# PhyloSofS User Manual

*PHYLOgenies of Splicing isOforms Structures*

Adel Ait-hamlat, Lelia Polit,

Hugues Richard and Elodie Laine.

Sorbonne Universités, UPMC University Paris 06, CNRS, IBPS, UMR 7238,
Laboratoire de Biologie Computationnelle et Quantitative (LCQB), France.

# 1 Dependencies

PhyloSofS requires the following Python packages:

- NetworkX, to handle phylogenetic trees,
- NumPy, to handle transcript tables,
- PyGraphviz, to allow visualization of the phylogenies.

PhyloSofS also uses some external tools:

- The HH-suite, to identify homologous templates,
- Modeller, to construct the 3D models,
- Naccess, to compute solvent accessible surface areas,
- Procheck, to assess the quality of the models.

# 2 Installing and running PhyloSofS

To install and run PhyloSofS, follow the instructions below:

- Unzip the archive in the directory of your choice.
- Set the environment variable PHYLO_HOME to the directory containing the program.
- Run the program by typing "python $PHYLO_HOME/phylosofs.py".
- For example, test the program on the example data:
"python $PHYLO_HOME/phylosofs.py -mode P -inSeq dat/JNK3.txt"

## 2.1 Mandatory arguments and input data

The **-mode** option is mandatory and specifies which task PhyloSofS should perform:

- **P** for reconstructing transcripts' phylogenies,
- **M** for modeling the tertiary structures of the isoforms,
- **PM** for both.

### 2.1.1 Phylogenetic reconstruction

A text file should be given in input with the option **-inSeq**. The file should contain:

- a first line containing the gene tree (in SMILES format) in which the reconstructed forest will be embedded,
- $n$ lines corresponding the $n$ leaves of the tree, where each line has the format $nameLeaf_i$ : $trans_i^1$,$trans_i^2$,...$trans_i^t$, with $t$ the number of transcripts in leaf $nameLeaf_i$ and $trans_i^j$ a string giving the exon composition of the $j^{th}$ transcript of leaf $nameLeaf_i$. See the file `JNK3.txt` in the `dat/` directory for an example.

### 2.1.2 Molecular modeling

The path to the input data should be given with the option **-inStruct**. The input data should be:

- either a directory with multifasta files, one file per species, each file containing the sequences of the transcripts observed in the considered species (see in the `dat/` directory for examples),
- or a single fasta file with the sequence of only one transcript (in that case, the **-uniq** option must be active)

A configuration file must also be provided with the option **-c**. It should contain the paths to the external tools and databases needed to perform the molecular modeling task. See `config.txt` for an example.

## 2.2 Optional arguments

Here is a list of optional arguments:

| | |
|---|---|
| **-o** | output directory (by default: $PWD) |

**Reconstruction of transcripts' phylogenies**

| | |
|---|---|
| **-b** | birth cost (by default: 5) |
| **-d** | death cost (by default: 3) |
| **-m** | mutation cost (by default: 2) |
| **-ni** | number of iterations (by default: 1) |
| **-noPrune** | disable the removal of exons that appear in only one transcript |
| **-printOnly** | perform only the generation of the PDF file enabling to visualize a transcripts' phylogeny given as input via the option -topo |
| **-s** | starting score (by default: not considered) |
| **-suff** | suffix (by default: _bdm_n) |
| **-topo** | initial topology (by default: maximum or random topology), or transcripts' phylogeny to be printed out (if the -printOnly option is active) |

**Generation of isoforms' 3D models**

| | |
|---|---|
| **-nt** | number of templates to retain (by default: 5) |
| **-only3D** | perform only the 3D modeling step (skip template search) |
| **-onlyQuality** | perform only the 3D models quality assessment |
| **-uniq** | treat only one transcript whose sequence is taken from the fasta file indicated by the -i option<br>the search by the topology corresponding to the maximum number of binary subnodes at each nodes (forest with the smallest possible number of trees), otherwise it starts from a randomly chosen topology |

The term *topology* designates a forest structure, which is specified by the numbers of binary subnodes ($b_n$), left subnodes ($l_n$) and right subnodes ($r_n$) for each node $n$.

If no score is given via the option **-s**, then the algorithm starts the search with the topology corresponding to the maximum number of binary subnodes at each nodes (forest with the smallest possible number of trees). Otherwise, it starts from a randomly chosen topology

## 2.3 Output files

### 2.3.1 Phylogenetic reconstruction

The transcripts' phylogeny(ies) of minimal cost is(are) written in the file `solution_$SUFF.pk`, where `$SUFF` is the suffix given with the **-suff** option. The file is readable by the Pickle module of Python. The object that will be loaded by the Pickle module is a list of phylogenies (NetworkX graphs) that correspond to different configurations obtained from the same forest structure. They differ only by their pairing of transcripts, while having the same cost. They represent different equivalent solutions for a given forest structure.

Additional files of the form `solution_$SUFF_config$i.xxx` are also provided, where $i is the index of the configuration and `xxx` is one of the following extensions:

- **dot**, the graph representing the phylogenetic forest (in DOT language).
- **info**, a text file giving the list of all transcripts, where each line has the format $i\_j$: $trans_i^j$, with $i$ the index of the node (species), $j$ the index of the subnode (transcript) and $trans_i^j$ a string giving the exon composition of the $j^{th}$ transcript of the $i^{th}$ species.
- **pdf**, the image of the phylogenetic forest.
- **sum**, a text file giving the number of trees, deaths and orphans in the phylogenetic forest.

**Note:** The user can use PhyloSofS to only generate those files (without any calculation) by using the option **-printOnly**. For this, he/she must give, via the option **-topo**, a file readable by the Pickle module of Python and containing all the configurations in only one NetworkX graph. Such a file is generated at the end of a phylogenetic reconstruction calculation: `tree_$SUFF.pk`.

Two directories are also created:

- **bestTopos/**, where every forest structure whose score is better than the best score encountered so far is written during the course of the calculation.
- **betterTrees/**, where all forests whose scores are better than or equal to the starting score (given by the option **-s**) are written.

The generated files are readable by the Pickle module of Python. The files stored in the directory **betterTrees/** are suitable for applying the **-printOnly** option.

### 2.3.2 Molecular modeling

One directory is created for each leaf (species), containing:

- the single fasta files containing the sequences of the transcripts,
- the annotated PDB files containing the coordinates of the 3D models of the isoforms (exon numbers in the B-factors column),
- the Naccess output files containing the solvent accessible surface areas computed for every residues,
- the Procheck output files containing information about the quality of the 3D models.

Additionally, a file `quality.sum` is written that contains a summary of the information collected on the 3D models:

- *transcript*, Ensembl accession id of the transcript
- *lenFull*, length of the full transcript's sequence
- *percentSS*, percentage of residues predicted in well-defined secondary structures
- *lenModel*, length of the generated 3D model
- *dihedrals*, Procheck G-factor computed for the torsion angles
- *covalent*, Procheck G-factor computed for the covalent bonds and angles
- *overall*, Procheck overall G-factor
- *dope*, normalized Dope score computed by Modeller
- *rSurf*, ratio between the number of surface residues and the total number of residues
- *rHydroph*, ratio between the number of hydrophobic surface residues and the total number of hydrophobic residues

# 3 Troubleshooting

For any question, feel free to contact Elodie Laine or Hugues Richard.