



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

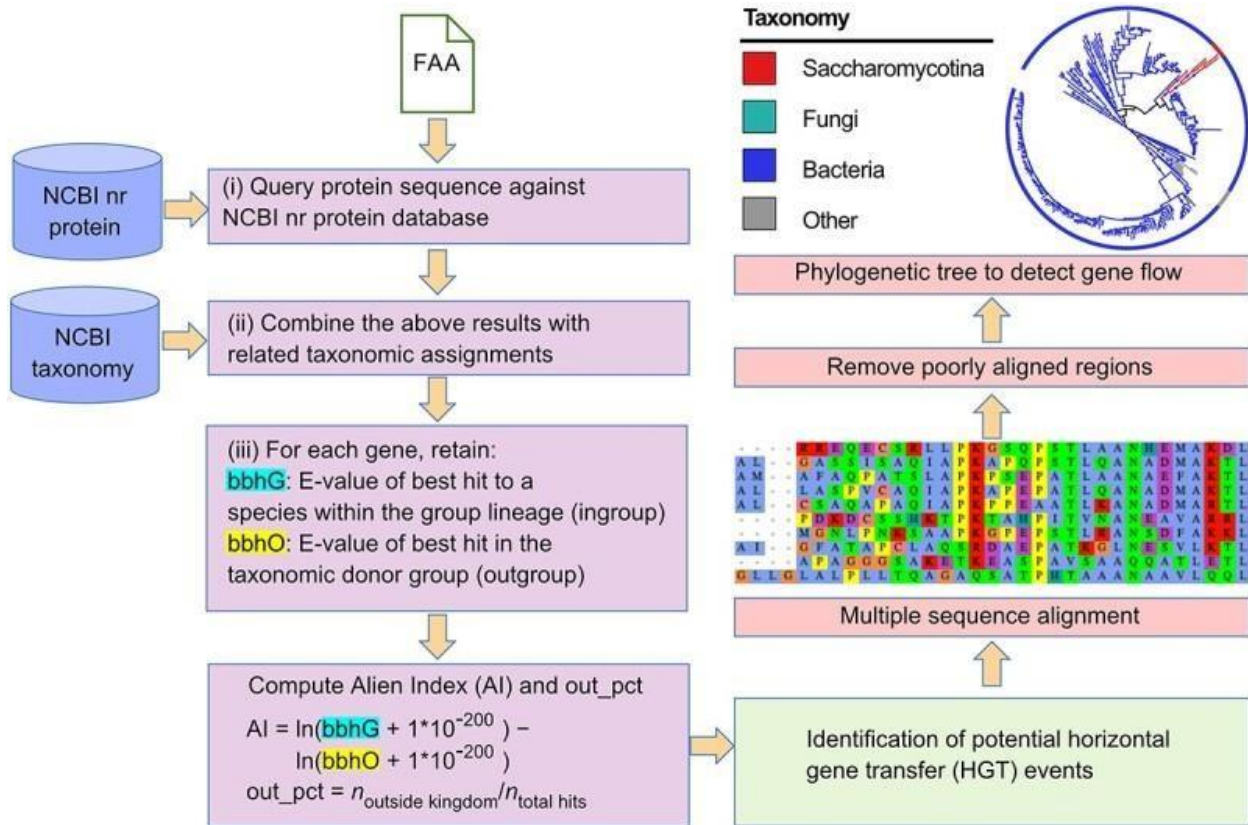


Figure (1): Overview of the HGTphylodetect pipeline for automated identification of HGT events from evolutionarily distant organisms (e.g. prokaryote to eukaryote).

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 uniprot [UniProt](#) (e.g. YIL166C(P40445), YOL164W (Q08347), where the YOL164W is the gene name and Q08347 is the accession number).

UniProtKB 85,631 results

Entry	Entry Name	Protein Names	Gene Names	Organism	Length
P53740	YN8S_YEAST	Phospholipid-transporting ATPase accessory subunit CRF1[...]	YNR048W, N3453	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	393 AA
P53261	PESC_YEAST	Pescadillo homolog[...]	NOP7, RRP13, YPH1, YGR103W	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	605 AA
Q00381	AP2S_YEAST	AP-2 complex subunit sigma[...]	APS2, YAP17, YJR058C, J1720	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	147 AA
P50082	AMA1_YEAST	Meiosis-specific APC/C activator protein AMA1[...]	AMA1, SPO70, YGR225W, G8541	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	593 AA
Q03533	TDA1_YEAST	Serine/threonine-protein kinase TDA1[...]	TDA1, YMR291W	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	586 AA
Q08985	SAM4_YEAST	Homocysteine S-methyltransferase 2[...]	SAM4, YPL273W	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	325 AA
P06105	SCP160_YEAST	Protein SCP160[...]	SCP160, HX, YJL080C, J1017	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	1,222 AA
Q08986	SAM3_YEAST	S-adenosylmethionine permease SAM3[...]	SAM3, YPL274W	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	587 AA
P21182	DCAM_YEAST	S-adenosylmethionine decarboxylase proenzyme[...]	SPE2, YOL052C, O1275	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	396 AA
P52870	SC6B1_YEAST	Protein transport protein SBH1[...]	SBH1, SEB1, YER087C-B, YER087BC	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	82 AA

P40445 · YIQ6_YEAST

Protein: Uncharacterized transporter YIL166C

Status: UniProtKB reviewed (Swiss-Prot)

Organism: *Saccharomyces cerevisiae* (strain ATCC 204508 / S288c) (Baker's yeast)

Amino acids: 542 (go to sequence)

Protein existence: Evidence at protein level

Annotation score: 3.0

Entry Variant viewer Feature viewer Genomic coordinates Publications External links History

BLAST Download Add Add a publication Entry feedback

Function

GO annotations:

Access the complete set of GO annotations on QuickGO

Slimming set: generic

all annotations

all molecular function

all biological process

Detection of HGT from distant and close organisms

Outline:

1. Step1: search gene accession_id(entry)

UniProtKB 3 results or search "YOL164W" as a Gene Name or Protein Name

Entry	Entry Name	Protein Names	Gene Names	Organism	Length
Q08347	BDS1_YEAST	Alkyl/aryl-sulfatase BDS1[...]	BDS1, YOL164W	<i>Saccharomyces cerevisiae</i> (strain ATCC 204508 / S288c) (Baker's yeast)	646 AA
Q3E7A5	YO16W_YEAST	Uncharacterized protein YOL164W-A	YOL164W-A	<i>Saccharomyces cerevisiae</i> (strain ATCC 204508 / S288c) (Baker's yeast)	60 AA
A0A8H4BV67	A0A8H4BV67_YEASX	YOL164W-A isoform 1	YOL164W-A, G1527_G0005193	<i>Saccharomyces cerevisiae</i> (Baker's yeast)	60 AA

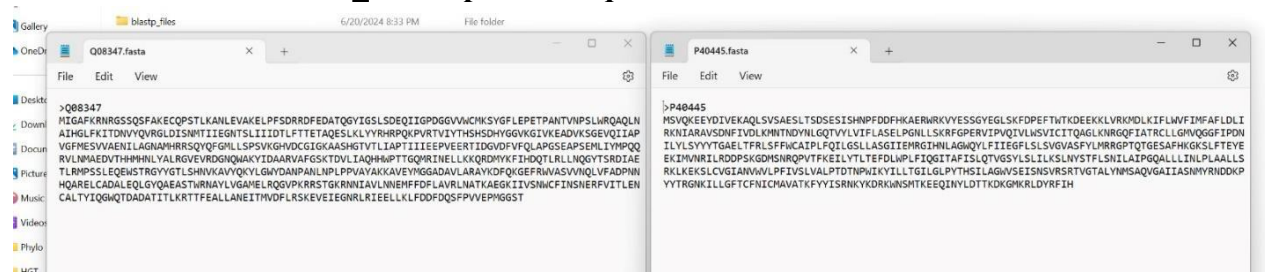
Figure (1): getting accession_id for distantly related gene

UniProtKB 144 results or search "YIL166C" as a Protein Name or Gene Name

Entry	Entry Name	Protein Names	Gene Names	Organism	Length
P40445	YIQ6_YEAST	Uncharacterized	YIL166C	<i>Saccharomyces cerevisiae</i> (strain ATCC 204508 / S288c) (Baker's yeast)	542 AA

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

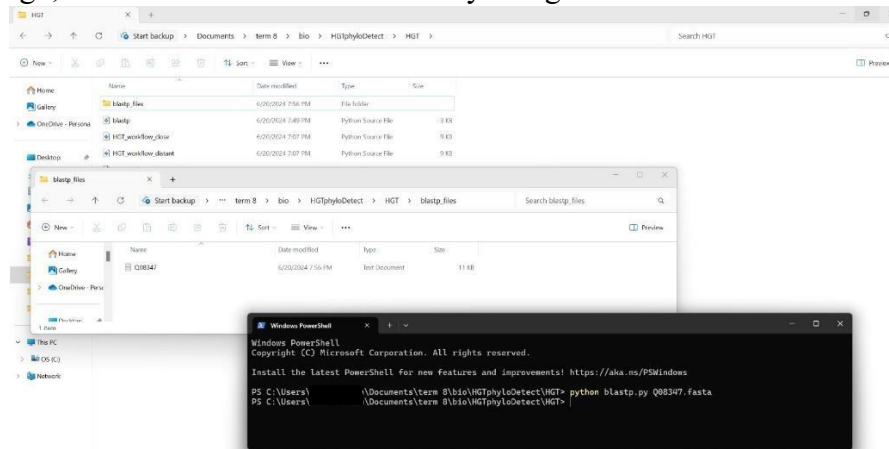


Figure (1): input file of blastp.py (distant)

```

# BLASTP 2.10.1+
# Fields: query acc., subject acc., evalue, bit score, alignment length, % identity
# 250 hits found
Q08347 NP_0144478 0.0 1355.12 646 100.00
Q08347 AJT92788 0.0 1351.55 646 99.69
Q08347 AJU05530 0.0 1348.95 646 99.54
Q08347 AJU05832 0.0 1348.57 646 99.54
Q08347 CAI4724642 0.0 1348.18 646 99.38
Q08347 CAI4770294 0.0 1347.41 646 99.38
Q08347 AJT72762 0.0 1347.03 646 99.38
Q08347 CAD6644263 0.0 1346.64 646 99.38
Q08347 CAI4737256 0.0 1345.49 646 99.38
Q08347 CAI4375500 0.0 1345.49 646 99.23
Q08347 CAI4755807 0.0 1343.95 646 99.07
Q08347 CAD6605965 0.0 1343.95 646 99.07
Q08347 AJU07963 0.0 1343.95 646 99.23
Q08347 CAI4903550 0.0 1268.06 607 99.34
Q08347 CAI4876181 0.0 1125.15 549 97.81
Q08347 XP_037144034 0.0 1082.01 642 78.04
Q08347 WP_007837899 0.0 932.939 633 68.25
Q08347 WP_116911865 0.0 925.62 632 68.04
Q08347 WP_184638849 0.0 924.85 632 67.56
Q08347 WP_280888742 0.0 924.85 638 67.87
Q08347 WP_257862860 0.0 924.08 638 68.34
Q08347 WP_206176581 0.0 923.694 643 67.03
Q08347 WP_093241869 0.0 923.309 628 68.63
Q08347 WP_093435402 0.0 922.924 638 67.55
Q08347 WP_17949 0.0 922.154 632 67.56
Q08347 WP_184605932 0.0 922.154 632 67.72
Q08347 WP_145759386 0.0 921.768 632 67.56
Q08347 WP_176665552 0.0 921.768 638 66.61
Q08347 WP_093195641 0.0 920.998 632 67.41
Q08347 WP_295988245 0.0 920.998 624 68.27
Q08347 WP_081267376 0.0 920.228 633 67.14
Q08347 WP_107646777 0.0 919.457 638 67.40
  
```

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

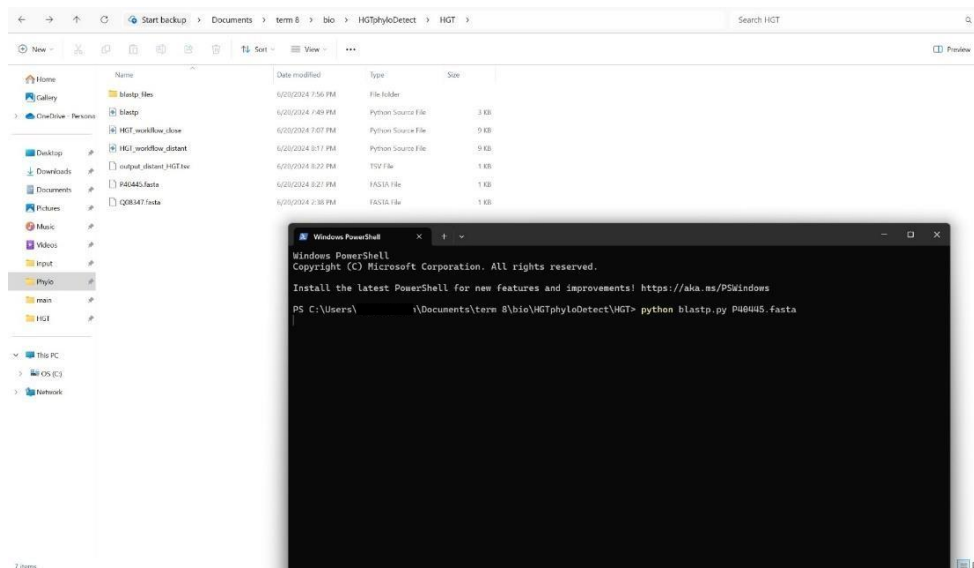


Figure (3): input file of blastp.py(close)

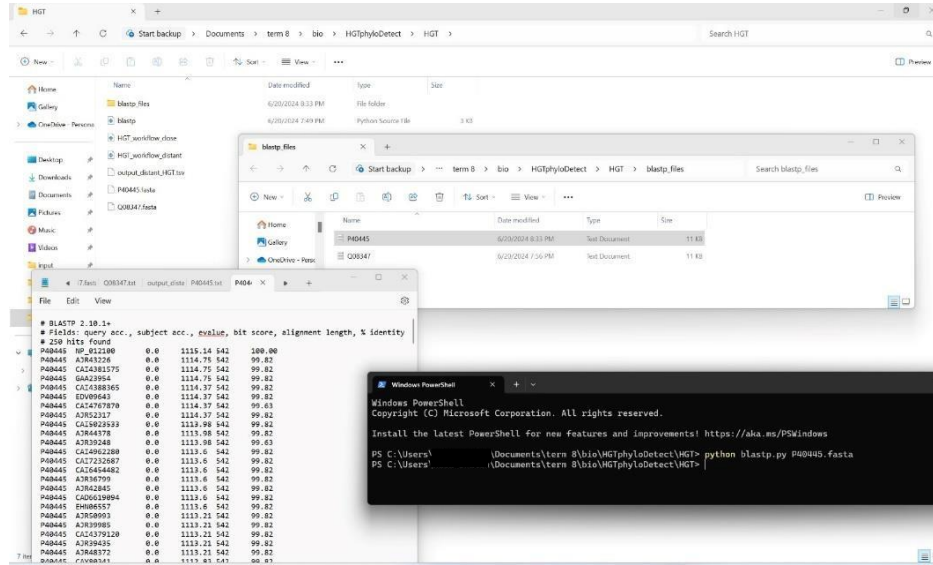


Figure (2): output text file of blastp.py(close)

4. **Step 4:** 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.

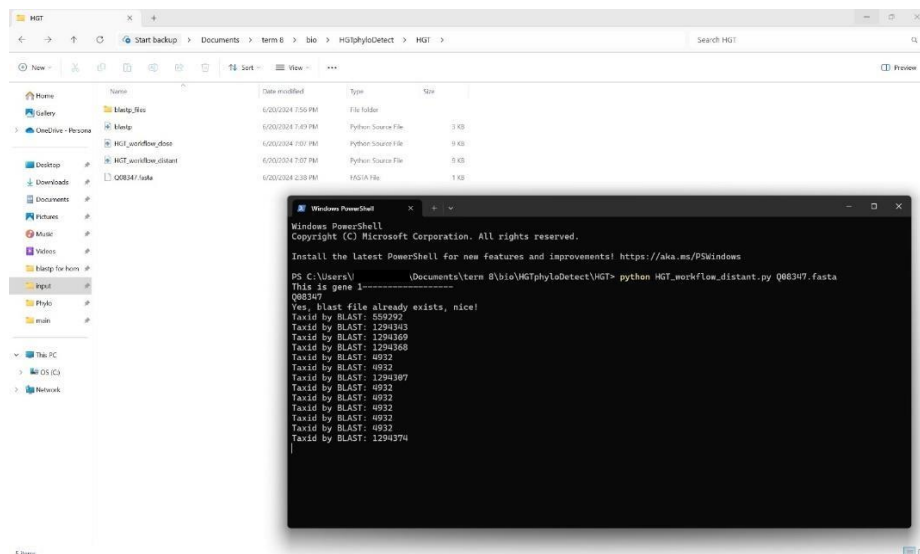


Figure (4): running distant workflow code

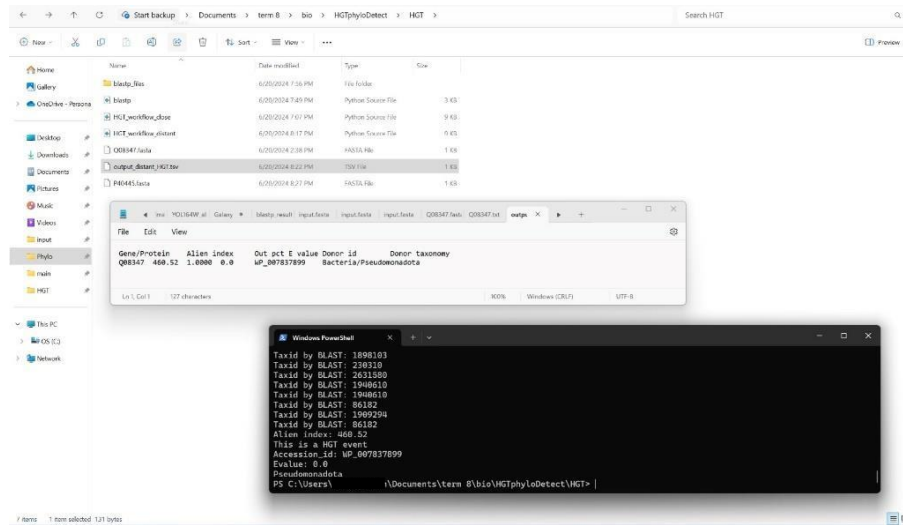


Figure (5): output of distant workflow code

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.

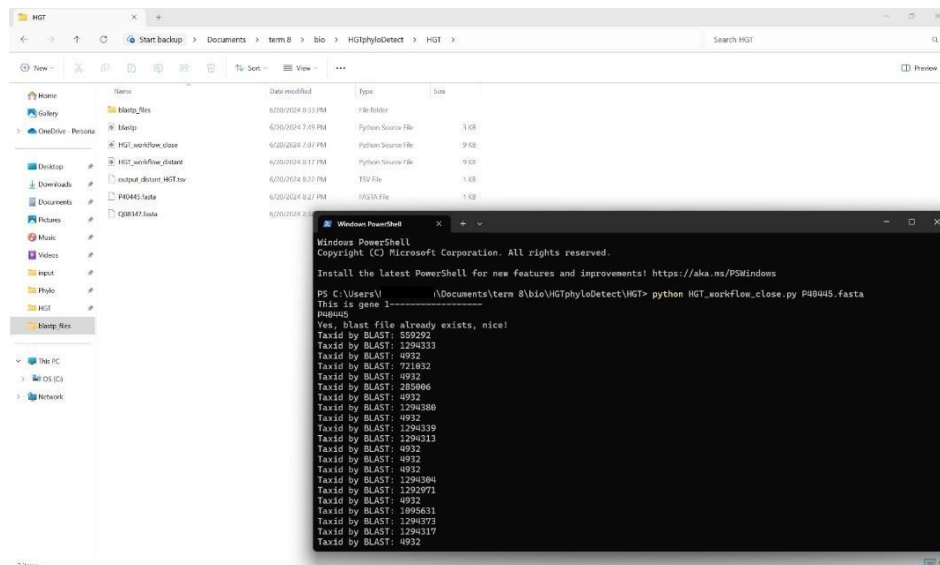


Figure (6): running of close workflow code

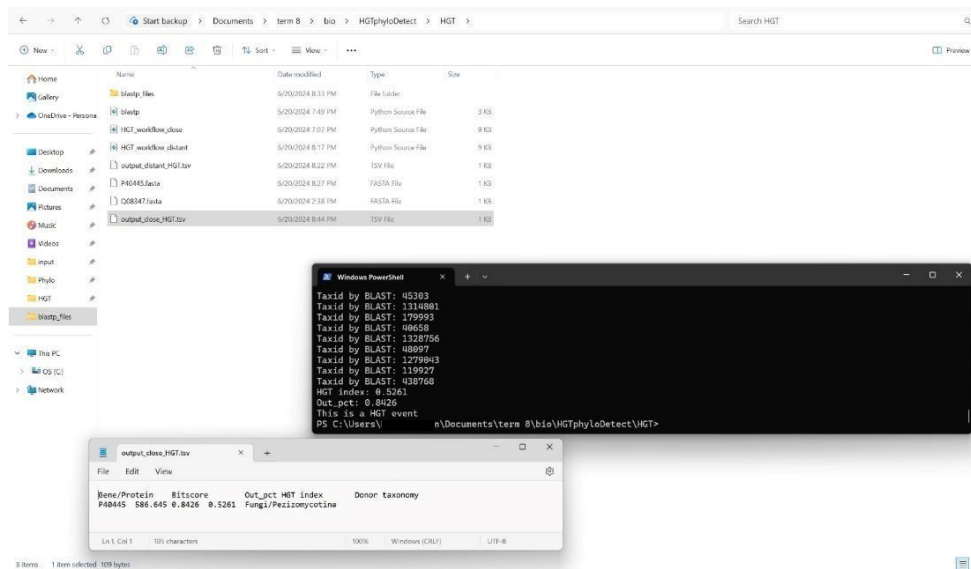


Figure (7): output of close workflow code

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`

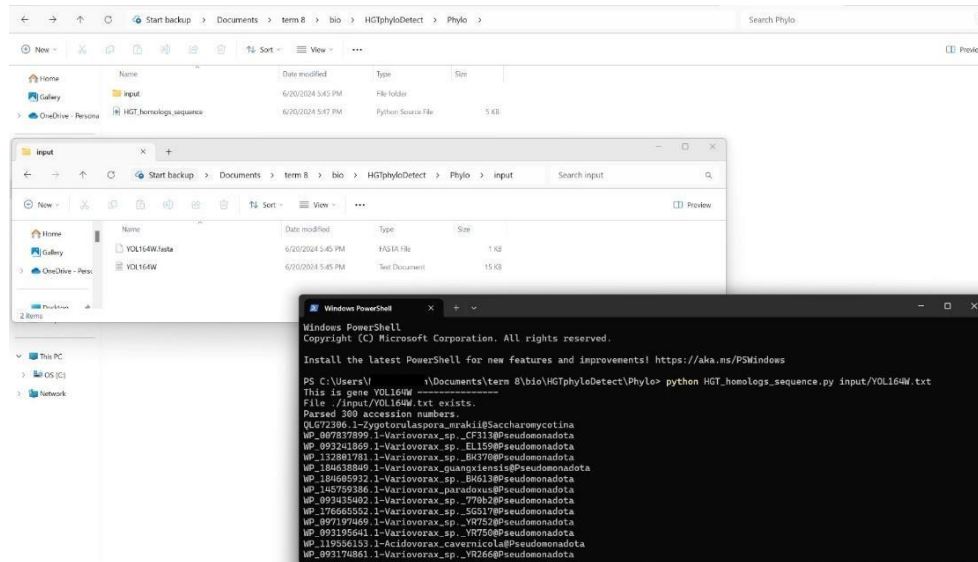


Figure (1): showing the running of the input files

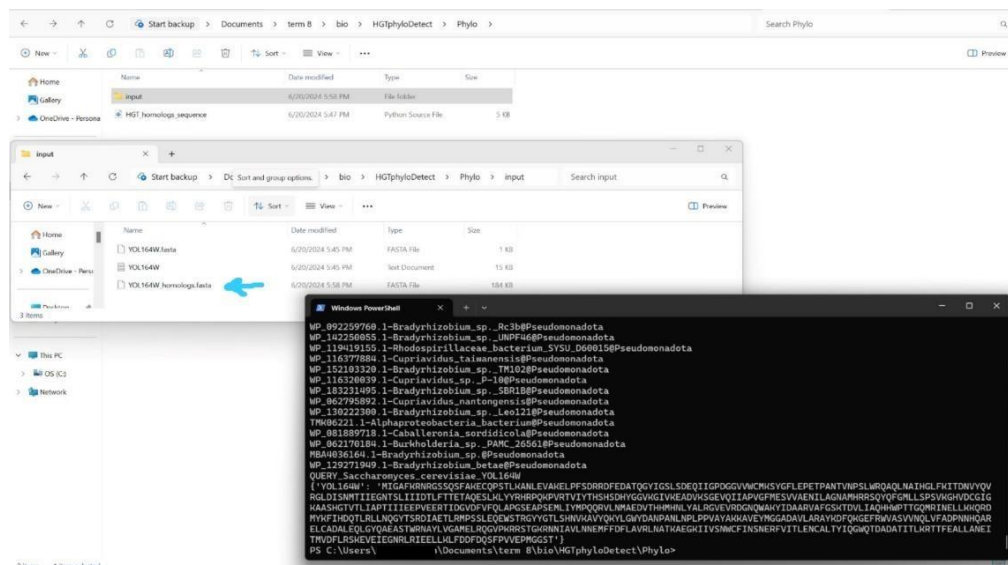


Figure (2): showcasing the output file

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

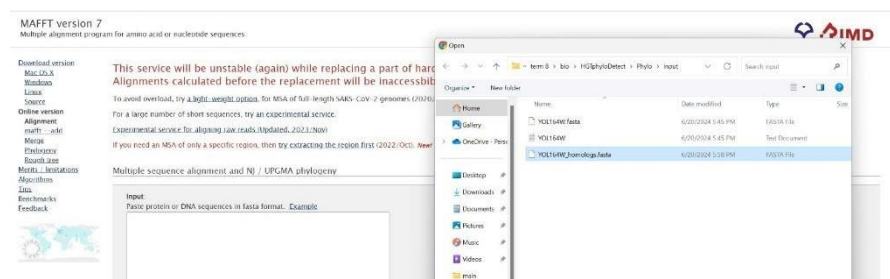


Figure (3): input file and MAFFT tool

Computing

Job id: _2406210017127495uZpZl2qgXKkC3Vjfsfnormal

Data size: 271 sequences x (714-599) aa

Command:

```
mafft --thread 8 --threadth 5 --threadit 0 --reorder --auto input > output
```

Load level of this queue:

Load level of the cluster:

Calculation status will be checked 2.4 seconds later.

[Check now](#)

[Progress \(in a new window\)](#)

[Cancel](#)

Notify when finished

[Set or change email address](#)

Figure (4): running the input file

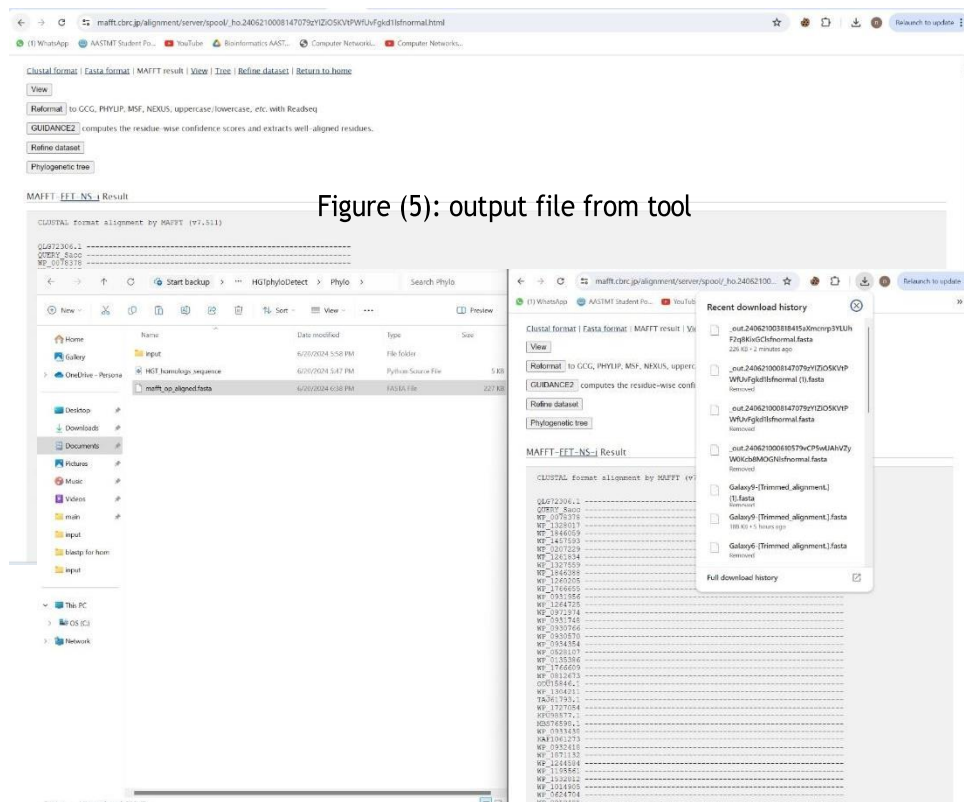


Figure (6): output renamed

- **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`

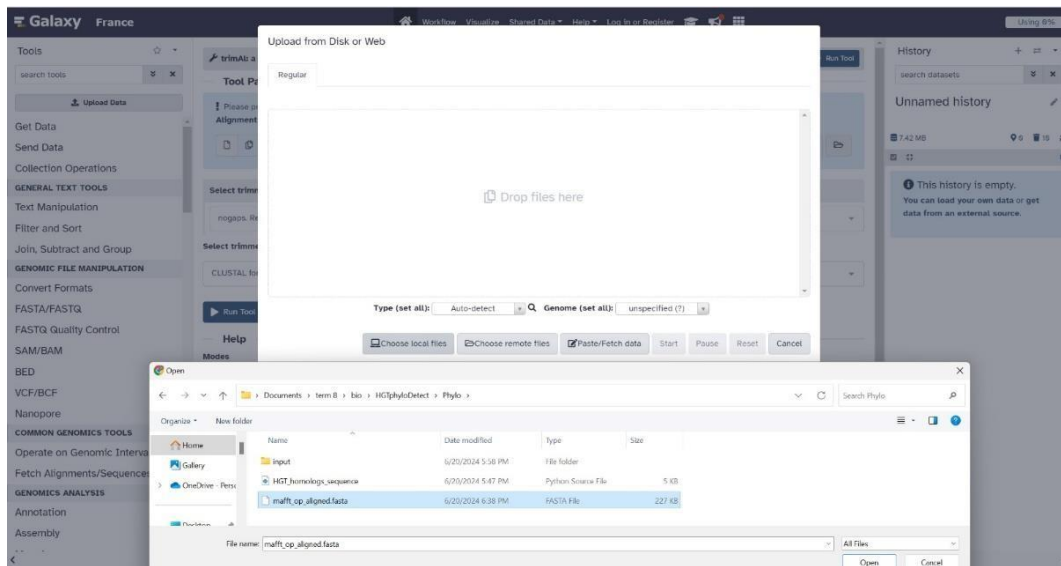


Figure (7): input file

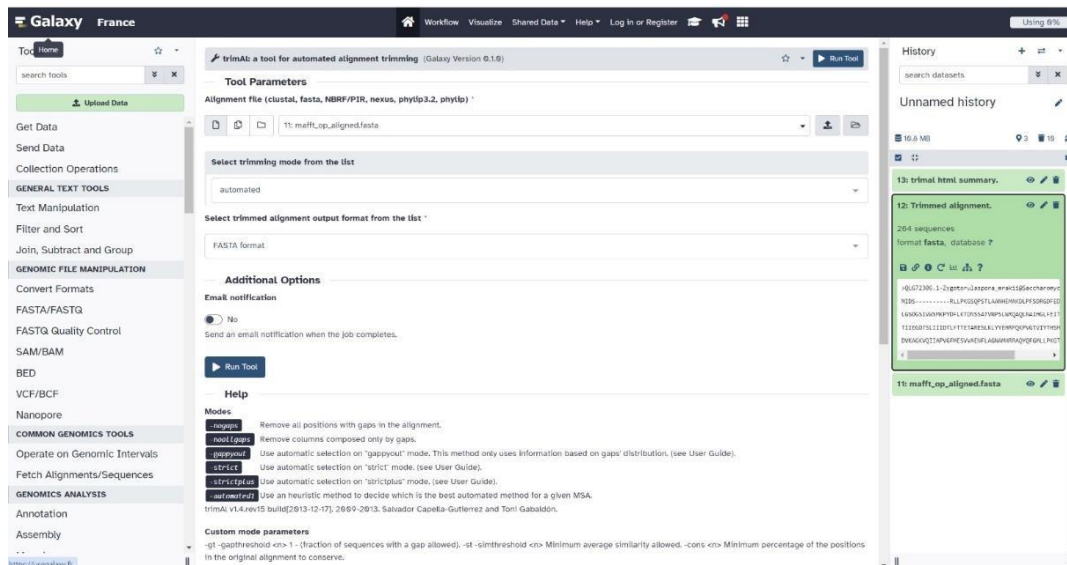


Figure (8): selecting fasta, automated 1, and producing output

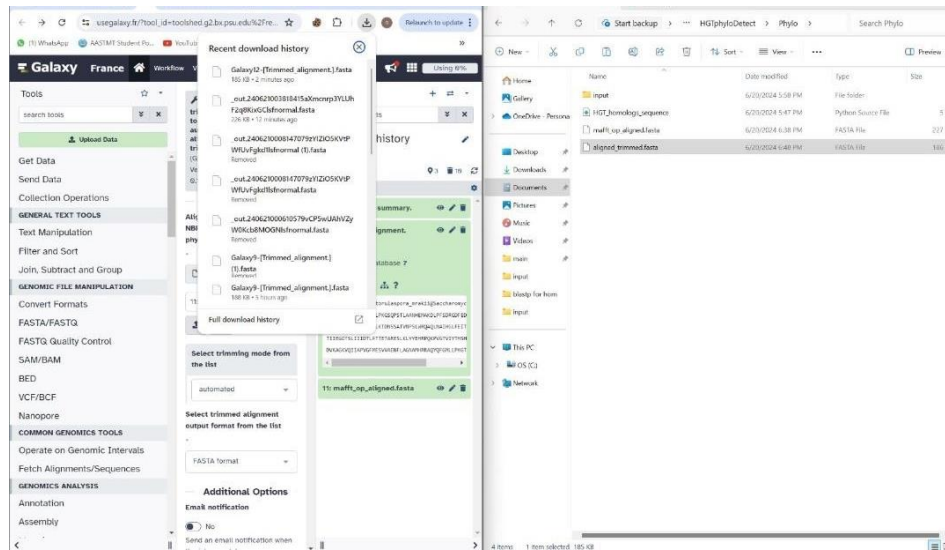


Figure (9): output file rename

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

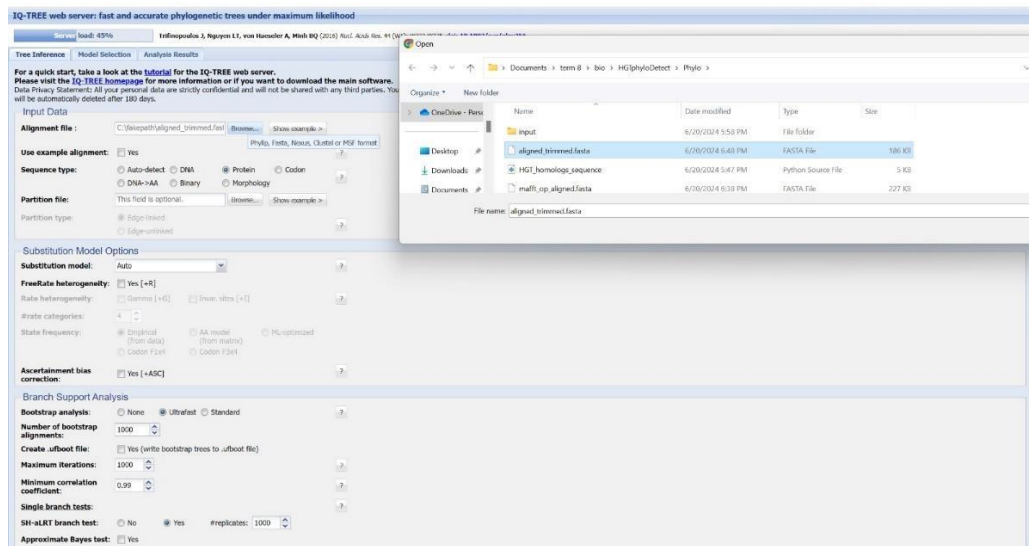


Figure (9): input file extension

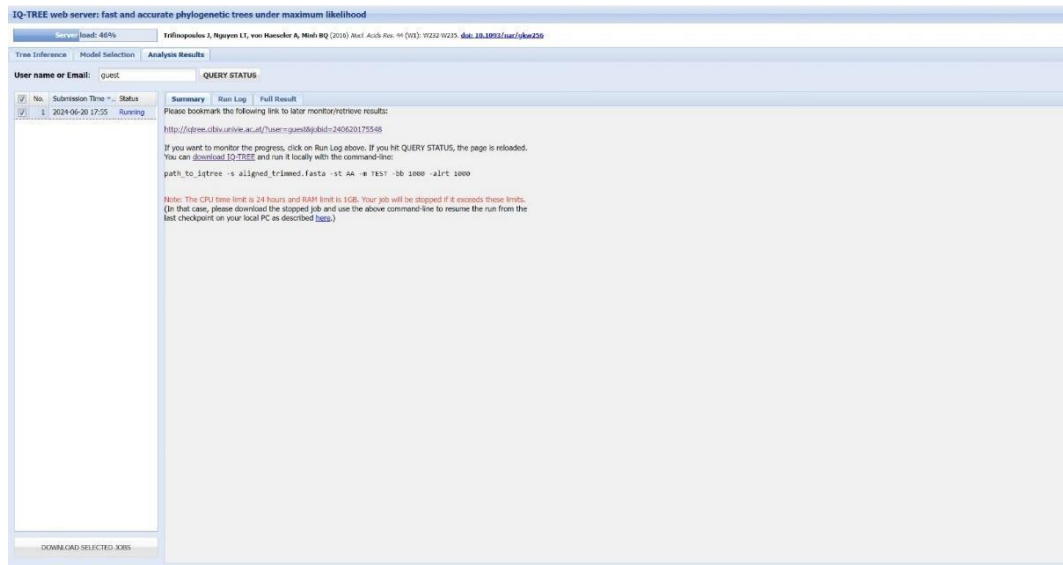


Figure (10): running

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

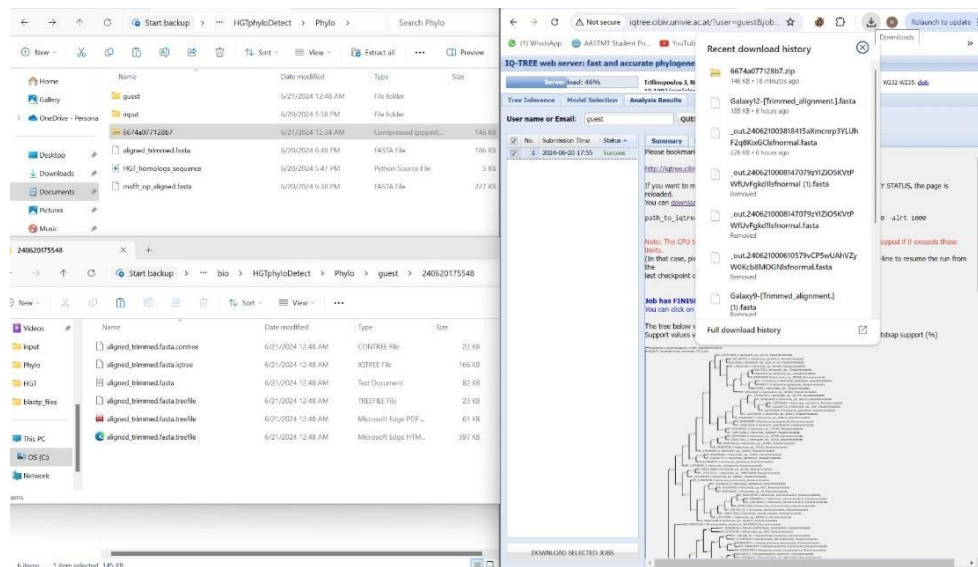


Figure (11): unzip downloaded folder of IQ-TREE output

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

- **Step 6:** 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)

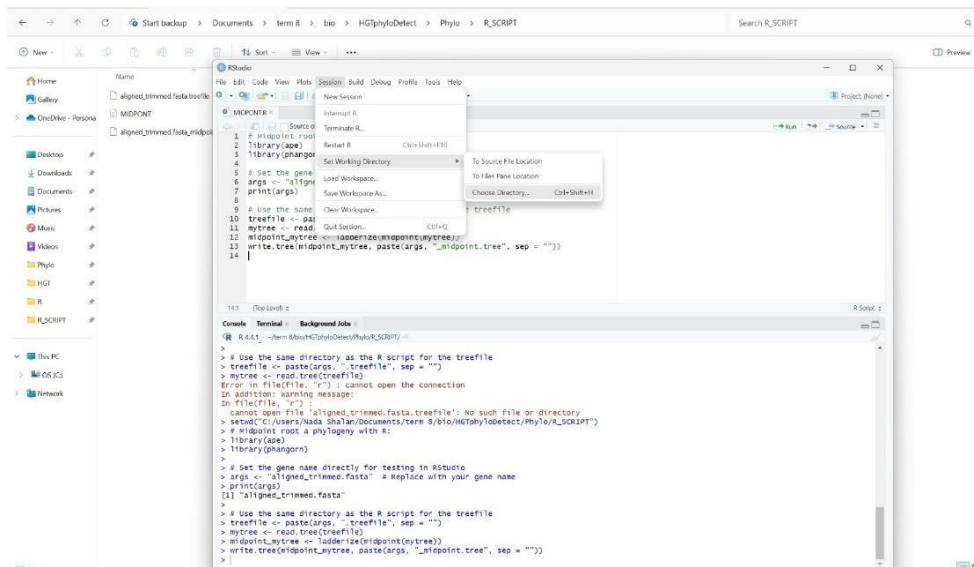


Figure (12): running R script (R studio)

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

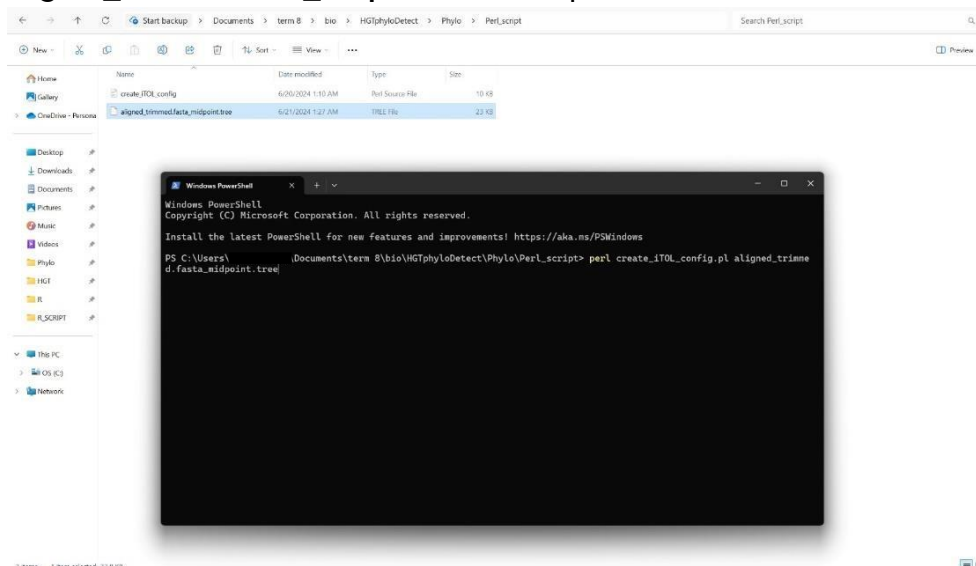


Figure (13) : running the input file

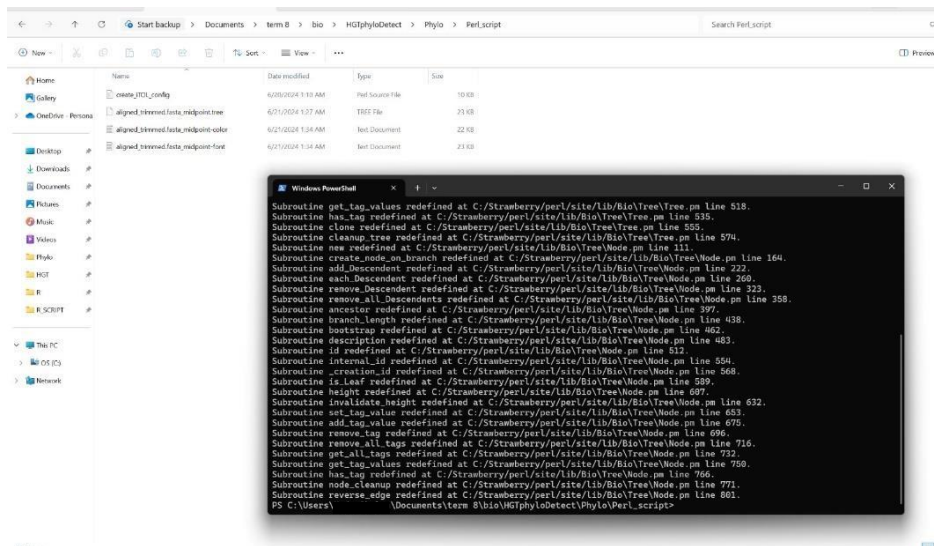


Figure (14): output file(annotations)

- **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

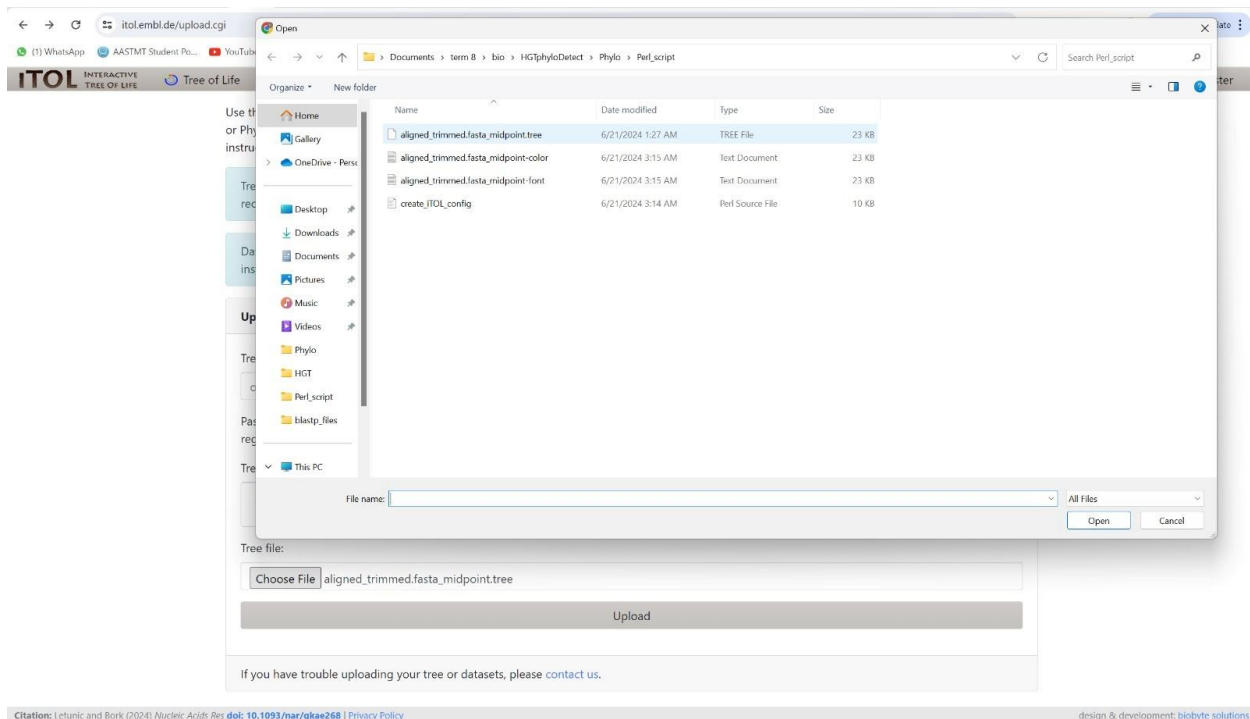


Figure (15): uploading(.tree)

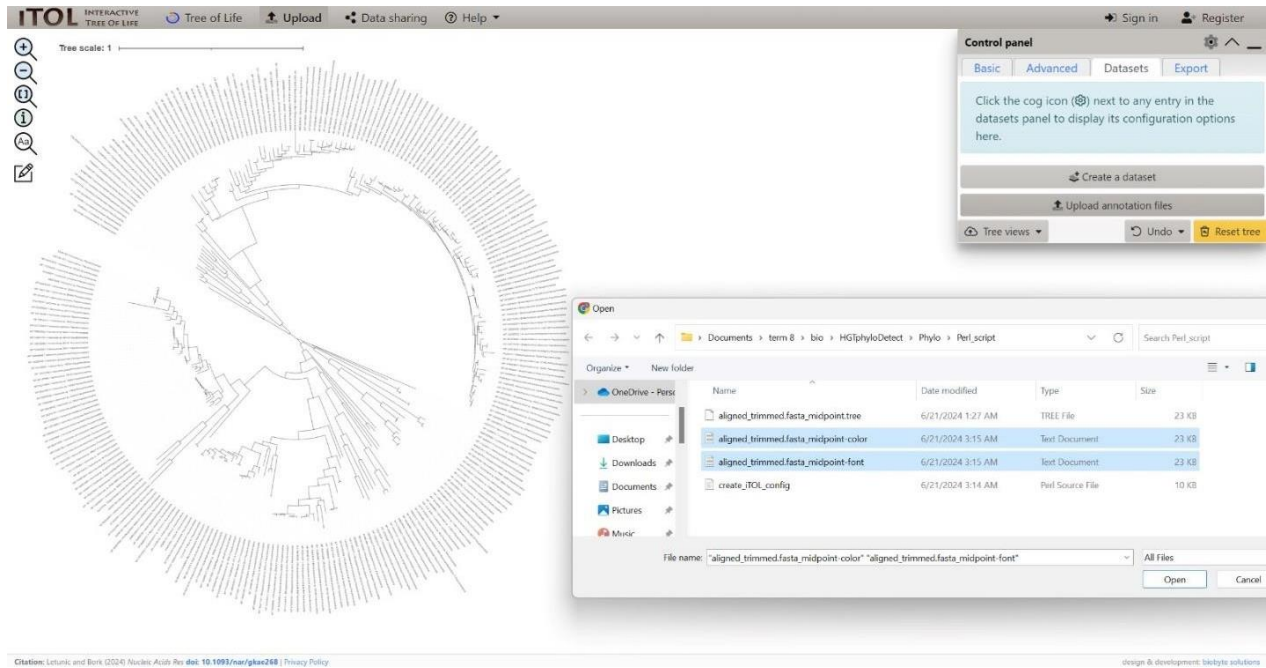


Figure (16): adding(annotations files)



Figure (16): circular output

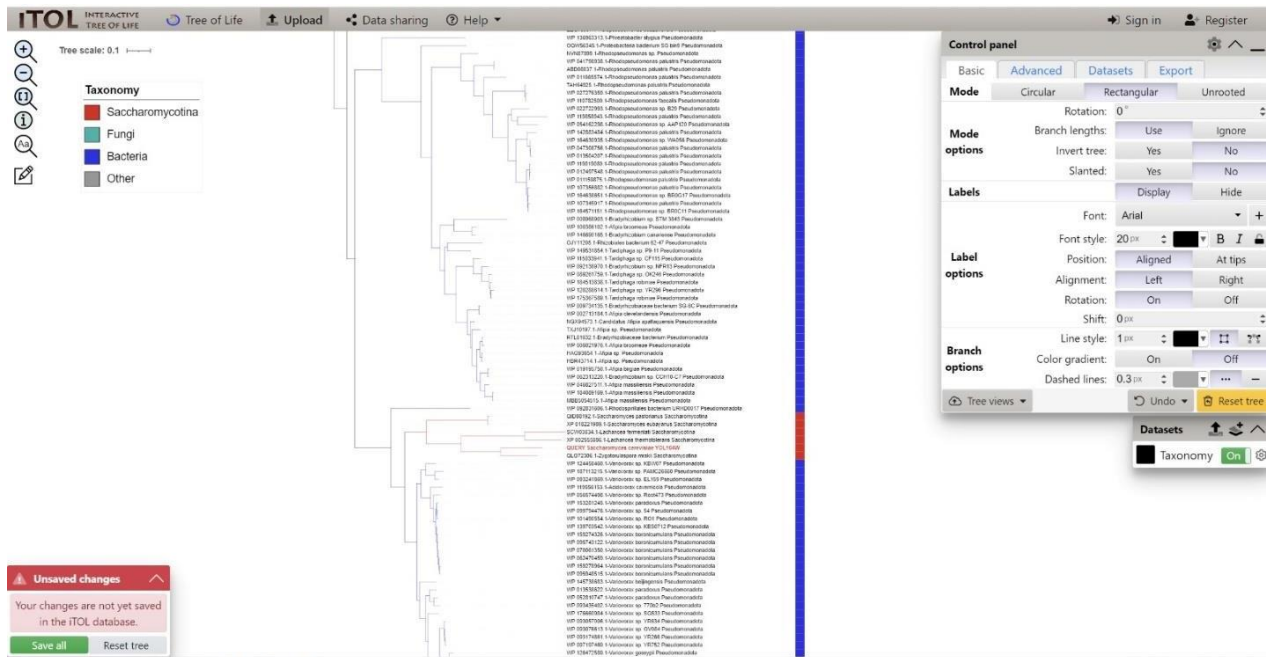


Figure (17): rectangular output