Multi-species Protein Function Prediction: Towards Web-based Visual Analytics

[Position Paper]

Paolo Perlasca, Giorgio Valentini, Marco Frasca, Marco Mesiti
Dipartimento di Informatica – Università di Milano
Via Comelico, 39 – Milano, Italy
{mesiti,valentini,perlasca,frasca}@di.unimi.it

ABSTRACT

The visualization and analysis of big bio-molecular networks is a key feature for the investigation and prediction of protein functions in a multi-species context. In this paper we present the design of a system that integrates data management, machine learning and visualization facilities to make effective the visual analysis of big networks by means of web-based interfaces.

Categories and Subject Descriptors

H.2.5 [**Heterogeneous Databases**]; C.2.4 [**Distributed Systems**]: Distributed applications; I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*learning*

General Terms

Algorithms

Keywords

Protein Function Prediction, Heterogeneous networks, graph visualization, visual analytics

1. INTRODUCTION

Many efforts have been devoted in the last decade to develop automated tools for large scale network-based automated function prediction of proteins (AFP) [7, 8]. These tools usually represent the available datasets through undirected graphs $G = (V, E)$, where nodes $v \in V$ correspond to proteins, and edges $e \in E$ are weighted according to the evidence of co-functionality implied by data sources [29]. The AFP problem consists in the identification of the class A of unlabeled proteins relying on their (direct or indirect) co-functionality with proteins known to belong to class A. A recent international challenge for the critical assessment of automated function prediction [11], highlighted that scalability and heterogeneity of the available data, and interpretation of the obtained results represent three of the main challenges posed by AFP.

The number of nodes and edges of the graphs that need to be processed can be very big. This requires the implementation of AFP algorithms that exploit: *a*) the memorization of the graphs in secondary memory (like GraphChi [14], Neo4J [31]) and the partial loading in main memory of the parts of the graphs needed for the computation; or, *b*) the use of NoSQL technologies (like GraphLab [17], Spark [12]) in which the map-reduce paradigm is used for the computation in a cluster of machines. Despite the use of these implementation strategies, the complexity of the available approaches poses serious obstacles for their scalability.

Datasets are usually made available from different public web sites, such as Entrez Gene, Ensembl, and UniProt, containing the annotations of the available organisms that exploit different conventions for identifying the proteins and their interactions. Specifically, they exploit different IDs for the presentation of the same molecule, most times without providing a matching with those of the other web sites. The continuously changed structure, organization and content of the data sources make the integration of their datasets particularly difficult. Many efforts are nowadays available to face their heterogeneity [22, 27] both from the perspective of the organization of the data and the perspective of their preparation for the application of machine learning algorithms to infer new biological knowledge from the data. Once these issues are faced, however, other problems may raise from a learning perspective. For instance, the coverage of functional annotations is different in different organisms and this make very difficult the effective transfer of the available functional knowledge from one organism to another.

A last, but not least, issue is the possibility to visually interpret the results of the AFP algorithms by means of visual tools. The size of the networks containing the prediction can be really big and their simple visualization, with off-the-shelf graphical visualization tools (like GraphViz, GeneMania, Neo4J) produces a cloud of points that is really hard to interpret and difficult to handle within a browser because of the consumption/saturation of the main memory. There is therefore the need of approximate views of the graph that allow an aggregate visualization of the graph and this representation should be navigable. This corresponds also to the intrinsic structure of protein networks, where multiple levels of structural and functional organization require multiple views, at different resolution levels, of the network itself. The user can decide the part of the graph that wishes

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
iiWAS 2016, Singapore
Copyright 2016 ACM 978-1-4503-2113-6/13/12 ...$15.00.

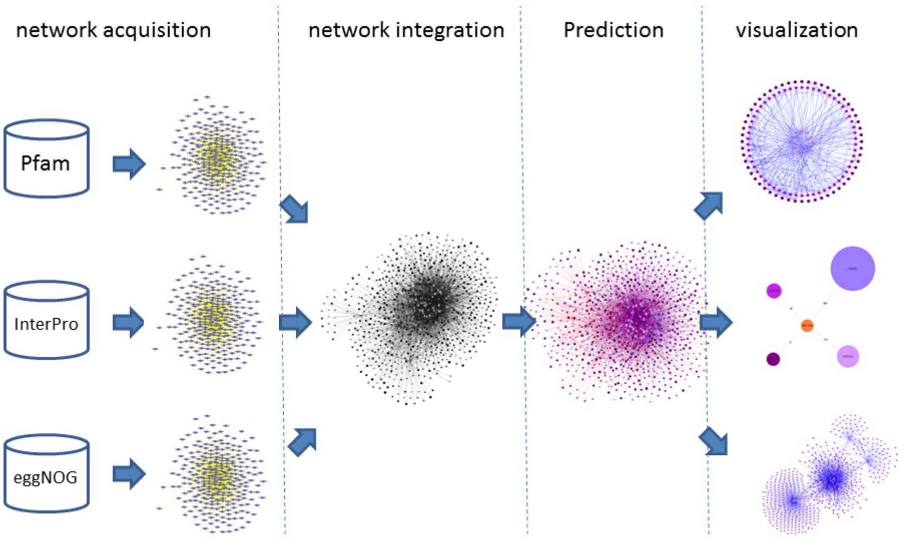


Figure 1: Steps required for the visual analysis of the AFP results

to explore and by clicking on the corresponding part obtaining a new enhanced visualization. Even if many approaches currently under development [3] move to this direction, none of them is specifically tailored for the AFP problem.

In this paper we propose a comprehensive Web-based solution that can be adopted to effectively tackle the scalability, heterogeneity, and interpretation challenges posed by AFP and so far discussed. The proposed solution combines data management, machine learning and visualization approaches for obtaining a system that can be used by domain experts for the acquisition and preparation of the component networks from available data banks, their integration and normalization in a multi-species network, the efficient application of AFP algorithms that can provide predictions in a reasonable amount of time, and the visual analysis of the results by means of different views calculated on the result of the AFP algorithms. The proposed solution has been sketched in Figure 1 and each component will be described in the remainder of the paper by pointing out their relevance w.r.t. the state of the art. Specifically, in Section 2 we advocate the use of the MergeGraphs system [21] for the preparation of the data to be integrated. By means of its Web-based interface it is possible to design the acquisition and integration processes and to prepare the datasets for the application of the AFP algorithms. Section 3 discusses the problem of the efficient design and implementation of network-based algorithms for AFP, introducing the main algorithmic and technological issues underlying the multi-species protein function prediction problem. Finally, Section 4 deals with the requirements and approaches for the visual analysis of AFP results. Our position is that a system that includes the aforementioned characteristics could be profitable exploited by biologists and bio-informaticians to investigate the functional role of the proteins.

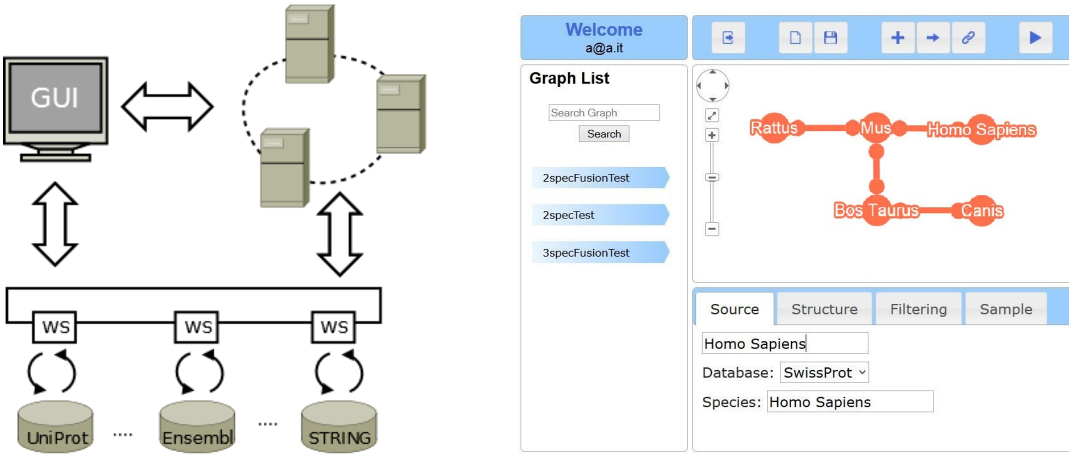
2. NETWORK ACQUISITION AND INTEGRATION

The construction of the networks that need to be integrated requires to download from the available public databases of the functional annotations of the considered organisms and the application of machine learning approaches for identifying the level of co-functionalities among the different proteins. This task poses several challenges due to the high frequency at which the public databases evolve both in terms of content and structures.

To address this issue we wish to adopt the recently proposed *MergeGraphs* system [21]. MergeGraphs is a web-based system for merging heterogeneous graphs that leverages the peculiarities of the single data sources in order to offer to the user a simple, yet powerful, environment for the specification and application of aggregation and integration operations to collect graph sources of different types. Single data sources are accessed by means of REST interfaces that expose services organized around the concepts of *vertices* and *edges* of the network. In this way, it is possible to obtain structures, sampling, and data of vertices and edges contained in the source independently from the format in which they are stored. Moreover, different operations can be applied on property graphs for extracting graph portions from data sources, for cleaning, integrating, and merging graphs and for solving mismatch among the adopted identifiers. Finally, MergeGraphs offers a Web interface where users can drag and drop different data sources and visually apply on them a set of operations that will lead to the specification and application of the execution plan.

In the context of the AFP problem, this system could be particularly helpful because web services can be developed for accessing single public databases by providing a common interface that can be easily customized to new requirements by changing configuration parameters expressed by means of JSON files. Moreover, the system could be integrated with new operations specifically tailored for extracting and integrating annotations from the public databases.

Figure 2 provides the general architecture of the MergeGraphs system that exploits a cluster of machines for making scalable the acquisition and integration of the required networks. Moreover, the right part of Figure 2 shows an execution plan for the acquisition of proteins of different organisms from UniProt and for the construction of their interspecies functional relationships from the Ensembl database.



Homo Sapiens

Database: SwissProt

Species: Homo Sapiens

Figure 2: Architecture and GUI of the MergeGraphs system

3. EFFICIENT AFP ALGORITHMS

The large scale network-based prediction of multi-species protein function poses significant algorithmic and technological challenges, due to the scalability of existing algorithms and the limitations of available memory in off-the-shelf desktop computers. Indeed most state-of-the-art network-based algorithms for AFP show a quadratic or cubic computational complexity [29], thus making infeasible their practical application to large multi-species networks. These approaches, indeed, usually rely on an in-memory adjacency matrix representation of the graph network, scale poorly with the size of the graph [15], and may have time complexity that becomes quickly prohibitive. Performance optimization is usually realized by adopting an adjacency-list representation of the graph to take its sparsity into account, or by using parallel strategies for matrix multiplication [4].

However problems arise when we cannot maintain entirely in primary memory the graph representing the biological network. In this case we can use two different technological strategies. The first one relies on distributed computation [9, 16, 18], by which the graph is spread on different machines and the results are finally collected. As a second strategy we can adopt secondary memory-based computation [31, 10], by efficiently using the secondary memory through appropriate techniques and engines well-suited for graph-based computation, such as *GraphChi*, a disk-based system that requires only a limited amount of primary memory to contain the edges and their associated values of only a relatively small subset of vertices at a time, while the rest of the graph is efficiently stored on disk [14].

Both these technological approaches require a rethinking of existing network-based algorithms, since most of them rely on the fact that the graph representing the protein network is entirely stored in main memory of a single computer. Unfortunately this in not true both when the graph is spread across different computers in a distributed environment, and when only a very small part of the graph can be available in primary memory and the secondary memory based computation paradigm is adopted.

A solution to this problem is the "local" implementation of existing network-based algorithms. That is, a "vertex-centric" (or symmetrically an "edge-centric") programming model [18] is adopted, by which only one vertex and its incident edges are iteratively processed at a time. It is worth noting that the most effective network-based algorithms for AFP learn by exploiting the overall topology of the networks [24, 13, 28], and their implementation usually requires to process in primary memory a large part or the overall underlying graph. The local "vertex-centric" implementation, by working iteratively across all the network, do not reject to think "globally" by exploiting the overall topology of the network, but "decomposes" global learning algorithms through local learning strategies that involve at each iteration a small local part of the graph [16]. Of course this is not always feasible for any network-based algorithms, but we showed that it is easy to obtain a vertex centric implementation for the popular random-walk algorithm [23]. The local implementation strategy can be applied to other network-based learning algorithms, ranging e.g. from simple guilt-by-association methods (that are inherently local) [19, 20] to more complex label propagation methods [32, 2], the recently proposed parametrized Hopfield networks [5, 6] and kernelized score functions [30].

4. AFP VISUAL ANALYTICS

The result of the application of the AFP algorithms discussed in the previous section is a network in which the proteins are annotated with their predicted classes and with the confidence of their membership. In a multi-species framework, the obtained networks can be quite big and their straightforward visualization in a browser poses issues from different points of view. The interpretation of the result is hard to grasp because a cloud of points is shown in the canvas and current visualization tools are not able to easily points out the occurrence of clusters of points. Moreover, the size of the network requires long time for its transfer from the server to the client and for the rendering in the canvas. Finally, the main memory of the machine where the browser is running is used for maintaining the network and the interaction with the browser can be really slow. Different approaches are thus required for improving the visualization of the network and the user experience with the tool. Several topology and energy-based approaches have been proposed for the visualization of both small and big graphs [3] . In our work we wish to adopt these standard approaches to provide different views of the data that allow to better grasp the re-

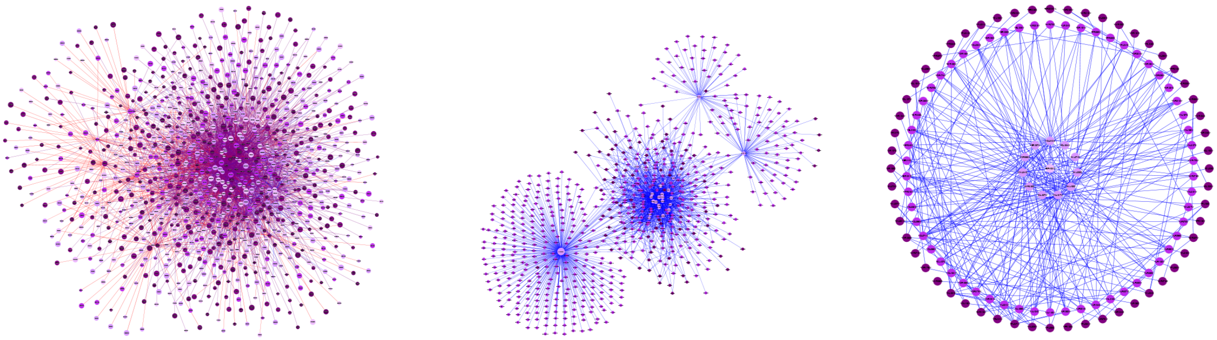


Figure 3: a) result of the prediction, b) vertex centered visualization, c) circle vertex centered visualization

sult of the AFP approach and to allow the user to navigate in a multi-resolution representation of the graph [26, 25], in order to access to the portions that are more significant for his own analysis.

In order to reach this goal, we adopt a “vertex-centric” approach for the exploration of the network that could be exact or approximate. With “vertex-centric” visualization approach we mean that the user can specify a vertex, named *target*, from which he wishes to explore the result of the prediction, and the size of the subnetwork he wishes to extract (in terms of nodes whose distance from the target is lower than a given threshold). Starting from these parameters, the system can produce the subgraph centered in the target. When this subgraph is small, a standard clustering algorithm can be used for aggregating together similar vertices (according to a given similarity measure) as shown in Figure 3.b. In this case three clusters are obtained from a subgraphs with threshold 3. Moreover, we can adopt a circle vertex centered visualization in which the vertices at distance one, two and three from the target are drawn in different concentric circles as shown in Figure 3.c. This rendering allows one to better understand the connectivity of the target with its neighborhood and how the co-functionality confidence degree has been propagated from the proteins already known belonging to a given class to those that have been predicted to belong to that class.

The aforementioned rendering techniques can be adopted when the size of the generated subgraph is still small and the available canvas is big enough for its visualization. When this is not possible, approximate visualization techniques can be adopted. For example the target node can be connected with bubbles of different sizes that distribute the nodes of the subgraph relying on the weight of their edges and the distance from the target. For example, in the left part of Figure 4 a target is shown in the center and then four bubbles are drawn representing the vertices at distance one that can be reached with a weight between $[0.0, 0.25)$, $[0.25, 0.50)$, $[0.50, 0.75)$, and , $[0.75, 1]$. This representation points out how much the co-functionality confidence degree has been propagated (or not) from the target node. Moreover, by clicking on one of the bubbles, the visualization can be expanded by showing other bubbles or single nodes (when the bubble only contains a singleton). Finally, the user can ask to show the bubbles representing nodes that can be reached with two steps from the target node.

Producing the exact and approximate rendering of the graph requires to apply algorithms for the computation of the clustering and aggregation of the data that need to be executed quickly on the server side. For this purpose noSQL solutions can be adopted in order to obtain efficient and scalable approaches.

5. CONCLUSIONS

The investigation of the functions of proteins in a multi-species framework requires the integration of data management, machine learning, visualization and web-based tools to allow biologists an interactive inspection of complex protein networks. In this way biologists can dispose of an easy-to-use interactive environment to discover novel functions of proteins with the support of underlying efficient and accurate machine learning methods for their functional characterization. We individuated MergeGraphs as a flexible web-based system that enables the specification and application of operations to combine different graph-based sources of information and for the visual application of operations to process biomolecular networks. This system, empowered with state-of-the-art network-based scalable graph-processing algorithms for AFP, makes feasible visual analytics procedures that can support the research activity of biologists, also through the proper multi-level visualization of complex biomolecular networks, constructed from data downloaded from public repositories. The same overall framework could be in principle adapted to other contexts characterized by the integration of different network-based source of information, such as the prioritization of disease genes or the prediction of abnormal phenotype - gene associations, and other relevant applications in the context of the so called“Network Medicine” [1].

6. REFERENCES

[1] A. Barabasi, N. Gulbahce, and J. Loscalzo. Network medicine: a network-based approach to human disease. *Nature Rev. Genet.*, 12:56–68, 2011.

[2] Y. Bengio, O. Delalleau, and N. Le Roux. Label Propagation and Quadratic Criterion. *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006.

[3] P. Eades and K. Klein. Graph visualization, EDBT School, 2015.

[4] J. Foster. *Designing and Building Parallel Programs.* Addison Wesley, 1995.

[5] M. Frasca, A. Bertoni, M. Re, and G. Valentini. A neural network algorithm for semi-supervised node label learning from unbalanced data. *Neural Networks*, 43:84–98, 2013.

[6] M. Frasca, A. Bertoni, and G. Valentini. Unipred: Unbalance-aware network integration and prediction of

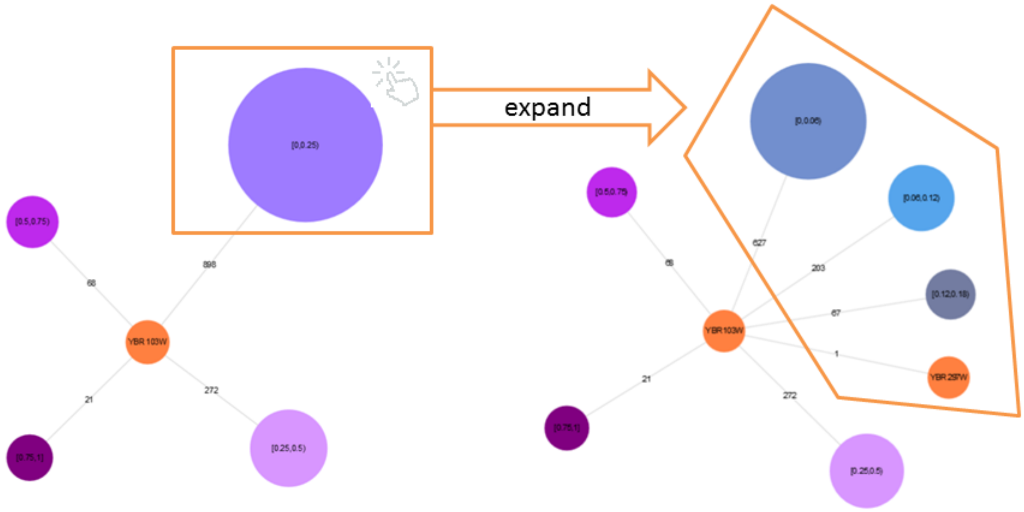


Figure 4: a) vertex centered approximate visualization, b) result of the expansion of a bubble

protein functions. *J. of Computational Biology*, 22(12):1057–1074, 2015.

[7] I. Friedberg. Automated protein function prediction-the genomic challenge. *Brief. Bioinformatics*, 7:225–242, 2006.

[8] J. Gillis and P. Pavlidis. Characterizing the state of the art in the computational assignment of gene function: lessons from the first critical assessment of functional annotation (CAFA). *BMC Bioinformatics*, 14((Suppl 3)):S15, 2013.

[9] J. Gonzalez, et al. PowerGraph: Distributed graph-parallel computation on natural graphs. In *OSDI'12*, pages 17–30, 2012.

[10] W.-S. Han, et al. Turbograph: A fast parallel graph engine handling billion-scale graphs in a single PC. In *Proc. of ACM SIGKDD*, pages 77–85, 2013.

[11] Y. Jiang et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, 17(184), 2016.

[12] H. Karau, et al. *Learning Spark: Lightning-Fast Big Data Analysis.* O'Reilly Media, 2015.

[13] S. Kohler, et al. Walking the interactome for prioritization of candidate disease genes. *Am. J. Human Genetics*, 82(4):948–958, 2008.

[14] A. Kyrola, G. Blelloch, and C. Guestrin. Graphchi: large-scale graph computation on just a pc. In OSDI'12, pages 31–46, 2012.

[15] W. Liu, J. Wang, and S.-F. Chang. Robust and scalable graph-based semisupervised learning. *Proc. of IEEE*, 100(9):2624–2638, 2012.

[16] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. Graphlab: A new parallel framework for machine learning. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[17] Y. Low, et al. Distributed GraphLab: A framework for machine learning in the cloud. *PVLDB*, 5(8):716–727, 2012.

[18] G. Malewicz, et al. Pregel: A system for large-scale graph processing. In *Proc. of SIGMOD*, pages 135–146, 2010.

[19] E. Marcotte, et al. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402:83–86, 1999.

[20] R. McDermott, J. Bumgarner and R. Samudrala. Functional annotation from predicted protein interaction networks. *Bioinformatics*, 21(15):3217–3226, 2005.

[21] M. Mesiti. Mergegraphs: a web-based system for merging heterogeneous big graphs. In *Proc. of Int'l Conf. on Information Integration and Web-based Applications & Services, iiWAS,* , pages 1–10, 2015.

[22] M. Mesiti, et al. XML-based approaches for the integration of heterogeneous bio-molecular data. *BMC Bioinformatics*, 10(12):S7, 2009.

[23] M. Mesiti, M. Re, and G. Valentini. Think globally and solve locally: secondary memory-based network learning for automated multi-species function prediction. *GigaScience*, 3:5, 2014.

[24] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9(S4), 2008.

[25] C. W. Muelder, et al. Multivariate social network visual analytics. *Multivariate Network Visualization. LNCS*, 8380:39 – 46, 2014.

[26] T. Munzner. Interactive visualization of large graphs and networks. Stanford, CA, USA, 2000.

[27] G. Valentini et al. An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods. *Artificial Intelligence in Medicine*, 61(2):63–78, 2014.

[28] M. Re, M. Mesiti, and G. Valentini. A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, 9(6):1812–1818, 2012.

[29] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Mol. Sys. Biol.*, 8(88), 2007.

[30] G. Valentini, G. Armano, M. Frasca, J. Lin, M. Mesiti, and M. Re. RANKS: a flexible tool for node label ranking and classification in biological networks. *Bioinformatics*, 32:2872–2874, 2016.

[31] J. Webber. A programmatic introduction to neo4j. In *Proc. of onf. on Systems, programming, and applications: software for humanity*, pages 217–218, 2012.

[32] D. Zhou et al. Learning with local and global consistency. In *Adv. Neural Inf. Process. Syst.*, 16:321–328, 2004.