Coding statements back up

# Good to know

I used the alias “$Sequences\_Anouk” a lot throughout my commands here as well as in my scripts. This is because it’s the easiest way to access my files on the G-drive, which is somewhat complicated by the fact there is a space in the Folder name “Group Crous”. So $Sequences\_anouk can be seen a as a sort of “Home” equivalent.

Furthermore, so commands may be broken due to the fact I have moved a lot of files after I was done with the project to export it all in a single .zip-file. Most importantly, the folder “scripts” used to be on my H-drive, but is now included in the current zip-file. Thus, if the re-running of a command is giving issues, make sure to check to locations of files referred to.

I used conda to download some of the necessary software, my conda environments at the end of the project were:  


If the cohesion of commands makes no sense, it could help to check “Thoughts while working”

If there are any questions, the best place to reach me is my personal email: jasperbell01@gmail.com

# Coding statements

**#Activating conda to get to linux environment**

wsl

**#Access Jerome group HPC**

Ssh j.bell@SRV-LNX-CMAR1

password: KeeperKiD#321#D10

**#Getting explicit output of what the command line is doing and thinking**

set -xv

**#Path to Sequences\_Anouk folder on G drive**

‘~/g/Group Crous/000INTERNS/Jasper/Sequences\_Anouk’

To use variable: “$Sequences\_Anouk”

**#Reading the lines of a file one by one**

while IFS=read -r line

do

echo “$line”

done < file.txt

**#Extracting only the strain name from the file name**

## into variable “name”

for file in Sequences/fusarium\_assemblies\_Germany/\*; do name=$(echo $file | cut -d "/" -f 3 | cut -d "-" -f 1); done

##Print to STDOUT

for file in Sequences/fusarium\_assemblies\_Germany/\*.fasta; do echo $file | cut -d "/" -f 3 | cut -d "-" -f 1; done

**#Running TeloVision** on all files in a folder and outputting based on only strain name  
## Might be necessary to activate conda television environment?

for file in $$$.fasta; do output\_name=$(basename "$file" | cut -d "." -f 1); telovision -i $file -o ~/h/TeloVisionOut/$output\_name; done

#Loop for running **general statistics** on all sequences

for file in Sequences/fusarium\_assemblies\_Germany/\*.fasta ; do ~/h/scripts/general\_info.sh $file; done

for file in "$Sequences\_Anouk"/assemblies/\*.fa\*; do ~/h/scripts/general\_info\_no\_busco\_summary.sh $file; done

**#Finding file names that are unique to one directory**

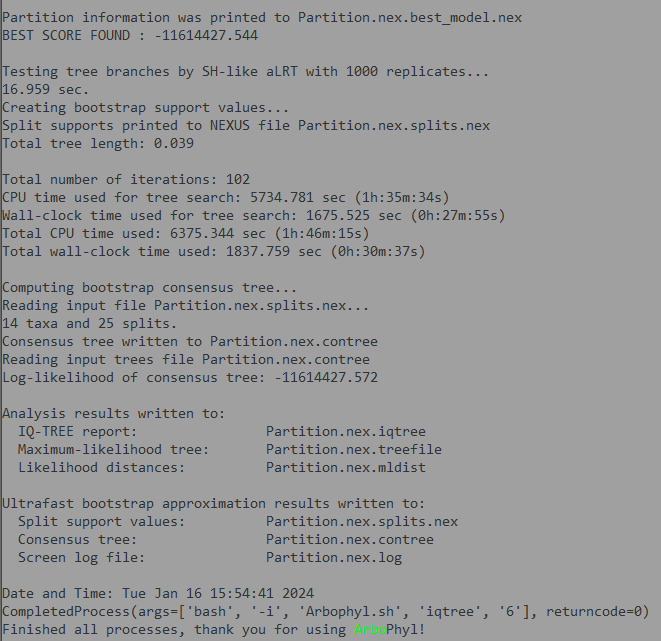
find . -mindepth 1 -maxdepth 2 -type f -printf '%p %f\n' | sort -t ' ' -k 2,2 | uniq -f 1 --unique | cut -d' ' -f1 | cut -d '/' -f3  
(original from <https://askubuntu.com/questions/48524/search-for-duplicate-file-names-within-folder-hierarchy>)

**download of Hypocreales lineage:**

from <https://busco-data.ezlab.org/v5/data/lineages/> comes file [hypocreales\_odb10.2020-08-05.tar.gz](https://busco-data.ezlab.org/v5/data/lineages/hypocreales_odb10.2020-08-05.tar.gz)

**Fusarium miscanthi genome downloaded from** <https://www.ncbi.nlm.nih.gov/datasets/genome/GCA_014898875.1/>

Strain found in doi 10.3114/sim.2022.104.02 fig 14.

**Arbophyl output results locations**

**Check\_parsimony\_sites**

#Running Check\_parsiminy\_sites on all iqtree files. Script is also used as a part of Compare\_genesets script.

## local command line

for file in ~/Sequences/True\_Fusarium\_Sequences/FilterBUSCOs\_output/MAFFT\_output/Trimmed\_MSAs/Passed\_MSA/Models/\*/\*\_trimmed.fna.iqtree; do ~/scripts/check\_parsimony\_sites.sh $file $OUTPUT\_FOLDER; done

## HPC Collemare

for file in ~/h/Sequences/True\_Oxysporum\_Sequences/FilterBUSCOs\_output/MAFFT\_output/Trimmed\_MSAs/Passed\_MSA/Models/\*/\*\_trimmed.fna.iqtree; do ~/h/scripts/check\_parsimony\_sites.sh $file $OUTPUT\_FOLDER; done

**Obtaining number of parsimony informative sites** from parsimonious\_genes.txt w

##When in True\_Oxysporum\_Sequences

sort -k 2n parsimonous\_genes.txt | cut -d "," -f2 | uniq -c | head -n 30

grep "at" non\_parsimonous\_genes.txt -c

I have many models now, I only want to keep those that are informative within FOSC. I have a list of what genes are informative in True\_Oxysporum\_Sequences/parsimonious\_genes. Any gene model not in this list should be removed

**BLAST**

# making a BLAST database from an assembly

makeblastdb -in ~/h/Sequences/fusarium\_assemblies\_Germany/Fusarium\_phialophorum-barcode15.fasta -dbtype nucl -parse\_seqids -blastdb\_version 5 -out OUTPUTNAME

Database name is “/home/jbell/Sequences/fusarium\_assemblies\_Germany/Fusarium\_XXX-barcodeYY.fasta”

Blast search “blastn -db /home/jbell/Sequences/fusarium\_assemblies\_Germany/Fusarium\_tanahbumbuense\_ex\_type-barcode17.fasta -query ~/h/BLAST/Tana\_genes/tef1a.fasta”

#Running local BLAST on Tana genome with genes from Ewald

for gene in Tana\_genes/\*.fasta; do out\_name=$(echo $gene | cut -d "." -f1); blastn -db /home/jbell/Sequences/fusarium\_assemblies\_Germany/Fusarium\_tanahbumbuense\_ex\_type-barcode17.fasta -query $gene -out $out\_name.out; done

for gene in Odo\_genes/\*.fasta; do out\_name=$(echo $gene | cut -d "." -f1 | cut -d "/" -f2); blastn -db ~/h/BLAST/databases/Indo12\_15 -query $gene -out "$out\_name.Indo12\_15.out"; done

grep "Iden" \*\_phia.Indo12\_15.out

**Compare\_genesets.sh**

#Running compare\_genesets.sh on two directories

## !! Directory names in the command have to finish on “/” otherwise code breaks

~/h/scripts/compare\_genesets.sh True\_Oxysporum\_Sequences/ Oxysporum\_two\_outgroup/

**IQtree**

#Running IQtree on all genetrees loop

Conda activate iqtree

for file in FilterBUSCOs\_output/MAFFT\_output/Trimmed\_MSAs/Passed\_MSA/Models/\*/\*\_trimmed.fna.log; do model=$(grep "Best-fit model:" $file | cut -d " " -f3); alignment\_file=$(echo ${file:0:-4}); output\_file=$((echo ${file:0:-16}) | sed 's/$/\_bootstrapped/'); test -e "$output\_file.treefile" && echo "$output\_file has already been computed" || iqtree -s "$alignment\_file" -m "$model" -nt AUTO --prefix "$output\_file" -redo; done

#Updated version for clade specific gene trees

for file in FilterBUSCOs\_output/MAFFT\_output/Trimmed\_MSAs/Passed\_MSA/Models/\*/\*\_trimmed.fna.log; do model=$(grep "Best-fit model:" $file | cut -d " " -f3); alignment\_file=$(echo ${file:0:-4}); output\_file=$((echo ${file:0:-16}) | sed 's/$/\_genetree/'); echo "working on $output\_file"; test -e "$output\_file.treefile" && echo "$output\_file has already been computed" || iqtree -s "$alignment\_file" -m "$model" -nt AUTO --prefix "$output\_file" -B 1000 -redo; done

**Compare\_trees.sh**

#Running compare\_trees.sh on a folder containing all genetrees (just cp them to a new folder)

Conda activate ete3

~/h/scripts/compare\_trees.py ~/h/Sequences/Oxysporum\_two\_outgroup/Partition.nex.treefile ~/h/Sequences/Oxysporum\_two\_outgroup/genetrees\_bootstrapped/ Fusarium\_lumajangense\_ex\_type-barcode10

# Copy genetree\_contree files to working space on H drive:

## When in Models/ folder

cp \*/\*\_genetree.contree ~/h/genetrees/Clade1/

#When in Genetrees, with species tree in Genetrees and gene trees in CladeX\_X folder

Conda activate ete3

python compare\_trees.py Clade1.contree ./Clade1 Fusarium\_lumajangense\_ex\_type-barcode10

**Gene concordance analysis**

#Running gCA

##Making the file that contains all gene trees

iqtree -s FilterBUSCOs\_output/MAFFT\_output/Trimmed\_MSAs/Passed\_MSA/ -S Partition.nex --prefix loci -T AUTO

~~## Calculating concordance factors~~

~~iqtree -t Partition.nex.contree --gcf loci.treefile --prefix concord~~

#Running sCA and gCA simultaniously

iqtree2 -t Partition.nex.contree --gcf loci.treefile --prefix Boot\_gCF\_sCF\_Tree --scf 100 -seed 977193 -p Partition.nex -T 1 -nt 4

**run\_OdoPhia\_blast**

conda activate blast

for genome in "$Sequences\_Anouk"/assemblies/\*; do ~/h/scripts/run\_OdoPhia\_blast.sh "$genome"; done

**Compare identity**

## Make sure to always add a number as second input as this is the minimal identity used in the script

~/h/scripts/compare\_identity.sh ~/h/BLAST/PurOdoPhia\_Blast/ 99

**Check what genes had no hits at all**

grep "No hits found" ./\*\_res/\*.out >> no\_alignment\_genes.txt

**Find the number of sequences that have a missing/short blast hit**

cat no\_alignment\_genes.txt | cut -d"." -f3 | sort | uniq >> lacking\_assemblies.txt

cat short\_alignments.txt | cut -d" " -f1 | sort | uniq >> lacking\_assemblies.txt

cat lacking\_assemblies.txt | sort | uniq | wc -l

**Working with output files of compare identity**

#Find the number of sequences that match highly similar to both odo and phia

grep "Both" sequence\_to\_species\_similarity.txt -c

#find the number of sequences with exactly 2 genes that were interpreted to be too short

grep "too" short\_alignments.txt | cut -d" " -f1 | uniq -c | grep "2 " -c

#Count the number of genes that did not have a BLAST hit at all

grep "No hits found" ./\*/\* -c | grep -v ":0" | wc -l

#Count the number of sequences that have exactly 1 gene without a BLAST hit

grep "." no\_alignment\_genes.txt | cut -d"." -f3 | uniq -c | sort | grep "1 " -c

#Find what sequences are listed as having both too short as well as missing BLAST hits

for Sequence in $(grep "." no\_alignment\_genes.txt | cut -d"." -f3 | uniq | sort); do if [ $(grep "$Sequence" short\_alignments.txt -c) -gt 0 ]; then echo "$Sequence" >> missing\_and\_short\_sequences.txt; fi ; done

#Make a list of the combination of percent identities to the query genes, sorted and counted by combination type

grep "Odoratissimum" all\_identities.txt | grep "rpb2 [7-8][0-9][0-9]" | grep "rpb1 1[2-9][0-9][0-9]" | grep "tef1 5[1-9][0-9]" | cut -d" " -f5,8,11 | sort -rn | uniq -c

**Make file of all incomplete (missing or short gene) sequences (and counting number of entries)**

grep "." no\_alignment\_genes.txt | cut -d"." -f3 | uniq | sort >> incomplete\_sequences.txt

grep "too" short\_alignments.txt | cut -d" " -f1 | uniq >> incomplete\_sequences.txt

grep "[A-B,0-9]" incomplete\_sequences.txt | sort | uniq >> temp.txt

rm incomplete\_sequences.txt

mv temp.txt incomplete\_sequences.txt

grep "[A-B,0-9]" incomplete\_sequences.txt | wc -l

**Count number of .faa files (BUSCO gene sequences) for each sequence in incomplete\_sequences.txt**

##NOT FULY TESTED YET – I think it works but there are many folders not unpacked yet, which causes the loop to raise many errors about no such file or directory.

while IFS= read -r line; do cd "$Sequences\_Anouk"/Busco\_files/BUSCO/busco\_"$line"/; ls \*.faa -l | wc -l ; done < ~/h/BLAST/PurOdoPhia\_Blast/incomplete\_sequences.txt

**Find sequences with non-sufficient hit that have Busco completeness > 90%**

while IFS= read -r line; do if [ $(grep "\_$line " "$Sequences\_Anouk"/low-busco\_assemblies.txt -c) -eq 0 ]; then grep "\_$line " "$Sequences\_Anouk"/busco\_completeness.txt; fi; done < ~/h/BLAST/PurOdoPhia\_Blast/incomplete\_sequences.txt

**Get the BLAST hit with the highest identity**

for file in ~/h/BLAST/PurOdoPhia\_Blast/SRR6118281\_res/\*; do echo $file; grep "Identities" $file | awk '{print $3 "\t" $4}' | cut -d "/" -f1,2 | cut -d "%" -f1 | sed 's/(//g' | sed 's/\//\t/g' | cut -d $'\t' -f2,3 |sort -k1,1 --numeric -r | head -n1; done

**Find blast searches that returned multiple hits**

for file in ~/h/BLAST/PurOdoPhia\_Blast/\*/\*; do if (( $(grep "Identities" $file -c) > 1 )); then echo $file; grep "Identities" $file; fi; done

**Remove all files with .gff suffix**

find . -name "\*.gff" -type f -delete

**Selecting strains to be reran**

for folder in ./\*; do if [ $(ls "$folder" | grep ".fna" -c) -eq 0 ]; then echo "${folder:8}" >> ../no\_fna\_files.txt; fi ; done

while IFS= read -r line; do completeness=$(grep "$line" ../../busco\_completeness.txt | cut -d" " -f2); if [ "$completeness" -gt 94 ]; then echo "$line" >> ../to\_rerun.txt ; fi; done < ../no\_fna\_files.txt

**Calculate number of busco genes found and percentage completeness**

for folder in "$Sequences\_Anouk"/Busco\_files/BUSCO/\*; do number\_of\_genes=$(ls "$folder"/run\_hypocreales\_odb10/busco\_sequences/single\_copy\_busco\_sequences | grep ".fna" -c); completeness=$(awk -v n="$number\_of\_genes" 'BEGIN{percent\_iden=n/4494\*100; printf "%.1f", percent\_iden }') ; strain=$(echo "$folder" | cut -d"/" -f11 | cut -d"\_" -f2,3,4,5,6) ; echo "working on: $strain" ; echo "$strain" "$number\_of\_genes" "$completeness" >> "$Sequences\_Anouk"/busco\_completeness.txt ; done

**Isolate assemblies with <95% busco completeness (4269.3 genes)**

while IFS= read -r line; do if [ $(echo "$line" | cut -d " " -f2) -lt 4270 ]; then echo "$line" >> low\_busco\_completeness\_assemblies.txt; fi; done < “$Sequences\_Anouk”/busco\_completeness.txt

**Counting the number of hits for strains in which the best BLAST hit is of at least 90% of the query length**

while IFS= read -r line  
do  
species=$(echo "$line" | cut -d" " -f16)  
strain=$(echo "$line" | cut -d" " -f1)  
if [[ "$species" == "Odoratissimum" ]]; then  
 if [ "$(echo "$line" | cut -d" " -f4)" -ge 789 ]; then  
 grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb2\_Odo."$strain".out -c  
 fi  
 if [ "$(echo "$line" | cut -d" " -f7)" -ge 1224 ]; then  
 grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb1\_Odo."$strain".out -c  
 fi  
 if [ "$(echo "$line" | cut -d" " -f10)" -ge 515 ]; then  
 grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/tef1\_Odo."$strain".out -c  
 fi  
fi  
if [[ "$species" == "Phialophorum" ]]; then  
 if [ "$(echo "$line" | cut -d" " -f4)" -ge 789 ]; then  
 grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb2\_phia."$strain".out -c  
 fi  
 if [ "$(echo "$line" | cut -d" " -f7)" -ge 1300 ]; then  
 grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb1\_phia."$strain".out -c  
 fi  
 if [ "$(echo "$line" | cut -d" " -f10)" -ge 515 ]; then  
 grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/tef1\_phia."$strain".out -c  
 fi  
fi  
done < ~/h/BLAST/PurOdoPhia\_Blast/all\_identities.txt | sort --numeric | uniq -c

**Same as above, but now exporting what sequences and what genes have how many hits**

while IFS= read -r line

do

species=$(echo "$line" | cut -d" " -f16)

strain=$(echo "$line" | cut -d" " -f1)

if [[ "$species" == "Odoratissimum" ]]; then

if [ "$(echo "$line" | cut -d" " -f4)" -ge 775 ]; then

n\_hits=$(grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb2\_Odo."$strain".out -c)

if [ "$n\_hits" -gt 1 ]; then

echo "$strain rpb2 $species $n\_hits" >> genes\_with\_multiple\_hits.txt

fi

fi

if [ "$(echo "$line" | cut -d" " -f7)" -ge 1224 ]; then

n\_hits=$(grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb1\_Odo."$strain".out -c)

if [ "$n\_hits" -gt 1 ]; then

echo "$strain rpb1 $species $n\_hits" >> genes\_with\_multiple\_hits.txt

fi

fi

if [ "$(echo "$line" | cut -d" " -f10)" -ge 515 ]; then

n\_hits=$(grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/tef1\_Odo."$strain".out -c)

if [ "$n\_hits" -gt 1 ]; then

echo "$strain tef1 $species $n\_hits" >> genes\_with\_multiple\_hits.txt

fi

fi

fi

elif [[ "$species" == "Phialophorum" ]]; then

if [ "$(echo "$line" | cut -d" " -f4)" -ge 789 ]; then

n\_hits=$(grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb2\_phia."$strain".out -c)

if [ "$n\_hits" -gt 1 ]; then

echo "$strain rpb2 $species $n\_hits" >> genes\_with\_multiple\_hits.txt

fi

fi

if [ "$(echo "$line" | cut -d" " -f7)" -ge 1300 ]; then

n\_hits=$(grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/rpb1\_phia."$strain".out -c)

if [ "$n\_hits" -gt 1 ]; then

echo "$strain rpb1 $species $n\_hits" >> genes\_with\_multiple\_hits.txt

fi

fi

if [ "$(echo "$line" | cut -d" " -f10)" -ge 515 ]; then

n\_hits=$(grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/tef1\_phia."$strain".out -c)

if [ "$n\_hits" -gt 1 ]; then

echo "$strain tef1 $species $n\_hits" >> genes\_with\_multiple\_hits.txt

fi

fi

fi

done < ~/h/BLAST/PurOdoPhia\_Blast/all\_identities.txt

**Find the length and identity of all hits in genes where the best hit is at least 90% query length**

while IFS= read -r line; do strain=$(echo "$line" | cut -d" " -f1); gene=$(echo "$line" | cut -d" " -f2); species=$(echo "$line" | cut -d" " -f3); if [ “$species” == “Odoratissimum” ]; then species=’Odo’; elif [ “$species” == Phialophorum ]; then species=’phia’; fi; echo "Searching in $strain $species $gene" ; grep "Identities" ~/h/BLAST/PurOdoPhia\_Blast/"$strain"\_res/"$gene"\_"$species"."$strain".out | awk '{print $3 "\t" $4}' | cut -d "/" -f1,2 | cut -d "%" -f1 | sed 's/(//g' | sed 's/\//\t/g' | cut -d $'\t' -f2,3 |sort --numeric -k1,1 -r ; echo " "; done < genes\_with\_multiple\_hits.txt

**Making corrected\_all\_identities\_long\_hits.txt**

while IFS= read -r line

do

species=$(echo "$line" | cut -d" " -f16)

rpb2\_length=$(echo "$line" | cut -d" " -f4)

rpb1\_length=$(echo "$line" | cut -d" " -f7)

tef1\_length=$(echo "$line" | cut -d" " -f10)

if [ "$species" == "Odoratissimum" ]; then

if [ "$rpb2\_length" -ge 775 ]; then

if [ "$rpb1\_length" -ge 1224 ]; then

if [ "$tef1\_length" -ge 515 ]; then

echo "$line" >> corrected\_all\_identities\_long\_hits.txt

fi

fi

fi

elif [ "$species" == "Phialophorum" ]; then

if [ "$rpb2\_length" -ge 775 ]; then

if [ "$rpb1\_length" -ge 1224 ]; then

if [ "$tef1\_length" -ge 515 ]; then

echo "$line" >> corrected\_all\_identities\_long\_hits.txt

fi

fi

fi

fi

done < corrected\_all\_identities.txt

while IFS= read -r line

do

species=$(echo "$line" | cut -d" " -f16)

rpb2\_length=$(echo "$line" | cut -d" " -f4)

rpb1\_length=$(echo "$line" | cut -d" " -f7)

tef1\_length=$(echo "$line" | cut -d" " -f10)

if [ "$rpb2\_length" -ge 774 ]; then

if [ "$rpb1\_length" -ge 1300 ]; then

if [ "$tef1\_length" -ge 517 ]; then

echo "$line" >> all\_identities\_long\_hits.txt

fi

fi

fi

done < all\_identities.txt

**Counting the number of assemblies that had long hits for all three query genes but low busco completeness**

while IFS= read -r line; do strain=$(echo "$line"| cut -d$'\t' -f1); if [ $(grep "$strain" "$Sequences\_Anouk"/busco\_complete\_assemblies.txt -c) -ne 1 ]; then echo "$strain"; fi; do

ne < sequence\_to\_species\_similarity.txt | sort | uniq | wc -l

**Making tree with MASH**

conda activate mashtree

cd "$Sequences\_Anouk "/assemblies

mashtree ./\* --outtree ~/h/Trees/mashtree\_complete --sort-order random --numcpus 3 --mindepth 0

## Clade specific pipeline – whole assessment

-- Make directory ~/h/BLAST/\*\*\*\_Blast --

-- Get ex-type genes from fusarium.org/NCBI –

-- Copy and adapt run\_\*\*\*\_blast.sh script for specific blast queries --

-- Copy ex-type assemblies to \*\*\*\_blast folder

Conda activate blast

for genome in ./\*.fasta; do ~/h/scripts/run\_\*\*\*\_blast.sh "$genome"; done

~/h/scripts/run\_\*\*\*\_blast.sh $STRAIN; done

-- Compare GenBank ex-type to assembly ex-type –

-- If needed, update gene-sequences –

Conda activate blast

for genome in "$Sequences\_Anouk"/assemblies/\*; do ~/h/scripts/run\_\*\*\*\_blast.sh "$genome"; done

-- Copy compare\_identity.sh and add Species definition as compare\_identity\_\*\*\*.sh –

-- Edit compare\_identity\_\*\*\*.sh to correct gene lengths and output locations/names --

~/h/scripts/compare\_identity.sh ~/h/BLAST/\*\*\*\_Blast/ 99

-- Interpret BLAST results --

**Running augustus to annotate genes**

Conda activate augustus

augustus --species=fusarium\_graminearum --gff3=on --codingseq=on "$Sequences\_Anouk"/assemblies/new\_assemblies/Fusarium\_odoratissimum\_ex\_type-barcode11.fasta > ~/h/Sequences/OdoPhia\_Outgroup/Synteny/odo\_ex\_type.gff3

**Loop for running the remaining genome predictions**

for file in SRR\*.fasta; do echo "working on $file" ; output\_name=$(echo "$file" | cut -d"." -f1); augustus --species=fusarium\_graminearum --gff3=on --codingseq=on $file > "$output\_name".gff3 ; done

**Transform gff output to fasta sequences for found genes – makes codingseq file**

getAnnoFasta.pl odo\_ex\_type.gff3

**Transforming gff output to bed format**

Conda activate mcscan

python -m jcvi.formats.gff bed odo\_ex\_type.gff3 -o odo\_ex\_type.bed

**Reformat codingseq file to proper fasta format (in .cds file)**

python -m jcvi.formats.fasta format odo\_ex\_type3.codingseq odo\_ex\_type.cds

**Change header names in cds file such that they match with .bed file column 4**

#CHECK BED FILE IF CONTIG NAMES NEED TO BE ADAPTED

sed -i 's/contig\_.\*g/g/g' purpur\_ex\_type.cds

sed -i 's/scaffold\_.\*g/g/g' purpur\_ex\_type.cds

sed -i 's/\.t1//g' phia.cds

sed -i 's/NODE.\*\.g/g/g' SRR3139015.cds

sed -i 's/\.t1//g' phia.cds

**Run actual homology comparison (automatically generates dotplot)**

python -m jcvi.compara.catalog ortholog odo\_ex\_type purpur\_ex\_type

**Make histogram of number of hits per anchor**

python -m jcvi.compara.synteny depth --histogram odo\_ex\_type.purpur\_ex\_type.anchors

**Make karyotype plot**

python -m jcvi.compara.synteny screen --minspan=30 --simple odo\_ex\_type.purpur\_ex\_type.anchors odo\_ex\_type.purpur\_ex\_type.anchors.new

python -m jcvi.graphics.karyotype seqids layout -o karyotype

#For C058 I had to edit the contig names as well

python -m jcvi.graphics.karyotype seqids\_temp layout\_temp -o karyotype.temp

**Make blocks plot thingy**

python -m jcvi.compara.synteny mcscan cuge\_ex\_type.bed cuge\_ex\_type.odo\_ex\_type.lifted.anchors --iter=1 -o cuge\_ex\_type.odo\_ex\_type.i1.blocks

**IN CASE NEEDED: editing .bed file contig names for easier plotting**

while IFS= read -r line ; do name=$(echo "$line" | cut -f1 | cut -d"\_" -f1,2); data=$(echo "$line" | cut -f2,3,4,5,6) ; echo -e "$name\t$data" >> SRR10428605\_2.bed; done < SRR10428605.bed

mv SRR10428605\_2.bed SRR10428605.bed

rm SRR10428605\_2.bed

**constrained\_tree\_comparison**

*#To call simply give a clade level folder (so the one that holds Arbophyl and FilterBUSCOs) to the function:*

Conda activate iqtree

bash ~/h/scripts/constrained\_tree\_comparison.sh "$Sequences\_Anouk"/Clade1

**Reverse contigs**

bash ~/h/scripts/reverse\_contig.sh Fusarium\_purpurascens\_ex\_type-barcode13.fasta "$Sequences\_Anouk"/Synteny\_ex\_types/purpur.bed contig\_2

awk -v contig\_length=6716093 -v contig="contig\_2\t" '

BEGIN{FS=OFS="\t"}

$1 ~ /contig/ {

$2 = (contig\_length - $2); $3 = (contig\_length - $3)}1' ../Synteny\_ex\_types/purpur.bed | head

**Computing genetrees with branch lengths and bootstrap for use in wASTRAL**

for gene in \*; do if [ $(grep -c "Working on $gene" "$Sequences\_Anouk"/rerun\_genetrees\_bootstrapped.log) -eq 0 ]; then echo "Working on $gene"; iqtree -s "$gene"/"$gene"\_trimmed.fna -m MFP --redo-tree -B 1000 --prefix "$gene"/"$gene"\_bootstrapped -quiet -nt AUTO; fi; done >> "$Sequences\_Anouk"/rerun\_genetrees\_bootstrapped.log

**wASTRAL-h (ASTER)**

cd “$Sequences\_Anouk”/ASTER

bin/wastral -t 10 --root Fusarium\_lumajangense\_ex\_type-barcode10 -x 100 -n 0 -u 2 -i ../Species\_tree/ASTRAL/genetrees\_bootstrapped.txt -o "$Sequences\_Anouk"/Species\_tree/ASTRAL/wASTRALh\_tree 2> "$Sequences\_Anouk"/Species\_tree/ASTRAL/wASTRALh\_log.log

bin/wastral -t 10 -r 16 -s 16 --root Fusarium\_lumajangense\_ex\_type-barcode10 -x 100 -n 0 -u 2 -i ../Species\_tree/ASTRAL/genetrees\_bootstrapped.txt -o "$Sequences\_Anouk"/Species\_tree/ASTRAL/wASTRALh\_tree2 2> "$Sequences\_Anouk"/Species\_tree/ASTRAL/wASTRALh\_log2.log

**Removing pp values from wASTRAL-h output tree to generate topology for CASTLES**

sed -i "s/\[[^]]\*\]//g" wASTRALh\_tree2\_topology

removed single quotes by hand

sed -i -E 's/\[pp1=([^;]+);[^]]\*\]/\1/g' wASTRALh\_tree2\_CU\_pp #Removing all but pp1 value

sed -i -E "s/'//g" wASTRALh\_tree2\_CU\_pp

**Transforming CU to SU for branch lengths on ASTRAL tree using CASTLES**

Cd "$Sequences\_Anouk/ASTER"

bin/astral4 -i "$Sequences\_Anouk/Species\_tree/ASTRAL/genetrees\_bootstrapped.txt" -C -c "$Sequences\_Anouk/Species\_tree/ASTRAL/wASTRALh\_tree2\_topology" -o "$Sequences\_Anouk/Species\_tree/ASTRAL/wASTRALh\_tree\_CUpp\_SU" --root Fusarium\_lumajangense\_ex\_type-barcode10 --genelength 1689 2> "$Sequences\_Anouk/Species\_tree/ASTRAL/wASTRALh\_tree\_CUpp\_SU\_log.txt"

**Getting branch lengths from species tree Newick file**

grep -oP ":(\d\.[\de-]\*?)[,\)]" wASTRAL\_pp\_SU\_final.nw | cut -d":" -f2 | cut -d"," -f1 | cut -d")" -f1 | sed 's/\./,/g' > branch\_lengths.txt

(to calculate average and mean import the data in excel, way easier and commas are already in place for correct formatting)

**Writing (almost all) subfiles of final xml file and compiling**

cd "$Sequences\_Anouk/Species\_tree/gene\_MSAs"

bash ~/h/scripts/xml\_line\_writer.sh

cd ../STACEY

*\*Copy files from Safekeeping to main STACEY folder\**

cat 1\_Start\_Data\_Sequences.xml 2\_maps.xml 3\_run\_taxonsets.xml 4\_run\_TreeIDs.xml 5\_run\_init.xml 6\_run\_distribution.xml 7\_run\_priors.xml 8\_run\_CompoundDistribution.xml 9\_run\_geneTree.xml 10\_run\_NodesNudge.xml 11\_run\_FNHS.xml 12\_run\_CPR.xml 13\_run\_TBA.xml 14\_run\_PPSS\_updown100.xml 15\_run\_updown350.xml 16\_run\_updown700.xml 17\_run\_updown900.xml 18\_run\_updown970.xml 19\_run\_updown990.xml 20\_run\_updown997.xml 21\_run\_updown999.xml 22.xml 23\_treescaler.xml 24.xml 25\_Tracelog.xml 26\_Other\_Loggers.xml >> TACEY\_setup.xml

Beast -validate STACEY\_setup.xml

**Select random subset of 355 genes from all genes for subset-STACEY run**

for file in $(ls | shuf -n355); do cp $file ../STACEY/RanSubset/Genes/ ; done

**Altering scalefactor for genes in XML file**

*sed -i "s/spec=\"ScaleOperator\" scaleFactor=\"0.5\"/spec=\"ScaleOperator\" scaleFactor=\"0.9\"/g" RanSub\_STACEY\_setup.xml*

**STACEY run** (seed for RanSub run)

*\*In directory with setup xml\**

beast -seed 245152 -prefix RanSub\_STACEY -threads 6 RanSub\_STACEY\_setup.xml

beast -seed 245151 -prefix run5\_RanSub/RanSub\_STACEY -resume -statefile run5\_RanSub/000\_Statefile\_run5 -threads 6 run5\_RanSub\_STACEY\_setup.xml >> run5\_RanSub/000\_ScreenLog\_STACEY.txt