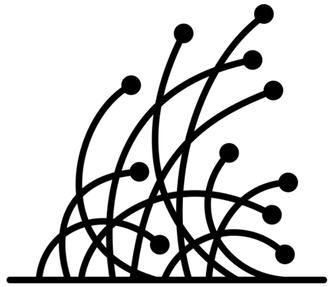

Workshop 2: *Trichophyton*

Phylogenetics, repetitive regions, R analysis/visualisation

Dr Johanna Rhodes

Learning outcomes

- **In practice**
  - Command line arguments
  - Troubleshooting errors

Trichophyton

- Zoophilic/anthropophilic species
- Cause tinea - skin infection/rash
- *Trichophyton mentagrophytes* is zoophilic - companion animals
    - some genetic variants (Type VII and VIII) known to transmit human-to-human and/or as an STD

Dr. Alireza Abdolrasouli

Trichophyton indotineae

New emerging pathogen

- Thought to originate in India
- Previously classified as *T. mentagrophytes* var. VIII
- Increasingly resistant to topical antifungal terbinafine
- Outbreaks linked to travel

Uhrlab *et al.* (2022) Journal of Fungi

Data

Short read data and reference genome

- WGS data: PRJEB75499
  - Download one set of paired end reads each
- Reference on GenBank (ASM2306590v1)
  - Thoughts on information here compared to *Candida auris* reference genome information?
- Obtain a vcf each with high-confidence SNP calls (with ‘LowConf’ labelled) —> combine all vcf together for rest of the workshop
  - Use repetitive regions file provided (GitHub repo under ‘Data’)

Repetitive regions

Excluding repetitive regions when calling variants

- RepeatMasker can be used to identify regions of a reference that are deemed repetitive
- After performing BQSR include the -XL option in HaplotypeCaller when calling variants and use the repetitive regions file provided.
  - This will exclude SNP calling in these regions

Building phylogenies with wgSNPs

How to handle low confidence SNPs?

- Do not remove them
  - Consider them as ‘missing’ instead
  - If that position is low confidence because it has low mapping quality or coverage, it may be we don’t have enough evidence for or against it being a real SNP
  - Change low confidence positions to ’N’, which will be read by phylogeny software as ‘missing’

```
grep -v "FILTER_GQ-50" isolate_name.filtered_snps_final.vcf | awk -vOFS='\t'
'$7 ~ /LowConf$/ {$5 = "N"}1' > isolate_name.finalSNPs.vcf
```

Building phylogenies with wgSNPs

Building the SNP dataset

- Index the vcf file
- First, compress using bgzip
- Then index using tabix

vcftools/bgzip isolate_name.finalSNPs.vcf

vcftools/tabix -p vcf isolate_name.finalSNPs.vcf.gz

Building phylogenies with wgSNPs

Building the SNP dataset

- Combine all the vcfs generated using vcftools ‘merge’ (can also use bcftools ‘merge’)

vcftools vcf-merge -R *vcf.gz > trichophyton.vcf

- This will print a lot of output to the screen. It annoyingly changes the name of the first isolate to ‘WGS’, and will include the full file name as the column header.
    - Can edit:

      sed -i ’s/.finalSNPs.WGS//g’ trichophyton.vcf

      sed -i ’s/WGS/isolate//g’ trichophyton. vcf

Building phylogenies with wgSNPs

Building the SNP dataset

- Convert the SNP matrix to FASTA file using bcftools:

```
vcf=merged.vcf

for samp in $(bcftools query -l ${vcf})

do

printf '>'${samp}'\n'

bcftools query -s ${samp} -f '[%TGT]' ${vcf}

printf '\n'

done
```

Building phylogenies with wgSNPs

RAxML

- Two models of approximation: CAT and GAMMA
  - Do not use CAT if you have less than 50-100 taxa in your input file - use GAMMA instead
  - CAT is better than GAMMA if using protein sequence as it accommodates rate heterogeneity
- Bootstrap over 100 replicates
- Can have BIN or GTR
  - BIN is for binary (presence/absence)
  - GTR = ‘General Time Reversible’ model of nucleotide substitution under Gamma model of rate heterogeneity, which reduced computational burden
  - -p random seed
  - -x rapid bootstrap random seed
  - -f a = bootstrap
  - -N = number of bootstrap iterations
- `raxml -s file.fa -m GTRGAMMA -p 12345 -f a -x 12345 -N 100 -n trichophyton -w /path/to/output/directory`

Phylogeny visualisation

FigTree

- Download FigTree and visualise your RAxML phylogeny
  - https://github.com/rambaut/figtree/)

Phylogeny and visualisation

R

- Can create basic phylogenies in R also
- Install packages ggtree (and ggplot2 if you don’t already have it), vcfR, ape, adegenet, phangorn, and some colour packages like RColorBrewer and viridis to make it look pretty (don’t forget to load after install)
- Import the vcf into R and convert into a DNAbin object:

```
Ti_vcf=read.vcfR(“trichophyton.vcf”)

Ti_dna=vcfR2DNAbin(Ti_vcf)
```

Phylogeny and visualisation

R

- Compute genetic distances using Tamura & Nei's model which allows for different rates of transitions and transversions, hetergeneous base frequencies, and between-site variation of substitution rate

distance=dist.dna(Ti_dna,model="TN93")

- Then build a basic neighbour joining tree to get a representation of genetic distance between individuals:

nj_tree=nj(distance)

plot(nj_tree)

plot(nj_tree,type="unrooted")

- Why is the second plot better for interpretation?

Phylogeny and visualisation
R

- Extras:
  - ‘bionj’ is an improved version of neighbour joining
  - Assess the quality of the NJ phylogeny - is it appropriate for the dataset?
    - The TN93 model we chose is flexible so should be a good choice

```
x=as.vector(distance)

y=as.vector(as.dist(cophenetic(nj_tree))

plot(x,y)

abline(lm(y~x),col=“red”)

cor(x,y)^2
```

- Is the NJ tree appropriate for the dataset?

Phylogeny and visualisation
R - extras

- Bootstrapping:
  - ‘boot.phylo’ to bootstrap the tree e.g. bootstrapped_tree=boot.phylo(nj_tree,Ti_dna,function(e)root(nj(dist.dna(e,model=“TN93)),1))
  - Takes a bit longer to run!
  - This gives the number of times each node was identified in bootstrapped analyses
- Maximum parsimony:
  - Simple way to infer phylogenies for data with low divergence i.e. low substitution rate
  - phangorn package - convert the data: Ti_dna2=as.phyData(Ti_dna)
  - Ti_pars=optim.parsimony(nj_tree,Ti_dna2)
    - Is this a better fit?

Phylogeny visualisation

R/ggtree

- Can compare the tree you built with RAxML and then the R tree. Import the RAxML tree:

library(ggtree)

library(ggplot2)

setwd() <- set to directory you're working in/where RAxML tree is

TrichophytonTree=read.tree("RAxML_bipartitions.trichophyton")

ggtree(TrichophytonTree)

Phylogeny visualisation

R/ggtree

- Can you add metadata?
- Download metadata.csv from the GitHub repo and import to R

```
metadata=read.csv(“metadata.csv”,header=T)

p=ggtree(TrichophytonTree)

p=p %<+% metadata

p1=p+geom_tippoint(mapping=aes(colour=Country),size=3,alpha=.75)
```