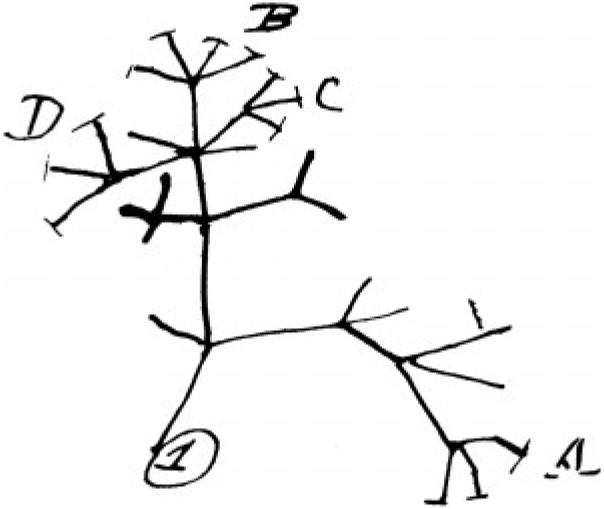
Phylogenetics and data processing

Phylogenetics Trees

- Representation of relationships of species, genes, groups
- Adjacent branches are more closely related entities than ones further away
- Generate trees based on computed distances or more complex liklihood models



Compact Tree Representation

Newick format

```
(A,B,(C,D),(E,F,(G,H)));
 ________________________ A
 |
 |________________________ B
 |
 |                         ________________________ C
_|________________________|
 |                        |________________________ D
 |
 |                         ________________________ E
 |                        |
 |________________________|________________________ F
                          |
                          |                         ________________________ G
                          |________________________|
                                                   |________________________ H
```

Molecular Phylogenetics

- Identify homologous sequences (e.g. genes)
- Align sequnence with Multiple alignment software
- Potentially trim sequence alignment or poor alignment region
- Construct phylogenetic tree

Identify homologous sequences

- Reciprocal BLAST
- Orthology Clustering
- Syntenic information (shared flanking regions)

- Molecular approaches (PCR, Clone, Sequencing)

- DNA and Protein considerations

- Store in FASTA file

Extract those sequences

```
$ module load cdbfasta
$ cdbfasta DATABASE
# make a file which has the sequence names, one each line
$ cdbyank DATABASE.cidx < listseqids > sequencefile.fa
```

Multiple Sequence Aligmment

- Construct MSAs: MUSCLE

```
$ module load muscle
$ muscle -in VMS1.aa.fasta -out VMS1.aa.fasaln
$ muscle -clw -in VMS1.aa.fasta -out VMS1.aa.aln
```

- Construct MSA of coding sequence: T-Coffee
- Align the sequence on Codon Boundaries (e.g. w/ knowledge of the amino-acid it codes for)

  ```
  $ module load tcoffee
  $ t_coffee VMS1.cds.fasta -method cdna_fast_pair
  ```

Trimming alignments

- Trimal or Gblocks to remove poorly aligning regions

```
$ module load trimal
$ trimal -in VMS1.aa.fasaln -out VMS1.aa.trim -automated1
```

- additional options for trimming

  ```
  # remove spurious sequences ('good site's are 75% seqs share a site;
  # 80% of the sites in sequence must be good to keep the sequence)
  $  trimal -in inputfile -out outputfile -resoverlap 0.75 -seqoverlap 80
  ```

- Trimal can also convert formats of files (fasta to PHYLIP, NEXUS, etc)

Phylogenetic anlayses

- Neighbor-Joining: Rapid, fast tree building. Distance based

```
#!/usr/bin/bash
#SBATCH --nodes 1 --ntasks 1
module load fasttree
FastTree VMS1.aa.trim > VMS1.aa.nj.tre
```

- FastTree automatically bootstraps 1000x
- FastTreeMP will run on multiple processors
- Many parameters to specify if nucleotide sequences, distributions of site variance, etc

Maximum Likelihood

- RAxML or IQTree on the cluster

  ```
  #!/usr/bin/bash
  #SBATCH --ntasks 2 --nodes 1
  module load IQ-TREE
  iqtree-omp -s VMS1.aa.trim -nt 2
  ```

```
#!/usr/bin/bash
#SBATCH --ntasks 4 --nodes 1
module load RAxML
raxmlHPC-PTHREADS-SSE3 -m PROTGAMMAAUTO -T 4 -s VMS1.aa.trim -n VMS1_Run1 -
```

Viewing trees

- iTOL - https://itol.embl.de/
- Figtree - http://tree.bio.ed.ac.uk/software/figtree/ https://github.com/rambaut/figtree/

```
# with X11 enabled on your laptop
$ module load figtree
$ figtree file.tre
```

Advanced Models

- Model fit - to identify most likely model of seq evolution : ModelTest, jModelTest, ProtTest
- Concatenated gene sets, allow partitioning of data : Partition Finder

more you can do in BioPython

BioPython tutorial http://biopython.org/wiki/Phylo and Cookbook http://biopython.org/wiki/Phylo_cookbook

- Read / Write Trees
- Construct trees from MSA (distance and parsimony)
- Bootstraps

Traverse a tree

```
#!/usr/bin/env python3

from Bio import Phylo
trees = Phylo.parse("treefile1.tre", "newick")

def all_parents(tree):
    parents = {}
    for clade in tree.find_clades(order='level'):
        for child in clade:
            parents[child] = clade
    return parents


for tree in trees:
    print(tree)
    Phylo.draw_ascii(tree)

    for tip in tree.get_terminals():
       print("terminal tip",tip)
       term_names = [term.name for term in tree.get_terminals()]

    parents = all_parents(tree)
    clades = tree.find_clades("E")
    for myclade in clades:  # get first instance of 'E'
        parent_of_myclade = parents[myclade]
        print(parent_of_myclade.get_terminals())
```