Introduction to molecular phylogenetic reconstruction with the ETE3 Toolkit

Frank Aylward

Abstract

This is a short tutorial on how to get started analyzing FASTA files via the command line.

Code is intended for use on an Ubuntu 16.04 LTS OS, but it may work on other Unix or Unix-like systems.

Here we will use the ETE3 toolkit, which is a very nice tool for phylogenetic reconstruction. The main website is here: http://etetoolkit.org/

We will also look at the EggNOG database, which is a very useful database of protein families: http://eggnogdb.embl.de/#/app/home

Citation: Frank Aylward Introduction to molecular phylogenetic reconstruction with the ETE3 Toolkit. **protocols.io** dx.doi.org/10.17504/protocols.io.qhydt7w
Published: 01 Jun 2018

Protocol

Download a working directory

Step 1.

This tutorial is designed to provide experience creating molecular phylogenies of marker genes. You may find yourself in a position where you have a gene of interest and you wish to know it's phylogenetic relationships to other known genes of the same gene/protein family. Marker genes can be markers of particular metabolic processes (functional marker genes) or markers for phylogenetic diversity (phylogenetic marker genes). An example of the former is nifH, which is a gene which encodes for a core component of the nitrogenase enzyme that is responsible for nitrogen fixation. If an organism has a nifH gene, there is a pretty good chance that it has the capacity to fix dinitrogen gas into ammonia. A good example of a phylogenetic marker gene would be rpoB, which encodes for the beta subunit of RNA polymerase and is a highly-conserved protein found across Bacteria, Archaea, and Eukaryotes, and can therefore be used to analyze phylogenetic relationships between many different organisms. From the perspective of microbial ecology, marker gene surveys can be useful

methods to learn about what kinds of microbes are present in a given environment, and what metabolic activities they may be engaged in.

Here we will practice creating a reference phylogeny for the marker gene narG, which encodes a key subunit in the nitrate reductase complex.

First let's download some practice data for GitHub:

git clone https://github.com/faylward/bioinfo-tutorials

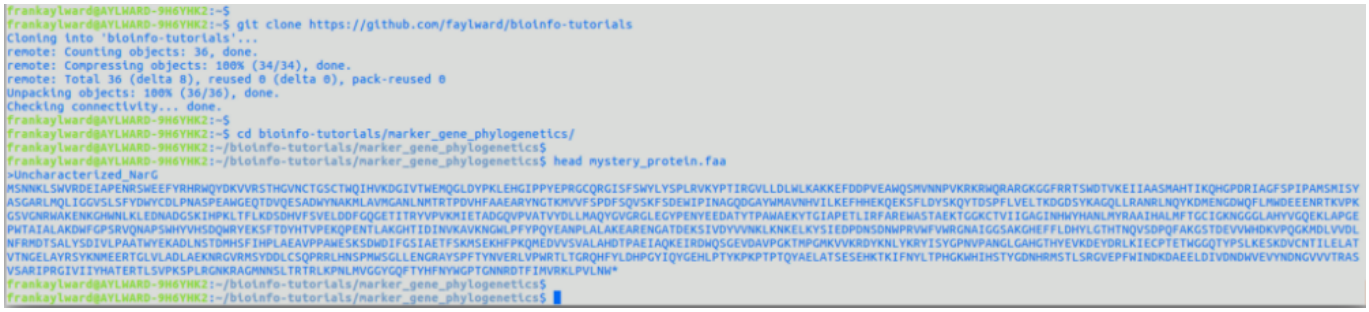
You should see a folder called 'bioinfo-tutorials', and inside that should be another folder called 'marker_gene_phylogenetics'. Let's navigate to that folder:

cd bioinfo-tutorials/marker_gene_phylogenetics

Inside this folder you should see a file called 'mystery_protein.faa' which contains the amino acid sequence of a single protein. Take a look in this file just to be sure:

head mystery_protein.faa

You can imagine that you found this protein encoded in a genome that you are analyzing and you think it may have been acquired through lateral gene transfer. Perhaps you did some gene annotation and you know it encodes for a NarG protein, but you don't know anything else about it. Creating a phylogeny with this protein and some reference sequences is a great way to investigate possible lateral gene transfers and see what other sequences are closely related. Or perhaps you found this gene in a particular environmental gene survey and you are interested in knowing what reference sequences are closely related. Either way, a phylogeny is informative.



Download a FASTA file of protein sequences

Step 2.

So let's proceed with getting some reference sequences. The EggNOG database has a very nice compilation of marker proteins, so we can download sequences from there:

wget -O narg.faa.gz http://eggnogapi.embl.de/nog_data/text/fasta/COG5013

And because the file is gzipped, let's uncompress is:

gunzip narg.faa.gz

Go ahead and take a look at the narg.faa file and make sure it's in the format you think it should be in (FASTA amino acid).

Some simple poking around and basic QC is always a good idea:

head narg.faa

grep -c '^>' narg.faa

I also like using seqtk and datamash on combination to get some basic stats about the proteins involved (see 'Introduction to analyzing FASTA files' for details here):

For example, if we want to know what the mean, minimum, maximum, and standard deviation for all protein lengths in in narg.faa is, we can use the command:

seqtk comp narg.faa | datamash mean 2 min 2 max 2 sstdev 2

For this I got mean = 1177.9, min = 54, max = 1296, and stdev = 211.1.

So assuming an average Prokaryotic gene length of 1000 bp, which translates to 330 amino acids, we see that NarG is a fairly long protein.

The code in this step should look like this:

Filter out short sequences

Step 3.

The mean and max protein lengths are not that different, but there do seem to be some shorter proteins present. We may wish to filter those proteins out, since shorter proteins will complicate sequence alignment and phylogenetic inference.

Let's try filtering out all proteins that are more than two standard deviations shorter than the mean (so 1177 - 2 X 211 = 755).

For this we can use the seqtk subseq command, which will allow us to specify a list of sequences that we wish to sub-sample from the full FASTA file. To do this we first need a file with the lengths of all of the proteins that we wish to retain, listed one after the other in a plain-text file. Let's use some pipes and AWK, which is a handy Unix-provided language that can be used for simple computations like

this:

seqtk comp narg.faa | awk '$2>755' | cut -f 1 > narg.long_proteins.list

This command does the following: 1) It takes the proteins in the narg.faa file and gets their general stats with seqtk, 2) uses an AWK command to filters through the seqtk comp output and provide only those lines where the second column value is > 755, and 3) then cuts out the first column and puts that in a file called narg.long_proteins.list.

Now we can use 'seqtk subseq' to get a FASTA file of only the proteins longer than 755 amino acids:

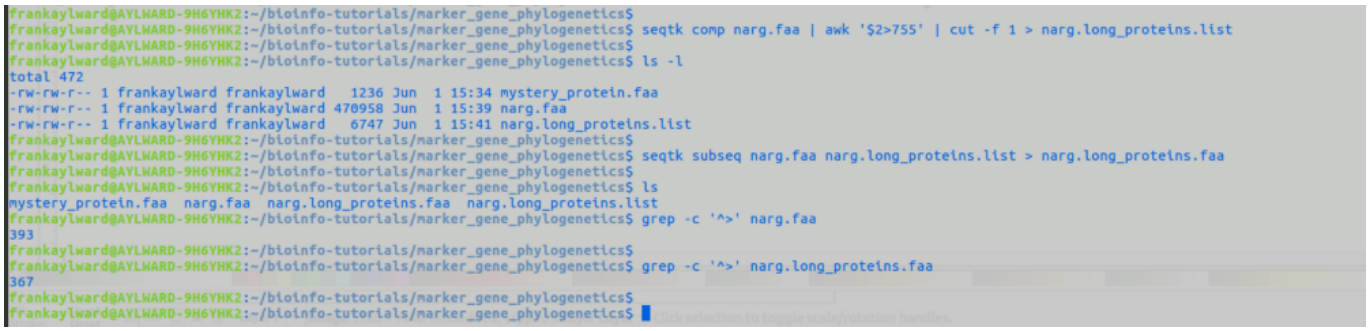
seqtk subseq narg.faa narg.long_proteins.list > narg.long_proteins.faa

How many proteins did we filter out with our length cutoff? We can check with:

grep -c '^>' narg.faa

grep -c '^>' narg.long_proteins.faa

I got 393 and 367, so we removed 26 of the shortest proteins with our length cutoff.



Remove redundant sequences from the reference file

Step 4.

Sometimes you will find that the reference sequences are highly redundant, meaning that a large number of sequences in the file are identical or nearly identical. Usually we are interested in getting a pretty broad view of phylogenetic relationships at this stage, so it may be worth while to de-replicate

the reference sequences rather than spend a lot of time later computing phylogenies of sequences that are 99% identical.

For sequence-based dereplication a handy tool is CD-HIT. You should be able to install this tool fairly easily using:

sudo apt install cd-hit

And if you just type cd-hit into the command line afterwards you should be able to view the many different options that this tool provides.

Be default cd-hit will take a FASTA file of proteins and identify clusters that are 90% or more identical over 90% the length of the shorter protein. These sequences are then grouped together, and one (usually the longest protein) is chosen as a 'representative'. This way we can just use cluster representatives in our final analysis rather than using every single sequence. Note that the % identity and % overlap thresholds can be modified using various flags in the help menu, so you could opt to remove redundancy only at the 99% amino acid identity level if you were interested in removing only the very very similar sequences, for example.

Let's try a simple command with the default parameters:

cdhit -i narg.long_proteins.faa -o narg.nr.faa

CD-HIT will print out some general log of what it's doing before finishing. You can check out the files that were created with 'ls -l':

You should see the output file we specified 'all_narg.nr.faa', which will have the cluster representatives, as well as a .clstr file which contains information about what sequences were clustered together. We will work mainly with the .nr.faa file.

Now let's check to see how many sequences were in the original file, and how many fewer are in new .nr.faa file (i.e, how many cluster representatives there are compared to raw sequences).

grep -c '^>' narg.long_proteins.faa

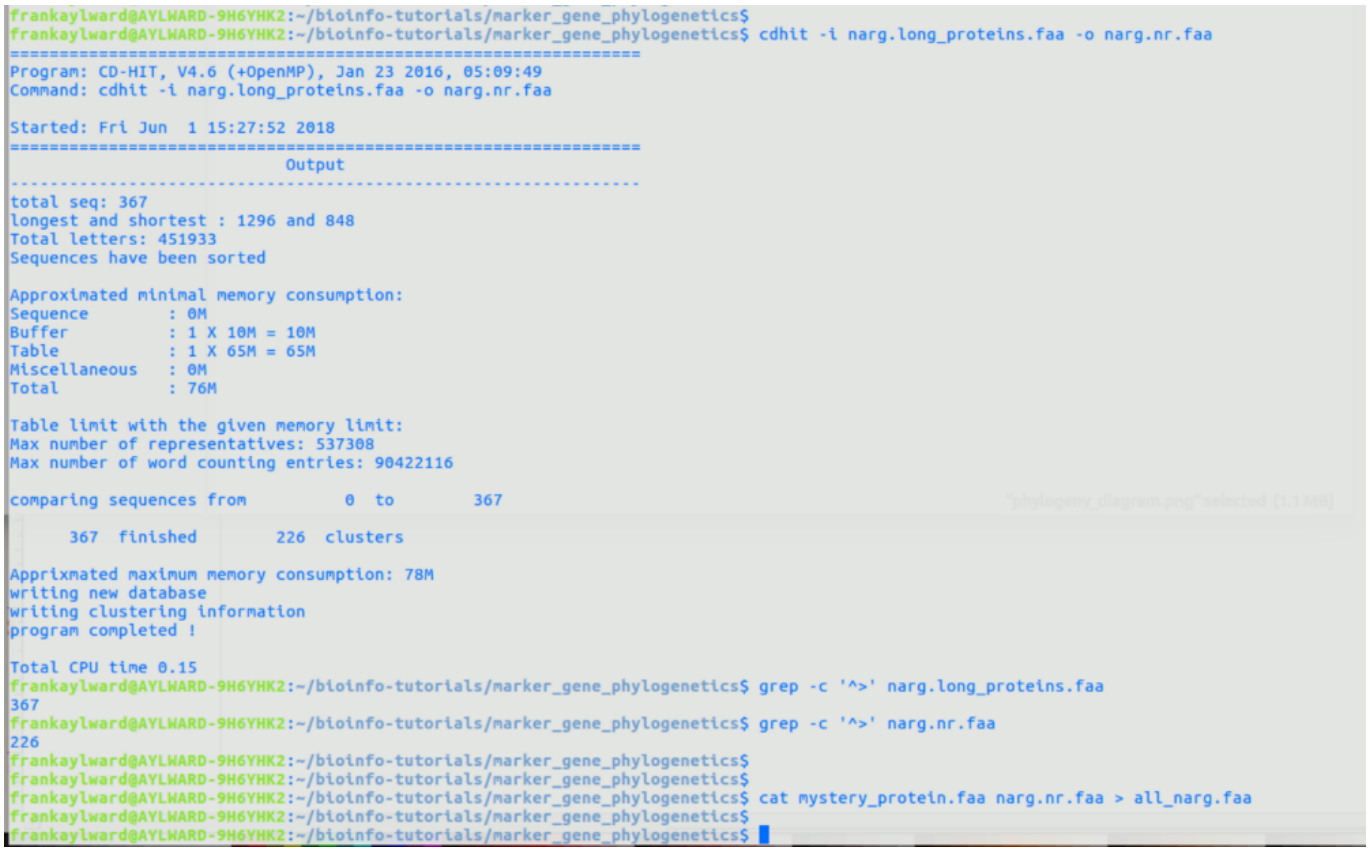
grep -c '^>' narg.nr.faa

I got 367 and 226, which indicates we removed 141 sequences. This will save us some time when we generate the phylogeny in the next steps.

Before we proceed, let's combined our non-redundant reference sequences with our mystery sequence so that we have one consolidated FASTA file to use for phylogenetic reconstruction:

cat mystery_protein.faa narg.nr.faa > all_narg.faa

Overall the commands should look something like this:



Generate the phylogeny

Step 5.

Now that we have our final NarG amino acid file, we can use it as input for ete3, which is a very useful tool that integrates different programs for sequence alignment, alignment trimming, and phylogenetic reconstruction into one interface.

Let's try the following command:

ete3 build -a all_narg.faa -o narg_phylogeny -w standard_trimmed_fasttree --spname-delimiter -C 16

This will take a bit to run (5-10 minutes), and it will provide a running log of the processes as they run. A few notes on the flags:

-a is the input. If we had wanted to use nucleic acids we would have used -d.

-o is the output directory. This is where all of the output files will go.

-w is the workflow name. There are many different workflows that are possible, depending on what programs you wish to use for alignment, trimming, and phylogenetic reconstruction. This particular workflow uses Clustal Omega for alignment, Trimal for alignment trimming, and FastTree for phylogenetic inference. A full list of options is available on the ete3 toolkit website:

At the end you should get some command line output that looks like this:

At the end you should have a directory called narg_phylogeny, and inside that should be another directory called something like 'clustalo_default-trimal01-none-fasttree_full'. Inside that directory will be the .nwk and .fa files with the trees and alignments, respectively, as well as several visualizations of the trees.

If you look at the figures it may take you a bit to find the mystery NarG protein we started with- it should be called 'uncharacterized narg' since that is the name that it had in the mystery_protien.faa file. Once you locate it you can see what other proteins are similar. Here, since we are using sequences from the EggNOG database, you will want to look up what some of the species codes stand for. For example, two of the proteins that is similar to our mystery protein have names that start with the prefix 314278.NB231, which, if we look up on EggNOG, is the species code for *Nitrococcus mobilis*, which is a marine nitrate-reducing bacterium. So the presence of NarG in this organism makes sense given what is known about its physiology.