Phylogenetics Tutorial 1:

Making Phylogenies

Finlay Maguire

Faculty of Computer Science

Table of contents

Overview

Protein Phylogeny Aims

- Get a protein
- Using pairwise alignment to find potential homologs
- Perform a multiple sequence alignment
- Trim the alignment
- Infer a NJ distance phylogeny
- Infer an approximate Maximum Likelihood phylogeny
- Infer an accurate Maximum Likelihood phylogeny
- Compare the trees

Core Genome Phylogeny

- Get genomes
- Find core genome
- Extract SNPs
- Infer a Maximum Likelihood phylogeny
- Visualise Phylogeny

Requirements

- mafft
- trimal
- aliview
- FastTree2
- iqtree
- FigTree
- prokka
- roary
- snp-sites

Installation

miniconda

If you don't have miniconda
`https://docs.conda.io/en/latest/miniconda.html`

```
conda create -n phylo -c bioconda mafft trimal prokka
                      fasttree iqtree roary snp-sites
```

```
conda activate phylo
```

or if older miniconda version:

```
source activate phylo
```

Other tools

Unfortunately, not everything is in bioconda:

- AliView
  `https://github.com/AliView/AliView/releases`
- FigTree
  `https://github.com/rambaut/figtree/releases`

Data

Starting Sequence

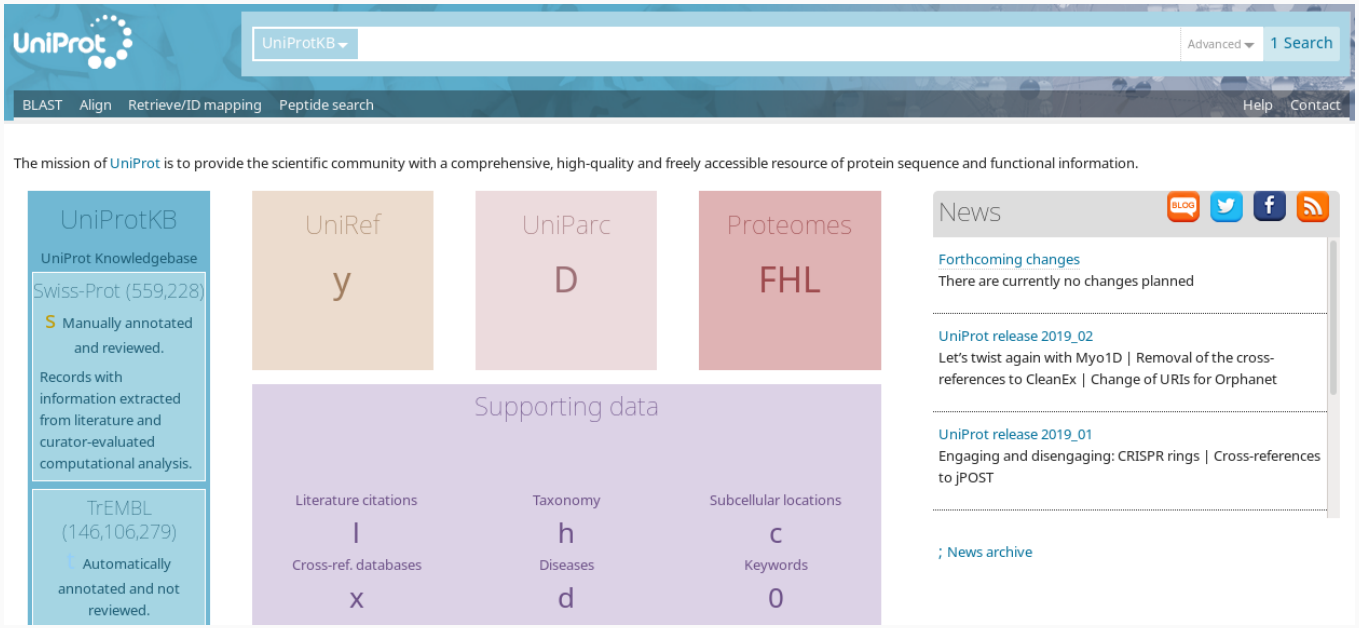

Figure 1: High-quality protein reference database: swiss-prot `http://www.uniprot.org`

Starting Sequence

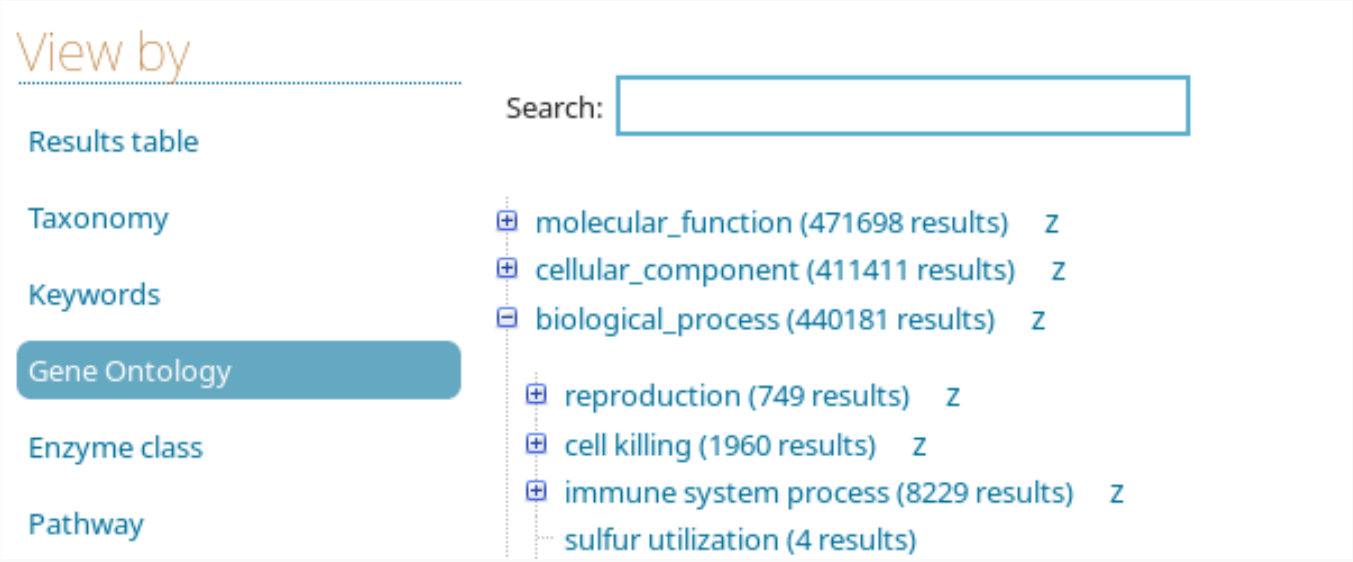

Figure 2: Choose ‘Gene Ontology’ and ‘biological process’

Starting Sequence

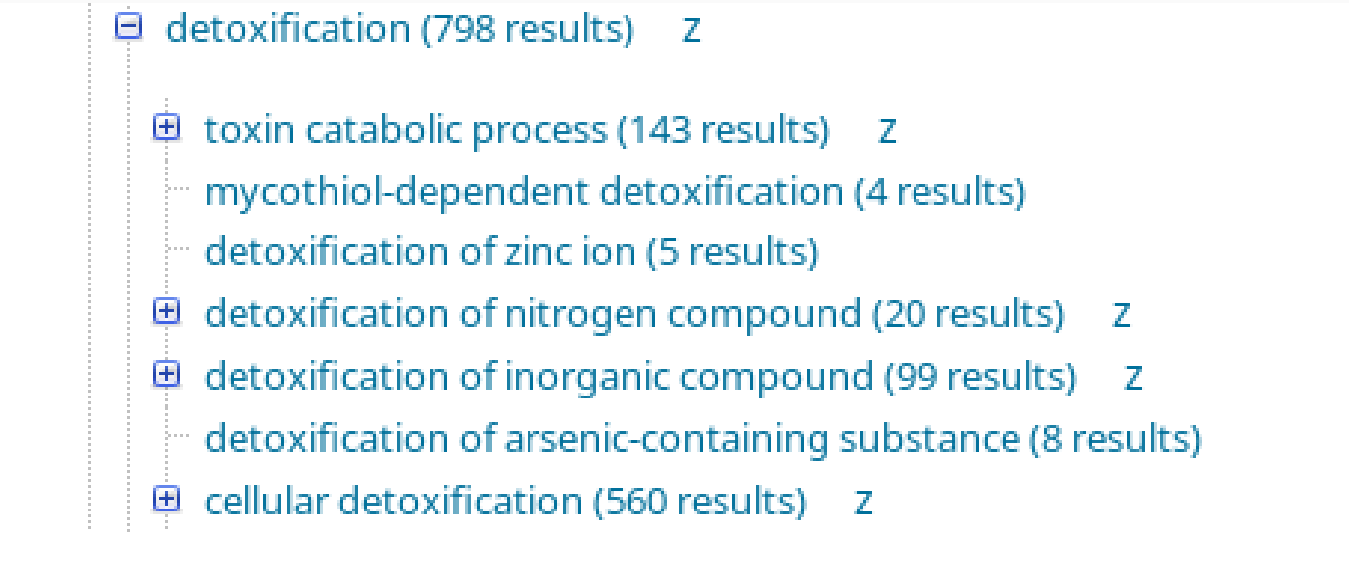

Figure 3: Go down to ‘detoxification’ and expand

Starting Sequence

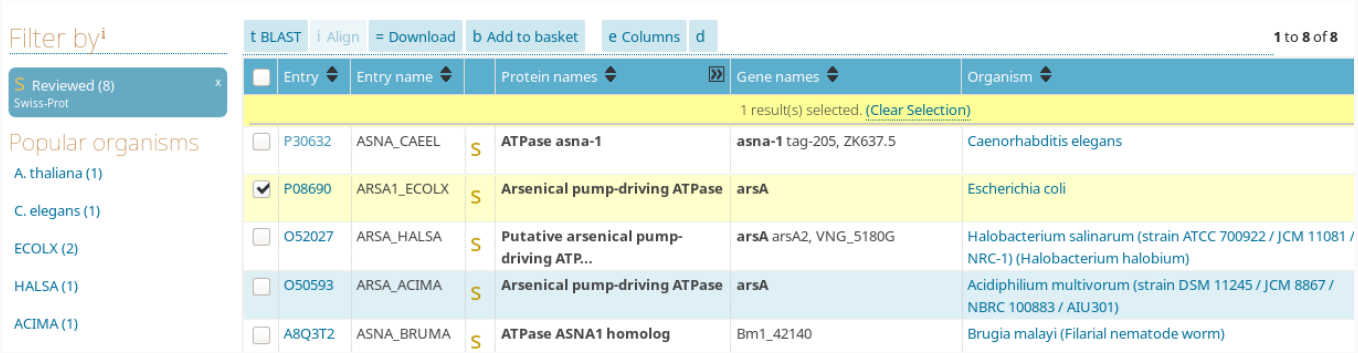

Figure 4: Select ‘8 results’ next to ‘detoxification of arsenic’

Using BLAST to find related sequences

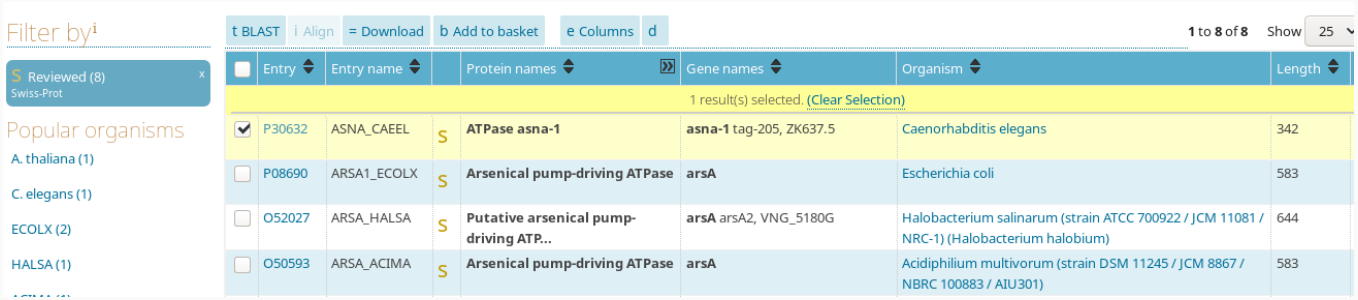

Figure 5: Select the *C. elegans* sequence and BLAST

Using BLAST to find related sequences

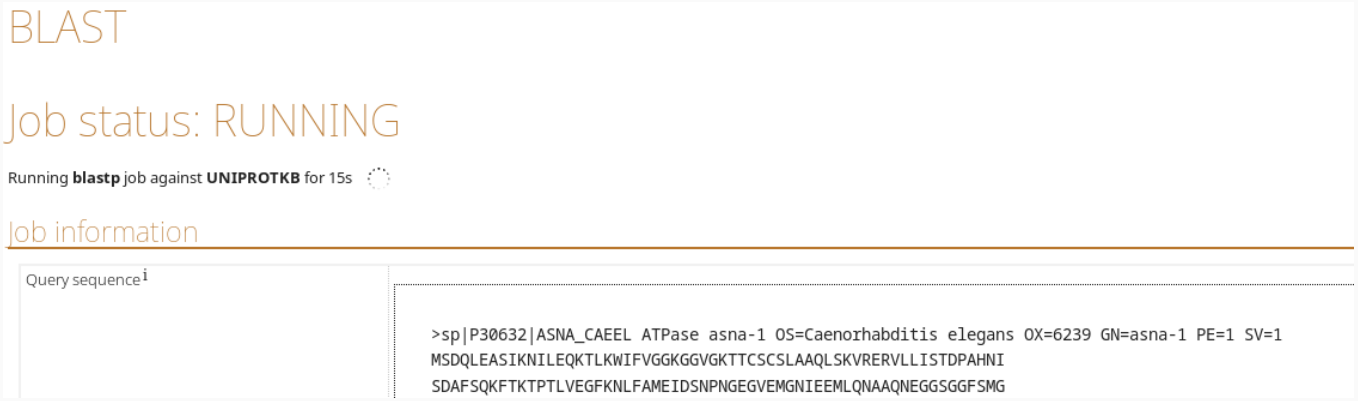

Figure 6: Wait...

Using BLAST to find related sequences

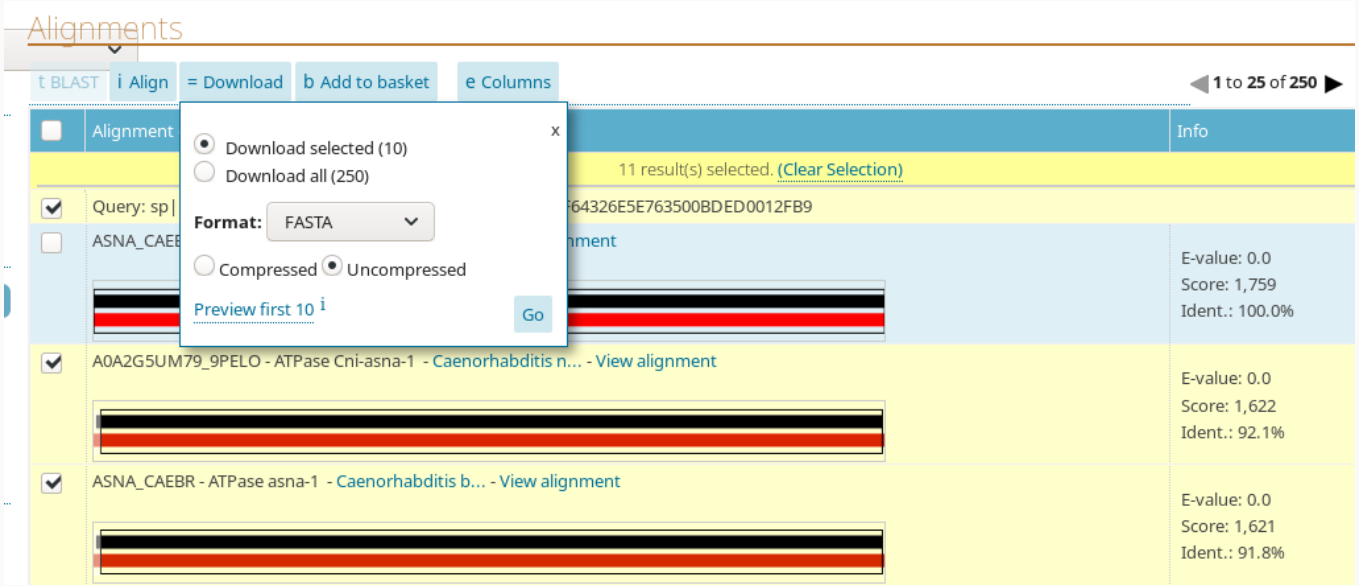

Figure 7: Download 10 sequences across a range of similarity

Multiple Sequence Alignemnt

MAFFT

```
mafft-linsi arsenic.faa > arsenic.afa
```

Trimming

Inspecting the alignment

```
java -jar aliview.jar
```

TrimAL

```
trimal -nogaps -in arsenic.afa -out arsenic_nogaps.mask
```

TrimAL

```
trimal -automated1 -in arsenic.afa -out arsenic_auto.mask
```

Compare Trimming

```
java -jar aliview.jar
```

Approximate ML Tree

FastTree

```
FastTree -lg arsenic_auto.mask > arsenic_dist.tree
```

Inspect

```
FigTree
```

Maximum-Likelihood Tree

IQ-Tree

- Generate 100 parsimony trees

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves
- Collect resulting unique topologies and optimise branch lengths

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves
- Collect resulting unique topologies and optimise branch lengths
- Select top 20 by likelihood

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves
- Collect resulting unique topologies and optimise branch lengths
- Select top 20 by likelihood
- Perform hill-climbing NNI (stochastic followed by hill-climbing) on each and optimise

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves
- Collect resulting unique topologies and optimise branch lengths
- Select top 20 by likelihood
- Perform hill-climbing NNI (stochastic followed by hill-climbing) on each and optimise
- Retain top 5 topologies as candidate trees

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves
- Collect resulting unique topologies and optimise branch lengths
- Select top 20 by likelihood
- Perform hill-climbing NNI (stochastic followed by hill-climbing) on each and optimise
- Retain top 5 topologies as candidate trees
- Randomly perturb candidates (stochastic NNI) and optimise (hill-climbing)

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves
- Collect resulting unique topologies and optimise branch lengths
- Select top 20 by likelihood
- Perform hill-climbing NNI (stochastic followed by hill-climbing) on each and optimise
- Retain top 5 topologies as candidate trees
- Randomly perturb candidates (stochastic NNI) and optimise (hill-climbing)
- If new tree is better than top candidate, replace

IQ-Tree

- Generate 100 parsimony trees
- Optimise all 100 with lazy SPR moves
- Collect resulting unique topologies and optimise branch lengths
- Select top 20 by likelihood
- Perform hill-climbing NNI (stochastic followed by hill-climbing) on each and optimise
- Retain top 5 topologies as candidate trees
- Randomly perturb candidates (stochastic NNI) and optimise (hill-climbing)
- If new tree is better than top candidate, replace
- If top candidate doesn't change after 100 random perturbations then output.

Running IQ-Tree

```
iqtree -mset LG,JTT,WAG -s arsenic_auto.mask
```

Note: IQTree does output a neighbour joining distance tree too (.bionj).

Inspect

```
FigTree
```

Phylogenomics

Get genomes

Download the 6 listeria genomes

```
wget finlaymagui.re/assets/listeria_genomes.tar.gz
tar xvf listeria_genomes.tar.gz
```

Annotate genomes

For genome GCA000008258:

```
prokka --kingdom Bacteria --outdir prokka_GCA_000008285
--genus Listeria --locustag GCA_000008285
GCA_000008285.1_ASM828v1_genomic.fna
```

Repeat for all genomes

Find shared parts

```
mkdir annotations
cp */*.gff annotations
roary -f core_genome -e -n -v annotations/*.gff
```

Extract SNPs

```
snp-sites -o listeria_snps.fna

    core_genome/core_gene_alignment.aln
```

Infer ML Phylogeny

```
iqtree -mset GTR -s listeria_snps.fna
```

Visualise Tree

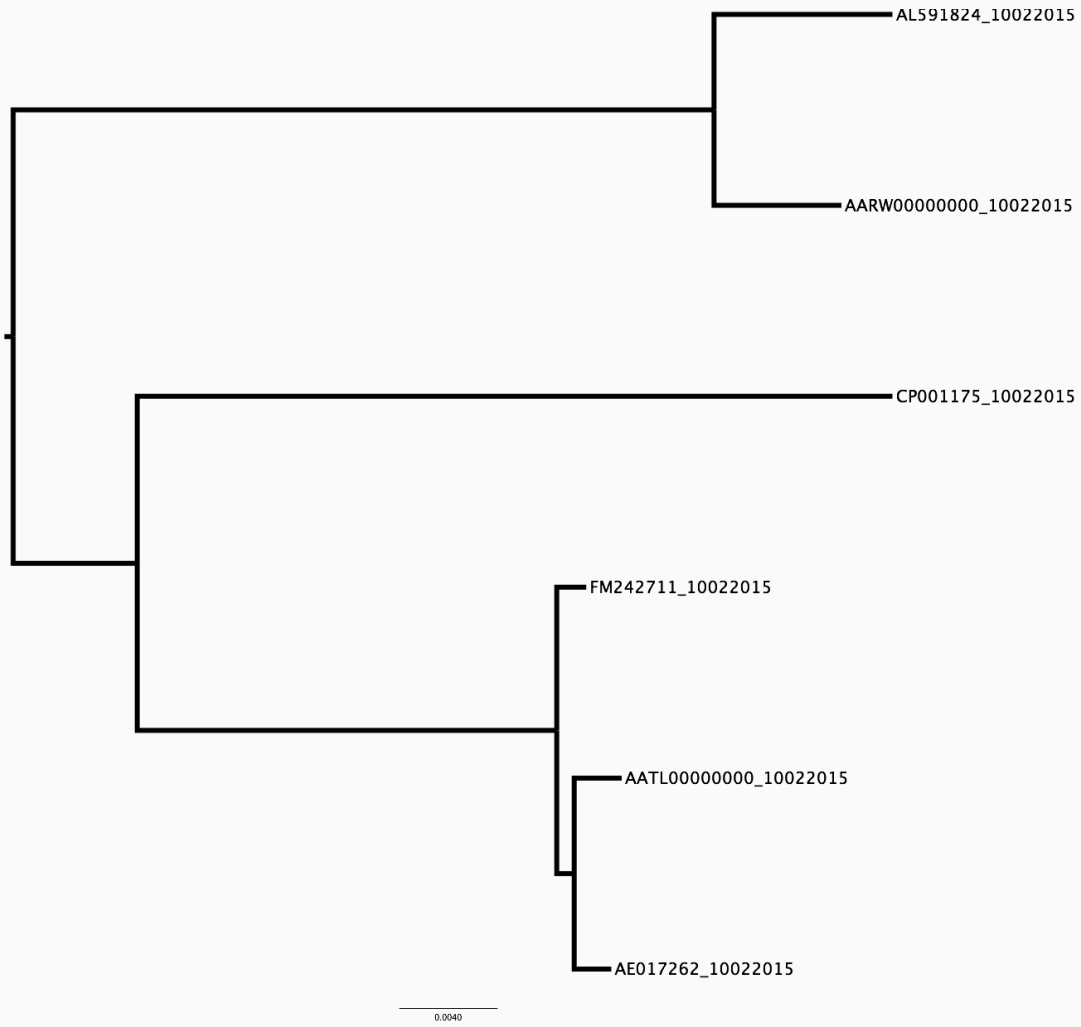

Figure 8: Roary Tutorial

Questions?