PhyloFlow: A Fully Customizable and Automatic Workflow for Phylogenetic Reconstruction

Jorge Álvarez-Jarreta, Gregorio de Miguel Casado and Elvira Mayordomo
Dept. de Informática e Ingeniería de Sistemas (DIIS)
& Instituto de Investigación en Ingeniería de Aragón (I3A),
Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain
Email: {jorgeal, gmiguel, elvira}@unizar.es

***Abstract*—Most phylogeny estimation systems such as SATé or DACTAL use fixed configurations and tools that make them suitable only for solving specific problems. Out of that scope, a hand-made combination of individual tools and methods has to be composed in order to get the desired phylogeny estimation.**
PhyloFlow is a new framework based on a workflow extendable to a wide range of tasks in phylogenetic analysis. This system is specially intended to build large phylogenies, where most of the methods do not provide a solution at all or the computing time required is not affordable. The workflow can scale to different phylogenetic estimation problems, the methods and stages already included can be fully customizable and once the user has set up the system, it will run automatically until the phylogenetic tree is completely estimated.
With the current version we have recreated two different phylogenetic systems: DACTAL and a study case for the human mitochondrial DNA. The first one displays the capabilities of our framework to reproduce the existing systems, in addition with the properties that a parallel system can provide. The second one shows the possibilities of building a real case workflow to estimate a phylogenetic tree for more than 23000 sequences of human mitochondrial DNA (16569 bp on average) applying biological knowledge to the process. Both workflows have been run sequentially and in parallel in a HTC cluster (HTCCondor and DAGMan).
PhyloFlow source code, the datasets and the workflow configurations are available by request to the first author.

***Keywords*—scientific workflow, phylogeny estimation, maximum likelihood, model selection, supertree**

I. Introduction

The construction and analysis of molecular phylogenetics based on proteins, RNA or DNA has become a mature research topic in Bioinformatics. Recent work focuses on specific questions about one or more species [1], [2] or proposes a partial solution within the whole phylogenetic process [3], [4]. In addition, we cannot forget the growing concern about the huge amount of data provided by the so called *next-generation sequencing methods* [5]. A good example is the increasing rate of the number of sequences stored in GenBank, which is doubling approximately every 35 months [6].

Using large datasets as input has disclosed overflow problems when stressing conventional methods and tools, and also when trying to upscale solutions based on them to HPC facilities. With respect to the former, it has being shown that some of them turn to be inaccurate when moving from small data sets to big ones [7]. With respect to the latter, there exists a growing interest in the development of specific frameworks for complex tasks in Bioinformatics capable of dealing with their challenging demands in terms of usability, data-sharing as well as computing [8].

In the context of phylogenetics, several tools and systems have been developed in order to generate a phylogeny composed of several thousands of biological sequences, e.g. SATé, ZaraMit, DACTAL [9], [10], [11], [12]. Based on parallelization, clustering, and divide-and-conquer techniques, these methods aim to achieve both high accuracy and short execution time. However, even though they intend to be generic for a wide range of molecular input data, they disregard useful biological knowledge. This fact challenges significantly not only the accuracy but also the time employed to obtain the phylogenetic tree. In particular, these systems achieve their objective with fixed processes and tools, with no easy possibility of changing any of the tools involved in the process or even parts of the process itself.

In this paper, we present PhyloFlow, a fully customizable and automatic workflow system which aims to provide a framework to build any system aimed to obtain a phylogenetic tree using clustering and divide-and-conquer techniques. The system allows choosing every step of the process as well as the associated tool within the whole phylogeny estimation process. We also present two small integrity tests run both sequentially and in a HTC cluster, and a case study centered in building a human mitochondrial DNA phylogeny. Note that for this case of study we included specific tools that make use of biological knowledge from experts for the human mitochondrial DNA as well as model selection tools valid for most phylogeny estimation processes.

II. Background

Throughout the years, the interest of studying the evolution processes and mutation associated pathogenies by means of phylogenetic trees has been reflected in the wide variety of methods and tools developed. From the point of view of constructing a complete automated phylogeny estimation process with minimum end user interaction, the desirable requirements of the underlying tools to be used in each stage are simple: parameterizable stand-alone applications with no GUI providing well defined output for their exit status.

This way, if we consider each step of the phylogeny estimation process, several tools providing partial solutions can be found: ClustalW [13] or Mafft [14] for the alignment; BioNJ [15] or PhyML [16] for the topology estimation; SeqBoot and

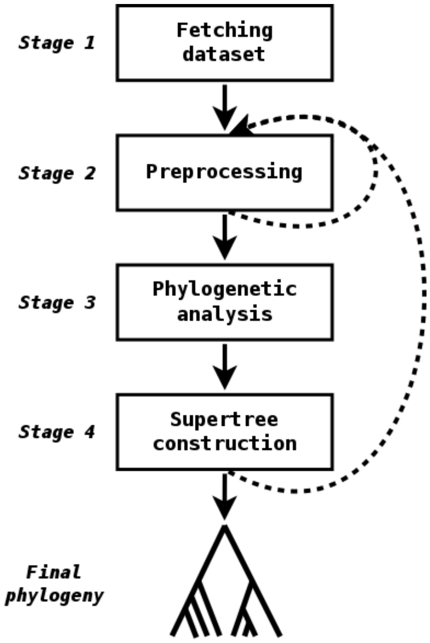

Fig. 1. PhyloFlow framework design with its possible recurrences of the preprocessing stage or iterations of the whole process.

Consense (PHYLIP package [17]) for bootstrapping and consensus, respectively; and SuperFine [18] or MinCut Supertree [19] for supertree estimation, among others.

From the point of view of usability, all of these tools and methods meet (averagely) the desirable requirements to be used in a workflow of tasks. In addition, they have proven their accuracy and reliability for the specific tasks for which they were conceived but not all of them are able to handle large datasets in an efficient way. Therefore, they must be used separately in order to build the final phylogenetic tree.

Few systems have been designed to overcome these problems. ZaraMit [10], [11] was designed to reconstruct the human mitochondrial phylogeny from raw DNA sequences. To our knowledge, this is one of the first examples of an automatic system intended for specific data. More generic and automatic systems for phylogeny estimation are SATé [9] and DACTAL [12]. Both of them use raw sequences as input data and they are able to cope with very large datasets. Nevertheless, from the point of view of usability they can not be easily customized to deploy the "logics and knowledge" provided by biologists for each phylogenetic inference task to be performed. Furthermore, it is necessary to take into account that their accuracy and reliability change depending on the kind of information within the given dataset.

We presented a previous system in 2011 [20] which was able to perform the phylogeny estimation for large datasets using model selection. In this paper we present a framework for building far more flexible and customizable systems.

III. PhyloFlow Framework

A. Design

PhyloFlow has been designed taking into account two main objectives: i) to provide a fully automatic and customizable framework able to build many different systems; and ii) to ease the addition of new tools (the existing ones not already included or those who will come out in the future). The framework is composed by four stages (Fig. 1) created with a *black box methodology* to provide modularity. Therefore, the user can select any configuration of any stage without concerning about the interaction between or within modules, providing a high robustness in the ad-hoc system built. These four stages correspond to the four most common processes that are usually made to obtain a phylogeny estimation:

1) Fetching dataset: Most studies done in phylogenetics are usually based on using existing systems configurations for new datasets [21], or testing different parameters or methods with well-known datasets and make the subsequent comparison with previous results [7]. Taking that into account, we have made a fully automatic fetching of the desired dataset. The system will first access to the public database selected with the query provided (or using a preconfigured query tagged with the data type it is intended for) or copy a local dataset in case the user has already one. The process would be the same if more than one dataset were fetched.

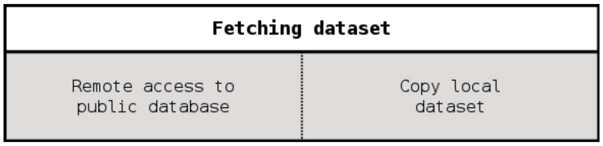

Fig. 2. Stage 1: Fetching datasets by remote access to a public database or by accessing them from a local resource.

2) Preprocessing: Most of the methods and tools aimed to build a phylogenetic tree require an alignment as input. Therefore, we will need at least to do an alignment of the fetched sequences in order to be able to get through the next stage successfully. On the other hand, there exist many methods that get more accurate phylogenies using divide-and-conquer strategies or adding biological knowledge to the process. In the first category we have systems like DACTAL[12], which uses the padded-Recursive-DCM3 decomposition (PRD) in order to get overlapping subsets of the input set of sequences. In the second category there are systems like the one we published in 2011[20], that applies a group and gene decomposition of the input dataset (2D-decomposition). In both cases, we get smaller datasets that can be handled individually in the following stages. Hence, the design of this stage has been divided in 3 different configurations: an alignment, a row division (by overlapping subsets or haplogrouping) or a column division (by genes). Some of the processes involved in this stage do not return aligned datasets as output, therefore we allowed to apply this stage recursively. Furthermore, taking into account those tools which are able to build phylogenies from unaligned sequences [22], we included the possibility of skipping this stage in the configuration process.

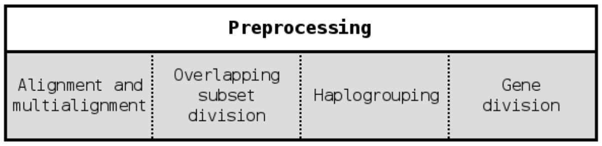

Fig. 3. Stage 2: Preprocessing of each input dataset by making a multiple sequence alignment, an overlapping subset decomposition, a haplogroup classification of the sequences or a gene division splitting the sequences into smaller fragments.

3) Phylogenetic analysis: There are a lot of methods that can estimate a phylogenetic tree, e.g. parsimony, maximum likelihood or Bayesian inference. We considered all these possibilities besides their parameter selection, which is as important as the methodology itself to improve the accuracy of the resultant phylogeny. We paid special attention to the model selection of the maximum likelihood method, which implicates a large increase of the number of processes needed to achieve it. We also considered that many of the methods mentioned above lack of statistical robustness and then an extension in the phylogenetic analysis is required. In biology, this problem is usually covered by a bootstrapping [24] of the input dataset, demanding a consensus process later on. Therefore, a bootstrapping analysis and a consensus process were also included in this stage.

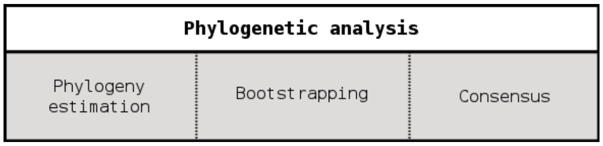

Fig. 4. Stage 3: This is the core stage of the framework, were the phylogeny is estimated for each input dataset. In addition, a bootstrapping phase can be included in the workflow process to add statistical robustness. Finally, a consensus tree can be computed to join related phylogenies.

4) Supertree construction: The last stage combines the phylogenetic trees obtained in the previous one, building the final supertree. The configuration process will allow the user to choose only viable methods within the processes selected so far, e.g., the system will not allow to run a merging of phylogenies with common leaves if the preprocessing stage has not obtained overlapping subsets at any point. In this case we also noticed the possibility of getting one single phylogeny as result of the previous stage, allowing the system to avoid this last phase.

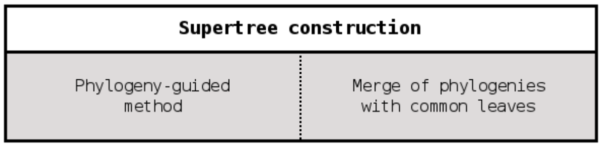

Fig. 5. Stage 4: The phylogenies obtained in the previous stage are joined in a single phylogeny with a supertree construction method.

B. Implementation

As mentioned before, PhyloFlow framework has been designed to produce workflow structures. These kind of design has several advantages. Firstly, we can make it work automatically once it has been configured, requiring the user interaction only in the configuration phase. Secondly, it allows changing easily the content of the different stages, even removing some of them if needed. As we have claimed before, customization is one of the main goals of our system, making it suitable for almost any kind of desired phylogenetic tree estimation. Thirdly, in order to facilitate the transition from the design to the implementation, we selected DAGMan [23], a meta-scheduler of HTCondor [24]. DAGMan allows to represent a system with direct acyclic graphs, which fits perfectly with the framework we present. Moreover, using the High Throughput technology of Condor, we can parallelize massive amounts of processes that can run in parallel in each phase, reducing considerably the time taken in each module. This technology in combination with the preprocessing stage, applied to the data decomposition process, allows for handling very large datasets with high accuracy. This is based on the same concepts as systems like DACTAL [12] or SATé [9].

Even though our design is represented by a directed acyclic graph, the accuracy of many systems is based on iterating over the main execution flow, using the results of the previous iteration as input of the new one (until some conditions are reached). In PhyloFlow this has been added using a global script that allows the user to iterate over the basic workflow presented. We also included one backup script right after each stage so it can recover from errors and shutdowns of the hardware system without the need of starting again from the beginning. The backups also provide a huge feedback of what has been done in each step, in case the user wants to reproduce any experiment or check them by other means.

All the scripts made within the system were programmed in Python and are compatible with both version 2.7 and version 3.3.

C. PhyloFlow 1.0 (July-2014 version)

PhyloFlow has several tools available with several possible configurations. On the first stage, where the dataset is fetched, there exist two options: download the dataset form GenBank or get it from a local database. For the first case we provided the option of fetching all human mitochondrial DNA sequences using the following query:
"Homo Sapiens"[Organism] AND mitochondrion[All Fields] AND 16000:16900[SLEN] NOT pseudogene[All Fields]
and all the rest data type IDs will be fetched in the local database.

On the second stage, we have three different choices: PRD (from DACTAL), group decomposition, gene decomposition, or alignment. For the first choice we copied the same binary used by DACTAL and we left the same default parameters that can be changed by the user. The group decomposition splits the input dataset into disjoint subsets, regarding the whole sequences, given certain biological conditions. In the case of the hmtDNA, we use the haplogroup information to make it. The gene decomposition uses the gene information to divide the input dataset into fully joint subsets, in terms of accessions of the sequences, and disjoint subsets in terms of the sequence strings. These two processes can be combined to get the 2D-decomposition mentioned in the design section, which is a common procedure when we handle hmtDNA sequences. For the alignment option we included two common and well known alignment tools: ClustalW [13] and Mafft [14]. ClustalW was added with two settings: DNA or protein. The system will automatically choose the one that corresponds to the data type selected. Due to the fact that Mafft has several parameters and options, we tried to make our system both simple and versatile at the same time, so we decided to include three configurations: parttree (the fastest choice), auto and l-ins-i (local pair alignment).

The third stage focuses on the phylogeny estimation. We took into account for this first version tools only for the maximum likelihood method because it is the most used nowadays given the impractical time cost of the Bayesian inference methods for big datasets [25] and the accuracy and statistical downsides of parsimony and some distance-based methods. Thus, we included FastTree [26] and PhyML [16] as the software tools available to build the phylogenetic tree. FastTree has been configured to get the GTR+CAT model when we are dealing with DNA and the JTT+CAT model when we are handling protein sequences. In both cases the *log* option is selected to create a log file to retrieve the score of the phylogenetic tree inferred using the former biological model. For PhyML we include the evolution model test composed by 88 models (22 basic models plus invariants, gamma distribution and both) for DNA sequences[20]. In the case of protein sequences, we test the same evolution models that were included in ProtTest [27]. For the bootstrapping generation and the consensus method we added seqboot and consense, both methods included in the Phylogenetic Inference package (PHYLIP [17]). We made two small modifications to these tools to make them able to accept a second parameter: the path of the output file. Originally, they create the output file with the bootstraps or the consensus tree in the same path from where the executable is called, which is not a desirable behavior in our system.

The last stage builds the supertree from the phylogenetic trees we estimated in the previous phase. We included the SuperFine [18] method and the same supertree method we included in the system of 2011 which we will call Profile. SuperFine is run in its default mode and in the configuration script we ensure that the phylogenetic trees provided as input have certain overlapping of the leaves (a must condition for SuperFine). Profile searches in the system for a profile or skeleton tree of the data type selected and replaces each leaf for its corresponding phylogenetic tree. E.g., when building the phylogenetic tree for the human mitochondrial DNA, we selected the 35 main haplogroups and we took the tree of those haplogroups from phylotree.org [28]. Then, Profile replaces each haplogroup ID at the leaves for its corresponding phylogenetic tree obtained in the previous stage.

IV. Results & Discussion

We made two different systems and we run both sequentially and in a HTC cluster to test the integrity between stages and the processes within them. In addition, we proposed a case study centered in building a human mitochondrial DNA phylogeny to show some of the possibilities of PhyloFlow in its current version. We also present the times taken for each case as a measure of the speedup of our framework.

1st test: human mitochondrial DNA system

The first workflow shown in Fig. 6. is a reproduction of the system we presented in 2011 [20] made to estimate the phylogenetic tree for all the sequences stored in GenBank for the human mitochondrial DNA (hmtDNA). Here we used a simplification of the whole system that will be presented later on.

We used one dataset of 27 well studied sequences of hmtDNA (16569 bp on average), 18 from haplogroup N and 9 from haplogroup T. We wanted to have a high difference in the number of sequences between this two groups in an attempt to reproduce the existing unbalance between haplogroups in the real case. After the system recovered the whole dataset, it was divided in two different sets of at most 15 sequences. The distribution of the sequences in the two sets is reproducible as long as the order of the sequences in the input dataset remains the same.

The preprocessing stage involves two processes which use specific biological knowledge to determine the haplogroup and the starting and ending site of each gene in a sequence of hmtDNA. Then, the two sets were processed individually to decide the haplogroup of each sequence and two new datasets, one for each haplogroup, were created. Afterwards, these two new datasets were processed again to extract two of the genes of the hmtDNA: NADH dehydrogenase 5 (ND5) and tRNA Threonine (Thr). These two genes were selected for their difference in length: 1812 bp and 66 bp, respectively. At the end of the second stage we got four datasets, one for each possible combination of haplogroup and gene.

In the phylogenetic analysis stage we included two main processes: model selection for the phylogeny estimation and bootstrapping to add statistical robustness. For the phylogeny estimation adding model selection we used PhyML and we selected JC and GTR as the two models to evaluate, the former with the least number of free parameters and the latest with the highest number of free parameters (of the 88 models in [20]). Thus, the system evaluated and selected the best model for each dataset from the previous stage. Later, two bootstraps were generated with Seqboot, from PHYLIP package, and then their phylogenetic tree was estimated with PhyML using its corresponding best model. Next, a bootstrap consensus was made with Consense, from PHYLIP package, for all the phylogenies of the bootstraps of the same dataset, getting a single consensus phylogeny for each pair of haplogroup and gene. Genes are closely related to each other in an evolutionary point of view for hmtDNA, so a consensus process can be placed again to join the phylogenetic information of all the genes of the same haplogroup. Hence, we got one phylogenetic tree for each haplogroup dataset.

Haplogroups are far enough from each other to include a final supertree construction in the system. Therefore, the final stage was made with a phylogeny-guided construction process with a synthetic and simple haplogroup tree, represented in Newick format by “*(N,T);*”.

For the whole test presented the system took 25 minutes in the sequential run and 11 minutes in the HTC cluster, which implies a speedup of 2,18. There could be some improvements in the sequential run that might slightly change this result. We could skip all the inner scripts that write DAGMan files

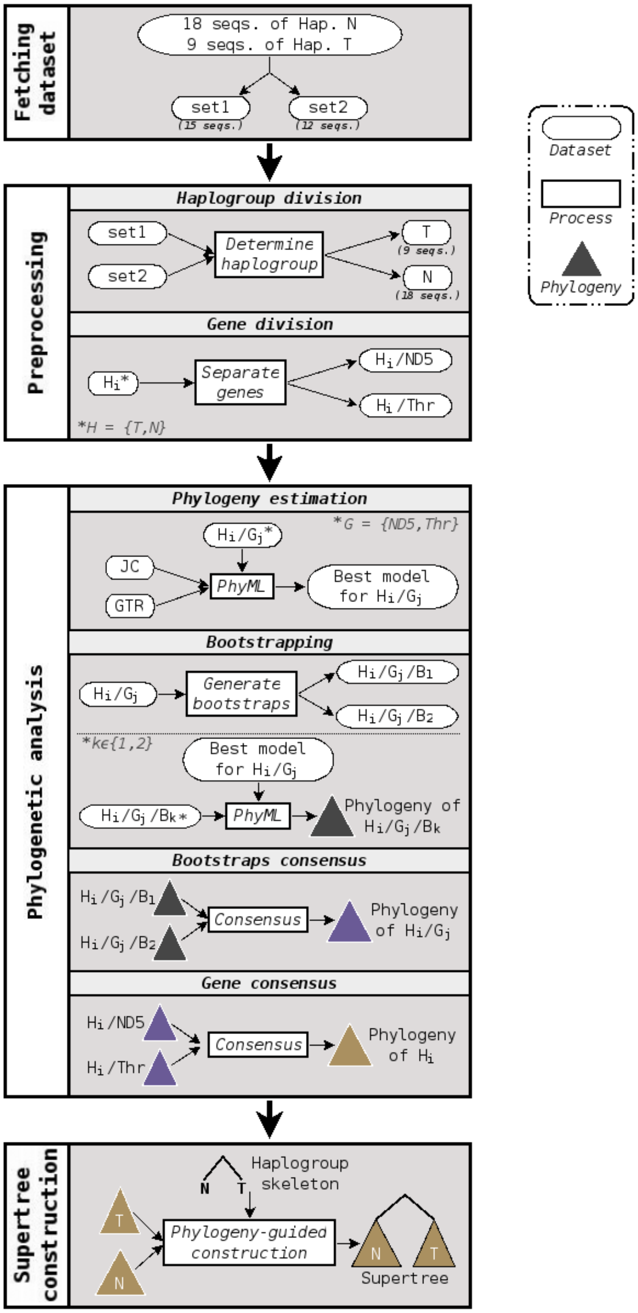

Fig. 6. Workflow of the system developed to test PhyloFlow for an example of 27 sequences of human mtDNA.

for the scheduler of the HTC cluster, as well as the first dataset division in the first stage, given that it is intended to be parallelized using the HTC cluster along the following stages. On the other hand, bigger datasets would improve latency times of data transmission in the cluster between nodes, and reduce penalty times added to those processes that took less time than the minimum required by the scheduler.

2^{nd} test: DACTAL-like system

The second workflow is a recreation of one iteration of DACTAL [12], as shown in Fig. 7. We wanted to reproduce it as close as possible so we were able to prove that PhyloFlow is capable of recreate most of the workflows we were able to find for the existing phylogeny estimation systems. We included a bootstrapping step in the phylogenetic analysis stage to make it a complete integrity test of our framework.

We used 50 sequences from a 1000-taxon simulated dataset studied in [9]. The dataset name is 1000M1 and we selected 25 sequences from the replica 0 and 25 sequences from the replica 1. The first stage just copies these two datasets from the local path.

To recreate the preprocessing process of DACTAL, we applied twice the second stage with different configurations. The first one aims to get the overlapping subsets for each input dataset. To do so, the system applies Mafft in its parttree setting (*mafft –parttree –retree 2 –partsize 1000*) to get a whole alignment of each dataset and set it as input of FastTree under GTRCAT to get a phylogenetic tree. This phylogeny is used to achieve the dataset division technique using a padded-Recursive-DCM3 decomposition (PRD) obtaining four overlapped subsets for each phylogeny composed by at most 16 sequences with at most 3 overlapping sequences among subsets. The output of the PRD process are the names of the sequences of each subset, so the final output of this first part is eight subsets of unaligned sequences. Therefore, the second part was intended to generate the alignments needed in the third stage. The system run Mafft in its L-INS-i setting (*mafft –localpair –maxiterate 1000*) to produce one alignment per subset.

In the phylogenetic analysis the system estimated the phylogenetic tree of each subset running FastTree with GTRCAT. We included a bootstrapping process to make a complete integrity test and also to show how easy can be to change or add new processes to an already existing system with our framework. We know there is no need on adding statistical robustness to DACTAL but it could be a different process or in a different workflow. Like in the previous test, the system run Seqboot to generate two bootstraps for each subset, and then it estimated again the phylogeny for each bootstrap executing FastTree with GTRCAT. Finally, it got the consensus tree for all the bootstraps of each subset with Consense.

At the end of the workflow, we selected SuperFine in the last stage as the tool to build the supertree for each initial set.

The complete execution of this workflow took 3 minutes in the sequential run and 8 minutes in the HTC cluster, implying a speedup of 0,375. The reduction in the speedup compared with the previous test is mainly based in the change of the tool for the phylogeny estimation: PhyML is much slower than FastTree for the same input, hence the time obtained in the parallel execution is higher for the last case due to the time penalties applied by the cluster. On the other hand, the same improvements presented in the previous test can be applied in this analysis.

Study case of human mitochondrial DNA phylogeny

In the first test we have presented a reduced and simple version of the study case of human mitochondrial DNA phylogeny reconstruction. Although most of the workflow configuration and the different processes remain the same, the changes in the first stage and in the values of the different parameters

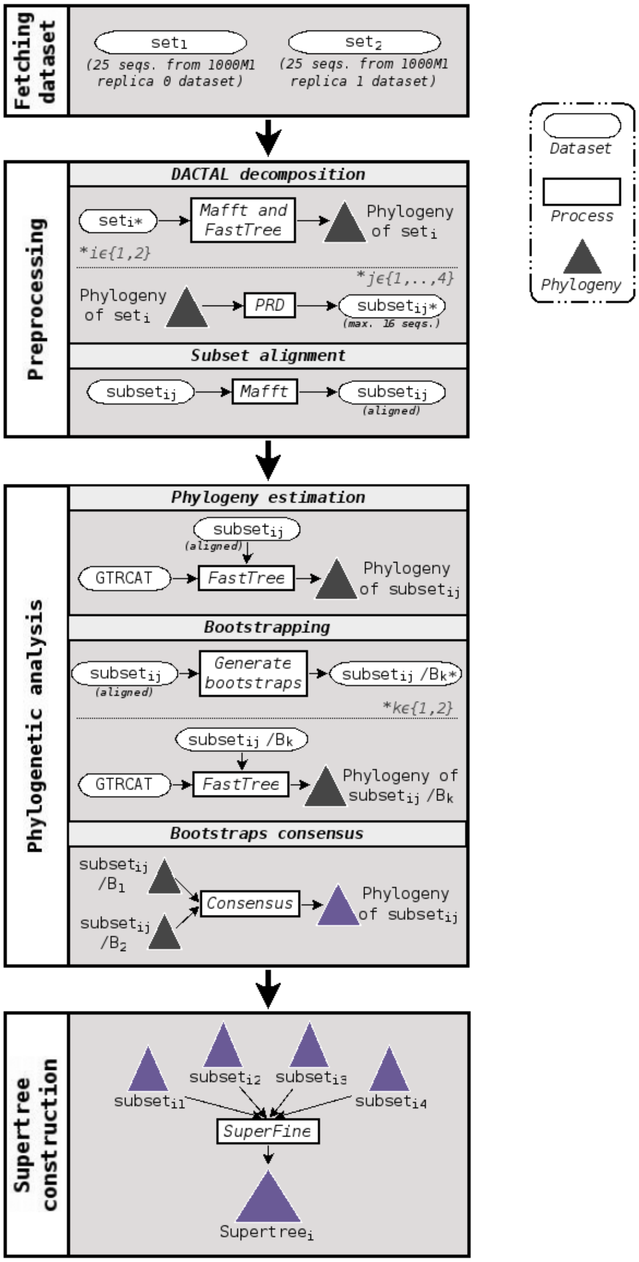

Fig. 7. Workflow of the system developed to test PhyloFlow for an example of 50 synthetic sequences in a DACTAL-like phylogeny estimation.

involved make a huge difference in the time cost of the whole system.

One of the most important steps in a real study case is to work with the most updated information. To do so, PhyloFlow has the possibility to download from GenBank all the sequences that belongs to certain biological type. In our case, downloading all the sequences of hmtDNA in FASTA format from GenBank in batches of 100 sequences (amount recommended by NCBI-Entrez) took 835 seconds (up to 14 minutes) for 23644 sequences (April, 6th of 2014), using the same query shown in the previous section. Afterwards, the first stage also includes a set division of the whole dataset into sets of up to 200 sequences. The order of these sequences may change from one time to another given the internal updates of GenBank database.

The real values of the parameters involved in the preprocessing stage involve a big increase in its time cost. Instead of 2 haplogroups, we choosed the 35 main haplogroups that can be found in *phylotree.org*[28], a website of reference for the haplogroup classification of hmtDNA. Processing each set took 66 minutes on average, making it a total of 10 days and 8 hours in the sequential case, and up to 180 minutes in the HTC cluster. After the haplogroup division, we found the first big disadvantage in these kind of systems: there exist haplogroups with less than 10 sequences (e.g. haplogroup O with 3 sequences) and others with more than 1000 sequences (e.g. haplogroup M with more than 1800 sequences), making it a really unbalanced scenario for the following processes. Then, the gene division was applied to each haplogroup set, generating 38 subsets for each input, one per each of the 37 existing genes in the hmtDNA plus the control region (also known as HVS). This second step took 87 minutes for the sequential case and 25 minutes in the HTC cluster. At the end of this stage we ended up with 1330 datasets.

The phylogenetic analysis stage we included the same 88 evolution models used in [20] for the model evaluation step and we selected the number of bootstraps at 200, a good value in the recommended range of 100 to 1000. We found several problems that made not possible to run this stage neither in sequential nor in the HTC cluster. The first one is the amount of processes that will be created in each step: for the model evaluation with PhyML we would have 117040 tasks, and for the bootstrapping step we would generate 266000 processes. The second problem is the total time needed for the whole stage. We have been measuring PhyML with some datasets of the total of 1330, and for the mean case, which implies the average gene in length and the average haplogroup in number of sequences for the dataset, and the measure of two evolution models, JC as the simplest and GTR+I+G as the most complex, it took 5175 minutes (more than 3 days and 12 hours). If we apply this time to the sum of the amount of processes exposed above, the third stage might take more than 3771 years in a sequential run. In addition, PhyML does not work with datasets of more than 4000 sequences, so some haplogroups need another step of preprocessing before the third stage.

With these results we want to show the problems that the systems like these one deals with when we use real datasets and we want to make and exhaustive phylogenetic analysis to get the best phylogenetic tree possible.

V. Conclusion

We have presented PhyloFlow, a workflow system that provides a framework to build fully customizable phylogeny estimation systems. Once the configuration process is finished, the new ad-hoc system will automatically build a phylogenetic tree from the selected input data. We have also displayed two possible scenarios that our framework was capable to reproduce in its current version. The DACTAL-like workflow has demonstrated the capability of PhyloFlow to reproduce an existing phylogeny estimation system, and the human mitochondrial DNA workflow has shown an application of more specific tools and processes, like model selection, that incorporates biological knowledge to the phylogeny estimation process. All the systems created can handle very large datasets given the modularity design and cluster implementation of our framework.

For future improvements we aim to include new tools and configurations for the already incorporated processes (e.g. fetching from different public databases), as well as new processes like alignment-free phylogeny constructions, homology detection or supermatrix constructions.

The first release version of the framework will be available at http://www.zaramit.org/phyloflow.

Acknowledgment

We would like to thank S. Mirarab, N. Nguyen, M.S. Bayzid and T. Warnow from the University of Texas at Austin for their invaluable help in the comprehension of DACTAL and in the reuse of its inner binaries. We also want to thank to the Instituto de Investigación en Ingeniería de Aragón (I3A) for they support with cluster Hermes.

Funding: The Spanish Ministry of Science and Innovation (Project TIN2011-27479-C04-01) and the Spanish Ministry of Education which provided J. Álvarez-Jarreta with a grant [FPU AP2010-1058] to carry out this investigation.

References

[1] J. Regier, J. Shultz, A. Zwick, A. Hussey, B. Ball, R. Wetzer, J. Martin, and C. Cunningham, "Arthropod relationships revealed by phylogenomic analysis of nuclear protein-coding sequences," *Nature*, vol. 463, pp. 1079–1083, Feb. 2010.

[2] J. James T. Harper, E. Waanders, and P. Keeling, "On the monophyly of chromalveolates using a six-protein phylogeny of eukaryotes," *Systematic and Evolutionary Microbiology*, vol. 55, no. 1, pp. 487–496, 2005.

[3] H. Schmidt, K. Strimmer, M. Vingron, and A. von Haeseler, "TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing," *Bioinformatics*, vol. 18, no. 3, pp. 502–504, Mar. 2002.

[4] Z. Yang, "PAML 4: Phylogenetic Analysis by Maximum Likelihood," *Molecular Biology and Evolution*, vol. 24, no. 8, pp. 1586–1591, 2007.

[5] A. Grada and K. Weinbrecht, "Next-Generation Sequencing: Methodology and Application," *Journal of Investigative Dermatology*, vol. 133, no. e11, 2013.

[6] D. Benson, I. Karsch-Mizrachi, D. Lipman, and E. Ostell, J.and Sayers, "GenBank," *Nucleic Acids Research*, vol. 38, pp. 46–51, 2010.

[7] K. Liu, C. Linder, and T. Warnow, "RAxML and FastTree: Comparing Two Methods for Large-Scale Maximum Likelihood Phylogeny Estimation," *PLoS One*, vol. 6, no. 11, p. e27731, Nov. 2011.

[8] B. Liu, R. Madduri, B. Sotomayor, K. Chard, L. Lacinski, U. Dave, J. Li, C. Liu, and I. Foster, "Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses," *Journal of Biomedical Informatics*, vol. 49, pp. 119–133, 2014.

[9] K. Liu, S. Raghavan, S. Nelesen, C. Linder, and T. Warnow, "Rapid and Accurate Large-Scale Coestimation of Sequence Alignments and Phylogenetic Trees," *Science*, vol. 324, p. 1561, 2009.

[10] R. Blanco and E. Mayordomo, "ZARAMIT: a system for the evolutionary study of human mitochondrial DNA," in *IWANN 2009, Part II*, ser. Lecture Notes in Computer Science, vol. 5518, 2009, pp. 1139–1142.

[11] R. Blanco, E. Mayordomo, J. Montoya, and E. Ruiz-Pesini, "Rebooting the human mitochondrial phylogeny: an automated and scalable methodology with expert knowledge," *BMC Bioinformatics*, vol. 12, p. 174, 2011.

[12] S. Nelesen, K. Liu, L.-S. Wang, K., C. Linder, and T. Warnow, "DACTAL: divide-and-conquer trees (almost) without alignments," *Bioinformatics*, vol. 28, pp. i274–i282, 2012.

[13] J. Thompson, D. Higgins, and T. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, no. 22, pp. 4673–4680, Nov. 1994.

[14] 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 Research*, vol. 30, no. 14, pp. 3059–3066, 2002.

[15] O. Gascuel, "BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data," *Molecular Biology and Evolution*, vol. 14, pp. 685–695, 1997.

[16] S. Guindon and O. Gascuel, "A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood," *Systematic biology*, vol. 52, no. 5, pp. 696–704, 2003.

[17] J. Felsenstein, "Phylogeny inference package (PHYLIP)," university of Washington, Seattle.

[18] M. Swenson, R. Suri, C. Linder, and T. Warnow, "SuperFine: fast and accurate supertree estimation," *Systems Biology*, vol. 61, no. 2, pp. 214–227, Mar. 2012.

[19] C. Semple and M. Steel, "A supertree method for rooted trees," *Discrete Applied Mathematics*, vol. 105, no. 1–3, pp. 147–158, 2000.

[20] J. Álvarez, R. Blanco, and E. Mayordomo, "Workflows with model selection: A multilocus approach to phylogenetic analysis," in *5th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2011)*, ser. Advances in Intelligent and Soft Computing, vol. 93, 2011, pp. 39–47.

[21] U. Purkhold, M. Wagner, G. Timmermann, A. Pommerening-Röser, and H. Koops, "16s rrna and amoa-based phylogeny of 12 novel betaproteobacterial ammonia-oxidizing isolates: extension of the dataset and proposal of a new lineage within the nitrosomonads," *International Journal of Systematic and Evolutionary Microbiology*, vol. 53, pp. 1485–1494, 2003.

[22] C. Daskalakis and S. Roch, "Alignment-free phylogenetic reconstruction," in *RECOMB 2010*, ser. Lecture Notes in Computer Science, vol. 6044. Springer, Berlin/Heidelberg, 2010, pp. 123–137.

[23] P. Couvares, T. Kosar, A. Roy, J. Weber, and K. Wenger, "Workflow in condor," in *In Workflows for e-Science*. Springer Press, 2007.

[24] J. Basney, M. Livny, and T. Tannenbaum, "High throughput computing with condor," *HPCU news*, vol. 1, no. 2, June 1997.

[25] M. Suchard and B. Redelings, "BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny," *Bioinformatics*, vol. 22, pp. 2047–2048, 2006.

[26] M. Price, P. Dehal, and A. Arkin, "FastTree: Computing Large Minimum-Evolution Trees with Profiles instead of a Distance Matrix," *Molecular Biology and Evolution*, vol. 26, pp. 1641–1650, 2009.

[27] F. Abascal, R. Zardoya, and D. Posada, "Prottest: selection of best-fit models of protein evolution," *Bioinformatics*, vol. 21, no. 9, pp. 2104–2105, 2005.

[28] M. van Oven and M. Kayser, "Updated comprehensive phylogenetic tree of global human mitochondrial DNA variation," *Human Mutation*, vol. 30, no. 2, pp. E386–E394, 2009. [Online]. Available: http://www.phylotree.org