Arab Academy for Science, Technology & Maritime Transport

College of Engineering: Computer department

Computer algorithms (CC412)

User tutorial

HGTphyloDetect facilitating the identification and Phylogenetic analysis of horizontal gene transfer

supervised by: Dr. Mohammed Ali Maher

Abeer Hossam Ali 20104789

Haneen Hazem Talaat 20104712

Salma Ismail Ibraheem 20105188

Nada Ahmed Fouad Shalan 20104602

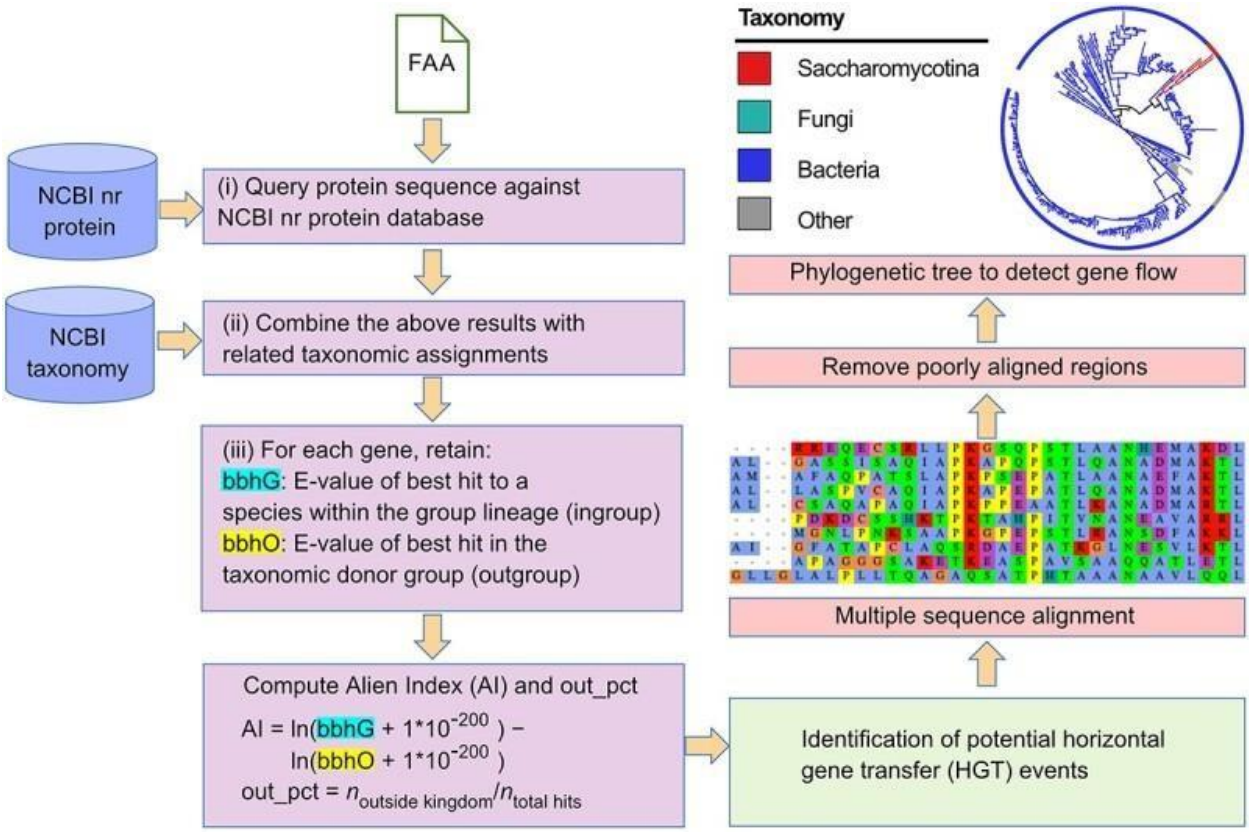
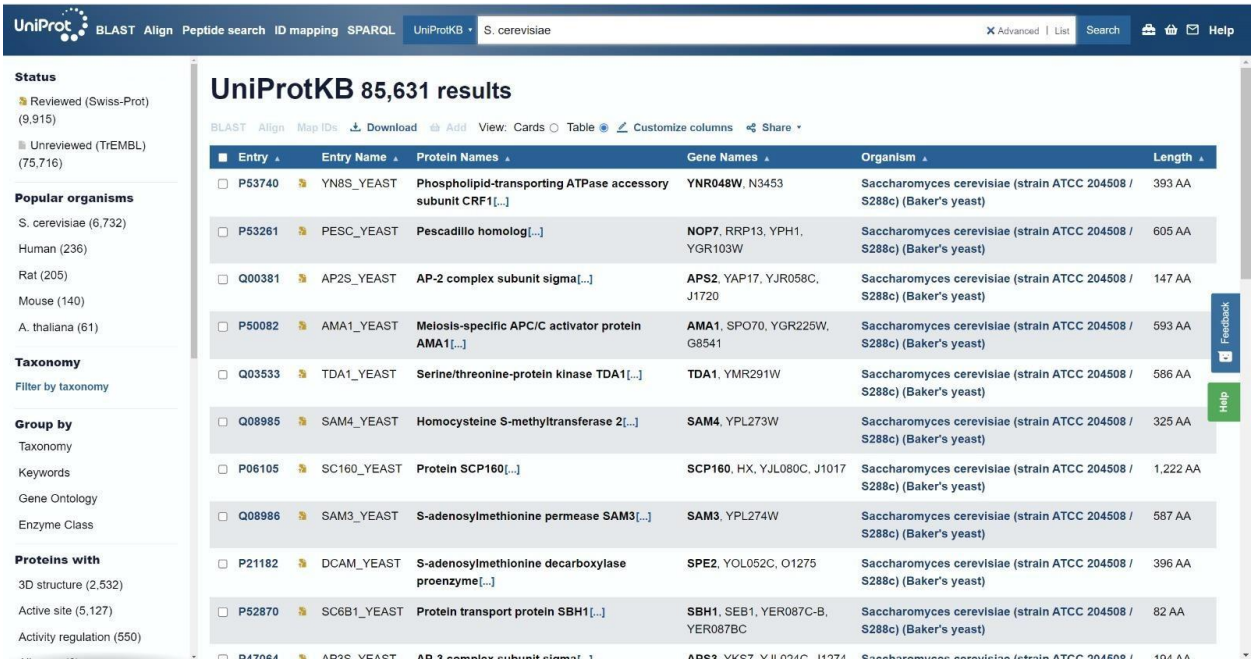


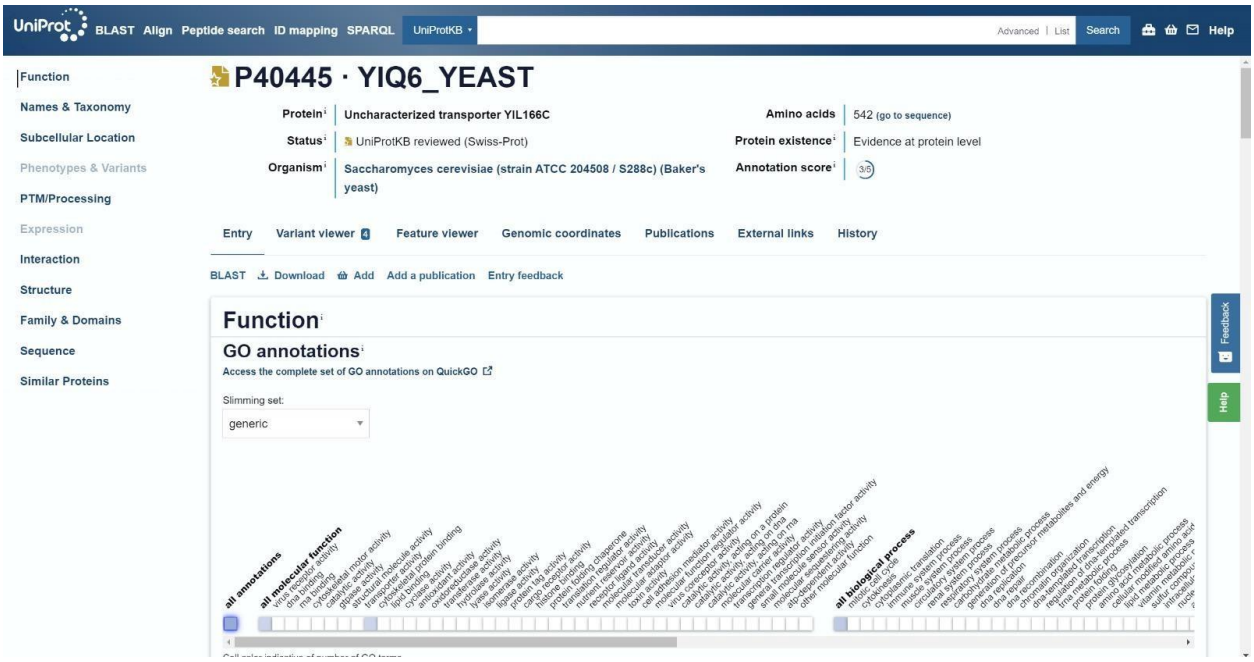
Figure (1): Overview of the HGTphyloDetect pipeline for automated identification of HGT events from evolutionarily distant organisms (e.g. prokaryote to eukaryote).

The project is divided into two parts HGT-detection , phylogenetic tree construction

Testing the performance of HGTphyloDetect

To evaluate the prediction performance of HGTphyloDetect, we applied this toolbox to one species (S. cerevisiae) that has manually curated HGT events described in previously published works, allowing benchmarking of our approach. By running HGTphyloDetect for all (more than 6000) genes in S. cerevisiae with the default parameters, we were able to identify the gene name using the accession number to search in uniport UniProt (e.g. YIL166C(P40445), YOL164W (Q08347),where the YOL164W is the gene name and Q08347 is the accession number).





Detection of HGT from distant and close organisms

Outline:

1. **<u>Step1</u>:** search gene accession_id(entry)

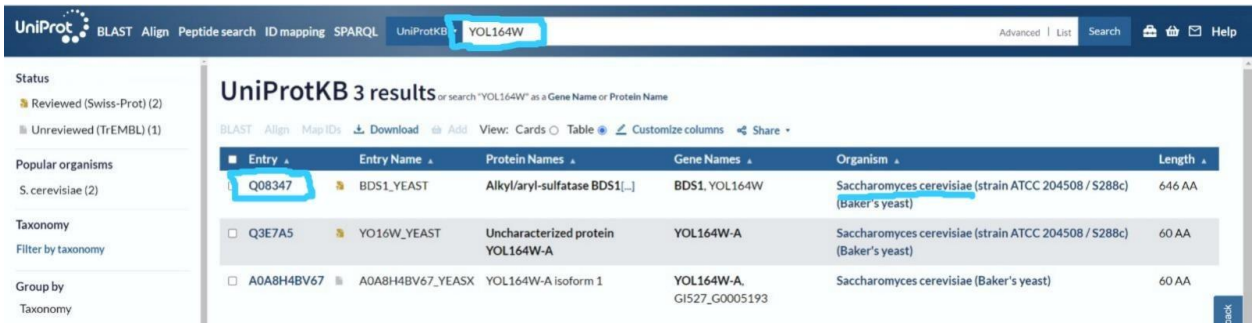


Figure (1): getting accession_id for distantly related gene

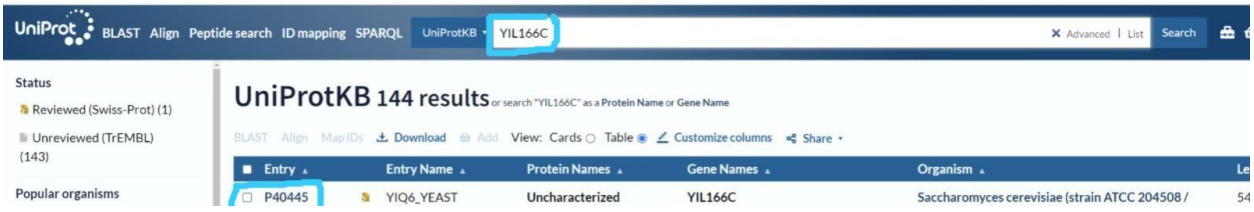
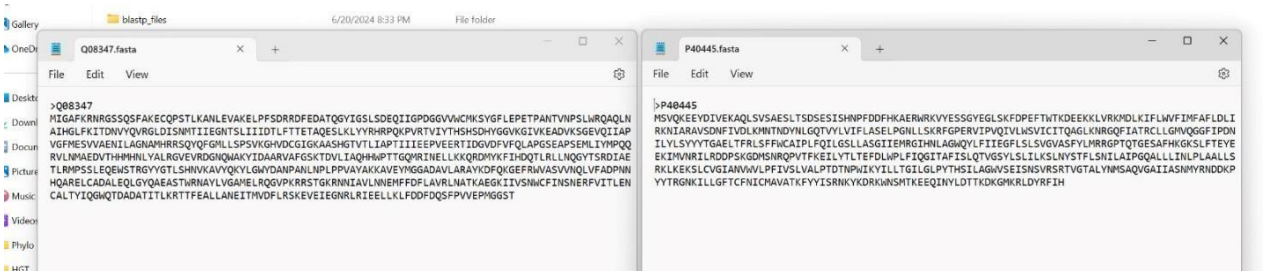


Figure (2): getting accession_id for closely related gene

2. **Step 2:** make the format of the fasta file by naming it by its accession_id (e.g. Q08347.fasta, P40445.fasta) containing**:**
   < accession_id and protein sequence>



3. **Step 3:** this fasta file that was created before will go through **blastp.py code** which will provide the output file in text format(accession_id.txt)**, dir**: will be in blastp_files, using the following command:

python blastp.py accession_id. Fasta

"this step takes time in computing up to 15 to 20 minutes according to the hitlist_size=250 "

(the input file name can change according to the sequence's accession id) and the output file will be text file. The provided distant and close python codes in GitHub can run multiple genes at once. Although, the submitted code will run only one gene at once to increase the performance

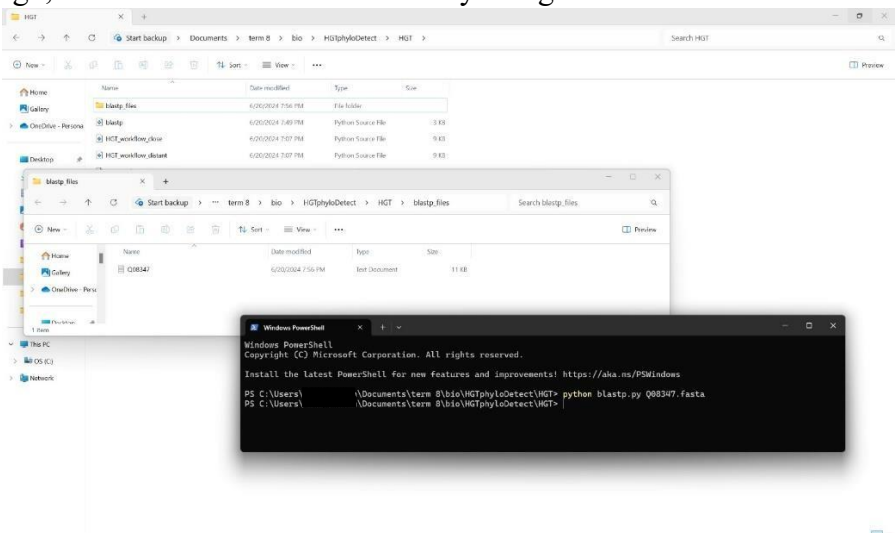


Figure (1): input file of blastp.py (distant)

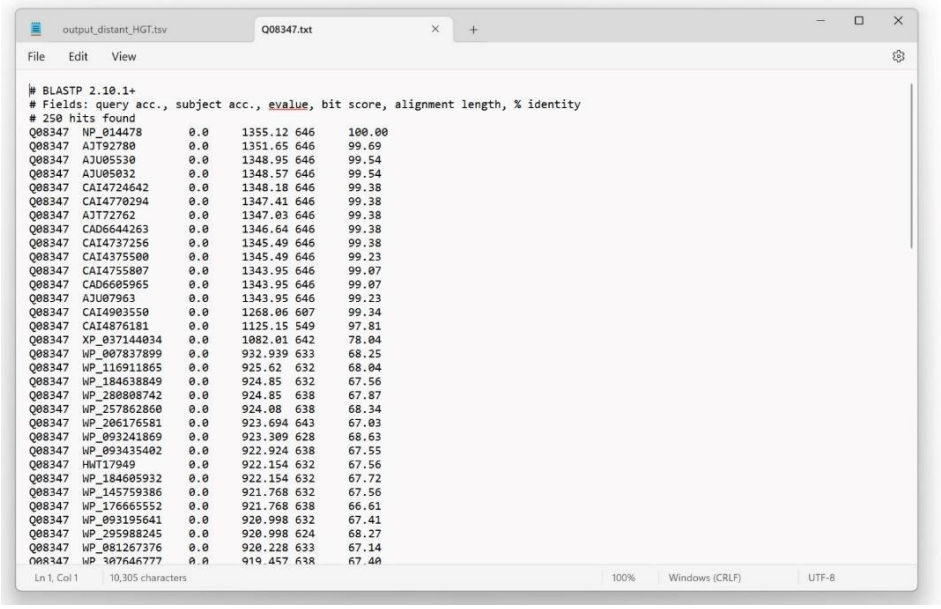


Figure (2): output text file of blastp.py(distant)

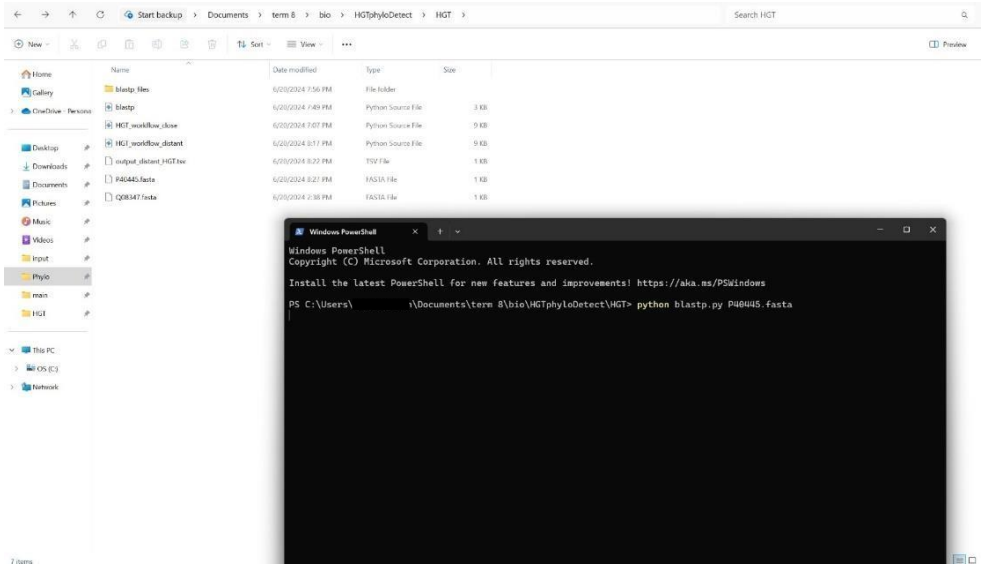


Figure (3): input file of blastp.py(close)

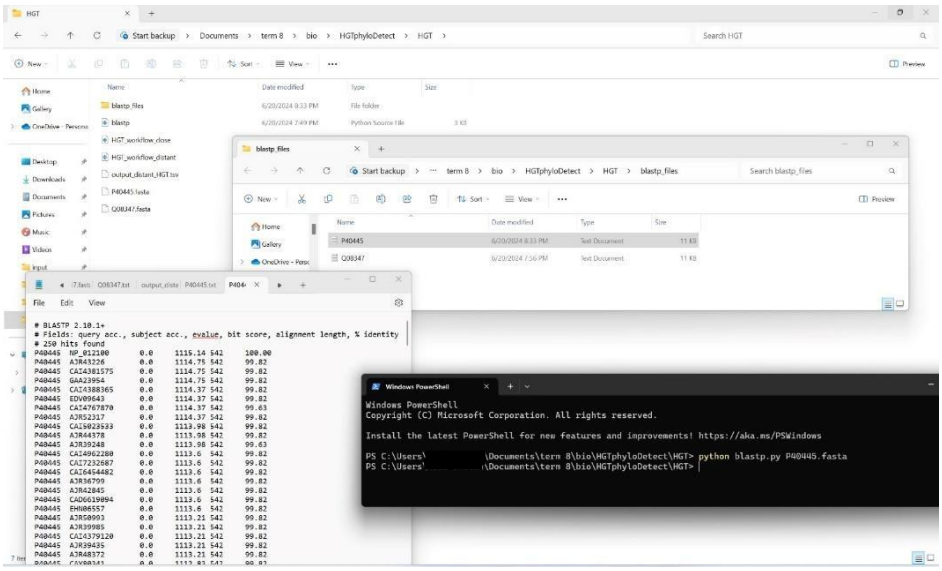


Figure (2): output text file of blastp.py(close)

4. **<u>Step 4</u>:** this fasta format file**(accession_id.fasta)** and blastp.py file**(accession_id.txt)** will be used in the distant and close codes

The following steps is illustrating how to run the python code of the distantly related organisms:

1. Open the cmd from the path of distant code location.
2. Type the following command into cmd:

   python HGT_workflow_distant.py Q08347.fasta
3. The generated output code will be in the format of fasta file.

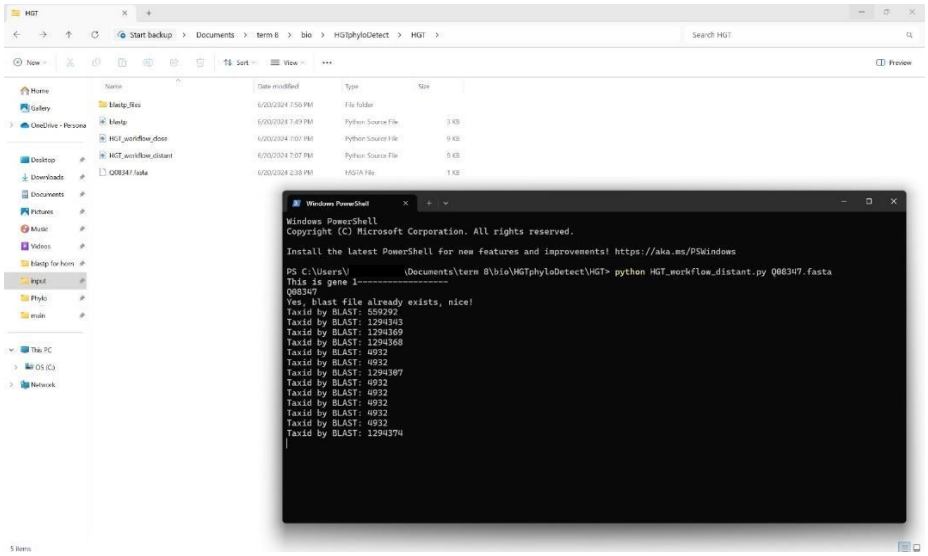


Figure (4): running distant workflow code

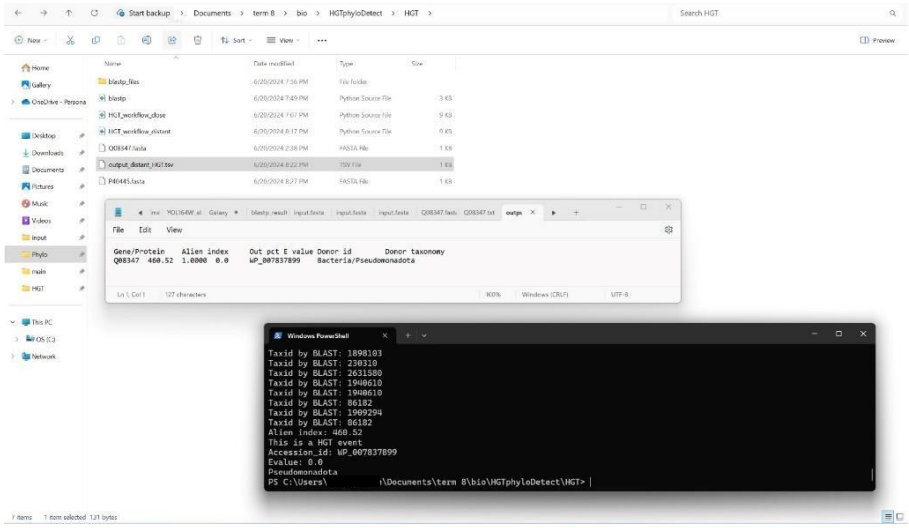


Figure (5): output of distant workflow code

The following steps is illustrating how to run the python code of the closely related organisms:

1. Open the cmd from the path of close code location.
2. Type the following command into cmd:

   python HGT_workflow_close.py P40445.fasta
3. The generated output code will be in the format of fasta file.

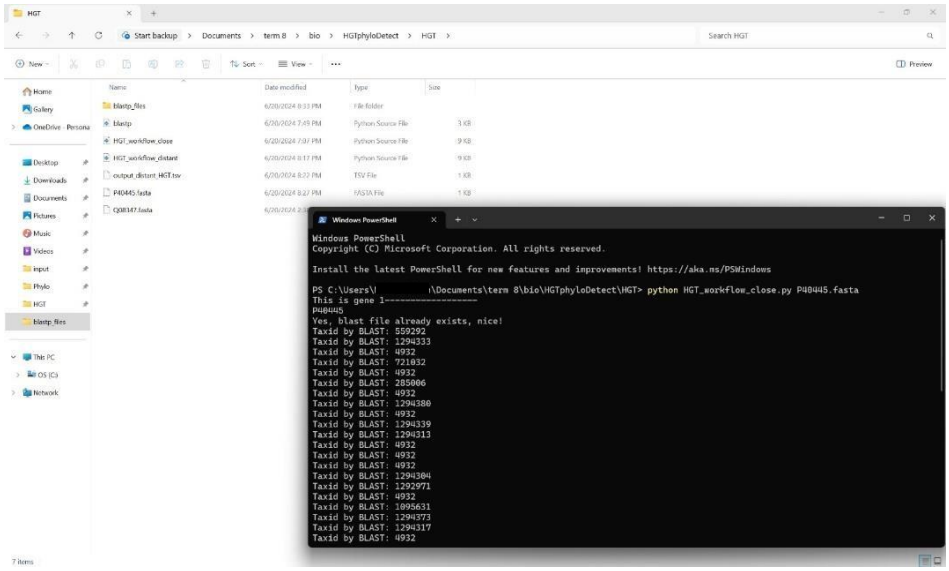


Figure (6): running of close workflow code

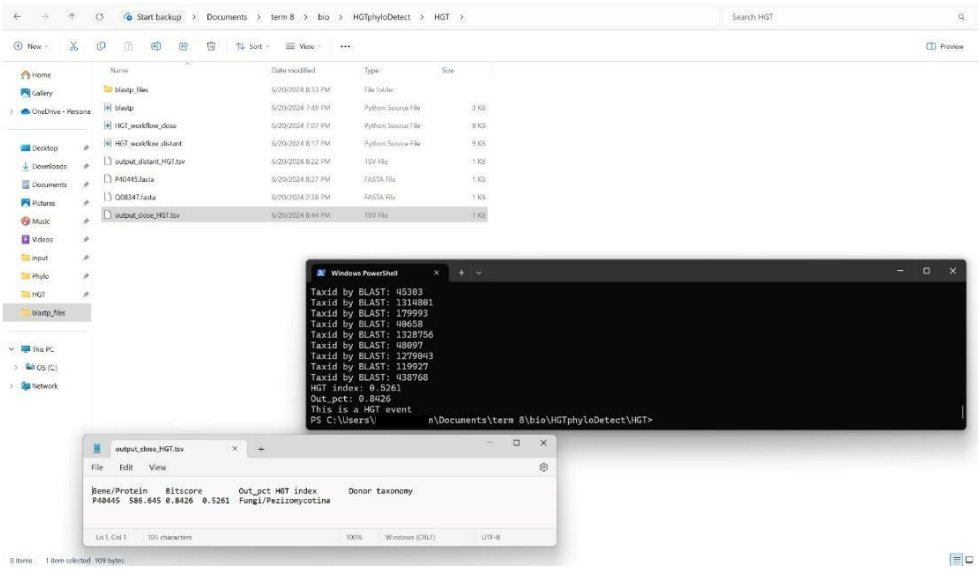


Figure (7): output of close workflow code

Construction of the phylogenetic analysis pipeline

The HGT phylogenetic analysis in HGTphyloDetect could mainly be divided into a series of steps as follows:

- **Step 1**: We start by entering two input files: YOL164W.fasta and YOL164W.txt (get them as mentioned before from HGT) autogenerating an output: YOL164W_homologs. fasta
- **Step 2**: using the command **python HGT_homologs_sequence.py input/YOL164W.txt**

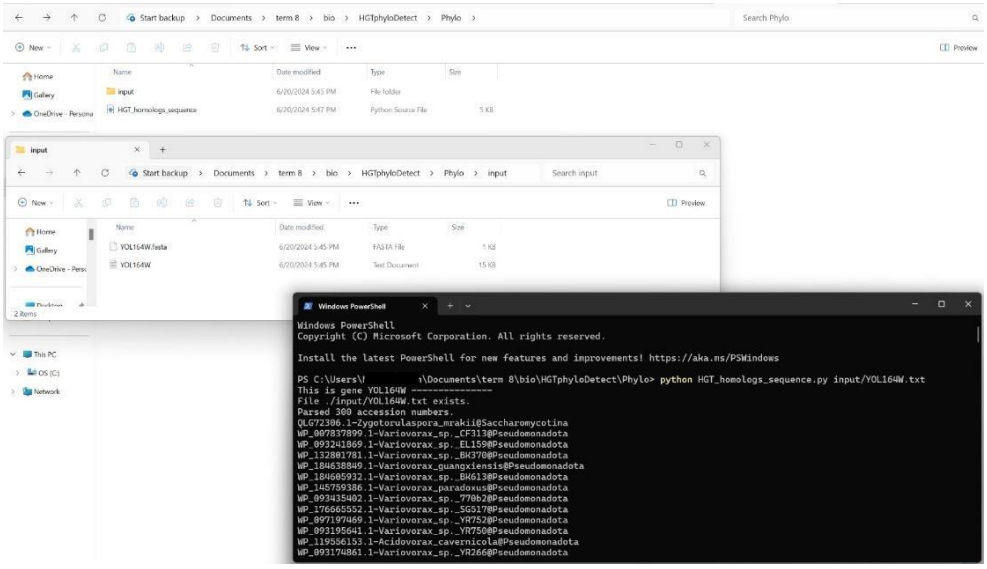


Figure (1): showing the running of the input files

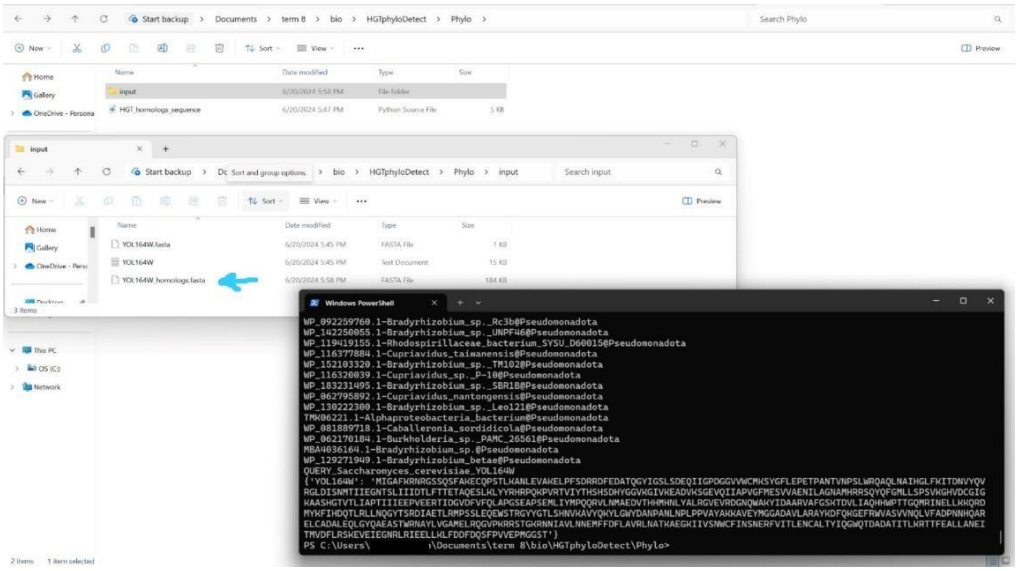


Figure (2): showcasing the output file

Step 3: Using the output file from last step as an input file entering it into MAFFT tool online (MSA) at link: https://mafft.cbrc.jp/alignment/server/index.html

Note: Not changing any of the default settings

1. Taking the input file named **YOL164W_homologs. fasta** into the tool
2. Generating an output file **mafft_op_aligned.fasta**

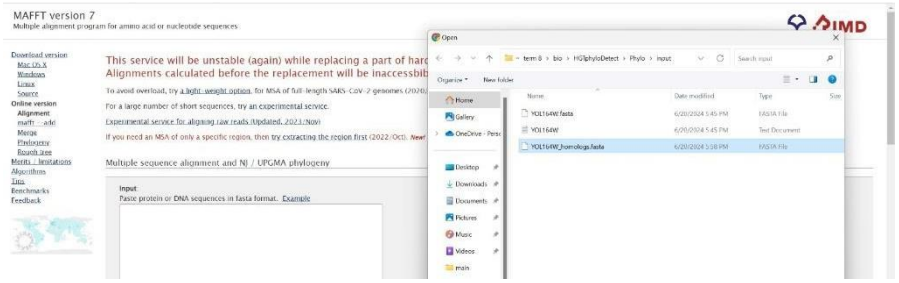


Figure (3): input file and MAFFT tool

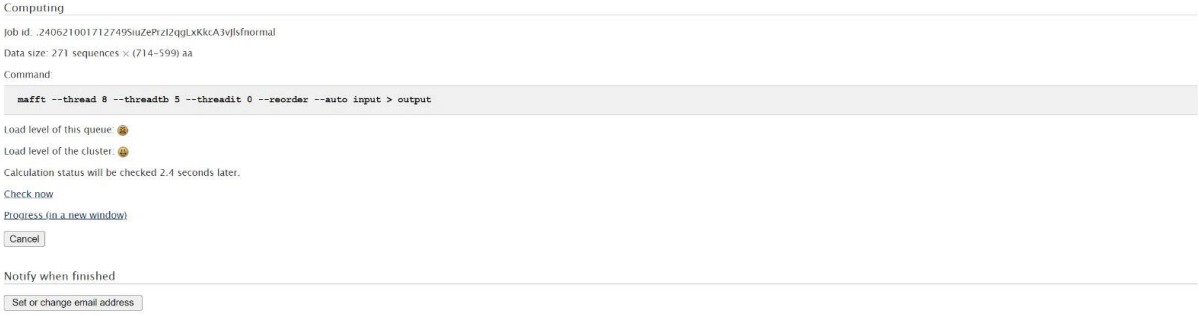


Figure (4): running the input file

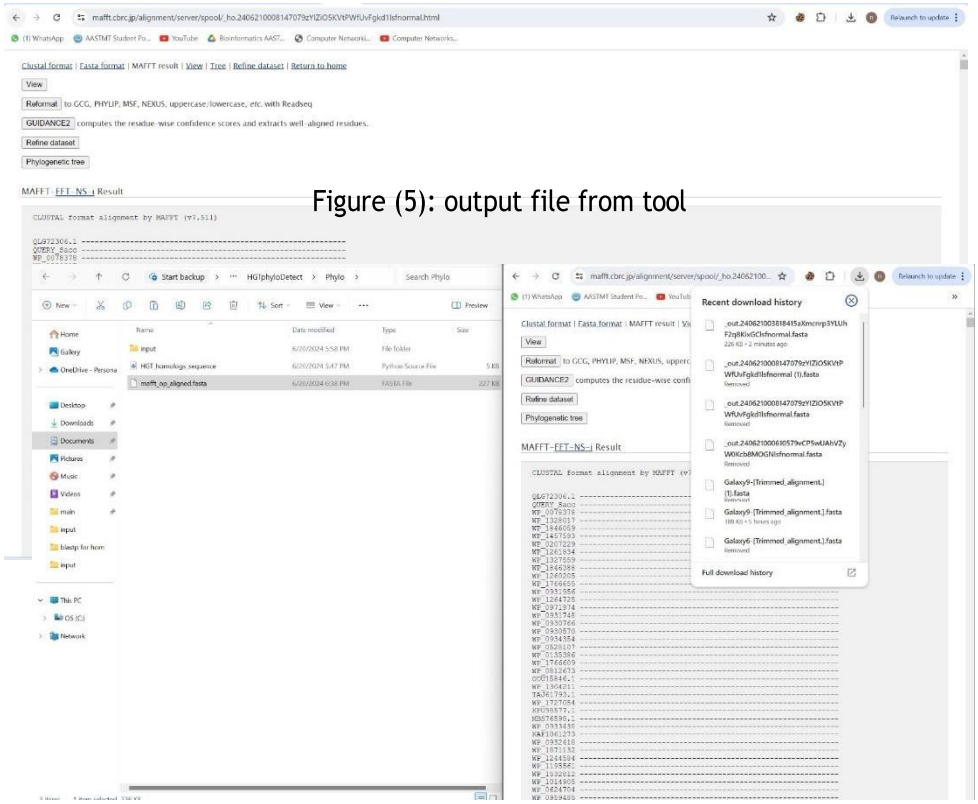


Figure (5): output file from tool

Figure (6): output renamed

- **Step 4**: Applying the output file as input file again **mafft_op_aligned.fasta,** entering it into

TrimAL tool online at link:

https://usegalaxy.fr/?tool_id=toolshed.g2.bx.psu.edu%2Frepos%2Fpadge%2Ftrimal%2Ftrimal%2F0.1.0

1. Using automated1 feature: designed to automatically detect and remove poorly aligned or ambiguously aligned regions from the MSA, ensuring that only well-aligned regions are used for downstream analyses.
2. The generated output file **aligned_trimmed.fasta**

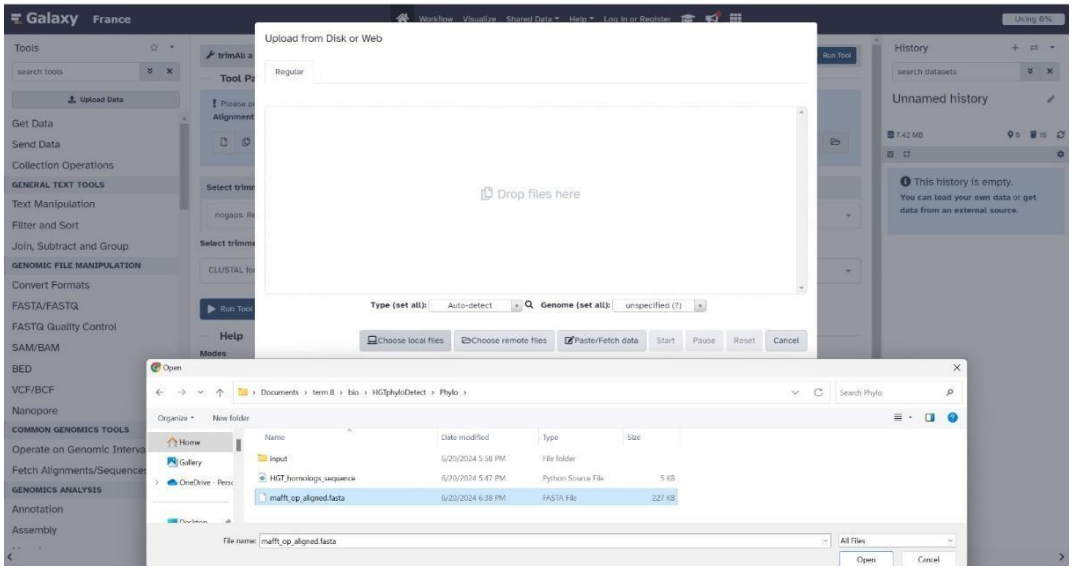


Figure (7): input file

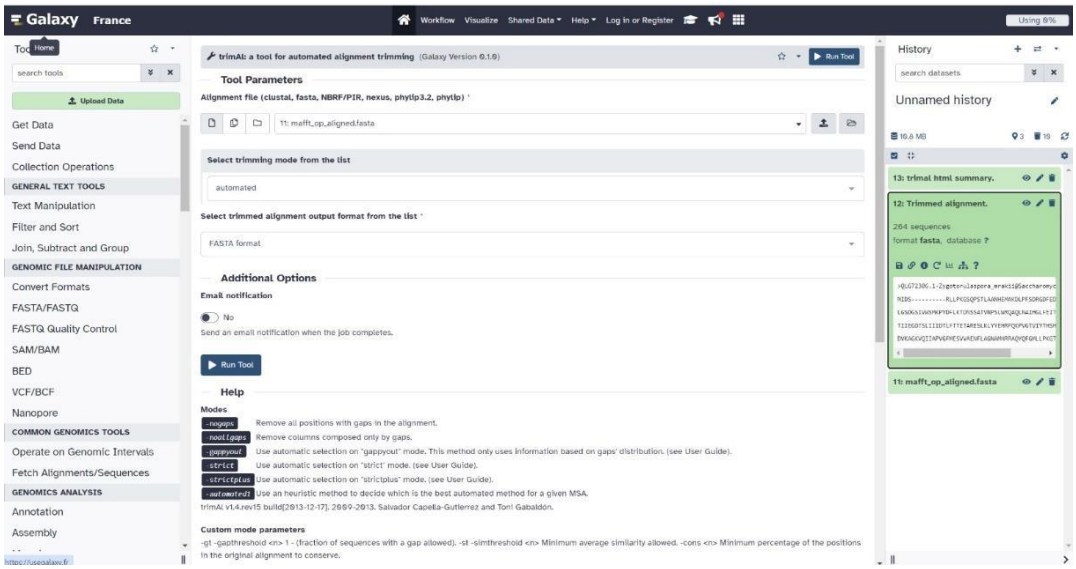


Figure (8): selecting fasta, automated 1, and producing output

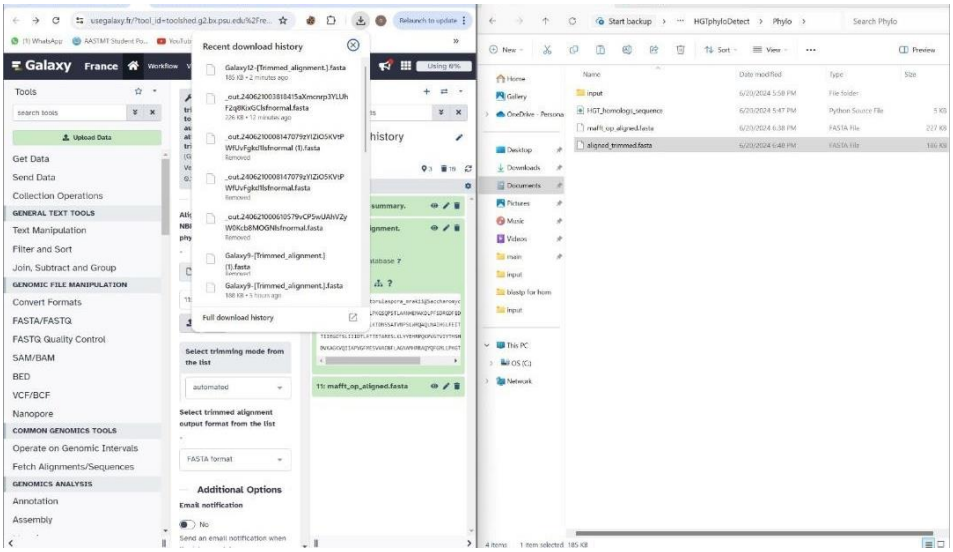


Figure (9): output file rename

- **Step 5**: Following the trimming, we will take the output from last step and enter as input into the IQ-TREE online tool at link: http://iqtree.cibiv.univie.ac.at/

    1. Take output file **aligned_trimmed.fasta** as input
    2. Enter it into IQ-TREE tool
    3. output file extension will be **aligned_trimmed.fasta.treefile**

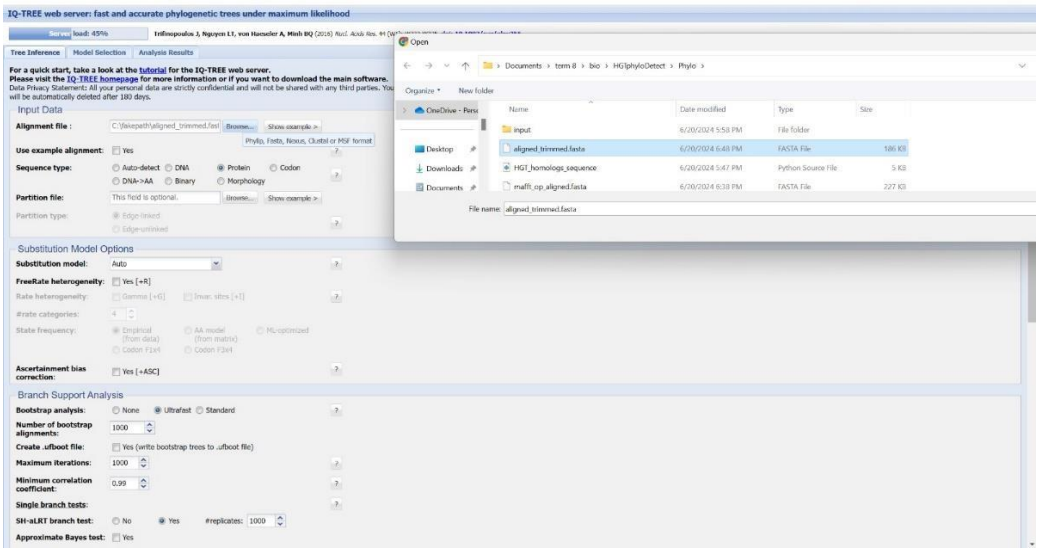


Figure (9): input file extension

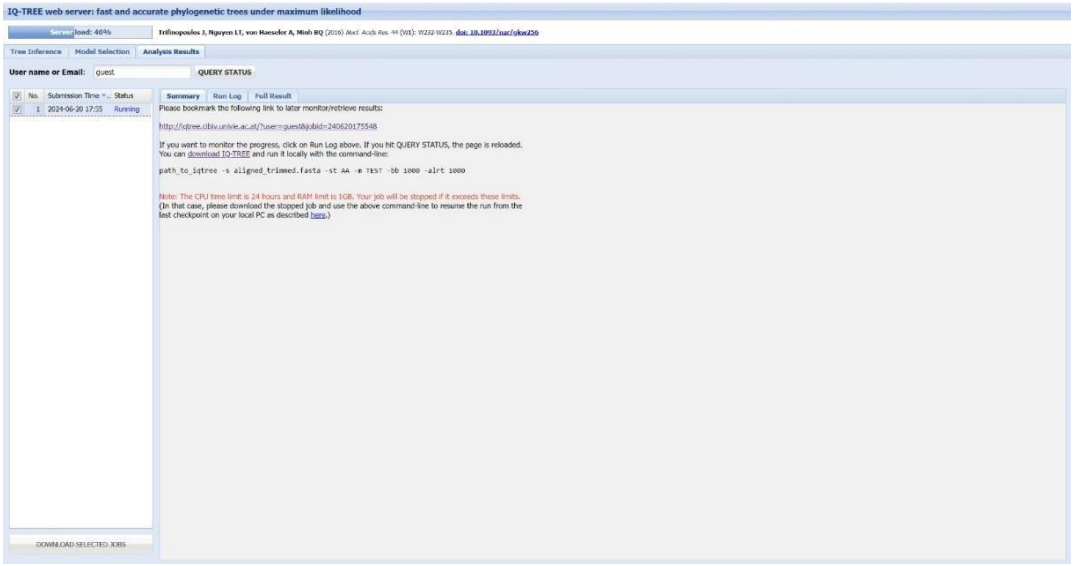


Figure (10): running

Note: the running stage may take time to generate result (7 hours)

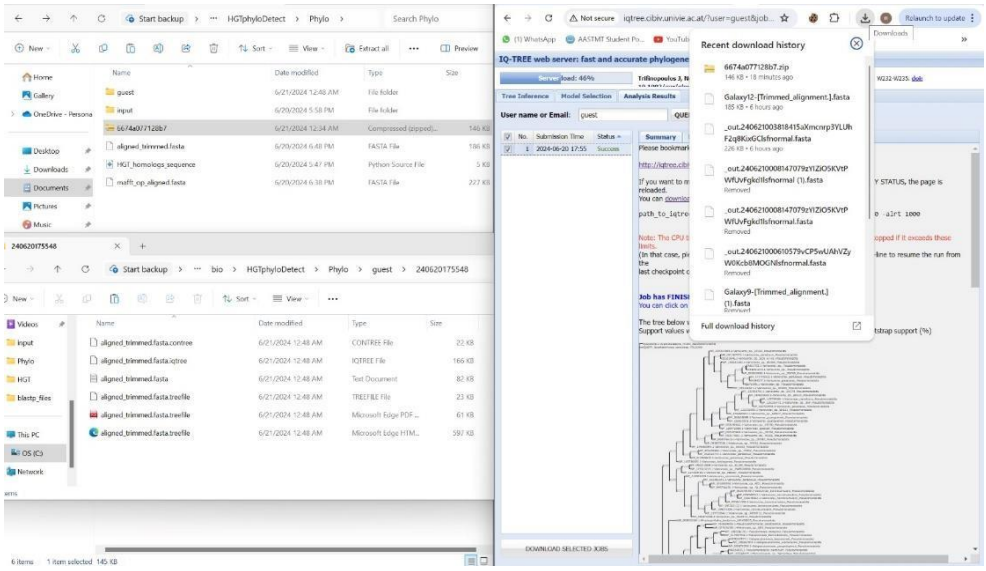


Figure (11): unzip downloaded folder of IQ-TREE output

Note: the only needed output is aligned_trimmed.fasta.treefile

- <u>Step 6</u>: taking the **aligned_trimmed.fasta.treefile** as an input to the **R script** getting output file **aligned_trimmed.fasta_midpoint.tree** (to produce rooted tree)

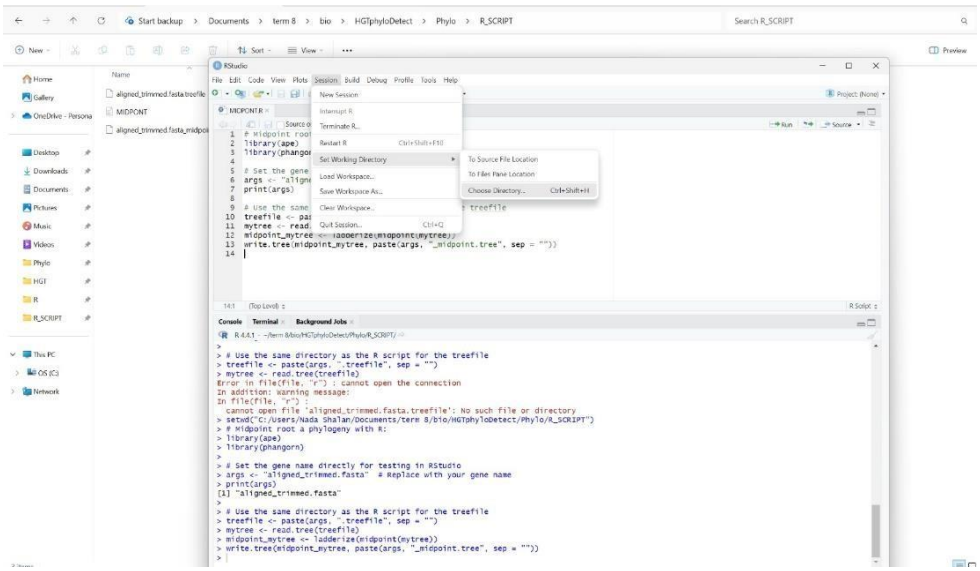


Figure (12): running R script (R studio)

Download: library(ape), library(phangorn) (packages)

Note: don't forget to Go to Session -> Set Working Directory -> Choose Directory... and select the appropriate directory.

Then select all ,run

- **Step 7**: running **create_iTOL_config.perl** script, taking **aligned_trimmed.fasta_midpoint.tree** as input

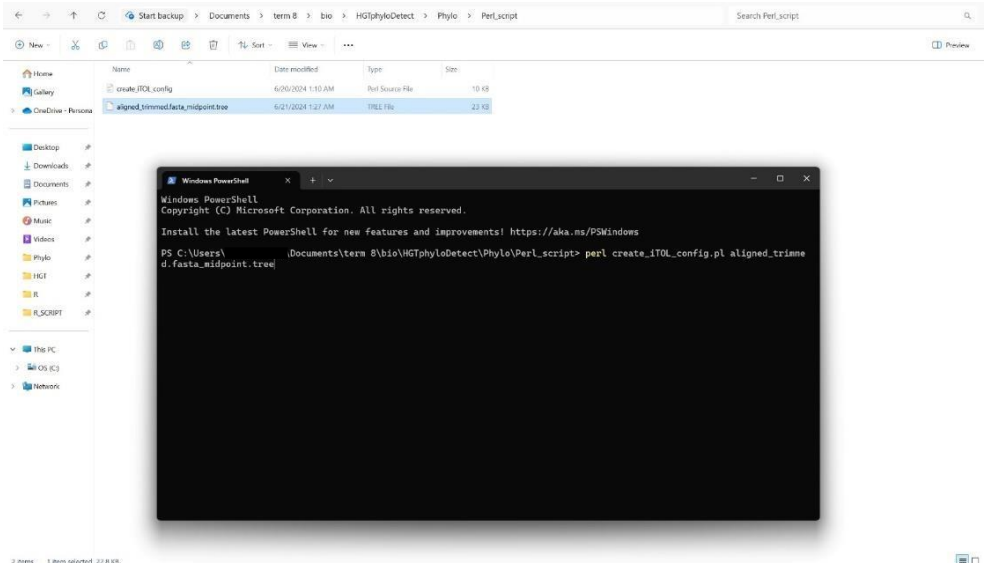


Figure (13) : running the input file

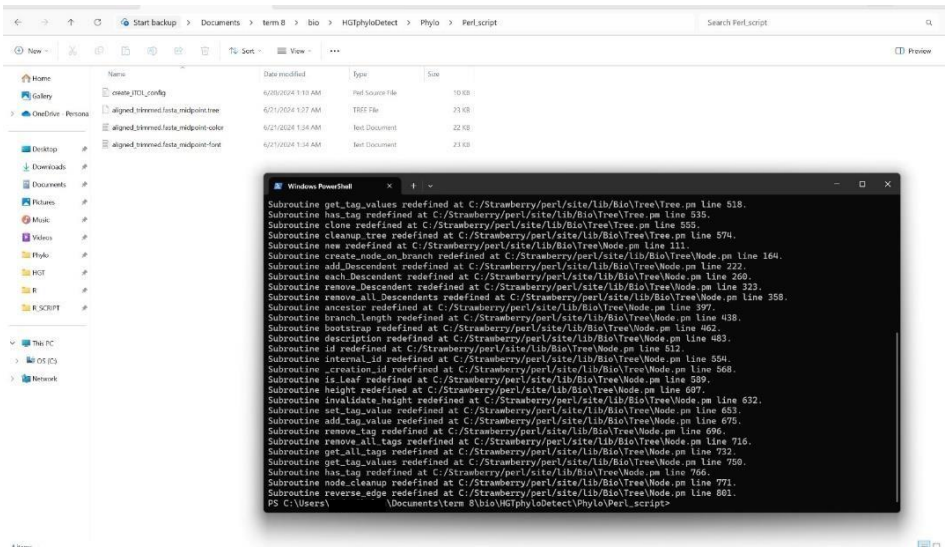


Figure (14): output file(annotations)

- **Step 8**: there are three files that could be output in the intermediate
- folder: **aligned_trimmed.fasta_midpoint.tree**, **aligned_trimmed.fasta_midpoint-font.txt** and **aligned_trimmed.fasta_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.
  The output is the tree visualizing the HGT

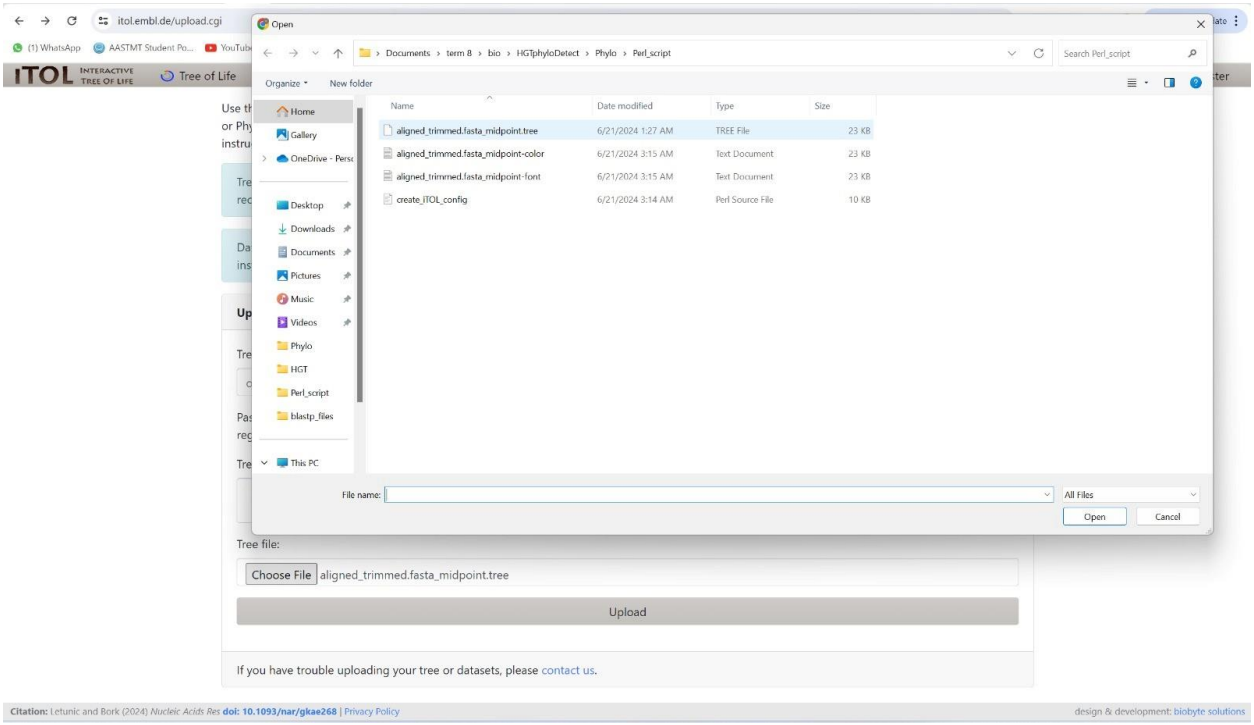


Figure (15): uploading(.tree)

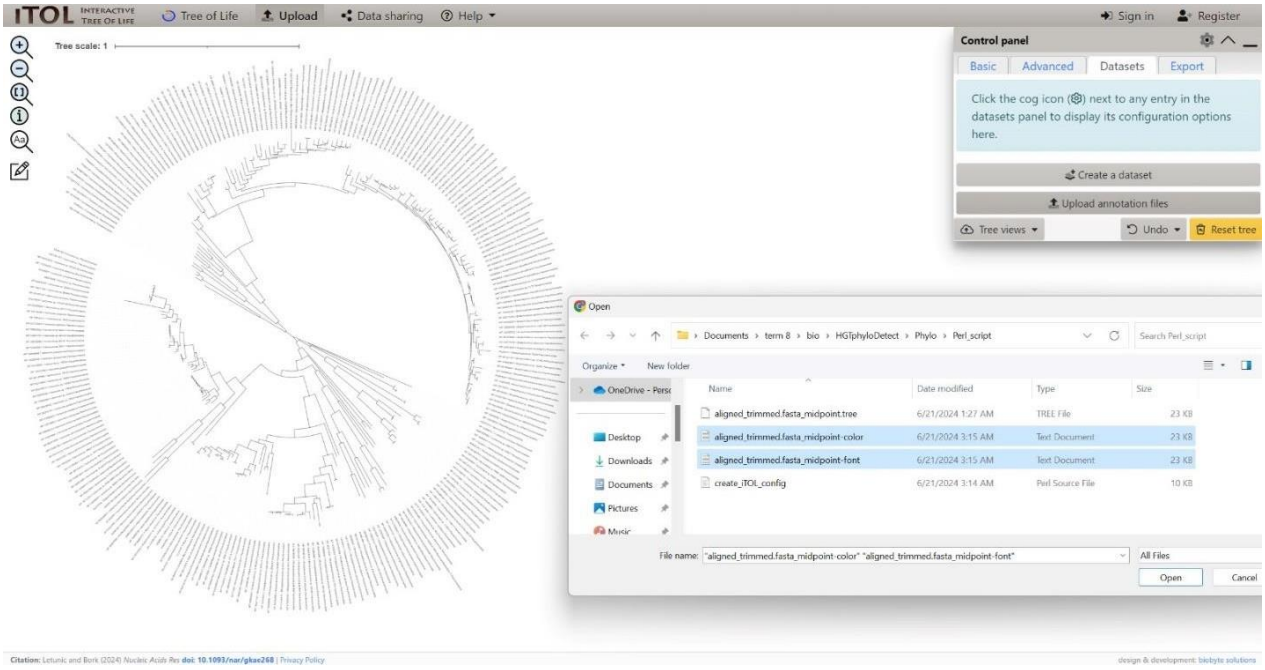


Figure (16): adding(annotations files)

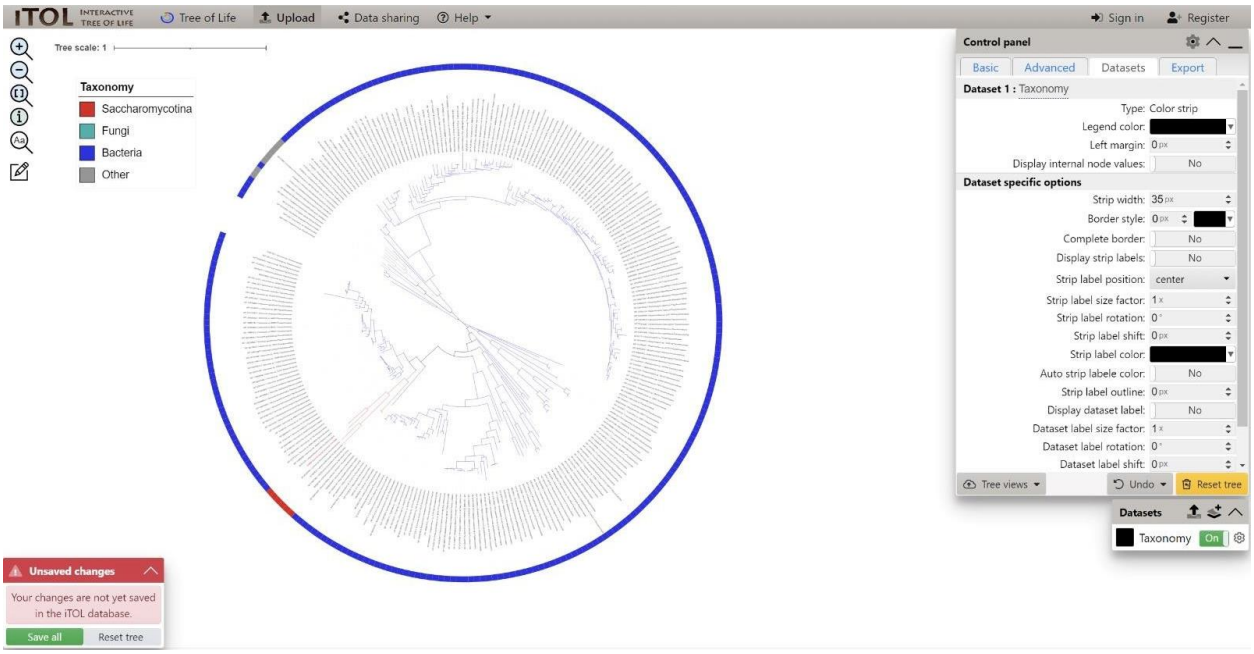


Figure (16): circular output

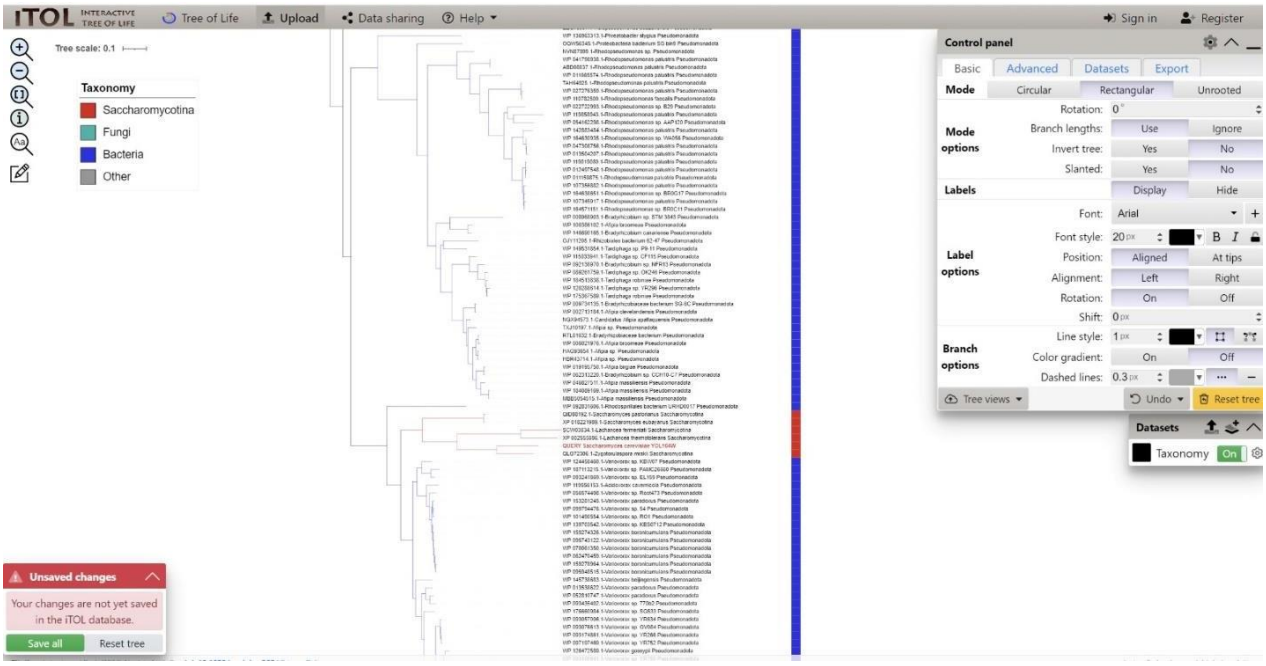


Figure (17): rectangular output