Unrooted Phylogenetic Orthology
(Short documentation)

Jesus A. Ballesteros

1 Quick recipe

1. Rename each fasta file to include the OTU of species name. Using the provided script `minreID.py`:

   ```
   minreID.py SequencesAA.fasta Species_Name1 \|
   ```

2. Concatenate all the renamed fasta files into a single reference file.

   ```
   cat *_withid.fasta > AllSeqs.faa
   ```

3. AllvsAll Blast. Create a blast database of the reference fasta file and search this file against this recently created database using the csv output format. The `blast_helper.sh` script is provided to facilitate and parallelize this procedure.

   ```
   Blast_helper.sh AllSeqs.faa
   ```

4. Cluster similar sequences into homologous families. And additional e value threshold can be enforced at this stage as well as a minimum taxa threshold to filter out clusters with not enough taxonomic representation.

   ```
   BlastResultsCluster.py -in blast_out.csv -d \| -e 1e-50 -m 5 -R AllSeqs.faa
   ```

5. Phylogenetic pipeline Perform multiple sequence alignement, gap masking, alignment sanitation and phylogenetic inference on the clusters.

   ```
   cd ClusteRs/
   paMATRAX+.sh
   ```

6. Orthology assessment with UPhO. Run all the gene family gene trees trhough UPhO.

   ```
   UPho.py  -in *.tre  -m 5 -S 0.95 -ouT -iP -R ../AllSeqs.faa}
   ```

2 Download & install

All the scripts described in this tutorial al available at https://github.com/ballesterus/UPhO.git. You can clone the latest version using git or download the files directly from the GitHub website. The scripts provided can be executed as stand alone programs or their contained functions imported as python modules in the interpreter. To execute the scripts from any directory add the folder containing the scripts to your PATH variable. Additionally, add this same folder to the PYTHONPATH variable so the functions can be imported by the python interpreter. Example: If UPhO folder is cloned (copied) in the folder

```
\Home\UserName
```

then add the following lines to the `.bash_profile`:

```
export PATH="$PATH:\Home\UserName\UPhO\"
export PYTHONPATH="$PYTHONPATH:\Home\UserName\UPhO\"
```

For the changes to take effect without logging off and on, run `source .bash_profile`

2.1 Dependencies

For orthology inference UPhO depends only of python 2.7.x with standard libraries and modules. Some other scripts, such as distOrth.py and Get_fasta_from_Ref.py make use of ETE2 and/or Biopython modules. Many of the steps for the workflow herein exemplified consists of common tasks for which a wide variety of programs are available including: gene homology, multiple sequence alignment, alignment trimming and sanitation, tree inference, etc.). Here we demonstrate a possible implementation of this pipeline but in the end users should be able to modify these tools according to the problem requirements or user preferences. For example, here we use mafft aligner but muscle or clustal could have been used for the same purpose.

Third party programs used in the workflow:

- gnu-parallel http://www.gnu.org/software/parallel/
- Mafft http://mafft.cbrc.jp/alignment/software/
- mcl http://micans.org/mcl/index.html
- NCBI-Blast+ http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download
- trimAl http://trimal.cgenomics.org
- raxml http://sco.h-its.org/exelixis/web/software/raxml/index.html
- FastTree http://meta.microbesonline.org/fasttree/

Python modules:

- biopython http:biopython.org
- ETE2 http://etetoolkit.org/docs/2.3/

The user can (and should) adjust the parameters of each of this programs according to their dataset complexity and research goals.

3 Sequence and leaves identifiers

For correctly parsing, the sequence identifiers in fasta files and leaf names in newick tree files, should be composed of at least two fields; the first being a species name consisting exclusively of alpha-numeric characters and underscore (a-z, A-Z, 0-9, _). The second field should correspond to a unique sequence identifier, also composed of alpha-numeric characters and underscore. Blank spaces and punctuation marks should be avoided in the sequence identifiers. Additional fields are effectively ignored and therefore tolerated.

The fields must be separated by a custom character (default "|") that is not in the set used for naming OTU's. Virtually any other character could be used although common sense should prevent the use of characters with special meanings in fasta and newick standards, such as ">", ":", "(", etc.

The script `minreID.py` can assist with the renaming of each file or the user can resort to stream line editors (awk, sed) to comply with the naming requirements.

This simple script takes as arguments the target file, species name to use and the custom delimiter. EXAMPLE:

```
assembly.fasta:
>c1212_g1_i2
GATACAGATACA

$ minreID.py assembly.fasta  mySpecies \|

assemly_withids.fasta:
>mySpecies|000001c1212_g1_i2
GATACAGATACA
```