Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

Augmented sparse reconstruction of protein signaling networks

D. Napoletani [a,b,*], T. Sauer [b], D.C. Struppa [c], E. Petricoin [a], L. Liotta [a]

[a] Center for Applied Proteomics and Molecular Medicine, George Mason University, Manassas, VA 20110, USA
[b] Department of Mathematical Sciences, George Mason University, 4400 University Drive MS 3F2, Fairfax, VA 22030, USA
[c] Department of Mathematics and Computer Science, Chapman University, Orange, CA 92866, USA

ARTICLE INFO

Article history:
Received 13 November 2007
Received in revised form
19 July 2008
Accepted 21 July 2008
Available online 25 July 2008

Keywords:
Sparse representations
Protein interaction models
Biochemical pathways

ABSTRACT

The problem of reconstructing and identifying intracellular protein signaling and biochemical networks is of critical importance in biology. We propose a mathematical approach called augmented sparse reconstruction for the identification of links among nodes of ordinary differential equation (ODE) networks, given a small set of observed trajectories with various initial conditions. As a test case, the method is applied to the epidermal growth factor receptor (EGFR) driven signaling cascade, a well-studied and clinically important signaling network. Our method builds a system of representation from a collection of trajectory integrals, selectively attenuating blocks of terms in the representation. The system of representation is then augmented with random vectors, and l_1 minimization is used to find sparse representations for the dynamical interactions of each node. After showing the performance of our method on a model of the EGFR protein network, we sketch briefly the potential future therapeutic applications of this approach.

1. Introduction

The problem of reconstructing a network of interacting variables from a small set of data generated by the network itself has attracted considerable attention especially since this problem arises so naturally in genomics, proteomics and more generally system biology problems (see for example Voit, 2000; Chou et al., 2006; Husmeier, 2003; Rogers and Girolami, 2005; Nachman et al., 2004; Gardner et al., 2003). In particular, the ability to reconstruct and identify intracellular protein signaling and biochemical networks is of critical importance in modern biology. However, the ability to dynamically measure and collect enough data from every protein/node within the network is impossible with current methodologies. We sought to develop a mathematical approach to this problem using one of the most well-studied and clinically important signaling networks, the epidermal growth factor receptor (EGFR) driven signaling cascade (Araujo et al., 2005a).

Interestingly, it is widely believed, and proven in some cases, that biological networks are scale free networks, with a few variables (hubs) densely connected to many others and most variables interacting only with a few others (Albert, 2005). Even the hubs do not interact with more than a dozen other variables in most reliable models, so that effectively we can say that these networks are *sparse*, with respect to the total number of all possible connections among variables. Such information can greatly help in reconstructing the network itself, as shown in Gardner et al. (2003), Yeung et al. (2002), and Tegner et al. (2003).

Many current algorithms to reconstruct networks from expression data are based on the application of powerful Bayesian methods after the seminal work in Friedman et al. (2000), but, as noted in Rogers and Girolami (2005) (see also Zak et al., 2002), these methods do not perform well with the limited amount of data that can be generated by microarray technologies. This limitation is especially pertinent for protein expression data. The other widely used approach for network reconstruction is based on parameter estimation of dynamical system models of the networks themselves (Voit, 2000). The fundamental difficulty of such approach is the very large number of parameters and reaction rates that need to be estimated (Chou et al., 2006), and this, again, leads to an inability to work efficiently with the limited data generated by microarrays and time series of expression profiles. Another viable alterative when analyzing microarray data is to simply perform some type of clustering analysis such as hierarchical or K-means clustering (Kaufman and Rousseeuw, 2005), or the recent exemplars clustering technique (Frey and Dueck, 2007). Clustering techniques do not require very large data sets to be applied, but they only identify similarly activated variables, and do not provide a causal understanding of the network structure.

[*] Corresponding author at: Department of Mathematical Sciences, George Mason University, 4400 University Drive MS 3F2, Fairfax, VA 22030, USA.
Tel.: +1 703 993 4269; fax: +1 703 993 1491.
E-mail address: dnapolet@gmu.edu (D. Napoletani).

0022-5193/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2008.07.026

To address the need for specialized network reconstruction methods that can work for the limited data generated by experiments, we restrict our attention in this work to ordinary differential equation (ODE) models of protein signaling networks of the form $\dot{x} = f(x)$, where x is the vector of variables in the system and $\dot{x}$ its componentwise derivative. The plausibility of a dynamical system model for biological networks is a well established approach in the literature (Voit, 2000; Chou et al., 2006). However, exact modeling and parameter estimation for such models is difficult in noisy environments and with small data sets. In addition, biological systems adapt their network structure over time, especially in the presence of diseases. It is likely more effective to search for equivalent, indistinguishable, classes of models (Judd and Smith, 2004) that project to the same network structure, in the sense that they give rise to trajectories that are qualitatively similar and they have similar overall topology of the connections among nodes.

With this general approach in mind, we ask whether the structure of sparse ODE networks can be inferred from a small set of trajectories with different initial conditions generated by the system. We show that, for a specific realistic case of ODE modeling of protein networks, it is possible to expand and adapt ideas from the theory of sparse regression (lasso) and signal reconstruction by l_1 minimization (Tibshirani, 1996; Chen et al., 1998; Hastie et al., 2001, Chapter 3; Donoho, 2006a), to develop a method that reconstructs a significant portion of these networks with good accuracy even in the presence of moderate uniform noise of intensity up to 20% of the maximum values of the trajectories.

Our method builds a system of representation by using a collection of integrals of all given trajectories and by attenuating block of terms in the representation itself. The system of representation is then augmented with random vectors, and l_1 minimization is used to find sparse representations for the dynamical interactions of each node. Augmentation by random vectors is crucial in the context of network reconstruction, since sparsity alone is not able to handle the large error-in-variables in the representation due to possible high noise in the trajectories.

One of the main strengths of our method is the ability to sharply distinguish relevant links, so that the rate of false links that are detected can be made very low. This is important in practice since it is difficult and expensive to follow up and validate experimentally potential links among proteins that are inferred by computational means (Hu et al., 2006).

The paper (Yeung et al., 2002) is a significant antecedent to our work, since in that paper the authors use a hybrid singular value decomposition (SVD) and l_1 minimization to find a sparse linear model that fits oligonucleotide microarray data. The l_1 minimization is used in that paper as a postprocessing of the reverse-engineering performed by the SVD. A similar preconditioning for large models is implemented in the recent paper (Debashis et al., 2008) and applied to several examples including microarray data. In this paper we will show how l_1 minimization methods can be modified to *directly* approach network reconstruction and model identification problems, without any preprocessing, for realistic, very limited sampling of the data and significant noise levels, even when large spaces of nonlinear models are considered. Moreover the results for the EGFR network show that our method can recover the topology of relatively small protein networks. We do not require explicit estimation of the noise level in the trajectories and we do not need multiple trajectories with same initial conditions to estimate the true trajectories in the presence of noise.

In Section 2 we show how to apply l_1 minimization methods to reconstruct sparse ODE networks, stressing the specific steps that are necessary in the network setting. In Section 3 we apply the algorithm presented in Section 2 to the EGFR model as described in Araujo et al. (2005a). We choose this network, strongly related to cell proliferation, because it plays a significant role in cancer development (Lacouture, 2006), so it is considered an ideal target for fine tuned potential therapies that do not impact the body at the systemic level. We will briefly mention some possible directions of research related to the medical applications of network mapping at the end of the paper.

2. Methods

2.1. Sparse signal processing

Suppose we have a discrete function $F(n)$, $n = 1, \ldots, N$ and a collection of functions $\mathscr{G} = \{g_1(n), \ldots, g_M(n),\ n = 1, \ldots, N\}$ with $M \geqslant N$. Then in general the representation of F in terms of $\mathscr{G}$ will not be unique, meaning that there will be many ways to write F as $F(n) = \sum_{m=1}^{M} a_m g_m(n)$, $n = 1, \ldots, N$. An important question when trying to extract the significant features of F with respect to $\mathscr{G}$ is to find, among the many possible representations for a signal, the one that is the most sparse, i.e. the representation that has as many zero coefficients a_m as possible. This problem is in general very difficult, but we can use linear programming techniques to find approximate sparse representations, that is, representations that have just a few large coefficients and many very small ones.

We briefly introduce this type of approximation to sparse solutions here following mostly Mallat (1998, Section 9.5.1), and we refer to Tibshirani (1996), Chen et al. (1998), Hastie et al. (2001), and Donoho (2006a, b) for a thorough analysis of the relations between l_1 optimization and sparsity. The key idea is to realize that if we minimize the 1-norm of the coefficients $|a| = \sum_{m=1}^{M} |a_m|$, this implies that the total energy of the coefficients is concentrated in just a few of them. We can gain an intuition on this by noting that a minimization of the 1-norm reduces cancellations among different elements of $\mathscr{G}$, since these cancellations increase the 1-norm.

Note that the problem $\min(\sum_{m=1}^{M} |a_m|)$, subject to $F(n) = \sum_{m=1}^{M} a_m g_m(n)$, $n = 1, \ldots, N$, is equivalent to the problem $\min(\sum_{p=1}^{2M} x_p)$, subject to $F(n) = \sum_{p=1}^{M} x_m g_m(n) - \sum_{p=M+1}^{2M} x_m g_m(n)$, with $x_p > 0$ for every $p = 1, \ldots, 2M$ and $x_p - x_{p+M} = a_p$. The linear optimization problem defined by the last two equations can be easily put in the standard format of linear programming problems, so that a solution can be quickly obtained using one of several algorithms as those described in Lustig et al. (1994) and Zhang (1998).

Given therefore a discrete signal of length N and a collection of M signals with $M \geqslant N$ we can easily find approximate sparse representations for F in the collection of signals. This result, first exposed in Chen et al. (1998), was the inspiration of a series of works that showed the great potential of l_1 minimization in signal processing, see for example the recent work in Candes and Tao (2004) and Candes et al. (2006). Regression using l_1 optimization has also been used extensively and to great effect in statistical learning for model identification under the name of lasso, after the pioneering work in Tibshirani (1996), see Hastie et al. (2001, Chapter 3) for a up to date review of use of the technique.

We will see in the next subsection the crucial adjustments that are required to make l_1 optimization effective and robust in network reconstruction problems.

2.2. Augmented sparse networks

Let us now write explicitly the general form of the dynamical systems of interest. We are assuming that an experiment provides noisy observations for N variables $x_1(t), \ldots, x_N(t)$, each measured at B time points $t_1, \dots, t_B$ (not necessary identical). Finally we assume we are given a series of R experiments. Thus, we have $N \times B \times R$ data points. The goal is to provide an algorithm for reconstructing the network structure of an autonomous system of ODEs for $x_1(t), \dots, x_N(t)$.

We measure noise levels in each time series of observations as follows: assume bounded noise, let m be the absolute value of the maximum displacement of the observations from the true trajectory, and let M be the maximum of the values of the true trajectory, then the noise level is defined as m/M, i.e. we consider the relative intensity of noise with respect to the true trajectory.

Signaling networks arising from protein interactions, even though nonlinear, are often modeled with differential equations that contain simple analytical forms that contain power function terms of variables $x_1, \dots, x_N$ of the type $x^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$, and hyperbolic terms of the type $x_i/(H + x_i)$, that take into consideration the presence of slow enzymatic kinetics (see Voit, 2000; Araujo et al., 2005a, and references therein). In this paper we assume for simplicity that the right-hand side of the dynamical system that we try to model has polynomial terms up to degree $d = 2$, and hyperbolic terms $x_i/(H + x_i)$, $H > 0$. Roughly speaking this modeling makes the assumption that linear terms correspond to uni-molecular interactions, quadratic terms correspond to bimolecular interactions, and hyperbolic terms correspond to Michaelis–Menton reactions.

Remark 2.1. We sample H at uniform intervals of length $\mathrm{d}H$ in a range $[0, S\mathrm{d}H]$ of interest with S some large positive integer. Assuming some fine sampling $\mathrm{d}H$, the proper choice of S would in principle depend on the magnitude of the trajectories, since when $\mathrm{d}HS$ is large compared to the maximum magnitude of the observations, the corresponding Michaelis–Menton term effectively will behave as a linear term, and therefore it will be redundant with one of the linear terms of the model. This implies that the right choice of S will depend on the dynamics of each variable, whoever in this study we do not specialize the choice of S to each variable, and we just assume that $\mathrm{d}HS$ is big compared to the magnitude of all variables, so that, for $s < S$ small enough, we have a properly hyperbolic term for each variable node.

If we denote by $\dot{x}_i$ the time derivative of x_i, we consider models of the form:

$$\dot{x}_n = a_0 + \sum_{i=1}^{N} l_i x_i + \sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij} x_i x_j + \sum_{i=1}^{N} \sum_{s=1}^{S} h_{is} \frac{x_i}{s\mathrm{d}H + x_i}, \tag{1}$$

where $n = 1, \dots, N$, and a_0, l_i, q_{ij}, h_{is} are parameters to be determined. We can in principle consider a model that contains on the right-hand side all monomials $x_i^{\alpha_i}$, all binomials $x_i^{\alpha_i} x_j^{\alpha_j}$, all the way to $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$, where the exponents have norm $|\alpha_i|$ less than a constant A and we assume a uniform sampling of the exponents. This would be a general setting compatible with the modeling approach taken in Voit (2000). However, the main complication of such general models is already present in our quadratic model with hyperbolic terms: when we have many nodes in the network, the combinatorial explosion of terms makes parameter fitting very difficult in the case only a limited amount of noisy data is available on the dynamics of each node.

We also assume sparsity of the network, i.e. *we define a network as sparse if each node interacts with only a small number of nodes compared with the total of possible nodes*. Note that this assumption implies that the number of terms in each equation in (1) with non-zero coefficients is also small compared to the total possible number of terms.

Given this assumption, a possible way to approach the fitting problem implicit in Eq. (1) is to find the model that minimize the l_1 norm of the parameters of the terms in the equation. From the background material summarized in the previous subsection, we know that l_1 optimization leads to a sparse representation of signals with very few terms with non-zero parameters, and that the optimization itself can be performed with linear programming techniques (Chen et al., 1998).

Since we noted in the introduction that actual biological networks seem to satisfy the sparsity assumption, the l_1 fitting method should, in principle, improve our ability to find the actual links among nodes. Exact parameter fitting is difficult in this case as well and we will see in the Results section that direct application of the l_1 fitting as used in signal processing leads to very poor results.

As noted at the beginning of the section, we assume that we sample variables $x_1, \dots, x_N$ at B points and that we have several trajectories $x_{n,r}$, $r = 1, \dots, R$, with R different initial conditions. We denote by $\dot{x}_1, \dots, \dot{x}_N$ the respective derivatives at each of the sampled points. If we write $X_n = [x_{n,1}, \dots, x_{n,R}]$, $\dot{X}_n = [\dot{x}_{n,1}, \dots, \dot{x}_{n,R}]$, and we denote by J the unit vector of same length as X_i, a formal substitution in Eq. (1) of x_n with X_n and $\dot{x}_n$ with $\dot{X}_n$ leads in effect to a problem of representation of discrete signals $\dot{X}_n$ in terms of the collection of signals $\mathscr{X} = \{J, X_i, X_j X_k, X_i/(s\mathrm{d}H + X_i)\}$ with $i, j, k = 1, \dots, N$, $s = 1, \dots, S$.

Remark 2.2. The total number of terms in $\mathscr{X}$ will be $M = N + N(N+1)/2 + NS$, since we have N linear terms, $N(N+1)/2$ distinct quadratic terms and NS hyperbolic terms. The length of each vector in $\mathscr{X}$ will be instead equal to$V = N \times B \times R$ where we recall that B is the number of data points in each time series and R is the number of distinct experiments (i.e. the number of different initial conditions).

This way of stating the problem of reconstructing a specific network of the form (1) highlights the potential of applying the l_1 sparsity techniques to recover the effective system from a collection of different trajectories. However, a direct application of l_1 optimization to the network data will not work in the presence of high noise and for very limited data. As much as sparsity is a powerful device to explore signal representations, it is not able by itself to deal with the large error-in-variables in the representation generated by the system trajectories when they are noisy. There are some crucial modifications that are necessary to get useful reconstruction results on protein networks, we term them model augmentation, attenuation of blocks of terms, and integral modeling. They lead to the following alternative model of the network structure at a specific node:

$$x_n(t) - x_n(t_0) = a_0 + \sum_{i=1}^{N} l_i \int_{t_0}^{t} x_i \,\mathrm{d}t + \beta_q \sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij} \int_{t_0}^{t} x_i x_j \,\mathrm{d}t + \beta_h \sum_{i=1}^{N} \sum_{s=1}^{S} h_{is} \int_{t_0}^{t} \frac{x_i}{s\mathrm{d}H + x_i} \,\mathrm{d}t + \sum_{g=1}^{G} w_g n_g, \tag{2}$$

where β_q and β_h are positive attenuation coefficients for quadratic and hyperbolic terms, both smaller than 1 and where n_g, $g = 1, \dots, G$, are discrete random vectors normally distributed, scaled to have norm 1 and multiplied by suitable coefficients w_g to be determined. We outline now why these adjustments are necessary for a successful application of l_1 sparsity methods.

Integral modeling: First of all we note that to go from Eqs. (1) to (2) we actually formally integrated the differential equations. We need to do this because in a realistic reconstruction setting we have few sample points and a relative noise that can be as high as 0.2 (i.e. 20%) of the maximum norm of the measured trajectories, making the estimation of the derivatives very difficult.

This problem transcends the specifics of our approach and is a key issue in the study of experimentally generated time series.

Note that to use l_1 optimization, we clearly do not need to use only local differential information. This integral representation avoids the implicit problem of finding a good estimation of the derivative from a limited number of samples of the trajectories. We can easily estimate multiples of the integrals on the right-hand side of Eq. (2) by summing up the samples that are given from t_0 to t, if sampling is uniform. If sampling is not uniform, which is very often the case for experimental data, we can scale the contribution of each summand multiplying by the size of the corresponding sampling interval.

The relative noise in the measurement of $x_n(t) - x_n(t_0)$ is comparable with the relative noise of the time series x_n itself when t is far from t_0. Moreover, for biological signals derived from proteomics and genomics, variables often represent intensity or concentration profiles that always assume positive values and therefore, in these cases, we expect the integrals on the right-hand side to be dominated by the integrals of the true values of the variables, when zero mean noise is added to them. Note that the constant term a_0 was used simply as a term to correct potential biases in (1), as it does not carry information on the nodes' links, so we use it similarly in (2) and we do not take its integral.

Model augmentation with random terms: The second very important modification that is apparent in Eq. (2) is the addition of discrete random vectors n_g, $g = 1, \ldots, G$, normally distributed, scaled to have norm 1.

Consider that the main issue that prevents an accurate reconstruction of the network is the presence of noise in the trajectories, what is often labeled as errors-in-variables (Voss et al., 2004). Noisy data are a problem especially when trying to fit a nonlinear model, since in this case large spurious terms are generated by nonlinear terms of the model. The representation system needs to account for such terms when fitting the models on the noisy data.

A simple example that shows how errors in the trajectories affect modeling is the following: denote the noisy measurements of X_i and X_j as $\tilde{X}_i = X_i + N_i$ and $\tilde{X}_j = X_j + N_j$, respectively, and assume that the differential model (1) includes a term X_iX_j in the representation of some $\dot{X}_n$ (the argument can be easily translated in the integral representation of Eq. (2)). Because we need to represent $\dot{X}_n$ with terms generated by the noisy trajectories, we want the term $\tilde{X}_i\tilde{X}_j$, to appear in the sparse representation with large non-zero coefficient. However, $\tilde{X}_i\tilde{X}_j = X_iX_j + X_iN_j + X_jN_i + N_iN_j$, that is, there are several spurious terms in $\tilde{X}_i\tilde{X}_j$, besides the effective term X_iX_j, that are not dynamically related to $\dot{X}_n$.

By augmenting the representation with a large number of random terms, we increase the chance that the noisy residue $X_iN_j + X_jN_i + N_iN_j$ is spread uniformly among many random terms that do not carry information on the potential links among nodes, so that the l_1 optimization will show only one large coefficient associated to $\tilde{X}_i\tilde{X}_j$, and many very small coefficients w_g associated to the random terms that do not carry information about the network. We want the number G of random terms to be much larger than the number V of data points available for each variable so that the energy of potential noisy residues like $X_iN_j + X_jN_i + N_iN_j$ is likely to be uniformly distributed among the parameters w_g of all the random vectors n_g, and therefore the overall contribution to the l_1 norm of these noisy residues is small.

Remark 2.3. Note that G is dependent on the particular instance of problem that is given, and more specifically on the type and number of trajectories and sample points in each trajectory, but the performance of the method we describe in this section is not strongly dependent on its specific value, as long as $G \gg V$, where V is the total number of data points available for each variable. As a strong numerical evidence of this fact, we will see in the next section, when analyzing a model of the EGFR network, that adding random terms tremendously increases the quality of the reconstruction of the network, see in particular Remark 3.3 and Fig. 5(b).

The augmentation of the basic model has far reaching consequences, since it assures that the new models are large enough to be able to perform an approximate sparse minimization, strongly retaining the dependence from the original terms of the 'effective', non-random model, while diffusing any potential noise in the data among the random terms of Eq. (2). The non-random portion of the matrix derived from the ODE network itself can be very ill conditioned. In particular, the hyperbolic terms generated by the same variable will be highly correlated among each other. There is an intrinsic inability to fully control the representation matrix generated by the trajectories and the error in variables that are bound to appear when trajectories are very noisy. This is a distinct characteristic of ODE reconstruction networks and one that makes this work diverge in methodology and outlook from standard l_1 signal reconstruction.

Attenuation of block of terms: Eq. (2) displays coefficients β_q and β_h, respectively, for all quadratic and all hyperbolic terms in the system. The reason for these extra coefficients is that we seek to have reconstructed models with low complexity, that is, models that, given two comparably good representations, will select the one with more terms of low degree. Given this requirement on the models, it is useful to enforce a way to explicitly diminish the influence of the terms belonging to more complex blocks of terms such as quadratic and hyperbolic ones, since the large number of quadratic and hyperbolic terms increases the chance, in a noisy setting, that several wrong terms from these blocks are selected in the representation of each node. Essentially, attenuation coefficients are one more device to keep the errors-in-variables from generating false links in the computed representation of each node, assuming that higher degree and higher complexity terms appear with frequency comparable to linear terms in the representations.

The partial suppression of these higher complexity terms can be done by using suitable attenuation coefficients. More specifically, we choose to attenuate uniformly all terms in a block by a factor $0 < \beta < 1$. Assuming that all terms were scaled to have l_2 norm equal to 1, we effectively multiply the inner product of any signal with each term in the attenuated block by $1/\beta$, which is bigger than 1, so the l_1 optimization will have the tendency to select fewer of the terms in this attenuated block for the representation with the minimal l_1 norm. This is another interesting point specific to the modeling of networks. The need of some attenuation is especially strong when we want very few selected false links and the trajectories are very noisy.

Note that quadratic terms will still be well represented in the l_1 representation, as it should be, since we know that biologically motivated systems will likely have quadratic terms when two molecules are interacting. We think of attenuation as a way to impose that the number of linear, quadratic and hyperbolic terms in the reconstruction are kept roughly equal, regardless of the potentially very different number of terms in each block.

Empirically, we find that the attenuation of quadratic and hyperbolic terms is important for obtaining the very best results in the reconstruction of the geometric structure of the network, for example an attenuation $\beta_q = \beta_h = 0.5$ for both quadratic and hyperbolic terms was near optimal for the EGFR network (see Fig. 6). The need of some attenuation is especially strong when we want very few selected false links and the trajectories are very noisy. The accuracy of each representation for different values of β_q and β_h can be compared with the other representations and therefore there is a way to make an online selection of these coefficients. We find that a wide range of small values of β_q and β_h give similar reconstruction results. See Remark 3.4 for a more detailed analysis of the use of attenuation coefficients in our case study.

2.3. Network reconstruction algorithm

The observations in the previous subsection can be gathered into a simple reconstruction algorithm based on l_1 optimization, which we call augmented sparse reconstruction. We label the variables involved in a slightly different way in the algorithm to highlight the flexibility in the choice of the input for the algorithm. Given trajectories from a sparse system that is believed to be of a certain generic form, for each trajectory $X_{n,r}$, $r = 1, \ldots, R$ discretely sampled at B points, let $\bar{X}_{n,r}$ be the vector $X_{n,r}(t) - X_{n,r}(t_1)$ where t takes all B sampled values, $\bar{X}_{n,r}$ corresponds to evaluating the vector of differences on the left-hand side of Eq. (2) for each variable. Moreover, for a given vector $g(t)$, $t = t_1, \ldots, t_B$, let $I(g)$ be the vector whose l-th component is the sum $\sum_{i=1}^{l} g(t_i)$, and let J denote the unit vector. The basic process to identify the nodes is the following:

A. Suppose we are given N node variables and that for each variable it is possible to generate R trajectories $X_{n,r}$, $r = 1, \ldots, R$, with different initial conditions, uniformly sampled at B points. Write $Y_n = [\bar{X}_{n,1}, \ldots ,\bar{X}_{n,R}]$, $L_i = [I(X_{i,1}), \ldots, I(X_{i,R})]$, $i = 1, \ldots, N$, $Q_{ij} = [I(X_{i,1}X_{j,1}), \ldots, I(X_{i,R}X_{j,R})]$, $i,j = 1, \ldots, N$ and $H_{js} = [I(X_{j,1}/(s\mathrm{d}H + X_{j,1})), \ldots, I(X_{j,R}/(s\mathrm{d}H + X_{j,R}))]$, $j = 1, \ldots, N$, $s = 1, 2, \ldots, S$ and where dH is the sampling interval for the hyperbolic terms. For each $n = 1, \ldots, N$:

B. Choose an attenuation coefficient β_q for the quadratic terms and another one, β_h, for the hyperbolic terms. Let n_g, $g = 1, \ldots, G$, be discrete random vectors normally distributed scaled to have norm 1. Denote by $||$ the 2-norm of a vector and let $\bar{L}$ be the matrix whose columns are all the vectors of linear terms $L_i/|L_i|$, $\bar{Q}$ be the matrix whose columns are all vectors of quadratic terms $Q_{ij}/|Q_{ij}|$ and $\bar{H}$ be the matrix whose columns are all allowed hyperbolic terms $H_{is}/|H_{is}|$. Let N_G be the matrix whose columns are the random vectors n_g scaled to have norm 1. Choose G large enough to have the matrix $Z = [J, \bar{L}, \beta_q \bar{Q}, \beta_h \bar{H}, N_G]$ with small condition number (say less that 10^2).

C. Let α be the vector of parameters to be fitted in Eq. (2). Find the minimal l_1 solution to the (generally underdetermined) system $Y_n = Z\alpha$, using for example LIPSOL (Linear Interior Point Solver) as described in Zhang (1998).

D. Choose a threshold T_n and let α_{T_n} be the vector whose components are equal to those of α if the corresponding components of α are bigger than T_n and zero otherwise. Let $\mathscr{I}_n$, the estimated set of directed links of node n, be the union of all node indexes that appear in terms of Z corresponding to non-zero components of α_{T_n}.

Basically in step A we use the sampled trajectories to estimate the integrals in the representation on the right-hand side of Eq. (2) from available data, and we string them into long vectors, for example the collection of all estimated integrals of the quadratic term x_1x_2 will be represented by Q_{12}, similarly integrals of linear terms will correspond to some L_i and so on.

In step B we scale the vectors of the representation to have norm one, we attenuate the portion of the representation matrix derived from quadratic and hyperbolic terms using attenuation coefficients β_q and β_h, and we augment the matrix of the model with scaled random terms to get an augmented matrix of representation Z.

In step C we apply l_1 minimization in the way outlined in Section 2.1 to the system $Y_n = Z\alpha$. Modulo scaling by the norm of the corresponding linear, quadratic and hyperbolic terms in the representation matrix Z, we can think of the components of the vector α as a_0, the parameter of the constant term, all the parameters l of the linear terms in (2), all the parameters q of quadratic terms, all the parameters h of hyperbolic terms and all the parameters w of random terms in Eq. (2). The specific l_1 algorithm that we use is LIPSOL, described in Zhang (1998).

In step D we identify the largest parameters in the representation, since the assumption is that these parameters correspond to the terms in Eq. (2) that significantly affect the dynamic of each node. Finally, by collecting all the indexes of terms with large parameters, we estimate the set of links that determine the dynamics of each node.

Remark 2.4. While it is true that the hyperbolic terms add a lot of parameters to the model and therefore they contribute significantly to the complexity of the problem, the basic difficulty of fitting the model with scarce very noisy data is significant also when we select just a few hyperbolic terms, in this case the l_1 optimization will be in general unfeasible, but the linear programming algorithm we use (LIPSOL) can find a solution by minimizing the constraints that are violated to the largest extent. Even in this case, the l_1 method avoids spreading the norm of the representation on the whole set of parameters, as it would do, for example, a quadratic regression. This is partially related to the fact that l_1 optimization for overdetermined systems of equations can be seen in many problems as a powerful error correcting method (Candès et al., 2005). In Fig. 6 we show how our method would perform for a variety of models, including the case when hyperbolic terms are fully suppressed.

Remark 2.5. The choice of the threshold in step D is very delicate and it is explored in depth computationally in the next section for the EGFR signaling network that we study. Our approach is to select a family of thresholds that is consistent with the fact that the size of the threshold must be relative to the size of the parameters used to represent each node. We define therefore a threshold, for each node, as a constant multiple of the standard deviation of the non-zero parameters of the non-random terms of each node, what we may call the deterministic parameters of the representation (in practice we neglect any parameter with norm smaller than 10^{-10}). Formally we define $T_i = K\sigma_i$, for each node $i = 1, \ldots, N$, where K is some fixed constant determined for the whole system, while σ_i is the standard deviation of the absolute value of the deterministic non-zero parameters of the representation of node i. This flexible definition of the threshold ensures that: (a) the threshold level is relative to the norm of the parameters of each node; (b) the threshold is larger if there are many sizeable non-zero parameters in the representation of a specific node. The main advantage of a uniform definition of threshold across all variables is that we need the proper estimation of a single threshold multiplier K, and we have the whole reconstruction data available to do that. If the network has very distinct behavior for different subsets of nodes, it may not be possible to use a single multiplier and we must resort to thresholds estimated for each node separately. The specific value of K must be in practice inferred form the data, and we device an heuristic for this purpose that is introduced at the end of Section 3.3 and fully described in Appendix A.

Remark 2.6. The emphasis in this paper is in reconstructing the network structure of the system of ODEs, but our algorithm can be used also to get a rough estimate of the norm of the parameter values of the leading terms in the differential equation (1). Suppose for example that, for a scaled quadratic term $Q_{ij}/|Q_{ij}|$ in the matrix of representation Z, the algorithm finds a parameter $\bar{q}_{ij}$ by using an attenuation coefficient β_q. Then this quadratic term gives a contribution to the overall representation of Y_n, the vector of all integrals of trajectories of $\dot{x}_n$, determined by $\bar{q}_{ij}\beta_q Q_{ij}/|Q_{ij}|$. This term is almost identical to the term we seek in Eq. (2), evaluated at each time sample of each trajectory of the node, except for the scaling by the norm of Q_{ij}. If we remember that by formally taking the derivative of Eq. (2) with respect to time we get Eq. (1), we see that we can recover the true parameter q_{ij} in the term $q_{ij}x_i x_j$ of the right-hand side of Eq. (1) by rescaling $\bar{q}_{ij}$ as $q_{ij} = \bar{q}_{ij}\beta_q/|Q_{ij}|$. An identical scaling procedure can be done for all leading linear and hyperbolic terms. However, in this way we get only an estimation by defect, since a significant portion of the norm of the parameter will be lost due to interference of the random terms in the representation.

We stress again that by no means we need to limit ourselves to linear, quadratic and hyperbolic terms. General power function expansions or higher degree polynomial terms are possible within the frame of this method, since random terms and l_1 minimization keep the reconstruction stable even for very underdetermined systems of representation. Note finally that the algorithm does not change in structure as we change the model, for example we could consider just linear and quadratic terms and the only difference would be to reduce the size of the representation matrix in step B.

3. Results

In this section we show the performance of the augmented sparse reconstruction method A–D on the EGFR protein network described in Araujo et al. (2005a) and explicitly shown in Appendix B, where each variable is labeled with the corresponding protein node. In Section 3.1 we give the details of the set up of the simulations we perform, in Section 3.2 we show the result of applying the algorithm at a single node. In Section 3.3 we work under the assumption that we can choose the multipliers for the thresholds introduced in Remark 2.5 so that the false positives rate is kept very low. In this way we are able to estimate the ideal performance of the algorithm and offer a proof of concept for the method. Finally we show that an heuristic argument allows to determine, from the set of all node representations provided by steps A–C, suitable thresholds that give true positive rates close to the ideal setting, the details of this heuristic are provided in Appendix A.

3.1. A model of the EGFR network: simulations and set up of the augmented sparse reconstruction

The EGFR network is one of the most well-studied and clinically important signaling networks in biology today and the ability of our method to reconstruct a model of such fundamental network is very promising. A portion of the EGFR network has been modeled in Araujo et al. (2005a) as a system involving only linear, quadratic and hyperbolic terms, so the general model in Eq. (1) (and therefore in Eq. (2)) is ideally suited for its analysis, however, the right-hand side of Eqs. (1) and (2) has a very large number of quadratic and hyperbolic terms for the EGFR network, both due to the large number of variables involved, and to the need of considering a sufficiently large range of hyperbolic terms. Therefore already in this case we are faced with the difficulty of finding from observed data the few relevant terms for the actual EGFR network. Despite this difficulty, augmented sparse reconstruction is able to find a very significant fraction of the links in the network. And we provide evidence that the method is robust with respect to changes in the size and type of system of representation.

The sparsity of links for the EGFR system has some variation between nodes; we have 11 variables with less than 4 distinct terms (linear, quadratic or hyperbolic) in the expression for their derivative, 9 variables with less than 8 terms and 1 variable, x_4, corresponding to protein node [RP], with 19 terms. This last variable is not sparse according to our definition of sparsity in Section 2.2, as it is connected to most other nodes, and it corresponds to the main 'hub' of the portion of the EGFR network that we analyze.

We assume that $R = 100$ time series with different initial conditions are available for each variable in the system, each sampled, for simulation purposes, at $B = 25$ equally spaced points in the time interval $[0, 27]$. However, in practice to be able to access 25 points for each time series may already be impractical, especially for proteomics applications. This sampling limitation contrast with the fact that, if the chosen sampling interval is too great, a signal peak and decline could be missed completely and this would mean that we would be unable to detect the main characteristics of the interactions. We compromised between these two opposing needs by making the assumption that we have access to the values of the signals on 5 equally spaced points and *also* to the averages of signals and their functions on 5 consecutive 5-point windows of the 25-point time series. This condition is stronger than having directly access only to 5 samples for each time series, but is sufficiently close to what can be obtained in an experimental setting. Because of this restriction on the number of samples, only $V = 5 \times R = 500$ uniformly selected points are actually used in the algorithm. We may be able to do without knowledge of averages on intervals by generating several possible representations from the data with a bootstrap-like technique similar to the one used in Napoletani and Sauer (2008).

To simulate the kind of noise observed in real protein networks, we add noise to the computed trajectories by taking the maximum M of each given time series and by adding uniform white noise in the interval $[-m, m]$ where m is equal to a given percentage of M. In particular, when $m = 0.15M$, i.e. when noise level is 0.15 of the maximum intensity, this procedure seems to give levels of noise consistent with experimental conditions. The characteristic shape of the noisy time series from the EGFR network is shown in Fig. 1 (noise level 0.15).

The initial conditions for each variable are chosen as uniformly distributed random numbers in the interval $[0, 40]$. In real systems the biologically significant ranges of initial conditions vary among different variables. This raises an interesting theoretical and practical question: Which is the minimal domain of initial conditions that allows the reconstruction of the network? This question is particulary relevant for networks that display simple dynamics, since each short trajectory may not carry the full information on the underlying network.

Now the generic shape of the activity levels of the network nodes is typical of a biological signaling system in steady state that is suddenly stimulated by a ligand, reaches a maximum activity and then decays to some baseline activity. The key point is to understand how the increase/decrease of activity of each node influence all others nodes, and the intensity and position of the peaks of the activity curve plays in this respect an important role. To gain a qualitative understanding of the shape of the activity levels of the nodes, and also to put the number of samples in perspective, we selected two time series from our simulation, one corresponding to protein node [R-Sh], in Fig. 1(a), and one corresponding to protein node [R-PLP] in Fig. 1(b). We selected these two trajectories because they highlight in a particularly striking way the fact that the speed of decay of the activity of each node can vary enormously, to the point of almost being

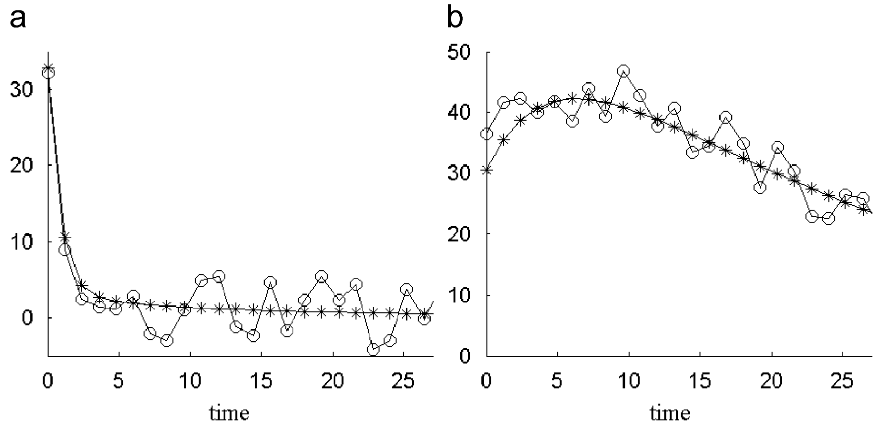

Fig. 1. In subplots (a) and (b) we show typical trajectories that are observed in the EGFR network, sampled uniformly 25 times in the time interval [0, 27]. Starred curves are the actual trajectories, circled curves are the trajectories with 0.15 relative noise added. Plot (a) shows a trajectory of x_9, i.e. [R-Sh], that settles within few samples points to a base value, plot (b) a trajectory of x_6, i.e. [R-PLP], with a much slower decay.

undetectable in an infrequent sampling setting. For the time series in Fig. 1(a) corresponding to protein node [R-Sh], we have fast initial decay, and, with our sampling rate, we detect only 2–3 points in the high varying region of the series, while for the time series in Fig. 1(b), corresponding to [R-PLP], we have slow decay and we can detect, with the same sampling rate, a larger proportion of points where the time series has not yet relaxed to its steady state. As we already stressed, this infrequent sampling is one of the reasons we had to move from the differential representation of the network to the integral one used in A–D, as it may be problematic to estimate derivatives in such infrequent sampling scenario.

We seek to reconstruct the EGFR network using models as in Eq. (2) (and implicitly Eq. (1)), i.e. we consider linear, quadratic and hyperbolic terms. The sampling interval of the hyperbolic terms is selected as $\mathrm{d}H = 10$ and the total number of hyperbolic terms for each variable as $S = 50$. The total number of terms for the model, and therefore the total number of parameters, is $M = N + N(N+1)/2 + NS = 1449$, far more than the $V = 500$ data points we use to find the links for each node. For most of the simulations, the number of random vectors to augment the model is chosen as $G = 2500$, the attenuation for the quadratic and the hyperbolic terms is chosen to be $\beta_q = \beta_h = 0.5$. In see Remarks 3.3 and 3.4 we explore instead the performance of the algorithm for several values of G and of attenuation coefficients.

Note that even a simpler model with only linear and quadratic terms would have $M = N + N(N+1)/2 = 299$ terms and therefore as many parameters to fit: a standard quadratic regression with $V = 500$ points would distribute the energy of the signals uniformly among all terms of the model. However, even without considering the added complexity of having hyperbolic terms, much bigger sets of data points V would be necessary to make quadratic regression competitive with our augmented sparse reconstruction algorithm. For a comparison of quadratic regression and augmented sparse reconstruction in the case of reconstructions using locally linear models see Napoletani and Sauer (2008).

3.2. Application of the algorithm at one single node

Now let us gain an understanding of the algorithm described in Section 2.3 by looking at a specific reconstruction for a node of the EGFR network.

In Fig. 2 we show a typical example of the sparse representation that can be obtained by applying A–D to the infrequently sampled, noisy trajectories of the EGFR network with the noise level as in Fig. 1. More specifically, we show the reconstructed representation for Y_2, the vector of all integrals of $\dot{x}_2$ on the left-hand side of Eq. (2), with respect to the vector of all integrals of linear, quadratic, and hyperbolic terms of the right-hand side of Eq. (2). We use step A to build all the numerical approximations of the integrals of all terms. With step B we scale all these integrals so that their norm is 1, this assures that the l_1 optimization procedure is not affected by the possibly very different activity levels of different nodes. At this stage we also pre-multiply quadratic and hyperbolic terms by the attenuation coefficients $\beta_q = \beta_h = 0.5$. With step C we use l_1 minimization to find a set of parameters that allow for an exact reconstruction of Y_2. Note that most of the parameters will be identically zero, a characteristic of solution found with l_1 minimization, as indicated in Section 2.1. Finally, we skip step D in this preliminary analysis at a single node, since our purpose in this subsection is only to get a visual qualitative sense of the sparsity and validity of the identified links.

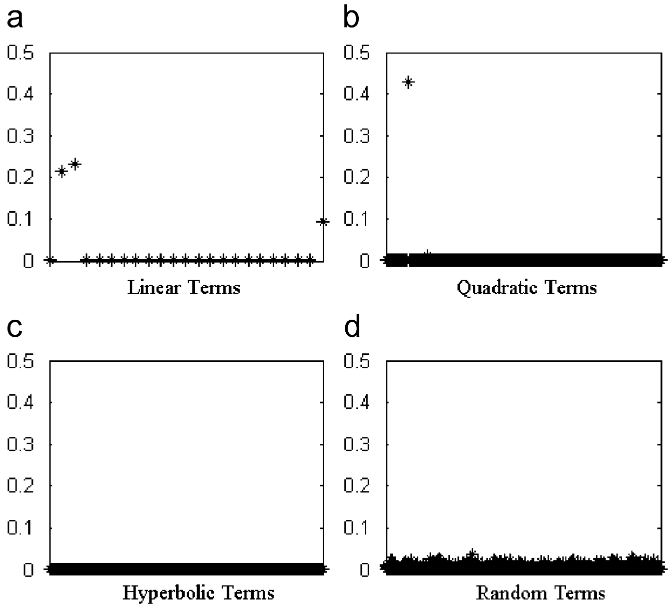

Fig. 2. This figure shows the norm of the parameters for the representation of Y_2, the vector of all integrals of $\dot{x}_2$ as defined in Eq. (2). From top left we have the parameters corresponding to: (a) the linear terms of Eq. (2); (b) the quadratic terms; (c) the hyperbolic terms; and (d) the random terms. Note that almost all deterministic parameters are identically zero.

We choose variable x_2, corresponding to [R$_a$], because it has very few terms in its actual differential equation, namely

$$\dot{x}_2 = -0.06x_2 + 0.2x_3 + 0.003x_1x_{23} - 0.02x_2^2,$$

where x_3 corresponds to [R$_2$], x_1 to [R] and x_{23} to [EGF]. [R$_a$] is involved in the early action of [EGF] in the network, as [EGF] binds to monomeric EGFR (labeled [R] in the network) and forms the receptor ligand complex [R$_a$], this in turn dimerizes as [R$_2$]. Even though we chose to focus our attention on [R$_a$] simply for its very clear linkage structure, we note that the ability to monitor the links that affect the dynamics of [R$_a$] could be useful to detect an early disruption of the whole EGFR network. Since [EGF] is the main input of the network and is used to stimulate the network, it is important to be able to detect its direct action on nodes such as [R$_a$]. We can also appreciate, from a modeling viewpoint, the special role of [EGF] by noting that it is the only variable in the network in Appendix B that does not depend in its dynamics from the other variables. Indeed a [EGF] ligand binding to a receptor can stimulate a variety of biologic functions and cellular responses including growth, differentiation, migration and survival. The same [EGF] receptor can connect to a plurality of downstream pathways which may have opposing or synergistic actions. Methodologies now exist for measuring the phosphorylated or activated state of 200 signal pathway protein nodes within a cellular network, see for example Petricoin et al. (2007), Sheehan et al. (2008), and Wulfkuhle et al. (2008). It would therefore be highly feasible to stimulate a cell with a ligand such as [EGF] and then measure the time course of fluctuation of network node activation states. As we said in Section 3.1, if the sampling interval is of sufficient frequency, such that the transient linkages are not lost between sample times, then the methodology described herein could discover new linkages and build linkage maps that are functionally relevant.

We plot the norm of the parameters of the model reconstructed with the augmented sparse reconstruction in Fig. 2, more particularly: the parameters of the linear terms of Eq. (2) are plotted in Fig. 2(a), from l_1 to l_{23}; the parameters of quadratic terms in Fig. 2(b), ordered from $q_{1,1},\ldots, q_{1,23}, q_{2,2},\ldots,q_{22,23}$; the parameters of the hyperbolic terms in Fig. 2(c), in the order $h_{1,1},\ldots,h_{1,10},\ldots,h_{23,1},\ldots,h_{23,10}$; the parameters of the random terms are plotted in Fig. 2(d). The three largest parameters across all terms correspond exactly to three of the terms in the actual differential equation of $\dot{x}_2$, namely x_2, x_3, corresponding to the unimolecular interactions with, respectively, [R$_a$] itself and [R$_2$] and x_1x_{23}, corresponding to the bimolecular interaction with [R] and [EGF]. We are missing instead the x_2^2 term that corresponds to a nonlinear self-interaction of [R$_a$]. The forth largest parameter in the reconstructed representation corresponds to the x_{23} term, so it repeats to some extent the information on the network linkage given by the x_1x_{23} term.

The strength of the algorithm is that almost all others linear quadratic and hyperbolic parameters are *identically* zero.The exceptions are the x_2x_{20} term, barely visible at the right of the leading quadratic term, that has a parameter that is at least 7 times smaller than any of the leading terms, and the x_1x_5 term that has a parameter at least 45 times smaller than any of the leading terms.

If we apply the reconstruction algorithm to this node with 20 different realizations of 0.15 relative noise, the x_2 term appears as dominant 18 times, the x_3 term 15 times, the x_1x_{23} term 20 times and the x_2^2 term 3 times. Note that the dominance of a term is not only due to the size of its actual parameter in the differential equation, but also to the intensity of the corresponding signal, for example x_1x_{23} has parameter 0.003 in the differential equation for $\dot{x}_2$ and yet it is recovered more often than x_2^2 that has larger parameter 0.02. We believe this has to do with the specific norm scaling of the terms of the representation. Different choices of norm scaling may be useful to improve further the performance of the method, refer as well to Remark 2.6.

The example of the reconstruction of the representation of $\dot{x}_2$ is typical: some terms not only may be missing, but they can be partially wrong, for example a term x_i may appear in the representation as x_i^2, or a term x_ix_j may be replaced by a term x_ix_k that gives similar shapes for the given initial conditions. Note that these two possibilities do carry some significant information on the geometry of the network, even though the specific terms are incorrect.

At this point it is instructive to check the ability in identifying links of a simpler technique such as correlation, which is very frequently used in finding potential links among nodes in a network when few trajectories are available. The setting in which we compute correlation coefficients is exactly the same as in our algorithm, i.e. we compute the correlation coefficients of the vector Y_2 and each of the vectors in the system of representation generated by linear, quadratic and hyperbolic terms. More specifically, in Fig. 3(a) we show the correlation coefficients of Y_2 and the vectors of all integrals of each of the linear terms of Eq. (2); correlation coefficients of Y_2 and the vectors of all integrals of each of the quadratic terms are plotted in Fig. 3(b); and correlation coefficients with the integrals of hyperbolic terms are in Fig. 3(c). The most negatively correlated linear term corresponds to x_2, the most negatively correlated quadratic term to x_2^2, and the cluster of most negatively correlated hyperbolic terms that can be seen in Fig. 3(c) correspond to x_2 as well. Note, however, that many terms show similar large negative correlation coefficients. There are especially a large number of negatively correlated quadratic terms, so that it is difficult to set a threshold on the norm of the correlation coefficients that would, for example, single out *only* x_1x_{23} as a relevant term in the differential equation of x_2.

Unlike usual correlation coefficients, the considerable sparsity of the coefficients of the reconstructions computed by our method (in Fig. 2) allows for an accurate distinction of false links and true links.

3.3. Global estimation of network structure

To evaluate globally the quality of the reconstruction results for different levels of noise we use the ratio of computed true links with respect to the total number of true links (true positives rate) and the ratio of computed false links with respect to the total number of false links (false positives rate). An important question when assessing the quality of reconstruction is the proper estimation of the thresholds T_n used in D. In general we expect these thresholds to vary according to the noise level in the time series. The sampling rate will also affect our degree of confidence in the computed links so we must find an automatic way to estimate the threshold from the data. Recall from Remark 2.5 that we defined, for each node i, the threshold $T_i = K\sigma_i$, $i = 1,\ldots,N$, where K, the threshold multiplier, is some constant determined for all the nodes, while σ_i is the standard deviation of the absolute value of the deterministic non-zero coefficients of the representation of node i.

To explore the performance of the algorithm of Section 2.3 with this class of threshold multipliers, suppose that for each noise level in the time series, we select K so that the false positives rate stays below 0.1. We perform such analysis for each possible relative noise in the trajectories from 0 to 0.25 in intervals of 0.05. In Fig. 4(a) we can see the result of such choice of thresholds: the true positives rate (starred curve) is high (around 0.65) even for realistic trajectories' noise of the order of 0.20. Fig. 4(a) shows the true positives rates (circled curve) when the false positives rate is

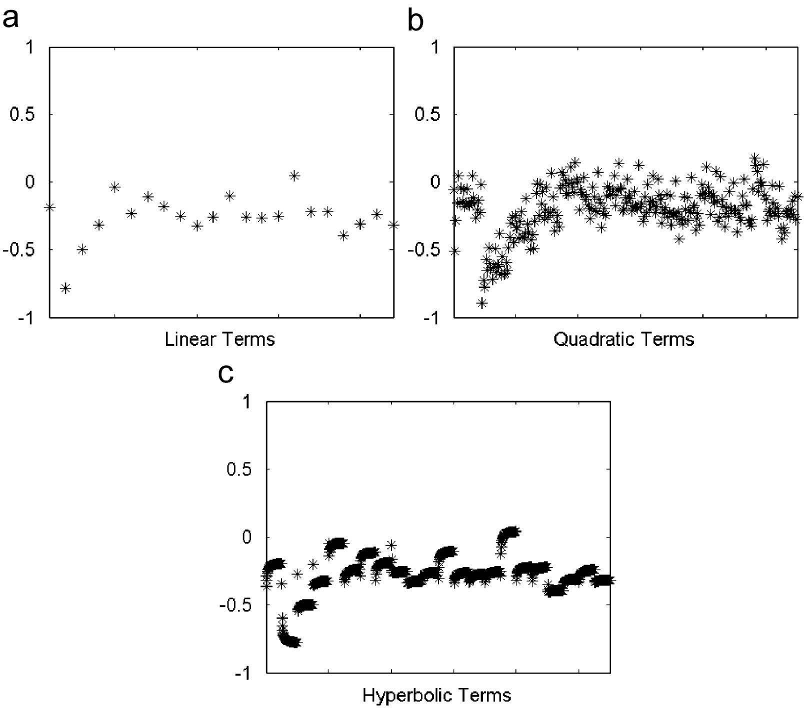

Fig. 3. From top left, we plot the correlation coefficients of Y_2, the vector of integrals of $\dot{x}_2$ as defined in the left-hand side of Eq. (2), and: (a) the vectors of integrals of linear terms of the right-hand side of Eq. (2); (b) the vectors of integrals of quadratic terms; (c) the vectors of integrals of hyperbolic terms.

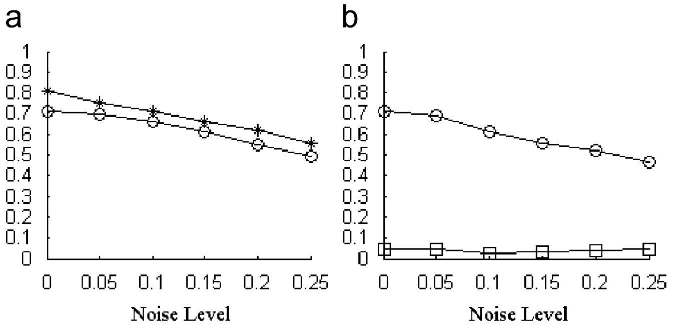

Fig. 4. In plot (a) we have the average true positives rates for relative noise in the trajectories from 0 to 0.25 when the value of the threshold multiplier K is set to keep the false positive rate at: 0.1 (starred curve) and 0.05 (circled curve). In plot (b) we have the average true positive rates (starred curve) and average false positive rates (squared curve) for relative noise in the trajectories from 0 to 0.25 where the value of the threshold multiplier K is found for each noise level by using the heuristic E of Appendix A.

kept at 0.05, in this case we observe lower true positive rates, but still sizeable: when relative noise is 0.20 we have true positive rates around 0.60.

Note that even a 0.60 true positive rate, if false positives rate can be kept to 0.1, has potentially very significant clinical applications. It is well established in biologic systems (Santos et al., 2007) that the wiring interconnections of cellular signaling networks can profoundly influence the cell phenotype. Depending on the linkage pattern, a cell can respond in a positive or negative manner to a stimulatory ligand, or can switch responses from one stimulatory ligand to another, therefore a cancer drug can either kill the cancer cell or stimulate its growth (Araujo et al., 2007). Consider that the environment of the cell, the treatment regimen, or its neoplastic state can change completely the linkages emanating from one class of receptor and one ligand. Therefore the ability of our method of detecting the leading links for each protein node could quickly determine, for example, whether a new pathway is active in a cancerous cell. Moreover activated and biologically important signal pathway nodes may be elevated 10 to a hundred fold more than quiescent nodes, improving therefore the actual experimental ability to detect nodes with our method. The knowledge of the new diseased pathways can then potentially be used to target specific receptors to suppress those pathways.

Remark 3.1. We generated Fig. 4 by repeating the analysis with 20 different realizations for each level of relative noise. In the case of false positive rates equal to 0.1, for all 20 realizations the computed true positive rates differed from the average by at most 0.054 units, with standard deviation, at each noise level, of at most 0.026. When we kept false positive rates at 0.05, for all 20 realizations the computed true positive rates differed from the average by at most 0.071 units, with standard deviation, at each noise level, of at most 0.03.

Note that even for noiseless trajectories, we still do not find all true links, this is due in part to the infrequent sampling of the trajectories that hides subtle interactions among the nodes. Note that the rates we display are obtained excluding from the average the reconstruction of variable 4, i.e. [RP], that does not have very sparse representation and which is a 'hub', so likely to be better known experimentally. We decided to exclude that specific protein in our estimates of the errors because only very few proteins are believed to perform the role of hub of a network in signaling pathways, therefore we believe that the error rates computed above are a better indicator of the errors we would find in computing the representation of a generic protein in a large network. Moreover the representation of x_4 is so different from the others that the use of the same threshold multiplier for it as for all other variables does not seem appropriate. Including variable 4 in computing the errors, without any modification in the choice of threshold multiplier, would lead true positive rates to slightly worsen.

Remark 3.2. It is possible in principle to have cases when much more accurate averages of the trajectories are experimentally available. To simulate this scenario, suppose we sample the trajectories of the EGFR network uniformly 100 times in the time interval [0, 27] and that we use these samples to make better estimates of the averages of all variables at 5 locations, so that we still use only $V = 500$ points in the algorithm, for direct comparison with the previous case. The error rates improve significantly in this case, if we keep false positive rate to 0.1, then average true positive rates are about 0.88 in the absence of noise, this is a improvement of almost 0.08 with respect to the same time series sampled uniformly only 25 times. In Fig. 5(a) we show average true positive rates for noise level from 0 to 0.25 in this accurate averages scenario. We use 20 realizations for each noise level in computing the averages of the true positive rates.

Remark 3.3. Fig. 5(b) shows that a large number of random terms is important for the proper functioning of algorithm A–D. Namely we show that the average true positive rates improve when we increase the number G of random terms from 0 to 3000 in intervals of 250. The averages of all variables used to generate this plot are computed at 5 equally spaced points from network trajectories sampled 25 times in the interval [0, 27]. False positive rates are kept at 0.1 and relative noise level is fixed at 0.15. The true positives rate is very low, just 0.13, when there are no random terms added. Addition of 1000 or more terms gives high true positive rates, about 0.68, that are comparable for several values of the number of terms G. These results show also the robustness of the algorithm with respect to the choice of G.

Remark 3.4. Our numerical results on the EGFR network suggest also that attenuation of both quadratic and hyperbolic terms improves true positive rates by roughly 0.10 with respect to a model in which no attenuation is performed, from about 0.60 to about 0.70, when false positive rates are kept at 0.1 and relative noise level is fixed at 0.15. The same improvement is observed with respect to a full suppression, where the attenuation coefficients are set to zero for both quadratic and hyperbolic terms. This scenario corresponds to a model in which only linear terms are used. While it is already surprising that the algorithm allows for a true positives rate close to 0.6 with only linear terms, it is clear that to have a larger model and suppress higher complexity terms strongly improves the reconstruction. The numerical results in Fig. 6 seems to show that a wide range of attenuations are suitable, as long as quadratic terms are not fully suppressed.

In practice we need to find a suitable value for the threshold multiplier K from the reconstruction data generated by the algorithm at the end of step C. We give an heuristic description of the method we developed to address this problem and we refer to Appendix A for a more detailed implementation.

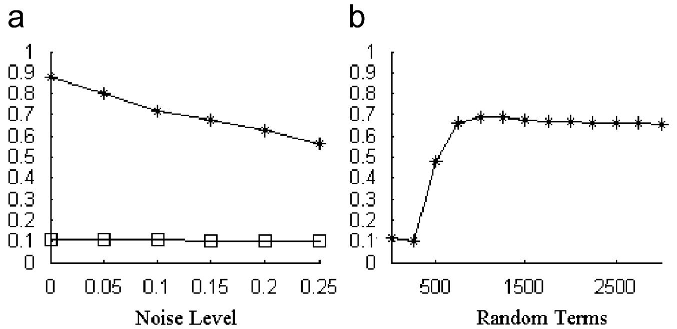

Fig. 5. In plot (a) we have the true positives rates (starred curve) in the accurate averages scenario of Remark 3.2, for relative noise in the trajectories from 0 to 0.25 when the value of the threshold multiplier K is set to keep the false positive rate (squared curve) at 0.1. In plot (b) we have the true positive rates (starred curve) as a function of the number of random terms added to the model. Noise level is 0.15. The value of the threshold multiplier K is set to keep the false positive rate at 0.1.

Suppose that we look at the total number of links selected by the algorithm for a given threshold, then if the threshold is small, we expect many links to be false positives, while if the threshold is large, we expect many links to be true positives. The true links are sparse, therefore when we increase the threshold we slowly reduce the number of true positive detected by the algorithm. The slow reduction of true positives implies an interesting assumption on the biology of the network, because one consequence is that there should be no large clusters of computed parameters, and this is possible if the true kinetic parameters of the network, scaled by the norm of the integrals of the trajectories, are roughly uniformly distributed. This is a reasonable assumption, even though it is difficult to have reliable information on the statistical properties of the true kinetic parameters: the absolute activation levels of the nodes are often very different, so that the true kinetic parameters are effectively divided by norms that are widely distributed, and if there is any cluster before scaling, it should be removed by the scaling itself.

Note that false positives are assumed to have small coefficients in the representation, so when the threshold is small, even a small change of threshold will significantly reduce the number of false positives that we detect. We can monitor how fast the decay of total links detected is, as a function of the threshold, and conjecture that, as soon as the decay goes from fast to slow, we have probably already eliminated most false positives. We then select this transition point as a suitable value of the threshold.

In Appendix A we will make all these assumptions quantitative enough to be automatically implemented. The automatic choice of threshold along the lines of this heuristic gives high true positive rates and low false positive rates. In Fig. 4(b) we plotted the average true positive rates and average false positive rates computed by using, in step D of the algorithm, the values of K estimated with the method of Appendix A. With 0.10 relative noise in the trajectories, we have a significant average true positives rate of about 0.61 with average false positives rate of only 0.025.

4. Discussion

There are still many open questions whose answers will shape the way augmented sparse reconstruction methods are applied to network reconstruction. What is the limiting node sparsity that

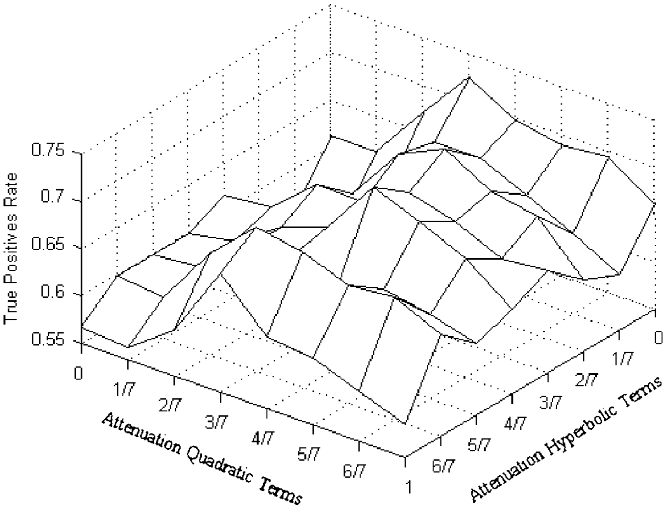

Fig. 6. True positive rates as a function of the attenuation coefficients of quadratic and hyperbolic terms. Noise level in the trajectories is 0.15. The value of the threshold multiplier K is set to keep the false positive rate at 0.1.

still allows the network itself to be recovered with this method? How are the error rates affected if only a subset of the variables is available? How does the network we compute on this subset of variables relates to the full network? If we have a node that is unrelated to the chosen subset of the network, it tends to have a greater portion of its norm accounted for by the random terms, and this can be used to decide whether it is well connected to the other variables, but further research in this direction is needed.

In some cases the 'skeleton' of the network may be available, for example for protein networks we may know roughly how the system is connected for healthy cells. Can we use this additional information to detect, with this method, whether patients with cancer develop additional strong links among nodes? Preliminary evidence suggests that small new links that do not make the system unstable are often detectable, but it would be interesting to use the available information on the skeleton of the network directly in the algorithm.

Even when previous information on the network is not available, it is very important for clinical applications to determine whether a specific protein has very distinct representations for cancerous cells and for healthy ones. If this is the case, then the reconstruction algorithm can predict changes in the signaling pathways that are likely due to the cancer itself.

One possible extension of our method is to perform the estimation of links only on local subsets of trajectories. This local application of the algorithm may highlight different links that could be dominant for different sets of initial conditions, in Napoletani and Sauer (2008) we show that this strategy is indeed feasible. Note that the augmented sparse reconstruction scheme has an edge over simple l_2 regression especially when there is a very limited set of initial conditions. Even in the case in which it is possible to span the entire phase space of the network, there is a limit to the density of the initial conditions that can be taken and therefore a local application of this method will be beneficial (because there are only a few local trajectories). By putting together the information on links that arise in different regions of the phase space it may be possible to find very tenuous links that would otherwise be undetectable in a global analysis. A clear advantage of a local version of the method is its generality, since simple, low degree polynomial models can always be used.

The augmented sparse reconstruction described in this paper is able to identify relevant links among nodes in very large systems of representation and with very noisy conditions, so we expect the algorithm to scale well to the use of cubic or higher degree terms in the representation and even to the use of general power functions, possibly with non-integer and negative exponents. The use of attenuation of blocks of terms in the representation will turn out to be even more important when the size of the dictionary of terms is increased.

We stated in the introduction that a promising approach to biological networks is to deemphasize exact modeling, in favor of a robust identification of classes of suitable models. If we take one step forward in this direction, then the techniques of network control, and the very notion of global stability of a network, must be changed in such a way that they are valid for entire classes of indistinguishable systems (Judd and Smith, 2004) that produce trajectories that are qualitatively similar. In this perspective, the reconstruction algorithm described in this paper could be used as an intermediate step of data-driven control schemes based on particle filter techniques, by providing an indistinguishable model that locally behaves as the real one. This potential application of the augmented sparse reconstruction method would be an interesting step in the direction of real time, personalized therapies that require an online estimation and control of specific pathways in the cell networks of individual patients (Liotta et al., 2001; Araujo et al., 2005b).

Appendix A. Automatic choice of threshold

Denote by $S(K)$ the total number of selected links that are found in step D of the augmented sparse reconstruction algorithm by using thresholds $T_n = K\sigma_n$. We can split $S(K)$ as $S(K) = S_t(K) + S_f(K)$ where $S_t(K)$ denotes the number of true computed links and $S_f(K)$ the number of false computed links. Since for each node we have only a small number of true links by assumption, and their corresponding coefficients in the representation are, in general, very large, we can conjecture that, as we let K increase from 0 to ∞ in intervals of $\mathrm{d}K$, $S_t(K)$ will decrease very slowly at the beginning. Since $S_t(K)$ assumes only integer values, this slow decay will appear as infrequent small jumps, this means that the value of the differences $\mathrm{d}S(K) = S(K + \mathrm{d}K) - S(K)$ of $S(K)$ will be dominated by the differences $\mathrm{d}S_f(K)$ of $S_f(K)$ for small values of K and by the differences $\mathrm{d}S_t(K)$ of $S_t(K)$, for larger values of K. If this conjecture is correct, it seems possible to infer some of the properties of $S_f(K)$ which is not known, from those of $S(K)$, which is generated by the algorithm.

In Fig. 7(a)–(c) we show approximations to $\mathrm{d}S_t(K)$, $\mathrm{d}S_f(K)$ and $\mathrm{d}S(K)$ for a specific reconstruction with relative noise in the time series of the order of 0.10. Note that $S(K)$ is identically zero for $K > 3.26$, we can immediately see the similarity of $\mathrm{d}S(K)$ and $\mathrm{d}S_f(K)$ in the frequency of negative jumps for small values of K. The frequency of jumps of $\mathrm{d}S(K)$ greatly decreases around $K = 0.29$. It looks like there is for that value of K a transition point such that frequency of jumps is much higher before the transition point, and much lower after it. To see this transition point more clearly, let $K_1, \ldots, K_M$ be the values of K, ordered from smallest to largest, for which $\mathrm{d}S(K) \neq 0$ and define the function $J(i) = K_i - K_{i-1}$, $i = 2, \ldots, M$ that computes the width of negative jumps. We plot J in Fig. 7(d) and we can see that for $i \approx 57$ we suddenly have much wider intervals between jumps, this value of i corresponds to $K_{57} \approx 0.29$ as we would expect from our analysis of $\mathrm{d}S(K)$. We argue that a suitable value K_f of the threshold multiplier K is the one for which $J(i)$ has very different local averages for $i < f$ and $i > f$. We use the following rule to find this transition value K_f that determines the passage from high to low frequency of jumps in $\mathrm{d}S(K)$:

E. Set an integer I, let $V(i) = (\bar{J}_{i-I})/(\bar{J}_{i+I})$, $i = I, \ldots, M - I$, where we denote by $\bar{J}_{i-I}$ the mean of J for values between $i - I$ and i and by $\bar{J}_{i+I}$ the mean of J for values between i and $i + I$. Denote by f the index for which $V(i)$ is minimum, and by K_f the corresponding threshold multiplier.

Basically, for each jump i, we compute the means of the length of intervals in-between jumps for the set of I consecutive jumps *before* the jump i and for the set of I consecutive jumps *after* the jump i. Then we take their ratio to measure how much the average length of intervals between the jumps is changing. Finally we select the jump for which this ratio is maximum.

In Fig. 4(b) from Section 3.3 we plotted the average true positive rates and average false positive rates computed by using, in step D of the algorithm, the values of K estimated with the heuristics of this section. With 0.10 relative noise in the trajectories, we have a significant average true positives rate of about 0.61 with average false positives rate of only 0.025.

Remark A.1. To generate Fig. 7 we chose a fine uniform sampling $\mathrm{d}K = 0.001$ of K, up to $K = 3.5$, so that $\mathrm{d}S(K)$ never goes below -2. To have the limiting case in which $\mathrm{d}S(K) \geqslant -1$ for all K seems to require excessively fine sampling rate.

Remark A.2. When the total number of jumps is about 120 (as in Fig. 7(d)), it is reasonable to take the size I of the two sets of

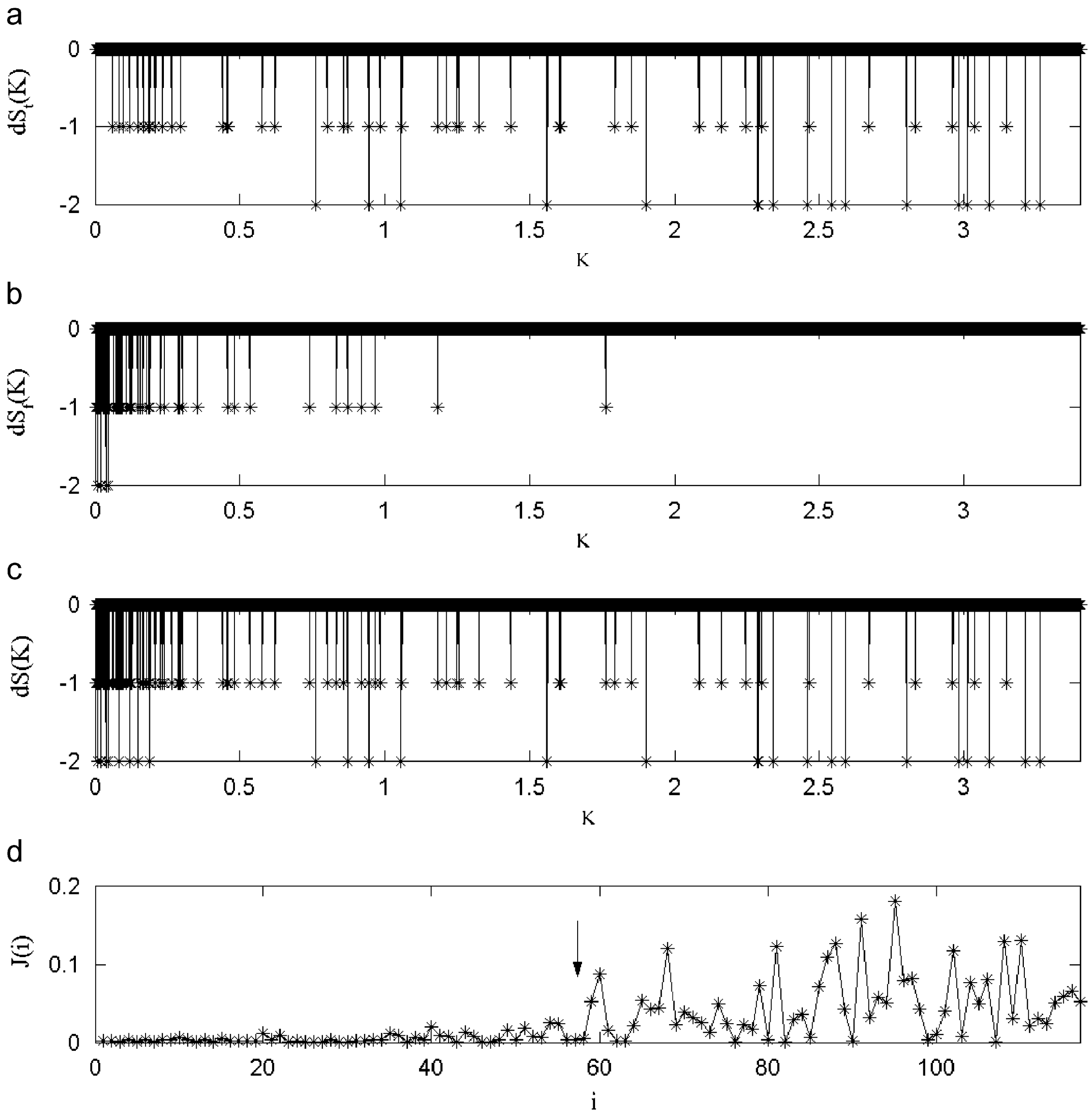

Fig. 7. Plots of: (a) the change $dS_t(K)$ of true links detected by the algorithm A–D for a specific reconstruction, as a function of threshold multiplier K; (b) the change $dS_f(K)$ of detected false links; (c) the change $dS(K)$ of all detected links. In plot (d) we show $J(i)$, the distance between the $i-1$-th and the i-th negative jumps in $dS(K)$. The arrow in plot (d) points to the index values, around $i = 57$, for which we have a large change of mean frequency of jumps.

consecutive jumps (before and after the i-th jump) to be small compared to this total, on the other hand we want a good estimate of the averages, so I should not be too small; we settled for $I = 20$ and we estimated the values of K by applying step E to the collection of representations for all nodes found by the algorithm. Other values of $I \approx 20$ give comparable threshold levels.

Remark A.3. We use 20 realizations for each noise level in computing the averages of the true and false positive rates shown in Fig. 4(b) in Section 3.3. For all 20 realizations the computed true positive rates differ from the averages by at most 0.11 units, with standard deviation, at each noise level, of at most 0.058. The false positives rate is below 0.11 for all realizations and all noise levels.

Appendix B. The EGFR network

This is a portion of the EGFR network as described in Araujo et al. (2005a). Before each differential equation, we label the protein whose concentration dynamics is described by that equation.

[R] $\dot{x}_1 = 0.06x_2 - 0.003x_1x_{23}$

[R$_a$] $\dot{x}_2 = -0.06x_2 + 0.2x_3 + 0.003x_1x_{23} - 0.02x_2^2$

[R$_2$] $\dot{x}_3 = -1.1x_3 + 0.01x_4 + 0.01x_2^2$

[RP] $\dot{x}_4 = x_3 - 0.01x_4 + 0.2x_5 + 0.3x_6 + 0.05x_7 + 0.03x_8 + 0.6x_9 + 0.3x_{10} + 0.3x_{11} + 0.12x_{12} - 0.0045x_4x_{13}$
$-0.0009x_4x_{14} - 0.0009x_4x_{15} - 0.06x_4x_{16} - 0.006x_4x_{17} - 0.003x_4x_{19} - 0.09x_4x_{20} - 0.00024x_4x_{22} - 450x_4/(50 + x_4)$

[R-PL] $\dot{x}_5 = -1.2x_5 + 0.05x_6 + 0.06x_4x_{16}$

[R-PLP] $\dot{x}_6 = x_5 - 0.35x_6 + 0.006x_4x_{17}$

[R-G] $\dot{x}_7 = -0.05x_7 + 0.06x_8 - 0.01x_7x_{21} + 0.003x_4x_{19}$

[R-G-S] $\dot{x}_8 = -0.09x_8 + 0.0045x_4x_{13} + 0.01x_7x_{21}$

[R-Sh] $\dot{x}_9 = -6.6x_9 + 0.06x_{10} + 0.09x_4x_{20}$

[R-ShP] $\dot{x}_{10} = 6x_9 - 0.07x_{10} + 0.0009x_4x_{14}$

[R-Sh-G] $\dot{x}_{11} = -0.4x_{11} + 0.0214x_{12} + 0.0009x_4x_{15} - 0.01x_{11}x_{21} + 0.003x_{10}x_{19}$

[R-Sh-G-S] $\dot{x}_{12} = -0.1843x_{12} + 0.00024x_4x_{22} + 0.009x_{10}x_{13} + 0.01x_{11}x_{21}$

[G-S] $\dot{x}_{13} = 0.03x_8 + 0.0429x_{12} - 0.0015x_{13} + 0.1x_{22} - 0.0045x_4x_{13} - 0.009x_{10}x_{13} - 0.021x_{13}x_{14} + 0.0001x_{19}x_{21}$

[ShP] $\dot{x}_{14} = 0.3x_{10} + 0.1x_{15} + 0.1x_{22} - 0.0009x_4x_{14} - 0.021x_{13}x_{14} - 0.003x_{14}x_{19} - 1.7x_{14}/(340 + x_{14})$

[Sh-G] $\dot{x}_{15} = 0.3x_{11} - 0.1x_{15} + 0.064x_{22} - 0.0009x_4x_{15} + 0.003x_{14}x_{19} + 0.03x_{15}x_{21}$

[PLC$_\gamma$] $\dot{x}_{16} = 0.2x_5 - 0.06x_4x_{16} + x_{17}/(100 + x_{17})$

[PLC$_\gamma$P] $\dot{x}_{17} = x_6 + x_{17} + x_{18} + x_4x_{17} + x_{17}/(1 + x_{17})$

[PLC$_\gamma$P-I] $\dot{x}_{18} = x_{17} - 0.03x_{18}$

[Grb] $\dot{x}_{19} = 0.05x_7 + 0.1x_{11} + 0.0015x_{13} + 0.1x_{15} - 0.003x_4x_{19} - 0.003x_{10}x_{19} - 0.003x_{14}x_{19} - 0.0001x_{19}x_{21}$

[Shc] $\dot{x}_{20} = 0.6x_9 - 0.09x_4x_{20} + 1.7x_{14}/(340 + x_{14})$

[SOS] $\dot{x}_{21} = 0.06x_8 + 0.0214x_{12} + 0.0015x_{13} + 0.064x_{22} - 0.01x_7x_{21} - 0.01x_{11}x_{21} - 0.03x_{15}x_{21} - 0.0001x_{19}x_{21}$

[ShG-S] $\dot{x}_{22} = 0.12x_{12} - 0.064x_{22} - 0.00024x_4x_{22} + 0.021x_{13}x_{14} + 0.03x_{15}x_{21}$

[EGF] $\dot{x}_{23} = 0$

References

Albert, R., 2005. Scale-free networks in cell biology. J. Cell Sci. 118, 4947–4957.

Araujo, R.P., Petricoin III., E.F., Liotta, L.A., 2005a. A mathematical model of combination therapy using the EGFR signaling network. Biosystems 80 (1), 57–69.

Araujo, R.P., Doran, C., Liotta, L.A., Petricoin, E.F., 2005b. Network-targeted combination therapy: a new concept in cancer treatment. Drug Discov. Today 1, 425–433.

Araujo, R.P., Liotta, L.A., Petricoin, E.F., 2007. Proteins, drug targets and the mechanisms they control: the simple truth about complex networks. Nat. Rev. Drug Discov. 6, 871–880.

Candes, E.J., Tao, T., 2004. Near-optimal signal recovery from random projections: universal encoding strategies. IEEE Trans. Inf. Theory 52, 5406–5425.

Candès, E.J., Rudelson, M., Tao, T., Vershynin, R., 2005. Error correction via linear programming. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 295–308.

Candes, E.J., Romberg, J.K., Tao, T., 2006. Stable signal recovery from incomplete and inaccurate measurements. Comm. Pure Appl. Math. 59 (8), 1207–1223.

Chen, S., Donoho, D., Saunders, M.A., 1998. Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. SISC 20 (1), 33–61.

Chou, I.C., Martens, H., Voit, E.O., 2006. Parameter estimation in biochemical systems models with alternating regression. Theor. Biol. Med. Model. 3 (25).

Debashis, P., Bair, E., Hastie, T., Tibshirani, R., 2008. "Pre-conditioning" for feature selection and regression in high-dimensional problems. Ann. Statist. 36 (4), 1595–1618.

Donoho, D., 2006a. For most large underdetermined systems of linear equations the minimal lscr1-norm solution is also the sparsest solution. Comm. Pure Appl. Math. 59 (6), 797–829.

Donoho, D., 2006b. For most large underdetermined systems of equations, the minimal lscr1-norm near-solution approximates the sparsest near-solution. Comm. Pure Appl. Math. 59 (7), 907–934.

Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. Science 315 (5814), 972–976.

Friedman, N., Linial, M., Nachman, I., Pe'er, D., 2000. Using Bayesian networks to analyze expression data. J. Comput. Biol. 7 (3–4), 601–620.

Gardner, T.S., di Bernardo, D., Lorenz, D., Collins, J.J., 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301 (5629), 102–105.

Hastie, T., Tibshirani, R., Friedman, J.H., 2001. The Elements of Statistical Learning. Springer, Berlin.

Hu, P., Bader, G., Wigle, D.A., Emili, A., 2006. Computational prediction of cancer-gene function. Nat. Rev. Cancer 7, 23–34.

Husmeier, D., 2003. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics 19, 2271–2282.

Judd, K., Smith, L.A., 2004. Indistinguishable states II: the imperfect model scenario. Physica D 196, 224–242.

Kaufman, L., Rousseeuw, P.J., 2005. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley-Interscience, New York.

Lacouture, M.E., 2006. Mechanisms of cutaneous toxicities to EGFR inhibitors. Nat. Rev. Cancer 6, 803–812.

Liotta, L.A., Kohn, E.C., Petricoin, E.F., 2001. Clinical proteomics: personalized molecular medicine. J. Am. Med. Assoc. 286, 2211–2214.

Lustig, I.J., Marsten, R., Shanno, D.F., 1994. Interior-point methods for linear programming: computational state of the art. ORSA J. Comput. 6, 1–14.

Mallat, S., 1998. A Wavelet Tour of Signal Processing. Academic Press, New York.

Nachman, I., Regev, A., Friedman, N., 2004. Inferring quantitative models of regulatory networks from expression data. Bioinformatics 20, i248–i256.

Napoletani, D., Sauer, T., 2008. Reconstructing the topology of sparsely-connected dynamical systems. Phys. Rev. E 77, 026103.

Petricoin, E.F., Espina, V., Araujo, R.P., Midura, B., Yeung, C., Wan, X., Eichler, G.S., Johann, D.J., Qualman, S., Tsokos, M., Krishnan, K., Helman, L.J., Liotta, L.A., 2007. Phosphoprotein signal pathway mapping: Akt/mTOR pathway activation association with childhood Rhabdomyosarcoma survival. Cancer Res. 67 (7), 3431–3434.

Rogers, S., Girolami, M., 2005. A Bayesian regression approach to the inference of regulatory networks from gene expression data. Bioinformatics 21, 3131–3137.

Santos, S.D.M., Verveer, P.J., Bastiaens, P.I.H., 2007. Growth factor-induced MAPK network topology shapes Erk response determining PC-12 cell fate. Nat. Cell Biol. 9, 324–330.

Sheehan, K.M., Gulmann, C., Eichler, G.S., Weinstein, J., Barrett, H.L., Kay, E.W., Conroy, R.M., Liotta, L.A., Petricoin, E.F., 2008. Signal pathway profiling of epithelial and stromal compartments of colonic carcinoma reveal epithelial–mesenchymal transition. Oncogene 27 (3), 323–331.

Tegner, J., Yeung, M.K.S., Hasty, J., Collins, J.J., 2003. Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. Proc. Natl. Acad. Sci. 100 (10), 5944–5949.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. B 58 (1), 267–288.

Voit, E.O., 2000. Computational Analysis of Biochemical Systems. Cambridge University Press, Cambridge.

Voss, H., Timmer, J., Kurths, J., 2004. Nonlinear dynamical system identification from uncertain and indirect measurements. Int. J. Bif. Chaos 14, 1905.

Wulfkuhle, J.D., Speer, R., Pierobon, M., Laird, J., Espina, V., Deng, J., Mammano, E., Yang, S.X., Swain, S.M., Nitti, D., Esserman, L.J., Belluco, C., Liotta, L.A., Petricoin, E.F., 2008. Multiplexed cell signaling analysis of human breast cancer: applications for personalized therapy. J. Prot. Res. 7 (4), 1508–1517.

Yeung, M.K.S., Tegner, J., Collins, J.J., 2002. Reverse engineering gene networks using singular value decomposition and robust regression. Proc. Natl. Acad. Sci. 99 (9), 6163–6168.

Zak, D., Doyle, F., Schwaber, J., 2002. Local identifiability: when can genetic networks be inferred from microarray data? In: Proceedings of the Third International Conference on Systems Biology, pp. 236–237.

Zhang, Y., 1998. Solving large-scale linear programs by interior-point methods under the MATLAB environment. Optim. Methods Software 10, 1–31.