Stability Analysis of Biological Network Topologies during Stochastic Simulation

Tommaso Mazza
Centre for Integrative Biology (CIBIO)
University of Trento
Via delle Regole, 101
38123 Mattarello (TN) - Italy
tommaso.mazza@unitn.it

Davide Prandi
The Microsoft Research - University of Trento
Centre for Computational and Systems Biology
Piazza Manci, 17
38122 Povo (TN) - Italy
prandi@cosbi.eu

ABSTRACT

Recent advances in the stochastic simulation of biological systems have exploited the *weighted dependency di-graph* as a compact representation of the computational workload. It was largely used to represent the causal relationships among reactions and then to determine their cause-effect implications. Although critical for several applications, the topology of the dependency graph has been little studied so far.

Here, we make use of some network topology indices to detect and characterize the *important* reactions of two real case studies. We measure the stability of such indices over time and make a case for considering them in parallel stochastic simulation.

Categories and Subject Descriptors

G.2.2 [**Graph Theory**]: Graph algorithms; I.6.8 [**Types of Simulation**]: Monte Carlo; J.3 [**LIFE AND MEDICAL SCIENCES**]: Biology and genetics

1. INTRODUCTION

Natural phenomena, ranging from chemical to ecological systems, are frequently summarized by networks, $G = (V, E)$, where the set V of vertices represents actors (e.g., atoms, proteins, cells, organisms), and the set E of edges collects their relationships (e.g., collisions, bindings).

Research on biological networks has considerably capitalized on the way of discovering/classifying the architectural structure of the biological systems. Topology indices were largely employed to measure static properties like *essentiality*, *lethality*, *fragmentation*, *reachability*, *connectivity* [20] on real case studies (see e.g., [1, 7, 8, 12, 13]). However, having nature also a strong dynamic component, other properties like *competition*, *cooperation*, *affinity* and *dissimilarity* were contemporarily targeted by complementary investigations. Hence, static and dynamic properties are deemed to be two precious fragments of a *Rosetta Stone*, whose joint understanding is the key for deciphering the functioning of nature.

SIMUTOOLS 2011, March 21-25, Barcelona, Spain

DOI 10.4108/icst.simutools.2011.245590

$$r_1 : X + X \xrightarrow{k_1} Y + Y$$
$$r_2 : X + Y \xrightarrow{k_2} X + X$$

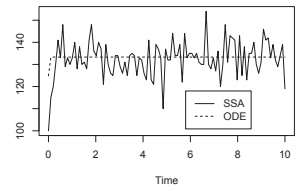


Figure 1: A biochemical system with two species, two reactions and same initial conditions: $| X |= 100$, $| Y |= 100$, $r_1 = 1$, $r_2 = 1$. **It is simulated with an SSA- and an ODE-based algorithm.**

Simulation is a notable representative of the dynamic analysis techniques and it is used to predict the temporal behavior of target systems. By simulation, a system (like the one in Fig. 1(a)) is realized in a number of traces, which may significantly vary if produced under continuous rather than discrete assumptions. As a matter of fact, traces in Fig. 1(b)) that have a similar steady state, significantly differ because of a *noisy* component. Several real case studies (e.g., [17, 22]) have proven the importance of such a stochastic component.

The Stochastic Simulation Algorithm (SSA) [10] is the de-facto standard algorithm for the stochastic simulation of biological systems. SSA considers a set $\{S_1, \ldots, S_N\}$ of well stirred biochemical species that evolve through $M > 1$ reactions $\{R_1, \ldots R_M\}$. The algorithm produces a trace in the solution space of the *Chemical Master Equation* [11] by computing the *next reaction density function* $p(\tau, j|\vec{s}, t) = a_j(\vec{s})e^{-a_0(\vec{s})\tau}$. This function defines the probability that in the current state $\vec{s}$ and at the time t, the next reaction will be R_j and will occur in the infinitesimal time interval $[t + \tau, t + \tau + dt)$. The *propensity function* $a_j(\vec{s})$ is proportional to the number of possible active instances of the reaction R_j [11]. For instance, the reaction r_2 of Fig. 1(a) has a propensity $a_2(\vec{s}) = c_2 \times | X | \times | Y |$, where $| X | \times | Y |$ is the number of active instances of r_2 in the current state, and the constant c_2 depends on the physics of X and Y. The algorithm iterates by taking a random sample for τ and j at each step, and then by updating $\vec{s}$ and, consequently, $a(\vec{s})$.

The computational precision provided by the stochastic

simulation comes at the price of a considerable simulation time or, even, of the overall impossibility to simulate large systems [2]. Parallel stochastic simulation aims at making the overall computation feasible by distributing the simulation of a single trajectory to many processing units. To do that, it is strictly required to optimally partition the workload (i.e., the reactions) of a simulation, conveniently represented by the *Dependency Graph* (DG) [9], into groups of reactions that are as much as possible independent [6, 16]. The nodes of a DG are reactions, linked by an edge if the execution of one changes the propensity of the other.

To capture the strength of the dependency between any two reactions i and j, we here consider a weighted DG (wDG), where the label of an arc (i, j) is the propensity, $a_i(\vec{s})$, of the reaction i. Over a wDG, we compute some topology indices in order to gather some information about the structure of the network. But such indices have been claimed to statically work on wDG, then an interesting problem arises: are they able to capture also the dynamical aspects of a wDG? In other words, given two reactions i and j and a topological index C that maps reactions into reals, from $C(i) > C(j)$, can we conclude something about the dynamic behavior of i and j by C? A positive answer would enable us to use such indices in the partitioning of the wDG and to easily simulate it in parallel.

In this work, we propose some preliminary investigations of the interplay between topological indices and stochastic simulation. The organization of the paper reflects this purpose. In Sect. 2 we provide the methods used in this work. We present (i) a formal definition of the wDG, (ii) the rationale and the mathematics behind some topological indices and (iii) some implementation aspects of our analysis library, which combines stochastic simulation and topological indices computing. Then, Sect. 3 presents the way we mixed stochastic simulation with topological analysis and gives some preliminary measures of the variability of the proposed indices over two real case studies. Sect. 4 concludes the paper.

2. METHODS

Formally, let $\mathcal{R}$ be a set of chemical reactions; let R_i and R_j be two reactions in $\mathcal{R}$; let $reactants\{R_i\}$ be the set of chemical reactants in R_i, and $product\{R_i\}$ be the set of chemical products of the reaction R_i:

DEFINITION 1. *R_j depends on (or is influenced by) R_i if there exists at least one species $s \in reactants\{R_j\}$ such that $s \in reactants\{R_i\} \cup products\{R_i\}$.*

On the basis of this definition, any biological system can be represented in terms of the dependency of its constituent reactions as a wDG. Vertices embody the biological interactions (i.e., reactions) and edges represent the causal dependencies among vertices in terms of the functions $a_j(\vec{x})$. Therefore, the relevance of a given dependency varies with time, since it equals the propensity of the reaction that causes it; a given edge has thus a weight which is the propensity of the node (i.e., reaction) where it issues from.

2.1 Centrality indices

In order to dig critical nodes out of a wDG, we put stress on different definitions of node centrality and we borrow some methods from the social science. *Degree, closeness, graph, betweenness and stress* centrality measures [23, 21] are just few among a many available indices. Some are based on enumeration of links or shortest paths (degree, stress and betweenness) and others derive from the measurement of the distances between pairs of nodes (graph and closeness).

Let us define a *path* from $s \in V$ to $t \in V$ as an alternating sequence of vertices and edges, beginning with s and ending with t, such that each edge connects its preceding with its succeeding vertex. The *length* of a path is the sum of the inverse weights of its edges. The idea is that faster reactions, i.e., those with the highest rates/weights, minimize the distance between nodes. Therefore, we compute the distance between two vertices s and t, written $d_G(s, t)$, as the minimum length of any path connecting s and t in G. By definition, $d_G(s, s) = 0$ for every $s \in V$. Note that, it is not required that $d_G(s, t) = d_G(t, s)$.

Degree centrality is based on the idea that important nodes are those with the largest number of ties to other nodes in the graph. It is often interpreted in terms of the immediate involvement of nodes in relationships established through the network. Let w_{us} be the weight of the arc connecting node u with node s, the degree centrality of a node u is:

$$C_d(u) = \sum_{u,s \in E} w_{us}$$

according to which the higher is the degree value, the more important (connected) is the node.

Closeness centrality is defined in a metric space where nodes are ranked because of their *geodesic distances* or 'proximity' to other nodes of the graph. Indeed, an important node is typically 'close' to, and can communicate quickly with the other nodes in the graph. In other words, it can be regarded as a measure of how important is a node in relationship to the reachability of any other node. Closeness is computed as

$$C_c(u) = \frac{1}{\sum_{t \in V \setminus u} d_G(u, t)}.$$

"Shallow" vertices have higher closeness.

Graph Centrality is the invert of the maximum of all geodesic distances from a node to all other nodes in the network. In not-connected networks, the centrality values of all nodes will be zero, since the distance to some nodes is infinite. It is formulated as:

$$C_{ce}(u) = \frac{1}{max_{t \in V \setminus u} d_G(u, t)}$$

Nodes with high C_{ce} have short distances to all other nodes.

Betweenness centrality measures the influence a node has over the spread of information through the network. Betweenness, in its basic version, is computed as the fraction of shortest paths between node pairs that pass through the node of interest. Its mathematical expression is

$$C_b(u) = \sum_{s \neq u \neq t \in V} \frac{\sigma_{s,t}(u)}{\sigma_{s,t}},$$

where $\sigma_{s,t}$ is the number of shortest paths from s to t, and $\sigma_{s,t}(u)$ is the number of shortest paths from s to t that pass through a vertex u. Vertices that occur on many shortest paths between other vertices have higher betweenness.

Stress centrality was designed to answer the question: 'how much work is done by each vertex in a communication network?'. The assumption is that, counting the number of

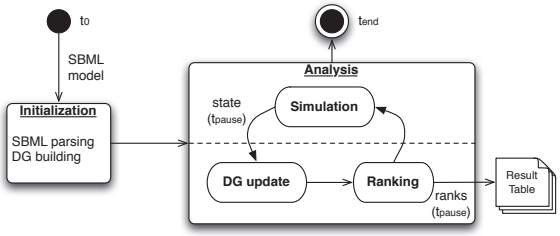


Figure 2: Pipeline of the simulation-ranking process. At time t_0 an SBML model is parsed and a related DG is built. DG is simulated by SSA and step-wise ranked by the 5 indices at each t_{paused} time step. Intermediate results are collected until time t_{end}.

shortest paths that contain an element u, gives an approximation of the amount of 'work' or 'stress' the node has to sustain in the network. It results from the sum:

$$C_s(u) = \sum_{s \neq u \neq t \in V} \sigma_{s,t}(u).$$

The higher $C_s(u)$ is, the more central (and busy) is u.

2.2 Implementation

The interleaving tasks of simulating and of measuring the importance of reactions have been implemented in the analysis library we make use of in this work[1]. It is written in `C#` and runs within `.Net framework 4`. It is made up of 4 modules that are respectively in charge of (i) loading a biological system written in `SBML` (Systems Biology Markup Language) [14], (ii) building the corresponding wDG, (iii) step-wise simulating the wDG and (iv) step-wise ranking the reactions through the indices (see Fig. 2).

`SBML` files are parsed through the `libSBML` library [4]. In this version of the library, we extract information about *species* (i.e. species names and initial copy numbers) and *reactions* only. Reactions are further dissected in their constituent pieces (i.e. *reactants* and *products*, stoichiometry values, and *kinetic laws*, which embody the propensities of the reactions). This minimal set of information is enough to build a wDG. The job is done by means of `NodeXL` [24], a third-party library.

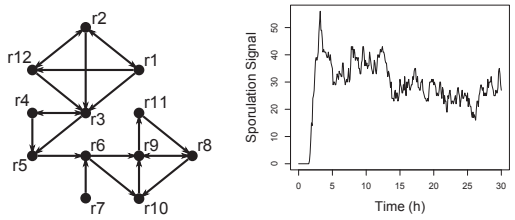
The wDG undergoes simulation by a tunable implementation of the SSA (*direct method*). From an initial state, our SSA is designed to simulate until a chosen time t_{pause}, and then, to return the current *state* of the system. The state of the system lies in the copy number of the species and, therefore, in a new set of weights of the edges of the wDG. At each pause, the wDG is updated with the latest system state and then ranked by the centrality indices.

Indices are based on two main concepts: *(shortest) path* and *distance*. Distance depends on the weight of the edges, in turn. Weight is a real positive number which equals zero if the propensity of the reaction is zero, i.e., the reaction is not enabled in the current state. We compute distances and paths together by a custom modification of the *Dijkstra's* algorithm. It returns a `path` for each pair of connected vertices, i.e. a list of `Dictionary<IVertex, Node>`, where `IVertex` is a node of the wDG and `Node` is a `struct` that stores both a link from `IVertex` to any preceding vertex and their distances. During the construction of all paths, the information about the number of shortest paths from any pair (s,t) of vertices that some other vertex (u) lies on is accounted (i.e., $\sigma_{s,t}(u)$), as well as the total number of computed paths (i.e., $\sigma_{s,t}$). Contrarily, the distance between any two vertices (i.e., $d_G(s,t)$), as well as the maximum distance between one vertex s and any other vertex $t \in V$ can be calculated off-line, namely by browsing the path(s) between them.

Summarizing, at each simulation pause, the wDG gets updated with the last computed state of the system and subsequently ranked by the 5 indices. The ranking process labels vertices with a value-pair made by the index name (taken out of an indices dictionary) and the computed score. Then, a snapshot of the wDG (vertices and edges metadata) is recorded in as many tables as the number of computed indices, where each row contains the time when the simulation paused and as many columns as the computed ranks, or, alternatively, as the number of reactions. The double process *simulation-ranking* repeats until simulation reaches a user-defined final time t_{end}.

3. EXPERIMENTAL RESULTS

We have tested our library over the models 37 and 260 of the BioModels Database [5][2]. With them, we have verified if the indices can capture the dynamic aspects of biological networks and if they can discriminate reactions and help in partitioning wDG.



(a) Dependency Graph (b) Simulation of Signal S

Figure 3: DG and stochastic simulation of model 37

Moldel 37 describes the kinetics of the sensory control of sporulation in Physarum polycephalum [19]. The model includes 12 reactions (see the DG of Fig. 3(a)) and describes the time evolution of the sporulation signal S (Fig. 3(a)). The signal S stabilizes around a value of 30, even if stochastic noise produces sensible variations in the steady state of S. Moreover, even if the network is quite small, it has a nice structure, because some nodes look like quite isolated while other nodes are more connected.

The five indices introduced in Sect. 2.1 are calculated every 0.1 hours, for a total simulation time of 30 hours. Fig. 4 summarizes these experiments. Two glyphs are used to denote the initial (▲) and the mean (●) values of the indices, which we have recorded during the simulation. Moreover, bars are plotted between the mean value plus and minus a standard deviation.

[1]The library is freely available upon request to the authors.

[2]We have chosen a more (model 37) and a less (model 260) noisy model. Both do not include continuous variables that are not currently supported by our library.

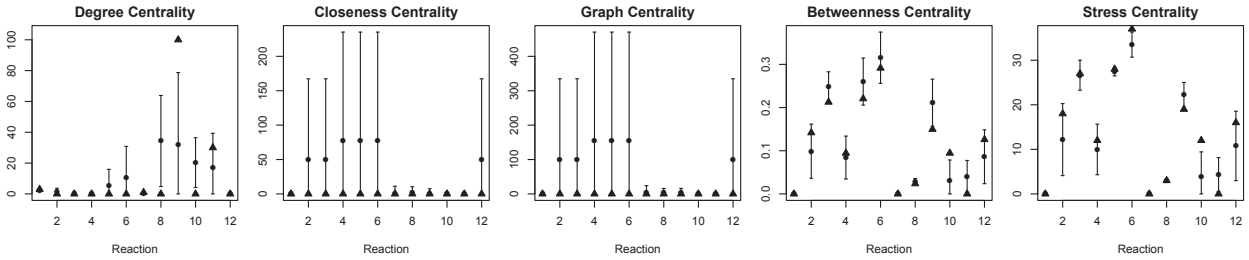


Figure 4: Centrality indices for each reaction of model 37: ▲ is the value of the index computed for the initial state; • is the mean value over the simulation time of Fig. 3(b) and bars represent standard deviation

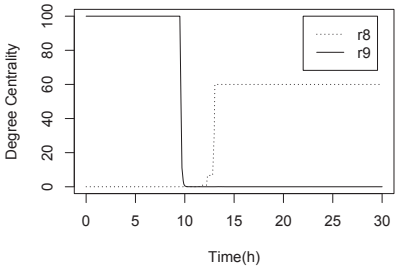


Figure 5: Time evolution of Degree centrality for reactions r9 and r10 of Fig. 3

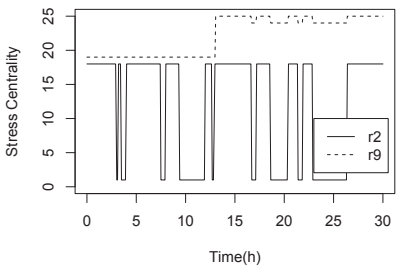
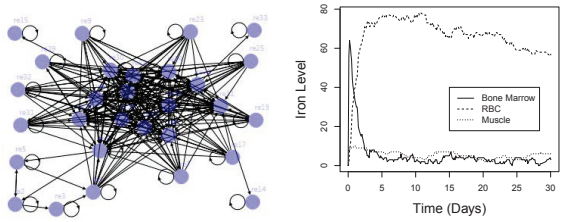


Figure 6: Time evolution of the Stress centrality index for reactions r3 and r9 of Fig. 3

With this setting we aim at evaluating if the computation of such static indices can give meaningful information about the dynamics of our systems. Concretely, we want to check how far the mean value of an index goes from its initial value during simulation. In any case, the evaluation of these indices can be easily misinterpreted. For instance, looking at the initial value of the Degree centrality index calculated for reactions 8 (r8) and 9 (r9), we may superficially conclude that r8 is less important than r9. However, the mean values of the two reactions confute this conclusion, since at the steady state it results that r8 is definitely more important than r9.

At a first inspection of Fig. 4, Degree, Closeness, and Graph centrality calculated at their steady state move greatly away from their initial values. Contrarily, Betweenness and Stress give more interesting results. Even if for some reactions the initial values of the indices are not within the boundaries of their standard deviations, the two indices are able to statically characterize the reactions r3, r5, and r6 as the most important. By stochastic simulation we confirm this observation. On the other hand, reactions r2, r9, and r12 are wrongly ranked. By inspecting the mean values of the Stress index we can conclude that r9 is more important than r2 and r12. The result is confirmed by Fig. 6, where the time evolution of the Stress index is plotted for the reactions r2 and r9. The same considerations are valid for Betweenness.

This bunch of experiments performed on the model 37 let us state that the indices that enumerate shortest paths (Betweenness and Stress) capture the dynamic behavior of a system better than those based on computing the distances between nodes (Graph and Closeness). This is due to the fact that distances among nodes (reactions) in a wDG can change over time, while the overall traffic is quite stable, i.e., nodes with highest importance are less sensible to the variation of time. Finally, Degree centrality seems to work badly in the context of this work, since it does not take into account the dependency among reactions (paths). We have tested these conjectures on the model 260 as well.



(a) Dependency Graph (b) Iron levels

Figure 7: DG and SSA simulation of model 260

Model 260 describes the dynamic behaviour of iron pools and fluxes in mammalian organisms [18]. In particular, the model traces iron levels in major tissues and organs for 28 days. Fig. 7(b) reports the time evolution of the iron levels in the bone marrow, the RBC, and the muscle tissue. The wDG in Fig. 7(a) has 29 reactions and it is a quite dense graph. The main difference with respect to model 37 is the impact of stochastic noise: the system reaches a stable state that is not very noisy.

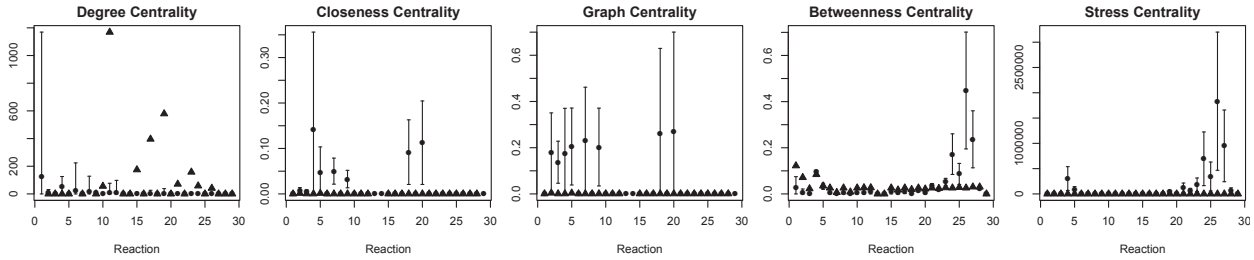


Figure 8: Centrality indices for each reaction of model 260: ▲ is the value of the index computed for the initial state; • is the mean value calculated over the simulation time of Fig. 7(b) and bars represent standard deviation

Fig. 8 plots a synthesis of our analyses. As for the previous case, we consider the initial value, the mean and the standard deviation for each reaction and for each index. A visual inspection of the plots is not really informative, even if it is evident that some reactions have initial and mean values sensibly different.

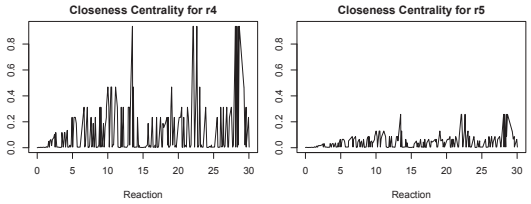


Figure 9: Time evolution of Closeness centrality for reactions r4 and r5

A deeper inspection of the time evolution of the indices gives some interesting insights. For instance, Fig. 9 shows the time evolution of Closeness centrality for reactions r4 and r5. It emerges that, even if model 260 is quite stable with respect to the stochastic noise, Closeness centrality is really noisy. The same is true also for the Graph centrality index. In the case of model 260, indices that enumerate shortest paths (Betweenness and Stress) do not behave better. For instance, Fig. 9 depicts the time evolution of Betweenness index for reactions r10 and r26. Even if on a different scale, the stochastic noise is dominant and does not allow to draw meaningful conclusion on the importance of these reactions.

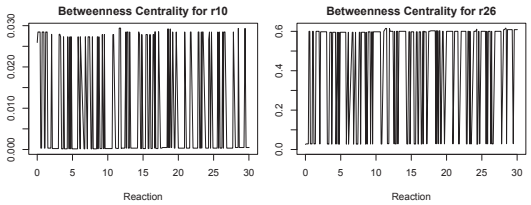


Figure 10: Time evolution of Betweenness centrality for reactions r10 and r26

We conclude the paper discussing these preliminary experimental results.

4. DISCUSSION

Most analysis procedures and routines acting on biological networks do their job mostly regardless of any dynamic considerations. In spite of this, we have here discussed the importance of considering the natural dynamism of some key parameters as, e.g., the reaction propensity in the field of stochastic simulation. In particular, we have studied the impact of change of 5 topological indices on weighted graphs. They have been calculated at fixed time during simulation. Index calculation and simulation were mixed in an interleaving process and tested on two real case studies.

Despite the preliminary stage of this work, some important implications came out. First of all, it emerged that none of the computed indices is almost constant during the simulation, namely, indices are not completely able to capture the dynamism of the biological systems. But by a deeper inspection we noticed a significant difference between results of both models. This advices us that the architecture of the considered network deeply affects the topological analysis, and therefore that it could be wrong comparing different networks on the only basis of the computed indices. In the case of model 37, Betweenness and Stress centrality give a sufficiently correct partition between more and less important reactions. Moreover, the static values of these indices, i.e., the values computed in the initial state, give meaningful information also about the dynamic evolution of the system. Unfortunately, this is not the case for the model 260. The motivation of this lies in the different structure of the wDG associated to the models: the network associated to model 37 is sparse, i.e., the overall number of dependencies is small in respect to the number of reactions, while the wDG of model 260 is quite dense. Nevertheless, this last consideration opens a panel of discussion in the field of parallel stochastic simulation.

Parallel stochastic simulation tries to reduce the total computing time by distributing the simulation of a trajectory to many processing units. A basic requisite is to find an (almost) optimal partition of the set of reactions, represented by a wDG. Reaction sets have to be chosen in a way that the propensity of the *boundary reactions* (those that link reactions not located in the same set) is the lowest possible. This guarantees that each two linked groups are maximally independent. As our experiments have shown,

dependencies change dynamically during a simulation and then require to be carefully monitored. In fact, large changes in the propensities of the boundary reactions would cause a forced run-time rearrangement of the groups in order to bring the boundary dependencies down. The analysis results on model 37 suggest that Betweenness and Stress could profitably point out reactions that are good candidate to be "boundary". The idea is that reactions with lower values of Betweenness (Stress) are less central to the network activity. In this context, the bad performances of the indices on model 260 are not surprising because, being the wDG dense, it is really challenging to find a good partition. These considerations suggest that parallel stochastic simulation could be a good approach only for sparse networks, that are, however, typical for biochemical systems.

Presented results are planned to be strengthened by the consideration that, sometimes, groups of nodes could be collectively ranked as more important than the sum of the individual ranks. Therefore, groups centrality indices will be included in future developments [3]. Furthermore, considering either only individual nodes or the whole groups could not be actually the key to assess the importance of a node. A *meso-scale* view could account for the strength of the direct effects of a node together with the indirect effects exerted on/by other nodes [15].

5. REFERENCES

[1] G. Bader and C. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(2):380, 2003.

[2] P. Ballarini, R. Guido, T. Mazza, and D. Prandi. Taming the complexity of biological pathways through parallel computing. *Briefings in Bioinformatics*, 2009.

[3] S. P. Borgatti. Identifying sets of key players in a social network. *Comput Math Organiz Theor*, (12):21, 2006.

[4] B. Bornstein, S. Keating, A. Jouraku, and M. Hucka. Libsbml: an api library for sbml. *Bioinformatics*, 6(24):880, 2008.

[5] L. Chen, D. Marco, R. Nicolas, and et al. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Systems Biology*, 4, 2010.

[6] L. Dematté and T. Mazza. On parallel stochastic simulation of diffusive systems. In *Computational Methods in Systems Biology*, page 191, 2008.

[7] A. M. Edwards, B. Kus, R. Jansen, and et al. Bridging structural biology and genomics:assessing protein interaction data with known complexes. *Trends Genet*, 18:529, 2002.

[8] E. Estrada. Virtual identification of essential proteins within the protein interaction network of yeast. *Proteomics*, 6(1):35, 2005.

[9] M. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, 104(9):1876, 2000.

[10] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340, 1977.

[11] D. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188(3):404, 1992.

[12] M. Girvan and M. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99:7821, 2002.

[13] Y. Ho, A. Gruhler, and A. H. et al. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868):180, 2002.

[14] M. Hucka, A. Finney, B. Bornstein, and et al. Evolving a lingua franca and associated software infrastructure for computational systems biology: the systems biology markup language (sbml) project. *Systems Biology*, 1(1):41, June 2004.

[15] F. Jordán and I. Scheuring. Searching for keystones in ecological networks. *Oikos*, (99):607, 2002.

[16] N. Jun-qing, Z. Hao-ran, C. Jiu-sheng, and et al. A Distributed-Based Stochastic Simulation Algorithm for Large Biochemical Reaction Networks. In *Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007.*, page 502, 2007.

[17] M. Kærn, T. Elston, W. Blake, and J. Collins. Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics*, 6(6):451, 2005.

[18] T. Lopes, T. Luganskaja, V. Spasić, M. Hentze, M. Muckenthaler, K. Sch ”umann, and J. Reich. Systems analysis of iron metabolism: the network of iron pools and fluxes. *BMC Systems Biology*, 4(1):112, 2010.

[19] W. Marwan. Theory of time-resolved somatic complementation and its use to explore the sporulation control network in Physarum polycephalum. *Genetics*, 164(1):105, 2003.

[20] T. Mazza, A. Romanel, and F. Jordán. Estimating the divisibility of complex biological networks by sparseness indices. *Brief Bioinform*, 11(3):364, 2010.

[21] M. Newman. The mathematics of networks. *The New Palgrave Encyclopedia of Economics*, pages 1–12, 2007.

[22] C. Rao, D. Wolf, and A. Arkin. Control, exploitation and tolerance of intracellular noise. *Nature*, 420(6912):231, 2002.

[23] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31:581, 1966.

[24] M. Smith, N. Milic-Frayling, B. Shneiderman, and et al. NodeXL: a free and open network overview, discovery and exploration add-in for Excel 2007/2010. WWW page (http://nodexl.codeplex.com/), 2010.