# User tutorial for HGTphyloDetect

## Table of Contents

## How to install HGTphyloDetect?

Install HGTphyloDetect:

git clone https://github.com/SysBioChalmers/HGTphyloDetect.git

or

pip install git+https://github.com/SysBioChalmers/HGTphyloDetect.git

To install all the required Python packages, run the command line:

cd HGTphyloDetect

pip install -r requirements.txt

## Any examples for quick start?

We provide a user-friendly example for small test, users just need to prepare a FASTA file including protein id and protein sequence, note that protein id should be from the GenBank protein database.

(1) If you are now in the HGTphyloDetect directory, just enter into the folder (example) via the command line:

cd example

(2.1) Then users can run the script for the input file (default AI value = 45, out_pct = 0.90):

python HGT_workflow.py input.fasta

(2.2) If users want to change the default values for the parameters used in the pipeline, e.g., AI value = 40, out_pct = 0.80, just reset the constant values and run the following:

python HGT_workflow.py input.fasta AI=40 out_pct=0.80

(3) Finally, our software could generate the output results as a file under the folder (example) for this gene/protein. The output file includes some important information, i.e., Alien index, E value and donor information. For example:

| Gene/Protein | Alien index | E value | Donor id | Donor taxonomy |
|---|---|---|---|---|
| AAT92670 | 199.18 | 3.15e-87 | WP_208929673 | Bacteria/Firmicutes |

## How to run high-throughput HGT identification?

Here, we recommend users to run BLASTP process separately, because it may take too much time especially if users want to execute BLASTP for many genes/proteins, such as 1,000 or 10,000. For BLASTP, users can enter into the folder (main), and run:

python blastp.py input.fasta

Then users could get BLASTP result under the generated folder (blastp_files).

To identify potential genes that have been horizontally acquired from evolutionarily distant organisms (i.e., prokaryote to eukaryote)
Users can enter into the directory (main) and run the command line:

```
cd main
python HGT_workflow_distant.py input.fasta
```

This python script mainly executes these three steps as follows:

- **Step I**: BLASTP hits are parsed to retrieve associated taxonomic information using the NCBI's taxonomy database.
- **Step II**: Calculate Alien Index (AI) values and out_pct based on the above information.
- **Step III**: Output the horizontal gene transfer (HGT) high-throughput identification results.

However, if users want to change the default values for the parameters used in this pipeline (HGT detection from evolutionarily distant organisms), e.g., AI value = 40, out_pct = 0.80, just reset the constant values and run the following:

```
cd main
python HGT_workflow_distant.py input.fasta AI=40 out_pct=0.80
```

To identify potential genes that have been horizontally acquired from more closely related organisms (e.g., eukaryote to eukaryote)
Users can enter into the directory (main) and run the command line:

```
cd main
python HGT_workflow_close.py input.fasta
```

If users want to change the default values for the parameters used in the pipeline (HGT detection from closely related organisms), e.g., bitscore = 150, HGT_index=0.6, out_pct = 0.70, just reset the constant values and run the following:

```
cd main
python HGT_workflow_close.py input.fasta bitscore=150 HGT_index=0.6 out_pct=0.70
```

## Software dependencies for HGT phylogenetic analysis?

- **MAFFT**

Users can install the MAFFT (multiple alignment program for amino acid or nucleotide sequences) software with conda by running one of the following command lines (referring to the website: https://anaconda.org/bioconda/mafft):

conda install -c bioconda mafft

conda install -c bioconda/label/cf201901 mafft

Or install the MAFFT software according to the tutorial provided by the MAFFT team (please refer to this link: https://mafft.cbrc.jp/alignment/software/source.html)

- **trimAl**

  Users can install the trimAl (a tool for automated alignment trimming in phylogenetic analysis) program with conda by running one of the following command lines (referring to the website: https://anaconda.org/bioconda/trimal):

  conda install -c bioconda trimal

  conda install -c bioconda/label/cf201901 trimal

  Or install the trimAl program according to the installation tutorial provided by the trimAl team (please refer to the GitHub: https://github.com/inab/trimal)

- **IQ-TREE**

  Users can install the IQ-TREE (an efficient software for phylogenetic inference) software with conda by running one of the following command lines (referring to the website: https://anaconda.org/bioconda/iqtree):

  conda install -c bioconda iqtree

  conda install -c bioconda/label/cf201901 iqtree

  Or install the IQ-TREE according to the tutorial provided by the IQ-TREE team (please refer to this link: http://www.iqtree.org/doc/Quickstart)

- **R and R packages (ape and phangorn)**

  Install R programming language at first based on the documentation provided by the R team (https://rstudio-education.github.io/hopr/starting.html).

  Then, copy & paste the following commands to your R command prompt to install the R packages ape and phangorn:

  install.packages("ape",repos="https://cloud.r-project.org",quiet=TRUE)

  install.packages("phangorn",repos="https://cloud.r-project.org",quiet=TRUE)

- **Perl and BioPerl (module Bio::TreeIO)**

  Install Perl: https://www.perl.org/get.html.

  Install BioPerl: https://github.com/bioperl/bioperl-live/blob/master/README.md.

## How many steps for HGT phylogenetic analysis?

**Overview**: To corroborate the accurate identification of HGT genes by their Alien Index (AI) values and out_pct, we extended HGTphyloDetect with a comprehensive phylogenetic analysis pipeline. First, the top 300 homologs with different taxonomic species names are selected from the BLASTP hits for each query sequence. HGTphyloDetect then aligns these homologs with MAFFT (Katoh, et al., 2005) using default settings for multiple sequence alignment, while ambiguously aligned regions are removed with trimAl using its '-automated1' option (Capella-Gutiérrez, et al., 2009). Phylogenetic trees are constructed from these alignments using IQ-TREE (Nguyen, et al., 2015) with 1000 ultrafast bootstrapping replicates. Subsequently, the phylogenetic tree is rooted at the midpoint using the R packages ape (Paradis, et al., 2004) and phangorn (Schliep, 2011). Finally, the resulting phylogenies are visualized using iTol (https://itol.embl.de/) to assess the mode of transmission of each gene.

Therefore, the HGT phylogenetic analysis in HGTphyloDetect could mainly be divided into eight steps (from step I to step VIII) as follows:

- Step I: If users are now located in the HGTphyloDetect directory, first enter into the phylogenetics directory.
- Step II: Obtain a specific FASTA file including a list of homolog identifiers and related sequences.
- Step III: Execute multiple sequence alignment for the above homologs.
- Step IV: Remove ambiguously aligned regions with trimAl using its '-automated1' option.
- Step V: Phylogenetic tree construction from the sequence alignments.
- Step VI: Root the phylogenetic tree at the midpoint.
- Step VII: Generate two annotation files for the visualization of the phylogenetic tree.
- Step VIII: Visualization of the phylogenetic tree.

**But do not worry about these complex steps!** We have compiled step I to step VII into one main script, users just need to run one main script together with their input (FASTA file format). Only the final step needs to be done by the users to ensure the high-quality of phylogenetic tree visualization.

## How to run phylogenetic analysis by using HGTphyloDetect?

To run phylogenetic analysis, we provided all the scripts under this folder (phylogenetics) in the repository. As described above, we have compiled the separate steps (step I to step VII) into one shell script (generate_tree.sh) shown in **Figure 1**. Users can directly run these steps by using the command line together with the input FASTA file:

sh generate_tree.sh input.fasta

After running this command line, the HGTphyloDetect toolbox would execute the program from step I to step VII, and then generate the phylogenetic tree and two annotation files, which are input files for the iTol visualization. And the prerequisite is that users have installed all the software dependencies regarding phylogenetic analysis.

```
#!/bin/bash

var=$1
gene=${var%.fas*}
echo 'This is gene '$gene' '
echo 'Begin to run!!!'

cd  ../

python scripts/HGT_homologs_sequence.py input/${gene}.fasta

mafft --thread 6 --auto ./input/${gene}_homologs.fasta > ./intermediate/${gene}_aln.fasta

trimal -in ./intermediate/${gene}_aln.fasta -out ./intermediate/${gene}_aln_trimmed.fasta -automated1

iqtree -nt 6 -st AA -s ./intermediate/${gene}_aln_trimmed.fasta -m TEST -mrate G4 -keep-ident -bb 1000 -pre ./intermediate/${gene}

Rscript scripts/midpoint_tree.R ${gene}

perl scripts/create_iTOL_config.pl ./intermediate/${gene}_midpoint.tree

echo 'Yep, finish!!!'
```

**Figure 1**. Screenshot of the integrative shell script about running phylogenetic analysis for horizontal gene transfer (HGT) event.

## Any examples regarding HGT phylogenetic analysis?

Here, we used an example (one important protein YOL164W in *Saccharomyces cerevisiae*) for easy understanding. Firstly, Users need to enter into the scripts folder under phylogenetics, and directly run the command line shown as below:

sh generate_tree.sh YOL164W.fasta

After running this command line, there are three files that could be output in the intermediate folder: YOL164W_midpoint.tree, YOL164W_midpoint-font.txt and YOL164W_midpoint-color.txt. Note that the first file is the generated phylogenetic tree, another two additional files are

the iTol annotation files for this tree. After that, users can use the iTol website (https://itol.embl.de/) to visualize the phylogenetic tree.

In order to facilitate the visualization of the phylogenetic tree, users just need to log into the iTol website (https://itol.embl.de/) and then upload these three files (YOL164W_midpoint.tree, YOL164W_midpoint-font.txt and YOL164W_midpoint-color.txt) to the website, and then the phylogenetic tree could be output by the 'Export' function on the website.

In this step, the input files and output files are:
Input files: YOL164W_midpoint.tree, YOL164W_midpoint-font.txt and YOL164W_midpoint-color.txt under the intermediate folder.
Output files: YOL164W_circle_tree.pdf or YOL164W_rectangular_tree.pdf under the output folder.
If users choose the 'Circular' mode for the phylogenetic tree visualization, then the tree would be generated like this (**Figure 2**):

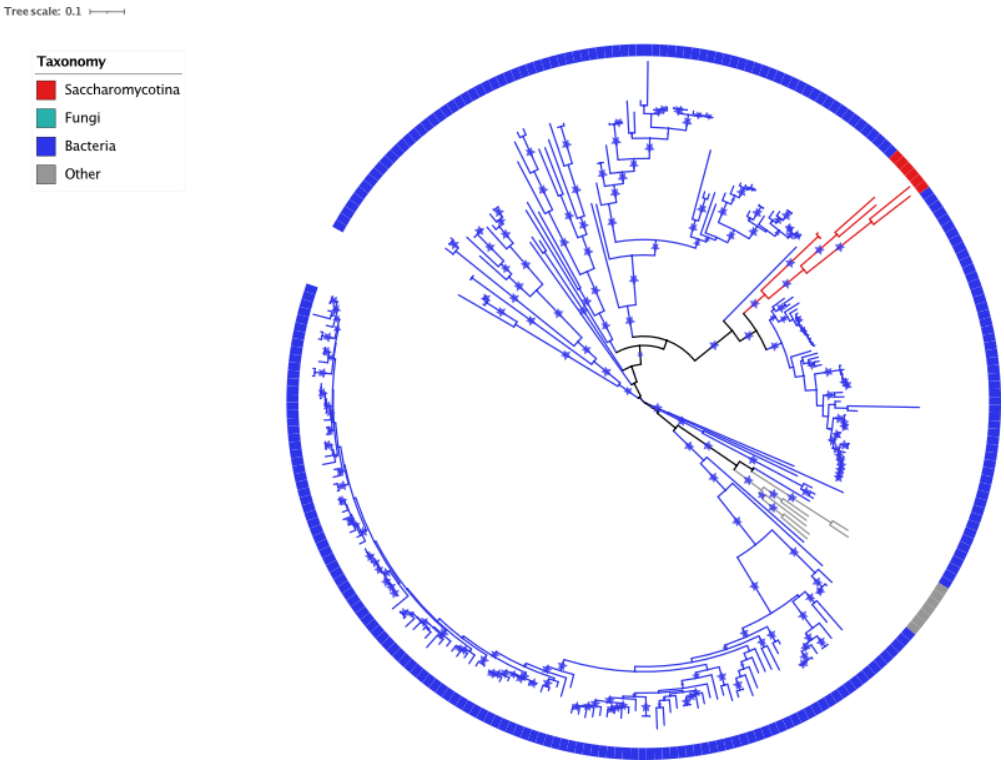


**Figure 2**. Maximum Likelihood (ML) phylogeny of a protein YOL164W in *Saccharomyces cerevisiae*. Branches with bootstrap support higher than 80% are shown by star.

If users choose the ‘Rectangular’ mode for the phylogenetic tree visualization, then the tree would be generated like this (**Figure 3**):

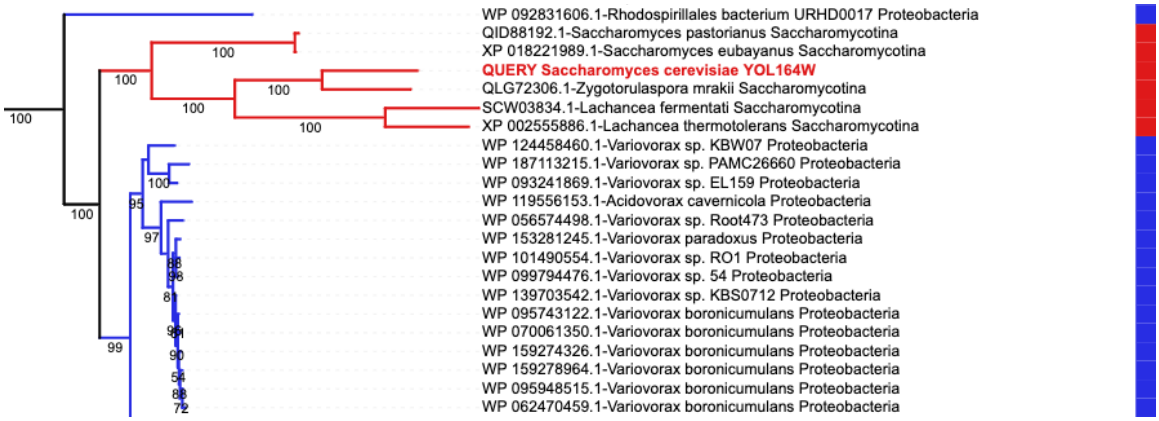


**Figure 3**. The detailed phylogenetic tree represents pruned ML phylogeny depicting the phylogenetic relationship between the protein YOL164W from the *Saccharomycotina* subphylum and its close relatives from other bacteria.

Please note the software dependencies that we used in the HGT phylogenetic analysis for this case (YOL164W) are: MAFFT v7.310, trimAl v1.4, IQ-TREE v1.6.12, ape v5.4-1, phangorn v2.5.5 and iTol v5.

Okay, now it is done for the HGT phylogenetic analysis, users can know the potential donors and the path of gene transmission by navigating the phylogenetic tree.

## Looking for help?

If you encounter any issues, feel free to contact the software developer: Dr. Le Yuan (leyu@chalmers.se)