Building Phylogenetic Tree Version 2

Elaina Graham,Benjamin Tully

Abstract

Making a Phylogenetic Tree

In this procedure we are going to build a phylogenetic tree! Throughout I'll refer to scripts available on my github here. We will be using the 16 ribosomal proteins used in Hug et al. 2016. The HMM models for these are available on my github in a file called `hug_ribosomalmarkers.hmm`. We will be running this tutorial using the genomes and files found in the folder `Example`.

As an overview I will be explaining
1. How to generate gene predictions using prodigal for genomes of interest
2. How to identify ribosomal proteins using an hmm model
3. How to align, trim, and concatenate proteins for a phylogenetic tree
4. building a phylogenetic tree

To cite this workflow reference this paper:
Graham, E. D., J. F. Heidelberg, and B. J. Tully. 'Potential for primary productivity in a globally-distributed bacterial phototroph.' *The ISME journal* (2018)
doi: https://doi.org/10.1038/s41396-018-0091-3

Citation: Elaina Graham,Benjamin Tully Building Phylogenetic Tree. **protocols.io**
dx.doi.org/10.17504/protocols.io.q2pdydn
Published: 15 Jun 2018

Before start

Before Starting Be sure you have all the required Pre-Requisites.

Python2.7
BioPython
HMMER
Prodigal
BinSanity
Muscle
TrimAL
FastTree

See this github page for links to all dependencies.

To cite this workflow reference this paper:
Graham, E. D., J. F. Heidelberg, and B. J. Tully. 'Potential for primary productivity in a globally-distributed bacterial phototroph.' *The ISME journal* (2018)
doi: https://doi.org/10.1038/s41396-018-0091-3

Protocol

Gathering the right files

Step 1.

Now to get all of the scripts and files do the command below to download the git repository.

The cd into `PhylogenomicsWorkflow/Example`

cmd COMMAND

```
$ ls Genomes

unknown1.fna  unknown2.fna  unknown3.fna  unknown4.fna  unknown5.fna

$ ls HugRef

ExampleRefSet.RpL14.faa  ExampleRefSet.RpL22.faa  ExampleRefSet.RpL4.faa   ExampleRefSet.R
pS17.faa
ExampleRefSet.RpL15.faa  ExampleRefSet.RpL24.faa  ExampleRefSet.RpL5.faa   ExampleRefSet.R
pS19.faa
ExampleRefSet.RpL16.faa  ExampleRefSet.RpL2.faa   ExampleRefSet.RpL6.faa   ExampleRefSet.R
pS3.faa
ExampleRefSet.RpL18.faa  ExampleRefSet.RpL3.faa   ExampleRefSet.RpS10.faa  ExampleRefSet.R
pS8.faa


..
```

You should now be in the PhylogenomicsWorkflow/Example directory which contains Genomes of interest and a small set of reference ribosomal proteins pulled from genomes on NCBI

Generate Gene Calls and Extract Ribosomal Proteins from Genomes Of Interest

Step 2.

The primary script in this workflow in `identifyHMM`. The script relies on the user providing the location of a file containing Hidden Markov Models (HMMs) for their genes of interest. Here we have provided you with a file called `hug_ribosomalmarkers.hmm`. This contains HMM models for 16 Ribosomal proteins: RpL14, RpL15, RpL16 ,RpL18, RpL22, RpL24 ,RpL2, RpL3, RpL4, RpL5, RpL6, RpS10, RpS17, RpS19, RpS3, RpS8.

Now you should enter the directory containing your genomes (in our case /PhylogenomicsWorkflow/Example/Genomes)

The help message for `identifyHMM` is given below.

```
usage: identifyHMM [-h] [--markerdb MARKERDB] [--performProdigal] [--cut_tc]
                   [--outPrefix OUTPREFIX] [--Num NUM] [-E E]
                   Input

Identify marker genes in in protein sequences of genomes.

positional arguments:
  Input                 Target file(s). Provide unifying text of desired
                        genome(s). Ext must be 'fna' or 'faa'.

optional arguments:
  -h, --help            show this help message and exit
  --markerdb MARKERDB   Provide HMM file of markers. Markers should have a
                        descriptive ID name.
  --performProdigal     Run Prodigal on input genome nucleotide FASTA file
  --cut_tc              use hmm profiles TC trusted cutoffs to set all
                        thresholding
  --outPrefix OUTPREFIX
                        Provide prefix of names for marker output files.
  -E E                  Set E-Value to be used in hmmsearch. Default: 1E-5
```

Note When using `identifyHMM` on your own data you need to remember that you should be in the directory containing your MAGs when you run the program and ensure that the only genomes in this directory are the genomes of interest. The program will parse through every file with the suffix given as the [input]. So in this case our input is `.fna`. Second, if you want to run identifyHMM with your own gene calls you will want to exclude the `--performProdigal` flag and be sure that your gene calls end with the suffix `.faa` and are in the same directory as the genomes of interest.

The HMM file that we will be using is found in `/PhylogenomicsWorkflow/hug_ribosomalmarkers.hmm` on the github. The HMM Models in this file were pulled from PFAM and represent 16 ribosomal proteins that tend to be syntenic/co-located (Hug et al. 2016)

Once in the `Genomes` directory run `identifyHMM` as Follows:

```
 identifyHMM --markerdb ../../hug_ribosomalmarkers.hmm --performProdigal --
cut_tc --outPrefix HUG .fna
```

The output of this will be 16 `.faa` files appended with the prefix spefied by `--outPrefix` (in this case we used `HUG`), five `.faa` files corresponding to the prodigal gene calls, and five hmm reports (Stored in the `.tbl` files). So the files that look like `HUG_RpL14.faa` are ribosomal proteins pulled from your genomes of interest.

```
hmmsearhc-log.txt  HUG_RpL16.faa  HUG_RpL24.faa  HUG_RpL4.faa
HUG_RpS10.faa      HUG_RpS3.faa   unknown1.fna   unknown2.fna
unknown3.fna       unknown4.fna   unknown5.fna   HUG_RpL14.faa
HUG_RpL18.faa      HUG_RpL2.faa   HUG_RpL5.faa   HUG_RpS17.faa
HUG_RpS8.faa       HUG_RpL15.faa  HUG_RpL22.faa  HUG_RpL3.faa
HUG_RpL6.faa       HUG_RpS19.faa  unknown1.faa   unknown2.faa
unknown3.faa       unknown4.faa   unknown5.faaunknown1.ribomarkers.tbl
unknown2.ribomarkers.tbl  unknown3.ribomarkers.tbl  unknown4.ribomarkers.tbl
unknown5.ribomarkers.tbl
```

cmd COMMAND

```
 identifyHMM --markerdb ../../hug_ribosomalmarkers.hmm --performProdigal --cut_tc --
outPrefix HUG --Num 16 .fna
```

The output of this will be 16 `.faa` files appended with the prefix spefied by `--outPrefix`, 5 `.faa` files corresponding to the prodigal gene calls, and 5 hmm reports (Stored in the `.tbl` files). So the files that look like `HUG_RpL14.faa` are ribosomal proteins pulled from your genomes of interest.

Merge Reference Proteins with Yours

Step 3.

Now that we have extracted marker genes from our genomes of interest we can merge together the Ribosomal Protein calls we just made on our genomes of interest with those in the folder `'/PhylogenomicsWorkflow/Example/HugRef'` accordingly.

To do this we will use a quick bash trick that uses the text file `hug_marker_list.txt` which contains a list of the marker proteins we are searching for. Use the following Bash command:

cmd COMMAND

```
while read p
                 do cat ../HugRef/ExampleRefSet."$p".faa HUG_"$p".faa > Dataset1_"$p".faa
done < ../../hug_marker_list.txt
```

This bash script is iterating through the list given in `hug_marker_list.txt` and concatenate appropriate reference and experimental protein sets. So for example it would concatenate `ExampleRefSet.RpL6.faa` and `Hug_RpL6.faa` into `Dataset1_RpL6.faa`.

Align

Step 4.

Now that we have 16 files containing Ribosomal proteins from our 5 unknown genomes and 100 references we can move on to aligning our proteins.
To align our proteins we can use muscle (You could also alternatively use MAFFT).

```
Dataset1_RpL14.faa  Dataset1_RpL24.faa    Dataset1_RpL6.faa
Dataset1_RpS8.faa
Dataset1_RpL15.faa  Dataset1_RpL2.faa   Dataset1_RpS10.faa        hmmsearhc-
log.txt
Dataset1_RpL16.faa  Dataset1_RpL3.faa   Dataset1_RpS17.faa
HUG_RpL14.faa
Dataset1_RpL18.faa  Dataset1_RpL4.faa   Dataset1_RpS19.faa
HUG_RpL15.faa
Dataset1_RpL22.faa  Dataset1_RpL5.faa   Dataset1_RpS3.faa        HUG_RpL16.faa
HUG_RpL18.faa       HUG_RpL4.faa        HUG_RpS19.faa
unknown1.ribomarkers.tbl
HUG_RpL22.faa       HUG_RpL5.faa        HUG_RpS3.faa
unknown2.faa
HUG_RpL24.faa       HUG_RpL6.faa        HUG_RpS8.faa
unknown2.fna
HUG_RpL2.faa        HUG_RpS10.faa        unknown1.faa
unknown2.ribomarkers.tbl
HUG_RpL3.faa        HUG_RpS17.faa        unknown1.fna
unknown3.faa
unknown5.faa        unknown5.fna        unknown5.ribomarkers.tbl
```

We will use the same trick as above where we feed the list of ribosomal markers in `hug_marker_list.txt` into a bash loop thar runs muscle on head of the protein fasta files we generated. This will end in 16 alignment (`.aln`) files.

cmd COMMAND

```
while read p
                  do muscle -maxiters 16 -in Dataset1_"$p".faa -out Dataset1_"$p".aln
done < ../../hug_marker_list.txt
```

Trim

Step 5.

Now that we have our alignments we need to trim those alignments. There are many programs that do this and I implore you to try a couple and see how different ones work, or even try manual trimming. Here we will use Trimal because its automated and works quickly when running alignments with lots of sequences.

You can run the following to trim your alignments:

cmd COMMAND

```
while read p
                 do trimal -automated1 -in Dataset1_"$p".aln -
out Dataset1_"$p".trimmed.aln
```

```
done < ../../hug_marker_list.txt
```

Concatenate Proteins

Step 6.

Now that you have the trimmed and aligned sequences you can concatenate these 16 files. To do this we will use the concat script packaged with BinSanity.

Usage is shown below:

```
usage: concat -f directory -e Alignment Extension --Prefix file linker -o
output

*****************************************************************************
*********************************BinSanity***********************************
    **     The `concat` script is used to concatenate multiple sequence
**
    **     alignments for conducting a phylogenomic analysis. Note that you
**
    **     receive an error if there are any duplicate sequence ids in an
**
    **     alignment.
*****************************************************************************

optional arguments:
  -h, --help  show this help message and exit
  -f          Specify directory where alignments are
  -e          Specify the extension for your alignments (must be in Fasta
format)
  --Prefix    Specify the prefix that links your alignments (ex: if you have
two alignments TOBG_RpL10, TOBG_RpL24, the --Prefix would be TOBG
  -o          Specify output file
  -N          Specify the minimum number of sequences needed to be included in
concatenation
```

cmd COMMAND

```
concat -f . -e .trimmed.aln --Prefix Dataset1 -
o Dataset1.HugRibosomal.trimmed.concat.aln -N 8
```

Build the Tree

Step 7.

You now have a concatenated alignment 'Dataset1.HugRibosomal.trimmed.concat.aln' which can be used to build a phylogenetic tree.

We will be using FastTree with the `-gamma` and `-lg` parameters. These parameters are optional and just indicate what models we would like to use for branch length calculation and amino acid evolution respectively. Feel free to adjust these for your own purposes.

cmd COMMAND

```
FastTree -gamma -
lg Dataset1.HugRibosomal.trimmed.concat.aln > Dataset1.HugRibosomal.trimmed.concat.newick
```

View Tree!

Step 8.

Now you have a newick file which can be viewed in a variety of tree views and edited. One of my favorite tools for making publication quality trees is ITOL!

To cite this workflow reference this paper:
Graham, E. D., J. F. Heidelberg, and B. J. Tully. 'Potential for primary productivity in a globally-distributed bacterial phototroph.' *The ISME journal* (2018)