Modeling and Evaluating the Phylogenetic Workflow

Phylogenetic analysis refers to constructing phylogenetic trees to confirm or refute a specific evolutionary hypothesis. Nowadays, phylogeny (and now phylogenomics) is also required to explore other new and complex bioinformatics fields as meta-transcriptomics, metagenomics, or proteomics.

Scientific workflows that construct phylogenetic trees, such as SciPhy, have been proposed to manage the complex interplay of multiple bioinformatics applications.

In terms of complexity and time requirements, this process is considered computing-intensive and demands monitoring High Performance Computing (HPC) executions.

In this scenario, performing phylogenetic experiments can be exhaustive due to the following issues: (i) there is a large number of programs/algorithms (applications) to be tested/compared and (ii) there is a huge quantity of biological data available in biological databases to be processed. Then, parallelism, distribution and HPC environments and technologies are needed.

Let's try out a couple of different phylogenetic scenarios! We'll be using programs that construct phylogenetic trees: RAxML, PhyML, MrBayes, PHYLIP, GARLI, or Weighbor. Also a variety of programs associated with previous processes (activities) in SciPhy is required (see **Table 1**).

For this practice, the following programs covered the four (4) activities in SciPhy: MAFFT for constructing multiple sequences alignments (MSA), ReadSeq for format conversion, ModelGenerator for choosing the evolutionary model and parameters, and RAxML for constructing phylogenetic trees. For installing the programs, please the following instructions. We considered these installations in an Ubuntu Linux operating system. However, they need to be adequate to Linux Cent OS 5 (64 or 32 bits) depends on the environment where they will be executed (desktop, clusters, grid, or clouds).

Table 1. Bioinformatics tools for phylogeny

Applications	Web link
Align-m	http://bioinformatics.vub.ac.be/software/software.html
ClustalW2	http://clustal.org/clustal2
Kalign2	http://msa.sbc.su.se/cgi-bin/msa.cgi
MAFFT	http://mafft.cbrc.jp/alignment/software
ModelGenerator	http://mcinerneylab.com/software/modelgenerator
MUSCLE	http://www.drive5.com/muscle
PHYLIP	http://evolution.genetics.washington.edu/phylip.html
ProbCons	http://probcons.stanford.edu/
RAxML	http://sco.h-its.org/exelixis/web/software/raxml/index.html
ReadSeq	https://sourceforge.net/projects/readseq/
T-Coffee	http://tcoffee.org

I. Modeling the experiment

a. Steps of the experiment

- Constructing MSA with "**MAFFT**, ClustalW, Kalign2, MAFFT, MUSCLE, ProbCons, T-Cofee"
- MSA format conversion (from MAFFT to PHYLIP format) with "**ReadSeq**"
- Evolutional model election with "**ModelGenerator**, JModelTest, ProTest"
- Constructing trees with "**RAxML**, ExaML, PhyML, MrBayes, PHYLIP, GARLI, Weighbor"

b. Checking required software for installation

There are two forms to install bioinformatics tools: by command line or download/installation/compilation.

i. Command line: Programs are installed with the command "**sudo apt-get install program**". Then, you need to store all information of programs *e.g.* version. Try:
   > sudo apt-get install mafft
   > sudo apt-get install raxml
   Obs. ReadSeq and ModelGenerator are ".jar" applications, which are available and ready to be executed.

ii. But, if any problem is detected (*e.g.* incompatibility to the environment or no available version by command line installation), programs can be directly downloaded from the Web.

- **RAxML: for link see Table 1**
  - RAxML, last version available
  - Uncompress (tar -xzvf) the directory, look the README document for the **requirements** and follow **instructions** for compiling/installing programs.
  - Add the RAxML PATH at the ".bashrc"
    - export PATH=$PATH:/path-dirRAxML
    - execute source .bashrc
  - Store information of versions for RAxML and the required libraries.
- **MAFFT: for link see Table 1**
  - MAFFT, last version available
  - Uncompress (tar -xzvf) the directory, look the README document for the **requirements** and follow **instructions** for compiling/installing programs.
  - If required, add the MAFFT PATH at the ".bashrc"
    - export PATH=$PATH:/path-dirMAFFT
    - execute source .bashrc
  - Store information of versions for MAFFT and the required libraries.

c. The biological dataset

We'll be working on a small genomic data set of cysteine protease genes.

d. Executing the programs by command line

Command lines for programs (activities) used in SciPhy are presented in Table 2.

e. SciPhy: a scientific workflow for phylogenetic analysis

SciPhy workflow [1] is composed of four main activities that are: MSA construction, MSA format conversion, a search for the best evolutionary model, and construction of the phylogenetic trees. They respectively execute the following bioinformatics applications: an MAFFT [2], ReadSeq [3], ModelGenerator [4] and RAxML [5].

Figure 1 presents the SciPhy workflow. The first activity of SciPhy (MSA construction) constructs an individual MSA using available MSA programs – *e.g.* ClustalW, Kalign, MAFFT, Muscle, or ProbCons. Each MSA program receives a multi-fasta file as an input from a set of given multi-fasta files, which then produces an MSA as an output. At this point, SciPhy obtains several individual MSA that have been produced by the elected MSA program. In the second activity, each MSA is converted to the PHYLIP format [6] using ReadSeq and then tested in the third activity to find the best evolutionary model using ModelGenerator. Both the individual MSA and the evolutionary model are used in the fourth activity to generate the phylogenetic trees using RAxML with configurable bootstrap replicate values. Consequently, multiple trees are obtained for each one of the MSA programs that were selected.

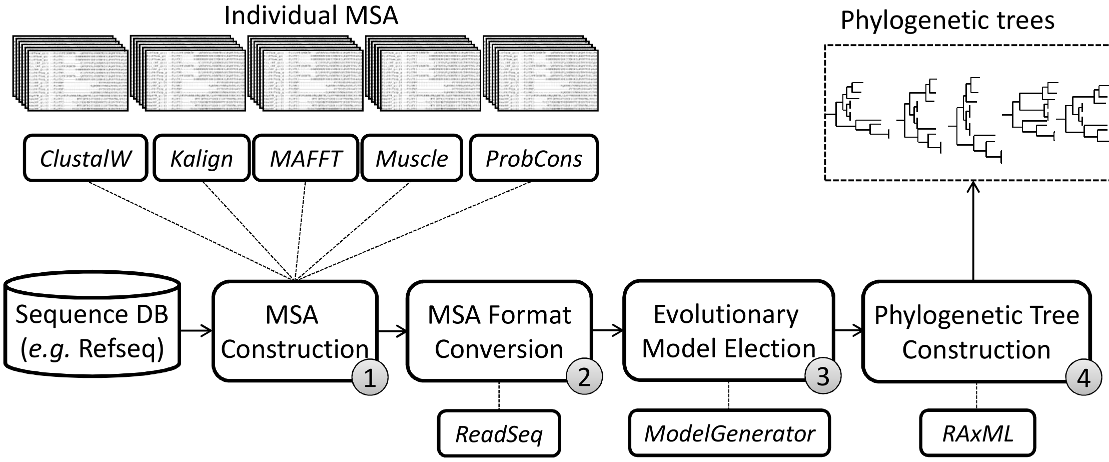

Figure 1. Conceptual view of SciPhy adapted from Ocaña *et al.* [1]

II. Designing the workflow for assembly with SciCumulus

The directory execution is called “SciPhy” (for RAxML). Each directory contains input/output files/directories and SciCumulus execution files.

a. Before the execution: Files and directories:

- **Files**

- *CModel.xml*: invokes SciCumulus for storing the configuration workflows
- *EModel.xml*: invokes SciCumulus for execute thr workflow experiments

- *machine.conf (only for cluster and clouds)*: shows the machine name, port and rank of the virtual machine(s) (VMs) that is(are) instantiated.
- *exp/Input.dataset*: contain the input file name, localized in the input directory. Note that the prefix ".fasta" (from the originally files downloaded) was deleted to allow the SWfMS manipulate/change all name adding or deleting the prefix for each activity.

- **Directories**

- *input*: contains the input files
- *output*: is empty and will store all resultant output files
- *template_activity1*, *template_activity2*, (…): contain scripts and files used to execute activities
- *exp*: directory where will be created all output directories for each task and activities executions
- *temp*: output directory that centralize the results pointed by scientists

b. After the execution: Files and directories:

- **Files**

- *output_activity1.txt*, *output_activity1.txt,:* are generated after the corresponding activity executions

- **Directories**

- *activity1*, *activity2*,: contain resultant files generated for each one of the activities.

c. Workflows execution

- In each workflow directory, invoke the following command lines:
  - ./exe1_cleanDir.txt
    - (to clean result directories)
  - ./exe2_ChironSetup.txt
    - (to insert/delete/update xml files to the SciCumulus database *e.g.* if there are changes)
  - ./exe3_ChironExe.txt
    - To execute the workflow
- **Note that before a new execution you NEED to update EModel.xml, by updating the number of execution:**
  - exectag="sciphy-execution_1"
  - exectag="sciphy-execution_2"
  - exectag="sciphy-execution_3"
  - …

Table 2. Command lines of programs used in the SciPhy activities

Program	**Coommand Line**
MAFFT	mafft **NAME.fasta** > **NAME.mafft**
ReadSeq	java -jar readseq.jar -all -f=12 **NAME.mafft** -o **NAME.phylip**
ModelGenerator	java -jar modelgenerator.jar **NAME.phylip** 6 > **NAME.mg**
RAxML	**1.- Estimating a Single Maximum-Likelihood Tree from Protein Sequences** raxmlHPC -s **NAME.phylip** -n phylip_raxml_tree1.singleTree -c 4 -f d -m **model** **2.- Estimating a Set of Non-Parametric Bootstrap Trees** raxmlHPC -s **NAME.phylip** -n phylip_tree2.raxml -c 4 -f d -m **model** -b 234534251 -N **bootstrap** **3.- Projecting Bootstrap Confidence Values onto ML Tree** raxmlHPC -f b -m **model** -c 4 -s **NAME.phylip** + -z **RAxML_bootstrap.phylip_tree2.raxml** –t **RAxML_bestTree.phylip_raxml_tree1.singleTree** -n **phylip_tree3.BS_TREE**

III. References

[1] K. A. C. S. Ocaña, D. de Oliveira, E. Ogasawara, A. M. R. Dávila, A. A. B. Lima, and M. Mattoso, “SciPhy: A Cloud-Based Workflow for Phylogenetic Analysis of Drug Targets in Protozoan Genomes,” in *Advances in Bioinformatics and Computational Biology*, 2011, pp. 66–70.

[2] K. Katoh and D. M. Standley, “MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability,” *Molecular Biology and Evolution*, vol. 30, no. 4, pp. 772–780, Apr. 2013.

[3] D. Gilbert, “Sequence file format conversion with command-line readseq,” *Curr Protoc Bioinformatics*, vol. Appendix 1, p. Appendix 1E, Feb. 2003.

[4] T. M. Keane, C. J. Creevey, M. M. Pentony, T. J. Naughton, and J. O. Mclnerney, “Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified,” *BMC Evol. Biol*, vol. 6, p. 29, 2006.

[5] A. Stamatakis, “RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies,” *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, May 2014.

[6] J. Felsenstein, “PHYLIP - Phylogeny Inference Package (Version 3.2),” *Cladistics*, vol. 5, pp. 164–166, 1989.