4.1.2 Comparison with CLR:

A further comparison has been performed between our algorithm and Context Likelihood of Relatedness (CLR), a gene network reconstruction method described in [31], which belongs to the class of algorithms that are based on the theory of relevance networks. Figs. 3 and 4 show the results of CLR over the same data sets used for the previous tests. The plot makes use of the box-and-whisker representation (see [32, p. 8-3]) to report the median PPV (middle line), the 25th and 75th percentile (lower and upper lines of the box) and the outliers (crosses), defined as those values that are more than 1.5 times the interquartile range away from the top or bottom of the box.

Note that CLR cannot take into account PK, therefore only the case with PK = 0% has been considered. Moreover, the networks inferred by CLR are undirected graphs: it is only possible to know whether there is an interaction between two nodes, but not the direction. Therefore the performance indices of both algorithms are computed assuming that the original and reconstructed connectivity matrices are undirected (see [15] for the definition of TP, TN, FP and FN in the undirected case). Finally, to compare the behaviour of the two algorithms at different level of sensitivity, we reported all the networks inferred by CORE-Net at each iteration.

From Figs. 3 and 4 it is evident that CORE-Net always performs better than CLR; moreover, the performances of CLR are comparable with those of the random inference algorithm.

4.1.3 Results with non-scale-free networks:

So far, numerical tests have shown that the PA mechanism actually provides some advantages when dealing with scale-free networks. The question arises of whether this advantage turns into a drawback if the network to be inferred is erroneously assumed to exhibit a scale-free topology.

Therefore here we show that the results of the algorithm adopting the PA mechanism do not undergo a significant degradation when the networks have a non-scale-free topology, but rather they are comparable to those obtained by CORE-Net$_{noPA}$. To this aim, both CORE-Net and CORE-Net$_{noPA}$ have been tested over a set of randomly generated networks, exhibiting Erdős–Rényi topology and the results are compared with those obtained by BANJO (see Figs. 5 and 6).

Concerning the topology assigned to these in silico networks, the number of non-zero off-diagonal elements

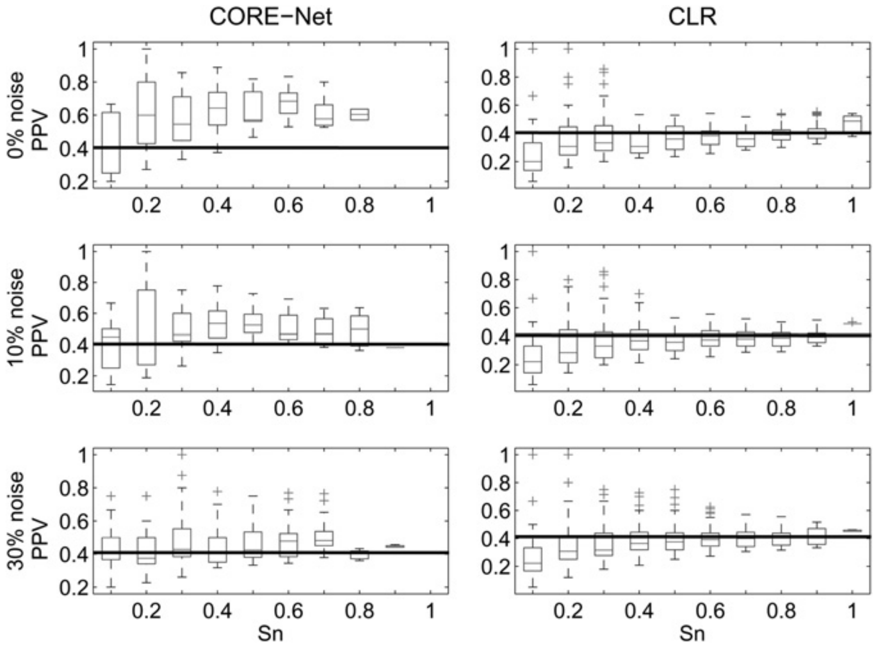

Figure 3 *PPV value at different sensitivity levels, with random scale-free (undirected) networks of ten nodes with 20 samples, reported in the box-and-whisker form*

Solid line represents the performance, in the same experimental conditions, of the random inference algorithm

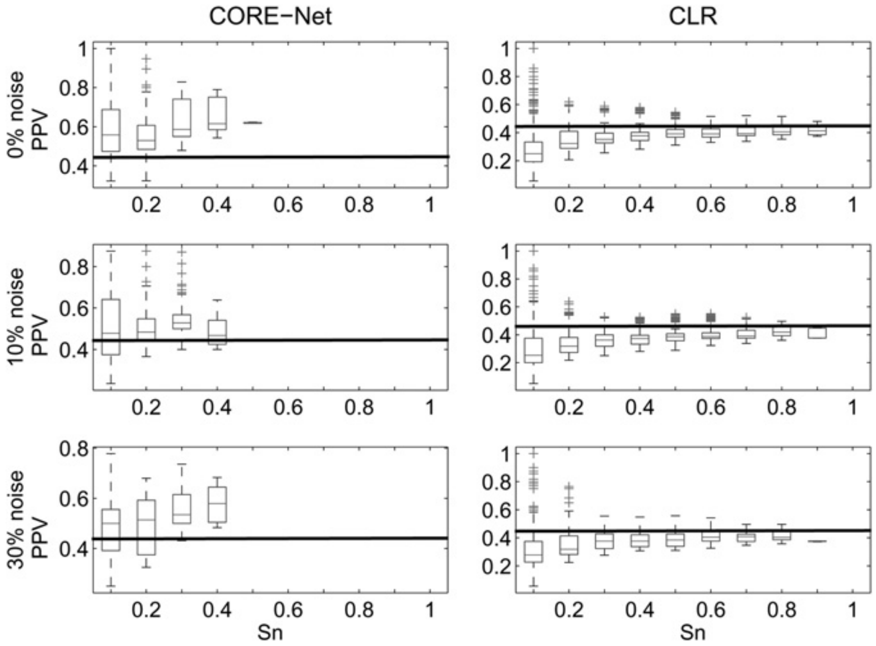

Figure 4 *PPV value at different sensitivity levels, with random scale-free (undirected) networks of 20 nodes with 20 samples, reported in the box-and-whisker form*

Solid line represents the performance of the random inference algorithm

on each row/column of the $\boldsymbol{A}$ matrix varies from 0 to 6 with a mean value of about 2 for the networks of ten nodes and from 0 to 11 with a mean value of about 4 for those of 20 nodes; thus, the average sparsity coefficient, η, is equal to 0.78 in the former case and 0.77 in the latter. Figs. 5 and 6 show that in this case the results of the algorithms with and without PA are comparable and both are better than BANJO.

However, looking at the results of LMI-IP, this seems to offer the best performances in the case of Erdős–Rényi networks, especially in the presence of noise.

With regard to the computational complexity of our technique, on a standard PC endowed with an AMD Athlon 2.14 GHz processor, a single test on a network of

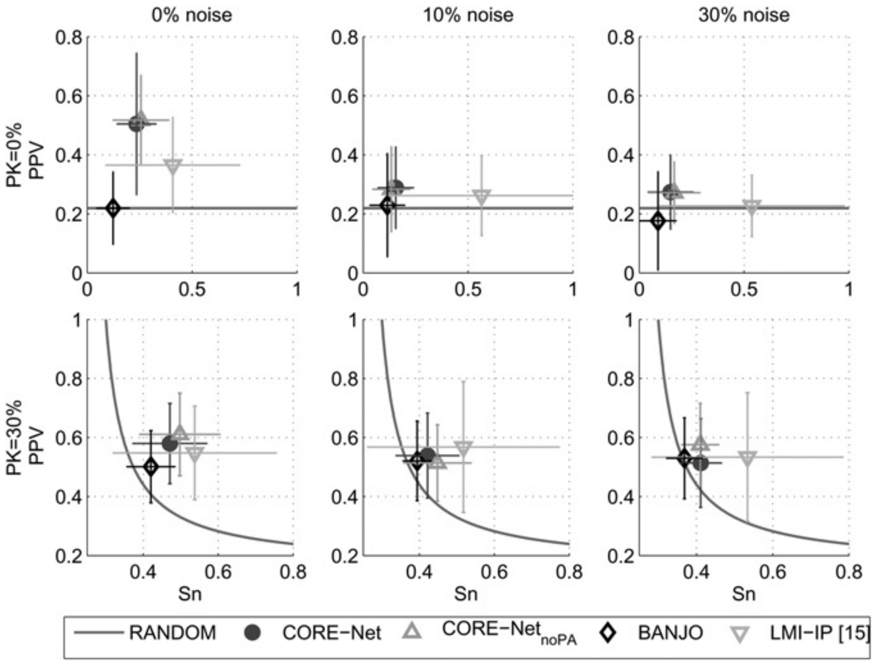

Figure 5 *Results averaged over a set of 20 random Erdős–Rényi networks of ten nodes, using 20 data points*

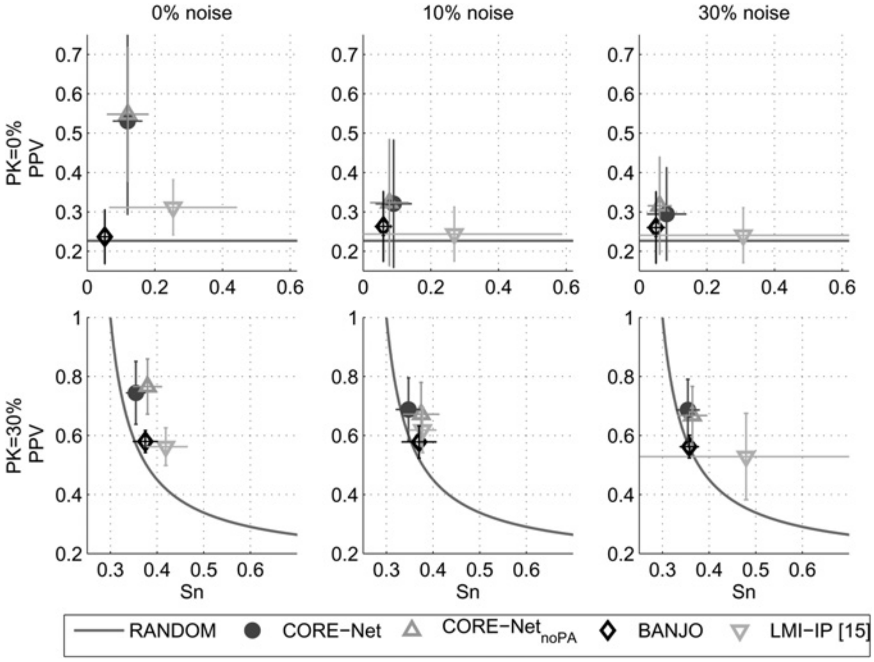

Figure 6 *Results averaged over a set of 20 random Erdős–Rényi networks of 20 nodes, using 20 data-points*

ten nodes takes about a couple of seconds; the computational time increases to about 1 min with a network of 20 nodes.

4.2 Biological case study

In order to assess the applicability of CORE-Net to real biological problems, it has been used to reconstruct a cell cycle regulatory subnetwork in *S. cerevisiae* from experimental microarray data. We have considered the model proposed by [33] for transcriptional regulation of cyclin and cyclin/CDK regulators and the model proposed by [34], where the main regulatory circuits that drive the gene expression program during the budding yeast cell cycle are considered. The network is composed of 27 genes: ten genes that encode for transcription factor proteins (ace2, fkh1, swi4, swi5, mbp1, swi6, mcm1, fkh2, ndd1, yox1) and 17 genes that encode for cyclin and cyclin/CDK regulatory proteins (cln1, cln2, cln3, cdc20, clb1, clb2, clb4, clb5, clb6, sic1, far1, spo12, apc1, tem1, gin4, swe1 and whi5). The microarray data have been taken from

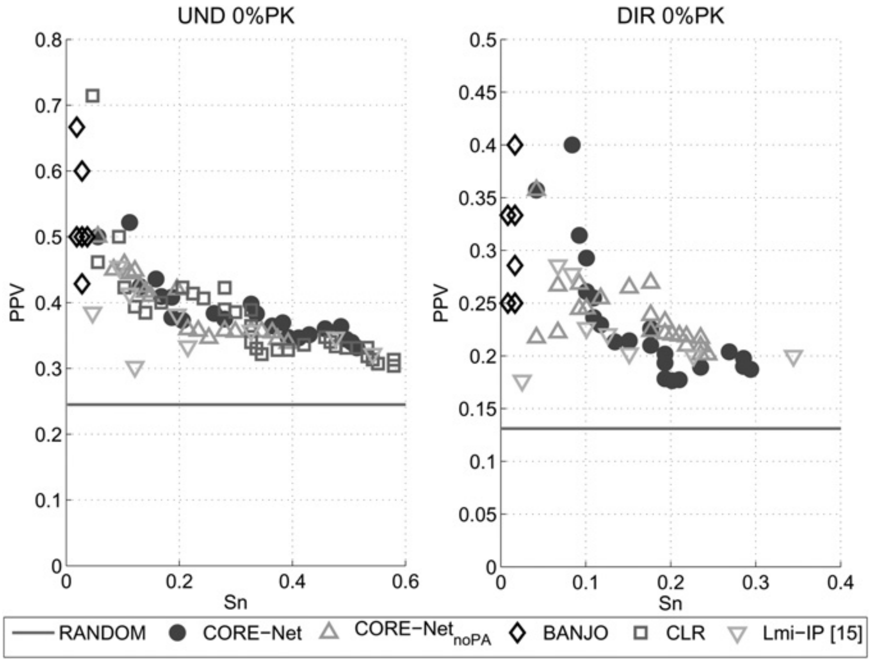

Figure 7 *Results for the cell-cycle regulatory subnetwork of S. cerevisiae obtained by the different techniques, without assuming prior knowledge (PK = 0%)*

[35], selecting the data set produced by alfa factor arrest method. Thus, the raw data set consists of 27 genes and 18 data points. A smoothing algorithm has been applied in order to filter the measurement noise and to increase by interpolation the number of observations, as done for the in silico data sets. The gold standard regulatory network comprising the chosen 27 genes has been drawn from the BioGRID database [36], taking into account the information of [33, 34]: the network consists of 119 interactions, not including the self-loops, yielding a value of $\eta = 0.87$.

Fig. 7 illustrates the results of CORE-Net, CORE-Net$_{noPA}$, LMI-IP, BANJO, CLR and the random inference algorithm, assuming no PK. To allow a comparison with CLR, we have reported the results considering both the case of undirected and directed networks. The performance of the different algorithm is very similar when no prior knowledge is available, except for BANJO, which is not able to achieve significant Sn levels, which is probably due to the low number of time points available.

Fig. 8, instead, shows the results obtained assuming different levels of PK. A comparison between our approach and BANJO is done; in particular, it is evident that the algorithm using the PA heuristic always performs better than the one without PA and BANJO. Moreover, the results show that the performances improve progressively when the level of prior knowledge increases. Thus, the PA seems to yield a significant improvement when some prior knowledge is included in the inference process, which sounds fairly reasonable.

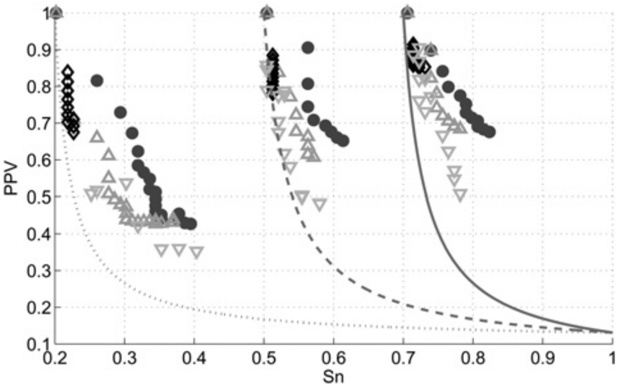

Figure 8 *Results for the cell cycle regulatory subnetwork of S. cerevisiae assuming different levels of prior knowledge (PK = 20, 50, 70%)*

Random reconstruction algorithm with 20% PK (···), 50% PK (- -) and 70% PK (–) and performance of CORE-Net (•), CORE-Net$_{noPA}$ (Δ), LMI-IP (▽) and BANJO (◇)

Finally, Fig. 9 shows the regulatory subnetwork inferred by CORE-Net, assuming 50% of the edges a priori known. Seven functional interactions, which are present in the gold standard network, have been correctly inferred. Moreover, other seven functional interactions have been returned, which are not present in the gold standard network. To understand if the latter have to be classified as TP or FP, we have manually mined the literature and the biological databases, and found the following results:

- The interaction between mbp1 and gin4 is reported by the YEASTRACT database [37]: mbp1 is reported to be a transcription factor for gin4.

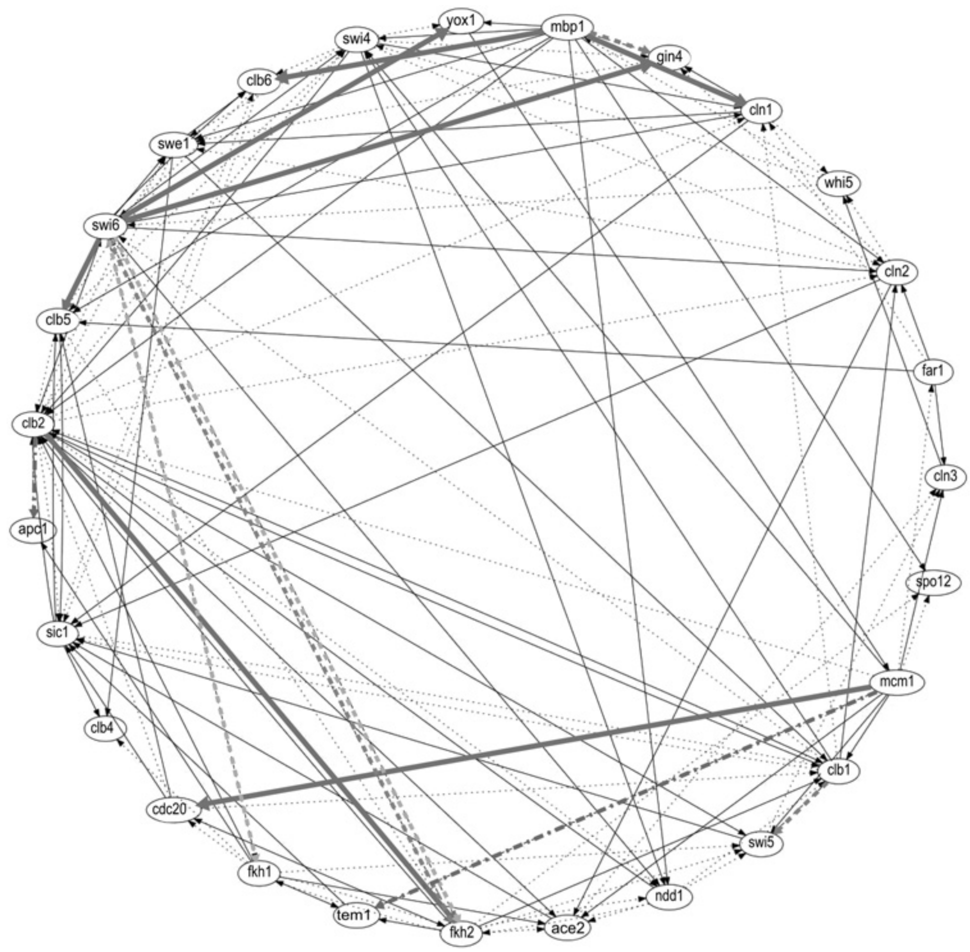

Figure 9 *Gene regulatory subnetwork of S. cerevisiae inferred by CORE-Net with 50% of the edges a priori known (thin solid edges)*

Results according to the gold standard network drawn from the BioGRID database: TP = thick solid edge, FN = dotted edge, FP = thick dashed edge. The thick dashed edges are not present in the BioGRID database; however, they can be classified as TP according to other sources. The FP thick dashed edges are indirect interactions mediated by ndd1. No information has been found regarding the interactions denoted by the thick dash-dot edges

- A possible interaction between fkh2 and swi6 is also reported by the YEASTRACT database: fkh2 is reported to be a potential transcription factor for swi6).

- The interaction between clb1 and swi5 appears in Fig. 1 of [34], where the scheme of the main regulatory circuits of budding yeast cell cycle is described.

Thus, these three interactions can be classified as TP as well and are reported as dashed edges in Fig. 9.

Concerning the other inferred interactions, two of them can be explained by the indirect influence of swi6 on fkh1 and fkh2, which is mediated by ndd1: in fact, the complexes SBF (Swi4p/Swi6p) and MBF(Mbp1p/Swi6p) both regulate ndd1 [33], which can have a physical and genetic interaction with fkh2. Moreover, fkh1 and fkh2 are forkhead family transcription factors, which positively influence the expression of each other. Thus, the inferred interactions are not actually between adjacent nodes of the networks and have to be formally classified as FP (these are reported as dashed edges in Fig. 9).

Concerning the last two interactions, that is clb2 → apc1 and mcm1 → tem1, since we have not found any information in the literature, in the absence of further experimental evidences they have to be classified as FP (reported as dash-dot edges in Fig. 9).

Finally, the results obtained in this example confirm that, also in a real biological case study, the exploitation of PK and of the PA mechanism together significantly improves the inference performances.

5 Conclusions

In the present work, a novel approach to the reverse engineering of functional interaction networks, named CORE-Net, has been presented, with particular emphasis to its application to the biological domain. In particular,

doi: 10.1049/iet-syb.2009.0047

the problem tackled in the present paper consists of identifying the topology of a gene regulatory network starting from the time-series measurements of the expression of each gene. The main feature of the devised approach is the combined exploitation of available prior knowledge and of an edges selection heuristic derived from the scale-free network generation model proposed by Albert and Barabási.

The method has been first statistically validated by means of numerical tests over a set of in silico simulated random networks. Subsequently, it has been applied to a real-world biological case study, that is the reconstruction of a cell-cycle regulatory subnetwork in yeast. The choice of this example was motivated by the large availability of information about the cell-cycle regulatory network of *S. cerevisiae*, which has allowed to assess the results of the inference algorithm.

The results from the numerical tests and the biological case study show that the proposed technique achieves a significant inference capability, outperforming two of the most well-known algorithms available in the literature, namely BANJO and CLR. An important indication arises from the comparison of the results obtained with and without using the PA mechanism; indeed, the exploitation of such a mechanism significantly improves the inference power of the algorithm, especially when PK is taken into account.

Certainly, it can be objected that different networks exhibit different topologies: assuming that the network to be inferred has a scale-free topology can be misleading. In such cases, the domain expert's knowledge can provide valuable insight to guess whether the biological network to be inferred is expected to exhibit a scale-free or Erdős–Rényi topology. Consequently, one can decide to apply methods like the one illustrated in this work, which is best suited for the former class of networks, or alternative approaches, for example, the LMI-IP, which has been shown to provide better performances with the latter class. Finally, the biological case study tackled in this paper suggests that the proposed approach can yield good results in real-world applications.

6 Supplementary material

The source code and the data used to run the tests illustrated in the paper are available at http://bioingegneria.unicz.it/~cosentino.

7 References

[1] FRIEDMAN N., LINIAL M., NACHMAN I., PE'ER D.: 'Using Bayesian networks to analyze expression data', *J. Comput. Biol.*, 2000, **7**, pp. 601–620

[2] STEUER R., KURTHS J., DAUB C.O., WEISE J., SELBIG J.: 'The mutual information: detecting and evaluating dependencies between variables', *Bioinformatics*, 2002, **18**, (Suppl. 2), pp. S231–S240

[3] GARDNER T.S., DI BERNARDO D., LORENZ D., COLLINS J.J.: 'Inferring genetic networks and identifying compound mode of action via expression profiling', *Science*, 2003, **301**, pp. 102–105

[4] BONNEAU R., REISS D.J., SHANNON P., *ET AL.*: 'The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo', *Genome Biol.*, 2006, **7**, (5), p. R36

[5] PRADERVAND S., MAURYA M.R., SUBRAMANIAM S.: 'Identification of signaling components required for the prediction of cytokine release in RAW 264.7 macrophages', *Genome Biol.*, 2006, **7**, (2), p. R11

[6] GUPTA S., MAURYA M.R., SUBRAMANIAM S.: 'Identification of crosstalk between phosphoprotein signaling pathways in RAW 264.7 macrophage cells', *PLoS Comput. Biol.*, 2010, **6**, (1), p. e1000654

[7] CHO K.-H., CHOO S.-M., JUNG S.H., CHOI H.-S., KIM J.: 'Reverse engineering of gene regulatory networks', *IET Syst. Biol.*, 2007, **1**, (3), pp. 149–163

[8] BANSAL M., BELCASTRO V., AMBESI-IMPIOMBATO A., DI BERNARDO D.: 'How to infer gene regulatory networks from expression profiles', *Mol. Syst. Biol.*, 2007, **3**, p. 78

[9] STOLOVITZKY G., PRILL R.J., CALIFANO A.: 'Lessons from the DREAM2 challenges', *Ann. N Y Acad. Sci.*, 2009, **1158**, pp. 159–195

[10] GONÇALVES J., WARNICK S.: 'Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks', *IEEE Trans. Autom. Control*, 2008, **53**, (7), pp. 1670–1674

[11] PALSSON B.Ø.: 'Systems biology: properties of reconstructed networks' (Cambridge University Press, Cambridge, UK, 2006)

[12] ZAK D.E., GONYE G.E., SCHWABER J.S., DOYLE III F.J.: 'Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network', *Genome Res.*, 2003, **13**, pp. 2396–2405

[13] KARNAUKHOV A.V., KARNAUKHOVA E.V., WILLIAMSON J.R.: 'Numerical matrices method for nonlinear system identification and description of dynamics of biochemical reaction networks', *Biophys. J.*, 2007, **92**, (10), pp. 3459–3473

[14] MAURYA M.R., BORNHEIMER S.J., VENKATASUBRAMANIAN V., SUBRAMANIAM S.: 'Mixed-integer nonlinear optimisation approach to coarsegraining biochemical networks', *IET Syst. Biol.*, 2009, **3**, (1), pp. 24–39

doi: 10.1049/iet-syb.2009.0047

[15] COSENTINO C., CURATOLA W., MONTEFUSCO F., BANSAL M., DI BERNARDO D., AMATO F.: 'Linear matrix inequalities approach to reconstruction of biological networks', *IET Syst. Biol.*, 2007, **1**, (3), pp. 164–173

[16] JULIUS A., ZAVLANOS M., BOYD S., PAPPAS G.J.: 'Genetic network identification using convex programming', *IET Syst. Biol.*, 2009, **3**, (3), pp. 155–166

[17] AUGUST E., PAPACHRISTODOULOU A.: 'Efficient, sparse biological network determination', *BMC Syst. Biol.*, 2009, **3**, p. 25

[18] JEONG H., TOMBOR B., ALBERT R., OLTVAI Z.N., BARABÁSI A.-L.: 'The large-scale organization of metabolic networks', *Nature*, 2000, **407**, pp. 651–654

[19] ALBERT R.: 'Scale-free networks in cell biology', *J. Cell Sci.*, 2005, **118**, pp. 4947–4957

[20] LUSCOMBE N.M., BABU M.M., YU H., SNYDER M., TEICHMANN S.A., GERSTEIN M.: 'Genomic analysis of regulatory network dynamics reveals large topological changes', *Nature*, 2004, **431**, pp. 308–312

[21] ALBERT R., BARABÁSI A.-L.: 'Topology of evolving networks: local events and universality', *Phys. Rev. Lett.*, 2000, **85**, (24), pp. 5234–5237

[22] DI BERNARDO D., GARDNER T.S., COLLINS J.J.: 'Robust identification of large genetic networks'. Proc. Pacific Symp. on Biocomputing (PSB'04), Hawaii, USA, January 2004, pp. 486–497

[23] GUTHKE R., MOLLER U., HOFFMAN M., THIES F., TOPFER S.: 'Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection', *Bioinformatics*, 2005, **21**, pp. 1626–1634

[24] KIM J., BATES D.G., POSTLETHWAITE I., HESLOP-HARRISON P., CHO K.-H.: 'Linear time-varying models can reveal non-linear interactions of biomolecular regulatory networks using multiple time-series data', *Bioinformatics*, 2008, **24**, pp. 1286–1292

[25] MEYER C.D.: 'Matrix analysis and applied linear algebra' (SIAM Press, Philadelphia, PA, 2000)

[26] BOYD S., EL GHAOUI L., FERON E., BALAKRISHNAN V. (EDS.): 'Linear matrix inequalities in system and control theory' (SIAM Press, Philadelphia, PA, 1994)

[27] GAHINET P., NEMIROVSKI A., LAUB A.J., CHILALI M.: 'LMI control toolbox' (The Mathworks, Inc., Natick, MA, 1995)

[28] FRANKLIN G.F., POWELL J.D., EMAMI-NAEINI A.: 'Feedback control of dynamic systems' (Prentice-Hall, Upper Saddle River, NJ, USA, 2002)

[29] OLSON D.L., DELEN D. (EDS.): 'Advanced data mining techniques' (Springer, 2008)

[30] YU J., SMITH V.A., WANG P.P., HARTEMINK A.J., JARVIS E.D.: 'Advances to Bayesian network inference for generating causal networks from observational biological data', *Bioinformatics*, 2009, **20**, pp. 3594–3603

[31] FAITH J.J., HAYETE B., THADEN J.T., *ET AL.*: 'Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles', *PLoS Biol.*, 2007, **5**, pp. 54–66

[32] The Mathworks: 'Statistics toolbox' (The Mathworks, Inc., Natick, MA, 2003)

[33] SIMON I., BARNETT J., HANNETT N., *ET AL.*: 'Serial regulation of transcriptional regulators in the yeast cell cycle', *Cell*, 2001, **106**, (1), pp. 697–708

[34] BÄHLER J.: 'Cell-cycle control of gene expression in budding and fission yeast', *Annu. Rev. Genetic.*, 2005, **39**, (1), pp. 69–94

[35] SPELLMAN P.T., SHERLOCK G., ZHANG M.Q., *ET AL.*: 'Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization', *Mol. Biol. Cell.*, 1998, **9**, (1), pp. 3273–3297

[36] STARK C., BREITKREUTZ B.J., REGULY T., BOUCHER L., BREITKREUTZ A., TYERS M.: 'BioGRID: a general repository for interaction datasets', *Nucleic Acids Res.*, 2006, **34**, (1), pp. D535–D539

[37] TEIXEIRA M.C., MONTEIRO P., JAIN P., *ET AL.*: 'The YEASTRACT database: a tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*', *Nucleic Acids Res.*, 2006, **34**, (1), pp. D446–D451

[38] BANSAL M., DI BERNARDO D.: 'Inference of gene networks from temporal gene expression profiles', *IET Syst. Biol.*, 2007, **1**, (5), pp. 306–312

[39] DE BOOR C.: 'A practical guide to splines' (Springer-Verlag, New York, USA, 2001)

[40] DE BOOR C.: 'Spline toolbox' (The Mathworks, Inc., Natick, MA, 2002)

8 Appendix

8.1 Sparsity pattern of the discrete-time model

The devised technique is based on the assumption that the sparsity pattern of the dynamical matrix and input matrix of system (1) can be recovered through the estimation of the corresponding matrices of the associated sampled-data

doi: 10.1049/iet-syb.2009.0047

discrete-time system. Here we want to validate such hypothesis, analysing the relationship between the dynamical matrices of the continuous-time and discrete-time systems.

For the sake of simplicity, in what follows we will assume that $\boldsymbol{A}$ has n distinct real negative eigenvalues, λ_i, $|\lambda_i| < |\lambda_{i+1}|$, $i = 1, \ldots, n$ and is therefore possible to find a non-singular matrix $\boldsymbol{P}$ such that $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$, with $\boldsymbol{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ (the case of non-diagonalisable matrices is beyond the scope of the present work and will not be treated here). Then, the matrix $\boldsymbol{A}_\mathrm{d}$ can be rewritten as ([25, p. 525])

$$\begin{aligned}\boldsymbol{A}_\mathrm{d} &= \boldsymbol{I} + \boldsymbol{A}T_\mathrm{s} + \frac{(\boldsymbol{A}T_\mathrm{s})^2}{2!} + \frac{(\boldsymbol{A}T_\mathrm{s})^3}{3!} + \ldots \\ &= \boldsymbol{P}\mathrm{diag}(\mathrm{e}^{\lambda_1 T_\mathrm{s}}, \ldots, \mathrm{e}^{\lambda_n T_\mathrm{s}})\boldsymbol{P}^{-1} \end{aligned} \quad (8)$$

If the sampling time is properly chosen, such as to capture all the dynamics of the system, then $T_\mathrm{s} \ll \tau_i := 1/|\lambda_i|$, $i = 1, \ldots, n$, which implies $\lambda_i T_\mathrm{s} \ll 1$. Therefore the following approximation holds

$$\mathrm{e}^{\lambda_i T_\mathrm{s}} = \sum_{k=0}^{\infty} \frac{(\lambda_i T_\mathrm{s})^k}{k!} \simeq 1 + \lambda_i T_\mathrm{s}$$

From this approximation and (8), we obtain

$$\boldsymbol{A}_\mathrm{d} \simeq I + \boldsymbol{A}T_\mathrm{s}$$

As for the input matrix $\boldsymbol{B}$, the following approximation holds

$$\boldsymbol{B}_\mathrm{d} = \boldsymbol{A}^{-1}(\boldsymbol{e}^{\boldsymbol{A}T_\mathrm{s}} - I)\boldsymbol{B} \simeq \boldsymbol{A}^{-1}(\boldsymbol{A}T_\mathrm{s})\boldsymbol{B} = \boldsymbol{B}T_\mathrm{s}$$

Note that the sparsity patterns of $I + \boldsymbol{A}T_\mathrm{s}$ and $\boldsymbol{B}T_\mathrm{s}$ are identical to those of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively; only the diagonal entries of $\boldsymbol{A}$ are different, which, however, are always assumed to be free optimisation parameters in our algorithm. What can be concluded from the previous calculations is that, in general, $(\boldsymbol{A})_{ij} = 0$ does not imply $(\boldsymbol{A}_\mathrm{d})_{ij} = 0$; however, one can reasonably expect $(\boldsymbol{A}_\mathrm{d})_{ij}$ to be much lower than the other elements on the ith row and jth column, provided that T_s is much smaller than the characteristic time constants of the system dynamics (the same applies for $\boldsymbol{B}$ and $\boldsymbol{B}_\mathrm{d}$). Such considerations can be readily verified by means of numerical tests.

The algorithm presented in Section 3 is based on these arguments; indeed, it chooses at each step only the largest elements of the (normalised) estimated $\boldsymbol{A}_\mathrm{d}$ and $\boldsymbol{B}_\mathrm{d}$ matrices, and is therefore expected to disregard the entries corresponding to zeros in the continuous-time matrices.

8.2 Generation of simulated data sets

The in silico networks, considered in this paper for benchmarking purposes, are mathematically represented as dynamical linear systems, in the form (1), whose $\boldsymbol{A}$ matrices determines both the dynamics and the topology of the network, as illustrated above. In particular, the matrix is generated such that the out-degree of the associated network exhibits a power law distribution, whereas the in-degree distribution is exponential [19] and the average sparsity coefficient, $\eta = 1 - \#\mathrm{edges}/(n^2 - n)$, is equal to 0.77 for the networks of ten nodes and 0.74 for those of 20 nodes (η is uniformly distributed in [0.65, 0.85]).

8.3 Network topology

The network topologies have been generated by using the scale-free network generation model proposed by Albert and Barabási [21]: the generation procedure starts with a network of n_0 isolated nodes (without links), subsequently one of the following operations is performed:

- With probability p add $m \le n_0$ new links: Randomly select a node i as the end point of the new link, whereas the start point of the link is selected with probability

$$p_j = \frac{K_j}{\sum_{l=1}^{n} K_l} \quad (9)$$

where p_j is the probability of inserting a new edge starting from node j and K_l is the number of outgoing connections from the lth node. This process is repeated m times.

- With probability q rewire m links: Randomly select a node i and an edge $\boldsymbol{A}_{ik}$ pointing to it, then remove the randomly selected link and replace it with with a new link $\boldsymbol{A}_{ij}$, where the new starting node j is chosen with probability p_j. This process is repeated m times.

- With probability $1 - p - q$ add a new node: The new node has m new incoming edges; to each edge the starting node j is assigned with probability p_j.

The iterative procedure terminates when the network has reached the desired size.

8.4 Data generation

Once the network topology has been determined, the associated dynamical system is generated through the following procedure:

- The rss routine in Matlab Control System Toolbox is used to generate a random linear system (the random systems are generated such that the eigenvalues have a normal distribution, see [38]); then $\boldsymbol{A}_{ij}$, for $i, j = 1, \ldots, n$ and $i \neq j$, is set to zero if there is no edge directed from node j to node i in the assigned network topology. Note that $\boldsymbol{A}_{ii} \neq 0$, because all the nodes are assumed to exhibit self-loops.

- In order to obtain a stable system, the eigenvalues are shifted to the left half-plane by subtracting a diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ to $\boldsymbol{A}$. The values λ_i, for

$i = 1, \ldots, n$, are chosen such that the eigenvalues of $\boldsymbol{A} - \Lambda$ have real part values uniformly distributed in the interval $[-2, -0.5]$.

- Concerning the input vector $\boldsymbol{B}$, the simulated experiments consist of perturbing a single node of the network, therefore all the elements of the vector but one, randomly chosen, are nullified.

- The initial condition of each state variable is chosen as a random value, with normal distribution with mean 0 and standard deviation 1.

The in silico data set is then generated by computing the state response of the system and sampling it at $h + 1$ equally spaced time points, in the interval $[0, T_f]$; T_f is the settling time at 5% and a Gaussian noise with distribution $\mathcal{N}(0, \sigma^2)$ is added at each data point. Different values of σ have been used as discussed in the following. When the data are noisy, the cubic smoothing spline technique (see [39, pp. 43–45]) (implemented in the csaps routine in Matlab Spline Toolbox [40]) is applied in order to filter the measurement noise and increase by interpolating the number of observations, before applying the inference algorithms.

doi: 10.1049/iet-syb.2009.0047