BIOINFORMATICS ORIGINAL PAPER
Vol. 28 no. 1 2012, pages 76–83
doi:10.1093/bioinformatics/btr620

Systems biology
Advance Access publication November 9, 2011

Decompositions of large-scale biological systems based on dynamical properties

Nicola Soranzo[1], Fahimeh Ramezani[2], Giovanni Iacono[3] and Claudio Altafini[3,*]
[1]CRS4 Bioinformatica, Loc. Piscina Manna, 09010 Pula (CA), Italy, [2]Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany and [3]SISSA, via Bonomea 265, 34136 Trieste, Italy

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Given a large-scale biological network represented as an influence graph, in this article we investigate possible decompositions of the network aimed at highlighting specific dynamical properties.

Results: The first decomposition we study consists in finding a maximal directed acyclic subgraph of the network, which dynamically corresponds to searching for a maximal open-loop subsystem of the given system. Another dynamical property investigated is strong monotonicity. We propose two methods to deal with this property, both aimed at decomposing the system into strongly monotone subsystems, but with different structural characteristics: one method tends to produce a single large strongly monotone component, while the other typically generates a set of smaller disjoint strongly monotone subsystems.

Availability: Original heuristics for the methods investigated are described in the article.

Contact: altafini@sissa.it

Received on September 5, 2011; revised on October 20, 2011; accepted on November 5, 2011

1 INTRODUCTION

One of the outstanding challenges that Systems Biology is currently facing is to provide the right tools for the investigation of the dynamical behavior of the large-scale networks used to represent complex biological systems, such as gene regulatory networks, signaling pathways and chains of metabolic reactions. Even if our knowledge of the interactions among the molecular species involved in these systems is growing at a fast pace, the details of the dynamics that they describe are seldom available and often unlikely to be obtainable in a near future. What is often more plausible to assume is that only an influence graph is available for these networks (Fages and Soliman, 2008; Klamt *et al*., 2006). An influence graph is a signed graph where an edge represents the action of a variable on another variable, and the signs may have the meaning of activatory/inhibitory action, or may simply represent the signature of the Jacobian linearization of a non-linear vector field which is unknown but sign constant over the entire state space (common forms of the kinetics, such as mass action and Michaelis–Menten, normally obey to this condition). In choosing this level of detail for our networks, we are guided by an abundant literature, see e.g. Fages and Soliman (2008); Huber *et al*. (2007); Klamt *et al*. (2006); Milo *et al*. (2002); Papin *et al*. (2005); Shen-Orr *et al*. (2002); Thieffry (2007). Important dynamical problems that can be investigated on an influence graph include:

(1) compute the equilibria of the system (Soulé, 2003);
(2) investigate the stability properties of the dynamics (Deangelis *et al*., 1986; Quirk and Ruppert, 1965);
(3) identify the largest open-loop subsystem of a given system (Ispolatov and Maslov, 2008);
(4) study the monotonicity and strong monotonicity properties of the dynamics (Sontag, 2007); and
(5) select a minimal intervention set for medical treatment (Klamt *et al*., 2006).

In this article, we are interested in the problems (3) and (4) of the list above.

In graph theoretical terms, finding the largest open-loop subsystem corresponds to identifying a maximum-size *directed acyclic graph* (DAG) within a network by dropping all feedback loops. In the computer science literature, this is called the *minimum feedback arc set* problem, and it is well known to be NP-hard (Karp, 1972). Although several heuristic methods are already available for it (Festa *et al*., 1999; Ispolatov and Maslov, 2008), the novel algorithm we propose in this article has the advantage that available *a priori* knowledge on the open-loop part of the system can be easily taken into account when computing a maximal DAG. We will show in the large-scale examples of Section 6 that the performances of our algorithm are comparable to those of the best heuristics.

In a series of papers by E. Sontag and colleagues (DasGupta *et al*., 2007; Ma'ayan *et al*., 2008; Sontag, 2007), it was shown that influence graphs can be used to study an important property of dynamical systems, namely monotonicity (Kunze and Siegel, 1994, 1999; Smith, 1988, 1995; Sontag, 2007). Monotone systems have nice properties of 'order' in their dynamical behavior. For example, they neither admit stable periodic orbits nor chaotic behavior. Moreover, for strongly monotone systems [i.e. monotone systems whose graph is irreducible, see Smith (1995); Sontag (2007)], Hirsch theorem states that almost all bounded solutions converge to the set of equilibria (Hirsch, 1983). The concept is particularly attracting for biological networks, because it is well known that these systems, though complex, have indeed outstanding stability properties, are largely devoid of spurious sustained oscillations and are definitively not chaotic. Hence, the paradigm of monotonicity has gained some momentum in recent years and there is by now a consistent literature on using these

*To whom correspondence should be addressed.

properties to study biological networks (DasGupta *et al.*, 2007; Iacono and Altafini, 2010; Iacono *et al.*, 2010; Ma'ayan *et al.*, 2008; Sontag, 2007).

Both monotonicity and strong monotonicity admit a graphical characterization: a system is monotone when all undirected cycles of its influence graph have positive sign (i.e. have an even number of negative edges); an irreducible system is strongly monotone when the same property holds for directed cycles (Sontag, 2007). While strong monotonicity implies monotonicity, the opposite implication is usually not true. For the stricter notion of strong monotonicity, the only study on large-scale biological networks we are aware of is Aswani *et al.* (2009).

In this article we propose two different methods aimed at extracting strongly monotone subsystems from large-scale influence graphs. The first method is based on the minimization of the total number of negative signs on the edges by means of 'switching equivalences' (Zaslavsky, 1982), i.e. changes in the direction of some of the axes of $\mathbb{R}^n$ in order to align the system as much as possible with the positive orthant of $\mathbb{R}^n$. This idea was developed in Iacono *et al.* (2010) for the monotonicity property and is extended here to the strong monotonicity properties.

The second method to decompose a network into strongly monotone subsystems relies instead on the notion of DAG introduced above. When on an open-loop subsystem represented as a DAG we start reinserting back the edges of the original network (i.e. the feedback loops for the original system), then strongly connected subgraphs begin to form. As long as all directed cycles of one of the strongly connected subgraphs have positive sign, then the corresponding subsystem will be strongly monotone.

In order to test the efficacy of the proposed algorithms, a number of large-scale biological networks are decomposed and their strongly monotone subsystems are identified. On these examples, the two methods we are proposing tend to highlight different features: a single large strongly monotone subnetwork is obtained in one case, and several medium-size strongly monotone subsystems in the other. Depending on the context, each of these approaches may be of help in better understanding the global structure of large systems and in investigating more properly their dynamical properties.

The organization of this article is as follows: the necessary background material is introduced in Section 2, and the construction of a maximal DAG is discussed in Section 3. The two methods for strong monotonicity decomposition are presented in Section 4. A small example [a yeast cell-cycle model from Li *et al.* (2004)] is studied in detail in Section 5 and finally, in Section 6, the algorithms are applied to large-scale biological networks.

2 BACKGROUND MATERIAL

2.1 Signed graphs

A basic reference for this Section is Deo (1974). A signed directed graph is an ordered pair $G=(V,E)$ where V is a set of vertices of cardinality $n=|V|$, and E is a set of signed edges $\ell_{i,j}\in\{\pm 1\}$ of cardinality $m=|E|$. A pair of edges $\ell_{i,j}$ and $\ell_{j,i}$ connecting the same vertices but of opposite direction is called a digon. When for all digons $sign(\ell_{i,j})=sign(\ell_{j,i})$, then we say that G admits an undirected graph (obtained by dropping all arrows in the edges). The sign of a path/cycle of G is positive (respectively negative) if it has an even (respectively odd) number of negative edges. We will denote $\mathcal{R}(v_i)\subseteq V$ the set of vertices reachable from v_i. An undirected (respectively directed) graph G is connected (respectively strongly connected) if any vertex is reachable from any vertex of G. In an undirected (respectively directed) graph G, a connected component (respectively strongly connected component, henceforth SCC) of G is a maximal connected (respectively strongly connected) subgraph of G. Given an undirected graph $G=(V,E)$, a spanning forest $T=(V,E_T)$ is a maximal acyclic subgraph of G. The number of edges of every spanning forest of G is equal to $|V|$ minus the number of connected components of G.

DAGs: a DAG is a directed graph without any directed cycle. When a DAG lacks also undirected cycles then it is called a polytree. Polytrees are typically obtained by considering a spanning forest T on the undirected graph of G and then restoring the original direction of the edges of T (dropping one of the arrows of each digon). For a directed graph G, a feedback arc set is a subset of edges whose removal from G leaves a DAG. A feedback arc set of G is minimal if no proper subset of it is a feedback arc set. A subgraph of G is a maximal DAG of G if it is the complement to a minimal feedback arc set of G.

Irreducible adjacency matrices and SCCs: denote A the signed adjacency matrix of a signed graph G. For simplicity of notation, we shall indicate $G(A)$ the graph obtained in correspondence of A, while $B\subseteq A$ will denote the adjacency matrix of the subgraph $G(B)$ of a graph $G(A)$. An $n\times n$ matrix A is reducible if $\exists$ a permutation matrix P s.t. $PAP=\begin{bmatrix} A_1 & A_2\\ 0 & A_3\end{bmatrix}$, with A_1, A_3 square submatrices. A is said irreducible if it is not reducible. A is irreducible if and only if the associated graph is strongly connected. For a non-strongly connected graph, finding the irreducible diagonal blocks of the matrix is equal to determining all the SCCs of the graph. Such operation can be carried out efficiently by e.g. the Tarjan algorithm (Tarjan, 1972). A directed graph $G(B)$, $B\subseteq A$, is a DAG if and only if $\exists$ a permutation matrix P such that PBP is upper triangular, see Deo (1974), Theorem 9.16. In other words, the adjacency matrix of a DAG is completely reducible.

2.2 Monotone dynamical systems

Dynamical systems and their signed influence graphs: consider the autonomous dynamical system

$$\dot{x}=f(x), \qquad x\in X\subseteq\mathbb{R}^n, \qquad f\in C^1(X), \tag{1}$$

and its linearization around an equilibrium point x_o, $\dot{z}=\mathcal{A}z$, where $\mathcal{A}=\left.\frac{\partial f(x)}{\partial x}\right|_{x=x_o}$, and $z=x-x_o$ is the vector of perturbations around x_o (signed, i.e. whose components z_i can assume both positive and negative values). In the context of large-scale biological networks, it is very difficult to have a precise knowledge of the functional form of the vector field $f(\cdot)$ or even of the Jacobian matrix $\mathcal{A}$. It is often more reasonable to assume that only the sign pattern is known of $\mathcal{A}$, i.e. $A=sign(\mathcal{A})$ has non-zero entries of unit amplitude $A_{ij}\in\{\pm 1,0\}$. A is the signed adjacency matrix of the so-called influence graph $G(A)$ of the network (Fages and Soliman, 2008; Klamt *et al.*, 2006), i.e. of the directed graph representing the effect of the j-th variable on the i-th variable, which can be activatory, $A_{ij}>0$, inhibitory, $A_{ij}<0$, or non-existent, $A_{ij}=0$. In general, in a system like (1), $\left(\left.\frac{\partial f(x)}{\partial x}\right|_{x=x_o}\right)_{ij}$ can change of sign with the operating point x_o, but we shall not consider this scenario here. In other words, we assume that the partial derivatives are sign constants, i.e. the sign patterns of $\left.\frac{\partial f(x)}{\partial x}\right|_{x=x_o}$ and $\left.\frac{\partial f(x)}{\partial x}\right|_{x=x_1}$ are the same for all x_o, x_1 in X. Conventionally, the self edges of the influence graph $G(A)$, i.e. the diagonal elements of A are disregarded when looking at monotonicity properties (Sontag, 2007). We shall tacitly assume this henceforth. The system (1) is said irreducible if A is irreducible. When $G(A)$ is a DAG then the system is completely reducible, i.e. A is triangular up to a permutation.

Monotonicity, strong monotonicity, and their graphical characterization: for a thorough introduction to the theory of monotone systems, the reader is referred to Kunze and Siegel (1999); Smith (1988, 1995); Sontag (2007). In $\mathbb{R}^n$, consider the cone K representing one of its orthants: $K=\{x\in\mathbb{R}^n \text{ such that } Dx\geqslant 0\}$ where D is a diagonal matrix $D=\text{diag}(\sigma)$ of diagonal elements $\sigma=(\sigma_1,\dots,\sigma_n)$, $\sigma_i\in\{\pm 1\}$, and denote by $\phi_t(x_1)$ the integral curve of (1) at time t in correspondence of the initial condition x_1. The system (1) is said *monotone* with respect to the partial order σ if $\forall x_1,x_2\in X$ such that $x_2-x_1\in K$ one has $\phi_t(x_2)-\phi_t(x_1)\in K\ \forall t\geqslant 0$. Likewise, the system (1) is said *strongly monotone* with respect to the partial order σ if $\forall x_1,x_2\in X$ such that $x_2-x_1\in K$, $x_2\neq x_1$, one has $\phi_t(x_2)-\phi_t(x_1)\in\text{int}(K)\ \forall t>0$ ($\text{int}(\cdot)$ is the interior of the cone). Monotonicity can be formulated in terms of the adjacency matrix A by means of the so-called Kamke condition, which states that the system (1) is monotone in X with respect to the orthant order σ if and only if

$$\sigma_i\sigma_j A_{ij}\geqslant 0 \qquad \forall i,j=1,\dots,n \text{ s. t. } i\neq j. \tag{2}$$

The starting point of our investigation is a graphical condition for orthant monotonicity. Assume that $G(A)$ admits an undirected graph, i.e. that all edge pairs of the digons of $G(A)$ have compatible signs, $A_{ij}A_{ji}\geqslant 0$. Denote A_U the adjacency matrix of the undirected graph obtained from $G(A)$. The following Lemma can be found in e.g. Sontag (2007).

LEMMA 1. *The system* (1) *is monotone in X with respect to some orthant order σ if and only if any of the following conditions holds:*

(1) *$\exists\,\sigma$ and a matrix $D=\text{diag}(\sigma)$ such that all off-diagonal entries of DA_UD are non-negative;*

(2) *all cycles of $G(A_U)$ have positive sign.*

The non-strict inequality in (2) implies that monotonicity is concerned not only with 'true' directed cycles and their sign, but also for example with 'parallel' directed paths starting and ending on the same nodes (and forming cycles on the undirected graph $G(A_U)$), see Iacono *et al.* (2010); Sontag (2007). The restriction to directed cycles is necessary when we are interested in strong monotonicity properties. A sufficient condition for strong monotonicity of a monotone system is the irreducibility of the system. From Lemma 1, we have the following graph-theoretical condition [see Smith (1995) and Sontag (2007)].

LEMMA 2. *Assume that the system* (1) *is irreducible in X. The system* (1) *is strongly monotone with respect to some orthant order σ if and only if any of the following conditions holds:*

(1) *$\exists\,\sigma$ and a matrix $D=\text{diag}(\sigma)$ such that all off-diagonal entries of DAD are non-negative;*

(2) *all directed cycles of $G(A)$ have positive sign.*

3 CONSTRUCTION OF A MAXIMAL DAG

In systems-theoretical terminology, since DAGs lack directed cycles, any dynamical system having a DAG as its influence graph can be considered as an open-loop system: no state variable of the system regulates in a feedback sense any other state. Various types of heuristics have been proposed to approximate a maximum-size DAG, see Festa *et al*. (1999) for a survey and Ispolatov and Maslov (2008) for a recent application in the context of biological networks. The aim of this Section is to propose a heuristic algorithm for computing a maximal DAG in which any available *a priori* information on the open-loop part can be easily taken into account. Our approach starts by choosing a spanning forest for the undirected graph, i.e. a polytree T for the directed graph G. The polytree is then incremented by adding edges to it, as long as these edges are guaranteed to preserve acyclicity. For this purpose, it is convenient to use the notion of height of a vertex. One possible way to define the height of a vertex is as the maximum length of any path from any source vertex to v, call it $h_{\max}(v)$ (this is normally called the depth in the graph-theoretical literature). Alternatively, one can use $h_{\min}(v)$, defined as the minimum length of any directed path from any source vertex to v. Similarly, the height of a DAG G is defined, respectively, as $h_{\max}(G)=\max_{v\in V}h_{\max}(v)$ or as $h_{\min}(G)=\max_{v\in V}h_{\min}(v)$. $h_{\min}$ corresponds to the maximum path length needed to reach any variable from at least one source, while $h_{\max}$ corresponds to the worst case path length from a source to all of its reachable vertices.

PROPOSITION 1. *Let $G=(V,E)$ be a DAG. If an edge $\ell_{i,j}$ such that $h_{\max}(v_i)\leq h_{\max}(v_j)$ is added to G, then the graph remains acyclic. In particular, if $h_{\max}(v_i)<h_{\max}(v_j)$ in G, then after adding the new edge the $h_{\max}$ of all vertices does not change. If instead $h_{\max}(v_i)=h_{\max}(v_j)$ in G, then after adding the new edge $h_{\max}(v_j)=h_{\max}(v_i)+1$, and $h_{\max}(v_r)=h_{\max}(v_r)+1$ for every $v_r\in\mathcal{R}(v_j)$ such that $\exists$ a path from v_j to v_r of length $h_{\max}(v_r)-h_{\max}(v_j)$.*

PROOF. A new cycle is created by the addition of the edge $\ell_{i,j}$ to a DAG G only if there is a path in G from v_j to v_i, but in this case $h_{\max}(v_i)$ must be at least $h_{\max}(v_j)+1$, which contradicts the hypothesis that $h_{\max}(v_i)\leq h_{\max}(v_j)$. Moreover, after the addition of the new edge, the $h_{\max}$ can change only for the nodes $v_r\in\mathcal{R}(v_j)$, and can only increase. This happens when a longer path from a source to v_r is created, passing through the new edge. This new path has length $h_{\max}(v_i)+1+k$, where $k\geq 0$ is the length of the longest path from v_j to v_r. Since there is already a path from v_j to v_r, then the original height of v_r should be at least $h_{\max}(v_j)+k$. So, if $h_{\max}(v_i)<h_{\max}(v_j)$ in G, then the original height is greater or equal than the new path length $h_{\max}(v_i)+1+k$, therefore the height of v_r cannot increase. If instead $h_{\max}(v_i)=h_{\max}(v_j)$ in G, when the edge $\ell_{i,j}$ is added to the DAG, then $h_{\max}$ of v_j becomes equal to $h_{\max}(v_i)+1$. Also for all vertices in $\mathcal{R}(v_j)$, the $h_{\max}$ can grow as a consequence. □

Proposition 1 allows to increment a DAG while preserving acyclicity. Iterating the argument to all edges in the complement of the polytree, we have a heuristic procedure for the construction of a maximal DAG.

ALGORITHM 1. *Construction of a maximal DAG*

Input: *polytree $T\subseteq A$*
Output: *maximal DAG $B\subseteq A$*

Procedure: $B = T$, $L = A \setminus B$
  calculate $h_{\max}$ *for the vertices of* B
  for each edge $\ell_{i,j} \in L$
  • *if* $h_{\max}(v_i) \leq h_{\max}(v_j)$ *then* $B = B \cup \{\ell_{i,j}\}$
  • *if* $h_{\max}(v_i) = h_{\max}(v_j)$ *then*
    ∘ $h_{\max}(v_j) = h_{\max}(v_i) + 1$
    ∘ $\forall v_r \in \mathcal{R}(v_j)$ *if* $\exists$ *a path from* v_j *to* v_r *of length* $h_{\max}(v_r) - h_{\max}(v_j)$ *then* $h_{\max}(v_r) = h_{\max}(v_r) + 1$

The heuristic steps are the initial choice of the polytree T and the order in which the edges are examined. In Algorithm 1, any available *a priori* knowledge on the open-loop part of the system can be included in the initial polytree T.

4 INVESTIGATING STRONG MONOTONICITY

4.1 Method I: generation of a single large SCC

When a systems like (1) is not exactly monotone, measuring how close it is to monotonicity is a computationally intense task. This measure (hereafter δ) consists in identifying the smallest number of edges whose sign switch (or removal) yields a graph with only positive undirected cycles. This problem is studied in detail in DasGupta *et al*. (2007); Hüffner *et al*. (2009); Iacono *et al*. (2010). The main idea behind the heuristics described in Iacono *et al*. (2010) for the computation of δ is to minimize the number of negative entries of $DA_U D$, where as before A_U is the symmetrized version of A and $D = \mathrm{diag}(\sigma)$. In terms of the dynamical system (1), this operation means reversing the partial order along certain axes of $\mathbb{R}^n$, in order to 'align' the cone K with the positive orthant $\mathbb{R}^n_+$ as much as possible. In Iacono *et al*. (2010), an empirical estimate for δ is found using a heuristic which repeatedly seeks for a vertex having more negative than positive incident edges and switches the sign to all its incident edges (i.e. sets to -1 the corresponding entry in the signature σ). The algorithms of Iacono *et al*. (2010) enabling the computation of the 'best' D are applicable also to directed graphs with only minor adjustments. The practical effect of a pre- and post-multiplication by $D = \mathrm{diag}(\sigma)$ is in fact to change sign to all rows and columns of the adjacency matrix in correspondence of the -1 entries in σ [with the observation that $\mathrm{sign}(\sigma_i \sigma_j A_{ij}) = \mathrm{sign}(A_{ij})$ when $\sigma_i = \sigma_j = -1$]. The minimization of the negative entries of A can be carried out also if A is not symmetric. Let $A_\sigma = DAD$ be the corresponding adjacency matrix.

Proposition 2. *Consider a signed directed graph* $G(A_\sigma)$. *Denote* A_σ^+ *and* A_σ^- *the two matrices containing, respectively, the positive and negative entries of* A_σ, $A_\sigma = A_\sigma^+ + A_\sigma^-$. *Assume* A_σ^+ *is irreducible. Then the subsystem of* (1) *having* A_σ^+ *as its influence matrix is strongly monotone.*

Proof. Since A_σ^+ has only non-negative entries, the corresponding system is cooperative hence monotone. Furthermore, since A_σ^+ is irreducible so is the corresponding system. But a cooperative irreducible system is strongly monotone, see Theorem 4.1.1 of Smith (1988). □

When A_σ^+ is not irreducible, then its SCCs should be considered. Needless to say, Proposition 2 is inefficient unless the number of negative entries of A is first minimized, as explained above. The approach is summarized in the following Algorithm.

Algorithm 2. *Strong monotonicity I*

Input: *signed adjacency matrix* A
Output: *set of strongly monotone subgraphs of* A
Procedure: *find orthant order* σ *so that the number of* $+1$ *entries of* $A_\sigma = DAD$, $D = \mathrm{diag}(\sigma)$, *is maximized*
  split $A_\sigma = A_\sigma^+ + A_\sigma^-$
  return the SCCs of $DA_\sigma^+ D$.

Since the maximization of $+1$ entries of A_σ is heuristic, the whole procedure is heuristic. As we will see in Sections 5 and 6, the peculiarity of the approach outlined in Algorithm 2 is that it often leads to a decomposition in which a single large strongly monotone subsystem is present.

4.2 Method II: construction of multiple small SCCs

In this Subsection we propose a different approach to the problem of decomposing a system into strongly monotone subsystems. This approach is more prone to building small disconnected SCCs. Starting with an acyclic subgraph $B \subseteq A$, at each step the subgraph is incremented by an edge and split into SCCs. On each SCC, strong monotonicity can be tested via Lemma 2. The edge is retained only if all SCCs are strongly monotone, then the procedure is iterated.

Algorithm 3. *Strong monotonicity II*

Input: *signed DAG* $B \subseteq A$
Output: *set of strongly monotone subgraphs of* A
Procedure: $C = B$; $L = A \setminus B$
  for each edge $\ell_{i,j} \in L$
  • *obtain the SCCs of* $C \cup \{\ell_{i,j}\}$
  • *if all SCCs are strongly monotone, then*
    ∘ $C = C \cup \{\ell_{i,j}\}$
  return the SCCs of C

Algorithm 3 is heuristic with respect to the choice of B and the order of the edges in L. Its performances tend to improve if the DAG we start with is maximal.

5 EXAMPLE: YEAST CELL CYCLE MODEL

The network shown in Figure 1a represents the influence graph of an extremely simplified model of the yeast (*Saccharomyces cerevisiae*) cell cycle, in response to an 'external' stimulation at the only source node cellsize. It was developed and studied in a boolean setting in Li *et al*. (2004). Its main characteristic is that it can reproduce faithfully the various phases of the yeast cell cycle, and the proper state transitions at the checkpoints between them. The influence graph shown in Figure 1a [with respect to the network of Li *et al*. (2004), we drop self-loops for convenience] is not a DAG and it is not monotone. Examples of frustrated cycles are the digon Clb1,2 ↔ Cdc20 or the cycles MBF → Clb5,6 → Clb1,2 → MBF and SBF → Cln1,2 → Sic1 → Clb1,2 → SBF. The last two cycles encode both the propagation of the replication order from the source cellsize and the feedback reaction of the system which concludes the S phase of the cycle, inactivating its transcription factors MBF and SBF, and consequently initiating mitosis. When we apply the

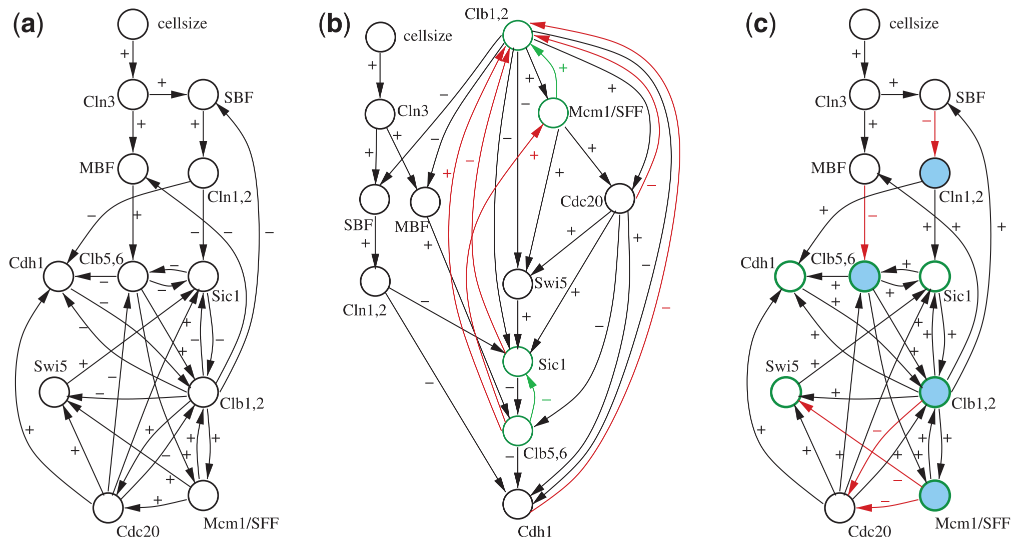

Fig. 1. Yeast cell cycle influence graph (Li *et al*., 2004). (**a**) The original signed network is shown. Self-loops are disregarded. (**b**) a DAG (edges in black) for the graph of (a). Using the height $h_{\max}$ to represent the graph, all edges of the DAG are 'descending'. Applying Algorithm 3 means adding the two green 'ascending' edges. In this way, we obtain two small strongly monotone SCCs (green nodes). Any of the red 'ascending' edges is instead forming negative directed cycles. (**c**) The graph of (a) is transformed by changing sign to all edges incident to the four nodes filled in blue. Dropping the five red edges the whole subsystem is monotone. Six nodes (circled in green) form a strongly monotone SCC.

procedure of Algorithm 1, we obtain a minimal feedback arc set of seven edges, five of which are digons. One possibility for the resulting DAG is shown in Figure 1b (DAG is in black), where the height $h_{\max}$ of the network is used to render the layout of the graph. For this DAG $h_{\min}(DAG)=2$ and $h_{\max}(DAG)=6$. Notice that the DAG has two sources, and both are needed to reach the entire DAG. In particular, for this choice of DAG the second source is Clb1,2, which is the master regulator of the entry and successive exit from the M phase of the cycle. The DAG breaks any path from the source cellsize to this critical vertex.

The adjacency matrix of the directed graph of Figure 1a has 14 negative edges out of a total of 30 (disregarding self-loops). When minimizing the number of negative edges (i.e. when computing A_σ in Algorithm 2), the sign of all incident edges is changed first for the vertex Clb1,2 (8 negative edges out of 12), then for Clb5,6 (5 negative edges out of 7, after the first round of switches), Mcm1/SFF (3 out of 5 after the second switches) and Cln1,2 (2 out of 3). After all these sign switches, we are left with the graph $G(A_\sigma)$ of Figure 1c in which there are only five negative edges left. In this case five is exactly the distance to monotonicity, and by dropping the five edges we are guaranteed that the subsystem is monotone. This monotone subsystem is not strongly connected and hence not strongly monotone. It has a SCC formed by the following six nodes: Clb1,2, Mcm1/SFF, Clb5,6, Cdh1, Swi5 and Sic1. The remaining six nodes instead form trivial (i.e. dimension 1) SCCs. Hence, although the complete network is a 'prototype' for negative feedback regulation, from Proposition 2, it hides in its structure a remarkably large strongly monotone subsystem involving half of the nodes of the network. In terms of the functioning of the cell cycle, the strategy behind this decomposition is far from obvious, except for the observation that the SCC is isolated from the source vertex cellsize, and that the influence of this last vertex is completely disconnected from the network by the cuts of the edges MBF → Clb5,6 and SBF → Cln1,2. Notice finally that deducing strong monotonicity of this SCC directly on the original graph (without the sign changes performed in Fig. 1c) is a non-trivial task even for this small-scale example.

When applying Algorithm 3 to the maximal DAG of Figure 1b, of the 7 edges dropped from the maximal DAG only two can be inserted without inducing negative directed cycles, and they both are in admissible digons (green edges in Fig. 1b). In this case, two small strongly monotone SCCs are created, both of dimension two (the two vertex pairs joined by digons) as opposed to the single SCC of dimension 6 obtained with Algorithm 2. Notice that 4 of the 5 edges that destroy strong monotonicity point to Clb1,2. As already mentioned, in this model Clb1,2 is the regulator whose activation and consecutive deactivation governs the entry and exit from the M phase, phase which constitutes the regulatory part of the cycle in response to the external stimulation, and allows the cycle to progress. In the full model, Clb1,2 rises after the S phase, due to Clb5,6 and due to the double inhibitions Cln1,2 → Cdh1 → Clb1,2 and Cln1,2 → Sic1 → Clb1,2. Hence, the three edges connecting Clb5,6, Cdh1 and Sic1 to Clb1,2 must be cut in order to have a strongly monotone subsystem.

6 LARGE-SCALE EXAMPLES

Only a limited number of large-scale biological networks are readily available as signed graphs [see e.g. DasGupta *et al*. (2007); Iacono and Altafini (2010); Ma'ayan *et al*. (2008)]. Those considered in this study are of two different types: three are transcriptional networks in which a directed edge represents the action of a transcription factor on one of its target genes, and the sign means activation (+) or inhibition (−). No stoichiometry is available for these networks. The other three networks instead represent signaling pathways. These are obtained from stoichiometric reactions, taking the signature of the

Table 1. Networks used in this study

Network	n	m	π_{in}; π_{ad}	ρ	δ	$\delta_{\max}$
Escherichia coli	1475	3320	4; 5	1452	371	1581
Yeast	690	1082	1; 0	688	41	401
Bacillus subtilis	918	1324	2; 2	912	71	415
EGFR	330	852	4; 65	138	193	376
Toll-like	679	2204	1; 413	267	468	873
Macrophage	697	1582	1; 155	359	330	704

n and m are the number of nodes and edges of the directed graph; π_{in} and π_{ad} the inadmissible/admissible digons; ρ is the number of SCCs in the original graph, δ the distance to monotonicity and $\delta_{\max}$ its theoretical upper bound.

Table 2. Maximal DAG found for the six networks

Network	γ (γ'; γ'')	ϵ	ω; ω_{tot}	$h_{\min}$; $h_{\max}$
Escherichia coli	9 (9; 376)	371	51; 65	5; 8
Yeast	1 (1; 77)	41	77; 87	4; 8
Bacillus subtilis	5 (5; 99)	71	663; 759	2; 7
EGFR	104 (94; 185)	169	38; 50	5; 37
Toll-like	452 (467; 665)	450	76; 85	8; 50
Macrophage	176 (175; 335)	316	100; 115	9; 48

The parameters shown are the size of the minimal feedback arc set (γ), the distance to monotonicity of the maximal DAG (ϵ), the minimal/total number of sources needed to cover the entire DAG (ω/ ω_{tot}) and min/max height of a graph. For γ our results are compared with those of Festa *et al*. (2001) (γ') and Ispolatov and Maslov (2008) (γ'').

Jacobian matrix, as described in Section 2.2, see also DasGupta *et al*. (2007); Kunze and Siegel (1999) for more details and a similar use. The details of the six networks are as follows:

- transcriptional networks
  - *Escherichia coli:* gene regulatory network of the *E.coli*, downloaded from *RegulonDB* database (http://regulondb.ccg.unam.mx), version 6.3.
  - *Yeast:* gene regulatory network of *S.cerevisiae* originally developed in Milo *et al*. (2002).
  - *Bacillus subtilis:* gene regulatory network for *B.subtilis*, downloaded from http://dbtbs.hgc.jp/.
- signaling networks
  - *EGFR:* network for the epidermal growth factor receptor pathway, created by Oda *et al*. (2005);
  - *Toll-like:* signaling network for the *Toll-like*-receptor. Assembled from Oda and Kitano (2006).
  - *Macrophage:* molecular interaction map of a macrophage obtained from Oda *et al*. (2004).

In the following, we shall simply refer to the networks as 'transcriptional' and 'signaling', but one should be aware that 'transcriptional, at functional level' and 'signaling, at stoichiometric level' is probably a more proper connotation for them. In Table 1, we report the data for the distance to monotonicity δ obtained in Iacono and Altafini (2010). It can already be noticed that there is a systematic difference between the two classes: the transcriptional networks are closer to monotonicity [$\delta/\delta_{\max} \sim 10-20\%$, where $\delta_{\max}$ is a theoretical upper bound on δ, see Iacono *et al*. (2010) for details] than the signaling networks ($\delta/\delta_{\max} \sim 50\%$).

When we use Algorithm 1 to construct a maximal DAG, then another key topological difference between the two classes emerges, namely that the transcriptional networks are essentially free from directed cycles, while in the signaling networks the number of edges that need to be dropped to get a DAG varies from $\sim 11\%$ to $\sim 20\%$, see Table 2. In Table 2, the performances of our Algorithm 1 are compared with those of other heuristics. In particular we choose a state-of-the-art local search method (GRASP: greedy randomized adaptive search procedure) from Festa *et al*. (2001), and a simulated annealing algorithm recently used in the context of biological networks (Ispolatov and Maslov, 2008). It can be observed that our heuristic and the algorithm of Festa *et al*. (2001) have similar performances.

If the influence graph of a system is a DAG, then the system may not be strongly monotone or not even monotone. In fact, multiple

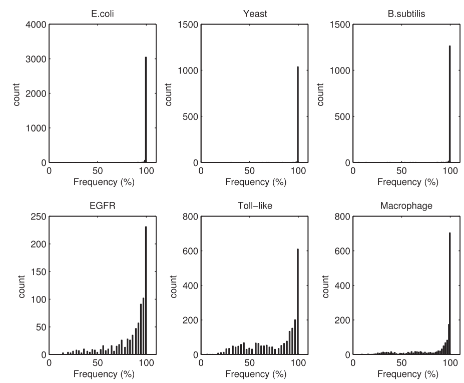

Fig. 2. Overlap between maximal DAGs in different runs of Algorithm 1. For each network, the histogram shows the distribution of the frequency of selection of an edge in a large number of nearly optimal trials. For the three transcriptional networks, there exists basically only a way to attain the maximal DAG. For the three signaling networks, instead, there is a degree of ambiguity in determining the 'open-loop' part of the dynamics, with only a fraction of the maximal DAG unanimously determined (from 1/3 for EGFR and Toll-like, to 1/2 of Macrophage).

paths originating in a fan-out node and ending in a fan-in node may have opposite signs, and hence carry opposite orders at the fan-in (activatory on one channel, inhibitory on the other), a 'frustration' (i.e. a negative undirected cycle) which is a trademark for lack of monotonicity. For all networks, the restriction to the maximal DAG still contains a large fraction of the δ 'frustrated' edges (see ϵ in Table 2), meaning that the systems have a complex and potentially incoherent open-loop dynamics. A qualitative difference between the two classes of networks can be observed looking at $h_{\max}$ on the DAGs (Table 2): the maximum length of a chain of events in the open-loop system is always much shorter in the transcriptional networks than in the signaling networks. On the contrary, the chain of events of minimum length required to reach every vertex (i.e. $h_{\min}$) is almost the same in both types of networks. Notice how the complex regulatory structure for the signaling networks implies that only a fraction of the maximal DAG is unanimously identified as open-loop subsystem over repeated runs of Algorithm 1, see Figure 2.

In Tables 3 and 4, we compare the two procedures for the construction of strongly monotone SCCs. Obviously, the difference

Table 3. Strongly monotone subsystems I: single large SCC

Network	ξ	λ	χ	ψ
Escherichia coli	10	1457	3	1
Yeast	3	688	3	1
Bacillus subtilis	7	914	3	0
EGFR	163	197	111	73
Toll-like	548	398	164	329
Macrophage	236	484	38	82

The following parameters are shown: the distance to strong monotonicity (ξ), the number of strongly monotone subsystems (λ), the size of the largest strongly monotone subsystem (χ) and the number of edges dropped that belong to a strongly monotone SCC (ψ).

Table 4. Strongly monotone subsystems II: multiple independent SCCs

Network	ξ (ξ')	λ	χ	ψ
Escherichia coli	7	1459	2	0
Yeast	1 (1)	690	1	0
Bacillus subtilis	2	914	3	0
EGFR	64 (45)	283	5	2
Toll-like	377	633	6	90
Macrophage	84 (75)	575	10	0

The same parameters of Table 3 are shown. For ξ also a comparison with the values reported in Aswani *et al*. (2009) is shown (ξ').

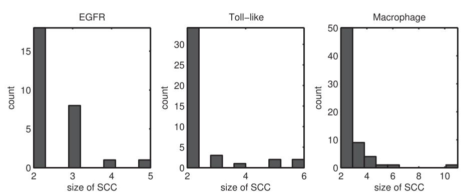

Fig. 3. Size of the non-trivial strongly monotone SCCs created by Algorithm 3 for the three signaling networks.

can be appreciated only on the three signaling networks, which have a sufficient amount of feedback regulations. As anticipated, the size of the largest strongly monotone SCC detected (i.e. χ) is consistently much higher for the method of Section 4.1 than for the one of Section 4.2. Apart from the large SCC, Algorithm 2 returns only trivial subsystems. For Algorithm 3, instead, the distribution of size of the non-trivial strongly monotone SCCs is shown in Figure 3. Notice that our numbers for this last case are still higher than those reported in Aswani *et al*. (2009) (and shown in Table 4), meaning that there is probably still room for improvement in our Algorithm 3.

7 CONCLUSION

The investigation of the dynamical properties of large-scale biological networks poses a problem and a challenge for the field of Systems Biology because of its complexity and lack of suitable methodology. By using simple tools from graph theory, we have shown in this article that nearly-optimal solutions for a couple of important dynamical problems, such as the identification of a minimum set of feedback loops whose removal leave the system without regulation, and the decomposition of the network into dynamically ‘simple’ subsystems, may be found with heuristics which are computationally efficient also for networks of the several hundreds / few thousands of molecular species. While not optimal and restricted to a specific class of network representations (influence graphs), our approach is promising and the insight it provides on the structure of the networks already significant.

Funding: This research was sponsored by a PRIN grant from MIUR, Italy.

Conflict of Interest: none declared.

REFERENCES

Aswani,A. *et al.* (2009) Graph-theoretic topological control of biological genetic networks. In *Proceedings of the American Control Conference* . IEEE Press, Piscataway, NJ, USA, pp. 1700–1705.

DasGupta,B. *et al.* (2007) Algorithmic and complexity results for decompositions of biological networks into monotone subsystems. *Biosystems*, **90**, 161–178.

Deangelis,D.L. *et al.* (1986) *Positive Feedback in Natural Systems*. Springer, New York, NY, USA.

Deo,N. (1974) *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Fages,F. and Soliman,S. (2008) From reaction models to influence graphs and back: A theorem. In Fisher,J. (ed.) *Formal Methods in Systems Biology*. vol. 5054 of *Lecture Notes in Computer Science*, Springer GmbH, Heidelberg, DEU, pp. 90–102.

Festa,P. *et al.* (1999) Feedback set problems. In Du,D.-Z. and Pardalos,P.M. (eds) *Handbook of Combinatorial Optimization*, Suppl. Vol. A, Kluwer Academic Publishers, Dordrecht, NLD, pp. 209–258.

Festa,P. *et al.* (2001) Algorithm 815: FORTRAN subroutines for computing approximate solutions of feedback set problems using GRASP. *ACM Trans. Math. Softwr*, **27**, 456–464.

Hirsch,M.W. (1983) Differential equations and convergence almost everywhere in strongly monotone semiflows. In Smoller,J. (ed.) *Nonlinear Partial Differential Equations (Durham, NH, 1982)*. American Mathematical Society, Providence, RI, USA, pp. 267–285.

Huber,W. *et al.* (2007) Graphs in molecular biology. *BMC Bioinformatics*, **8**, S8.

Hüffner,F. *et al.* (2009) Separator-based data reduction for signed graph balancing. *J. Comb. Optim.*, **20**, 335–360.

Iacono,G. and Altafini,C. (2010) Monotonicity, frustration, and ordered response: an analysis of the energy landscape of perturbed large-scale biological networks. *BMC Syst. Biol.*, **4**, 83.

Iacono,G. *et al.* (2010) Determining the distance to monotonicity of a biological network: a graph-theoretical approach. *IET Syst. Biol.*, **4**, 223–235.

Ispolatov,I. and Maslov,S. (2008) Detection of the dominant direction of information flow and feedback links in densely interconnected regulatory networks. *BMC Bioinformatics*, **9**, 424.

Karp,R.M. (1972) Reducibility among combinatorial problems. In Miller,R.E. and Thatcher,J.W. (eds) *Complexity of Computer Computations*. Plenum Press, New York, NY, USA, pp. 85–103.

Klamt,S. *et al.* (2006) A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, **7**, 56.

Kunze,H. and Siegel,D. (1994) A graph theoretical approach to monotonicity with respect to initial conditions. In Liu,X. and Siegel,D. (eds) *Comparison Methods and Stability Theory*. vol. 162 of *Lecture notes in Pure and Applied Mathematics.* Marcel Dekker, New York, NY, USA, pp. 207–216.

Kunze,H. and Siegel,D. (1999) A graph theoretical approach to monotonicity with respect to initial conditions II. *Nonlinear Anal.*, **35**, 1–20.

Li,F. *et al.* (2004) The yeast cell-cycle network is robustly designed. *Proc. Natl Acad. Sci. USA*, **101**, 4781–4786.

Ma’ayan,A. *et al.* (2008) Proximity of intracellular regulatory networks to monotone systems. *IET Syst. Biol.*, **2**, 103–112.

Milo,R. *et al.* (2002) Network motifs: simple building blocks of complex networks. *Science*, **298**, 824–827.

Oda,K. *et al.* (2004) Molecular interaction map of a macrophage. *AfCS Res. Reports*, **2**, 14 DA.

Oda,K. *et al.* (2005) A comprehensive pathway map of epidermal growth factor receptor signaling. *Mol. Syst. Biol.*, **1**, 2005.0010.

Oda,K. and Kitano,H. (2006) A comprehensive map of the toll-like receptor signaling network. *Mol. Syst. Biol.*, **2**, 2006.0015.

Papin,J.A. *et al.* (2005) Reconstruction of cellular signalling networks and analysis of their properties. *Nat. Rev. Mol. Cell Biol.*, **6**, 99–111.

Quirk,J. and Ruppert,R. (1965) Qualitative economics and the stability of equilibrium. *Rev. Econ. Stud.*, **32**, 311–326.

Shen-Orr,S.S. *et al.* (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.*, **31**, 64–68.

Smith,H.L. (1988) Systems of ordinary differential equations which generate an order preserving flow. A survey of results. *SIAM Rev.*, **30**, 87–113.

Smith,H.L. (1995) *Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems*. vol. 41 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, USA.

Sontag,E.D. (2007) Monotone and near-monotone biochemical networks. *Syst. Synth. Biol.*, **1**, 59–87.

Soulé,C. (2003) Graphic requirements for multistationarity. *ComPlexUs*, **1**, 123–133.

Tarjan,R.E. (1972) Depth-first search and linear graph algorithms. *SIAM J. Comput.*, **1**, 146–160.

Thieffry,D. (2007) Dynamical roles of biological regulatory circuits. *Brief. Bioinform.*, **8**, 220–225.

Zaslavsky,T. (1982) Signed graphs. *Discrete Appl. Math.*, **4**, 47–74.