Phylogenetic Tree

Malay (malay@uab.edu)

Nov 9, 2018

Contents

1 Creating a distance tree

We will use `phylip` to create a distance tree from the p53 protein alignment that we created eariler using `muscle`. We can use a script to convert an existing aligment into `phylip` format (I wrote such a script called `fasta2phylip.pl`) or we can realign the sequences using `muscle`, but this time we can directly output in `phylip` format. We will use the following program from the `phylip` package:

1. `protdist` creates a distance matrix from an alignment
2. `neighbor` will create a neighbor-joining tree

Remember, all programs in phylip will take an `infile` and will create an `outfile`.

1.1 Creating the alignment

Copy the fasta file containing the p53 sequences into current directory. `muscle` can output two different `phylip` formats. We will use "interleaved". Note, we could have output several formats from `muscle` simultaneously during alignment, like this:

```
muscle -in p53.fas -fastaout p53.afa -phyiout p53.phy
```

1.2 Get `phylip`

```
wget http://evolution.gs.washington.edu/phylip/download/phylip-3.696.tar.gz
tar -xvzf phylip-3.696.tar.gz
cd phylip-3.696/src
make -f Makefile.unx install
cd ../exe
export PATH=$PATH:`pwd`
```

1.3 Running `protdist`

```
cp p53.phy infile
protdist
```

This will create the distance matrix in a file `outfile`.

1.4 Running `neighbor`

```
rm infile
mv outfile infile
neighbor
```

The tree will be produced in a file called `outtree`:

```
cat outtree
(gi|1890836:0.06010,(((gi|1204070:0.00000,gi|3328472:0.00001):0.02029,
gi|1140518:0.02400):0.06893,(gi|2884992:0.11207,(gi|5097897:0.09463,
(gi|4258767:0.37349,gi|5005442:0.34037):0.24453):0.00796):0.01474):0.10408,gi|1487472:0.07587);
```

This is a newick format tree.

1.5 Viewing the tree

To view the tree we can use `FigTree`. You can download it from here:

http://tree.bio.ed.ac.uk/software/figtree/

Download and run the software as

```
java -jar figtree.jar
```

1.6 Bootstrapping

To get the statistical measurement of tree, we can use "bootstrapping". We have to start with the raw `phylip` file.

```
cp p53.phy infile
seqboot
rm infile
mv outfile infile
```

Now follow the exact step described earlier. Don't forget to change the option **"Multple data sets"**. The treefile will now have 100 trees (if you have used 100 bootstrap datasets).

After you have created 100 trees. Run `consense` to get a consensus tree:

```
mv outtree intree
consense
```

2 Maximum likelihood tree using RAxML

RAxML is a very efficient maximum likelihood phylogenetic analysis software that can run on a HPC settings. The software is generally used for very large datasets.

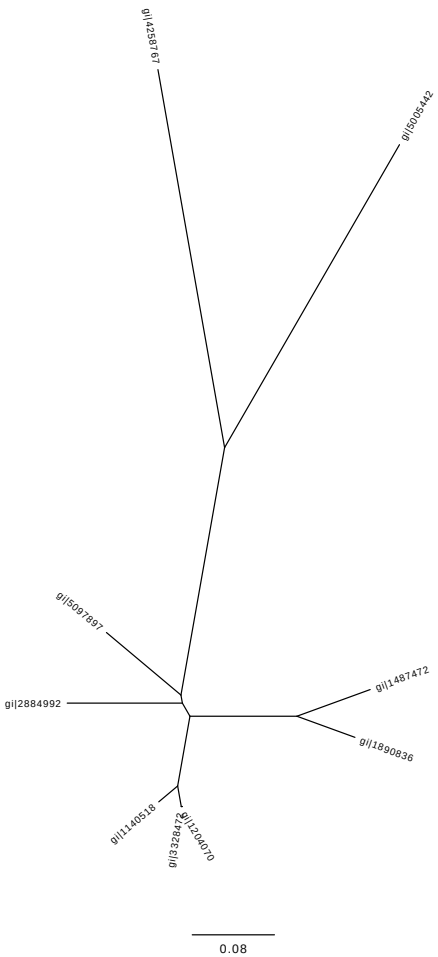

Figure 1: Tree generated using `neighbor`

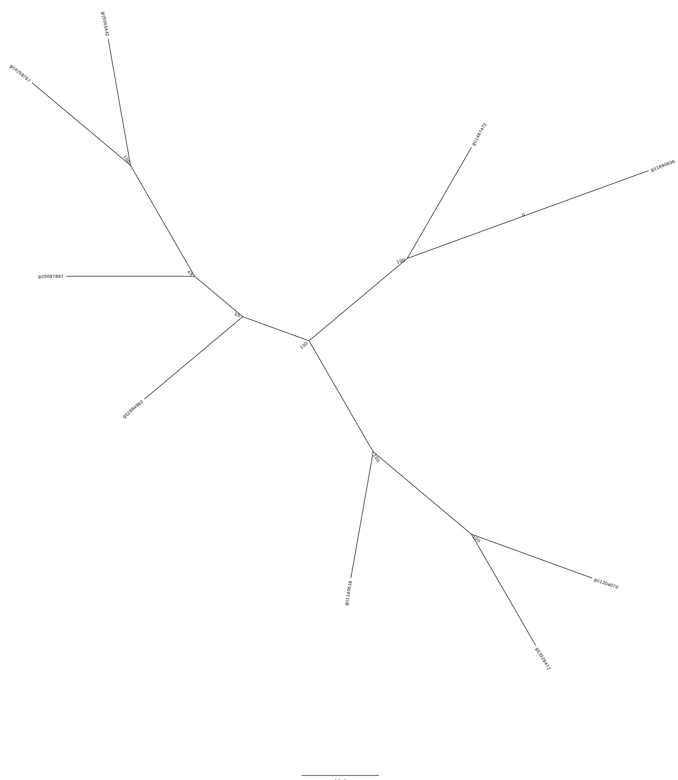

Figure 2: Bootstrap consensus tree using `consense`.

Download RAxML from https://github.com/stamatak/standard-RAxML. There are several way to compile RAxML. For use with our virtual machine we will use a very simple option.

```
git clone https://github.com/stamatak/standard-RAxML.git
cd standard-RAxML/
make -f Makefile.gcc
rm *.o
```

RAxML requires specification of a model. For nucleotide sequence the best option GTRGAMMA. For protein sequence any of the JTT models are fine. They are expanded PAM matrices.

But we can tell RAXML for find the best model for us.

```
standard-RAxML/raxmlHPC -m PROTGAMMAAUTO -s p53.afa -p 12345 -n p53
```

This will create the tree as `RAxML_bestTree.p53`.

We can also run a bootstrap analysis:

```
standard-RAxML/raxmlHPC -m PROTGAMMAJTT -s p53.afa -p 12345 -# 10 -f a -n p53_c -x 12345
```

Unlike `consense` in `phylip`. RAxML already geneartes a tree will bootstrap values already included. This is the file `RAxML_bipartitions.p53_c` How nice!