On parallelizing graph theoretical approaches for identifying causal genes and pathways from very large biological networks

Jeethu V. Devasia
Priya Chandran
Garge Shreya
Aparna R.
Abijith R.
Department of Computer Science and Engineering
National Institute of Technology Calicut
INDIA

ABSTRACT

Biological networks by nature are hard to process in real time for most of the applications. This work focuses on improving the speed of processing biological networks, in particular, faster traversal of genomes which have been mapped into a network for the detection of causal genes and associated pathways. Inference of disease causing genes and their pathways has achieved a crucial role in computational biology because of its practicality in understanding the major causal genes and their interactions that lead to a disease state, and suggesting new drug targets. In this work, Hadoop's distributed storage system has been used to store the molecular interaction network. Graph parallel processing techniques of Hadoop MapReduce, in conjunction with graph theoretical approaches have been utilized to improve the accuracy of results and execution time on benchmark data.

CCS CONCEPTS

• **Computing methodologies** → **MapReduce algorithms;**
• **Applied computing** → *Recognition of genes and regulatory elements; Health informatics;*

KEYWORDS

MapReduce, Parallelization, Causal genes, Dysregulated pathways, Molecular interaction networks

ACM Reference format:
Jeethu V. Devasia, Priya Chandran, Garge Shreya, Aparna R., and Abijith R.. 2017. On parallelizing graph theoretical approaches for identifying causal genes and pathways from very large biological networks. In *Proceedings of Second International Conference on Internet of Things and Cloud Computing, Cambridge, United Kingdom, March 22 2017 (ICC '17),* 6 pages.

ICC '17, March 22 2017, Cambridge, United Kingdom

DOI: http://dx.doi.org/10.1145/3018896.3056793

DOI: http://dx.doi.org/10.1145/3018896.3056793

1 INTRODUCTION

Hadoop MapReduce is a software framework for developing applications which process massive amount of data that may span multi-terabytes in size, in parallel in a reliable, fault tolerant manner on large clusters on thousands of nodes of commodity hardware. The input data is usually split by the MapReduce job into independent chunks of data which are input to the map job thereafter. The map tasks process and reformat the data in a completely parallel manner which then undergo a sort and shuffle by the framework before these are input to the reducer tasks. Typically, a file-system is used to store both the input and the output of the job. Scheduling and monitoring tasks and re-executing the failed tasks are managed by the framework itself. In this way, the map and reduce functions can be executed on smaller subsets of large datasets, which in effect provides the scalability that is critical for big data processing [24].

Graph processing on Hadoop has become a popular and fast developing research topic in the recent past because of its applications in diverse fields. Hadoop MapReduce technology enables us to tackle huge graphs by scaling up many graph algorithms to run on clusters of machines in a parallel fashion. For this, parallel processing model with data partitioning is used in graph processing on Hadoop.

The common pattern followed by the algorithms in Hadoop MapReduce follows:

(1) In the **map** phase, input is processed, and certain key-value pairs are generated based on the algorithm.
(2) In the **reduce** phase, results are obtained based on the algorithm and incoming values.

Here, the aim is improvements to the execution time and accuracy of results for inferring disease causing genes and associated pathways in molecular interaction networks using map/reduce framework in conjunction with graph theoretical approaches. The complexity of computations on molecular interaction networks is posed by the incredibly large number of genetic connections, which leads to very large processing times for the network. So, parallelization of approaches that are efficient in terms of accuracy of results and execution

time, has been attempted in this work to get the results in pragmatic time. Parallelization of all the related works in literature for tackling the problem of identifying causal genes and pathways has been implemented using map/reduce framework and analyzed based on serial execution of all the approaches using multiple real datasets. To the best of the authors' knowledge, this is the first parallel implementation for tackling the problem of inferring causal genes and associated pathways.

This paper is organized as follows. Section 2 gives an overview of the reported works in literature and Section 3 provides necessary information for understanding the problem of identifying disease causing genes and dysregulated pathways and a brief summary of the parallelized algorithms for addressing the problem. Section 4 explains parallelization of the algorithms using MapReduce approach. Section 5 deals with experimentation on real datasets, Section 6 gives analyses of the results and Section 7 concludes the paper.

2 RELATED WORKS

2.1 MapReduce framework for graph processing

The fundamentals of Hadoop Distributed File System (HDFS) and the working of MapReduce and large graph analysis using Hadoop are presented by Tom White in *Hadoop : The Definitive guide* [24]. It provides a detailed explanation of the anatomy of a MapReduce Job run, map phase, shuffle and sort phase and reduce phase. Extension of the MapReduce paradigm to graph data and the appropriate generation of key-value pairs to enable parallelization of graph processing are given in [2]. An iterative algorithm in the MapReduce framework, drawbacks of using MapReduce in iterative processing, major issues based on execution of tasks and design of modified MapReduce are addressed in [19]. A detailed study of parallel data accession distributed file systems like HDFS and the problems caused by not focusing on the distributed I/O resources and global data distribution are presented in [27]. [25] focuses on finding structures and characters of large-scale graphs and explains how the MapReduce framework and its implementation on Hadoop can be used to deal with large, complex graphs. 3-clique enumeration for large scale graphs using cluster system with the help of MapReduce is also presented.

2.2 MapReduce framework for computational biology

An overview of the current usage of Hadoop and associated open source software projects in bioinformatics that focus on next-generation sequencing analysis are explained in [22]. A survey of MapReduce framework operation in different applications of bioinformatics is presented in [29].

3 IDENTIFICATION OF CAUSAL GENES AND DYSREGULATED PATHWAYS

The basics of the techniques for identification of causal genes and dysregulated pathways and graph algorithms for finding the same is presented in this section. The proposed approach is to improve the time taken for the identification of causal genes and dysregulated pathways by traversing the molecular interaction network using parallelization. It involves the usage of Hadoop distributed storage and MapReduce paradigm and tweaking them to apply for graph traversal algorithms for causal gene detection and associated pathways for improved accuracy of results.

3.1 Biologic information

A *gene* consists of a segment of DNA and it is the molecular unit of heredity of a living organism. It is possible to quantify the level at which a particular gene is expressed during the process of making a biologically functional molecule of either RNA or protein [12]. i.e., Each gene is associated with a numerical *expression value* [10], [12], [16]. Apart from these, each gene interacts with some other gene [11]. These genetic interactions can be represented as a network known as *protein network*. These networks form the basis of understanding cellular processes [13], [30].

Expression values of genes are useful in identifying diseases [17]. If *expression levels (or values)* of a gene change between two sample groups of healthy and affected individuals, then that gene is said to be a *differentially expressed* gene [6], [8], [10]. In many human diseases, some genes' expression values vary significantly. There is an increase or decrease in expression values, from that of disease free persons [6], [26].

The set of disease causing genes that lead to a disease state is known as *causal genes*. The set of genes that may cause a disease state is known as *candidate causal genes*. Since, Transcription Factors alter the expression of genes, the set of candidate causal genes that are linked with Transcription Factors is referred to as *target genes* [5], [23]. The basic problem is to find the set of *causal genes*, where the set of *causal genes* is a subset of the set of *candidate causal genes* and *candidate causal genes* and *target genes* are known. The set of pathways in which *causal genes* are the major members is known as *dysregulated pathways*.

Since, this paper deals with the parallelization of the major works for identifying causal genes and dysregulated pathways simultaneously, a brief summary of each parallelized algorithm is given next.

3.1.1 Random walk approach. An interaction network is assembled using molecular interaction data such as protein-protein interactions, phosphorylation events and transcription factor-DNA interactions. A random walk is performed from the target genes and continued by selecting a vertex based on transition probabilities at each step. Pathways connecting causal genes and target genes and a subset of causal genes that affect the disease state are identified [23].

3.1.2 eQED algorithm. The expression quantitative trait loci (eQTL) electrical diagrams (eQED) approach attempts to improve on the Random walk approach by modeling the biological network into a very large and complex electrical circuit. There is a considerable work in available literature showing the equivalence between electric networks and random walks [1], [7], [28]. The large electric circuit is solved using Kirchoff's law and Ohm's law and currents flowing through all the edges are calculated. Causal gene is determined as the one with the highest current flowing through it. This approach gives a deterministic solution as opposed to stochastic random walk. The essence of the electric circuit model is that current flow on the edges indicate information flow direction in the biological system and the path from a source which has maximum flow would indicate the direction with the strongest signal transmission [20].

3.1.3 Current flow algorithm with multiple sources and sinks. Motivated by [20], molecular interaction is considered as an electric circuit and conductance of each edge is defined based on gene expression values. The magnitude of the current flow is calculated after finding voltages, by solving a set of linear equations. The genes with significant value of current flow are taken as causal genes. The shortest paths in the set of all maximum current paths for each pair of source and sink are considered as dysregulated pathways [14], [28].

3.1.4 Randomized Rounding algorithm. Randomized Rounding integrates thresholding of edges with the current flow approach [14], [28]. This algorithm brings improvement in both the paths traversed and the time required by selectively eliminating edges based on some chosen threshold value. This is done based on the assumption that in a very large network, edges with weights less than a certain threshold contribute to less probable paths. In the reduced network, maximum current flow paths are selected using randomized rounding approach which are considered as the dysregulated pathways and members of dysregulated pathways are considered as causal genes [3].

3.1.5 Rounding with Min-Heap algorithm. In this heuristic algorithm, edge-weights are defined based on expression values of genes for the molecular interaction network. It attempts to improve on the time taken to find all the causal genes and pathways by integrating approximation algorithm with vertex pruning techniques. The set of the explored paths is taken as the dysregulated pathways and the members of the pathways are taken causal genes [4].

3.1.6 Graph pruning approach. This algorithm combines both vertex pruning and edge pruning techniques in an iterative manner to reduce the size of the graph (pruning) by making sure that only less important members are getting removed each time. The collection of selected paths is considered as the dysregulated pathways and the members of the pathways are considered as causal genes [5].

4 PARALLELIZATION USING MAPREDUCE APPROACH

For all the algorithms that identify causal genes and associated pathways, the input is a genetic interaction sequence mapped into a network. The aim is to find the genes that affect the interactions most, and their associated pathways. Parameters like currents flowing through edges, correlation co-efficients are used as approximations for the strength of interactions between the genes. Genes that are most relevant in altering these interactions are found by finding most appropriate pathways in the network and checking for genes which occur the most in these pathways. All the algorithms for finding causal genes and dysregulated pathways work by traversing a graph for finding paths based on certain approach-specific parameters.

The high level paradigm that applies to all these algorithms is the parallelization of the basic graph traversal problem. i.e, given the network, sources, sinks and the parameters, find paths and members of the paths. The first step in the design is the storage of the input network. The network is loaded into HDFS, where it is broken down into smaller blocks and stored across the cluster.

But, this data cannot be worked up in parallel because information corresponding to neighbours is stored on physically different locations across the cluster. To solve this, the Map phase is used to reformat the input data to make it a 'property graph' such that all the information pertaining to computation at a certain vertex/edge is stored with that vertex/edge itself [9]. i.e., each vertex has an identifier called VertexID and edge has corresponding source and destination vertex identifiers and edge attributes. These properties are stored with each edge and vertex in the graph.

The mapper generates key-value pairs from the input. Parallelization techniques described in this paper converts the molecular interaction networks depicting different types of edges linking genes/proteins into property graphs [3], [14], [18], [21]. To enable smooth processing despite using distributed storage, which may put connected vertices on different clusters, the abstraction used for the representation of the network is the adjacency list/matrix data structure and the special data structure introduced in Spark called the 'triplets' [9]. Networks are represented as key-value pairs, in which the VertexID is considered as the key and a complex record called a 'tuple' that contains the list of end vertices corresponding to edges and other attributes of the edges is considered as the value asscciated with the key. The entire network is thus distributed across the cluster.

Even though different parts of the network may be stored on different machines, such a network can be processed using MapReduce triplets. The mapper reads the input and for each source as a key, it stores the input network and parameters corresponding to that source in the value corresponding to that key. The concept of generating Mapper output in such a way that the computation of the reducer on a certain key value pair is independent of other pairs is employed in the implementation. i.e., each source is considered as a key and

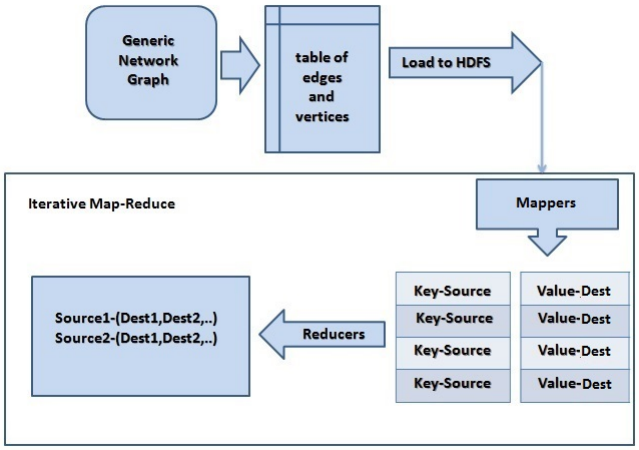

Figure 1: General overview of the design

the corresponding network or property graph is considered as the value associated with it.

Since these key-value pairs contain all the information required to find paths from a particular source to any destination, these pairs are passed on and are worked in parallel by the reducers. Reducers execute the corresponding algorithm for causal gene and pathway identification. So, when one reducer finishes executing, the paths from one source to all destinations are found. Since, the reducers are being executed in parallel, the execution time that would be consumed while executing sequentially, would get divided by the number of sources.

Figure 1 gives a pictorial representation of the design.

5 EXPERIMENTATION

Random walk approach, eQED algorithm, Current flow algorithm with multiple sources and sinks and Randomized rounding algorithm have been implemented on Hadoop-2.6.2 single node cluster (16GB RAM) with Java Runtime Environment-7 and Rounding with min-heap has been implemented on a three node virtual machine cluster and Graph pruning approach has been implemented on a multi node cluster with 64GB RAM. All the mappers have been coded in Python. Reducers for Random walk approach, eQED algorithm, Current flow algorithm with multiple sources and sinks, Randomized rounding algorithm and Rounding with min-heap algorithm have been coded in Python and Graph pruning approach has been coded in C. The serial versions of the approaches discussed in Section 3.1 have been implemented using R/C languages.

5.1 Selection of the input data and benchmark data

For experimentation, gene expression data of Pancreatic cancer of the species, Mus musculus (Mouse) and Rattus norvegicus (Rat) have been collected from the National Center for Biotechnology Information (NCBI) sponsored Gene Expression Omnibus data repository [3]. These data contain the set of *differentially expressed* genes and associated *gene expression values* of multiple healthy and disease instances. Data are normalized using Robust Multi-array Average method and statistically significant data are selected using t-test. This set of *differentially expressed* genes is considered as *candidate causal genes*. Gene interaction networks of different cases are downloaded from BioGRID database for each *candidate causal gene*. Then, the genes that are *differentially expressed* are selected from the fetched data. Target genes defined in Section 3.1, are taken as the source nodes and remaining genes are taken as sink nodes [5].

For Rattus norvegicus dataset, the set of target genes consists of 46 genes and the set of candidate causal genes consists of 140 genes. For Mus musculus dataset, the set of target genes consists of 30 genes and the set of candidate causal genes consists of 28 genes. Different cardinalities of molecular interaction networks follow: 1135 edges and 58 nodes with a total of 30 such graphs for Mus musculus dataset and 729 edges and 186 nodes with a total of 46 such graphs for Rattus norvegicus dataset [5].

Data from different databases and literatures have been curated as benchmark data for different cases. For this, NCBI sponsored Gene database, Aceview, Uniprot database and literatures from NCBI, Nature, Nucleic Acid Research have been considered [15]. Benchmark data have been obtained for the species Mus musculus and Rattus norvegicus as 58 genes and 92 genes respectively [5].

5.2 MapReduce Implementation

Sources, sinks and all the edge weights corresponding to all the sources comprise the input to the mapper. What enables parallel processing of graphs is the generation of key-value pairs such that the computation pertaining to any key is independent of the other keys. i.e, The output key-value pairs generated by the mapper should be such that for any key, all the information needed to compute the specific parameters associated with each algorithm should be available in the value associated with it. This enables smooth computation when different keys are being processed by different reducers on different nodes.

For each algorithm, the key would be the source, and to find the path, the list of target genes and the edge weights in the network corresponding to the sources are needed. So, these would constitute the value associated with the source key. When the reducer finishes processing a single source property-graph (key-value) pair corresponding to a particular algorithm, it will have found all the resultant paths from that particular source (key) to all the destinations for that algorithm.

Initial input size is approximately 9MB-10MB and because of the large number of interconnections among vertices in the network, even one iteration of the algorithm, i.e., finding paths from one source to destination itself consumes a large amount of time and a larger amount of memory, ranging from Megabytes to Gigabytes.

6 ANALYSES OF THE RESULTS

Comparison between serial and parallel execution time for different approaches on all the datasets follows. Here, CPU time used in executing the program alone is given. Large amount of execution time in certain methods like current flow approach in serial mode is due to the way by which these methods process the intermediate results. As seen from the results presented in Figures 2 and 3, reduction and uniformities in execution time of these algorithms have been obtained.

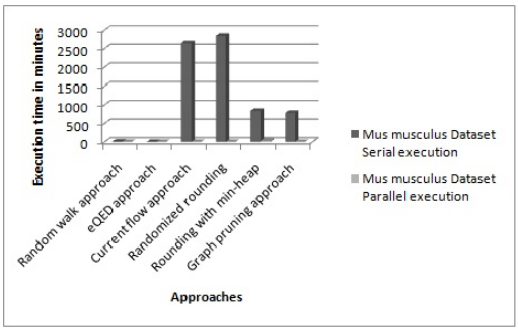

Figure 2: Execution time of Mus musculus

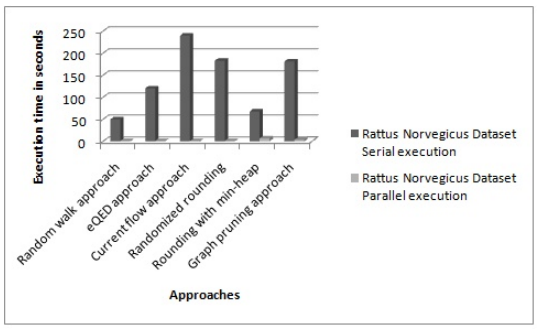

Figure 3: Execution time of Rattus norvegicus

Accuracy of the results based on benchmark data is presented in the Figure 4. Accuracies of the parallel execution are comparable to the values obtained sequentially.

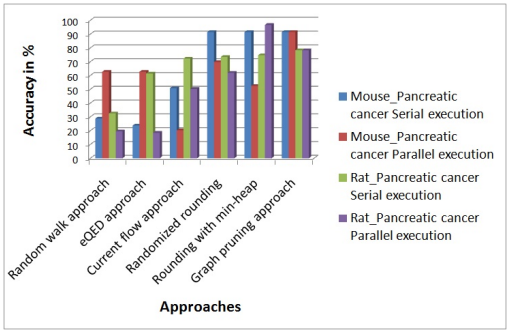

Figure 4: Accuracy of the results

7 CONCLUSION

This paper focuses on reducing the execution time of several causal genes and pathways identification algorithms because they are computationally intensive while executing sequentially. Proposed approach has attained this objective by glueing Hadoop MapReduce framework together with graph theoretical approaches. But, the number of reducers and thereby reduction in execution time depends on the number of sources of the input network. This approach can be largely extended to adapt to any other applications with the same input structure.

REFERENCES

[1] Przytycka TM Cho D-Y, Kim Y-A. 2012. Chapter 5: Network Biology Approach to Complex Diseases. *PLoS Computational Biology* 8(12) (2012).

[2] Jonathan Cohen. 2009. Graph Twiddling in a MapReduce World. *Computing in Science Engineering* 11, 4 (2009), 29–41.

[3] Jeethu V. Devasia and Priya Chandran. 2014. Towards an Improved Algorithm for Modeling Information Flow in Biological Networks. In *International Conference on Advances in Computing, Communications, and Information Science*. Elsevier, 88–95.

[4] Jeethu V. Devasia and Priya Chandran. 2016. Inferring disease causing genes and their pathways: A mathematical perspective. *Cornell University Library* (2016). Temporary submission ID: 1700287.

[5] J. V. Devasia and P. Chandran. 2016. Who are the key players behind a disease state?: Outcomes of a new computational approach on cancer data. In *2016 International Conference on Bioinformatics and Systems Biology (BSB)*. 1–4.

[6] Alison Devonshire, Ramnath Elaswarapu, and Carole Foy. 2010. Evaluation of external RNA controls for the standardisation of gene expression biomarker measurements. *BMC Genomics* 11, 1 (2010), 662.

[7] Peter G Doyle and J Laurie Snell. 2000. *Random Walks and Electric Networks*. Mathematical Association of America, Washington DC.

[8] Alexessander Couto Alves Fredrik Barrenäs, Sreenivas Chavali and others. 2012. Highly interconnected genes in disease-specific networks are enriched for disease-associated polymorphisms. *Genome Biology* (2012).

[9] Mohammed Guller. 2015. *Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis*. Apress.

[10] Joshua W.K. Ho, Maurizio Stefani, Cristobal G. dos Remedios, and Michael A. Charleston. 2008. Differential variability analysis of gene expression and its application to human diseases. *Bioinformatics* 24, 13 (2008), i390–i398.

[11] Miko I. 2008. Epistasis: Gene interaction and phenotype effects. *Nature Education* 1(1):197 (2008).

[12] Ernest S. Kawasaki. 2010. The End of the Microarray Tower of Babel: Will Universal Standards Lead the Way? *Journal of Biomolecular Techniques* 17:200206 (2010).
[13] Ryan Kelley and Trey Ideker. 2005. Systematic interpretation of genetic interactions using protein networks. *Nat Biotechnology* 23(5) (2005).
[14] Przytycka TM Kim Y-A, Wuchty S. 2011. Identifying causal genes and dysregulated pathways in complex diseases. *PLOS Computational Biology* 7 (2011).
[15] Donna Maglott, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. 2005. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research* 33, suppl 1 (2005), D54–D58.
[16] Frank Millenaar, John Okyere, Sean May, Martijn van Zanten, Laurentius Voesenek, and Anton Peeters. 2006. How to decide? Different methods of calculating gene expression from short oligonucleotide array data will give different results. *BMC Bioinformatics* 7, 1 (2006), 137.
[17] Omar Odibat and Chandan K. Reddy. 2012. Ranking differential hubs in gene co-expression networks. *Journal of Bioinformatics and Computational Biology* 10 (2012), 1240002 (15 pages).
[18] Georgios Pavlopoulos, Maria Secrier, Charalampos Moschopoulos, Theodoros Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis Bagos. 2011. Using graph theory to analyze biological networks. *BioData Mining* 4, 1 (2011), 10.
[19] H. Singhal and R. M. R. Guddeti. 2014. Modified MapReduce framework for enhancing performance of graph based algorithms by fast convergence in distributed environment. In *Advances in Computing, Communications and Informatics (ICACCI), 2014 International Conference on*. 1240–1245.
[20] Silpa Suthram, Richard M Karp Andreas Beyer, and Trey Ideker Yonina Eldar. 2008. eQED: an efficient method for interpreting eQTL associations using protein networks. *Molecular Systems Biology* 4:162 (2008).
[21] Damian Szklarczyk, Andrea Franceschini, Michael Kuhn, Milan Simonovic, Alexander Roth, Pablo Minguez, Tobias Doerks, Manuel Stark, Jean Muller, Peer Bork, Lars J. Jensen, and Christian von Mering. 2010. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research* (2010).
[22] Ronald C. Taylor. 2010. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. In *Proceedings of the 11th Annual Bioinformatics Open Source Conference (BOSC) 2010*. BMC Bioinformatics.
[23] Zhidong Tu, Li Wang, Michelle N. Arbeitman, Ting Chen, and Fengzhu Sun. 2006. An integrative approach for causal gene identification and gene regulatory pathway inference. *Bioinformatics* 22, 14 (2006), e489–e496.
[24] Tom White. 2012. *Hadoop: The Definitive Guide*. O'Reilly Media / Yahoo Press.
[25] B. Wu, Y. Dong, Q. Ke, and Y. Cai. 2011. A parallel computing model for large-graph mining with MapReduce. In *Natural Computation (ICNC), 2011 Seventh International Conference on*, Vol. 1. 43–47.
[26] Chao Wu, Jun Zhu, and Xuegong Zhang. 2013. Network-based differential gene expression analysis suggests cell cycle related genes regulated by E2F1 underlie the molecular difference between smoker and non-smoker lung adenocarcinoma. *BMC Bioinformatics* 14, 1 (2013), 365.
[27] J. Yin and J. Wang. 2015. Optimize Parallel Data Access in Big Data Processing. In *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*. 721–724.
[28] Stefan Wuchty Yoo-Ah Kim, Jozef H Przytycki and Teresa M Przytycka. 2011. Modeling information flow in biological networks. In *Physical Biology*, Vol. 8. 2011 IOP Publishing Ltd, 1–9.
[29] Quan Zou, Xu-Bin Li, Wen-Rui Jiang, Zi-Yu Lin, Gui-Lin Li, and Ke Chen. 2013. Survey of MapReduce frame operation in bioinformatics. *Briefings in Bioinformatics* (2013).
[30] Gabriel stlund, Mats Lindskog, and Erik L. L. Sonnhammer. 2010. Network-based Identification of Novel Cancer Genes. *Molecular & Cellular Proteomics* 9, 4 (2010), 648–655.