12

Network reconstruction based on steady-state data

Eduardo D. Sontag[1]

Department of Mathematics, Hill Center, 110 Frelinghuysen Rd,
Rutgers, The State University of New Jersey Piscataway,
NJ 08854-8019, U.S.A.

Abstract

This paper discusses a theoretical method for the "reverse engineering" of networks based solely on steady-state (and quasi-steady-state) data.

Introduction

A central concern of systems biology is that of elucidating the mechanisms and overall architecture underlying the observed behaviour of biomolecular networks. In this context, the "reverse engineering problem" concerns itself with the unravelling of the web of interactions among the components of protein and gene regulatory networks, so as to map out the direct or *local* interactions among components. These direct interactions capture the topology of the functional network.

An intrinsic difficulty in capturing these direct interactions, at least in intact cells, is that any perturbation to a particular gene or signalling component -using tools such as traditional genetic experiments, RNAi (RNA interference), hormones, growth factors, or pharmacological interventions- may rapidly propagate throughout the network, thus causing *global* changes which cannot be easily distinguished from direct effects. Thus, a major goal in reverse engineering is to use these observed global responses -such as steady-state changes in concentrations of active proteins, mRNA levels, or transcription rates- in order to infer the local interactions between individual nodes.

[1] *sontag@math.rutgers.edu*

One approach to solving this global-to-local problem is the "unravelling," or "Modular Response Analysis" (MRA) method proposed in [1] (see also [2]) and further elaborated upon in [3–6] (see [7,8] for reviews). The MRA experimental design compares the steady states which result after performing independent perturbations with each "modular component" of a network. These perturbations might be genetic or biochemical. For example, in eukaryotes they might be achieved through the down-regulation of mRNA, and therefore protein, levels by means of RNAi, as done in [9]. That work employed MRA in order to quantify positive and negative feedback effects in the Raf/MEK (MAPK/ERK kinase)/ERK (extracellular-signal-regulated kinase) MAPK (mitogen-activated protein kinase) network in rat adrenal pheochromocytoma (PC-12) cells; using the algorithms from [3] and [4], the authors of [9] uncovered connectivity differences depending on whether the cells are stimulated with epidermal growth factor (EGF) or instead with neuronal growth factor (NGF).

This paper presents a mathematical synopsis of the MRA method.

Problem formulation

We assume that there are n quantities $x_i(t)$ that can be in principle measured, such as the levels of activity of selected proteins, or the transcription rates of certain genes. Later, we discuss how to weaken this assumption, so as to be able to include systems for which some variables are hidden from measurement. These quantities are thought of as state variables in a dynamical system, and are collected into a time-dependent vector $x(t) = (x_1(t),\ldots, x_n(t))$. The dynamical system is described by a system of differential equations:

$$\begin{aligned}\dot{x}_1 &= f_1(x_1,\ldots,x_n,p_1,\ldots,p_m)\\ \dot{x}_2 &= f_2(x_1,\ldots,x_n,p_1,\ldots,p_m)\\ &\vdots\\ \dot{x}_n &= f_n(x_1,\ldots,x_n,p_1,\ldots,p_m)\end{aligned}$$

or, in more convenient vector form,

$$\dot{x} = f(x,p)$$

(dot indicates time derivative, and arguments t are omitted when clear). We will assume that $m \geq n$. The p_i's are parameters, collected into a vector $p = (p_1,\ldots, p_m)$. These parameters can be manipulated, but, once changed, they remain constant for the duration of the experiment. An example would be that in which the variables x_i correspond to the levels of protein products corresponding to n genes in a network, and the parameters reflect translation rates, controlled by RNAi. Another example would be that in which the parameters represent total levels of proteins, whose half-lives are long compared with the time scale of the processes (such as phosphorylation modifications of these proteins in a signalling pathway) described by the variables x_i. Yet another example would be one in which the parameters represent concentrations of enzymes that control the reactions, and whose turnover is slow.

The ultimate goal is to obtain, for each pair of variables x_i and x_j, the relative signs and magnitudes of the partial derivatives

$$\frac{\partial f_i}{\partial x_j}$$

which quantify the *direct* effects of each variable x_j upon each variable x_i. These partial derivatives are the entries of the Jacobian matrix with respect to x of f (which is assumed to be continuous differentiable). See Figure 1.

The entries of $\partial f_i/\partial x_j$ of the Jacobian F are functions of x and p. The steady-state version of MRA attempts to estimate this Jacobian when $x = \bar{x}$ is an "unperturbed" steady state attained when the vector of parameters has an "unperturbed" value $p = \bar{p}$. The steady-state condition means that $f(\bar{x}, \bar{p}) = 0$. Ideally, one would want to find the matrix F, since this matrix completely describes the influence of each variable x_j upon the rate of change of each other variable x_i. Unfortunately, such an objective is impossible to achieve from only steady-state data, because, for any parameter vector p and associated steady-state x, $f(x, p) = 0$ implies that $\Lambda f(x, p) = 0$, for any diagonal matrix $\Lambda = \text{diag}\ (\lambda_1,\ldots, \lambda_n)$. In other words, the best that one could hope for is for steady-state data to uniquely determine each of the rows

$$F_i = (F_{i1},\ldots, F_{in}),\ \ i = 1,\ldots, n$$

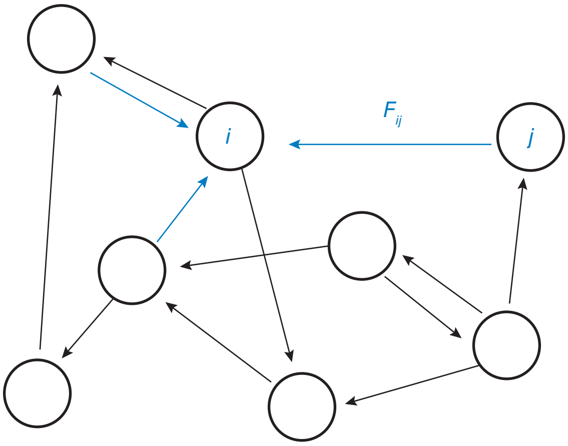

Figure 1. Network interaction graph
Shown in bold are the edges ending at node *i*, which correspond to the nonzero elements F_{ij} of the *i*th row of the Jacobian. The numerical value of F_{ij} reflects the interaction strength, quantifying how the rate of change of activity of node *i* depends on the activity of node *j*.

of *F only up to a scalar multiple*. For example, if we impose the realistic condition that $F_{ii} \neq 0$ for every i (these diagonal Jacobian terms typically represent degradation and/or dilution effects, and are in fact negative), one could hope to have enough data to estimate the ratios a_{ij}/a_{ii} for each $i \neq j$. Note that F_i is the same as the gradient ∇f_i of the ith coordinate f_i of f, evaluated at steady states.

The critical assumption for MRA, and indeed the main point of [1,3,10], is that, while one may not know the detailed form of the vector field f, often one does know which parameters p_j directly affect which variables x_i. For example, x_i may be the level of activity of a particular protein, and p_i might be the total amount (active plus inactive) of that particular protein; in that case, we know that p_i only directly affects x_i, and only indirectly affects the remaining variables.

The steady-state MRA experimental design consists of the following steps:

1. measure a steady state $\bar{x}$ corresponding to the unperturbed vector of parameters $\bar{p}$;
2. separately perform a perturbation to each entry of $\bar{p}$, and measure a new steady state.

The "perturbations" are assumed to be small, in the sense that the theoretical analysis will be based on the computation of derivatives. Under mild technical conditions, this means that a perturbed steady state can be found near $\bar{x}$. Note that there are $m+1$ experiments, and n numbers (coordinates of the corresponding steady state) are measured in each. In practice, of course, this protocol is repeated several times, so as to average out noise and obtain error estimates, as we discuss later. For our theoretical analysis, however, we assume ideal, noise-free measurements, and so we may assume that each perturbation is done only once.

Using these data (and assuming that a certain independence condition, which we review later, is satisfied), it is possible to calculate, at least in the ideal noise-free case, the Jacobian of f, evaluated at $(\bar{x}, \bar{p})$, except for the unavoidable scalar multiplicative factor uncertainty on each row.

The obtained results typically look as shown in Figure 2, which is reproduced with permission from [9]. The authors of that paper used MRA in order to infer positive and negative feedback effects in the Raf/MEK/ERK MAPK network in PC-12 cells, employing perturbations in which total mRNA, and thus protein, levels are down-regulated by means of RNAi. The numbers in the arrows in Figure 2 have been normalized to −1's in the diagonal of the Jacobian. An example of the raw data for the algorithm is provided by Figure 3, which shows the level of active (doubly phosphorylated) ERK1/2 when PC-12 cells have been stimulated by EGF and NGF. (The Figure shows only responses in the unperturbed case. Similar plots, not shown, can be derived from the data for the perturbation experiments given in [9].) The response to NGF stimulation allows the application of the steady-state MRA method, and leads to the results shown in the right-most panel in Figure 2.

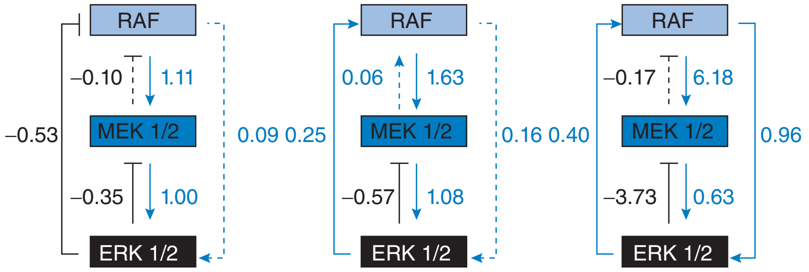

Figure 2. Three reconstructed local interaction maps, in MRA experiments from [9]
Topologies derived from data obtained after stimulation by EGF (left-hand panel, 5 min) or NGF (middle panel, 5 min, and right-hand panel, 15 min). Adapted from Santos, S.D.M, Verveer, P.J & Bastiaens, P.I.H. (2007) Growth factor induced MAPK network topology shapes Erk response determining PC-12 cell fate. *Nat. Cell Biol.* **9**, 324–330, with permission © 2007 Macmillan Publishers Ltd.

However, the plots in Figure 3 indicate that, in certain problems, steady-state data cannot be expected to provide enough information, even for only finding the Jacobian rows up to multiplicative factors. Such a situation occurs when the system *adapts* to perturbations. In Figure 3, notice that the steady-state response to EGF stimulation is (near) zero (this holds for perturbed parameters as well, not shown). Thus, measuring steady-state level of activity of ERK1/2 after parameter perturbations, in the EGF-stimulated cells, will not provide non-trivial information. One needs more than steady-state data. A variant of MRA, which allows for the use of general non-steady-state, time-series data was developed in [3]. However, that method requires one to compute

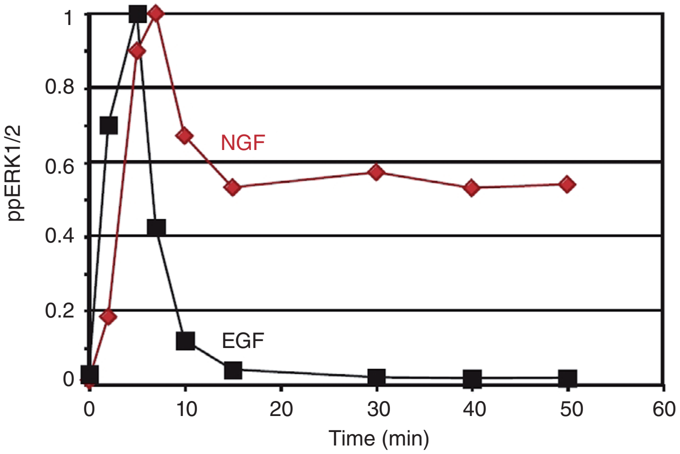

Figure 3. Active form of ERK1/2, in MRA experiments from [9]
Data shown only for unperturbed case. Adapted from Santos, S.D.M, Verveer, P.J & Bastiaens, P.I.H. (2007) Growth factor induced MAPK network topology shapes Erk response determining PC-12 cell fate. *Nat. Cell Biol.* **9**, 324–330, with permission © 2007 Macmillan Publishers Ltd.

second-order time derivatives, and hence is especially hard to apply when time measurements are spaced far apart and/or are noisy. In addition, as shown for 5 min and 15 min NGF stimulation by the middle and right-most panels in Figure 2, the relative strengths of functional interactions may change over time, so that a time-varying Jacobian may not be very informative from a biological standpoint. An appealing intermediate possibility is to use quasi-steady-state data, meaning that one employs data collected at times when a variable has been observed to attain a local maximum (peak of activity) or a local minimum. Indeed, this is the approach taken in [9], which, for EGF stimulation, measured network responses at the time of peak ERK activity (approximately 5 min), and not at steady state. The left-most and middle panels in Figure 2 represent, respectively, the networks reconstructed in [9] when using quasi-steady-state data (at approximately 5 min) for EGF and NGF stimulation.

We next describe both the steady-state and quasi-steady-state reconstruction problems.

Mathematical details

We assume given a parameter vector $\bar{p}$ and state $\bar{x}$ such that $f(\bar{x},\bar{p})=0$ and so that the following generic condition holds for the Jacobian of $f: F(\bar{x},\bar{p}) = \frac{\partial f}{\partial x}(\bar{x},\bar{p}) \neq 0$. Therefore, we may apply the implicit function theorem and conclude the existence of a mapping φ, defined on a neighbourhood of $\bar{p}$, with the property that, for each row i,

$$f_i(\varphi(p), p) = 0 \qquad \text{for all } p \approx \bar{p} \tag{1}$$

and $\varphi(\bar{p}) = \bar{x}$ (and, in fact, $x = \varphi(p)$ is the unique state x near $\bar{x}$ such that $f(x, p) = 0$).

We next discuss how one reconstructs the gradient $\nabla f_i(\bar{x},\bar{p})$, up to a constant multiple. (The index i is fixed from now on, and the procedure must be repeated for each row f_i.) We do this under the assumption that it is possible to apply $n-1$ independent parameter perturbations. Mathematically, the assumption is that there are $n-1$ indices $j_1, j_2,\ldots, j_{n-1}$ with the following two properties:

(a) f_i does not depend directly on any p_j: $\partial f_\mathrm{i}/\partial p_j \equiv 0$, for $j \in \{j_1, j_2,\ldots, j_{n-1}\}$, and

(b) the vectors $v_j = (\partial\varphi/\partial p_j)(\bar{p})$, for these j's, are linearly independent.

Assumption (a) is structural, and is key to the method and non-trivial, but assumption (b) is a weaker genericity assumption. We then have, taking total derivatives in (eqn 1):

$$\nabla f_i(\bar{x},\bar{p})\, v_j = 0, j \in \{j_1, j_2,\ldots, j_{n-1}\}.$$

Thus, the vector $\nabla\, f_i(\bar{x},\bar{p})$ which we wish to estimate, and which we will denote simply as F_i, is known to be orthogonal to the $n-1$ dimensional subspace spanned by $\{v_1,\ldots, v_{n-1}\}$. Therefore, it is uniquely determined, up to multiplication by a positive scalar. The row vector F_i satisfies

$$F_i\,\Sigma = 0 \tag{2}$$

where Σ is defined as the $n \times (n-1)$ matrix whose columns are the v_i's. Generically, we assume that there is no degeneracy, and the rank of Σ is $n-1$. Thus, F_i can be computed by using Gaussian elimination, as any vector which is orthogonal to the span of the columns of Σ. Another way to phrase this is to say that F_i is in the (one-dimensional) left nullspace of the matrix Σ, or that (if nonzero) the transpose of this gradient can be found as an (any) eigenvector associated to the zero eigenvalue of the transpose of Σ.

Handling noise

We next briefly discuss how to modify the algorithm to account for repeated but noisy measurements. In principle, such noise may be due to combinations of internal sources, such as stochasticity in gene expression, external sources affecting the process being studied, or measurement errors. Our discussion is tailored to measurement noise, although in an approximate way may apply to internal noise; however, the effect of internal noise on MRA has not been studied in any detail.

In practice, one would estimate not merely the results of just $n-1$ perturbation experiments, but many repetitions, collecting the data into a matrix $\Sigma^\#$ whose columns are derived from the different experiments. We will think of each column of $\Sigma^\#$ as having the form $v + e$, where v is a vector $(\partial\varphi/\partial p_j)(\bar{p})$, for some parameter p_j for which f_i does not depend directly on p_j, and where e is an "error" vector. In matrix notation, $\Sigma^\# = \Sigma + E$, where E denotes an error matrix. Note that eqn (2) implies that Σ has rank $n-1$. On the other hand, because of noise in measurements, $\Sigma^\#$ will have full rank n, which means that there is no possible nonzero solution F_i to eqn (2) with the data matrix $\Sigma^\#$ used in place of the (unknown) Σ. So, we proceed as follows. Assuming that the signal to noise ratio is not too large, the experimental matrix $\Sigma^\#$ should be close to the ideal (noise-free) matrix, Σ. The best least-squares estimate of Σ, in the sense of minimization of the norm of E, is obtained (Eckart-Young Matrix Approximation Theorem [11]) by a singular value decomposition (SVD) of $\Sigma^\#$, as follows. An SVD of $\Sigma^\#$ is a factorization $\Sigma^\# = UMV^T$, where U and V are (square) orthogonal matrices and M is a matrix (of the same size as $\Sigma^\#$) with non-negative numbers on the diagonal and zeros off the diagonal. The diagonal elements $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ of M are the singular values of $\Sigma^\#$, and we have that $\sigma_1,\ldots, \sigma_{n-1}$ are all nonzero, and $\sigma_n = 0$, in the ideal case, but in general σ_n will be nonzero and represents the magnitude of the noise. The columns of U and of V are, respectively, the left and right singular vectors of $\Sigma^\#$. Then (see details in e.g. [12], Appendix A, or in [13,14]), the matrix Σ of rank $n-1$ for which $\|E\|$ is minimized is $\Sigma = UM_{n-1}V^T$, where M_{n-1} is the matrix obtained from M by setting the smallest singular value σ_n to zero.

We now replace eqn (2) by $F_i\Sigma^\# = 0$, which, because V is nonsingular, is the same as $F_i\ U\ M_{n-1} = 0$. Under the generic assumption that $\sigma_1,\ldots, \sigma_{n-1}$ are nonzero, this means that $F_iU = \alpha e_n^T$, where α is a scalar and $e_n^T = (0, 0, \ldots, 0, 1)$. We then conclude that, up to a constant multiple, $F_i^T = Ue_n$ is the right singular vector corresponding to the smallest singular value σ_n.

This procedure can also be interpreted as follows (see [4] for details). If we normalize F_i to have its ith entry as "−1" (in other words, we normalize the diagonal of the Jacobian to −1's), then the equation $F_i\Sigma^\# = 0$ can also be written as "$Az = b$" where z represents the unknown n−1 remaining entries of F_i, b is the ith column of $\Sigma^\#$, and A is the matrix in which this column has been removed from $\Sigma^\#$. The estimation method outlined above is the "total least squares" or "errors in variables" procedure [13,14]. Statistically, the method is justified if the elements of the noise matrix E are independent and identically distributed normal (Gaussian) random variables. If these entries are normal and independent but have different variances, then one must modify the above procedure to add an appropriate weighting, but in the general non-Gaussian case nonlinear SVD techniques are required; see for instance [15–17].

Modular approach

Modularity is an important feature of biochemical networks [18,19]. As its name implies, one of the main advantages of the steady-state MRA method is that only "communicating intermediates" in-between "modules" need to be measured. When applying MRA in a modular fashion, only perturbation data on communicating intermediaries are collected, The connectivity strength among a pair of intermediates (such as activated levels of MAPK cascade proteins) can be derived, even if this apparent connectivity it is not due to a direct biochemical interaction. An obvious advantage of the modular approach is that it can be applied regardless the degree of complexity of the network analysed, as "hidden" variables (such as non-activated forms of a MAPK protein, for example) only affect connectivity in an indirect fashion. Thus, functional interactions among communicating variables can be deduced without requiring detailed knowledge of all the components involved.

Let us suppose that the entire network consists of an interconnection of n subsystems or "modules", each of which is described by a set of differential equations such as:

$$\begin{aligned} \dot{x}_j &= g_j(\mathbf{y}_j, x_1, \ldots, x_n, p_1, \ldots, p_m), \qquad j = 1, \ldots, n \\ \dot{\mathbf{y}}_j &= G_j(\mathbf{y}_j, x_1, \ldots, x_n, p_1, \ldots, p_m), \qquad j = 1, \ldots, n \end{aligned} \tag{3}$$

where the variables x_j represent "communicating" or "connecting" intermediaries of module j that transmit information to other modules, whereas the *vector variables* $\mathbf{y}_j$ represent chemical species that interact within module j. Each vector $\mathbf{y}_j$ has dimension ℓ_j. The integers $\ell_j, j = 1, \ldots, n$ are in general different for each of the n modules, and they represent one less than the number of chemical species in the jth module respectively. Observe that, for each j, the rate of change of the communicating variable depends only on the remaining communicating variables x_i, $i \neq j$, and on the variables $\mathbf{y}_j$ it its own block, but does not directly depend on the internal variables of other blocks. In that sense, we think of the variables $\mathbf{y}_j$ as "hidden" (except from the communicating variable in the same block).

We will assume, for each fixed module, that the Jacobian of G_j with respect to the vector variable $\mathbf{y}_j$, evaluated at the steady state corresponding to $\bar{p}$ (assumed to exist, as before) is nonsingular. The Implicit Mapping Theorem then implies that one may, in a neighbourhood of this steady state, solve

$$G_j(\mathbf{y}_j, x_1, \ldots, x_n, p_1, \ldots, p_m) = 0$$

for the vector variable $\mathbf{y}_j$, as a function of $x_1, \ldots, x_n, p_1, \ldots, p_m$, the solution being given locally by a function

$$\mathbf{y}_j = \mathbf{M}_j(x_1, \ldots, x_n, p_1, \ldots, p_m).$$

Then, those steady states obtained by small perturbations of $\bar{p}$ are precisely the same as the steady states of the new system

$$\dot{x}_j = h_j(x_1, \ldots, x_n, p_1, \ldots, p_m), \qquad j = 1, \ldots n$$

where we defined $h_j(x_1, \ldots, x_n, p_1, \ldots, p_m)$ as:

$$g_j(\mathbf{M}_j(x_1, \ldots, x_n, p_1, \ldots, p_m), x_1, \ldots, x_n, p_1, \ldots, p_m)$$

From here, the analysis then proceeds as before, using the h_j's instead of the f_j's. A generalization to the case of more than one communicating intermediate in a module, namely a vector $(x_{j,1}, \ldots, x_{j,}k_j)$, is easy.

Using quasi-steady-state data

Recalling the discussion in the Introduction regarding the need for a method based on quasi-steady-state data, we next consider the following scenario. For any fixed variable, let us say the ith component x_i of x, we consider some time instant t_i at which $\dot{x}_i(t)$ is zero. Under the same independence hypothesis as in the steady-state case, plus the non-degeneracy assumption that the second time derivative $\ddot{x}_t(\bar{t}_i)$ is not zero (so that we have a true local minimum or local maximum, but not an inflection point), we show here that the MRA approach applies in exactly the same manner as in the steady-state case. Specifically, the ith row of the Jacobian of f, evaluated at the vector $(\bar{x}, \bar{p})$, is recovered up to a constant multiple, where $\bar{x} = x(\bar{t}_i)$ is the full state x at time $\bar{t}_i$. The main difference with the steady-state case is that different rows of f are estimated at different pairs $(\bar{x}, \bar{p})$, since the considered times $\bar{t}_i$ at which each individual

$\dot{x}_i(t)$ vanishes are in general different for different indices i, and so the state $\bar{x}$ is different for different i's.

We fix an index $i \in \{1, \ldots, n\}$, and an initial condition $x(0)$, and assume that the solution $x(t)$ with this initial condition and a given parameter vector $\bar{p}$ has the property that, for some time $\bar{t} = \bar{t}_i$, we have that both $\dot{x}_i(t) = 0$ and $\ddot{x}_i(t) \neq 0$. At the instant $t = \bar{t}_i$, x_i achieves a local minimum or a local maximum as a function of t. We describe the reconstruction of the ith row of the Jacobian of f, which that is, the gradient ∇f_i, where f_i is the ith coordinate of f, evaluated at $x = \bar{x}$ and $p = \bar{p}$, where $\bar{x} = (\bar{t})$.

To emphasize the dependence of the solution on the parameters [the initial condition $x(0)$ will remain fixed], we will denote the solution of the differential equation $\dot{x} = f(\bar{x}, \bar{p})$ by $x(t, p)$. The function $x(\mathrm{t}, p)$ is jointly continuously differentiable in x and p, if the vector field f is continuously differentiable (see e.g. [12], Appendix C). Note that, with this notation, the left-hand side of the differential equation can also be written as $\partial x/\partial t$, and that $x(\bar{t}, \bar{p}) = \bar{x}$.

We introduce the following function:

$$\alpha(t,p) = \frac{\partial x_i}{\partial t}(t,p) = f_i(x(t,p),p)$$

Note that $\alpha(\bar{t}, \bar{p}) = 0$. Also,

$$\frac{\partial \alpha}{\partial t}(t,p) = \frac{\partial^2 x_i}{\partial t^2}(t,p) = \nabla f_i(x(t,p),p) f(x(t,p),p).$$

The assumption that $\ddot{x}_i(\bar{t}) \neq 0$ when $p = \bar{p}$ means that $\frac{\partial \alpha}{\partial t}(t,p) \neq 0$. Therefore, we may apply the implicit function theorem and conclude the existence of a mapping τ, defined on a neighbourhood of $\bar{p}$, with the property that

$$\alpha(\tau(p), p) = 0 \text{ for all } p \approx \bar{p}$$

and $\tau(\bar{p}) = \bar{t}$ [and, in fact, $t = \tau(p)$ is the unique value of t near $\bar{t}$ such that $(\partial x_i/\partial t)(t, p) = \alpha(t, p) = 0$]. Finally, we define, also in a neighbourhood of $\bar{p}$, the differentiable function

$$\varphi(p) = x(\tau(p), p)$$

and note that $\varphi(\bar{p}) = \bar{x}$. Observe that, from the definition of α, we have that eqn (1) holds, exactly as in the steady-state case. From here, the reconstruction of $\nabla f_i(\bar{x}, \bar{p})$ up to a constant multiple proceeds as in the steady-state case, again under the assumption that it is possible to apply $n-1$ independent parameter perturbations.

A noise analysis similar to that in the steady-state case can be done here. However, there are now many more potential sources of numerical and experimental error, since measurements at different times are involved. In addition, internal (thermal) noise may introduce additional error, since, in the quasi-steady-state case, the state probability distributions (solutions of the Chemical Master Equation) have not converged to steady state.

Implementation: numerical approximation by finite differences

Of course, the sensitivities represented by the vectors v_i (entries of the matrix Σ, or $\Sigma^{\#}$ in the noisy case) cannot be directly obtained from typical experimental data. However, approximating the vectors v_j by finite differences, one has that $\nabla f_i(\bar{x},\bar{p})$ is approximately orthogonal to these differences as well. Explicitly, suppose that we approximate $v_j = (\partial(\varphi/\partial p_j)(\bar{p})$ by:

$$\frac{1}{h}(\varphi(\bar{p}+he_j)-\bar{x})$$

where h is small and where e_j is the vector having a one in the jth position and zeros elsewhere. Then, $\nabla f_i(\bar{x},\bar{p})$ is (approximately) orthogonal to the differences

$$\varphi(\bar{p}+he_j)-\bar{x}$$

which form a set of $n-1$ linearly independent vectors (if h is small). A simple matrix inversion (after fixing an arbitrary value for one of its entries) allows the computation of $\nabla f_i(\bar{x},\bar{p})$. Observe that division by the potentially small number h is not required in performing these operations, In fact, no knowledge whatsoever about parameter values is needed by the algorithm. In the quasi-steady-state case, note that $\varphi(\bar{p}+he_j)$ is the state $x(t)$ at the time t at which *the particular coordinate* x_i achieves a local extremum value, if the parameters have been perturbed to $p = \bar{p} + he_j$. To be more precise, t is the unique time close to $\bar{t}$ such that $\dot{x}_i(t) = 0$ when parameter vector p is being used. Theoretically, we must have $p \approx \bar{p}$ so h must be very small, but, in practice, quite large perturbations of p also work fine.

A simple numerical example

We illustrate the calculations by working out a very simple example, the following system (writing x instead of x_1 and y instead of x_2, and p and q instead of p_1 and p_2):

$$\begin{aligned}\dot{x} &= -3qx + \tfrac{10}{1+y}\\ \dot{y} &= px + 1 - 3y.\end{aligned}$$

We take the initial state $(0,0)$ and reference parameters $\bar{p} = 2$ and $\bar{q} = 1$. These equations might model, for example, a simplified dynamics of a two-gene network, in which the first gene enhances the expression of the second gene, which in turn represses the rate of expression of the first one, there are constitutive rates of production of each gene, and both protein products decay at rate 3 s^{-1}. (For simplicity, we do not introduce separate variables for mRNA and protein concentrations.) Parameter p represents a promoter strength, and we assume that there is a way to perturb it (perhaps by duplication or sequence change). Parameter q affects the degradation of the first gene product.

Figure 4 shows (solid lines) the solution $[x(t), y(t)]$ with initial state $(0,0)$, $p = \bar{p} = 2$ and $q = \bar{q} = 1$. By time $t = 2.5$, the system has settled approximately to steady state, $(x(2.5), y(2.5)) \approx (1.45, 1.3)$. At this point, the true Jacobian, to be estimated, is approximately:

$$\begin{pmatrix} -3 & -1.9 \\ 2 & -3 \end{pmatrix}$$

In order to estimate the first row of the Jacobian, which does not depend on the parameter p, we next perturb to $p = 4$ (keeping $q = 1$; see Figure 4). We now have $(x(2.5), y(2.5)) \approx (1.16, 1.88)$. Thus we are looking for a vector orthogonal to $(1.16, 1.88) - (1.45, 1.3)$, which provides an estimated first row of approximately $(-3, -1.5)$. (We normalized the first entry to -3 merely in order to compare our result with the true gradient; the algorithm does not know the value "-3". In practice, one might know that the first entry of the vector is negative, reflecting degradation or dilution effects, so the algorithm will give the correct sign for the second term, as well as its magnitude relative to the rate of degradation or dilution.) Observe that we have retrieved the correct sign for

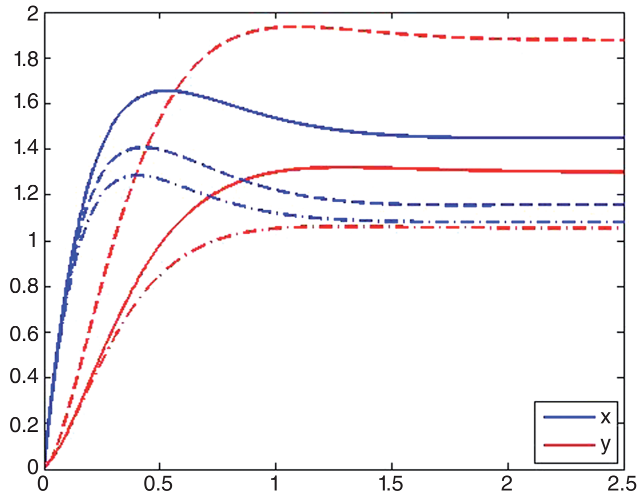

Figure 4. Trajectories
$p = 2, q = 1$ (solid); $p = 4, q = 1$ (dash); $p = 2, q = 1.5$ (dash-dot).

the second element, and the relative error is less than 20%, even though we have used a 100% perturbation. To estimate the second row of the Jacobian, one perturbs q, and the results are even better. For example, a 50% perturbation to $q = 1.5$ (keeping $p = 2$; see Figure 4) gives $(x(2.5), y(2.5)) \approx (1.09, 1.05)$, which leads to perfect recovery (to two decimal digits) of the second row.

Next we illustrate the quasi-steady version of the algorithm, using the same example. We study the equation for the first variable, which is the one that has a peaking behaviour. The solid lines in Figure 5 (and also in Figures 6 and 7) again show plots of the solution coordinates $x(t)$ and $y(t)$, but now zooming-in on an initial interval where peaks occur.

Let us pose the following problem: not knowing the above equations, estimate the relative strength of the second gene's effect on the rate of expression of the first one. The only data to be used are the levels of both gene products [$x(t)$ and $y(t)$] at the time when $x(t)$ achieves its local maximum. We do assume known the fact that the parameter p affects *directly* only the rate of expression of the second gene, not the first. Observe that the maximum of x is attained at $t \approx 0.5275$, and the values there are (approximately) $x(t) = 1.6553$ and $y(t) = 1.0138$. The gradient ∇f_1 of $-3x+\frac{10}{1+y}$, evaluated at (1.407, 1.3695), has the true (but unknown to the algorithm) value:

$$\left(-3, -\left.\frac{10}{(1+y)^2}\right|_{y=1.0138}\right) \approx (-3, -2.4659).$$

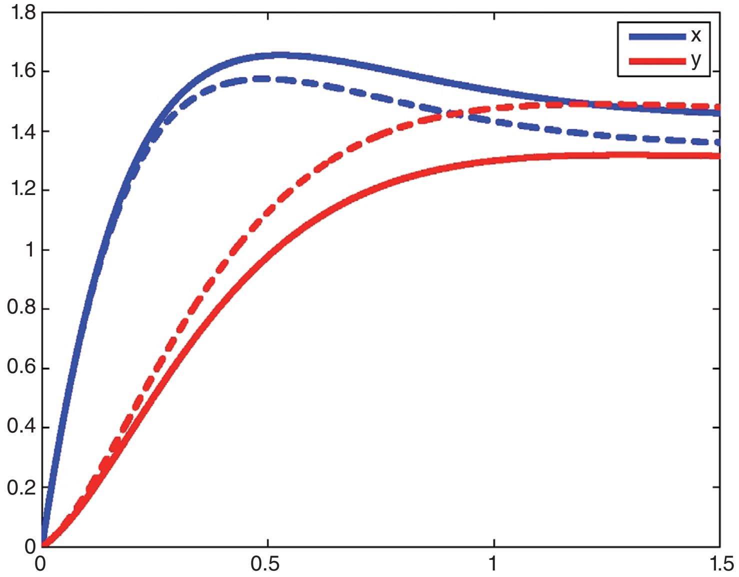

Figure 5. Trajectories
Dashed is perturbed motion with 25% change in parameter.

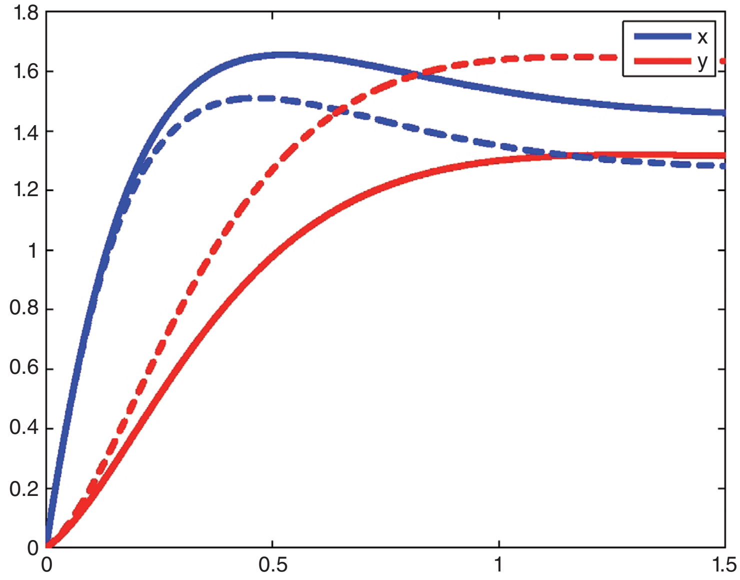

Figure 6. Trajectories
Dashed is perturbed motion with 50% change in parameter.

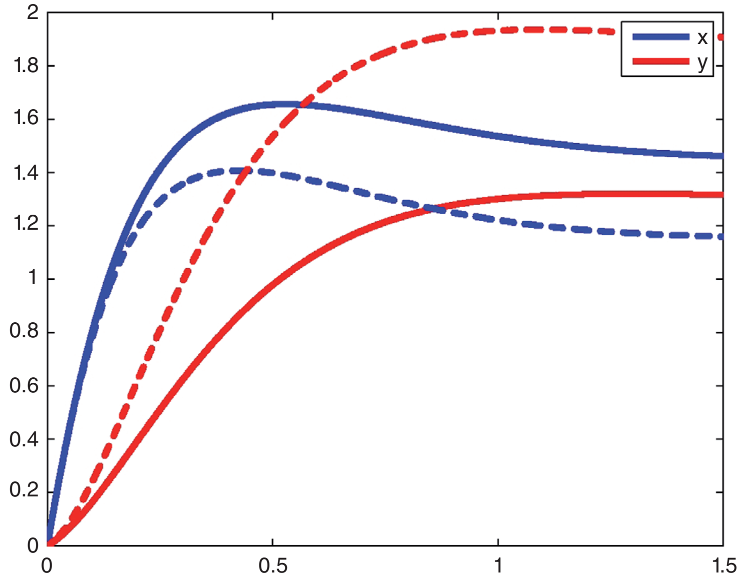

Figure 7. Trajectories
Dashed is perturbed motion with 100% change in parameter.

Next, we perform the "experiment" in which p is up-perturbed by 25%. With the new parameter $p = 2.5$, we obtain plots as shown by the dotted lines in Figure 5. Now the maximum of x is attained at $t \approx 0.4268$, and the values there are $x(t) = 1.407$ and $y(t) = 1.3695$. Letting $\delta = (1.407, 1.3695) - (1.6553, 1.0138)$, the unknown (to the algorithm) gradient ∇f_1 is known to be (approximately) orthogonal to δ. Any vector perpendicular to δ must be a multiple of $(-3, -2.3455)$. The relative error in our estimate is less than 5%.

Even larger perturbations may be performed. For example, a 50% perturbation from $\bar{p} = 2$ to $p = 3$, provides the dashed lines in Figure 6. Now the maximum for x is attained at $t \approx 0.4658$, and there $x(t) = 1.5103$ and $y(t) = 1.2073$. The estimated gradient is now $(-3, -2.2476)$, which gives a relative error of less than 9%. Finally, a 100% perturbation to $p = 4$ provides the dashed lines in Figure 7. Now the maximum for x is attained at $t \approx 0.4268$, and there $x(t) = 1.4071$ and $y(t) = 1.3695$. The estimated gradient is now $(-3, -2.0936)$, which gives a relative error of about 15%.

Conclusions

The MRA method provides a tool for reconstructing network topology, and relative input strengths, from steady-state and quasi-steady-state data. Much future work remains to be done, especially concerning the analysis of noise.

Summary

- *Local interactions can be reverse engineered from global responses, so long as enough independent perturbations can be performed.*
- *The same method can be applied in a modular fashion.*
- *Measurement errors are dealt with by a total least squares procedure.*

This research was supported in part by NSF (National Science Foundation) grants DMS-0504557 and DMS-0614371, and an AFOSR (Air Force Office of Scientific Research) grant.

References

1. Kholodenko, B.N., Kiyatkin, A., Bruggeman, F., Sontag, E.D., Westerhoff, H. & Hoek, J. (2002) Untangling the wires: a novel strategy to trace functional interactions in signaling and gene networks. *Proc. Natl. Acad. Sci. U.S.A.* **99**, 12841–12846
2. de la Fuente, A., Brazhnik, P. & Mendes, P. (2002) Linking the genes: inferring quantitative gene networks from microarray data. *Trends Genet.* **18**, 395–398
3. Sontag, E.D., Kiyatkin, A. & Kholodenko, B.N. (2004) Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data. *Bioinformatics* **20**, 1877–1886
4. Andrec, M., Kholodenko, B.N., Levy, R.M. & Sontag, E.D. (2005) Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy. *J. Theor. Biol.* **232**, 427–441

5. Berman, P., Dasgupta, B. & Sontag, E.D. (2007) Algorithmic issues in reverse engineering of protein and gene via the modular response analysis method. *Ann. N.Y. Acad. Sci.* **1115**, 132–141
6. Berman, P., Dasgupta, B. & Sontag, E.D. (2007) Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks. *Discrete Appl. Math. Special Ser. Comp. Mol. Biol.* **155**, 733–749
7. Stark, J., Callard, R. & Hubank, M. (2003) From the top down: towards a predictive biology of signalling networks. *Trends Biotechnol*. **21**, 290–293
8. Crampin, E.J., Schnell, S. & McSharry, P.E. (2004) Mathematical and computational techniques to deduce complex biochemical reaction mechanisms. *Prog. Biophys. Mol. Biol.* **86**, 77–112
9. 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
10. Kholodenko, B.N. & Sontag, E.D. (2002) Determination of functional network structure from local parameter dependence data. Technical report, arXiv physics/0205003
11. Eckart, C. & Young, G. (1936) The approximation of one matrix by another of lower rank. *Psychometrika* **1**, 211–218
12. Sontag, E.D. (1998) Mathematical Control Theory: Deterministic Finite Dimensional Systems, Springer-Verlag, New York
13. Golub, G.H. & Van Loan, C.F. (1980) An analysis of the total least squares problem. *SIAM J. Numer. Anal.* **17**, 883–893
14. Van Huffel, S. & Vandewalle, J. (1991) The Total Least Squares Problem: Computational Aspects and Analysis, volume 9 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA
15. Gerhold, G.A. (1969) Least-squares adjustment of weighted data to a general linear equation. *Am. J. Phys.* **37**, 156–161
16. De Moor, B. (1993) Structured total least squares and L_2 approximation problems. *Linear Algebra Appl.* **188/189**, 163–205
17. Premoli, A. & Rastello, M.L. (2002) The parametric quadratic form method for solving TLS problems with elementwise weighting. In Total Least Squares and Errors-in-variables Modeling (Leuven, 2001) (van Huffel, S. & Lemmerling, P., eds), pp. 67–76. Kluwer Academic Publishing., Dordrecht
18. Del Vecchio, D., Ninfa, A.J. & Sontag, E.D. (2008) Modular cell biology: retroactivity and insulation. *Mol. Syst. Biol.* **4**, 161
19. Sontag, E.D. (2007) Monotone and near-monotone biochemical networks. *Syst. Synth. Biol.* **1**, 59–87