A cloud-based approach for Gene Regulatory Networks dynamics simulations

A Cloud-Based Approach for Gene Regulatory Networks Dynamics Simulations

Alessandro Vasciaveo, Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino
Politecnico di Torino
Dipartimento di Automatica e Informatica
Corso Duca degli Abruzzi 24, I-10129, Torino, Italy
Email: {name.familyname}@polito.it

Fabrizio Bertone, Giuseppe Caragnano, Olivier Terzo
Istituto Superiore Mario Boella
Via P. C. Boggio 61, Torino, Italy
Email: {familyname}@ismb.it

Abstract—**Gene Regulatory Networks (GRNs) are one of the most investigated biological networks in Systems Biology because their work involves all living activities in the cell. A powerful but simple model of such GRNs are Boolean Networks (BN) that describe interactions among biological compounds in a qualitative manner. One of the most interesting outcomes about GRNs's dynamics are the so called network attractors, since they seem to well represent the stable states of a living cell. Though collecting state space trajectories is a quite simple task when the network topology consists of few nodes, it becomes not so trivial when nodes are of the size of hundreds or thousands. Thus, we exploit the MapReduce algorithm in order to cope this complexity on a cloud architecture built for the purpose. We found that scaling-out the problem is a better solution rather than increasing resources on single machine, thus allowing simulations of large networks.**

Keywords-component; Boolean Networks; Cloud Computing; Gene Regulatory Networks; MapReduce Algorithm; Network Attractors; Network Dynamics Simulation; Systems Biology; Computational Biology; Big Data

I. INTRODUCTION

One of the most important challenges today is that the latent knowledge embedded in biological networks is not currently fully exploited [1]. Since networks are useful to model complex systems, they are heavily adopted in many research areas, and they find applications in systems biology too. More specifically, interactions among biological compounds are mapped into a particular kind of networks called biological pathways (or simply pathways), which acts as a knowledge representation about biological phenomena involved in cell metabolism, cell signaling or gene regulation in living cells. By modeling the last phenomenon, pathways are usually called Gene Regulatory Networks (GRNs) and they are the subject of our work. Furthermore, scientists discovered that genes never act alone in a biological system, but they participate in a cascade of networks [2] increasing the complexity of the system to be analyzed.

Since computational models are well suited to cope with these complex systems, many of them have been proposed since '70s to properly model GRNs and their dynamics [11-14]. Among them, we chose a qualitative modeling approach by adopting a Boolean Network model similar to the one introduced by Kauffman as reviewed by Gershenson in [14]. Major differences between the Random Boolean Network model described in [14] and our model consist in a deterministic selection of the node updating function (instead of a random selection from a predefined set of them) and the inclusion of gene/product and post-transcriptional information in the network topology [28]. A common approach adopted by most of the network dynamics simulators is to assume a synchronism while updating the state of each node in the network. Our implementation of the model follows this assumption, thus the network dynamics are those generated by a Finite State Machine (FSM) and trajectories in the network state space are composed of a finite set of nodes univocally determined by the network attractor, which is the termination node. This property of a deterministic state space is the key to understand why using the MapReduce paradigm is an interesting and valid approach to analyze the dynamics of large GRNs modeled by Boolean Networks. Details about implementation will follow in next sections.

There are a few tools in literature that allow the simulation of Boolean Networks as those proposed in [7-10], but all of them share common drawbacks like limitations in network size, i.e., tens or few hundreds of nodes, and a limitation in the degree connectivity per node because of their implementation that uses truth tables for representing updating functions in memory. For example, when modeling post-transcriptional regulation, a node could have many incoming connections due the presence of hub nodes like MicroRNAs or Transcription Factors. This leads to model node updating functions with truth tables that need huge memory resources (e.g., truth table for a network with a node v targeted by other 50 nodes would need 2^{50} rows, just for v, to be stored in central memory). Therefore, to allow the simulation of large networks we need tools that are able to handle this bottleneck.

Few of these tools let the user to analyze the network state space of the whole simulation. We adopted as Boolean Network Simulator the Enhanced Boolean Network Toolkit

Publication funding for this article has come from Grant No. CUP B15G13000010006 awarded by the Regione Valle d'Aosta for the project: "Open Health Care Network Analysis"

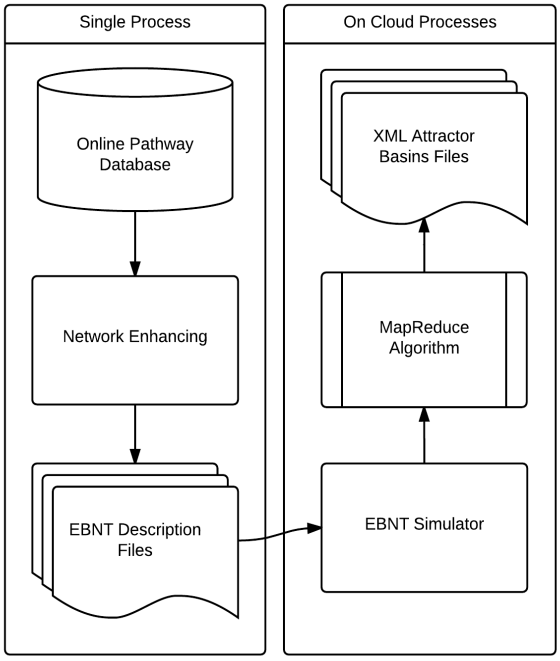

Figure 1 - The overall data flow from network topology to XML attractors set.

(EBNT Simulator), which is the one available at [36] because it is implemented in C++, it runs fast, it uses multithreaded libraries, and it is an open source solution.

II. Objectives and motivation

The whole human genome consists of more than 30,000 genes [3] and a single GRN model is not able to take into account all interactions among all genes. This is due to limitation of our knowledge about a general and complex GRN model and to the huge computational resources needed to analyze such a model. Moreover, a single pathway is a model of just a small subset of these genes for which the interconnecting mechanisms are well known in the scientific community. Nevertheless, the increasingly knowledge about a specific GRN allows scientists to infer links between related GRNs, providing cascades of these networks formin bigger GRNs. Furthermore, there are more and more tools that provide a network enhancing based from the retrieval of information from many online databases. Thus, there is an increasing requirement for tools able to simulate very large GRNs.

The aim of our study is to allow the simulation of GRNs modeled by large Boolean Networks, which consist of thousands of nodes highly interconnected. This goal is reached through a cloud architecture based upon the well-know MapReduce algorithm. The final outcome consists of a set of XML files containing a subset of basins of attraction of the simulated GRN and their network attractors. These XML files can be imported in Cytoscape [29] to be further analyzed.

III. Materials and methods

The data flow process, as shown in Fig. 1, starts with the selection of a pathway from online databases. These pathways are manually curated and a large collection is classified and available online from many repositories [5-6]. A useful tool which is able to download a biological pathway, enhance it, and provide as output a boolean network topology representation is a Cytoscape plugin named ReNE [30]. The second step is the network enhancing. Although, it is an optional step and can be skipped if just a classical simulation is desired without post-transcriptional elements modeled. After that, the files containing the model representation of the GRN are ready to be shipped as input for the EBNT Simulator. Until now, the process is performed on a single machine by a single user.

The general architecture on which the workflow in Fig.1 is based is composed of the following components.

A. Enhanced Boolean Network Toolkit

The Boolean Network model is implemented in a software toolkit (EBNT) that allows to analyze GRNs from both a structural and a dynamic point of view. Among its capabilities, we used the EBNT to simulate the network dynamics in order to find those stable states of a GRN called attractors. This process is well described in [31,37]. The open-source toolkit is compatible with available visualization tools like Cytoscape and allows to run detailed analysis of the network topology as well as of its attractors, trajectories, and state-space.

B. Cloud Computing Integration

Since the network analysis algorithm requires considerable resources in terms of temporary memory and permanent storage, an effort was made in order to enhance the computational power and the storage capabilities of the system. Moreover, the massive amount of data generated by the EBNT Simulator limits the scope of information available for even computationally sophisticated users.

To overcome these limitations in simulation and analysis stages, an experimental virtualized cloud computing environment was implemented in the form of a *Platform as a Service* (**PaaS**).

The problem of efficiently dealing with extremely large quantity of data is recurring in modern industry and a variety of solutions and infrastructures have been proposed. Both storage and computation are aspects that need to be taken into account and optimized in order to serve an efficient Big Data analysis system. MapReduce, proposed by Google [25], has soon emerged as a leading paradigm for Big Data processing due to its scalability and reliability [32]. Hadoop [33] is an open-source implementation, which is reminiscent of GFS (Google File System) and MapReduce [35], and is released under the umbrella of the Apache Software Foundation [34].

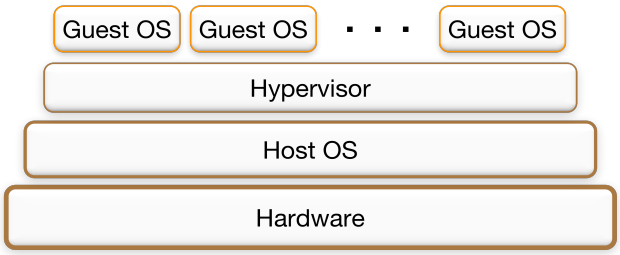

Figure 2 - The virtualization stack.

MapReduce algorithms have been widely used in bioinformatics applications such Next Generation DNA Sequencing analysis [15-16], exploiting the computational capabilities of the cloud architecture, and in storage and retrieval of large biomedical data as in [17-18]. An interesting review on bioinformatics application using MapReduce and Hadoop is done by Taylor in [22]. Moreover, the MapReduce algorithm performs better when data are homogeneous, and this is the data type that we designed to work with.

C. Cloud Environment

The base of the distributed system was built using the Apache Hadoop framework [19] deployed inside a Cloud constituted by Ubuntu Server Virtual Machines [20].

The Cloud environment was built using open source virtualization tools on top of IBM BladeServer hardware. The complete virtualization stack depicted in Fig. 2 is composed of several layers which consists in, from bottom to top:

- Hardware (IBM Blade)
- Host OS (Cent OS)
- Hypervisor (KVM)
- Guest OS (Ubuntu Server 14.04)

The Cloud Management is performed through OpenNebula [21]. We used the Hadoop framework version 2.0 deployed inside Guest OS VMs. This technique permits to dynamically allocate computing clusters of the required size.

Inside the Hadoop cluster, two Virtual Machines (VMs) are dedicated exclusively to do the Master Agent. One machine, the *HDFS Master*, controls the Distributed File System (DFS) and the other, the *YARN Master*, leads computational resources. All the remaining allocated VMs act as *slaves* and provide active storage and Workers at the same time.

In the proposed cluster architecture, YARN master serves as an interface node for scheduling/dispatching purposes. File management, including propagation of EBNT description files and splitting/indexing of big files is held by the HDFS master.

D. MapReduce Algorithm

The generation of the results is divided into various stages and sub-steps, where every step is coordinated between the various machines by a central entities. The workflow, which is going to be described is depicted in Fig. 4.

The goal in the first stage is to generate the inputs that will be supplied to the EBNT Simulator. The second stage uses the data generated in the previous step (initial states) to simulate and track the network evolution down to the final states (attractors) saving the trajectories in the state space. The third stage, the *map* phase in the MapReduce paradigm, parses the results files and assign to each attractor a *key* and the number of hits encountered as its value. The fourth and last stage, the *reduce* phase, identifies multiple instances of the same attractor generated in different simulations and sum up all its hits values.

The details of the various stages of the whole process are shown in Fig. 3 and explained in the following.

A request to start a simulation is initially performed using a custom developed User Interface (a web page which exposes custom Web Services APIs). The data needed in order to start the simulation is inserted in the DFS and made available to all the connected Workers. The total number of simulations to perform is divided among all the available slave VMs in the system, in order to reduce the complexity of the generation stage. For example, if the cluster is composed of 100 Slaves VMs, a request of 10^{10} simulations will be split among the slaves by performing 10^{8} simulations in each VM. Some measurements will be needed to identify the best compromise between minimum subset and framework overhead. The

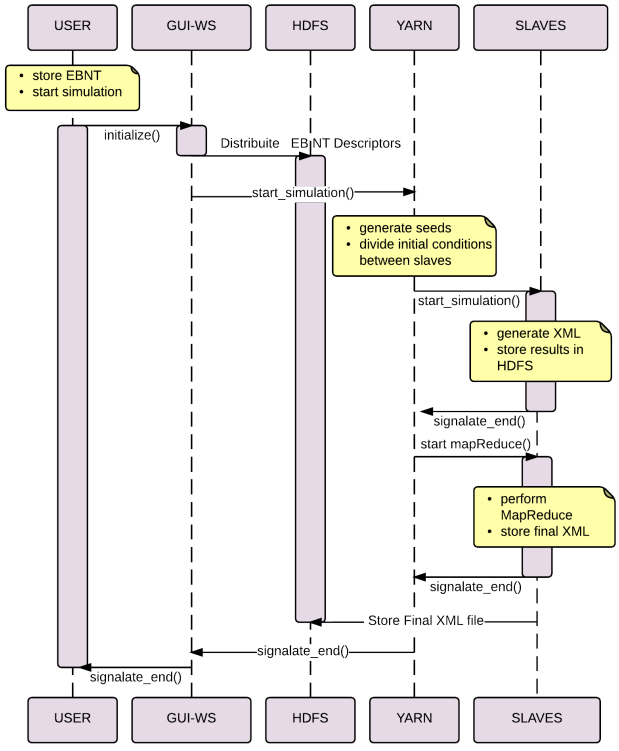

Figure 3 - *UML Sequence Diagram of data flow on the cloud architecture.*

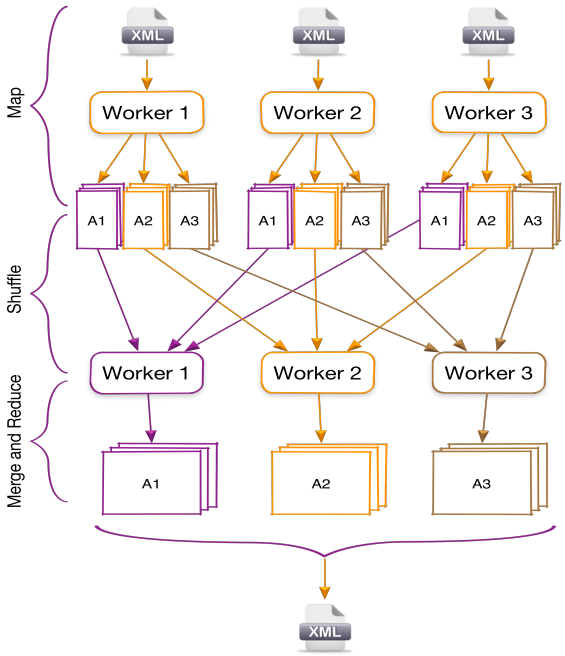

Figure 4 – The MapReduce algorithm applied to simulation results.

YARN Master, which acts as a controller, generates a random seed for each Slave VM. Using the EBNT Simulator and the assigned seed, each machine generates a subset of initial states from which to start the simulation. Managing the seeds generation in a central controller has the purpose to avoid running simulations with the same initial conditions, which could lead to useless identical datasets. The initial states are stored locally on each VM in order to avoid unnecessary network traffic. A custom application developed for the YARN Framework [23] deals with running the necessary commands on each VM setting the right parameters to the EBNT Simulator. Afterwards, each VM is able to perform the simulation running the attractor finder algorithm, which is multi-threaded, collecting all trajectories in the state space.

When a VM completes all the simulations, the output produced is a XML file, which stores the set of network attractors, each with all its basins (the set of trajectories). This file is stored on the HDFS [24] distributed filesystem provided by Hadoop, allowing the availability of the results at a global scope.

Next, it comes the analysis performed on all computed datasets by the MapReduce paradigm, which can be loosely summarized in two phases. In the first one, the *map* phase, files are parsed and mapped in order to obtain sets of key-value pairs. The second phase is called the *reduce* phase and consists in the aggregation of all the key-value pairs on the basis of the key value, while performing a algorithm of interest. In our implementation, the map phase consists in the indexing of all attractors found (*keys*) by assigning, as value for each key, a edge which composes a trajectory. After, the reduce phase will collect keys with same value (i.e. a network attractors) from all simulation instances performed in parallel by the VMs. Then, the basin size is computed counting all edges that belong to the same attractor while taking into account possible duplicates.

We are still collecting results about simulations and till now we ran a test comparing the simulation time of the mTOR pathway, both on a single machine and on the cloud environment. The mTOR pathway is taken from the KEGG Online Database [37] (with the KEGG code hsa04150) and processed with the Cytoscape plugin ReNE [30]. The resulting pathway after the enhanced phase is a GRN, which consists in 1668 nodes (among genes, Transcription Factors and MicroRNAs) and 12603 edges.

Running the simulations for a total of 10^6 initial states, we got the computational time on a single VM and on different clusters each composed of a certain number of allocated VMs as shown in Fig. 5.

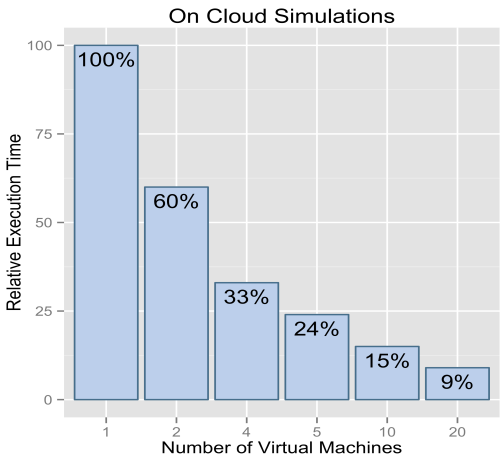

Figure 5 - Simulation timings

From the same Fig. 5 it's clearly noticeable a performance boost of a 10x factor using a cluster of 20 VMs. Indeed, using 20 VMs, the total time is just the 9% of the simulation time performed by a single allocated VM.

IV. Conclusions and Future Works

By using established methods derived from cloud computing we have proposed a fully scalable architecture for the analysis of the dynamics of GRNs modeled by Boolean Networks.

Thus, the study of real networks dynamics with significant size it is now possible also thanks to the many cloud platforms and services that are providing support to the technology we used.

A possible improvement in the proposed cloud architecture consists in replacing XML files by a NoSQL DBMS like MongoDB [26] during the MapReduce algorithm beacuse it

seems to scale better [27]. We are working in that direction, also considering to other kind of knowledge discovery using data mining techniques to exploit the huge amount of data (i.e. Big Data) generated by simulations and stored in the cloud.

REFERENCES

[1] Draghici, S., Khatri, P., Tarca, A. L., Amin, K., Done, A., Voichita, C., ... & Romero, R. (2007). A systems biology approach for pathway level analysis.Genome research, 17(10), 1537-1545.

[2] Hasty, J., McMillen, D., Isaacs, F., & Collins, J. J. (2001). Computational studies of gene regulatory networks: in numero molecular biology. Nature Reviews Genetics, 2(4), 268-279.

[3] Deloukas, P., Schuler, G. D., Gyapay, G., Beasley, E. M., Soderlund, C., Rodriguez-Tome, P., ... & Vega-Czarny, N. (1998). A physical map of 30,000 human genes. Science, 282(5389), 744-746.

[4] Shannon, Paul, et al. "Cytoscape: a software environment for integrated models of biomolecular interaction networks." *Genome research* 13.11 (2003): 2498-2504.

[5] Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., ... & Stein, L. (2005). Reactome: a knowledgebase of biological pathways.Nucleic acids research, 33(suppl 1), D428-D432.

[6] Cerami, E. G., Gross, B. E., Demir, E., Rodchenkov, I., Babur, Ö., Anwar, N., ... & Sander, C. (2011). Pathway Commons, a web resource for biological pathway data. Nucleic acids research, 39(suppl 1), D685-D690.

[7] Bock, M., Scharp, T., Talnikar, C., & Klipp, E. (2014). BooleSim: an interactive Boolean network simulator. Bioinformatics, 30(1), 131-132.

[8] Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics,26(10), 1378-1380.

[9] Albert, I., Thakar, J., Li, S., Zhang, R., & Albert, R. (2008). Boolean network simulations for life scientists. Source code for biology and medicine, 3(1), 1-8.

[10] Zheng, J., Zhang, D., Przytycki, P. F., Zielinski, R., Capala, J., & Przytycka, T. M. (2010). SimBoolNet—a Cytoscape plugin for dynamic simulation of signaling networks. Bioinformatics, 26(1), 141-142.

[11] Weaver, D. C., Workman, C. T., & Stormo, G. D. (1999, January). Modeling regulatory networks with weight matrices. In *Pacific symposium on biocomputing* (Vol. 4, pp. 112-123).

[12] Matsuno, H., Doi, A., Nagasaki, M., & Miyano, S. (2000). Hybrid Petri net representation of gene regulatory network. In *Pacific Symposium on Biocomputing* (Vol. 5, No. 338-349, p. 87). Singapore: World Scientific Press.

[13] Chen, T., He, H. L., & Church, G. M. (1999, January). Modeling gene expression with differential equations. In *Pacific symposium on biocomputing* (Vol. 4, No. 29, p. 4).

[14] Gershenson, C. (2004). Introduction to random Boolean networks. *arXiv preprint nlin/0408006*.

[15] McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., ... & DePristo, M. A. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome research, 20(9), 1297-1303.

[16] Meng, M., Gao, J., & Chen, J. J. (2013, December). Blast-Parallel: The parallelizing implementation of sequence alignment algorithms based on Hadoop platform. In Biomedical Engineering and Informatics (BMEI), 2013 6th International Conference on (pp. 465-470). IEEE.

[17] Grace, R. K., Manimegalai, R., & Kumar, S. S. (2014, March). Medical Image Retrieval System in Grid Using Hadoop Framework. In Computational Science and Computational Intelligence (CSCI), 2014 International Conference on (Vol. 1, pp. 144-148). IEEE.

[18] Zhang, Y., Zhang, R., Chen, Q., Gao, X., Hu, R., Zhang, Y., & Liu, G. (2012, September). A Hadoop-based massive molecular data storage solution for virtual screening. In ChinaGrid Annual Conference (ChinaGrid), 2012 Seventh (pp. 142-147). IEEE.

[19] Hadoop - Apache Software Foundation project home page. [http://hadoop.apache.org/].

[20] Xu, G., Xu, F., & Ma, H. (2012, August). Deploying and researching Hadoop in virtual machines. In Automation and Logistics (ICAL), 2012 IEEE International Conference on (pp. 395-399). IEEE.

[21] Llorente, I. M., & Montero, R. S. (2011). OpenNebula.

[22] Taylor, R. C. (2010). An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC bioinformatics, 11(Suppl 12), S1.

[23] Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., ... & Baldeschwieler, E. (2013, October). Apache hadoop yarn: Yet another resource negotiator. In Proceedings of the 4th annual Symposium on Cloud Computing (p. 5). ACM.

[24] Borthakur, D. (2008). HDFS architecture guide. Hadoop Apache Project, 53.

[25] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.

[26] MongoDB - MongoDB project home page. [http://www.mongodb.org/].

[27] Dede, E., Govindaraju, M., Gunter, D., Canon, R. S., & Ramakrishnan, L. (2013, June). Performance evaluation of a mongodb and hadoop platform for scientific data analysis. In *Proceedings of the 4th ACM workshop on Scientific cloud computing* (pp. 13-20). ACM.

[28] Benso, A., Di Carlo, S., Politano, G., Savino, A., & Vasciaveo, A. (2014). An extended gene protein/products boolean network model including post-transcriptional regulation. *Theoretical Biology and Medical Modelling*, *11*(Suppl 1), S5.

[29] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., ... & Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, *13*(11), 2498-2504.

[30] Politano, G., Benso, A., Savino, A., & Di Carlo, S. (2014). ReNE: A Cytoscape Plugin for Regulatory Network Enhancement. *PloS one*, *9*(12), e115585.

[31] Politano, G., Savino, A., Benso, A., Di Carlo, S., Rehman, H. U., & Vasciaveo, A. (2014). Using Boolean networks to model post-transcriptional regulation in gene regulatory networks. *Journal of Computational Science*, *5*(3), 332-344.

[32] Yao, Y., Wang, J., Sheng, B., Lin, J., & Mi, N. (2014, June). HaSTE: Hadoop YARN Scheduling Based on Task-Dependency and Resource-Demand. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on* (pp. 184-191). IEEE.

[33] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on* (pp. 1-10). IEEE.

[34] The Apache Software Foundation - ASF home page. [http://www.apache.org/].

[35] Yang, H. C., Dasdan, A., Hsiao, R. L., & Parker, D. S. (2007, June). Map-reduce-merge: simplified relational data processing on large clusters. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (pp. 1029-1040). ACM.

[36] EBNT Simulator Software Repository - [https://github.com/sysbio-polito/ebnt-simulator]

[37] Benso, A., Di Carlo, S., Rehman, H. U., Politano, G. M. M., Savino, A., Squillero, G., Vasciaveo A., & Benedettini, S. (2013, April). Accounting for post-transcriptional regulation in boolean networks based regulatory models. In *International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2013, Granada, ES, 18-20 March , 2013*. pp. 397-404.

[38] Kanehisa, M. and Goto, S.; KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 28, 27-30 (2000)