Journal of Biology and Today's World

Journal home page: http://journals.lexispublisher.com/jbtw

Received: 23 March 2017 • Accepted: 30 May 2017

Short C

doi:10.15412/J.JBTW.01060602

ClusPhylo: Spark Based Fast and Reliable Approach for Reconstruction of Phylogenetic Network Using Large Databases

Shamita Malik[1*], Sunil Kumar Khatri[2], Dolly Sharma[3]

[1] Amity School of Engineering and Technology, Amity University, Uttar Pradesh, India

[2] Amity Institute of Information Technology, Amity University, Uttar Pradesh, India

[3] Computer Science and Engineering, Shiv Nadar University, India

*Correspondence should be addressed to Shamita Malik, Amity School of Engineering and Technology, Amity University, Uttar Pradesh, India; Tel: +911204392277; Fax: +911204659009; Email: smalik@amity.edu.

ABSTRACT

Phylogenetic examination has turned out to be fundamental part of investigation for evolution of "tree of life". This investigation is most vital in logical research for development of life; it is a measure of impressions among creatures. It is important during examination that is required in process of arranging scattered information. Due to the expansion of more information in the field of proteomics, the computational biology algorithms should be extremely productive and near to accuracy. The inference of expansive and precise phylogenetic trees has expanded in most recent couple of years. Early methodologies for phylogenetic derivation depended on single processor PCs. Nonetheless, for expansive number of taxa, it is not feasible to utilize single processor. This represents a test for more proficient and adaptable calculations that utilizates parallel and conveyed processing for phylogenetic surmising. In this research paper, a new algorithm ClusPhylo based on clusters is introduced for large datasets. The proposed algorithms upgrades tree development issue by partitioning input arrangement into groups builds beginning sub-trees from arrangements of clusters and consolidations sub-trees into a solitary tree by additive methodology. ClusPhylo is implemented on Apache Spark. The execution of calculation as far as conclusive log probability qualities and execution time is contrasted with understood calculations. The outcome comes about demonstrating that the proposed calculation is computationally effective, delivers better probability values and is versatile on fluctuating number of processors too.

Key words: Phylogenetic reconstruction, Apache Hadoop, Apache Spark, Sequence alignment.

Journal of Biology and Today's World is published by *Lexis Publisher*; Journal p-ISSN 2476-5376; Journal e-ISSN 2322-3308.

1. INTRODUCTION

Computational science and bioinformatics is an interdisciplinary field that creates and applies computational strategies to investigate huge accumulations of biological information, for example, hereditary successions, cell populaces or protein tests, to make new forecasts or find new science. The computational techniques utilized incorporate systematic strategies, scientific displaying and simulation. Progressions in sequencing advances have brought about uncommonly fast gathering of biological information, likewise displaying chances to lead huge scale developmental examinations for uncovering more important information than any time in recent research. Phylogeny reconstruction is an examination for generally developmental related reviews, deciding and envisioning transformative connections among numerous qualities or species (1-3). To endeavor maximum potential of enormous information, novel computational approaches are required for quick evaluation of datasets. Most such calculations are theoretically direct. In any case, the information is normally vast and the calculations must be disseminated crosswise over hundreds or a large number of machines with a specific end goal to complete in a sensible measure of time. The issues of how to parallelize the calculation circulate the information and handle disappointments plot to cloud the first straightforward calculation with a lot of complex code to manage these issues. Mostly algorithms based on maximum likelihood (4, 5). These strategies have turned out to be exceptionally

prominent for building phylogenetic trees from sequence data information. In any case, regardless of perceptible late advance, with huge and troublesome datasets current ML programs still require immense registering time and can get to be distinctly caught in terrible neighborhood optima of the probability work. At the point when this happens, the subsequent trees may in any case demonstrate a portion of the deformities (6). Currently algorithms for reconstructing phylogenetics are in view of single process model or parallel models (7, 8), but they cannot scale well with the drastically expanding size of info dataset. Keeping in mind the end goal to handle this challenge, Apache Hadoop came into picture to solve mining of enormous datasets. Many algorithms have been proposed for parallel and distributed programming structure generally received in the field of data innovation for Apache Hadoop. In these algorithms, data is basically stored on distributed file system. In such programming concepts, they are able to support application execution and accomplish great versatility from one machine to a bunch containing any size of nodes. Although Hadoop encourages various extensive online corporate like Amazon, Facebook, Google etc. that utilizes it for different sorts of information warehousing purposes (9-11). In any case, for iterative calculations, Hadoop absences of an effective algorithm that reserve calculation estimations. These calculation estimations are part of cache during MapReduce turns which can be required in several cases. A MapReduce turn also, in this way, endures an unavoidable and critical execution misfortune. Likewise, another distributed computing stage named Apache Spark has been proposed to beat this inadequacy. Spark effortlessly outflanks Hadoop by about up to 100x speedup (12, 13). Especially, little consideration has been committed to building up a Spark-based algorithm that can deal with huge dataset for quick and adaptable phylogeny reconstruction. In this research work, ClusPhylo that empowers programmed parallelization and dispersion of vast scale calculations, joined with an execution of this interface that accomplishes superior on vast groups of clusters. The programming model can likewise be utilized to parallelize calculations over different cores of a similar machine. Section 2 explains the implementation of Apache Spark algorithm that depicts an execution of cluster based computing. Section 3 has performance measurement of the ClusPhylo on real datasets in terms of computing speed and efficiency. Section 4 has future work.

2. MATERIALS AND METHODS

With the tremendous development in bioinformatics, there is a requirement for algorithm that empower fast mining of information on big data store. Sometimes these algorithms are so restricted to small number of data store. The brief methodology followed is shown with the help of flowchart shown in the Figure 1.

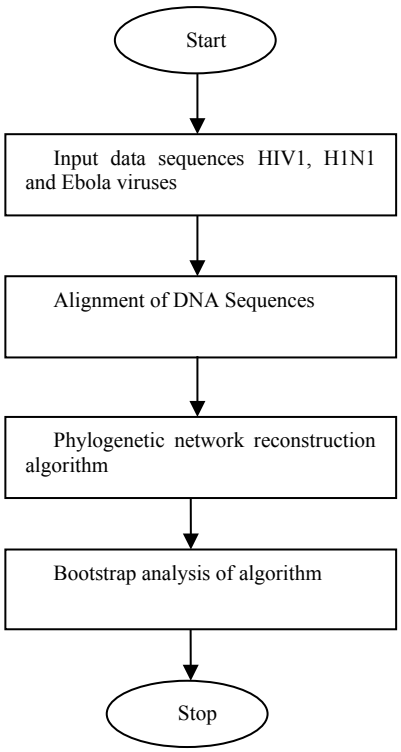

Figure 1. Flowchart depicting methodology in brief

The newly designed algorithm will mine information in relatively small fraction of time. For better runtime ClusPhylo is implemented in Scala, a JVM (Java Virtual Machine). The objective of this newly designed parallel algorithm is to investigate the required information utilizing data clusters to beat the restrictions of unique

Apache Hadoop. Likewise, there is a need to do the execution analysis for various arrangements of datasets and additionally for various numbers of hubs in the cluster. The Spark uses master/worker architecture (Figure 2). In this architecture single coordinator referred as master, will calls driver that further manages in which executor's runs.

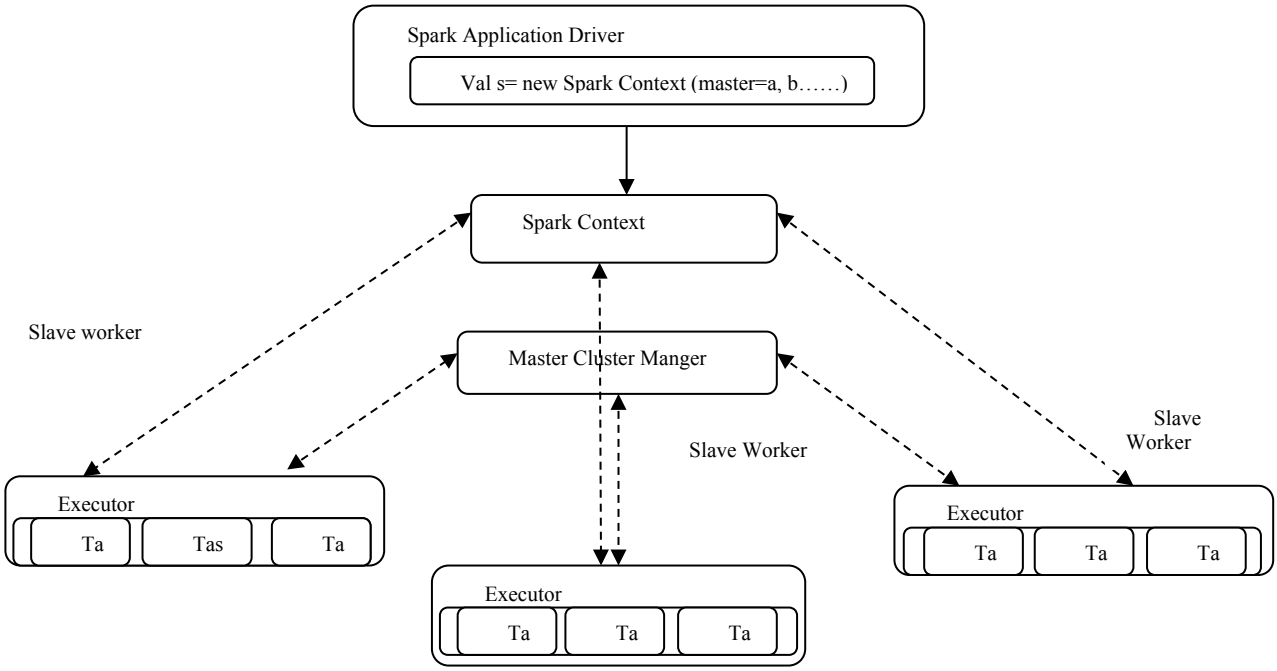

Figure 2. Spark architecture used in current methodology

The data is freely available on National Centre for Biotechnology Information. Three datasets are taken of viruses namely HIV1 (1.2 GB), H1N1 (1.66 GB) and Ebola (1.3 GB). The proposed system uses input in form of FASTA format. The objective function is to make clusters of highly correlated clusters (Figure 3).

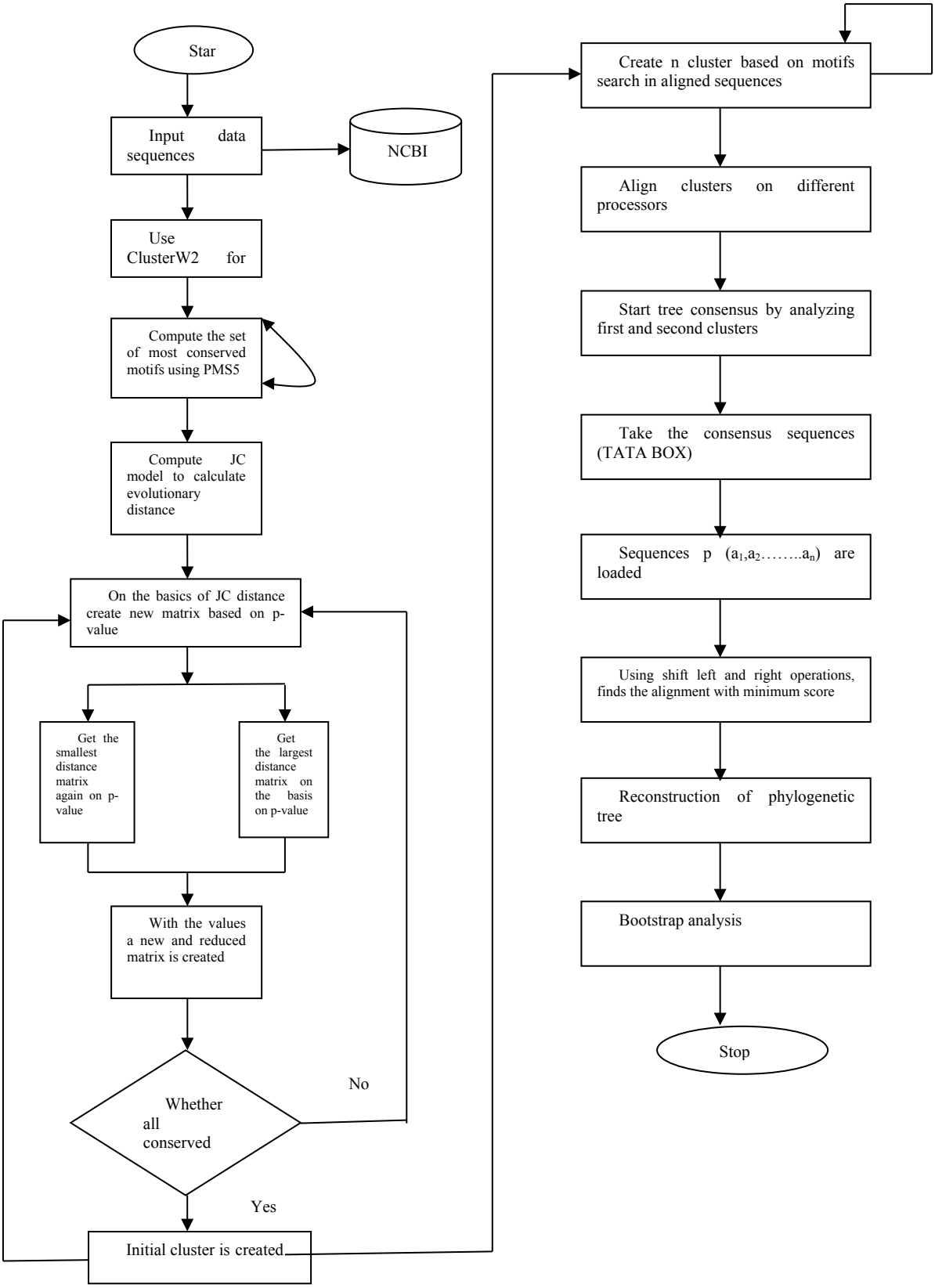

Figure 3. Flowchart for ClusPhylo algorithm

Initially while making clusters, we group the data-sets into sites based on identical data. The similarity matrix is created based on maximum likelihood parameter. One of the speediest and conceivable methods for dealing with this undertaking is to create parallel and distributed calculations for maximum likelihood. The objective in this review is to ascertain the precision and effectiveness of a parallelized calculation in following datasets surpassing 1000 taxa with most extreme probability investigation. The proposed solution considers both data and the computation

parallelism by improving performance, throughput and accuracy. The parallel approach is done by dividing the sequence into set of blocks and processing or making it to run parallel (14). Dividing the input data into set of blocks makes the execution of algorithm fast. Figure 3 shows methodology used in ClusPhylo.

3. RESULTS AND DISCUSSION

3.1. Experimental design

In this research, the performance of ClusPhylo is evaluated on more than 15 nodes. Every cluster is assigned on one particular node and every node have same configuration running Ubuntu server 14.04 with Apache Spark version 1.6.0 and Oracle JDK version 8u45 installed, equipped with a quad-core 2.1 GHz CPU, 36 GB RAM, 100 GB disk space, and a gigabytes network interface card. In this test environment, Spark cluster runs under standalone mode, sharing files with HDFS (Hadoop Distributed File System) bundled with Apache Hadoop version 2.4. The dataset is available on National Centre for Biotechnology Information. We have taken datasets of HIV1 (1.2 Gb), H1N1 (1.66 Gb) and Ebola (1.3 Gb) viruses.

3.2. Evaluation

In this research, the dataset sizes of input sequences are changed. As compared to previous research ReTF algorithm [8] the dataset was in megabytes but in the current methodology, it is in Gb. The prevalent of Spark is clear; the more evident execution is shown below in Figure 4.

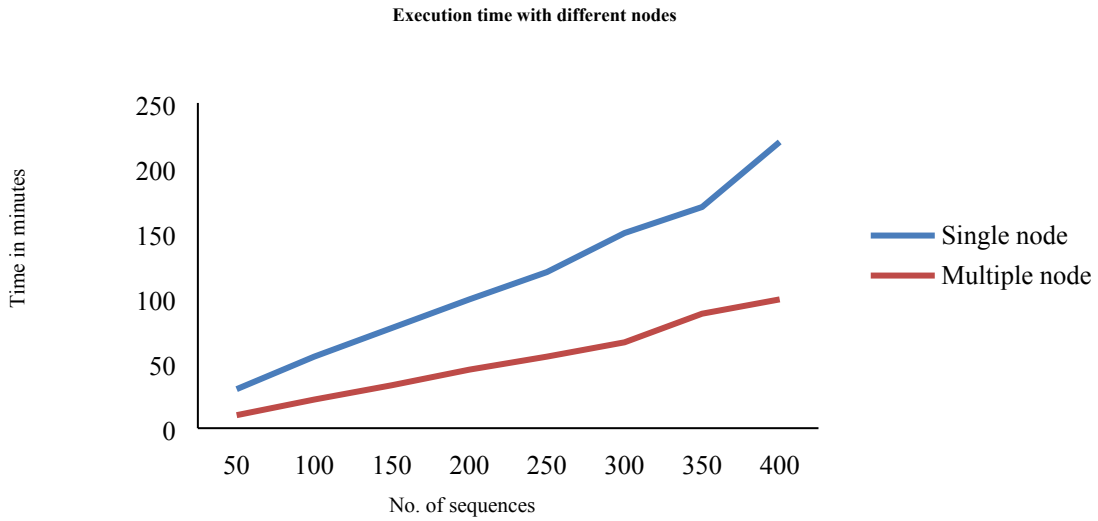

Figure 4. Execution time for 500 length sequence

The throughput here is number of succession arrangements made every minute (Figure 5). As normal an aggregate of three input sets were taken that have expanding number of groupings. Throughput was figured, if there should be an occurrence of both single node and multi nodes grouping. The throughput has progressively expanded as size of succession set increments. Additionally the distinction between the throughputs of a solitary node group and number of nodes additionally incremented bit by bit. The difference was low due to parallelizing overheads. The more number of nodes added means, the time is reduced drastically.

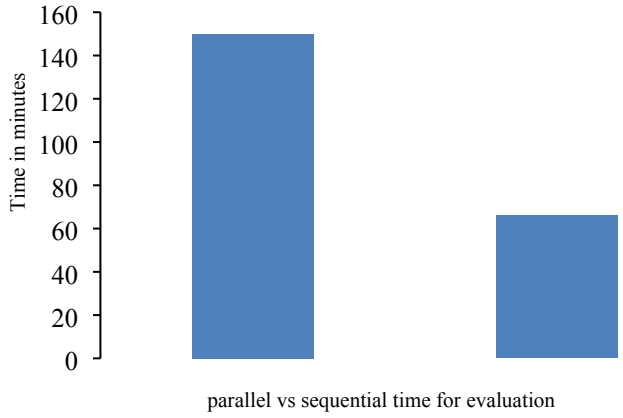

Figure 5. Bar graph for run time in sequential and parallel process

The three versions of algorithms, namely ReTF algorithm sequential algorithm (15), D-Phylo parallel implementation (16) and ClusPhylo are compared in this research. The ClusPhylo shows better performance (Figure 6).

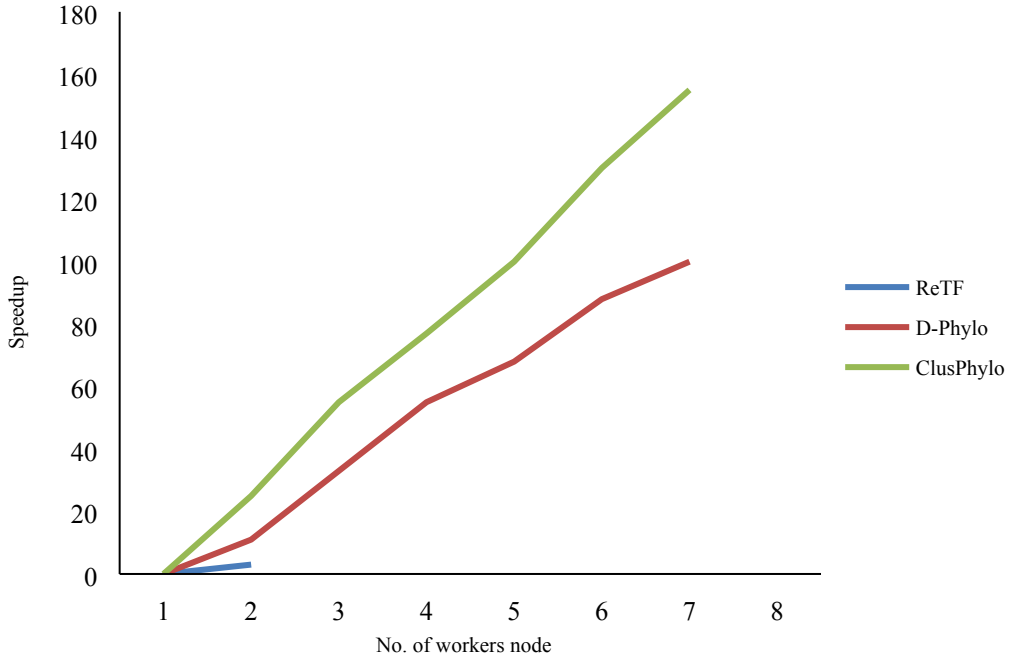

Figure 6. Speedup for reconstruction of phylogenetic network

4. CONCLUSION

Although these problems are biological problems but the huge amount of computations and number crunching makes them more suitable for computer scientists. The way in which Computer Science researchers can look at these problems is as follows: We are provided with a database which consists of sequences of DNA and our job is to mine this database and come up with patterns, their occurrences or relationships. Even though these problems have been studied for a very long time, but the mathematical or computational model built to solve the biological problems sometimes fail while working on real biological data. This is due to the fact that the assumptions based on which the model was built does not always hold in reality. The objective of this design is to create more realistic models algorithm to solve problems related to gene sequencing and mining of important patterns. These problems are not only important because we need to know how we came to existence and how we migrated, but majorly in identifying diseases like cancer, treatment of such diseases and drug discovery. The outcomes demonstrate that the proposed arrangement enhances the execution of phylogenetic analysis by utilizing Apache Spark .It provided results in much lesser time than single node algorithm. The proposed strategy is time effective approach and it catches the virus families better when contrasted with different algorithms in different technologies. A parallel approach of phylogenetic investigation is built utilizing processing powers that requires top of the line machines. However it may, at few points additionally required be moved forward, which incorporates the technique to part databases and parse the reports of ClustalW2. Furthermore, more assessment will be finished to adjust to more quality and protein succession

with much diverse length.

ACKNOWLEDGMENT

The authors thank Dr. Ashok K. Chauhan, Founder President, Amity University, for his support and encouragement along with providing us with the necessary infrastructure for research.

FUNDING/SUPPORT

This study was not supported by any research grant.

AUTHORS CONTRIBUTION

This work was carried out in collaboration among all authors.

CONFLICT OF INTEREST

The authors declared no potential conflicts of interests with respect to the authorship and/or publication of this paper.

REFERENCES

1. Hallett MT, Lagergren J, editors. Efficient algorithms for lateral gene transfer problems. Proceedings of the fifth annual international conference on Computational biology; 2001: ACM.
2. Sneath PH. Cladistic representation of reticulate evolution. Systematic Zoology. 1975;24(3):360-8.
3. Posada D, Crandall KA. The effect of recombination on the accuracy of phylogeny estimation. Journal of molecular evolution. 2002;54(3):396-402.
4. Hansen DR, Dastidar SG, Cai Z, Penaflor C, Kuehl JV, Boore JL, et al. Phylogenetic and evolutionary implications of complete chloroplast genome sequences of four early-diverging angiosperms: Buxus (Buxaceae), Chloranthus (Chloranthaceae), Dioscorea (Dioscoreaceae), and Illicium (Schisandraceae). Molecular phylogenetics and evolution. 2007;45(2):547-63.
5. Altekar G, Dwarkadas S, Huelsenbeck JP, Ronquist F. Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics. 2004;20(3):407-15.
6. Guindon S, Gascuel O. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic biology. 2003;52(5):696-704.
7. Olsen GJ, Matsuda H, Hagstrom R, Overbeek R. fastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. Computer applications in the biosciences: CABIOS. 1994;10(1):41-8.
8. Sakamoto C, Miyazaki T, Kuwayama M, Saisho K, Fukuda A. Design and implementation of a parallel pthread library (ppl) with parallelism and portability. Systems and Computers in Japan. 1998;29(2):28-35.
9. White T. Hadoop: The definitive guide: " O'Reilly Media, Inc."; 2012.
10. Ekanayake J, Pallickara S, Fox G, editors. Mapreduce for data intensive scientific analyses. eScience, 2008 eScience'08 IEEE Fourth International Conference on; 2008: IEEE.
11. Taylor RC. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC bioinformatics. 2010;11(12):S1.
12. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: Cluster Computing with Working Sets. HotCloud. 2010;10(10-10):95.
13. Wiewiórka MS, Messina A, Pacholewska A, Maffioletti S, Gawrysiak P, Okoniewski MJ. SparkSeq: fast, scalable, cloud-ready tool for the interactive genomic data analysis with nucleotide precision. Bioinformatics. 2014:btu343.
14. Matsunaga A, Tsugawa M, Fortes J, editors. Cloudblast: Combining mapreduce and virtualization on distributed resources for bioinformatics applications. eScience, 2008 eScience'08 IEEE Fourth International Conference on; 2008: IEEE.
15. Malik S, Sharma D, editors. Reconstructing phylogenetic network with ReTF algorithm (rearranging transcriptional factor). Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on; 2013: IEEE.
16. Malik S, Sharma D, Khatri SK. Parallel implementation of D-Phylo algorithm for maximum likelihood clusters. IET Nanobiotechnology. 2016.

Special Issue on:
'Role of IT in Bioinformatics and Neuro-informatics'

Guest Editors:

Prof. Deepshikha Bhargava
Amity University Rajasthan – India
dbhargava1@jpr.amity.edu

Dr. Ramesh C. Poonia
Amity University Rajasthan – India
rameshcpoonia@gmail.com

Dr. Swapnesh Taterh
Amity University Rajasthan – India
staterh@jpr.amity.edu

This paper is published as one of the selected papers that were presented at:
International Conference on Smart Computing and Communication (SmartTech-2017).