Other Projects **Fungal Phylogeny**

Description

The idea is to create a large scale phylogenomic tree and then test things like codon-usage, GC content and gene-functions. I will be using BUSCO to create a set of orthologous genes. The genomes were downloaded from NCBI unsing biomartr: github.com/ropensci/biomartr look at the corresponding note NCBI Genome download, to see how this was done.

The projects folder is in /data/scratch/philipp/projects/fungal_phylo.

Todo list:

- [x] Download genomes
- [x] get taxonomy
- [x] run busco
- [x] create alignments
- [x] align alignments
- [x] run trimal
- [ ] run iqtree on individual genes (graz)

Running BUSCO on the set of downloaded ge…

For this I use the script 01_run_busco_all.sh:

```
#!/bin/bash
cd results
mkdir busco
cd busco
files=$(ls ../../data/fungi_incl_some_lichens/*.gz)
for file in $files
do
 gunzip $file
 echo "Preparing to run BUSCO on: "
 echo $file
 uncompressed_file=$(echo $file | sed -e "s/\.gz//g")
```

```
outname=$(echo $uncompressed_file | sed -e "s/\.fna//g")
#outname=$(echo $outname | sed -e "s/\.\. //g")
outname=$(basename $outname)
echo $outname
echo $file
python /usr/local/src/busco3/scripts/run_BUSCO.py -i $uncompressed_file -o
busco_$outname -l /usr/local/src/busco3/fungi_odb9 -m genome -c 8
gzip $uncompressed_file
done
```

Preparing concatenated phylogenomic analysis

Getting the taxon names for use with concat:

```
awk '{print $1}' busco_table_asco_processed.txt | sed -e 's/\"//g' >
IDS_for_tree.txt
```

I then created a concatenated alignment using the phyloscript pipeline. Only afterwards I ran trimAl to remove uninformative sites.

Selecting genes for phylogenetic analysis and…

I wrote a small python script to create a summary table of all the buscos that are there:

```
#!/usr/bin/env python2
# This script will get all ascomycota busco3 hmms and look in the busco
results of 700+ fungal genomes for the presence of the files

import os

hmms = os.listdir("/usr/local/src/busco3/ascomycota_odb9/hmms/")
hmms = [hmm.strip(".hmm") for hmm in hmms]

genomes = os.listdir("/data/scratch/philipp/projects/fungal_phylo/results/")

string = "species" + "\t".join(hmms)
print string
for species in genomes:
```

```
outstring = species +"\t"
for hmm in hmms:
     buscos = os.listdir("/data/scratch/philipp/projects/fungal_phylo/
results/" + species + "/single_copy_busco_sequences/")
     if hmm+".fna" in buscos:
          outstring += "\t"
          outstring += "1"
     else:
          outstring += "\t"
          outstring += "0"
print outstring
```

Using this table as an input for R I created two plots showing the relative BUSCO completenes of each species: percent_overview_species.pdf

and also for each BUSCO gene: percent_overview_busco.pdf

Alternatively (and much better) it is to run multiqc on the busco runs. Which I did.

I will remove species for which less than 80% of BUSCOs are present. The corresponding R script for this analysis is here: busco_overview.r

The output table is then used as input to the next python script which generates FASTA files for each busco and species: It is called 03_create_alignments.py

```
#!/usr/bin/env python
# this script will create fasta files for all the buscos from all ascomycete
species with >80% of buscos present
import os
import pandas as pd
from Bio import SeqIO
busco_overview = pd.read_csv("/data/scratch/philipp/projects/fungal_phylo/
results/busco_table_asco_processed.txt", sep="\t")
genomes = os.listdir("/data/scratch/philipp/projects/fungal_phylo/results/
busco/")
species_list = busco_overview["species"].tolist()
print(len(species_list))
#print(species_list)
#first remove species with too low busco coverage
```

```
busco_overview = busco_overview.set_index("species")
for sp in species_list:
 if busco_overview.loc[sp, "percent_complete"] < 0.8:
      busco_overview = busco_overview.drop([sp])
species_list =  list(busco_overview.index)
print(len(species_list))
#now loop through each busco and extract sequence for each species
buscos = list(busco_overview.columns.values)
for busco in buscos:
 print("Processing: " + busco)
 outfile = open("results/alignments/"+busco+"_all.fas", "w")
 for species in species_list:
      for genome in genomes: # this loop is to get the correct directory
name, it is very unelegant
           if species in genome:
                try:
                     seqfile = open("results/busco/"+genome+"/
single_copy_busco_sequences/"+busco+".faa", "r")
                     for seq_record in SeqIO.parse(seqfile, "fasta"):
                          name = ">" +species+"\n"
                          outfile.write(name)
                          outfile.write(str(seq_record.seq)+"\n")
                except: # skip missing buscos
                     continue
 outfile.close()
```

The next script will align all the files with mafft (04_align_sequences.sh):

```
#!/bin/bash
files=$(ls /data/scratch/philipp/projects/fungal_phylo/results/alignments/
*.fas)
cd results
cd alignments
for file in $files
do
 echo $file
 mafft --auto $file > $file"_aligned"
done
cd ..
mkdir alignments_aligned
mv alignments/*_aligned alignments_aligned/
```

The next script will filter the alignments with trimal (05_trimal_alignments.sh):

```
#!/bin/bash
files=$(ls results/alignments_aligned/*_aligned)
for file in $files
do
 echo $file
 trimal -gappyout -in $file -out $file"_trimmed"
done
cd results
mkdir alignments_trimmed
mv alignments_aligned/*_trimmed alignments_trimmed/
```

The next step is to run single-gene trees on the single busco alignments.