Estimating the divisibility of complex biological networks by sparseness indices

Tommaso Mazza, Alessandro Romanel and Ferenc Jordán
Submitted: 7th September 2009; Received (in revised form): 6th November 2009

Abstract

In order to understand the complex relationships among the components of biological systems, network models have been used for a long time. Although they have been extensively used for visualization, data storage, structural analysis and simulation, some computational processes are still very inefficient when applied on complex networks. In particular, any parallel simulation technique requires a network previously divided into a number of clusters in numbers equal to that of the available processors. At the same time, let maximally disconnected clusters be chosen in order to minimize extra-communication overhead and to optimize the overall computational efficiency. Obtaining such a disconnection becomes a computationally hard problem when disconnection conditions are complex in themselves, like in the case of parallel simulation. Before applying any clustering method, topological indices might contribute to give an a priori insight about the divisibility of a network. Here we present a class of them, the sparseness indices. As particular topological indices provide either local or global quantification of network structure, they can help in identifying locally dense, but globally sparsely connected subgraphs.

Keywords: biological networks; graph sparseness; parallel computing; partitioning

Corresponding author. Tommaso Mazza, The Microsoft Research, University of Trento Centre for Computational and Systems Biology, Piazz Manci 17, 38123 Povo (Trento), Italy, Tel: +39 0461 282821, Fax: +39 0461 282814. E-mail: mazza@cosbi.eu

Tommaso Mazza was born in Catanzaro, Italy, on 8 June 1979. He studied Computer Science Engineering at the University of Calabria and received a PhD in Computer Science and Biomedical Engineering in November 2007 from the ‘Magna Græcia’ University of Catanzaro. In 2004, he worked on the Cofin 2003 Project titled ‘Tumori ereditari della mammella: studi genetici ed analisi del proteoma’ (‘Hereditary tumours: genetic studies and proteomic analysis’). In 2005, he worked on the design of a common interface to translate mass spectrometry raw data (MALDI-TOF and LC) into mzData. In March 2006, he visited CoSBi, where he dynamically collaborated in the design and implementation of Cyto-Sim, a stochastic simulator of biochemical processes. Moreover, he worked on the definition of a common interface for unmarshaling SBML models into the Cyto-Sim syntax and vice-versa. In summer 2006, Tommaso visited Microsoft Research Cambridge where he investigated the possibility of parallelizing and distributing simulation algorithms. In 2007, he joined the Bioinformatics Italian Society (B.I.T.S.). Tommaso joined CoSBi in January 2008.

Alessandro Romanel attended the Faculty of Science at the University of Trento (Italy), where he received a PhD in Computer Science in 2010. In 2004, he worked as a postgraduate researcher at CREATE-NET International Research Center. From September until December 2008, Alessandro was a visiting student at the Computer and Information Science Department at the University of Pennsylvania, and was supervised by Professor Benjamin C. Pierce. Currently, he is carrying out his research activities in the field of computational systems biology at CoSBi.

Ferenc Jordán was born in Budapest, Hungary, on 11 June 1973. He studied Biology (MSc 1996) and received his PhD in Genetics (1999) from Eötvös University, Budapest. Ferenc was an assistant professor at the Department of Genetics, Eötvös University (1999–2002), a postdoctoral researcher at the Department of Ecology and Plant Taxonomy, Eötvös University (2002–2003), an associate professor at the Institute of Ecology and Botany of the Hungarian Academy of Sciences (2003–2006) and a research professor at the Animal Ecology Research Group of HAS (2007–2008). He was also a Junior Fellow (2000/2001) and Branco Weih Fellow (2003–2008) at the Collegium Budapest, Institute for Advanced Study. The latter was supported by the Society in Science foundation, ETH, Zürich, Switzerland. Ferenc has given a number of talks in different countries (from India to Taiwan and from the United States to South Africa) and maintains a rich network of collaborators in 8–10 countries. He was a participant in the Darwinian Homeland Security workshop organized at NCEAS, Santa Barbara, CA (2005), and is an editor for Ecology Letters. Ferenc joined CoSBi in October 2008.
BACKGROUND
Computational systems biology has increasingly relied on the expressive power of graphs, as mathematical models for studying biological interactive systems. Network modeling is supported by the rapidly developing capacity of computers, using some unambiguous and artificial formalism. The most common one is using coupled ordinary differential equations (ODEs) [1]. Alternatively, several other promising modeling techniques have been employed, including Boolean networks [2], Petri nets [3], Bayesian networks [4], graphical Gaussian models [5], process calculi [6,52] and Automata theory [7]. Standard, xml- (SBML [8], CellML [9]) or graphical- (SBGN [10], BlenX4Bio [53]) based languages have been further proposed to share and integrate our knowledge. All of the above draw particular inspiration from graph theory, since they share a common simple principle: interacting agents (e.g. genes, proteins, enzymes, etc.) are represented by graph nodes and their interactions (e.g. dimerization, phosphorylation, collision, etc.) are represented by graph edges. The relation between the sets of nodes and edges is always problem-specific (e.g. protein A is linked to protein B if and only if protein A phosphorylates protein B). The nature of connections (e.g. strength, frequency, symmetry) is usually modeled by weighting the edges (e.g. by kinetic parameters of enzymes catalyzing reactions).

The advantage of network models in computational systems biology is twofold. First, on the quantitative side, they make it possible to summarize a huge amount of information on both nodes and edges, i.e. agents and interactions among them. It was not possible much earlier, even if the need for network studies were evident (see [11] or later [12]). Second, on the qualitative side, it is clear that by integrating, not only summarizing, this information, we are able to reveal emergent properties as well as to better understand the relationships between various levels of biological organization (e.g. the effect of a gene knock-out experiment on the performance of the whole metabolic network (see [13]).

Apart from this hierarchical complexity, we are also interested in the relationship between structure and functioning [14], as well as its spatio-temporal aspects [15]. Furthermore, edges in relevant network models can be directed (or undirected, i.e. symmetrical), weighted (or unweighted, i.e. binary) as well as signed (or unsigned, i.e. homogeneous). Particular problems call for the adequate data type and network model. For example, the direction of network edges is typically very important information: several biochemical reactions, gene regulatory or catalytic processes are irreversible, represented by a directed edge in the graph model (in other cases, considering the direction of edges is less important than weighting them). This diversity of data, approaches and problems poses a big challenge for computability and efficiency in computational systems biology.

Recent progress in modern network analysis, aided by the tools of computational systems biology, has provided fundamental information on biological networks. We know better their local [16] and global [17] properties, the structure of their dynamic relationships [18] and, very importantly, our comparative knowledge is rapidly increasing [19]. Beyond the local-scale analyses (e.g. positional similarity of network nodes, see an application in [20]), the most important global results concern the complexity, vulnerability, robustness and connectivity of biological networks. Connectivity is among the oldest studied properties of biological networks [21], deserving continuous attention as probably being a proxy to the previous properties. Connectivity itself encompasses a number of concepts and measurements. It can be a mathematical quality (whether the graph is connected or it is composed of several components), a quantity (to what extent a graph is connected, how dense is it, how large is the average distance, etc.) or a functional property (whether efficient flows are realized on the graph; see a study on disease spread on the illness network in [22]). All these issues are interrelated and describe another aspect of connectivity.

NEED FOR PARTITIONING
Apart from the existing graph-based representation languages and beyond exploiting their own intrinsic properties, scientists have developed a plethora of methodologies to investigate new theoretical conjectures. One of the most known analysis techniques is simulation. Its main characteristic lies in the ability to capture the dynamics of the modeled system, that is, to explore how formally encoded input models change over time. The goodness and reliability of the outcome of a simulation process depend on the accuracy of the information considered by the model, as well as by its intrinsic nature.
Stochastic and deterministic simulators contended the ‘gold medal’ in working with noisy models with few chemicals and with smooth-trended models with large populations of chemicals, respectively. Neither of them has ever been proved to be superior to the other, because both suffered relevant limitations. Whereas the stochastic simulation algorithm (SSA) by Gillespie [23] and the most derivations require that systems are (strictly) well-stirred mixtures of colliding molecules and that the non-reactive collisions are more common than the reactive ones, the deterministic approaches completely disregard the random fluctuations, which are typical of individual genes, mRNAs and proteins. Moreover, a general drawback is common: all concurrency aspects are neglected and hence the real biological parallelism phenomenon looks forcibly flattened on a sequential space.

Although the parallel simulation of biological systems is a really tricky task, the additional computing power provided by multi-processor computers has encouraged scientists to investigate two complementary approaches. Multiple replications in parallel (or MRiP) means parallelizing across the simulation, namely computing simultaneous and independent stochastic trajectories of the same system (e.g. [24–26]), while single replication in parallel (or SRiP) means parallelizing across the method, namely dividing up the realization of a stochastic trajectory among each available processors and coordinating the process (e.g. [27–29]). Contrary to the MRiP methods, any development in SRiP is drastically delayed by the problematic task of partitioning the so-called workload graph (WG), a graph that represents the dependencies between computational units and/or input data.

In a WG, nodes and edges denote computational units and communication among them, respectively. Partitioning of the WG means aggregating some of its subgraphs into single clusters (we note here that ‘clusters’ should not be confused with ‘clustering coefficient’, defined in Table 1). This is based on a simple and general criterion: communication between clusters (i.e. the number and weight of cutting edges) should be minimized. Moreover, it is based on some ad hoc conditions (i.e. inter-cluster path length should be minimized, inter-cluster loops should be avoided, etc). A whole set of solutions exists in the literature, mentioning here only the two most important families, geometric and structural algorithms. Solutions in both families are based on bisection, according to which a partition is determined by recursive application of a division procedure that splits the original WG into two disjoint subgraphs.

Geometric algorithms require that nodes of the input graph be described by geometric coordinates, which are used to calculate the bisection. Furthermore, geometric algorithms rely on the assumption that the probability of the edge between two nodes $i$ and $j$ (connectivity) is proportional to the Euclidean distance of $i$ and $j$ (their geometric proximity): a strong assumption that is typically unrealistic. Systems that require these algorithms are those for which the macroscopic description of kinetics is usually insufficient. In other words, they are spatially heterogeneous systems where intracellular diffusion of chemicals is a limiting constraint. Therefore, when dealing with such processes, other than considering the geometry of the cell, it is mandatory to explicitly take into account the spatial heterogeneity (e.g. diffusion gradients). Molecules move at rates specified by diffusion coefficients and the distance of their random movement is determined by the Fick’s second law. Therefore, the assumptions made by Gillespie (thermal equilibrium and instantaneous diffusion) typically do not hold here. Thus, SSA may work well locally, but cannot be surely used to represent complex systems with reactions that take place in inhomogeneous media. A proposed extension lies in discretizing the space, namely in subdividing it into logical sub-volumes, often referred to as cells (or compartments). The dimension of a cell is chosen to be small enough for the SSA assumptions to hold. This method, proposed by Bernstein [30], is depicted in Figure 2. Over the years, it inspired further developments (reviewed in [31]).

Structural algorithms, on the other hand, determine a bisection of the WG, based exclusively on graph topology. In particular, often a sort of dependency graph is employed. It is characterized by having biochemical reactions as nodes and inter-reaction dependencies as edges. We note here that the relevance of a given dependency varies with the propensity function of the reaction that causes the dependency. That is, a directed edge from $i$ to $j$ is weighted by the dependency function of the reaction represented by the $i$ starting node. More formally, let $R$ be the set of chemical reactions, let $R_A$ be a reaction in $R$ and let reactants($R_A$) be the set of reactant chemicals in $R_A$. We call propensity
of the reaction $R_A$, the product: $k_A \Pi_i [r_i]$, where $\forall r_i \in \text{reactants}(R_A)$, $[r_i]$ stands for the population of the species $r_i$ and $k_A$ for the kinetic law of the reaction $R_A$. Hence, a reaction $R_A$ depends on another reaction $R_B$ if and only if: $\exists r_i \in \text{reactants}(R_A) \mid r_i \in \text{reactants}(R_B) \cup \text{products}(R_B), i = 1, \ldots, |R| \ (\text{Figure 1}).$

Although in [32] Gibson and Bruck used this kind of graph just as a clever data structure to boost the performances of their simulation algorithm, other scientists made subsequently use of it to epitomize the overall burden of computation required to simulate biological systems. From this last perspective, they conjectured about the way of dividing such a graph and assigning its clusters to individual processors in a way to maximize the load balance and to minimize the inter-communication among processors. Apart from the intrinsic computational complexity of dividing such a graph in an optimal manner, the partitioning problem is further made heavier by the fact that weights on edges change dynamically over simulation, since the reaction propensities do so. This means that a suitable partitioning effort would work by statically dividing the WG at the beginning of the simulation and consequently rearranging the nodes every time that the

Table I: Some centrality and topological measures (considering a graph with $V$ nodes and $E$ edges).

<table>
<thead>
<tr>
<th>Topology measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>The number of edges incident upon a node. In a directed network, in-degree (the number of edges that point to a node) is distinguished from out-degree (the number of edges that start from it). Undirected networks are characterized by the average degree $k = 2E/V$.</td>
</tr>
<tr>
<td>Degree distribution</td>
<td>The degree distribution $P(E)$ denotes the probability that a selected node has exactly $E$ edges, i.e. the fraction of nodes in the network with degree $E$.</td>
</tr>
<tr>
<td>Clustering coefficient</td>
<td>The clustering coefficient of a node $A$ is defined as the probability that two nodes $B$ and $C$, which are connected to the node $A$, are themselves connected.</td>
</tr>
<tr>
<td>Average clustering coefficient</td>
<td>The average clustering coefficient for a whole network is the average of nodal clustering coefficients: it measures the ‘cliquishness’ of the network.</td>
</tr>
<tr>
<td>Path length, shortest path and mean path length</td>
<td>A path from node $A$ to node $B$ is an alternating sequence of nodes and edges, beginning with $A$ and ending with $B$, such that each edge connects its preceding node with its succeeding. The length of a path between two nodes is the number of such edges. In weighted networks, path length is the sum of the weights on edges. The shortest path between two nodes is the path composed of the smallest number of edges. In directed networks, the path length from $A$ to $B$ is often different from the path length from $B$ to $A$. The length of the shortest path between nodes $A$ and $B$ is the distance between $A$ and $B$. The mean path length is the average length of the shortest paths between all pairs of nodes.</td>
</tr>
<tr>
<td>Betweenness centrality</td>
<td>For node $A$ ($B_A$), it is the ratio of shortest paths going through node $A$.</td>
</tr>
<tr>
<td>Closeness centrality</td>
<td>For node $A$ ($C_A$), it is the mean distance from node $A$ to others. In directed networks, node $A$ has in-closeness and out-closeness values.</td>
</tr>
</tbody>
</table>

Figure I: Dependency graph construction: an example. $k_{i,m}, \ldots, u$ corresponds to the kinetic law of the $i^{th}$ reaction. Contrarily, $p_{i,m}, \ldots, u$ represents the propensity function of the $i^{th}$ reaction.
established minimization conditions between any two neighbor clusters were no longer respected. Various algorithms exist, dealing with this problem, and a detailed overview on the subject can be found in [33].

Irrespective of the kind of WG, we note that the literature seems to lack papers on studying the predisposition of WG to be partitioned. Thus, since the partitioning is a hard problem from a computational point of view, it is our belief that a set of indices intended to quantify a priori the tendency of a WG to be divided would be essential. The application of such indices to a WG would eventually help preventing the use of an expensive method (such as those introduced before) that would spend the most time in communication rather than in simulation, thereby resulting in a loss of performance (and in a wrong simulation). The problem that we will discuss is about the sparseness character of biological networks. We will review the most known sparseness indices and will provide the reader with some insights about how to combine them with other topological indices in order to estimate the overall network divisibility.

TOOLS FOR ESTIMATING THE DIVISIBILITY OF NETWORKS
A denser graph is widely meant as a graph in which the number of edges \((E)\) is closer to the maximal number of edges or, conversely, it is sparser when the number of edges is less than the possible number of edges. Preiss gave the first pseudo-quantitative definition of dense graphs [34]. Starting from the assumption that a graph is dense if \(E = \Theta(V^2)\) (where \(V\) is the number of nodes; \(V''\) stands for ‘vertex’) or, alternatively, sparse if \(E = O(V)\), he stated that a class of graphs might be considered dense if \(E = \Theta(V^k)\), with \(k\) chosen in a range between 1 and 2. As Dienstel also noticed in [35], this formulation of density makes sense only for families of graphs whose order tends to infinity; and hence, not for individual graphs.

Over time, a couple of further contributions tried to better elucidate these definitions. The density has been defined in terms of the edge to node ratio \(E/V\) [36] or as the ratio between the number of edges and the square of the nodes number \(E/V^2\). According to the former definition, graphs of proportionally increasing order and size show the same density estimation, but different relationships between the number of zeros and non-zeros in their adjacency matrices (i.e. how sparse is the matrix; see Figure 3). On the contrary, the latter index scales with order [37]. However, it exhibits a weak point. It fails in classifying a graph whenever the number of edges corresponds exactly to half of the square of the nodes. In such case, density equals 1/2 and the graph can be classified neither dense nor sparse.

Another measure, introduced in [38], values the relative density of a graph as the ratio between the number of its edges \((E)\) and that of a complete graph with the same number of nodes \((E^*)\). According to the graph typology, the relative density \((E/E^*)\) can be expressed in different forms. Some relevant examples are as follows:

1. \(2E/V(V + 1)\) for graphs with loops;
2. \(2E/V(V - 1)\) for graphs
3. \(E/V^2\) for digraphs with loops;
4. \(E/V(V - 1)\) for digraphs.

Although this index scales with orders and sizes (see Figure 3), it introduces undetermined values, as well. To overcome this issue, the authors of [39] proposed the compactness index, with the twofold aim of balancing the concept of sparseness and density and discarding the ambiguity of critical unclassifiable values. This index is the quotient of the relative density \(E/E^*\), calculated by the formula \(2E/V(V - 1)\), and the relative sparseness \(Z/Z^*\), obtained by the ratio between the number of zeros in the adjacency matrix \(A\) (the adjacency matrix of a finite graph \(G\) on \(n\) vertices is the \(n \times n\) matrix \(A\) where the non-diagonal entry \(a_{ij}\) holds the number of edges from vertex \(i\) to vertex \(j\), and the diagonal entry \(a_{ii}\) holds...
the number of loop edges from the vertex $i$ to itself. In this context, graphs have integer value entries ranging from 0 to 1) of a graph $(Z = V^2 - 2E)$ and the maximum number of allowed zeros $(Z^* = V^2)$. The final formula becomes

$$\rho = \frac{E/E^*}{Z/Z^*} = \left(\frac{V^2}{2E} - 1\right)^{-1} \left(1 - \frac{1}{V}\right)^{-1},$$

where $\rho$ has value 1 when the sparseness and density contributions are equal. This represents the critical value that separates dense and sparse graphs. Those with $\rho > 1$ are classified as dense, while graphs with $\rho < 1$ are classified as sparse. Every graph exhibits either a greater or a smaller $\rho$ number than the critical value, since this last is not analytically obtainable by the formula (i.e. the $\rho$ function gets asymptotically closer to 1, but never reaching it).

However, even though the compactness index succeeds to avoid undermined values, it fails in classifying graphs different from undirected graphs.

Referring to the kind of WG introduced in the previous section, we are particularly interested in directed graphs (the presence of loops is irrelevant in our scope). Thus, in order to make the index working also with them, the previous formula can be reinterpreted as the ratio between $E/E^* = E/V (V - 1)$ and $Z/Z^* = (V^2 - E)/V^2$. The final formula results

$$\rho_d = \frac{E/E^*}{Z/Z^*} = \left(\frac{V^2}{E} - 1\right)^{-1} \left(1 - \frac{1}{V}\right)^{-1}.$$

As before the unitary value of $\rho_d$ represents the critical value that separates dense and sparse graphs and no graphs can be classified to be at the boundary.

Figure 3: From left to right, the graph A, of order 5 and size 7, shows a quite balanced quantity of zeros and non-zeros, making it neither sparse nor dense. Graph B, with 10 vertices and 14 edges, displays 26 non-zeros on 86 zeros elements. Graph C displays 185 non-zeros on 40 zeros elements. In summary, they are matrices with the same density estimation but with incremental sparseness.
Note that for both undirected and directed graphs, the sparseness and denseness contributions in the formulas are not balanced within the domain, i.e. the denseness contribution is halved in respect to the sparseness one. We may conjecture here that this escamotage has been meant with a three-fold aim: (i) to make both formulas defined for all the possible value pairs \( [E, V] \) (indeed, the ratio \( Z/Z' \) can never be zero), (ii) for scaling purposes and (iii) to avoid critical values. However, even though these may be justifiable reasons, they are obviously questionable. In particular, we do not see much the need (at least in the context of the topic of this work) of having an asymptotic behavior over the critical value.

Referring to the fuzzy Harary’s sentence that we dug out of the footnote 1 at page 205 of [40] ‘In the literature, a sparse matrix has been defined as one with many zeros’ and considering the widely accepted definition of density given by Preiss in [34], we feel confident to consider dense those graphs that have \( \rho_d \) or \( \rho = 1 \). Furthermore, we reckon such an index be more general, usable on a wider class of graphs (e.g. directed, undirected, with/without loops) and balanced within its analytical domain.

These features can be obtained by slightly modifying the formulation of the compactness index. We call the new index: completeness. Instead of considering the ratio between the relative density and relative sparseness, we simply use here the ratio between the number of non-zero and zero entries in the adjacency matrix of a graph \( G \). The completeness index can be formulated as follows:

\[
\frac{E}{Z} = \frac{\sum_{i \in V} \sum_{j \in V} a_{ij}}{\sum_{i \in V} \sum_{j \in V} 1 - a_{ij}}
\]

\( E/Z = 1 \) is the critical value. Graphs with \( E/Z \geq 1 \) are classified as dense, while graphs with \( E/Z < 1 \) are classified as sparse. As said before, we explicitly assume that graphs on the border are dense. Moreover, complete graphs are not considered in the domain of the index (because they are clearly dense). Note that if not interested in self-loops, an alternative formulation of the completeness index is

\[
\frac{E}{Z} = \frac{\sum_{i \in V} \sum_{j \in V, j \neq i} a_{ij}}{\sum_{i \in V} \sum_{j \in V, j \neq i} 1 - a_{ij}}
\]

In Figure 4, we compare the completeness and compactness indices on randomly generated directed and undirected graphs. Although the two indices look really close in most cases, it can be noticed how their difference increases when the size and the density of the graphs increase. This is a direct consequence of the unbalanced formula of the compactness index which underestimates (and hence misclassifies) large and extremely dense graphs.

Although sparseness/density indices can give a first insight about how prone are graphs to be divided, they do not take the graphs peculiar topology into consideration. An easy consequence of this is that a very dense but inhomogeneous graph might be much more divisible than a very sparse but homogeneous one. The concept of homogeneity is here related to that of node degree. A homogenous graph has nodes with same degree. A graph is inhomogeneous in the opposite case. Literature offers a variety of global and local graph metrics that can be potentially used as complementation of the previously shown indices. Degree, as the most local network index, characterizes the immediate (direct) neighborhood of graph nodes (i.e. how many neighbors a node \( A \) has that equals \( E_{A} \)). Clustering coefficient for a node \( A \) provides information also about the edges among its neighbors: it is the density of the subgraph composed of its neighbors. For a pair of non-neighbor nodes, their distance is the length of the shortest path between them (see Table 1). There are a number of meso-scale network indices combining degree and distance, all measuring some aspect of centrality and position. Some of them are: betweenness centrality, closeness centrality, information centrality (see all in Table 1) and more sophisticated metrics. Global indices characterize the whole network by single numbers. These include metrics that are statistics for all over the nodes (e.g. average clustering, average distance) and ‘emergent’ metrics not defined locally for individual nodes (e.g. diameter: the largest distance in the network, or structural balance [41]). For a review on the application of local, meso-scale and global network indices in biology, see [42].

Networks divisibility is high if relatively dense clusters can be produced in such a way that the network of clusters is sparse. In this case, there is a higher chance to simulate the WG graph in parallel and more efficiently. Particular needs determine how to optimize between-cluster density and large-scale sparseness. The presence of nodes with high betweenness centrality may be a key to find optimal aggregation algorithms. One possibility is to use edge-betweenness, the form of betweenness centrality used to characterize edges instead of nodes [43]. According to this approach, it is possible (i) to find
the edge of highest betweenness and aggregate the neighbor nodes of their end-nodes and (ii/a) to find the edge of highest betweenness and aggregate its end-nodes, followed by the edge of the second highest betweenness or (ii/b) to find the edge of highest betweenness, aggregate its end-nodes, calculate again edge betweenness for edges and re-start the process. During these processes, the clusters can be characterized by density, while the network of clusters can be monitored by sparseness and number of cycles (see Pseudocode 1).

Several alternative tools exist to approach the problem, including the application of subgraph centrality [44] and communicability [45], the identification and analysis of dominator trees [46], measuring regular equivalence [47] and detecting network modules [48–50]. All these approaches can help in finding quantitative relationships among subgraphs (like clusters).

**Figure 4:** Comparison of the compactness and completeness indices. We generated randomly 100,000 directed and 100,000 undirected graphs over a range of the number of nodes varying randomly between 1 and 1000. The figures on the right show a zoomed view of those on the left.

**PSEUDOCODE 1**
Naïve clustering algorithm describing one scenario mentioned in the text (ii/b). It computes three sequential macro-steps. 1. The rank_edge routine ranks all the edges of a DG according to a chosen centrality measure (e.g. betweenness). An ordered edge list is returned. 2. The sub-routine (i) takes the highest-rank edge, (ii) merges its end-nodes with their neighbor nodes, (iii) calculates new centrality values and ranks the edges again and (iv) evaluates a termination condition (e.g. 'the number of computed clusters be less than the number of available cores' or 'the inter-cluster sparseness/centrality measure be under a certain threshold') and repeats until the termination condition is verified. 3. The last task optimizes the computed clustered DG by moving nodes through clusters so that inter-cluster paths and loops are minimized.
1. CALL rank_edge(DG) RETURNING edge_list
2. REPEAT
   2.1 POP edge
   2.2 MERGE neighbor nodes for end-nodes
   2.3 UPDATE edge_list
   UNTIL NOT EVALUATE (termination.condition)
3. OPTIMIZE DG

However, the combination of sparseness indices and centrality measures may help in two ways. First, the network can be aggregated based on its properties quantified by centrality measures (e.g. edge-betweenness) and the aggregation process can be tested against the sparseness measures of aggregates (here, termination conditions are defined by sparseness indices). Second, the exact aggregation rules themselves can be defined by the local sparseness properties of subgraphs (for example, n-cliques can be automatically aggregated) and the centrality of aggregates is then studied. Combining centrality measures with sparseness indices, we can provide an a priori assessment of how clusters will be related to each other: to what extent they are connected, how hierarchical is their ordering, how global link density is locally distributed and, for example, what is the minimal number of relevant clusters.

CONCLUSION
In order to simulate and predict the behavior of natural systems, it is of key importance to understand their complexity. This mostly means the rich network of interactions among biological components. Since huge numbers of interactions happen at the same time, parallel computational techniques are essential for realistic modeling. We note here that the problem is very general: one of the key challenges for evolutionary biology is that selection acts on each trait in parallel; sequential thinking based on single-locus models can be totally misleading. However, parallel computation cannot be applied blindly: in order to assess whether it is reasonable, we need a priori knowledge on the divisibility of biological networks, i.e. whether structurally defined subgraphs can be functionally quasi-independent. Since these tools basically apply for all kinds of networks, they may also help to better understand the differences between basic topological classes of networks [51].

Over the years, parallel and distributed computing relied on both the intrinsic parallelism of nature and the power of multi-processors architectures. On the one hand, real life exhibits various independent levels of parallelism (e.g. a given chemical reaction does not take place only for two molecules but for two large sets of the two kinds of molecules; hundreds of independent biological transformations take place at the same time). On the other hand, the characteristics of HPC architectures and software have improved dramatically. Thus, the very nature of large-scale computing has changed from systems relying on one or a few powerful custom-designed processors to scalable parallel systems or farms of computers. Complex problems (often mapped on complex networks) are now broken down into a set of smaller jobs that run concurrently on multi-processor machine nodes. Automatic procedures are approaching alternative parallel implementations and several, sometimes not trivial, ad hoc synchronization policies are coming up to support their final deployments.

In particular, partitioning WGs is a crucial step since the quality of the overall parallel simulation often depends on it. It has been frequently recognized as a renowned computationally hard task. The WG of a biological simulation is divisible if we can define a clustering method providing clusters of desirable properties. In order to assess this, we need two kinds of information. First, we have several techniques (sparseness indices) to measure the patterning of links (number and distribution). Second, we also have a number of network indices to quantify the relationships among subsets of nodes (subgraphs, clusters). The combination of these methods helps in determining how independent, how hierarchical or how connected various clusters are (i.e. divisibility). Providing exact solutions is still in progress, but it is already of critical importance to call attention for these unavoidable tasks.

Sequential thinking about complex systems is not simplifying the problem but pouring the baby out with the bathwater.

Key Points
- Biological networks represent the complexity of interactions among the components of natural systems (e.g. interactions among molecules in the cell);
- Dividing biological networks into maximally disconnected sub-networks is the main aim of parallel simulation techniques;
- Clustering methods are computationally complex;
- Topological network indices (like sparseness) might be useful to estimate the divisibility of networks before applying any clustering method on them.

References


