Algorithms for biological graphs: analysis and enumeration*

Andrea Marino

Dipartimento di Informatica, Università di Milano, Milano, Italy

The aim of enumeration is to list all the feasible solutions of a given problem satisfying some constraints. Enumeration algorithms are particularly useful whenever the goal of a problem is not clear and all its solutions need to be checked. Since one peculiar property of biological networks is the uncertainty, a scenario in which enumeration algorithms can be helpful is biological network analysis. Modelling biological networks indeed introduce bias: arc dependencies are neglected and underlying hyper-graph behaviours are forced in simple graph representations to avoid intractability. Moreover regulatory interactions between all the biological networks are omitted, even if none of the different biological layers is truly isolated. Last but not least, the dynamical behaviours of biological networks are often not considered: indeed most of the currently available biological network reconstructions are potential networks, where all the possible connections are indicated, even if edges/arcs and vertices are hardly present all together at the same time. More details about these aspects of the biological networks can be found in [8].

Our Contribution. We have shown four examples of enumeration algorithms that can be applied to efficiently deal with some biological problems modelled by using biological networks: enumerating central and peripheral nodes of a network, enumerating stories, enumerating paths or cycles, and enumerating bubbles. Notice that the corresponding computational problems we define are of more general interest and our results hold in the case of arbitrary networks.

1 Enumerating central and peripheral vertices

Structural analysis allows the identification of important and not important vertices within a network and also for this reason has become very popular in many disciplines. In the biological domain, the importance of a vertex can be defined in many different ways. With neighbourhood-based centrality measures, such as degree, the importance of the vertices is inferred from their local connectivity and the more connections a vertex has the more central it is. Closeness, eccentricity, and shortest path based betweenness relies on global properties of a network, such as distance between vertices.

* The author wants to thank the PhD advisor Pierluigi Crescenzi, the PhD School of Dipartimento di Sistemi e Informatica, Università di Firenze (Italy), and all the coauthors of the papers.

We have focused on the enumeration of the radial and diametral vertices, i.e. vertices that are central and peripheral according to the eccentricity notion of centrality, and on the computation of the radius and diameter of biological networks and of real world graphs in general. The diameter and radius of a graph are respectively the maximum and minimum eccentricity among all its nodes, where the eccentricity of a node x is the distance from x to its farthest node. Thus, intuitively, the diametral source vertices are the vertices that hardly reach the other ones, the diametral target vertices are the vertices hardly reachable from the other ones, and the radial vertices are the vertices that easily reach all the vertices of the network. In order to calculate the vertices that can be easily reached from any other vertex, it is sufficient to consider the transposed graph.

We have presented the DiFUB Algorithm, which is able to list all the diametral sources and targets and to compute the diameter of (strongly) connected components of a graph $G = (V, E)$ in time $O(|E|)$ in practice, even if, in the worst case, the complexity is $\Theta(|V||E|)$. Analogously, we have presented a new algorithm to list all the central vertices and to compute the radius of (strongly) connected components of a graph in *almost* $O(|E|)$ time in practice.

The analysis of real world networks in general, such as citation, collaboration, communication, road, social, and web networks, has attracted a lot of attention. The fundamental analysis measures have been reviewed in [12]. Moreover the size of these networks has been increasing rapidly, so that in order to study such measures, algorithms able to handle huge amount of data are needed. Since the algorithms available until now were not able to compute diameter and radius in the case of huge real world graphs, the contribution of our algorithms is not just limited to biological networks analysis, but extends also to the analysis of complex networks in general. We thus have shown their effectiveness also for several other kinds of complex networks. More details can be found in the work [5], which has been the generalization of [4, 3]. Our algorithm in [3] has been used to compute the diameter of Facebook Network (721.1M vertices, 68.7G edges, and diameter 41) with just 17 BFSes in a popular work ([9], divulged by New York Times on November 22, 2011).

2 Enumerating stories

The problem of enumerating stories was motivated initially by the biological question in [10] related to Metabolic networks, in particular to compound graphs, in which vertices are compounds and there is an arc from a compound x to a compound y if there is a metabolic reaction that consumes x and produces y. A subset $\mathbb{B}$ corresponds to compounds that have been experimentally identified as having a significantly higher or lower production in a given condition (for instance when an organism is exposed to some stress). The aim is then to extract all the interaction dependencies among the compounds in $\mathbb{B}$ which do not create cycles but at the same time involve as many compounds as possible. These may require intermediate steps that concern compounds not in $\mathbb{B}$, but the initial and

final steps must involve only compounds in $\mathbb{B}$. A solution, that is a possible scenario of metabolic dependencies, is called a *(metabolic) story*.

A metabolic story has to capture the relationship between the vertices of interest in a way that allows us to define a flow of matter from a set of sources to a set of target compounds. The need for this hierarchy between the compounds led us to consider acyclic solutions. The maximality condition has been added in order to capture all alternative paths between the sources and the targets. The problem is then to "tell" all possible stories given as input a graph G and a subset $\mathbb{B}$ of the vertices of G.

We have presented a polynomial algorithm to find one story and an exact but exponential approach for the enumeration problem [1]. This definition is a generalization of a well-known problem which is the *feedback arc set* problem. However, any polynomial-delay algorithm to enumerate feedback arc sets (ex: [14]) can only be used in some particular instances. Moreover we have shown that finding a story with a specified set of sources or targets is NP-hard.

Our contribution appeared in [1] and its biological application in [11].

3 Enumerating cycles or paths

Studying paths or cycles of biological networks can be useful for several purposes. In the case of interaction graphs, such as Gene Regulatory networks, the importance of enumeration has been shown in [7]. These networks are directed, their vertices are genes, and their arcs are signed, where the sign or weight of the arcs indicates the causal relationship between the vertices, such as activation or inhibition. In particular cycles and paths can be useful for studying dependencies among vertices, the steady state and multistationarity of dynamic models.

We have considered the problem of enumerating paths and cycles in the case of undirected graphs. This result can be useful for undirected Protein-Protein Interaction networks, where nodes are proteins and edges are interactions, but in the case of interaction networks in general, our approach neglects the effects of the controls, i.e. the sign and direction of the arcs. In this latter case, the cycles can be enumerated in the underlying undirected graph and *a posteriori* filtered or *ad hoc* algorithms can be applied. The main question arising from our work, is whether it is possible to extend our result to directed graphs in order to efficiently deal also with this kind of networks.

On the other hand, our contribution is not just restricted to biological undirected networks, but extends also to arbitrary undirected graphs. Listing all the paths and cycles in a graph is a classical problem whose efficient solutions date back to the early 70s. The best known solution in the literature is given by Johnson's algorithm [6] and takes $O((|\mathcal{C}(G)| + 1)(|E| + |V|))$ and $O((|\mathcal{P}_{st}(G)| + 1)(|E| + |V|))$ time for a graph $G = (V, E)$, where $\mathcal{C}(G)$ and $\mathcal{P}_{st}(G)$ denote respectively the set of cycles and (s, t)-paths in G. However there exists graphs for which this algorithm is not optimal.

We have presented the first optimal algorithm to list all the paths and cycles in an undirected graph G. Our algorithm requires $O(|E| + \sum_{c \in \mathcal{C}(G)} |c|)$ time

and is asymptotically optimal: indeed, $\Omega(|E|)$ time is necessarily required to read G as input, and $\Omega(\sum_{c\in\mathcal{C}(G)}|c|)$ time is necessarily required to list the output. Moreover, our algorithm lists all the (s,t)-paths in G optimally in $O(|E|+\sum_{\pi\in\mathcal{P}_{st}(G)}|\pi|)$ time, observing that $\Omega(\sum_{\pi\in\mathcal{P}_{st}(G)}|\pi|)$ time is necessarily required to list the output.

Our algorithm exploits the decomposition of the graph into biconnected components and without loss of generality restricts to study paths and cycles in a same biconnected component. Thus it recursively lists the cycles or (s,t)-paths using the classical binary partition: given an edge e in G, list all the solutions containing e, and then all the solutions not containing e, at each time modifying the graph. In order to avoid recursive calls (in the binary partition) that do not list solutions, we have used a *certificate*, as a data structure, whose cost for dynamically updating is constant with respect to the number of solutions produced. In order to prove the complexity obtained, we have exploited the properties of the binary recursion tree corresponding to the binary partition. For more details, see [2].

4 Enumerating bubbles

A DNA fragment, that is an RNA-coding sequence, is transformed in a Pre-mRNA sequence, through the transcription phase, in which sequences of *exons* and sequences of *introns* alternatively occur. The removal of all the sequences of introns and of some sequences of exons leads to the mRNA sequence, that is a protein-coding sequence, that translated leads to a protein. Since not any exon is transcribed in the mRNA sequence, there can be many possible mRNA sequences. For instance, let $\langle e_1, i_1, e_2, i_2, e_3, i_3, e_4, i_4\rangle$ be a fragment of DNA, where for any j, with $1 \leq j \leq 3$, e_j and i_j are the j-th sequence of exons and introns respectively. The possible resulting mRNA sequences containing e_1 are $\langle e_1, e_2, e_3, e_4\rangle$, $\langle e_1, e_2, e_3\rangle$, $\langle e_1, e_2, e_4\rangle$, $\langle e_1, e_3, e_4\rangle$, $\langle e_1, e_2\rangle$, $\langle e_1, e_3\rangle$, $\langle e_1, e_4\rangle$. The underlying phenomenon is called alternative splicing and checking all the alternative events has been shown in [13] to correspond to checking recognisable patterns in a de Bruijn graph built from the reads provided by a sequencing project. The pattern corresponds to an (s,t)-bubble: an (s,t)-bubble is a pair of vertex-disjoint (s,t)-paths that only shares s and t.

Since the k-mers correspond to all words of length k present in the reads (strings) of the input dataset, and only those, in relation to the classical de Bruijn graph for all possible words of size k, the de Bruijn graph for NGS data may then not be complete. We have ignored all the details related to the treatment of NGS data using De Bruijn graphs, and consider instead the more general case of finding all (s,t)-bubbles in an arbitrary directed graph. In particular we show the first linear delay algorithm to identify all bubbles. A previous known algorithm presented in [13] was an adaptation of Tiernan's algorithm for cycle enumeration [15] which does not have a polynomial delay. In the worst case the time elapsed between the output of two solutions is proportional to the number of paths in the graph, i.e. exponential in the size of the graph. Our algorithm

is a non trivial adaptation of Johnson's cycle enumeration algorithm [6] in a directed graph with the same theoretical complexity. Notably, the method we propose enumerates all bubbles with a given source with $O(|V|+|E|)$ delay. The algorithm requires an initial transformation of the graph, for each source s, that takes $O(|V|+|E|)$ time and space; this transformation reduces the enumeration of bubbles to the enumeration of constrained cycles in a special graph.

References

1. V. Acuña, E. Birmelé, L. Cottret, P. Crescenzi, F. Jourdan, V. Lacroix, A. Marchetti-Spaccamela, A. Marino, P. V. Milreu, M.-F. Sagot, and L. Stougie. Telling stories: Enumerating maximal directed acyclic graphs with a constrained set of sources and targets. *Theor. Comput. Sci.*, 457:1–9, 2012.
2. E. Birmelé, R. A. Ferreira, R. Grossi, A. Marino, N. Pisanti, R. Rizzi, and G. Sacomoto. Optimal listing of cycles and st-paths in undirected graphs. In *SODA*, pages 1884–1896, 2013.
3. P. Crescenzi, R. Grossi, M. Habib, L. Lanzi, and A. Marino. On computing the diameter of real-world undirected graphs. *Theor. Comput. Sci.*, 514:84–95, 2013.
4. P. Crescenzi, R. Grossi, C. Imbrenda, L. Lanzi, and A. Marino. Finding the diameter in real-world graphs - experimentally turning a lower bound into an upper bound. In *ESA (1)*, pages 302–313, 2010.
5. P. Crescenzi, R. Grossi, L. Lanzi, and A. Marino. On computing the diameter of real-world directed (weighted) graphs. In *SEA*, pages 99–110, 2012.
6. D.B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.*, 4(1):77–84, 1975.
7. S. Klamt and A. von Kamp. Computing paths and cycles in biological interaction graphs. *BMC Bioinformatics*, 10:181, 2009.
8. C. Klein, A. Marino, M.-F. Sagot, P.V. Milreu, and M. Brilli. Structural and dynamical analysis of biological networks. *Briefings in functional genomics*, 2012.
9. B. Lars, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four degrees of separation. In *WebSci*, pages 33–42, 2012.
10. G. Madalinski, E. Godat, S. Alves, D. Lesage, E. Genin, P. Levi, J. Labarre, J.-C. Tabet, E. Ezan, and C. Junot. Direct introduction of biological samples into a ltq-orbitrap hybrid mass spectrometer as a tool for fast metabolome analysis. *Analytical Chemistry*, 80(9):3291–3303, 2008.
11. P. V. Milreu, C. Klein, L. Cottret, V. Acuña, E. Birmelé, M. Borassi, C. Junot, A. Marchetti-Spaccamela, A. Marino, L. Stougie, F. Jourdan, P. Crescenzi, V. Lacroix, and M.-F. Sagot. Telling metabolic stories to explore metabolomics data: a case study on the yeast response to cadmium exposure. *Bioinformatics*, 30(1):61–70, 2014.
12. M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.
13. G. Sacomoto, J. Kielbassa, R. Chikhi, R. Uricaru, P. Antoniou, M.-F. Sagot, P. Peterlongo, and V. Lacroix. Kissplice: de-novo calling alternative splicing events from rna-seq data. *BMC Bioinformatics*, 13(S-6):S5, 2012.
14. B. Schwikowski and E. Speckenmeyer. On enumerating all minimal solutions of feedback problems. *Discrete Applied Mathematics*, 117(1-3):253 – 265, 2002.
15. J. C. Tiernan. An efficient search algorithm to find the elementary circuits of a graph. *Communonications ACM*, 13:722–726, 1970.