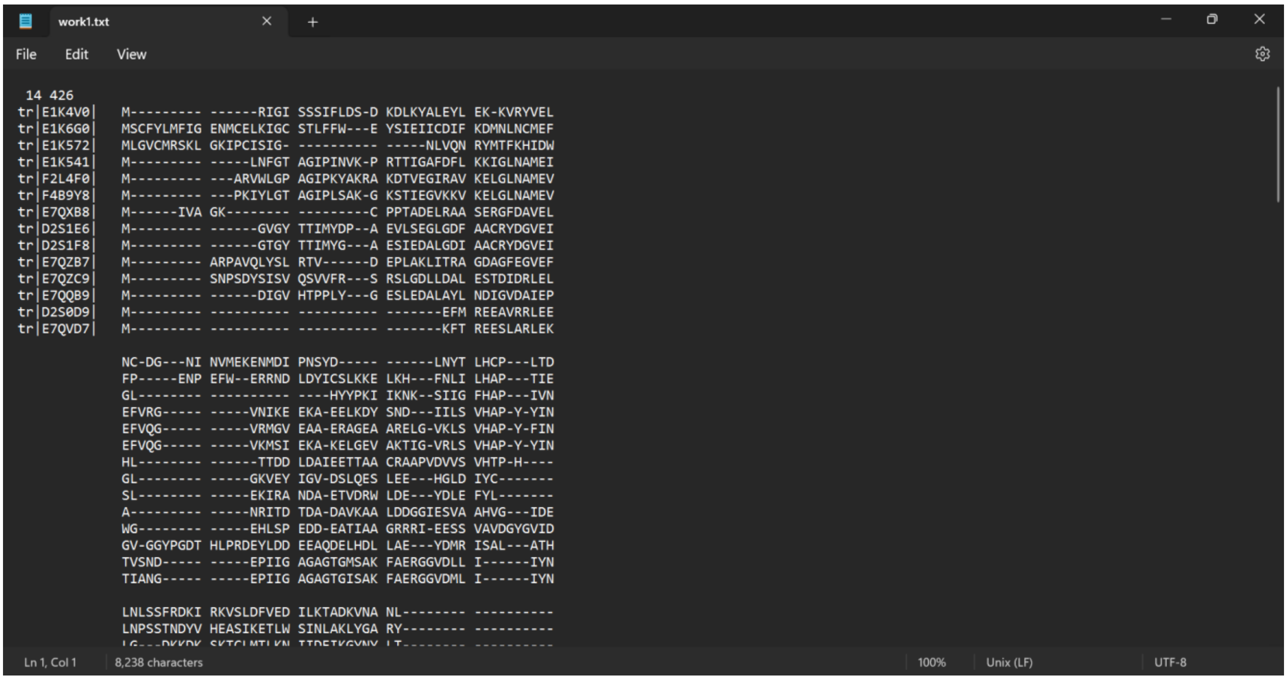
Anirudh Rao
BE21B004

BT3040 – Bioinformatics

Practical 10

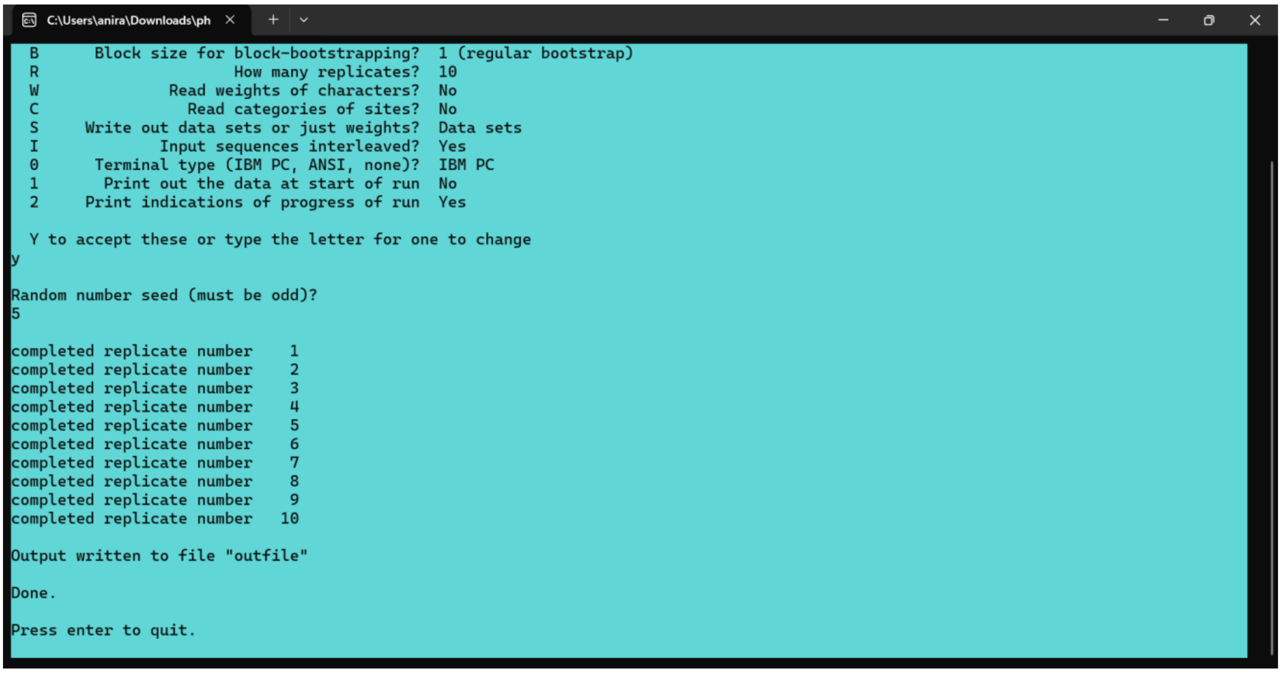
1

After downloading PHYLIP, a MSA was performed on the first set of sequences using the MAFFT webserver. The output was downloaded in PHYLIP format. This was saved in the folder containing the .exe files of PHYLIP.



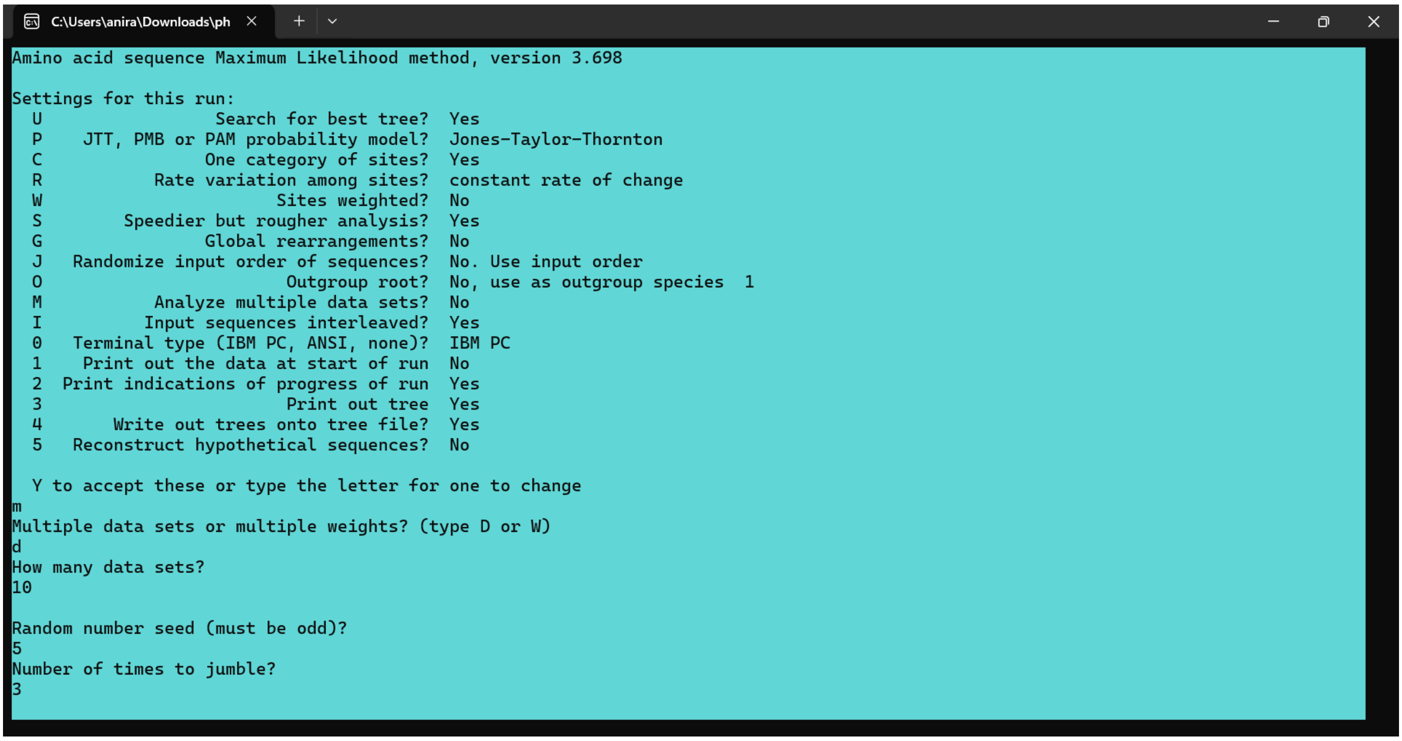
```
 14 426
tr|E1K4V0|   M--------- ------RIGI SSSIFLDS-D KDLKYALEYL EK-KVRYVEL
tr|E1K6G0|   MSCFYLMFIG ENMCELKIGC STLFFW---E YSIEIICDIF KDMNLNCMEF
tr|E1K572|   MLGVCMRSKL GKIPCISIG- ---------- -----NLVQN RYMTFKHIDW
tr|E1K541|   M--------- -----LNFGT AGIPINVK-P RTTIGAFDFL KKIGLNAMEI
tr|F2L4F0|   M--------- ---ARVWLGP AGIPKYAKRA KDTVEGIRAV KELGLNAMEV
tr|F4B9Y8|   M--------- ---PKIYLGT AGIPLSAK-G KSTIEGVKKV KELGLNAMEV
tr|E7QXB8|   M------IVA GK-------- ---------C PPTADELRAA SERGFDAVEL
tr|D2S1E6|   M--------- ------GVGY TTIMYDP--A EVLSEGLGDF AACRYDGVEI
tr|D2S1F8|   M--------- ------GTGY TTIMYG---A ESIEDALGDI AACRYDGVEI
tr|E7QZB7|   M--------- ARPAVQLYSL RTV------D EPLAKLITRA GDAGFEGVEF
tr|E7QZC9|   M--------- SNPSDYSISV QSVVFR---S RSLGDLLDAL ESTDIDRLEL
tr|E7QQB9|   M--------- ------DIGV HTPPLY---G ESLEDALAYL NDIGVDAIEP
tr|D2S0D9|   M--------- ---------- ---------- -------EFM REEAVRRLEE
tr|E7QVD7|   M--------- ---------- ---------- -------KFT REESLARLEK

             NC-DG---NI NVMEKENMDI PNSYD----- ------LNYT LHCP---LTD
             FP-----ENP EFW--ERRND LDYICSLKKE LKH---FNLI LHAP---TIE
             GL-------- ---------- ----HYYPKI IKNK--SIIG FHAP---IVN
             EFVRG----- -----VNIKE EKA-EELKDY SND---IILS VHAP-Y-YIN
             EFVQG----- -----VRMGV EAA-ERAGEA ARELG-VKLS VHAP-Y-FIN
             EFVQG----- -----VKMSI EKA-KELGEV AKTIG-VRLS VHAP-Y-YIN
             HL-------- ------TTDD LDAIEETTAA CRAAPVDVVS VHTP-H----
             GL-------- -----GKVEY IGV-DSLQES LEE---HGLD IYC-------
             SL-------- -----EKIRA NDA-ETVDRW LDE---YDLE FYL-------
             A--------- -----NRITD TDA-DAVKAA LDDGGIESVA AHVG---IDE
             WG-------- -----EHLSP EDD-EATIAA GRRRI-EESS VAVDGYGVID
             GV-GGYPGDT HLPRDEYLDD EEAQDELHDL LAE---YDMR ISAL---ATH
             TVSND----- -----EPIIG AGAGTGMSAK FAERGGVDLL I------IYN
             TIANG----- -----EPIIG AGAGTGISAK FAERGGVDML I------IYN

             LNLSSFRDKI RKVSLDFVED ILKTADKVNA NL-------- ----------
             LNPSSTNDYV HEASIKETLW SINLAKLYGA RY-------- ----------
             LG---DKKDK SKTCLMTLKN TTDETKGVNY LT-------- ----------
```

Next, bootstrapping was performed using PHYLIP's `seqboot` program.



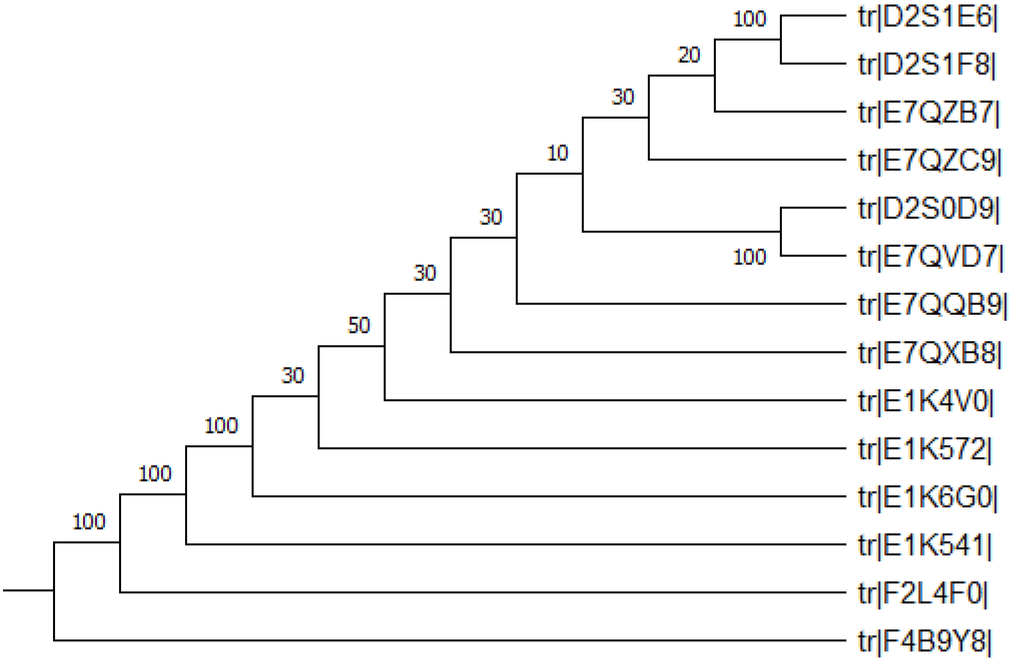
```
  B      Block size for block-bootstrapping?  1 (regular bootstrap)
  R                     How many replicates?  10
  W              Read weights of characters?  No
  C                Read categories of sites?  No
  S     Write out data sets or just weights?  Data sets
  I          Input sequences interleaved?  Yes
  0      Terminal type (IBM PC, ANSI, none)?  IBM PC
  1       Print out the data at start of run  No
  2     Print indications of progress of run  Yes

  Y to accept these or type the letter for one to change
y

Random number seed (must be odd)?
5

completed replicate number    1
completed replicate number    2
completed replicate number    3
completed replicate number    4
completed replicate number    5
completed replicate number    6
completed replicate number    7
completed replicate number    8
completed replicate number    9
completed replicate number   10

Output written to file "outfile"

Done.

Press enter to quit.
```

The outfile produced by this is used as input for the `proml` program, which computes the phylogenetic tree based on maximum likelihood.

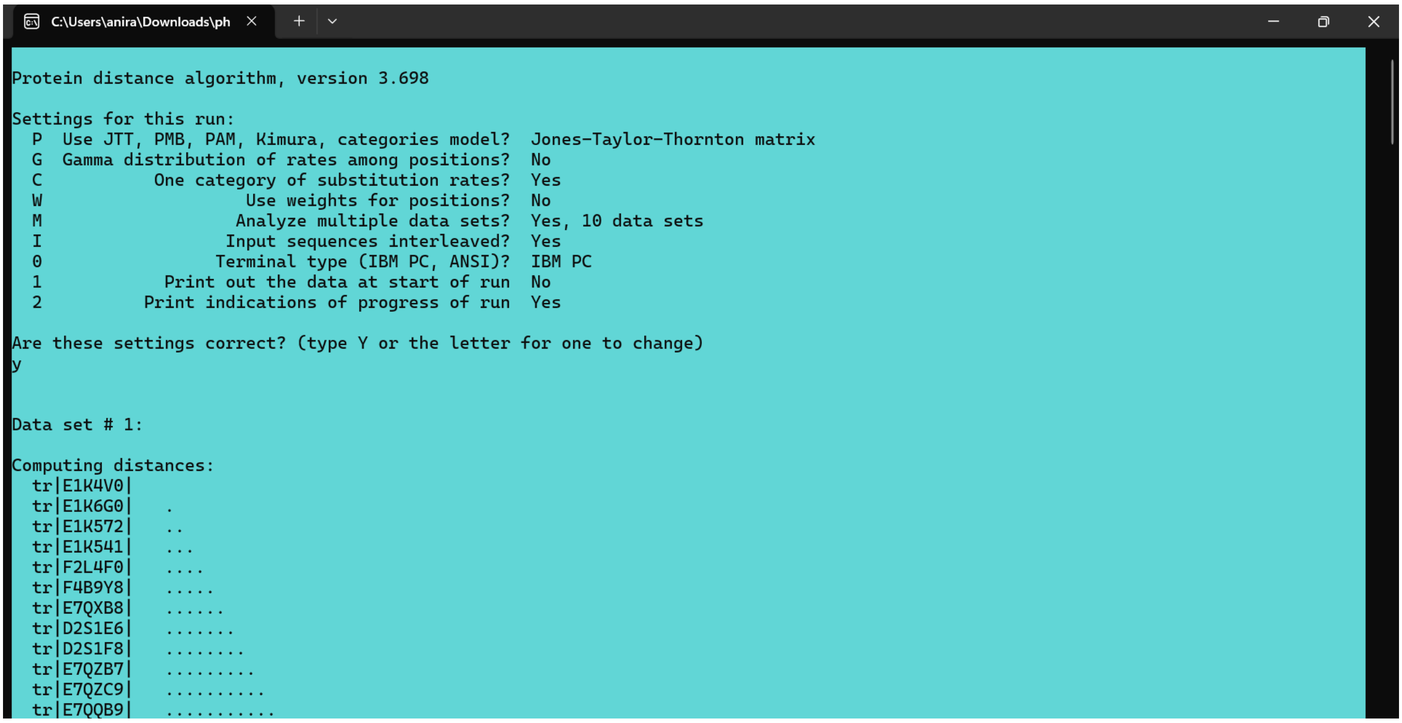


```
Amino acid sequence Maximum Likelihood method, version 3.698

Settings for this run:
  U                 Search for best tree?  Yes
  P    JTT, PMB or PAM probability model?  Jones-Taylor-Thornton
  C                One category of sites?  Yes
  R           Rate variation among sites?  constant rate of change
  W                       Sites weighted?  No
  S        Speedier but rougher analysis?  Yes
  G                Global rearrangements?  No
  J   Randomize input order of sequences?  No. Use input order
  O                        Outgroup root?  No, use as outgroup species  1
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  IBM PC
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4       Write out trees onto tree file?  Yes
  5   Reconstruct hypothetical sequences?  No

  Y to accept these or type the letter for one to change
m
Multiple data sets or multiple weights? (type D or W)
d
How many data sets?
10

Random number seed (must be odd)?
5
Number of times to jumble?
3
```

The output of this is fed into the `consense` program to obtain the consensus tree. MEGA-X is then used to visualise the tree.

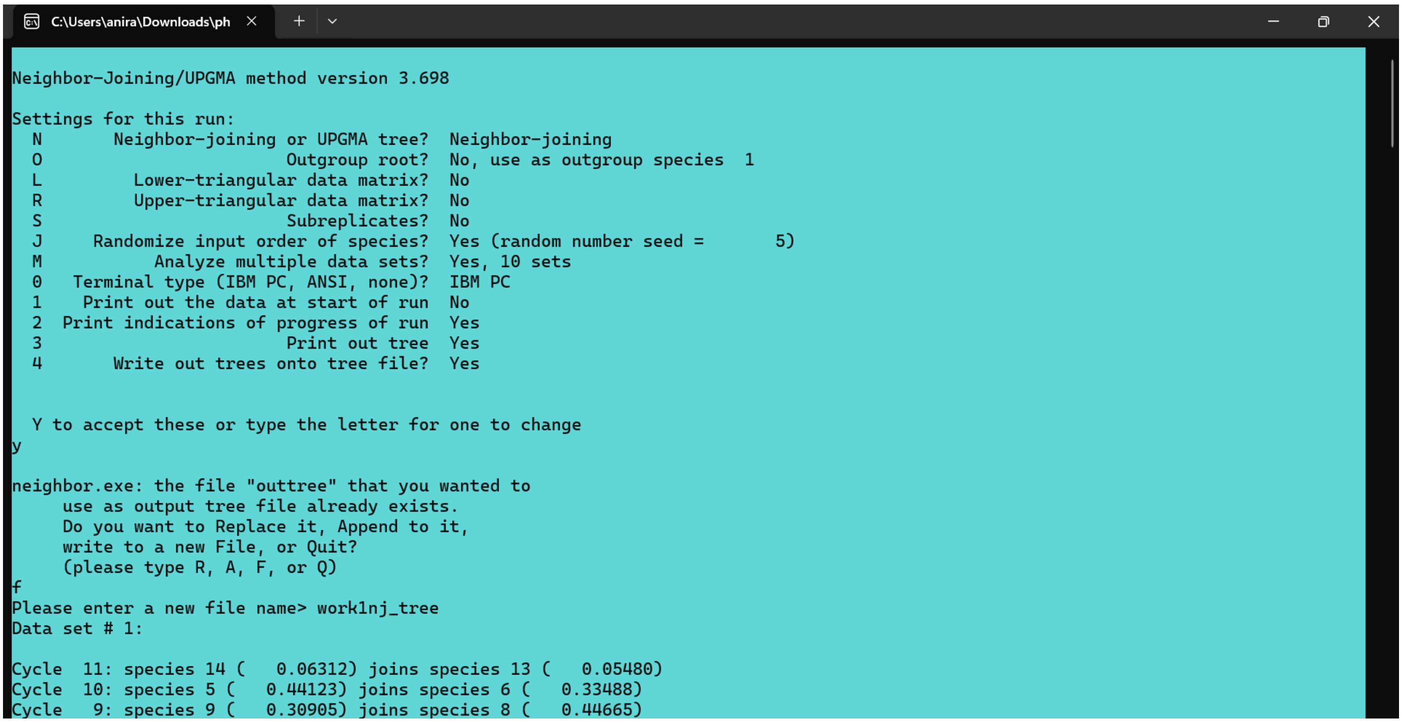


A similar tree is constructed using the `protdist` program.



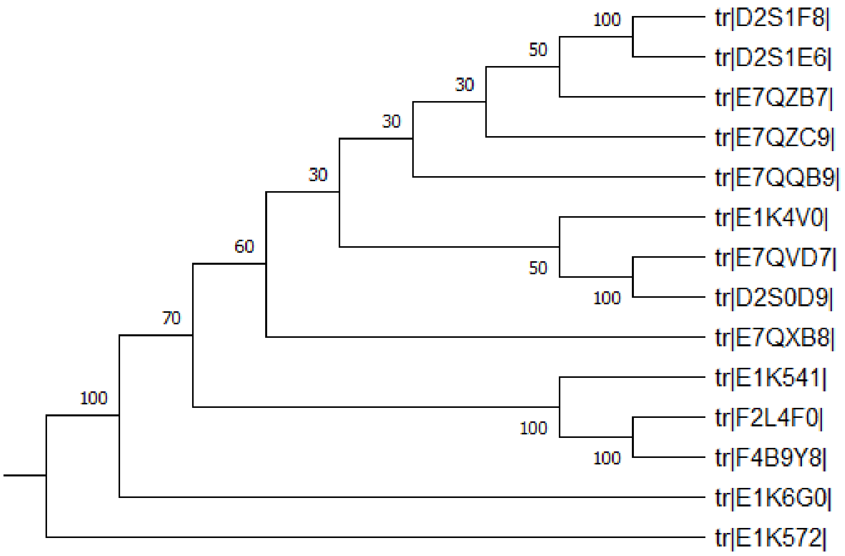
```
Protein distance algorithm, version 3.698

Settings for this run:
  P  Use JTT, PMB, PAM, Kimura, categories model?  Jones-Taylor-Thornton matrix
  G  Gamma distribution of rates among positions?  No
  C           One category of substitution rates?  Yes
  W                    Use weights for positions?  No
  M                   Analyze multiple data sets?  Yes, 10 data sets
  I                  Input sequences interleaved?  Yes
  0                 Terminal type (IBM PC, ANSI)?  IBM PC
  1            Print out the data at start of run  No
  2          Print indications of progress of run  Yes

Are these settings correct? (type Y or the letter for one to change)
y


Data set # 1:

Computing distances:
  tr|E1K4V0|
  tr|E1K6G0|   .
  tr|E1K572|   ..
  tr|E1K541|   ...
  tr|F2L4F0|   ....
  tr|F4B9Y8|   .....
  tr|E7QXB8|   ......
  tr|D2S1E6|   .......
  tr|D2S1F8|   ........
  tr|E7QZB7|   .........
  tr|E7QZC9|   ..........
  tr|E7QQB9|   ...........
```

The output of this is fed into the `neighbor` program.

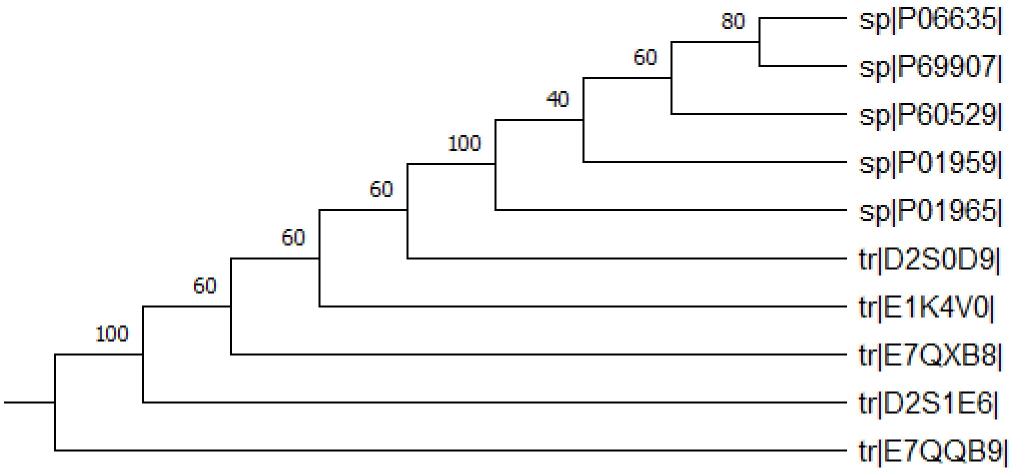


```
Neighbor-Joining/UPGMA method version 3.698

Settings for this run:
  N       Neighbor-joining or UPGMA tree?  Neighbor-joining
  O                        Outgroup root?  No, use as outgroup species  1
  L         Lower-triangular data matrix?  No
  R         Upper-triangular data matrix?  No
  S                        Subreplicates?  No
  J     Randomize input order of species?  Yes (random number seed =       5)
  M           Analyze multiple data sets?  Yes, 10 sets
  0   Terminal type (IBM PC, ANSI, none)?  IBM PC
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4       Write out trees onto tree file?  Yes


  Y to accept these or type the letter for one to change
y

neighbor.exe: the file "outtree" that you wanted to
     use as output tree file already exists.
     Do you want to Replace it, Append to it,
     write to a new File, or Quit?
     (please type R, A, F, or Q)
f
Please enter a new file name> work1nj_tree
Data set # 1:

Cycle  11: species 14 (   0.06312) joins species 13 (   0.05480)
Cycle  10: species 5 (   0.44123) joins species 6 (   0.33488)
Cycle   9: species 9 (   0.30905) joins species 8 (   0.44665)
```

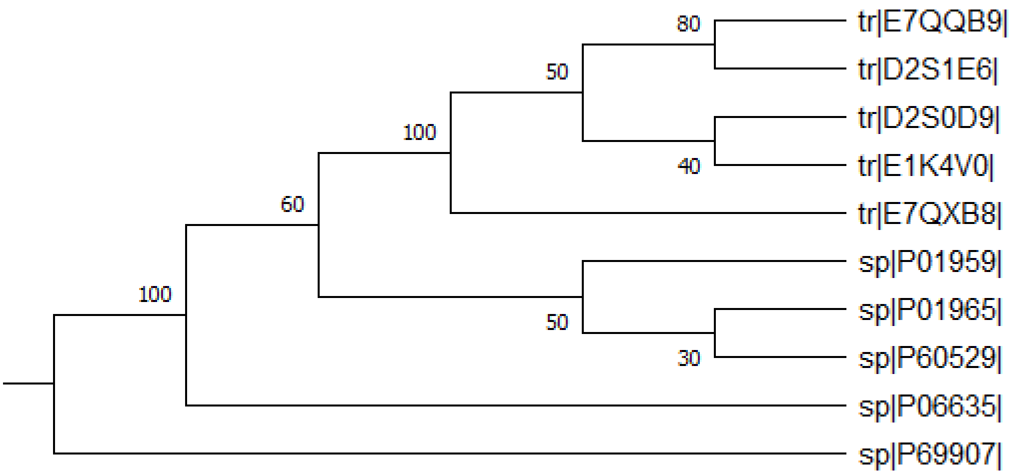
The consensus tree for this is then produced using the `consense` program and visualised using MEGA-X.



The same steps are repeated for the second set of sequences. The consensus tree based on `proml` is:



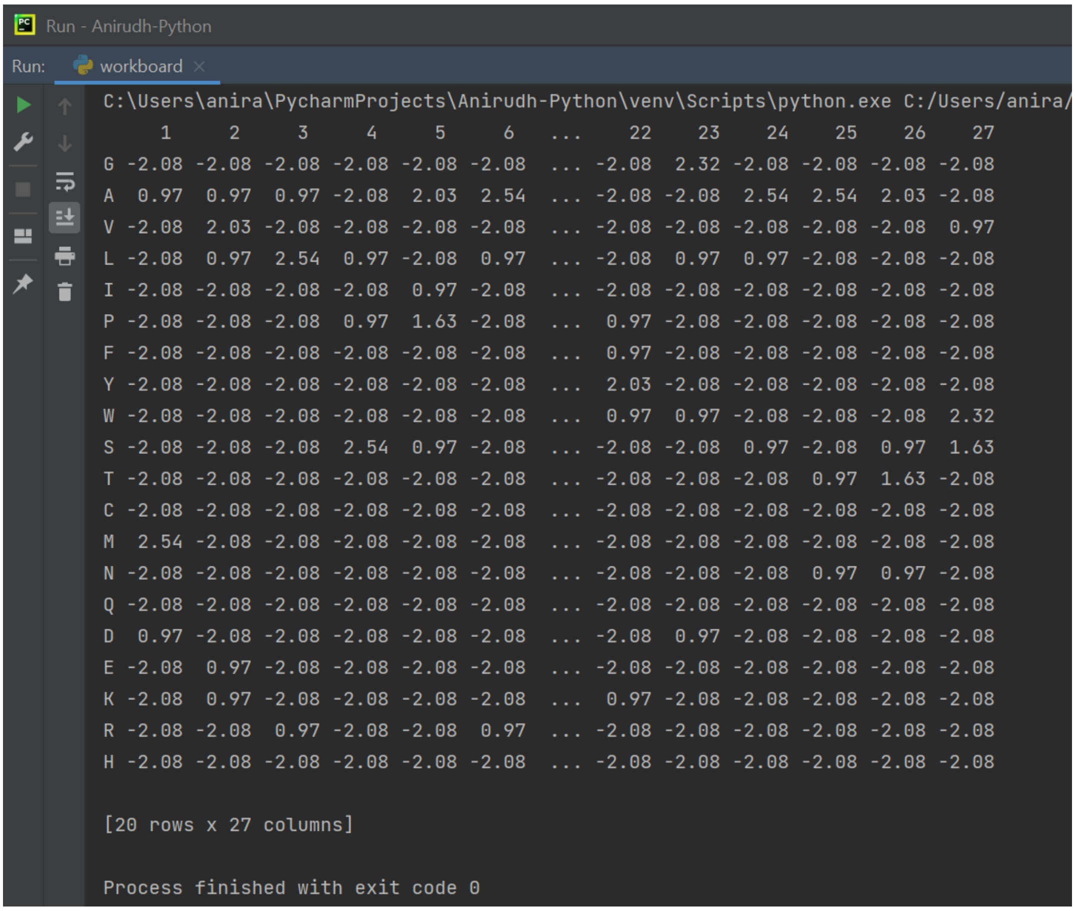
The consensus tree based on `protdist` and `neighbor` is:



The Python code to compute the weight matrix of the given sequences is shown below:

```python
import numpy as np
import pandas as pd

alignment = [
    "MVLSPADKTNVKGKVGAHAGEYGAAAW",
    "MKRLPADPPCVKGKVKAKAGDYGATTW",
    "MALSAADKTNVKSKVGGHAGEYGAATS",
    "MVLSAADKTNVKSKAGGNAGEWWAAAW",
    "MVLSAADKTNVKSKVLANAGEFGAAAW",
    "ALLPIRTTYHKKCASGHIPEEKDLNNV",
    "DEASSLKGHHIKKLEADALLIPLSASS"]
residues = ["G", "A", "V", "L", "I", "P", "F", "Y", "W", "S", "T", "C", "M", "N", "Q", "D", "E", "K", "R", "H"]
alignment_matrix = {residue: [0] * len(alignment[0]) for residue in residues}
for i in range(len(alignment[0])):
    for seq in alignment:
        alignment_matrix[seq[i]][i] += 1

N = len(alignment)
p = 1 / len(residues)
weight_matrix = {
    residue: [round(np.log((alignment_matrix[residue][i] + p) / (p * (N + 1))), 2) for i in range(len(alignment[0]))]
    for residue in residues}
df = pd.DataFrame(weight_matrix).transpose()
df.columns = range(1, len(alignment[0]) + 1, 1)
print(df)
```

The output of this is:



```
C:\Users\anira\PycharmProjects\Anirudh-Python\venv\Scripts\python.exe C:/Users/anira/
     1     2     3     4     5     6  ...    22    23    24    25    26    27
G -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08  2.32 -2.08 -2.08 -2.08 -2.08
A  0.97  0.97  0.97 -2.08  2.03  2.54  ... -2.08 -2.08  2.54  2.54  2.03 -2.08
V -2.08  2.03 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08 -2.08 -2.08  0.97
L -2.08  0.97  2.54  0.97 -2.08  0.97  ... -2.08  0.97  0.97 -2.08 -2.08 -2.08
I -2.08 -2.08 -2.08 -2.08  0.97 -2.08  ... -2.08 -2.08 -2.08 -2.08 -2.08 -2.08
P -2.08 -2.08 -2.08  0.97  1.63 -2.08  ...  0.97 -2.08 -2.08 -2.08 -2.08 -2.08
F -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ...  0.97 -2.08 -2.08 -2.08 -2.08 -2.08
Y -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ...  2.03 -2.08 -2.08 -2.08 -2.08 -2.08
W -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ...  0.97  0.97 -2.08 -2.08 -2.08  2.32
S -2.08 -2.08 -2.08  2.54  0.97 -2.08  ... -2.08 -2.08  0.97 -2.08  0.97  1.63
T -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08  0.97  1.63 -2.08
C -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08 -2.08 -2.08 -2.08
M  2.54 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08 -2.08 -2.08 -2.08
N -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08  0.97  0.97 -2.08
Q -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08 -2.08 -2.08 -2.08
D  0.97 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08  0.97 -2.08 -2.08 -2.08 -2.08
E -2.08  0.97 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08 -2.08 -2.08 -2.08
K -2.08  0.97 -2.08 -2.08 -2.08 -2.08  ...  0.97 -2.08 -2.08 -2.08 -2.08 -2.08
R -2.08 -2.08  0.97 -2.08 -2.08  0.97  ... -2.08 -2.08 -2.08 -2.08 -2.08 -2.08
H -2.08 -2.08 -2.08 -2.08 -2.08 -2.08  ... -2.08 -2.08 -2.08 -2.08 -2.08 -2.08

[20 rows x 27 columns]

Process finished with exit code 0
```