Proc. of the Intl. Conf. on Advances in Environmental and Bio-Technology -- AEBT 2014

ISBN: 978-981-07-8859-9

Development and Optimization of Integrated Pipelines for Phylogeny and Evolutionary Analyses

Santosh Servisetti, Guoqing Lu and Zhengxin Chen

***Abstract*— Recently development of automated and integrated pipelines for phylogeny and evolutionary analyses has received a lot of attention. In this paper, we describe our approach of pipeline development. Our design focuses on integration, not only in the integrated result (i.e., the pipelines), but also in the integrated approach in the design process, so that we can exploit all kinds of resources available to us. In addition, we want integrated pipelines not only work effectively, but also efficiently. In this paper, we describe our two-step design: A basic approach and an optimized approach which takes advantage of parallelism offered by supercomputers. We present the major phases of pipeline development, describe methods used in optimization (including the sequence-split-logic), show the experimental result, and compare it with non-optimized implementation.**

***Keywords*—Integrated pipelines, phylogeny, evolutionary analyses, parallelism, supercomputer, optimization**

I. Introduction

One of the major objectives in bioinformatics is to study Tree of Life in which phylogenetic analysis is a common research pathway. By conducting phylogenetic analysis researchers can find the origin of disease pathogens and may also learn the pattern which initiated the process, which in turn leads to a discovery of cure for a particular disease [8].

The computational process of finding the origin is very exhaustive and it requires initiating different processes at different points of time. Developing an automated, integrated pipeline for phylogeny and evolutionary analyses is an important and urgent task. To be more specifically, our goal is to develop a system which can get all the process of phylogenetic analysis done under a common hood so that upon a user submits his/her request, the system is able to generate the final results in reasonable amount time with certain accuracy.

Although there have been numerous efforts in developing automated and integrated pipelines, each has different features.

Santosh Servisetti, and Zhengxin Chen
College of Information Science and Technology,
University of Nebraska at Omaha, USA

Guoqing Lu
Department of Biology,
University of Nebraska at Omaha, USA

For example, the *CIPRES* project [9] was a big 5-year project with a global perspective, while the Hal project [15] has a much more manageable size project and shares some aspects in common with our own project. Yet our project differ from these existing approaches in that while conducting our project we have the following specific considerations in mind:

(a) Our design focuses on integration, not only in the integrated result (i.e., the pipeline), but also in the process: we are aimed to take an integrated approach in the design process, so that we can exploit all kinds of resources available to us. For example, we have employed our knowledge on XML and supercomputing, etc. The motivation behind this is that we believe a system-theoretical perspective holds the key for a holistic study of developing useful bio-techniques in our environment. After all, general systems theory was proposed by the *biologist* von Bertalanffy [17].

(b) We not only want to develop an automated, integrated pipeline which works effectively, but also efficiently. That means we also have optimization in mind for pipeline development. In particular, we explore how to take advantage of parallelism of computation which supercomputers can offer.

The two major approaches of this pipeline system are: 1) Basic pipeline approach for phylogenetic analysis on tools like PAUP, MRBAYES and GARLI; and 2) Advanced pipeline approach for BEAST.

The rest of this paper is organized as follows. In Section II, we provide a brief discussion on necessary background, including various kinds of existing software used in building the pipelines. In Section III, we present the basic approach for pipeline development and present experimental results. In Section IV, we further present an optimized approach and the splitting algorithm used (i.e., the sequence-split-logic), as well as the experimental result which is compared with the non-optimized approach. We conclude the paper in Section V where wrap up our paper by summarizing important findings

II. Background

Numerous tools are developed based on the need and are used at different phases of phylogenetic analyses. Below are the list of tools which are installed in the system and their descriptions. These include alignment tools, phylogenetic tools and result viewers, along with the supercomputer we have used. We also tried to provide the reference URL sites (whenever possible) so if a problem arises or a new version releases researchers can visit their site and get the update.

Proc. of the Intl. Conf. on Advances in Environmental and Bio-Technology -- AEBT 2014

ISBN: 978-981-07-8859-9

A. *Alignment tools*

In our pipeline system we use two tools for alignment: MUSCLE and MAFFT. A user can select the appropriate tool and align the sequence file. Result was then processed to the next stage.

MUSCLE [3] is a new computer program for creating multiple alignments of protein sequences. Elements of the algorithm include fast distance estimation using k-mer counting, progressive alignment using a new profile function referred to as the log expectation score, and refinement using tree dependent restricted partitioning.

MAFFT [7] offers various multiple alignment strategies. They are classified into three types: (a) the progressive method, (b) the iterative refinement method with the WSP score, and (c) the iterative refinement method using both the WSP and consistency scores. In general, there is a tradeoff between speed and accuracy.

MODELTEST [10] is a program for the selection the model of nucleotide substitution that best fits the data. The program chooses among 56 models, and implements three different model selection frameworks: hierarchical likelihood ratio tests (hLRTs), To use ModelTest we will need PAUP (see below).

B. *Phylogenetic tools*

Below are the tools our pipeline system uses for sequence classification. Classification is mainly used for the creation of names for groups whereas systematics goes beyond this to elucidate new theories of the mechanisms of evolution [4].

PAUP (Phylogenetic Analyses Using Parsimony) [16] version 4.0 is a major upgrade and new release of the software package for inference of evolutionary trees, for use in Macintosh, Windows, UNIX/VMS, or DOS-based formats. PAUP 4.0 and MacClade 3 use a common data file format (NEXUS), allowing easy interchange of data between the two programs. We use likelihood criteria for Paup while we calculate the results since it is highly compatible [11].

GARLI [18] performs heuristic Phylogenetic searches under the General Time Reversible (GTR) model of nucleotide substitution and its sub models, with or without gamma distributed rate heterogeneity and a proportion of invariant sites. The implementation of this model is exactly equivalent to that in PAUP, so that likelihood scores obtained by each program are directly comparable.

MrBayes [5] is a program for Bayesian inference [14] and model choice across a wide range of Phylogenetic and evolutionary models. MrBayes uses Markov chain Monte Carlo (MCMC) methods to estimate the posterior distribution of model parameters.

BEAST [2] is a cross-platform program for Bayesian Bayesian Markov Chain Monte Carlo (MCMC) analyses of molecular sequences. It is entirely orientated towards rooted, time-measured phylogenies inferred using strict or relaxed molecular clock models. It can be used as a method of reconstructing phylogenies but is also a framework for testing evolutionary hypotheses without conditioning on a single tree topology. Since BEAST uses MCMC to average over tree space, each tree is weighted proportional to its posterior probability. See [6] for more on Bayesian inference of phylogeny.

C. *Result Viewers*

Tracer [13] is a program for analyzing the trace files generated by Bayesian MCMC runs (that is, the continuous parameter values sampled from the chain). It can be used to analyze runs of BEAST, MrBayes, LAMARC and possibly other MCMC programs.

Figtree [12] is designed as a graphical viewer of phylogenetic trees and as a program for producing publication-ready figures. In particular it is designed to display summarized and annotated trees produced by BEAST.

D. *Firefly supercomputer*

Holland Computing Center at University of Nebraska at Omaha, supports a diverse collection of hardware that is here as a campus resource, and anyone on campus is welcome to apply for an account on these machines. Access to these resources is by default shared with the rest of the user community via various job schedulers. The main reason of using this supercomputer is to take advantage of parallelism, so that multiple sequence files can be processed in parallel.

III. Basic structure of phylogenetic pipeline

In order to build a pipeline, we need to identify all the different stages in it. Additionally we also need to find a way to process one after other storing intermediate results of each stage and passing it to the next stage. The flow of pipeline consists of four phases: Alignment, data formatting (using a tool developed earlier from our research group [1]), model test, and generating resultant trees. The four general phases of our overall design process are depicted in Fig. 1.

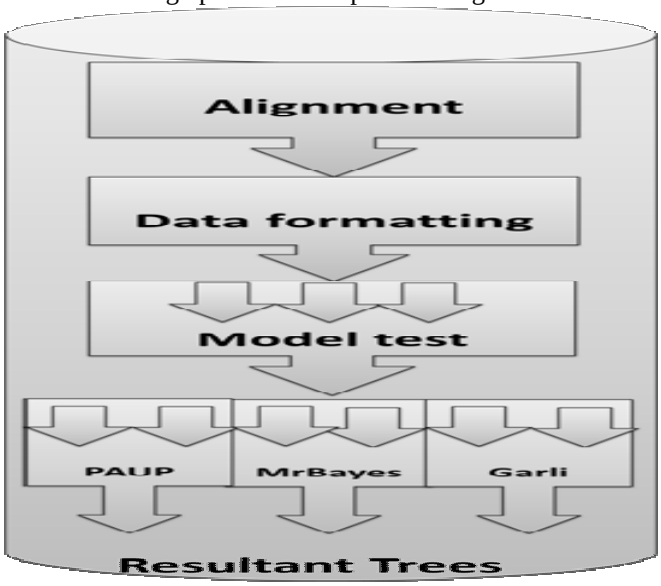

Figure 1. Major phases in the basic pipeline

Proc. of the Intl. Conf. on Advances in Environmental and Bio-Technology -- AEBT 2014

ISBN: 978-981-07-8859-9

IV. Development and optimization of Beast Phylogenetic Pipeline

BEAST is a tool using an advanced approach for evolutionary analyses and this pipeline is hosted on Firefly to make use of parallel execution. To take advantage of parallel computing, here are four important phases in development and optimization of Beast Phylogenetic Pipeline: (a) Data preprocessing, (b) Extracting and splitting sequence files, (c) Generating XML files using Beauti and (d) compiling XML files using BEAST. Fig. 2 shows the flow of Beast pipeline.

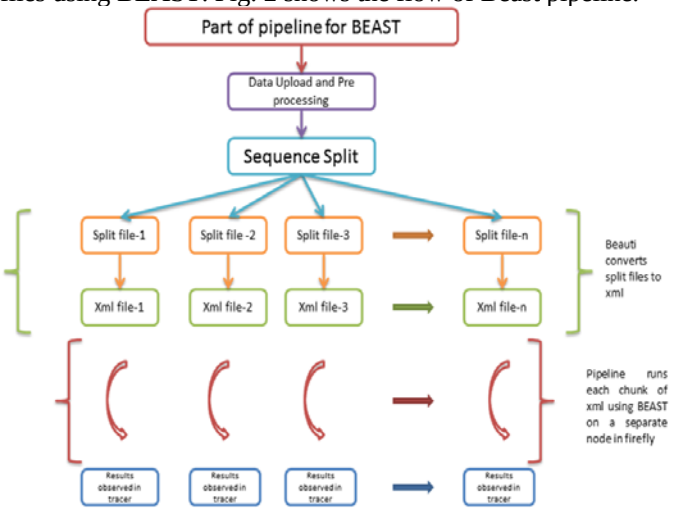

Figure 2 Data flow for Beast phylogenetic pipeline

To demonstrate the result produced by the pipeline, Fig. 3 shows the trace files of all the subsets of data and it also shows the total chain length 20000000 which is used to calculate results. The user can also observe the estimates and analyze how co-related the input file is. Effective Sample Size (ESS) is important estimate to check whether the results highly correlated or not. If ESS is greater than 350 then it is considered that the results generated are good for analyses. The user can observe all the resultant logs at once and can also analyze the combined results of all the files.

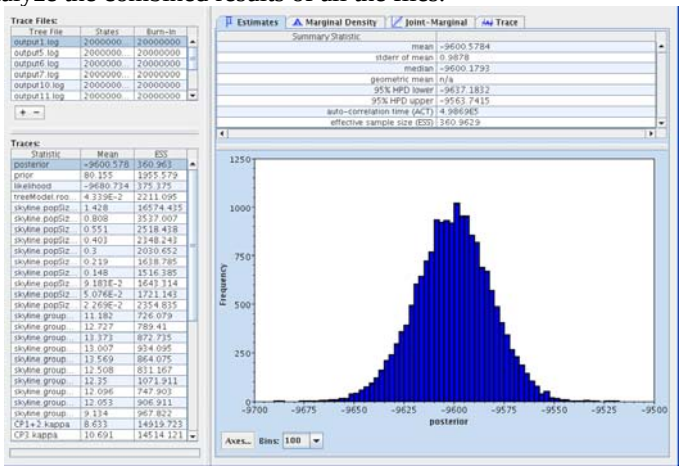

Figure 3. Resultant of file of first input dataset

In addition, the user can observe all the resultant logs at once and can also analyze the combined results of all the files. Fig. 4 shows the trace of all the 10 subsets of data with mean values in y-axis and states in x-axis.

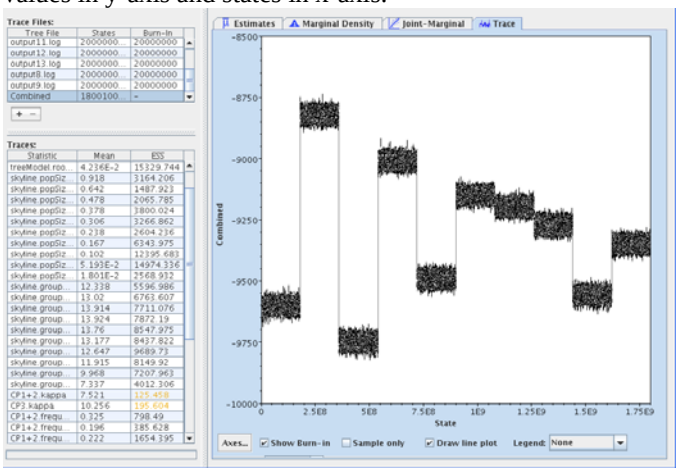

Figure 4. Trace of all the log files of subsets of data

The reason for a separate pipeline of BEAST is that it is very time consuming of performing analysis in BEAST and lots of computational resources are needed to generate results. For large sets of data even a supercomputer may have to take several weeks and even months to generate results. Therefore we have to optimize the implementation involving BEAST so that it can generate results in less time, yet without sacrificing accuracy. For this purpose we take a “divide and conquer” approach to split the original set of sequences into several files and execute them in parallel. Yet experiments have indicated that splitting the data completely by random does not guarantee good result. In order to optimize the pipeline, we have developed sequence extraction utilities Sequence-split-logic (Fig. 12), which specifies splits the given input sequence file into ‘N’ parts, Where Number ‘N’ will be calculated dynamically during the program execution. Upon the user submit his/her request, the script developed by us will call the ***sequence extraction utility*** to split the sequences into several files, based on the year and location grouping.

Since there are different criteria for splitting, we have developed several types of sequence split versions. By default the pipeline will select Sequence extraction version-1.Two alternatives are: Sequence Extraction v2 (Split based on Week Information) and Sequence Extraction v3 (which works for both FASTA/NEXUS files).

Sequence-split-logic will first of all run some tests on user uploaded data and get some information, including number of different years, number of different locations and total number of output files to be generated. Then the utility will group all the sequences based upon the year and locations. The next step is to create split files by randomly selecting sequences from all different years plus all different year-locations until the number of sequences in that file reaches 120. This process will execute until it generates all the output files.

Proc. of the Intl. Conf. on Advances in Environmental and Bio-Technology -- AEBT 2014

ISBN: 978-981-07-8859-9

Once the files are created, regular/split phase was complete and pipeline script will move to the next phase which is creating XML files.

Finally, pipeline script will create shell script files for every XML file and run each file on each node of Firefly.

After the Sequence-split-logic all the chunks of files are executed on Beauti with customized parameters to generate XML files. Here we can also speed up the process of generating XML files by saving the customized parameters in a template. We can obtain the template while we generate XML files for the rest of the Nexus split files. We have developed a shell script file and use it for each XML file to request a node in Firefly to run. After the Sequence-split-logic As this utility was built using java we can compile and execute like a normal java program. Once the program was executed resultant files are generated.

Fig. 5 shows the algorithm of sequence extraction. Data flow of sequence extraction utility is shown in Fig. 6.

```
Number of files: [Total Number of sequences/ Number of sequences in each file];
While (file < Number of files)
Begin
    While (sequence length in file < 120) Begin
    RandomSequence = GetRandomSequence(Input File);
    If((CheckSequenceExist(RandomSequence, file))
    Then  // skip sequence; continue to next sequence;
    Else
        If((CheckSequenceYearExist(RandomSequence, file) &&DiffYear< Distinct
Years)
        Then  //skip sequence; continue to next sequence;
        Else
            If((CheckSequenceLocationExist(RandomSequence, file) &&
DiffLoc < Distinct Locations)
            Then  //skip sequence; continue to next sequence;
            Else  //Add Sequence to the file and update location and year
information; Increment sequence length;
End; Save file; Increment file Number; End;
```

Figure 5. .Algorithm of sequence extraction

Based on the number of sequences in input file, several subsets of output files are generated. In this phase, by using sequence extraction utility we have observed good results in BEAST pipeline both in terms of accuracy and computational time. When BEAST pipeline was executed without using Sequence-split-logic and optimization, results were generated in 28 days. When we employed Sequence-split-logic in BEAST we got results in 3-4 days, the quality of results was not good. Table I shows the variety of optimization logic we used and the time in which results are generated. When we run BEAST using Sequence extraction utility with location and year clustering logic, We have observed good results and resultant log files are generated in 3-4 days which have an ESS of greater than 350.

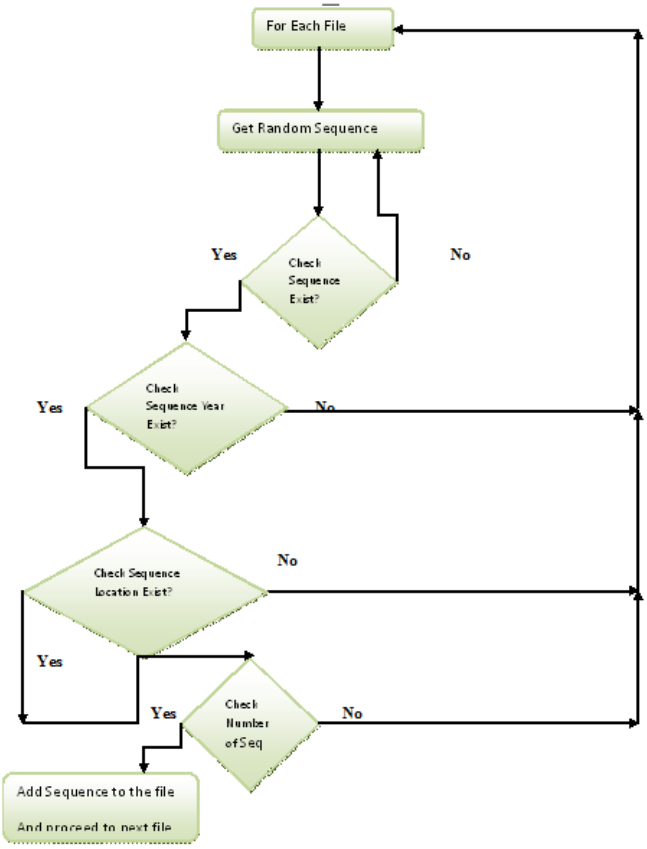

Figure 6. Data flow of sequence extraction utility

.

v. Conclusion

The aim of this research is to build a pipeline which integrates all the stages in the phylogenetic analysis and to optimize the flow of data in these stages to generate results in less computational time. We have succeeded in connecting all the stages in phylogenetic analysis and completed the objective of minimal user intervention. Our future work includes extending the current pipeline approaches to run in GPU clusters would be a nice improvement. In addition, sequence extraction utility can be further improved. Another possibility is consider how to take advantage of cloud computing.

Acknowledgment

This research is supported by NIH grant numbers P20 RR016469 from the INBRE program of the National Center for Research Resources and R01 LM009985-01A1 to G. Lu.

ISBN: 978-981-07-8859-9

TABLE I. TABLE TYPE STYLES STATISTICS OF SEQUENCE EXTRACTION UTILITY

Type of Optimization (logic)	Time to Generate results	Number of nodes:	Analysis
Single file with pipeline logic on Firefly super computer	28 days	1 node on Firefly (serial)	-Good evolutionary rate. -As per the Tracer log result was pretty good and accurate. - Effective Sample Size (ESS) was greater than 350
Number of sequences: 1700			
Number of chars/seq: 1745			
Splitting the input file randomly without Sequence-split-logic.	3 - 4 days	14 nodes on Firefly (parallel)	-Bad evolutionary rate for most of the files - Results are not similar for all the output files - Effective Sample Size (ESS) was less than 100
Details of input file:			
Number of sequences: 1700			
Details of splitted files:			
Number of chars/seq: 1745			
Number of sequences: 120			
Number of output files: 14			
Number of chars/seq: 1745			
Splitting the input file randomly using Sequence-split-logic.	3 - 4 days	14 nodes on Firefly (parallel)	-Good Evolutionary rate for all the files. -Results are similar for all the output files - Effective Sample Size (ESS) was greater than 350.
Details of input file:			
Number of sequences: 1700			
Details of splitted files:			
Number of chars/seq: 1745			
Number of sequences: 120			
Number of output files: 14			
Number of chars/seq: 1745			
Number of different years: 47			
Number of different locations: 208			

References

[1] P. K. Attaluri, M. C. Christman, Z. Chen and G. Lu, SeqMaT: A sequence manipulation tool for phylogenetic analysis, Bioinformation. 5(9), 2011, pp. 400-401.

[2] A. J. Drummond, R. A. Drummond AJ & Rambaut A (2007) "BEAST: Bayesian evolutionary analysis by sampling trees." BMC Evolutionary Biology 7, 214. *BMC Evolutionary Biology 7* , 214, 2007.

[3] R. C. Edgar, MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research 32(5), 2004, pp. 1792-97.*

[4] W. Hennig, Phylogenetic systematics. University of Illinois Press, Urbana, 1966.

[5] J. P. Huelsenbeck, MRBAYES: "Bayesian inference of phylogeny." *Bioinformatics 17* , 2001, pp. 754-755.

[6] K. Jenny, M. E., Archibald, "Bayesian inference of phylogeny: a non-technical primer." *taxon 52* , 2001, pp. 187-191,

[7] K. Katoh, K. Misawa, K. Kuma and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. ". *Nucleic Acids Res. 30,* 2002, pp. 3059-3066.

[8] D. Maddison, "*Tree of life Web Project,"* http://tolweb.org/tree/.

[9] M. Miller, M. Holder, R.,Vos, P. Midford, R. Liebowitz, L. Chan., P. Hoover, and T. Warnow, "The CIPRES Portals. CIPRES." URL:http://www.phylo.org/sub_sections/portal. Accessed: 2009-08-04. (Archived by WebCite(r) at http://www.webcitation.org/5imQlJeQa)

[10] D. Posada and K. A. Crandall, "Modeltest: testing the model of DNA substitution." Bioinformatics 14 (9): 817-818. . *Bioinformatics 14 (9)* , 1998 , pp. 817-818,

[11] K. de Queiroz and S. Poe, Philosophy and Phylogenetic Inference: A Comparison of Likelihood and Parsimony Methods in the Context of Karl Popper's Writings on Corroboration, Syst. Biol.50(3): 2001, pp. 305–321,

[12] A. Rambaut, *Tracer v1.4*. Retrieved from Tracer: http://beast.bio.ed.ac.uk/Tracer, 2007.

[13] A. Rambaut, A. *Figtree v1.0*. Retrieved from Figtree: http://beast.bio.ed.ac.uk/Figtree, 2006.

[14] B. Rannala and Z. Yang, "Probability distribution of molecular evolutionary trees: a new method of phylogemolecular evolutionary trees: a new method of phylogenetic." *J. Molec. Evol.43* , 1996, pp. 304–311.

[15] B. Robbertse, R. J. Yoder, A. Boyd, J. Reeves, and J. W. Spatatora, "Hal: An automated pipeline for phylogenetic analyses of genomic data", PLoS Curr., Feb. 7, 3: RPN1213, 2011. .Retrieved from

[16] D. L. Swofford, "PAUP. Phylogenetic Analysis Using Parsimony," 2002. Retrieved from http://paup.csit.fsu.edu/about.html.

[17] L. von Bertalanffy, General System Theory, New York, 1981.

[18] D. J. Zwickl, "Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion." PhD dissertation, UT Austin, 2006. Retrieved from http://repositories.lib.utexas.edu/handle/2152/2666?show=full.