Reverse Engineering Partially-Known Interaction Networks from Noisy Data

Francesco Montefusco ∗,∗∗ Carlo Cosentino ** Jongrae Kim ***
Francesco Amato ** Declan G. Bates *

* College of Engineering, Mathematics and Physical Sciences,
University of Exeter, Harrison Building, North Park Road, Exeter
EX4 4QF, UK (e-mail: {F.Montefusco, D.G.Bates}@exeter.ac.uk)
** School of Computer and Biomedical Engineering, Università degli
Studi Magna Gracia di Catanzaro, Viale Europa, Campus di
Germaneto, 88100 Catanzaro, Italy (e-mail: {f.montefusco,
carlo.cosentino, amato}@unicz.it)
*** Division of Biomedical Engineering, University of Glasgow,
Glasgow G12 8QQ, UK (e-mail: Jongrae.Kim@glasgow.ac.uk)

Abstract: One of the most difficult challenges associated with the problem of inferring functional interaction networks from experimental data is that of dealing with the effects of measurement noise in the data used for reverse engineering. A second important challenge is that of taking full advantage of prior knowledge about some elements of the network to improve the results of the reconstruction process. This paper introduces a new inference algorithm, PACTLS, which addresses both of the above issues. The algorithm combines methods to exploit mechanisms underpinning scale-free networks generation, i.e. network growth and preferential attachment (PA), with a technique to optimally reduce the effects of measurement noise in the data on the reliability of the inference results, i.e. the Constrained Total Least Squares (CTLS) algorithm. The technique is assessed through numerical tests on in silico random networks and is shown to consistently outperform approaches based on Bayesian networks.

Keywords: Identification and modelling.

1. INTRODUCTION

The general problem of reverse engineering the topology of functional interaction networks from time-course experimental data has received a considerable attention in the literature, due to the potential applications in the most diverse fields, comprising engineering, biology, economics and social sciences. A number of conceptually different approaches have been proposed in the literature to tackle this problem, focusing on the case of biological interaction networks (Cho et al. [2007], Sima et al. [2009] and references therein). A significant difficulty for all of the proposed approaches is the detrimental effect of measurement noise on the reliability of the inference results (see Cosentino et al. [2007], Bansal and di Bernardo [2007]).

Approaches based on statistical models, such as Bayesian networks and Mutual Information theory, usually also require large data sets and/or assume that the samples are independent. In certain situations, however, only a small number of experimental data points may be available, and the assumption of independent samples is clearly not true when we consider the measurements of the expression of the same gene at two consecutive time-points. For such problems, the family of inference methods that use dynamical systems theory to identify linear models interpolating experimental data, see Yeung et al. [2002], Gardner et al. [2003], di Bernardo et al. [2005], Cosentino et al. [2007], Montefusco et al. [2010], have been shown to be a useful alternative or complement to statistical approaches, especially when the size of the network to be reconstructed is moderate. In the recent literature, several authors have proposed techniques based on regression algorithms for the identification of linear models interpolating experimental data (Chen et al. [1999], Yeung et al. [2002], Gardner et al. [2003], di Bernardo et al. [2004, 2005], Amato et al. [2006]). A common limitation to the practical application of these methods, besides the detrimental effect of measurement noise, is their inability to exploit any prior knowledge which may be available about the network topology. For example, in the field of biology, it is often possible to derive qualitative information about some part of a network from biological journals and databases, of the form “protein A inhibits the expression of gene B”. In contrast to statistical approaches such as Bayesian networks, information of this kind cannot be exploited when using standard regression techniques, since in these approaches an variable can be fixed only if its actual numerical value, as opposed to its sign, is known a priori. In Cosentino et al. [2007], the authors addressed this issue by formulating the inference problem as a convex optimization problem in the form of linear matrix inequalities (LMIs). This formalism allows qualitative information to be directly taken into account within the optimization procedure and achieves a significant inference capability, but the performance of the algorithm was also seen to degrade significantly in the presence of noise in the measurements. Attempts to extend
this LMI-based approach to explicitly take account of the effects of measurement noise have so far been unsuccessful.

In this paper, therefore, we consider an alternative approach, named PACTLS, which is devised to (a) optimally deal with the presence of correlated noise in the measurements, by using the Constrained Total Least Squares (CTLS) algorithm (see Abatzoglou et al. [1991]) and (b) take into account qualitative prior knowledge about the network topology by representing this information as additional constraints for the reconstruction problem.

A key point underlying our approach is the measurements on the structure of the World-Wide Web, Internet, social and biological networks (see Jeong et al. [2000] and references therein, Wagner [2001], Featherstone and Broadie [2002]), show that, independently of the nature of the system and the identity of its constituents, these networks have a small number of hubs and many poorly connected nodes. A plausible hypothesis for the emergence of such a feature, as shown in Albert and Barabási [2000], is the so-called preferential attachment (PA) mechanism during network growth and evolution, where, when a new node is added to the network, it is more likely to be connected with one of the few hubs than with one of the many other loosely connected nodes. In large networks, this evolution rule may generate particular degree distributions, e.g., the well known power-law distribution of scale-free networks.

In our proposed approach, the PA mechanism is combined with the CTLS technique, to produce the PACTLS algorithm. Interestingly, while the CTLS algorithm is now routinely used in advanced signal and image processing applications, its usefulness in Systems Biology applications is only just beginning to be appreciated, see for example Kim et al. [2007], where the method was used to identify the parameters of differential equation-based systems biology models from noisy data.

The proposed PACTLS algorithm is tested on a set of randomly generated in silico networks, with varying levels of noise in the corresponding measurement data and different levels of prior knowledge about the topology of the network, and is shown to consistently outperform approach based on Dynamic Bayesian networks (DBNs). The practical applicability of the PACTLS algorithm has also been demonstrated on a real biological case-study with in vitro experimental data, namely the reverse engineering of a regulatory subnetwork of the cell-cycle in S. cerevisiae (see Supplementary Material).

The paper is structured as follows: Section 2 describes the network model and the generation of in silico networks and data sets. Section 3 illustrates the reverse engineering technique. The results obtained in the numerical tests are reported in Section 4. Finally, conclusions are given in Section 5.

2. METHODS

2.1 Graphs and dynamical systems used for network inference

The interaction network we aim to reconstruct can be described via the formalism of directed graphs (or digraphs). Mathematically a digraph is an ordered pair of sets \( V := (V, A) \), where \( V \) is a set of vertices (or nodes) and \( A \) is a set of ordered pairs of nodes, called links or edges (see Kocay and Kreher [2004], p. 223). If a weight (i.e. a real scalar) is assigned to every edge, the result is a weighted digraph. 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 \( m \) nonzero coefficients. The main goal of a topological inference algorithm, then, is to recover the connectivity matrix.

Looking at the problem from the system theorist point of view, the dynamical evolution of a functional interaction 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 (ODE) in the continuous-time case, or difference equations in the discrete-time case (see Gardner et al. [2003], di Bernardo et al. [2004] and references therein). In this work we consider the continuous-time LTI model

\[
\dot{x}(t) = Ax(t) + Bu(t),
\]

where, \( x(t) = (x_1(t), \ldots, x_n(t))^T \in \mathbb{R}^n \), represents the quantities or levels of the different entities associated to the state variables \( x_i, i = 1, \ldots, n, \) of the system, \( A \in \mathbb{R}^{n \times n} \) is the dynamic matrix and \( B \in \mathbb{R}^{n \times 1} \) is a vector that determines the effect of external perturbations, \( u(t) \in \mathbb{R}^n \).

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 \( A_{ij} \), respectively. In view of these considerations, if we look at the state variables as quantities associated to the nodes of a network, the matrix \( A \) 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 of this approach could be raised with respect to the use of a linear model, which is certainly inadequate to capture the complex nonlinear dynamics of certain networks, like molecular reactions. However, this criticism would only be reasonable if we aimed to identify an accurate model of large changes in the states of a biological system over time; here instead, the goal is to recover only the qualitative functional relationships between the states of the system, which can usually be well represented by a first-order linear approximation.

2.2 Generation of in silico networks and data sets

The in silico networks used in this paper for benchmarking purposes are mathematically represented as dynamical linear systems, in the form of (1), whose \( A \) matrices determines both the dynamics and the topology of the network, as illustrated in Section 2.1. The matrices are generated such that in the associated network there are more nodes with low in-degree (\( k_{in} \)) and out-degree (\( k_{out} \)), and fewer nodes (the hubs) with high \( k_{in} \) and \( k_{out} \). A set of 20 networks of twenty nodes is generated and the average sparsity coefficient, \( \eta = 1 - \text{#edges}/(n^2 - n) \), is equal to 0.74 (\( \eta \) is uniformly distributed in \([0.65, 0.85]\)).

The network topologies have been generated by using the network generation model proposed in Albert and Barabási [2000]. Once the network topology has been
determined, the associated dynamical system is generated through the rss routine in Matlab Control System Toolbox (the random systems are generated such that the eigenvalues have a normal distribution, see Bansal and di Bernardo [2007]); then $A_{ij}$, 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 $A_{ii} \neq 0$, 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 $\Lambda = \text{diag} (\lambda_1, \ldots, \lambda_n)$ to $A$. The values $\lambda_i$, for $i = 1, \ldots, n$, are chosen such that the eigenvalues of $A - \Lambda$ have real part values uniformly distributed in the interval [-2,0.5]. Concerning the input vector $B$, 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 a uniform distribution in [-1,1]. 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]; $T_f$ is the settling time at 5% of the steady–state value and a Gaussian noise with distribution $N(0,\sigma^2)$ is added at each data point. Different values of $\sigma$ have been used as discussed next.

3. THE PACTLS INFECTION ALGORITHM

3.1 Using the CTLS method for network inference

In our approach, the general problem of reverse engineering a functional interaction network from temporal evolution data is tackled via methods based on dynamical linear systems identification theory. The basic step of the inference process consists of estimating, from experimental measurements, the matrix $A$ and the vector $B$ of the linear system (1), which represent the weighted connectivity matrix of the network and the vector that models the effect of external perturbations, respectively. To this end, we will make use of an extension of the classical Least Squares Estimator (LSE), namely the CTLS.

Assume that $h + 1$ experimental observations, $x(\kappa) \in \mathbb{R}^n$, $\kappa = 0, \ldots, h$, are available, then we can recast the problem in the discrete–time domain as $X := (x(0) \ldots x(T)) = \Theta \Omega$, where

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

Since we are dealing with a linear model, it is possible to separately estimate each row, $\Theta_{i,*}$, of the connectivity matrix $\Theta$ to be identified. Let $Z = \Theta^T \in \mathbb{R}^{h \times (n+1)}$, $X_i = (x_i(h), \ldots, x_i(T))^T \in \mathbb{R}^h$ and $\beta = (\Theta_{i,*})^T \in \mathbb{R}^{n+1}$. The problem to be solved in a standard least squares (LS) setting can then be formulated as follows:

$$X_i = Z \cdot \beta. \quad (2)$$

Now, if we assume that the measurements are noisy, the relation (2) can be written in the following form:

$$X_i + \Delta X_i = (Z + \Delta Z) \cdot \beta, \quad (3)$$

where

$$\Delta Z = \begin{pmatrix} 1_{n_0} \ldots n_{n_0} & 0 \\ \vdots & \ddots & \vdots \\ 1_{n_h} \ldots n_{n_h} & 0 \end{pmatrix} \in \mathbb{R}^{h \times (n+1)},$$

and $\Delta X_i$ is the $i$-th noise component of $x_i$, for $i = 1, \ldots, n$ and for $j = 0, \ldots, h$. $\Delta Z$ and $\Delta X_i$ are unknown terms caused by the noise in the data. Although the exact values of the correction terms, $\Delta Z$ and $\Delta X_i$, are not known, the structure, i.e. how the noise appears in each element, is known. If the unknown terms are ignored, then the problem is solved by the standard least squares (LS) method as follows:

$$\beta_{LS} = (Z^T Z)^{-1} Z^T X_i.$$

To write (3) in a more compact form, we make the following definitions:

$$C := (Z \ X_i), \quad \Delta C := (\Delta Z \ \Delta X_i).$$

Then the relation (2) is written as

$$(C + \Delta C) \begin{pmatrix} \beta \\ -1 \end{pmatrix} = 0.$$ 

An extension of the LS algorithm, namely the Total Least Squares (TLS) technique, was developed to solve exactly this problem by finding the correction term $\Delta C$. The (TLS) problem is then posed as follows (Golub and Loan [1980]):

$$\min_{\beta,\delta} ||\Delta C||^2_F \quad \text{s.t.} \quad (C + \Delta C) \begin{pmatrix} \beta \\ -1 \end{pmatrix} = 0,$$

where $||\cdot||_F$ is the Frobenius norm defined by $||A||_F = \sqrt{\text{tr}(AA^T)}$ for a matrix $A$ in which $\text{tr}(AA^T)$ is the trace of the matrix, i.e. the sum of the diagonal terms. When the smallest singular value of $(Z^T Z)$ is not repeated, the solution of the TLS problem is given by:

$$\beta_{TLS} = (Z^T Z - \lambda^2 I)^{-1} Z^T X_i,$$

where $\lambda$ is the smallest singular value of $(Z \ X_i)$. The TLS solution is not optimal when the two noise terms in $Z$ and $X_i$ are correlated, since one of the main assumptions in this method is that the two noise terms are independent of each other. If there is some correlation between them, this knowledge can be used to improve the solution by using the constrained total least squares (CTLS) technique (Abatzoglou et al. [1991]). In the case of our problem (3), the two noise terms are obviously correlated because $Z$ is a function of the noise term from the sampling time $k$ equal to 0 to $h - 1$ and $X_i$ is a function of the noise term from $k$ equal to 1 to $h$. The CTLS formulation can be reduced to the following unconstrained minimisation problem:

$$\min_{\beta} \left[ \beta^T - 1 \right] C^T \left( H_\beta H_\beta^T \right)^{-1} C \begin{pmatrix} \beta \\ -1 \end{pmatrix}, \quad (4)$$

where $H_\beta$ is given in a special form which is a function of the structure of the correction terms due to noise and also of $\beta$ — the details can be found in Abatzoglou et al. [1991] and Kim et al. [2007]. The starting guess for $\beta$ used in the above optimisation problem is simply the value returned by the solution of the standard least squares problem.

The problem to be solved 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 are always strictly greater than the number of variables, that is \( h > n + 1 \), then the problem admits a unique globally optimal solution. In the other case, \( h \leq n + 1 \), the interpolation problem is undetermined, thus there exist infinitely many values of the optimization variables that equivalently fit the experimental measurements. In this case, several expedients that can be adopted: first, it is possible 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 \( h > n + 1 \). 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 variables) is iteratively increased and can be limited to satisfy the above constraint. Finally, the introduction of sign constraints on the optimization variables, derived from qualitative prior knowledge of the network topology (as described below), results in 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 CTLS optimization procedure. Since each element of \( A \) can be interpreted as the weight of the edge between two nodes of the network, this goal can be achieved by constraining some of the optimization variables to be zero and others to be strictly positive (or negative), and using a constrained optimization problem solver, e.g. the nonlinear optimisation function \texttt{fmincon} from the MATLAB Optimization Toolbox, to solve (4). Similarly, we can impose a sign constraint on the \( i \)-th element of the input vector, \( b_i \), if we \textit{a priori} know the qualitative (i.e. promoting or repressing) effect of the perturbation on the \( i \)-th node. Alternatively, an edge can be easily pruned from the network by setting to zero the corresponding entry in the minimization problem.

Note that, since we are starting from a discrete data-set, \( \hat{A} \) and \( \hat{B} \) are not actually the estimates of \( A \) and \( B \) in (1), but rather of the corresponding matrices of the discrete–time system obtained through the Zero-Order-Hold (ZOH) discretization method (Franklin et al. [2002], p. 676) with sampling time \( T_s \) from system (1), that is

\[
x(k + 1) = A_dx(k) + B_du(k),
\]

where \( x(k + 1) \) is a shorthand notation for \( x(kT_s + T_s) \), \( x(k) \) for \( x(kT_s) \), \( u(k) \) for \( u(kT_s) \), and

\[
A_d = e^{AT_s}, \quad B_d = \left( \int_0^{T_s} e^{A\tau} d\tau \right) B.
\]

In general, the sparsity patterns of \( A_d \) and \( B_d \) differ from those of \( A \) and \( B \). However, if the sampling time is suitably small, \((\hat{A})_{ij} = 0 \) implies that \((\hat{A}_d)_{ij}\) exhibits a very low value, compared to the other elements on the same row and column, and the same applies for \( B_d \) and \( B \) (see the Appendix in Montefusco et al. [2010] 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, as described next.

### 3.2 Reconstruction of the network model structure

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

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

**P1** A first matrix, \( A \), is computed by solving (4) for each row, without setting any optimization 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 \( A \) before solving the optimization problem, as explained in the previous subsection. Since it typically exhibits all nonzero entries, matrix \( A \) is not representative of the network topology, but it is rather used to weight the relative influence of each entry on the system’s dynamics. This information will be used to select the edges to be added to the network at each step. Each element of \( A \) is normalized with respect to the values of the other elements in the same row and column, which yields the matrix \( \tilde{A} \), whose elements are defined as

\[
\tilde{A}_{ij} = \frac{A_{ij}}{(\|A_{i\cdot}\| \cdot \|A_{\cdot j}\|)^{1/2}}.
\]

**P2** At the \( k \)-th iteration, the edges ranking matrix \( \tilde{G}^{(k)} \) is computed, by applying the formula

\[
\tilde{G}_{ij}^{(k)} = \frac{|\tilde{A}_{ij}|p_{ij}^{(k)}}{\sum_{i=1}^{n} p_i^{(k)}|\tilde{A}_{i\cdot}|}, \quad p_{ij}^{(k)} = \frac{K_i^{(k)}}{\sum_{i=1}^{n} K_i^{(k)}}
\]

where \( p_{ij}^{(k)} \) is the probability of inserting a new edge starting from node \( j \) and \( K_i^{(k)} \) is the number of outgoing connections from the \( i \)-th node at the \( k \)-th iteration. The \( \mu(k) \) edges with the largest scores in \( \tilde{G}^{(k)} \) 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 smaller numbers of nodes at each iteration. The form of the function \( p(\cdot) \) stems from the so–called \textit{preferential attachment} (PA) mechanism, which states that in a growing network new edges preferentially start from \textit{popular} nodes (those with the highest connectivity degree, i.e. the hubs). By exploiting the mechanisms of network growth and PA, we are able to guide the network reconstruction algorithm to increase the probability of producing a network with a small number of hubs and many poorly connected nodes.

**P3** The structure of nonzero elements of \( \hat{A}^{(k)} \) is defined by adding the entries selected at point P2 to those selected up to iteration \( k-1 \) (including those derived \textit{a priori} information), and the set of inequality constraints is updated accordingly; then Problem 4
for each row, with the additional constraints, is solved to compute $\hat{A}^{(k)}$.

P4) The residuals generated by the identified model are compared with the values obtained at the previous iterations; if the norm of the vector of residuals has decreased, in the last two iterations, at least by a factor $\epsilon_r$ 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 sparsity pattern of $\hat{A}^{(k-2)}$. The factor $\epsilon_r$ is inversely correlated with the number of edges inferred by the algorithm: on the other hand, using a smaller value of $\epsilon_r$ raises the probability of obtaining false positives. By conducting numerical tests for different values of $\epsilon_r$, we have found that setting $\epsilon_r = 0.1$ 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 nonzero elements) of $\hat{B}$ is preassigned at the initial step and the corresponding constraints are imposed in all the subsequent iterations.

4. RESULTS

A statistical evaluation is performed by testing the algorithm against a set of randomly generated in silico networks. The results are compared with one of the most popular statistical methods, dynamic Bayesian networks.

4.1 Performance Indexes

The performance of the algorithms are evaluated by using two common statistical indices (see Olson and Delen [2008], p.138):

- **Sensitivity** ($Sn$), defined as
  \[
  Sn = \frac{TP}{TP + FN},
  \]
  which is the fraction of actually existing interactions (TP:=true positives, FN:=false negatives) that the algorithm infers, also termed Recall, and

- **Positive Predictive Value** (PPV),
  \[
  PPV = \frac{TP}{TP + FP},
  \]
  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 an edge, but only its existence, so the network is considered as a directed graph.

4.2 Inference of in silico networks

In this subsection the performance of the algorithm is assessed by applying it to a set of 20 random networks, each with twenty nodes. Different tests have been conducted on each network, using data sets of 25 samples, two levels of measurement noise, corresponding to $\sigma$ equal to 10% and 30% of the sample value, and assuming three different levels of prior knowledge (PK), 10%, 20% and 60%.

![Fig. 1. Results averaged over a set of twenty random networks, each of twenty nodes, using twenty-five data-points.](image)

The performance of the PACTLS algorithm is compared with a reverse engineering method based on Bayesian networks (BN), a theoretical framework that allows prior knowledge to be taken into account in an elegant and principled manner using probability theory. For our case studies, we used the software BANJO (Bayesian Network inference with Java Objects), a tool developed by Hartemink and coworkers (Yu et al. [2004]), that performs network structure inference for static and dynamic Bayesian networks (DBNs). We ran BANJO by searching for the “Best” DBN: for each network and test, we defined a setting file that describes our data and how the inference has to be performed by BANJO (see Supplementary Material). To reverse engineer the network BANJO employs two search strategies: greedy and simulated annealing searches. For our tests, we used the greedy algorithm and assumed as stopping criteria the maximum time allotted for the search, which is equal to 1 hour. In this way, the computational time allowed for PACTLS and BANJO was made very similar. In terms of computational complexity, indeed, on a standard PC endowed with an AMD Athlon 2.14 GHz processor, a single test of the PACTLS algorithm took approximately 1 hour (the exact prediction of the computational time is not possible, as it depends on the number of iterations, data-points and constraints imposed at each optimization step).

The performance of both approaches is compared in Fig. 1. In order to further validate the inference capability of the algorithms, in the figure we show also the results obtained by a random selection of the edges, based on a binomial distribution: given any ordered pair of nodes, the existence of a directed edge between them is assumed true with probability $p_r$, and false with probability $1 - p_r$. By varying the parameter $p_r$ in $[0, 1]$, the random inference algorithm produces results shown as the solid curves on the (PPV, Sn) plot in Fig. 1. As shown in the figure, the results of the PACTLS algorithm are consistently better than the standard statistical approach. In these experiments we have assumed prior knowledge of 10%, 20% and 60% of the edges, randomly selected, corresponding to about 10, 20 and 58 edges out of 97, respectively. Moreover the results obtained by PACTLS with two different levels of noise
(10% and 30%) are very close, which suggests that the detrimental effects of measurement noise are minimised. The performance of the BDN method, on the other hand, deteriorates significantly as the level of measurement noise is increased, so that for a 30% noise level the results obtained are hardly better than those obtained by the random algorithm.

5. CONCLUSIONS

The results from the numerical tests show that the proposed technique achieves a significant inference capability, consistently outperforming one of the most well known algorithms available in the literature, namely BANJO. In particular, when the level of noise in the measurements increases, the CTLs technique combined with the PA mechanism provides significantly improved predictions with respect to the Bayesian approach.

We believe that the PACTLS algorithm offers significant advantages for network inference, and should be suitable for use as a complement or alternative to standard statistical approaches in many problems. The algorithm is particularly suitable for the inference of moderate sized networks, where there exists a significant level of noise in the measurement data, a reasonable amount of prior knowledge about some aspects of the network’s structure, and a relatively small number of time-series data points.

6. SUPPLEMENTARY MATERIAL

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

ACKNOWLEDGEMENTS

This work was carried out under EPSRC Research Grant: EP/F057016/2.

REFERENCES