Molecular BioSystems

This article was published as part of the Computational and Systems Biology themed issue

Please take a look at the full table of contents to access the other papers in this issue.
On the basic computational structure of gene regulatory networks

Carlos Rodríguez-Caso,*a Bernat Corominas-Murtraa and Ricard V. Soleab

Received 11th March 2009, Accepted 5th June 2009
First published as an Advance Article on the web 21st July 2009
DOI: 10.1039/b904960f

Gene regulatory networks constitute the first layer of the cellular computation for cell adaptation and surveillance. In these webs, a set of causal relations is built up from thousands of interactions between transcription factors and their target genes. The large size of these webs and their entangled nature make it difficult to achieve a global view of their internal organisation. Here, this problem has been addressed through a comparative study of Escherichia coli, Bacillus subtilis and Saccharomyces cerevisiae gene regulatory networks. We extract the minimal core of causal relations, uncovering the hierarchical and modular organisation from a novel dynamical/causal perspective. Our results reveal a marked top-down hierarchy containing several small dynamical modules for E. coli and B. subtilis. Conversely, the yeast network displays a single but large dynamical module in the middle of a bow-tie structure. We found that these dynamical modules capture the relevant wiring among both common and organism-specific biological functions such as transcription initiation, metabolic control, signal transduction, response to stress, sporulation and cell cycle. Functional and topological results suggest that two fundamentally different forms of logic organisation may have evolved in bacteria and yeast.

1. Introduction

The pattern of regulatory interactions linking transcription factors (TFs) to their target genes constitutes the first level of a multilayered network of gene regulation, the so-called gene transcriptional regulatory networks (GRNs). Some of these patterns have been recovered from genome-wide approaches, particularly well established for Escherichia coli and Bacillus subtilis as well as the unicellular eukaryote Saccharomyces cerevisiae. In such a picture, both hierarchical and modular components have been repeatedly highlighted, although no general agreement exists on which scale of analysis more accurately captures global complexity.

The conceptualisation of cellular interaction maps within the framework of graph theory has provided powerful insights into their hierarchical, modular organisation as well as shedding light on the topological constraints imposed on its dynamical behaviour. However, their quantification, even their identification, has led to a nonuniform concept of a module under functional, topological, evolutionary and developmental criteria. Similarly, the observed hierarchy seldom matches an ideal feed-forward relation between components.

An alternative approach considers looking at GRNs in a more fundamental way, namely as sets of causal relations, featured by the directed nature of the link between a transcription factor and its target gene. Causal links, namely what acts on what, make it possible to actually define the skeleton underlying dynamics by attending to the cyclic and linear nature of the genetic circuits.

![Fig. 1](image_url) A small toy model (a) and its dynamic behaviour. The observed pattern is generated by the activity of two basic feedback loops, indicated as A and B in (a). These subsets are responsible for the qualitative dynamics exhibited by the net, as shown in (b) with filled circles. Moreover, this attractor results from the combination of the two different periodic orbits displayed by the two basic modules, whose time series are shown in (c–d). For a mathematical treatment of this specific example see appendix.
relational picture of a biological computation. This is a computational perspective and, not surprisingly, GRNs have been compared to computers. Cellular computations pervade the diverse responses to external stimuli as well as cell robustness and plasticity. A number of dynamical approximations suggest a link between network organisation and its dynamical behaviour. However, due to the large size of these systems, only a few small ones have been fully analysed.

Ideally, it would be desirable to have a method to construct a graph capturing all non-trivial causal relations and thus all potentially important computational links. In this paper we show that such organisation of GRNs is captured by the principle of causality depicted by the directed nature of gene–gene relations. This can be uncovered by exploiting the properties of directed graphs, in particular by the use of leaf removal algorithms (LRAs) and the identification of the so-called strongly connected components (SCCs). As we will see, LRAs recover the network fraction responsible for potentially non-trivial computations. Furthermore, the internal organisation of this special subgraph can be topologically dissected by SCC identification.

An SCC is defined for a set of vertices if there is a walk—attending to the directedness of edges—from every vertex to every other vertex of this set (see Fig. 2). The identification of SCC has been applied to a diverse number of systems from the Internet to metabolism and neural wiring networks. Recently, it has been reported that yeast GRN presents a giant SCC contrasting with an acyclic and feed-forward organisation of E. coli.

In this work, we explore the topological constraints derived from wiring patterns of causal dependencies in a comparative analysis of yeast, E. coli and B. subtilis by defining their qualitatively relevant causal cores. We explore their internal organisation in terms of irreducible computational entities at the level of gene regulatory systems. As will be shown below, our analysis (using an updated version of E. coli GRN) revealed relevant differences in relation to previous results published in the literature.

2. Results

2.1 Causality, dynamics and topology

Causal relations in GRNs can be described in terms of directed graphs (in short digraphs). A graph constitutes a set of vertices or nodes (here, the genes) and the set of edges linking them (the relations among genes). The regulatory effect of a TF gene \( v_i \) on a specific target gene \( v_j \) is depicted by an arrow \( (v_i \rightarrow v_j) \) in the graph plot.

If the vertices are genes, a chain of connections of different TFs corresponds in graph theory with a directed walk and it can be interpreted as a causal chain. Interestingly, all TFs exhibit outgoing links, whereas non-TF genes (the target ones) only receive arrows from the TF set. The number of outgoing links of a vertex is known as out-degree (denoted by \( k_{out} \)) whereas the number of incoming edges is the in-degree \( (k_{in}) \). Since a TF can be a regulatory target of other TFs, they can exhibit both incoming and outgoing links, allowing feed-backs to occur.

Especially important information about the organisation of directed graphs is provided by the analysis of its connected components, i.e. the graph in which every pair of distinct vertices has a walk—see appendix. In digraphs, directed walks are non-trivially organised. The Internet is an excellent illustrative example of this, exhibiting a bow-tie structure with three different connected components: (1) a central strongly connected component (SCC), namely a subgraph for which every two vertices are mutually reachable; (2) a set of incoming components (IC) composed of the set of feed-forward pathways starting from vertices without in-degree and ending in SCC and (3) a set of outgoing components (OC) where pathways starting from SCC end in vertices without out-degree (see Fig. 2).

Typically, the pattern of activation of a given gene has a time-dependent causal relation with the state of the set of genes affecting it, and every vertex \( v_i \) can acquire a given state from a number of possible states. Specifically, the state of \( v_i \) at time \( (t+1) \) is influenced by the state of the vertices affecting it at time \( t \)—see appendix. The equations describing the dynamical behaviour of a given vertex \( v_i \) at time \( t+1 \) can be formulated in different ways, including Boolean dynamics, threshold nets as Fig. 1 illustrates or coupled differential equations. These models are different but all show a common principle of causality: the state of a given vertex \( v_i \) at time \( t+1 \) is exclusively defined by the vertices affecting it at time \( t \). No matter what our choice of dynamical equations, the patterning of links must strongly influence a system’s behaviour.

The presence of a cyclic walk—a directed walk where the first and the ending vertices are the same—is a necessary (but not sufficient) requisite for a periodic solution. This is due to the fact that, in interpreting directed edges as causal relations, a cyclic graph implies that every vertex is indirectly affected by itself. When we consider the whole graph, an overlapping of cycles originates the SCC set that can be governed by the IC set. As we shall show below, these two structures represent the core of the graph that qualitatively constrains its dynamical complexity. By contrast, in linear directed walks, the most upstream element fully determines the final state of the elements belonging to this walk.

A set able to properly capture the relevant components qualitatively affecting global behaviour should remove linear

---

Fig. 2 Bow-tie structure depicting an incoming component (IC), two strongly connected components (SCCs) and the outgoing component (OC).

§ For the sake of simplicity, we present here the concepts related to graph theory in an intuitive way. For formal definitions and construction, see appendix.
paths from the graph. Under this view, we can identify this set by means of a straightforward iterative algorithm. Given the underlying influence of causal relations on a system’s dynamics, we will use the label dynamic for most of our definitions.

2.2 Dynamical backbone

We compute the dynamical backbone (DB) of a given directed graph $\mathcal{G}$ (denoted by $DB(\mathcal{G})$) by the iterative pruning of vertices with $k_{\text{out}} = 0$ from the initial graph, but maintaining those vertices whose $k_{\text{in}} = 0$ (Fig. 3). This algorithm belongs to the family of so-called leaf removal algorithms$^{9,26}$ that have been previously used for the analysis of GRNs$^{6,9,26}$.

Fig. 3a illustrates the mechanism of pruning. Notice that, in contrast with previous proposals of leaf removal algorithms, the $DB(\mathcal{G})$ also keeps the single root vertices i.e. those such that $k_{\text{in}} = 0$ which appear isolated. Single root vertices are special because their state is only externally changed and is not influenced by other genes. We observe that, in general, the $DB(\mathcal{G})$ displays more than a single connected component (see Fig. 3). Interestingly, this dynamically relevant subgraph coincides with the union of the previously described IC and SCC sets of a directed graph. By contrast, OC is associated with another interesting subgraph, to be indicated as $DB'(\mathcal{G})$, formed by the fraction of the net that exclusively displays feed-forward structures. See appendix for its formal construction.

The application of the above defined algorithm leads to a drastic reduction of network complexity, preserving the computational structures responsible for the qualitative behaviour of the net.

2.3 Dynamical modules and hierarchy

As we argued above, the core of causal relations of a GRN is captured by its $DB$. Furthermore, the causal relations inside this $DB$ can display some kind of hierarchy. As we shall see, $DB$ organisation can be studied by SCC detention and the so-called graph condensation process. The mathematical appendix formally details the treatment of GRNs that uncovers the order relation (i.e. the hierarchical order) among the elements of the net.

Fig. 3b illustrates the process for module identification by SCC detection. Interestingly, the nodes belonging to an SCC are mutually affected and therefore, they all contribute to the state definition of all the set (see A and B regions of the toy model described in Fig. 1a). The complexity of these topological entities cannot be reduced and thus, the SCC concept defines an irreducible unit of causal relations (see mathematical appendix for formal definition). In this sense, the SCC can be identified with the module concept, since it constitutes an identifiable semiautonomous part of the system,$^{27}$ capturing a part of the dynamical complexity of the system. Therefore, we say that, within the context of GRNs, SCCs represent a sort of dynamical modules. We finally note that, consistently with definition, a single vertex belonging to the $DB$ but not participating in any cycle is itself a dynamical module.

Interestingly, when the SCC is represented by a single vertex, the resulting graph is a directed acyclic graph (see Fig. 3c). Such an operation is commonly referred to in the literature as the condensation of a graph.$^{43}$

The result after the condensation process is precisely the feed-forward organisation which enables us to define an order relation among the elements of the $DB$. It is worth stressing that an order relation cannot be established in cyclic wiring. The condensation process allows us to identify such cycles in SCCs representing dynamical modules. In this way, the internal organisation in a hierarchy of causal relations is revealed for the $DB$.

2.4 E. coli dynamical backbone

We extracted the GRN for *E. coli* K-12 prokaryote from the available information in the RegulonDB 6.0 database.$^{24}$ The resulting network was a directed graph with 1607 vertices (43 of them with $k_{\text{in}} = 0$) and 4141 links with a giant component of 1589 vertices and average degree $\langle k \rangle = 5.1$ [see Methods section for network construction]. The network included a total of 156 vertices with $k_{\text{out}} > 0$, corresponding with transcription and $\sigma$ trans-acting factors, and 1451 vertices with $k_{\text{out}} = 0$, i.e., the target genes.

From the *E. coli* GRN, the $DB$ subgraph was obtained after the second iteration of the algorithm. The $DB$ subgraph
consists of 142 vertices (8.9% of the GRN) distributed as follows: 33 single root vertices and a set of subgraphs with 109 vertices (10 of them with $k_{in} = 0$) and 279 edges. This set is organised in nine graphs: a giant component of 100 vertices ($\langle k \rangle = 3.7$), another component displaying two elements and seven isolated self-interacting vertices. We found that the size of the studied DB is markedly smaller than that expected in a null model obtained by a randomisation process. In this methodology, node degree is conserved but correlations among nodes are destroyed (see Methods section). Differences in the DB size between *E. coli* and a randomised counterpart feature the biological fact that only a small fraction of genes in the genome operate over a majority of genes without a direct role in the regulation of transcription. As expected and in agreement with the direct observation of RegulonDB, the genes belonging to DB are described as either transcription or effectors. Only two exceptions were found: the transcription anti-terminator cspE and trmA, a tRNA methyltransferase (according to RegulonDB 6.0 [see SI1 for the biological function of DB genes]). Functional analysis based on gene ontology annotation confirmed the significant ($P < 10^{-5}$) overabundance of general functions related to regulation of transcription, signal transduction and transcription initiation (see Table 1 and SI2 for a detailed statistical analysis).

Concerning the previous observations, we obtained a graph resulting from the interactions among TFs (topologically, the nodes with out-degree) in GRN. According to its definition, DB is also a subset of a TF subnet. Fig. 4a inset shows that DB size for a randomised TF–TF subgraph is somewhat smaller than the *E. coli* DB. We also found that the number of root nodes expected by chance in both GRN and TF subgraph follow a behaviour similar to that observed for DB size. These results show that the DB structure for *E. coli* exhibits an organisation markedly different to that expected in a null model (see Fig. 4b).

Fig. 5a shows the DB organisation after the condensation process. The analysis of the condensed DB revealed five dynamical modules larger than one vertex (Fig. 5b). Interestingly, the sizes of SCCs are markedly smaller in both GRN and TF randomised sets. Notice that Fig. 4 depicts large values for SCC sizes equal to one in randomised networks, however, they correspond with the fraction of the non-condensed network. Typically, randomised networks exhibited a single large SCC component (within the range of 800–900 nodes), occasionally together with a very small SCC, rarely larger than three nodes (data from SCC distributions within networks are not shown). It is worth noting that no randomised web, neither for GRN nor TF–TF subgraph sets, produced exclusively small SCCs as observed in real data. This indicates that GRN is not devoid of cycles but they are distributed along the DB and confined in dynamical modules of small size.

We found that the condensed DB captures the hierarchical behaviour of the largest graph component evidenced by a feed-forward order relation with six layers of downstream dependencies (see Fig. 5). By definition, the first layer contains all the vertices with $k_{in} = 0$ but we can see that it also includes the largest dynamical module.

The largest module (A in Fig. 5) contains four of the seven effectors described for *E. coli* (see Table 1 and SI2 for statistical details). These elements are responsible for transcription initiation. Together with the primary initiator factor rpoD ($\sigma^{II}$), we found the alternative ones operating under heat shock stress (rpoH and rpoE, corresponding with $\sigma^{12}$ and $\sigma^{24}$, respectively). In addition, rpoN ($\sigma^{24}$, initiator of nitrogen metabolism genes) is also part of the module. The second largest hub (a highly connected vertex) in the DB is the cyp gene, also known as CAP (catabolite activator protein). CAP is a general regulator that exerts a positive control of many of the catabolite sensitive operons as a sensor of glucose starvation. Interestingly, the members of module A participate in 44% of the total number of links in the GRN, indicating the relevance of this module in defining the state of the whole network.

Other relevant factors are co-localised in this dynamical module such as those TFs related to nutrient sensor and assimilation (phosphate sensor system, phoB, as well as nucleoside (cvtR) and arginine (argP) transport control). Notice that the initiator factor of DNA replication initiator (dhaA) is associated with this group, and we also found two specific TF expressed under stress conditions (lexA and cpxR). Similarly, the other four modules include key genes associated with adaptive responses to changing environmental cues.

Table 1 A selection of statistically over-represented biological function terms from gene ontology annotation (GOA) is shown for DB, the root node set and dynamical modules (A to E), according to functional analysis for *E. coli*. Notice that three genes of the DB were not found in GOA annotation (see SI2 for a detailed analysis). The number of genes belonging to a specific GOA term is depicted in the first column. The fractions of genes associated with specific GOA with respect to the total number of genes for each particular set are shown for $P \leq 0.05$ ($P$ is the level of significance). Highly significant results ($P < 10^{-5}$) are indicated with (**). Values within the interval of $10^{-5} < P < 10^{-3}$ are marked (*)

| Biological function | # Genes in GO | Fraction of the GO term in the analysed set |
|---------------------|--------------|------------------------------------------|
| Regulation of transcription, DNA-dependent | 354 | 135/136* 37/37* |
| Regulation of metabolism | 399 | 136/136 37/37** |
| Two-component signal transduction system (phosphorelay) | 91 | 26/136* 6/37** |
| Transcription initiation | 7 | 7/136* |
| Negative regulation of cellular process | 36 | 10/136* 4/37** |
| DNA replication | 62 | 3/11** |
| Response to heat | 4 | 1/1 |
| Response to antibiotic | 60 | |
| Carbohydrate transport | 112 | |

This journal is © The Royal Society of Chemistry 2009
These include homeostasis in acid environment (stress responses to high osmolarity, module B), antibiotic resistance (led by $\text{mar}A$ and $\text{mar}R$, module C), glucitol use (module D) or responses to oxygen changes (module E).

In summary, the $E. \ coli$ DB describes a hierarchical feed-forward network with a strong fan-like pattern dominated by a single irreducible subset of TFs. This pattern reveals, in terms of a computational design, a strongly centralised organisation, with a well defined node affecting multiple layers of activity. The largest hub has no input from other members of the DB and thus only affects each other’s behaviour. Within the DB, we identify the dynamical modules responsible for the control of transcriptional replication under both normal and stress conditions, control of metabolism, DNA replication as well as assimilation of essential sources of nitrogen and phosphorus.

2.5 $B. \ subtilis$ dynamical backbone

$B. \ subtilis$ GNR was obtained from DBTBS$^{55}$ [see Methods section for network construction]. The resultant network consists of 922 nodes (159 with $k_{\text{out}} > 0$, 83 of them with $k_{\text{in}} = 0$ and 763 with $k_{\text{out}} = 0$) and 1366 directed links ($\langle k \rangle = 2.96$). A component of 892 nodes was predominant.
The fractions of genes associated with specific GOA with respect to the total number of genes for each particular set are shown for \( P \) DB depicted in the first column. Notice that eight genes of 144 belonging to \( P \) DB participate in 26% of the total number of links of the GRN.

and the remaining nodes formed part of very small isolated subgraphs (one of five nodes, one of three, 10 of two nodes and two isolated self-interacting nodes).

The computation of \( DB \) required four iterations before the acquisition of the stable graph. The process removed 778 nodes, resulting in a \( DB \) of 144 genes (10.5% of the GRN). In this set, 73 were isolated root nodes and 71 nodes were distributed in eight connected components (a big one of 59 nodes, one of five, one of two and five self-interacting isolated nodes) with 108 links. Analogously to the results observed for \( E. coli \), statistical analysis of GOA terms showed an overabundance of regulation of transcription metabolism and cell signalling (see Table 2 and SI2 for statistical details) in both \( DB \) and root nodes set (see Table 2 and SI2 for statistical details). Other functions relating to transcription termination and positive regulation process appear in \( B. subtilis DB \).

We note that comparison of functional analysis among species must be taken with caution due to the different coverage of genes in GOA terms that may produce a bias in the interpretation of results. In order to provide a more complete picture, we consider functional information annotated in DBTBS. However, we cannot exclude some discrepancies derived from the use of different sources. Similarly to \( E. coli \), statistical functional analysis is complemented by a direct function examination of the genes belonging to \( DB \) in DBTBS (see SI for functional information).

Concerning null model comparison, Fig. 6a and b suggest a similar behaviour for \( DB \) and the set of root nodes in randomised networks to that observed for \( E. coli \).

SCC calculation revealed five dynamical modules as illustrated in Fig. 7. Four of them are in the same 54-connected component and the other one is in the 5-connected component. Similarly to \( E. coli \), a single markedly large SCC is expected in \( E. coli \) bacterial counterpart. In this case, module A consisted of a cross-regulation of \( sigA \) and \( sigD \) sigma factors. These two nodes participate in 26% of the total number of links of the GRN.

This agrees with the essential role of these genes. \( SigA \), also known as \( \sigma^{32} \) or \( rpoD \), is an essential gene, primary \( \sigma \) factor of this bacterium. It is also worth noting that the equivalent \( rpoD \) gene occupies a similar position on the top of the \( E. coli \) condensed \( DB \) (module A in Fig. 5). The module partner of this gene in \( B. subtilis \), \( sigD \) or \( \sigma^{38} \), is a sigma factor required for flagellum and motility genes involved in the chemotaxis process. This is different from \( E. coli \), since its equivalent gene \( flhA \) (also \( \sigma^{28} \)) does not belong to the \( E. coli \) module A but it has a direct downstream dependence on this module.

As illustrated in Fig. 7, modules C and D receive a direct input from the master module A. They appear related to signal transduction systems (\( resD \) and \( comK \) in modules C and D, respectively). Interestingly, a cross dependence between respiration and phosphate assimilation is captured in module C. It has been reported that \( resD \) can also play a regulatory role in respiration whereas \( phoD \) forms part of a molecular system involved in the regulation of alkaline phosphatase and phosphodiesterase alkaline phosphate, two enzymes involved in the uptake of free phosphate groups from the environment. Moreover, a catabolite repression factor, \( ccpA \), responsible for the repression of carbohydrate utilisation and the activation of excretion of excess carbon affects this module. This suggests a strong relation between phosphorus and carbon assimilation and respiratory regulation. On the other hand, the \( rok-comk \) genetic circuit described in module E appears to be involved in the expression of the late competence genes.

In contrast to what occurs in \( E. coli \), the most complex module is not at the top of the hierarchy. Module E is a three-component gene (much smaller than for \( E. coli \)) closely related to the sporulation process. It is noteworthy that this module receives inputs from the competence-related module E and another sporulation module (see Table 2) is found in an independent component. This is the case for module B which does not belong to the same \( DB \) connected component as the remaining modules. It contains a sporulation-specific \( \sigma \) factor (\( sigG \), together with \( spoT \) controlling factor). Interestingly, this module, significantly associated with sporulation function (see Table 2), is affected by another sporulating \( \sigma \) factor (\( sigF \)). Another relevant trait is the link between sporulation and

---

**Table 2** A selection of statistically over-represented biological function terms from gene ontology annotation (GOA) is shown for \( DB \), the root node set and dynamical modules (A to E), according to functional analysis for \( B. subtilis \). The number of genes belonging to a specific GOA term is depicted in the first column. Notice that eight genes of 144 belonging to \( DB \) were not found in GOA and therefore not considered in this analysis. The fractions of genes associated with specific GOA with respect to the total number of genes for each particular set are shown for \( P \) ≤ 0.05 \( (P \) is the level of significance). Highly significant results \((P < 10^{-5})\) are indicated with (**). Values within the interval of \( 10^{-5} > P > 10^{-3} \) are marked (*)

| Biological function | # Genes in GO | Fraction of the GO term in the analysed set |
|---------------------|--------------|------------------------------------------|
| Regulation of transcription, DNA-dependent | 357 | 113/121* | 58/62* | 2/2 | 2/2 | 2/2 | 2/2 | 3/3 |
| Regulation of metabolism | 368 | 114/121* | 58/62* | 2/2 | 2/2 | 2/2 | 2/2 | 3/3 |
| Two-component signal transduction system (phosphorelay) | 99 | 8/121* | 12/62* | — | 2/2 | — | — | — |
| Transcription initiation | 26 | 14* | 5/62** | 2/2 | 2/2 | — | — | 1/3 |
| Negative regulation of cellular process | 15 | 8/121* | 3/62 | — | — | — | — | — |
| Positive regulation of cellular process | 8 | 6/121* | 6/62* | — | — | — | — | — |
| Transcription termination | 6 | 3/121 | 2/62** | — | — | 2/2 | — | — | 3/3 |
| Sporulation | 258 | — | — | 2/2 | — | — | — | — | — |
| Phosphate transport | 10 | — | — | 1/2 | — | — | — | — | — |
response to nutritional stress performed by spo0A and the catabolite repression role of abrB in module D.

From a topological point of view, the most obvious difference between E. coli and B. subtilis is the simpler composition of dynamical modules for the bacillus. We must stress that the knowledge of these organisms—specially of B. subtilis, which is much less known than its E. coli counterpart—is constantly advancing. Thus, a variation in the wiring of these modules cannot be neglected. However, differences in module conformation might also be due to the large phylogenetic distance among these bacteria, as well as the extremely different environments in which they live. E. coli is a gram-negative enterobacterium unable to sporulate that lives in the intestine of warm animals, whereas B. subtilis is a free living gram-positive organism, usually found in soils and with the ability to sporulate. However, in spite of these differences, a very similar architecture with a strong hierarchical character of its DB is shared. Once again, a single only-output hub is observed. As shown below, this is markedly different from what is found in our eukaryotic example.

2.6 Yeast dynamical backbone

GRN of Saccharomyces cerevisiae eukaryote was obtained from a compilation of different genome scale transcriptional analyses8 [see Methods for network construction]. The resulting network was a directed graph consisting of a single connected component with 4441 vertices (29 of them with $k_{in} = 0$) and 12,900 links, leading to a $\langle k \rangle = 5.8$. The network included a total of 157 vertices with $k_{out} > 0$ and 4284 vertices with $k_{out} = 0$, corresponding with the transcription factors and target genes analysed in the different datasets compiled in ref. 8.
The yeast DB subgraph was obtained after three iterations of DB pruning. The DB was formed by a set of 109 vertices (2.45% of the GRN), 17 of them being isolated single root vertices, and a single connected component displaying 92 vertices and 318 edges, leading to $\langle k \rangle = 6.1$. Analogously to bacteria GRNs, the randomisation process produced networks with much larger DB with a smaller number of root nodes and a very large SCC compared to real yeast GRN (see Fig. 8). However, a different behaviour was observed from the yeast TF–TF subgraph producing a fairly bigger DB after the randomisation process. This suggests that although bacteria and yeast DB exhibited a similar size, they would differ in their DB organisation.

Functional analysis revealed that DB consists of elements responsible for regulation of transcription and metabolism as observed in bacteria. In addition, cell cycle related functions and response to chemical stimulus appeared significantly over-represented in DB, the root node set and SCC (see Table 3 and SI2 for detailed statistical analysis).

All the 109 genes of the DB appear included in the list obtained from ref. 8 defining the TF genes of the yeast GRN. In addition, we checked that all DB elements but five were clearly identified as TFs in the current version of the Saccharomyces genome database. However, they were all classified as regulators of gene expression [see SI for biological details of DB genes].

The condensation process of yeast DB revealed a single dynamical module of 60 vertices (module A in Fig. 9a) with a high average degree ($\langle k \rangle = 6.6$) that contrasts to SCC distribution in bacteria. The module actually included more than half of the DB, tied to a much less marked hierarchy (see Fig. 9b) than that observed in the E. coli DB. However, the number of layers (seven) was very close in both organisms, as Fig. 9a indicates.

We can see that the large dynamical module occupies a central position within the DB. Overall, the yeast DB resembles the so-called bow-tie organisation observed in the Internet and metabolism where incoming inputs are integrated in a large component leading to a set of outgoing outputs. This differs substantially from the hierarchical character of E. coli DB and from the point of view of computation it reveals a much higher integration and pre-processing of the input–output paths.

Table 3 A selection of statistically over-represented biological function terms from gene ontology annotation (GOA) is shown for DB, the root node set and dynamical modules (A to E), according to functional analysis for S. cerevisiae. Notice that only two genes of the DB were not found in GOA annotation (see SI2 for a detailed analysis). The number of genes belonging to a specific GOA term is depicted in the first column. The fractions of genes associated with specific GOA with respect to the total number of genes for each particular set are shown for $P < 10^{-5}$ (P is the level of significance). Highly significant results ($P < 10^{-5}$) are indicated with (**). Values within the interval of $10^{-5} < P < 10^{-3}$ are marked (*)

| Biological function                                      | # Genes in GO | Fraction of the GO term |
|----------------------------------------------------------|---------------|-------------------------|
| S. cerevisiae DB                                         | Root nodes    | DM A                    |
| Regulation of metabolism                                 | 466           | 80/107*                 |
| Regulation of transcription, DNA-dependent               | 327           | 69/107*                 |
| Positive regulation of cellular process                  | 101           | 33/107*                 |
| Negative regulation of cellular process                  | 191           | 17/107*                 |
| Response to chemical stimulus                            | 270           | 20/107*                 |
| Cell cycle                                               | 412           | 21/107**                |
| Transcription initiation                                 | 56            | 5/107                   |
| G1/S-specific transcription in mitotic cell cycle        | 12            | 9/107*                  |
|                                                          |               | 51/60*                  |
|                                                          |               | 44/60*                  |
|                                                          |               | 21/60*                  |
|                                                          |               | 13/60*                  |
|                                                          |               | 11/60**                 |
|                                                          |               | 17/60*                  |
database information indicated TFs for a cell cycle such as swi4, swi6 and swi5 (they control G1 related genes and they are also involved with DNA repair), hcm1 (S phase related genes), yhp1 and yox1 (M/G1 phase) and fkh1-2 (G2 phase), among others. Moreover, the swi TFs located in module A form part of the five TFs contained in the minimal dynamical network suggested for the yeast cell cycle. Although this network combines protein modification and transcription, we observe that all TFs in Li’s network are included in the yeast DB. The two remaining TFs in Li’s module that are not inside module A (mcm1, mbp1) are located upstream in the hierarchy, indicating a master control over the swi factors (and the remainder module A partners) at transcriptional level. It is also found that a number of TF control the assimilation of carbon sources, amino acid assimilation (gal, adr1, aye7, myg1-2, put3, arg81, leu3, gln3 and gcn4), nitrogen compound degradation (dal80–81, gzf3, gat1), stress response (msn4, sut1, yap1–6, sko1, smp1 and msn2) and sporulation (rim101). A detailed description of biological functions derived from manual analysis of module A and yeast DB is available in SI.†

3. Discussion

The final goal behind any network-based analysis is in most cases to reach a systematic picture of what is relevant and what is not. According to this philosophy, we have presented a comparative study of the best known up-to-date GRNs by exploiting their most fundamental trait: the principle of causality. The GRN can be seen as a roadmap of the computational processes linking external stimuli with adaptive cell responses and this can be reduced to the wiring pattern of causal dependencies among TFs and their target genes. Although here we ignore any dynamical features, stability properties and other relevant components of GRN complexity, the construction process is unambiguous and leads to a unique, logical description of the causal core of a GRN.

By using the directed nature of the network, we can systematically reduce network complexity from thousands of elements to a much smaller subset including all the causally relevant parts of the GRN. Such a causal/computational perspective is consistent with the nature of regulatory maps. No matter how they behave exactly (and thus how they would be modelled) the set of non-reducible causal links should capture most of what is dynamically relevant. The approach presented here combines the use of leaf removal and SCC identification algorithms for directed networks. These methods do not make any assumption or require any probabilistic estimator providing a unique solution where the presence of a modular hierarchy can be easily depicted.

This approximation to the identification of the basic computational/causal units differs from others such as network motif identification. Dissecting networks into motifs provides a systematic classification of GRNs in their small building blocks by using the abundance of small subgraphs in a given network. However, the detection of a subgraph in a given net requires predefined decisions on what is to be measured. By contrast, SCC detection is not restricted to the search of a particular subgraph configuration and this makes possible the identification of entities that cannot be further partitioned due to the cyclic behaviour of this special set. As we stated in the Introduction section, cycles are at the basis of a non-trivial computation and, as we have shown, SCC is a topological concept that may be interpreted as a sort of dynamical modules (identifiable semiautonomous entities) within the framework of GNR. Attending to this, we have shown that these dynamical modules are organised within the core of computationally relevant dependencies that is the DB. Interestingly, this interpretation is also valid for any network depicting a relation of causal dependencies.

The analysis of GRNs shows that dynamical backbones are very small (compared with the whole network) and thus the basic logic of GRNs can be markedly simplified. The very small number of iterations required to obtain the DB suggests a direct control of the TF set over the pool of target genes, without a long chain of command responsible for information transmission. This is reinforced by the fact that in all cases more than 90% of genes were eliminated after the first iteration of DB pruning (data not shown). Functional analyses did not reveal significant differences between DB and the set of root nodes. Since they are at the top of the hierarchy, we could...
conclude that they might be related to stimulus sensing. However, we must stress that the TF–gene target relation is a part of the regulation process and TF and promoter alteration through cell signalling processes or metabolism can operate in different elements of the regulatory set. In this context, further efforts in the integration of different regulatory levels would contribute to clarifying this issue.

The strong hierarchical DB structure found for bacteria is consistent with previous observations.\textsuperscript{5,10,47} Our analysis reveals that bacterial GRNs are almost devoid of cycles, although they are not totally excluded, as previous studies have suggested.\textsuperscript{49} Moreover, our results indicate that these cycles are organised into dynamical modules with essential biological functions, such as module A in E. coli. The pattern of connections of this module reveals a non-trivial computational link between transcription and metabolic control under both normal and stress growth conditions. As we have shown, it is a common shared trait in bacteria despite the relatively simple organisation of B. subtilis when compared with E. coli (most probably due to a lesser knowledge of its regulatory wiring). In contrast with the prokaryotic DBs, the yeast DB is fairly different, with a high content of cycles organised into a single SCC (see ref. 46). Our analysis shows that this large dynamical module occupies a central position in the yeast DB, suggesting a bow-tie causal organisation.

It is worth noting that, according to their definition, SCC components may be substantially modified by the addition of new links or by rewiring them. In this context, novel findings in the gene regulatory circuitry of these organisms may alter the current picture of these networks. Now an interesting question arises. From an evolutionary perspective, mutations in regulatory boxes of target genes may alter the structure of SCCs. Then, it is reasonable to think that the regulatory system should be protected against this kind of perturbation. Interestingly, when comparing real networks with their null model counterparts both DB and SCCs are markedly smaller than what is expected by chance. At this point, the reduction of the dynamically relevant core and the hierarchical organisation of the smallest possible size SCCs (attending to dynamical requisites of the cell) would contribute to reducing the impact of this type of mutation. In addition, another layer of complexity must be considered to play a relevant role in minimising the impact of this type of mutation. This is the distribution of the activatory and inhibitory weight in the genetic wiring. Interestingly, it has been seen that an experimental rewiring of the E. coli gene regulatory network does not have a dramatic impact on organism surveillance.\textsuperscript{59}

Uncovering the genetic circuitry of these organisms is still an ongoing task, and our results can only provide a tentative approximation to the final picture. However, due to the good quality of data for the E. coli network, only slight modifications in the feed-forward architecture are expected. In the case of DB yeast, its cyclic character is a distinctive trait. Notice that the lack of knowledge of gene connections would not alter the cyclic character of DB. In addition, only a tree-like structure (similar to bacteria) could be suggested if further evidence demonstrates that current data are enriched with a high number of false connections causing loops in the network. This is a technical limitation in yeast due to the fact that TF–gene connections are defined according to a statistic value [see yeast GRN construction in Methods]. In our case, very restrictive conditions were used for yeast network construction in order to reduce the impact of possible false connections. This provides strong arguments for different strategies for GRN organisation being present in these prokaryotic and eukaryotic studied organisms.

With regard to the biology of bacteria and yeast, common patterns can be identified. Both share a unicellular organisation, simple life cycles (only sporulation and replication states for yeast and bacillus) and similar nutritional requirements (all of them are chemoheterotrophs absorbing molecules from the medium). However, a more complex behaviour appears to be associated with the cell cycle control in yeast. As a consequence, these relatively simple organisms may operate under comparable external inputs and a similar GRN organisation would be expected. This is consistent with the different output behaviour observed in yeast and bacteria. It is reasonable to think that this difference should originate with the different wiring displayed in these GRNs.

Yeast and bacteria drastically differ in their genomic organisation and cellular compartmentalisation, the two most distinctive features distinguishing eukaryotes from prokaryotes. It is widely accepted that spatial compartmentalisation and intronic organisation confer a more diverse regulatory behaviour\textsuperscript{60–63} and genomic plasticity. GRN structures could thus be understandable as alternative evolutionary solutions. More complex wiring in yeast, in contrast to bacteria, would be the basis for this differential computational behaviour.

4. Methods

4.1 Construction of GRNs

4.1.1 E. coli network definition and construction. GRN for E. coli is an overlapping of two files obtained from RegulonDB 6.0:\textsuperscript{34} NetWorkSet.txt, containing TFs and their target genes, and SigmaNetWorkSet.txt, containing the Sigma factors and the genes promoted by them. Both files contain information about the relations, as well as the activator/repressor behaviour, of TF (and Sigma factors) over the target genes. Biological information was obtained from the EcoCyc database.\textsuperscript{64} In this work we exclude elements contributing to a TF modification such as phosphorylation or ligand–TF binding. Graph pictures were created using Cytoscape software (http://www.cytoscape.org/).

4.1.2 B. subtilis network definition and construction. GRN for B. subtilis was obtained by the combination of gene, tfac and sigma field information compiled in DBTBS (release 5).\textsuperscript{35} Biological information was obtained from DBTBS, SubList\textsuperscript{65} and Uniprot\textsuperscript{66} databases. Graph pictures were created using Cytoscape software (http://www.cytoscape.org/).

4.1.3 S. cerevisiae network definition and construction. Yeast GRN was obtained from the compilation of different sources performed in ref. 67. Self-interactions were not initially included in that work and they were directly provided by the authors. Data correspond with highly confident experiments (\(P = 0.001\) and three positive replicas). Biological
information was obtained from Saccharomyces Genome database SGD\textsuperscript{56} and Uniprot. Graph pictures were created using Cytoscape software (http://www.cytoscape.org/).

### 4.2 Null model construction and analysis

Null model networks were obtained by a randomisation process of the real GRN consisting of twice the number of links rewiring events. A rewiring event is realised by an end node interchange in two randomly selected pairs. Then, the arrows of the new pairs are arbitrarily inverted. Once the new connections are obtained, the algorithm verifies that they were not previously in the network. If the rewired connection matches with existing links, the solution is rejected and the algorithm proceeds to a selection of two new links. The number of autoloops is kept constant in this randomisation process. This randomisation makes it possible to destroy local correlations keeping the number of nodes, links and degree distribution. A thousand randomised networks were constructed for each of the three GRNs. In addition, we also obtained null models for the set of nodes corresponding with TF activity, \textit{i.e.}, the nodes with $k_{\text{out}}>0$ (1000 randomised networks for each of the three GRNs). DB size, SCC size components and number of root nodes were calculated per null model network. Frequencies were normalised by the number of replicas (1000 for each of the six conditions, \textit{i.e.}, one GRN and the respective TF subset for each of the three studied organisms).

### 4.3 Statistical analysis of biological functions

Statistics for the estimation of the overabundance of biological functions were estimated for the DB set, root nodes and the dynamical module sets for the three different studied organsisms. A hypergeometric test with Benjamini and Hochberg false discovery rate correction, and selected significance level of 0.05 was applied using the gene ontology biological functions. In the case of \textit{B. subtilis}, gene ontology annotation was achieved by parsing information containing \textit{B. subtilis} taxon ID. The source file, gene\_association\_goa\_ uniprot (submission date 4/25/2009), was obtained from http://www.geneontology.org/GO.current.annotations.shtml. Specific \textit{E. coli} GO annotation was directly obtained from the same website. The analyses were performed using Bingo 2.3 Cytoscape plugging.\textsuperscript{68} Notice that gene annotation for yeast is included by default in this package. Detailed statistical results are provided in the SI2 supplementary file.\textsuperscript{†}

In addition, biological functions of the DB were collected manually from the respective organisations’ databases (EcoCyc and RegulonDB for \textit{E. coli}, DBTBS and Subtilist for \textit{B. subtilis} and SDG for \textit{S. cerevisiae}).

#### 4.3.1 Software implementation

Dynamical backbone algorithm and dynamical module detection were implemented in a software package using Perl language and unix/linux commands. This package is available for linux/unix platforms as supplementary material.\textsuperscript{†} Graph pictures were created using Cytoscape software (http://www.cytoscape.org/).

### 5. Mathematical appendix

#### 5.1 Causality, dynamics and topology

A directed graph (or digraph), $\mathcal{G} = \{V_{\mathcal{G}}, E_{\mathcal{G}}\}$, is constituted by a set of vertices, $V_{\mathcal{G}} = \{v_1, \ldots, v_N\}$, and the set of edges linking them, $E_{\mathcal{G}} = \{e_1, \ldots, e_k\}$. Formally, an edge from $v_j$ to $v_i$ is described by an ordered pair $e_k = (v_i, v_j)$, depicted by an arrow in the picture of the graph $v_j \rightarrow v_i$. An alternating sequence of vertices $v_1, \ldots, v_N \in V_{\mathcal{G}}$ and edges $e_1, \ldots, e_{k-1} \in E_{\mathcal{G}}$, defines a directed walk in a digraph $\mathcal{G}\textsuperscript{43}$ if there is a set $e_1, \ldots, e_{k-1} \in E_{\mathcal{G}}$ such that, for all $i < n$, $e_i = (v_i, v_{i+1})$. We denote a directed walk (if it exists) from $v_1$ to $v_n$ as $\pi(v_1, v_n)$. We can also define a walk between $v_1$ and $v_n$, \textit{i.e.}, a sequence of vertices and edges connecting $v_1$ and $v_n$ no matter the directed nature of the graph.

The number of outgoing links of a given vertex $v_i \in V_{\mathcal{G}}$ is known as $out$-degree (denoted by $k_{\text{out}}(v_i)$) whereas the number of incoming edges is the $in$-degree ($k_{\text{in}}(v_i)$). Once $k_{\text{in}}$ and $k_{\text{out}}$ have been defined, we can obtain the average connectivity or average degree, denoted by $\langle k \rangle$:

$$\langle k \rangle = \frac{1}{|V_{\mathcal{G}}|} \sum_{v_i \in V_{\mathcal{G}}} k_{\text{in}}(v_i) + k_{\text{out}}(v_i) \quad (1)$$

Furthermore, given a vertex $v_i \in V_{\mathcal{G}}$, it is interesting to define the set of vertices affecting it, to be denoted by $\Gamma_i$:

$$\Gamma_i = \{v_k \in V_{\mathcal{G}}: (v_k, v_i) \in E_{\mathcal{G}}\} \quad (2)$$

Attending to their reachability, a directed graph can display three types of components, namely:

1. The strongly connected components (SCC), which are the subgraphs for which every two vertices are mutually reachable.\textsuperscript{43,48}

2. The incoming components (IC), composed by the set of feed-forward pathways starting from vertices without in-degree and ending in SCC.

3. The outgoing components (OC) where pathways starting from SCC end in vertices without out-degree (see Fig. 2).

#### 5.1.1 Threshold network model

In this subsection, we briefly define the toy model used in the example provided in Fig. 1.

In a dynamical setting, the state $\sigma(t) = (\sigma_1(t), \ldots, \sigma_N(t))$ of a system $\sigma$ formed by $N$ elements would be updated under some class of dynamical process. An example of such dynamics is a threshold-like equation, in which the state of a vertex $v_i \in V_{\mathcal{G}}$ at time $t + 1$ is updated by the state of the vertices $v_k \in \Gamma_i$, namely

$$\sigma_i(t + 1) = \Theta \left( \sum_{j=1}^{N} \omega_{ij} \sigma_j(t) - \theta_i \right)$$

where $\Theta(x) = 1$ if $x > 0$ and zero otherwise. Here $\theta_i$ is a threshold and the weights $\omega_{ij} \in \{-1,0,1\}$ define the type of interaction between genes. If the state of each element is Boolean, \textit{i.e.} $\sigma_i \in \{0,1\}$, the previous model provides, for a given initial state, a closed description of the system's
behaviour. Here the matrix \( W = (\omega_{ij}) \) captures the structure and nature of causal links.

The small threshold network depicted in Fig. 1 starts from an initial state where \( \sigma_4(0) = \sigma_5(0) = 1 \) and \( \sigma_i(0) = 0 \) for \( i \neq 4, 5 \); for other units, a cyclic attractor (of period 12) is obtained. Here only two elements have a non-zero threshold, namely \( \theta_1 = \theta_4 = 1/2 \). Arrows and end circles indicate \( \omega = +1 \) and \( \omega = -1 \) links, respectively. The global state \( M(t) \) is given by \( M(t) = \sum_{i=1}^{N} \sigma_i(t) \) (thick lines, b).

5.2 Dynamical backbone pruning

Consider the pruning function \( \Phi : G \rightarrow B \), where \( B \subseteq G \). This function takes a directed graph as input, its output being the graph without all the vertices having \( k_{out} = 0 \) (and the links pointing to them). Accordingly,

\[
\Phi(G) = B_1 \{ V_{Bi}, V_{Ei} \},
\]

where

\[
\begin{align*}
V_{Bi} &\equiv \{ v_k \in V_G : k_{out}(v_k) > 0 \} \\
E_{Bi} &\equiv \{ (v_k, v_i) \in E_G : v_k, v_i \in V_{Bi} \}.
\end{align*}
\]

Thus, the computation is an iterative operation:

\[
\Phi(G) = B_1, \\
B_2 = \Phi(B_1), \\
\ldots \\
B_n = \Phi(B_{n-1})
\]

The resulting graph at the \( i \)-th iteration is denoted by \( B_i = B(V_{Bi}, E_{Bi}) \) and the computation ends when no further vertex elimination occurs, i.e., \( B_n = B_{n-1} \). Notice that vertices with \( k_{in} = 0 \), the root ones, can lose all their links in the pruning process. As we shall see, this special set of nodes is included by definition in the \( DB \). At every iteration, this collection of single root vertices defines a set \( r_i \) and, from these sets, we build the set \( R_i \) of all the single root vertices found until the \( i \)-th step:

\[
R_i = \bigcup_{k \leq i} r_k
\]

We now have all the ingredients to provide a formal definition of the dynamical backbone of a directed graph \( G, DB(G) \). Let us assume that, when performing recursively the operation \( \Phi \) over a directed graph, we reached the stable state, i.e., \( B_n = B_{n-1} \). The dynamical backbone is a subgraph of \( G, DB \subseteq G \), defined as:

\[
DB(G) \equiv DB(V_{DB(G)}, E_{DB}) = DB(V_{Bi} \cup R_n, E_{Bi}).
\]

Once \( DB(G) \) is defined, we can also be interested in the fraction of the net that exclusively displays feed-forward structures, to be denoted by \( DB'(G) \), which does not coincide exactly with the graph \( DB(G) \), obtained from the set of vertices \( v_i \in V_G \backslash V_{DB(G)} \) and the set of edges \( \{ v_4 = (v_k, v_i) \in E_G : v_k, v_i \in V_{DB(G)} \} \). To properly identify it, we need to define the subgraph \( L_n \) as the set of connections linking \( DB(G) \) to \( DB(G) \) and the vertices linked to them. Note that this subgraph may display many components. Its main feature is that the links end in vertices outside the \( DB(G) \) but they come from vertices belonging to \( DB(G) \). We obtain the maximal feed-forward subgraph \( DB'(G) \) from

\[
DB'(G) = DB(G) \cup L_n.
\]

5.3 Dynamical modules and hierarchy

Let us assume that we are in the \( k \)-th connected component of \( DB(G) \). The \( i \)-th dynamical module of the \( k \)-th connected component of \( DB(G) \), \( DM_{ik} \), is a set of vertices (and the directed edges among them) that constitutes an irreducible unit of causal relations. As we said above, the existence of cycles inside the \( DB(G) \) is responsible for the possible non-trivial unit of causal relations. Thus, if the \( k \)-th component of the \( DB(G) \) is not a single root node, the concept of a dynamical module is topologically equivalent to the SCC. As there can be more than one dynamical module in a directed graph, we can define \( \Delta_k(G) \) as the set of dynamical modules of the \( k \)-th component of the \( DB(G) \). Interestingly, when the dynamical module is contracted to a single vertex, the resulting graph is a directed, acyclic graph (Fig. 3c). Such an operation is commonly referred to in the literature as the condensation of a graph.\(^{43}\) Accordingly, we can construct a new condensed graph,

\[
H_k(G) \equiv H_k(V_{H_k}, E_{H_k}),
\]

where \( V_{H_k} = \Delta_k(G) \) and \( E_{H_k} \) is the set of links connecting the different dynamical modules. In other words, we collapse the elements of every dynamical module into a single node and we allow links connecting different modules of the component of the \( DB(G) \) we are working in. Notice that, as a consequence, we cannot state that

\[
H_k(G) \not\subseteq G.
\]

Interestingly, when we consider these \( DM \)'s as single vertices of \( H_k(G) \), we obtain a feed-forward graph. It is precisely the feed-forward organisation which enables us to define an order relation among the elements of \( DB(G) \). Such an order relation can be interpreted as the dynamical hierarchy of the network’s dynamical core and it is defined among the different modules of \( DB_k \). If we define \( \Pi(H_k(G)) = \{ \pi_1, \ldots, \pi_m \} \) as the set of all existing directed walks over all nodes of \( H_k(G) \), we can define the order relation “\( \succ \)” as:

\[
(DM_{ik} \succ DM_{jk}) \iff (\exists \pi(DM_{ik}, DM_{jk}) \in \Pi(H_k))
\]

Such an order relation is called the Transitive closure of the graph \( H_k(G) \).\(^{43}\) The above order relation provides our definition of causal hierarchy.

Acknowledgements

We thank the members of the Complex Systems Lab for their fruitful suggestions. This work was funded by the 6th Framework project ComplexDis NEST-043241 (CRC), the James McDonnell Foundation (BCM) and the Santa Fe Institute (RVS).
