Code

TF Phylogenetic Trees - Lab 2 rotation

Download data and IPR#'s from uniprot website (https://www.uniprot.org/ (https://www.uniprot.org/)).

This will include the fasta and tsv files for each of the four superfamilies of interest in each of the species.

The four transcription superfamilies of interest include: Homeobox Domain = IPR001356; BHLH = IPR011598; HMG = IPR036910; T-box = IPR046360

The five species to download for each of the above IPR#'s: Human; C. elegans; Drosophila; Amphimedon queenslandica; Mizuhopecten yessoensis

Transfer downloaded fasta and tsv files from my computer to lisc server

Hide

```
scp -r ~/Downloads/lab2_fasta_tsv janicek@login02.lisc.univie.ac.at:/scratch/molevo/janicek/Lab2/data/lab2_fasta_tsv/
```

Fasta and TSV files

can be found in the separate species directories

Hide

```
cd /scratch/molevo/janicek/Lab2/data/amphimedonqueensland/
cd /scratch/molevo/janicek/Lab2/data/mizuhopectenyessoensis/
cd /scratch/molevo/janicek/Lab2/data/human
cd /scratch/molevo/janicek/Lab2/data/celegans
cd /scratch/molevo/janicek/Lab2/data/drosophila/
```

Multiple fasta files for each IPR

contains the five reference species and are found in specific directories.

Hide

```
cd /scratch/molevo/janicek/Lab2/data/multiple_fasta_IPR001356/
cd /scratch/molevo/janicek/Lab2/data/multiple_fasta_IPR011598/
cd /scratch/molevo/janicek/Lab2/data/multiple_fasta_IPR036910/
cd /scratch/molevo/janicek/Lab2/data/multiple_fasta_IPR046360/


ls >> fasta_001356_list.txt #creates a list for each of the 5 species fasta files
ls >> fasta_011598_list.txt
ls >> fasta_036910_list.txt
ls >> fasta_046360_list.txt

#change the following code to get the correct IPR fasta files into their respective l
ists
#ls /scratch/molevo/janicek/Lab2/data/multiple_fasta_IPR001356/IPR001356/IPR001356-19
*.fasta* > /scratch/molevo/janicek/Lab2/data/multiple_fasta_IPR001356/IPR001356/fasta
_001356_list.txt
```

IPS script (version 5.61-93.0-11.0.4)

currently set to AAUR (Aurita/Aurelia) data, but change values for NV2 (Nvectensis). IPS or interproscan is used to sequence the raw internal data so that it can be compared to the Uniprot/Interpro data for the other species. IPS enables the user to identify the gene families that a protein sequence belongs to. This was critical to be able to select out only the genes of interest.

Hide

```
#!/bin/bash

#SBATCH --job-name=ips
#SBATCH --nodes=1
#SBATCH --partition=basic
#SBATCH --cpus-per-task=16
#SBATCH --mem=8GB
#SBATCH --time=5:0:0
#SBATCH --output=/scratch/molevo/janicek/Lab2/logs/aaurita/ips_%j_%a.log #../../nvect
ensis/ips_%j_%a.log
#SBATCH --error=/scratch/molevo/janicek/Lab2/logs/aaurita/ips_%j_%a.err ##../../nvect
ensis/ips_%j_%a.err
#SBATCH --export=ALL
#SBATCH --mail-type=ALL
#SBATCH --mail-user=a12110422@unet.univie.ac.at

###ENVIRONMENT
module load interproscan
module list

###CONSTANTS
wd="/scratch/molevo/janicek/Lab2"
res="${wd}/results"
od="${res}/ips"
prots=( ${wd}/data/aaurita/chunk*.fa ) #/data/nvectensis/chunk*.fa
ips="/scratch/mirror/interpro/interproscan-5.61-93.0/interproscan.sh"

###VARIABLES
prot=${prots[$SLURM_ARRAY_TASK_ID]}
base=`basename $prot`
out=${base%.*}_ips

###EXECUTION
echo "Started at `date`"

echo "mkdir -p ${od}"
mkdir -p ${od}

echo "bash ${ips} -cpu 16 -b ${od}/${out} -etra -f GFF3,TSV,XML -goterms -pa -i ${pro
t} -t p -T ${TMPDIR}"
bash ${ips} -cpu 16 -b ${od}/${out} -etra -f GFF3,TSV,XML -goterms -pa -i ${prot} -t
p -T ${TMPDIR}

echo "Finished at `date`"
```

Generates a list from tsv files for nematostella and aurelia

Hide

```
# make sure in correct directory. /scratch/students/janicek/Lab2/results/ips.nvectens
is or aaurita
fgrep -hw "IPR001356" chunks_*.tsv | cut -f 1 | sort | uniq > IPR001356.list
fgrep -hw "IPR011598" chunks_*.tsv | cut -f 1 | sort | uniq > IPR011598.list
fgrep -hw "IPR036910" chunks_*.tsv | cut -f 1 | sort | uniq > IPR036910.list
fgrep -hw "IPR046360" chunks_*.tsv | cut -f 1 | sort | uniq > IPR046360.list

cat IPR001356.list #will show what is in the list
cat IPR011598.list
cat IPR036910.list
cat IPR046360.list
```

PFAM download

These sequences will be based on the specific proteins of interest and will be needed to be able to extract coordinates within the motifs. Using the uniprot website, the PFAM hmm files to download: PF00046 = IPR001356(homeobox); PF00010 = IPR011598(bHLH); PF00505 = IPR036910(HMG); PF00907 = IPR046360(Tbox)

#Transfered the downloaded hmm files for the PFAM from my computer to lisc server

Hide

```
scp -r ~/Downloads/PF001356.hmm.gz janicek@login02.lisc.univie.ac.at:/scratch/molevo/
janicek/Lab2/data/
scp -r ~/Downloads/PF011598.hmm.gz janicek@login02.lisc.univie.ac.at:/scratch/molevo/
janicek/Lab2/data/
scp -r ~/Downloads/PF036910.hmm.gz janicek@login02.lisc.univie.ac.at:/scratch/molevo/
janicek/Lab2/data/
scp -r ~/Downloads/PF046360.hmm.gz janicek@login02.lisc.univie.ac.at:/scratch/molevo/
janicek/Lab2/data/
```

#Decompress the transferred hmm files

Hide

```
pigz -dp8 PF001356.hmm.gz
pigz -dp8 PF011598.hmm.gz
pigz -dp8 PF036910.hmm.gz
pigz -dp8 PF046360.hmm.gz
```

#Link files for Nvectensis and Aaurita in ips directory

Hide

```
cd /scratch/molevo/janicek/Lab2/results/ips/nvectensis
ln -s /scratch/molevo/jmontenegro/nvectensis/results/annotation/tcs2_internal/tcs2.in
ternal.pep.fa

cd /scratch/molevo/janicek/Lab2/results/ips/aaurita/
ln -s /scratch/molevo/jmontenegro/alison/aaurita/results/combine/aaur2.dedup.pep.fa
```

Extracting proteins from Aaurtia and Nvectensis data. Samtools (version 1.17)

Hide

```
module load samtools
cd /scratch/molevo/janicek/Lab2/results/ips/nvectensis/
list=(`cat IPR001356.list`)
samtools faidx tcs2.internal.pep.fa ${list[@]} > nv2_IPR001356.fa

list=(`cat IPR011598.list`)
samtools faidx tcs2.internal.pep.fa ${list[@]} > nv2_IPR011598.fa

list=(`cat IPR036910.list`)
samtools faidx tcs2.internal.pep.fa ${list[@]} > nv2_IPR036910.fa

list=(`cat IPR046360.list`)
samtools faidx tcs2.internal.pep.fa ${list[@]} > nv2_IPR046360.fa

cd /scratch/molevo/janicek/Lab2/results/ips/aaurita/
list=(`cat IPR036910.list`)
samtools faidx aaur2.dedup.pep.fa ${list[@]} > aaur2_IPR036910.fa

list=(`cat IPR046360.list`)
samtools faidx aaur2.dedup.pep.fa ${list[@]} > aaur2_IPR046360.fa
```

#IPR001356 and IPR011598 from Aaurita These two IPRs had suffixes that had been removed from the original names. Therefore the suffixes had to be also removed so that the proteins could be extracted.

Hide

```
module load samtools

sed -e 's/\.p[0-9]\+//' IPR001356.list > tmp
mv tmp IPR001356.list
list=(`cat IPR001356.list`)
samtools faidx aaur2.dedup.pep.fa ${list[@]} > aaur2_IPR001356.fa

sed -e 's/\.p[0-9]\+//' IPR011598.list > tmp
mv tmp IPR011598.list
list=(`cat IPR011598.list`)
samtools faidx aaur2.dedup.pep.fa ${list[@]} > aaur2_IPR011598.fa
```

CD-hit is used to de-duplicate the proteins from the multifasta files.

A CD-hit Slurm script was made which clusters the protein sequences while reducing any duplications within them. CD-hit (version 4.8.1)

Hide

```
#!/bin/bash

#SBATCH --job-name=cdhit
#SBATCH --nodes=1
#SBATCH --cpus-per-task=4
#SBATCH --mem=1GB
#SBATCH --time=5:0:0
#SBATCH --partition=basic
#SBATCH --output=/scratch/molevo/janicek/Lab2/logs/cdhit-%j_%A.log
#SBATCH --error=/scratch/molevo/janicek/Lab2/logs/cdhit-%j_%A.err
#SBATCH --mail-type=ALL
#SBATCH --mail-user=a12110422@unet.univie.ac.at

###ENVIRONMENT
module load cdhit
module list

###CONSTANTS
wd="/scratch/molevo/janicek/Lab2"
prots=( ${wd}/data/multiple*/*.fasta.*.gz )
res="${wd}/results"
od="${res}/dedup"

###VARIABLES
prot=${prots[$SLURM_ARRAY_TASK_ID]}
base=`basename ${prot}`
ipr="${base%%-*}"
specie="${base##*fasta.}"
specie="${specie%.gz}"
out="${ipr}_${specie}.dedup.fa"

###EXECUTION
echo "Started at `date`"

echo "mkdir -p ${od}"
mkdir -p ${od}

echo "cd-hit-est -i ${prot} -aL 0.1 -aS 1 -T 4 -d 50 -c 1 -o ${od}/${out}"
cd-hit-est -i ${prot} -aL 0.1 -aS 1 -T 4 -d 50 -c 1 -o ${od}/${out}

echo "Finished at `date'"
```

Hmmer(version 3.3.2) is used to find motifs within the protein sequences.

Merges the PFAM file and the deduplicated fasta files into a single file used to generate a table of coordinates.

Hide

```
module load hmmer

hmmsearch --domtblout 001356dt --cpu 4 ../data/PF001356.hmm.gz dedup/IPR001356.allSpe
cies.fasta > PF001356.hmmer.out
fgrep -v "#" 001356dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3>1{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 001356dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 001356dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3==1' >
IPR001356.tblout.coords

hmmsearch --domtblout 011598dt --cpu 4 ../data/PF011598.hmm.gz dedup/IPR0011598.allSp
ecies.fasta > PF011598.hmmer.out
fgrep -v "#" 011598dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3>1{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 011598dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 011598dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3==1' >
IPR011598.tblout.coords

hmmsearch --domtblout 036910dt --cpu 4 ../data/PF036910.hmm.gz dedup/IPR03910.allSpec
ies.fasta > PF036910.hmmer.out
fgrep -v "#" 036910dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3>1{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 036910dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 036910dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3==1' >
IPR036910.tblout.coords

hmmsearch --domtblout 046360dt --cpu 4 ../data/PF046360.hmm.gz dedup/IPR046360.allSpe
cies.fasta > PF046360.hmmer.out
fgrep -v "#" 046360dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3>1{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 046360dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '{prin
t$1}' | sort | uniq | wc -l
fgrep -v "#" 046360dt | awk '{print $1"\t"$10"\t"$11"\t"$20"\t"$21}' | awk '$3==1' >
IPR046360.tblout.coords

mkdir hmmer
 mv *out *coords *t hmmer/ #to clean up results folder and find files easier
```

Samtools(version 1.17) faidx creates a list from the de-duplicated proteins.

Using the de-duplicated fasta files, a list is created that includes the protein family and all seven species. This list generates a table of coordinates. The table of coordinates for each protein family only includes the relevant information of name, hmm to and from, align to and from, and env to and from

Hide

```
#module load samtools (make sure loaded)

list=( `awk '{print $1":"$4"-"$5}' IPR001356.tblout.coords` ) #creates a table of coo
rdinates for each protein family
samtools faidx ../dedup/IPR001356.allSpecies.fasta ${list[@]} > PF001356.oneDomain.al
lSpecies.fa #moves the list into a multifasta file containing the IPR and PFAM fasta
together

list=( `awk '{print $1":"$4"-"$5}' IPR011598.tblout.coords` )
samtools faidx ../dedup/IPR0011598.allSpecies.fasta ${list[@]} > PF011598.oneDomain.a
llSpecies.fa

list=( `awk '{print $1":"$4"-"$5}' IPR036910.tblout.coords` )
samtools faidx ../dedup/IPR03910.allSpecies.fasta ${list[@]} > PF036910.oneDomain.all
Species.fa

list=( `awk '{print $1":"$4"-"$5}' IPR046360.tblout.coords` )
samtools faidx ../dedup/IPR046360.allSpecies.fasta ${list[@]} > PF046360.oneDomain.al
lSpecies.fa

# vim PF001356.oneDomain.allSpecies.fa #can be used at anytime after the file is crea
ted to visualize the contents, just change the PF# to the correspoinding protein fami
ly number
```

Samtools faidx will then be used to extract these coordinates from the motifs

This creates a position to focus on for the alignment of our transcription factors in each of the species. Ultimately the product of this step results in the protein sizes that are selected to run the alignment on.

Hide

```
list=( `tail -n +2 PF001356.size.select.tsv | cut -f1` ) #creates a list from the tsv
file
samtools faidx PF001356.oneDomain.allSpecies.fa ${list[@]} > PF001356.oneDomain.allSp
ecies.sel.fa #creates a fasta file from the selected list
list=( `tail -n +2 PF011598.size.select.tsv | cut -f1` )
samtools faidx PF011598.oneDomain.allSpecies.fa ${list[@]} > PF011598.oneDomain.allSp
ecies.sel.fa
list=( `tail -n +2 PF036910.size.select.tsv | cut -f1` )
samtools faidx PF036910.oneDomain.allSpecies.fa ${list[@]} > PF036910.oneDomain.allSp
ecies.sel.fa
list=( `tail -n +2 PF046360.size.select.tsv | cut -f1` )
samtools faidx PF046360.oneDomain.allSpecies.fa ${list[@]} > PF046360.oneDomain.allSp
ecies.sel.fa
```

Create plots for visualization of protein sizes using R studio(version 2023.03.0 Build 386)

Plots for each of the four IPR numbers were generated to visualize the protein sizes. The first plot will show all the proteins and show the ranges in which the sizes of the proteins fall. This allows us to choose the area of optimum size ranges. The second plot will show the proteins of interest after the size range has been adjusted.

Hide

```
library(ggplot2)
getwd() #to make sure in the correct directory of "/scratch/molevo/janicek/Lab2/resul
ts/hmmer"
#setwd() #use if not in the above directory

sizes001356<-read_tsv("PF001356.oneDomain.allSpecies.fa.fai", col_names=F) #shows how
many rows and columns are in the PFAM-IPR for all species
ggplot(sizes001356) + geom_density(aes(x=X2, y=after_stat(ndensity))) + theme_bw() #p
lots graph
ggplot(sizes001356, aes(x=X2)) +
  geom_density(aes(y=after_stat(ndensity))) +
  theme_bw() +
  xlim(40, 70) #re-plots the graph to show only the proteins that fall within the pea
ks of interest


sizes011598<-read_tsv("PF011598.oneDomain.allSpecies.fa.fai", col_names=F)
ggplot(sizes011598) + geom_density(aes(x=X2, y=after_stat(ndensity))) + theme_bw()
ggplot(sizes011598, aes(x=X2)) +
  geom_density(aes(y=after_stat(ndensity))) +
  theme_bw() +
  xlim(40, 70)

sizes036910<-read_tsv("PF036910.oneDomain.allSpecies.fa.fai", col_names=F)
ggplot(sizes036910) + geom_density(aes(x=X2, y=after_stat(ndensity))) + theme_bw()
ggplot(sizes036910, aes(x=X2)) +
  geom_density(aes(y=after_stat(ndensity))) +
  theme_bw() +
  xlim(50, 80)

sizes046360<-read_tsv("PF046360.oneDomain.allSpecies.fa.fai", col_names=F)
ggplot(sizes046360) + geom_density(aes(x=X2, y=after_stat(ndensity))) + theme_bw()
ggplot(sizes046360, aes(x=X2)) +
  geom_density(aes(y=after_stat(ndensity))) +
  theme_bw() +
  xlim(100, 210)
```

Alignment of protein sequences using Clustalomega(version 1.2.4)

This tool aligns the extracted coordinates for the protein sequences of interest which can show meaningful divergence. Alignment of multiple sequences allows for visualization of the divergence of between species.

Hide

```
module load clustalomega

clustalo -i PF001356.oneDomain.allSpecies.sel.fa -o PF001356.oneDomain.allSpecies.aln
--hmm-in ../../data/PF001356.hmm --thread 8
#vim PF001356.oneDomain.allSpecies.aln #this command allows visualization of the alig
ned sequence to know it worked, can be skipped in the command lines and run at the en
d

clustalo -i PF011598.oneDomain.allSpecies.sel.fa -o PF011598.oneDomain.allSpecies.aln
--hmm-in ../../data/PF011598.hmm --thread 8
#vim PF011598.oneDomain.allSpecies.aln

clustalo -i PF036910.oneDomain.allSpecies.sel.fa -o PF036910.oneDomain.allSpecies.aln
--hmm-in ../../data/PF036910.hmm --thread 8
#vim PF036910.oneDomain.allSpecies.aln

clustalo -i PF046360.oneDomain.allSpecies.sel.fa -o PF046360.oneDomain.allSpecies.aln
--hmm-in ../../data/PF046360.hmm --thread 8
#vim PF046360.oneDomain.allSpecies.aln

mkdir clustalo
mv hmmer/*aln clustalo/
```

Iqtree(version 2.2.2.4) for builing trees

In bash, iqtree will reconstruct evolutionary trees from the alignment data that was performed in clustalomega. This uses a bootstrapping technique, running simulations on 10,000 trees per IPR, to give the most likely cases for the TFs in the trees. The script below uses the aligned proteins of each of the four TFs in each of the seven species. A slurm script is necessary due to the long run times needed to accomplish this task.

Hide

```
#!/bin/bash
#SBATCH --job-name=iqtree
#SBATCH --nodes=1
#SBATCH --cpus-per-task=4
#SBATCH --mem=1G
#SBATCH --time=24:00:00
#SBATCH --partition=basic
#SBATCH --output=/scratch/molevo/janicek/Lab2/logs/iqtree-%j_%A.log
#SBATCH --error=/scratch/molevo/janicek/Lab2/logs/iqtree-%j_%A.err
#SBATCH --mail-type=ALL
#SBATCH --mail-user=a12110422@unet.univie.ac.at

###ENVIRONMENT
module load iqtree
module list

###CONSTANTS
wd="/scratch/molevo/janicek/Lab2"
genes=( ${wd}/results/clustalo/*.allSpecies.fasta ) #these are the proteins that were
selected per IPR#
res="${wd}/results"
od="${res}/iqtree"

###VARIABLES
gene=${genes[$SLURM_ARRAY_TASK_ID]}
base=`basename ${gene}`
BOOTSTRAP=10000
out="${base%%.*}.iqtree"

###EXECUTION
echo "Started at `date`"

echo "mkdir -p ${od}"
mkdir -p ${od}

echo "cd $od"
cd $od

echo "iqtree2 -s $gene --seqtype AA -m MFP -B $BOOTSTRAP --threads-max 4 --prefix ${o
ut}"
iqtree2 -s $gene --seqtype AA -m MFP -B $BOOTSTRAP --threads-max 4 --prefix ${out}

echo "Finished at `date`"
```

Create metadata tables using tsv files

Hide

```
cd /scratch/molevo/janicek/Lab2/results/clustalo

module load samtools

samtools faidx PF001356.oneDomain.allSpecies.fasta #creates a fasta.fai file to be us
ed in metadata table construction, do for each IPR.fast file
samtools faidx PF011598.oneDomain.allSpecies.fasta
samtools faidx PF036910.oneDomain.allSpecies.fasta
samtools faidx PF046360.oneDomain.allSpecies.fasta
```

#Using R studio, the metadata tables for five of the seven species (celegans, drosophila, human, mizuhopectensis, and amphimedon queensland) are built and merged together to complete the table for each TF superfamily.

Hide

```
setwd("/scratch/molevo/janicek/Lab2/results/clustalo") #had to manually setwd in cons
ole had issue changing directory in the chunk
library(tidyverse)

alig<-read_tsv("PF001356.oneDomain.allSpecies.fasta.fai", col_names=F)
ce<-read_tsv("../../data/celegans/IPR001356-20.07.46.99.tsv.celegans") #gives rows an
d columns for the species within the tsv file
alig2 <- alig %>% separate(X1, c("prefix", "Entry", "EntryName"), sep="\\|") #creates
separate headings for each column, makes it easier to read and identify in tables lat
er
left_join(alig2, ce, by="Entry") #merges the alig table and species tables - will cha
nge for each species added

dm<-read_tsv("../../data/drosophila/IPR001356-20.07.07.77.tsv.drosophila")
metaTab<-bind_rows(ce,dm) #merges the species tables together and is cumulative - cha
nges as more species added
#left_join(alig2, metaTab, by="Entry") #can be left until the last species is added

hs<-read_tsv("../../data/human/IPR001356-20.08.08.77.tsv.human")
metaTab<-bind_rows(metaTab, hs)
#left_join(alig2, metaTab, by="Entry")

mi<-read_tsv("../../data/mizuhopectenyessoensis/IPR001356-20.06.23.98.tsv.mizuhopecte
nyessoensis")
metaTab<-bind_rows(metaTab, mi)
#left_join(alig2, metaTab, by="Entry")

am<-read_tsv("../../data/amphimedonqueensland/IPR001356-20.04.50.40.tsv.amphimedonque
ensland")
am[, 5]<-as.character(am[, 5]) #needed to change all entries to characters - differen
t organization from uniprot download for this particular species as compared to the o
thers
metaTab<-bind_rows(metaTab, am)
left_join(alig2, metaTab, by="Entry")

mergedTab<-left_join(alig2, metaTab, by="Entry") #merges all tables to 1 metadata tab
le per IPR
write_tsv(mergedTab, file="IPR001356.metadata.tsv") #creates metadata.tsv file and sa
ves to directory

alig<-read_tsv("PF011598.oneDomain.allSpecies.fasta.fai", col_names=F)
ce<-read_tsv("../../data/celegans/IPR011598-20.20.23.46.tsv.celegans")
alig2 <- alig %>% separate(X1, c("prefix", "Entry", "EntryName"), sep="\\|")
left_join(alig2, ce, by="Entry")

dm<-read_tsv("../../data/drosophila/IPR011598-20.21.13.12.tsv.drosophila")
metaTab<-bind_rows(ce,dm)

hs<-read_tsv("../../data/human/IPR011598-20.19.14.08.tsv.human")
metaTab<-bind_rows(metaTab, hs)
#left_join(alig2, metaTab, by="Entry")
```

```
mi<-read_tsv("../../data/mizuhopectenyessoensis/IPR011598-20.24.26.17.tsv.mizuhopecte
nyessoensis")
metaTab<-bind_rows(metaTab, mi)

am<-read_tsv("../../data/amphimedonqueensland/IPR011598-20.23.04.26.tsv.amphimedonque
ensland")
am[, 5]<-as.character(am[, 5])
metaTab<-bind_rows(metaTab, am)
left_join(alig2, metaTab, by="Entry")

mergedTab<-left_join(alig2, metaTab, by="Entry")
write_tsv(mergedTab, file="IPR011598.metadata.tsv")

alig<-read_tsv("PF036910.oneDomain.allSpecies.fasta.fai", col_names=F)
ce<-read_tsv("../../data/celegans/IPR036910-20.26.36.03.tsv.celegans")
alig2 <- alig %>% separate(X1, c("prefix", "Entry", "EntryName"), sep="\\|")
left_join(alig2, ce, by="Entry")

dm<-read_tsv("../../data/drosophila/IPR036910-20.27.29.02.tsv.drosophila")
metaTab<-bind_rows(ce,dm)

hs<-read_tsv("../../data/human/IPR036910-20.25.43.76.tsv.human")
metaTab<-bind_rows(metaTab, hs)
#left_join(alig2, metaTab, by="Entry")

mi<-read_tsv("../../data/mizuhopectenyessoensis/IPR036910-20.28.31.03.tsv.mizuhopecte
nyessoensis")
metaTab<-bind_rows(metaTab, mi)
#left_join(alig2, metaTab, by="Entry")

am<-read_tsv("../../data/amphimedonqueensland/IPR036910-20.29.44.77.tsv.amphimedonque
ensland")
am[, 5]<-as.character(am[, 5])
metaTab<-bind_rows(metaTab, am)
left_join(alig2, metaTab, by="Entry")

mergedTab<-left_join(alig2, metaTab, by="Entry")
write_tsv(mergedTab, file="IPR036910.metadata.tsv")

alig<-read_tsv("PF046360.oneDomain.allSpecies.fasta.fai", col_names=F)

ce<-read_tsv("../../data/celegans/IPR046360-20.31.30.54.tsv.celegans")
alig2 <- alig %>% separate(X1, c("prefix", "Entry", "EntryName"), sep="\\|")
left_join(alig2, ce, by="Entry")

dm<-read_tsv("../../data/drosophila/IPR046360-20.32.28.01.tsv.drosophila")
metaTab<-bind_rows(ce,dm)

hs<-read_tsv("../../data/human/IPR046360-20.30.41.78.tsv.human")
metaTab<-bind_rows(metaTab, hs)
```

```
#left_join(alig2, metaTab, by="Entry")

mi<-read_tsv("../../data/mizuhopectenyessoensis/IPR046360-20.34.30.54.tsv.mizuhopecte
nyessoensis")
metaTab<-bind_rows(metaTab, mi)
#left_join(alig2, metaTab, by="Entry")

am<-read_tsv("../../data/amphimedonqueensland/IPR046360-20.33.24.03.tsv.amphimedonque
ensland")
am[, 5]<-as.character(am[, 5])
metaTab<-bind_rows(metaTab, am)
left_join(alig2, metaTab, by="Entry")

mergedTab<-left_join(alig2, metaTab, by="Entry")
write_tsv(mergedTab, file="IPR046360.metadata.tsv")
```

Manual compilation of the metadata tables for Aurelia and Nvectensis.

Hide

```
module load samtools

samtools faidx nv2_IPR001356.fa #creates a .fa.fai file to begin the metadata table u
sing the tsv files per IPR
cut -f 1 nv2_IPR001356.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 nv2_IPR001356.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR001356 | cut -f 1,6,12
-14 | head #takes the pertinent heading information to add to the metadata table
cut -f 1 nv2_IPR001356.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR001356 | cut -f 1,6,12
-14 > nv2_IPR001356.metadata.tsv #creates the metadata table to add
cut -f 1 nv2_IPR001356.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR001356 | cut -f 1,12-1
4 | sort | uniq > nv2_IPR001356.metadata.tsv #takes out the duplicates that were in t
he data due to imputting changes
head nv2_IPR001356.metadata.tsv #checks the data is there

samtools faidx nv2_IPR011598.fa
cut -f 1 nv2_IPR011598.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 nv2_IPR011598.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR011598 | cut -f 1,6,12
-14 | head
cut -f 1 nv2_IPR011598.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR011598 | cut -f 1,12-1
4 | sort | uniq > nv2_IPR011598.metadata.tsv
#head nv2_IPR011598.metadata.tsv

samtools faidx nv2_IPR036910.fa
cut -f 1 nv2_IPR036910.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 nv2_IPR036910.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR036910| cut -f 1,6,12-
14 | head
cut -f 1 nv2_IPR036910.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR036910 | cut -f 1,12-1
4 | sort | uniq > nv2_IPR036910.metadata.tsv
#head nv2_IPR036910.metadata.tsv

samtools faidx nv2_IPR046360.fa
cut -f 1 nv2_IPR046360.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 nv2_IPR046360.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR046360| cut -f 1,6,12-
14 | head
cut -f 1 nv2_IPR046360.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR046360 | cut -f 1,12-1
4 | sort | uniq > nv2_IPR046360.metadata.tsv
#head nv2_IPR046360.metadata.tsv

samtools faidx aaur2_IPR001356.fa
cut -f 1 aaur2_IPR001356.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 aaur2_IPR001356.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR001356| cut -f 1,6,1
2-14 | head
cut -f 1 aaur2_IPR001356.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR001356 | cut -f 1,12
-14 | sort | uniq > aaur2_IPR001356.metadata.tsv
#head aaur2_IPR001356.metadata.tsv

samtools faidx aaur2_IPR011598.fa
cut -f 1 aaur2_IPR011598.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 aaur2_IPR011598.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR011598 | cut -f 1,12
-14 | sort | uniq > aaur2_IPR011598.metadata.tsv
#head aaur2_IPR011598.metadata.tsv
```

```
samtools faidx aaur2_IPR036910.fa
cut -f 1 aaur2_IPR036910.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 aaur2_IPR036910.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR036910 | cut -f 1,12
-14 | sort | uniq > aaur2_IPR036910.metadata.tsv
#head aaur2_IPR036910.metadata.tsv

samtools faidx aaur2_IPR046360.fa
cut -f 1 aaur2_IPR046360.fa.fai | fgrep -wf - *.tsv | head
cut -f 1 aaur2_IPR046360.fa.fai | fgrep -hwf - *.tsv |  fgrep IPR046360 | cut -f 1,12
-14 | sort | uniq > aaur2_IPR046360.metadata.tsv
#head aaur2_IPR046360.metadata.tsv
```

Merging Aurelia and Nvectensis metadata tables to the completed IPR metadata tables

Using R studio, Nvectensis and Aurelia metadata tables are combined with the metadata tables of the other five species creating a complete metadata table consisting of all seven species

Hide

```
#setwd("/scratch/molevo/janicek/Lab2/results/ips/aaurita") #make sure to either be in
this directory or be able to get into it

###IPR001356
alig<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR001356.fa.f
ai", col_names=F)
aaur2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR001356.met
adata.tsv", col_names=F)
unique(aaur2$X3) #makes sure you only have 1 unique IPR and in the correct IPR
meta001356<-read_tsv("/scratch/molevo/janicek/Lab2/results/clustalo/IPR001356.metadat
a.tsv")
meta001356$Entry<-ifelse(is.na(meta001356$Entry), gsub(":.+$", "", meta001356$prefi
x), meta001356$Entry) #replaces multiple columns
meta001356 #visualizes changes; do not have to keep repeated this step
meta001356$EntryName<-ifelse(is.na(meta001356$EntryName), meta001356$prefix, meta0013
56$EntryName) #combines entries
#meta036910 #visualizes changes
names(meta001356) #shows column names
meta001356<-meta001356[, c(2,3,8,10:13)] #removes duplicate columns
names(meta001356)[c(4,5)]<-c("ProteinName", "GeneName") #removes spaces
#meta001356 #visulaizes chnanges
aaur2 #makes sure the aaurita species is there and of the same IPR and superfamily
alig #checks that you are working with aaur2 data
alig<-alig[, c(1,2)] #removes unwanted data columns; keeps the important ones
names(alig)<-c("Entry", "Length") #changes names of columns
alig #verifies changes
meta001356 %>% left_join(alig, by="Entry") %>% mutate(Length=coalesce(Length.y, Lengt
h.x)) %>% dplyr::select(-Length.x, -Length.y) #joins tables and removes duplicate len
gth columns
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR001356.fa.f
ai", col_names=F) #gets data for nvectensis to add to aaurita and metadata tables
nv2<-nv2[, c(1,2)] #keeps only certain columns specified
names(nv2)<-c("Entry", "Length") #changes the name of X-variables
nv2 #verifies changes
meta001356<-meta001356 %>% left_join(nv2, by="Entry") %>% mutate(Length=coalesce(Leng
th.y, Length.x)) %>% dplyr::select(-Length.x, -Length.y) #joins tables
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR001356.meta
data.tsv", col_names=F)
nv2 #makes sure correct species, IPR, and all the same superfamily
unique(nv2$X3) #makes sure only 1 type of superfamily
unique(aaur2$X3) #verifies the aaur2 and nv2 are the same superfamily
meta001356 #shows table and what columns still show NA to add data to
meta001356$ProteinName<-ifelse(grepl("NV2", meta001356$Entry), "Homeobox domain", met
a001356$ProteinName) #adds info for the superfamily to the protein name for nvectensi
s
meta001356$ProteinName<-ifelse(grepl("AAUR2", meta001356$Entry), "Homeobox domain", m
eta001356$ProteinName) #adds info for the superfamily to the protein name for aaurita
meta001356 #visulaizes changes
write_tsv(meta001356, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR001356.met
adata.complete.tsv") #copies the completed tsv file to the directory
```

```
###IPR011598
alig<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR011598.fa.f
ai", col_names=F)
aaur2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR011598.met
adata.tsv", col_names=F)
unique(aaur2$X3)
meta011598<-read_tsv("/scratch/molevo/janicek/Lab2/results/clustalo/IPR011598.metadat
a.tsv")
meta011598$Entry<-ifelse(is.na(meta011598$Entry), gsub(":.+$", "", meta011598$prefi
x), meta011598$Entry)
#meta011598
meta011598$EntryName<-ifelse(is.na(meta011598$EntryName), meta011598$prefix, meta0115
98$EntryName)
#meta011598
names(meta011598)
meta011598<-meta011598[, c(2,3,8,10:13)]
names(meta011598)[c(4,5)]<-c("ProteinName", "GeneName")
#meta011598
aaur2
alig
alig<-alig[, c(1,2)]
names(alig)<-c("Entry", "Length")
alig
meta011598 %>% left_join(alig, by="Entry") %>% mutate(Length=coalesce(Length.y, Lengt
h.x)) %>% dplyr::select(-Length.x, -Length.y)
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR011598.fa.f
ai", col_names=F)
nv2<-nv2[, c(1,2)]
names(nv2)<-c("Entry", "Length")
nv2
meta011598<-meta011598 %>% left_join(nv2, by="Entry") %>% mutate(Length=coalesce(Leng
th.y, Length.x)) %>% dplyr::select(-Length.x, -Length.y)
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR011598.meta
data.tsv", col_names=F)
nv2
unique(nv2$X3)
unique(aaur2$X3)
meta011598
meta011598$ProteinName<-ifelse(grepl("NV2", meta011598$Entry), "Myc-type, basic helix
-loop-helix (bHLH) domain", meta011598$ProteinName)
meta011598$ProteinName<-ifelse(grepl("AAUR2", meta011598$Entry), "Myc-type, basic hel
ix-loop-helix (bHLH) domain", meta011598$ProteinName)
meta011598
write_tsv(meta011598, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR011598.met
adata.complete.tsv")

###IPR036910
alig<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR036910.fa.f
ai", col_names=F)
aaur2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR036910.met
adata.tsv", col_names=F)
```

```
unique(aaur2$X3)
meta036910<-read_tsv("/scratch/molevo/janicek/Lab2/results/clustalo/IPR036910.metadat
a.tsv")
meta036910$Entry<-ifelse(is.na(meta036910$Entry), gsub(":.+$", "", meta036910$prefi
x), meta036910$Entry)
#meta036910
meta036910$EntryName<-ifelse(is.na(meta036910$EntryName), meta036910$prefix, meta0369
10$EntryName)
#meta036910
names(meta036910)
meta036910<-meta036910[, c(2,3,8,10:13)]
names(meta036910)[c(4,5)]<-c("ProteinName", "GeneName")
#meta036910
aaur2
alig
alig<-alig[, c(1,2)]
names(alig)<-c("Entry", "Length")
alig
meta036910 %>% left_join(alig, by="Entry") %>% mutate(Length=coalesce(Length.y, Lengt
h.x)) %>% dplyr::select(-Length.x, -Length.y)
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR036910.fa.f
ai", col_names=F)
nv2<-nv2[, c(1,2)]
names(nv2)<-c("Entry", "Length")
nv2
meta036910<-meta036910 %>% left_join(nv2, by="Entry") %>% mutate(Length=coalesce(Leng
th.y, Length.x)) %>% dplyr::select(-Length.x, -Length.y)
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR036910.meta
data.tsv", col_names=F)
nv2
unique(nv2$X3)
unique(aaur2$X3)
meta036910
meta036910$ProteinName<-ifelse(grepl("NV2", meta036910$Entry), "High mobility group
(HMG) box domain", meta036910$ProteinName)
meta036910$ProteinName<-ifelse(grepl("AAUR2", meta036910$Entry), "High mobility group
(HMG) box domain", meta036910$ProteinName)
meta036910
write_tsv(meta036910, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR036910.met
adata.complete.tsv")

###IPR046360
alig<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR046360.fa.f
ai", col_names=F)
aaur2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/aaurita/aaur2_IPR046360.met
adata.tsv", col_names=F)
unique(aaur2$X3)
meta046360<-read_tsv("/scratch/molevo/janicek/Lab2/results/clustalo/IPR046360.metadat
a.tsv")
meta046360$Entry<-ifelse(is.na(meta046360$Entry), gsub(":.+$", "", meta046360$prefi
x), meta046360$Entry)
```

```
#meta046360
meta046360$EntryName<-ifelse(is.na(meta046360$EntryName), meta046360$prefix, meta0463
60$EntryName)
#meta046360
names(meta046360)
meta046360<-meta046360[, c(2,3,8,10:13)]
names(meta046360)[c(4,5)]<-c("ProteinName", "GeneName")
#meta046360
aaur2
alig
alig<-alig[, c(1,2)]
names(alig)<-c("Entry", "Length")
alig
meta046360 %>% left_join(alig, by="Entry") %>% mutate(Length=coalesce(Length.y, Lengt
h.x)) %>% dplyr::select(-Length.x, -Length.y)
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR046360.fa.f
ai", col_names=F)
nv2<-nv2[, c(1,2)]
names(nv2)<-c("Entry", "Length")
nv2
meta046360<-meta046360 %>% left_join(nv2, by="Entry") %>% mutate(Length=coalesce(Leng
th.y, Length.x)) %>% dplyr::select(-Length.x, -Length.y)
nv2<-read_tsv("/scratch/molevo/janicek/Lab2/results/ips/nvectensis/nv2_IPR046360.meta
data.tsv", col_names=F)
nv2
unique(nv2$X3)
unique(aaur2$X3)
meta046360
meta046360$ProteinName<-ifelse(grepl("NV2", meta046360$Entry), "T-box transcription f
actor (Tbox) DNA-binding domain", meta046360$ProteinName)
meta046360$ProteinName<-ifelse(grepl("AAUR2", meta046360$Entry), "T-box transcription
factor (Tbox) DNA-binding domain", meta046360$ProteinName)
meta046360
write_tsv(meta046360, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR046360.met
adata.complete.tsv")
```

Building phylogenetic trees using ggplot in Rstudio.

This step uses the iqtree data and the completed metadata tsv files to build the trees for each IPR

Hide

```
library(tidyverse)
library(treeio)
library(tidytree)
library(ggtree)
library(ggsci)
library(ggstar)
library(ggplot2)
library(ggstance)
library(ape)
BiocManager::install("ggtreeExtra")
library(ggtreeExtra)
library(dbplyr)


#location of data
iqTreeFile001356<-"/scratch/molevo/janicek/Lab2/results/iqtree/PF001356.iqtree.contre
e"
iqTreeFile011598<-"/scratch/molevo/janicek/Lab2/results/iqtree/PF011598.iqtree.contre
e"
iqTreeFile036910<-"/scratch/molevo/janicek/Lab2/results/iqtree/PF036910.iqtree.contre
e"
iqTreeFile046360<-"/scratch/molevo/janicek/Lab2/results/iqtree/PF046360.iqtree.contre
e"

metadata001356<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR001356.metad
ata.complete.tsv") #loads and reads the tsv file
metadata011598<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR011598.metad
ata.complete.tsv")
metadata036910<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR036910.metad
ata.complete.tsv")
metadata046360<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR046360.metad
ata.complete.tsv")

###Fixes NA fields within the metadata files - should not have to do this again

#Adds entry to GeneName
metadata001356$GeneName<-ifelse(is.na(metadata001356$GeneName), metadata001356$Entry,
metadata001356$GeneName)
metadata011598$GeneName<-ifelse(is.na(metadata011598$GeneName), metadata011598$Entry,
metadata011598$GeneName)
metadata036910$GeneName<-ifelse(is.na(metadata036910$GeneName), metadata036910$Entry,
metadata036910$GeneName)
metadata046360$GeneName<-ifelse(is.na(metadata046360$GeneName), metadata046360$Entry,
metadata046360$GeneName)

#Adds unreveiwed to table
metadata001356$Reviewed<-ifelse(is.na(metadata001356$Reviewed), "unreviewed", metadat
a001356$Reviewed)
metadata011598$Reviewed<-ifelse(is.na(metadata011598$Reviewed), "unreviewed", metadat
a011598$Reviewed)
metadata036910$Reviewed<-ifelse(is.na(metadata036910$Reviewed), "unreviewed", metadat
```

```
a036910$Reviewed)
metadata046360$Reviewed<-ifelse(is.na(metadata046360$Reviewed), "unreviewed", metadat
a046360$Reviewed)

#Adds Organism name to table
metadata001356$Organism<-ifelse(grepl("NV2", metadata001356$Entry), "Nematostella vec
tensis", metadata001356$Organism)
metadata001356 #checks the changes
metadata001356$Organism<-ifelse(grepl("AAUR2", metadata001356$Entry), "Aurelia sp.",
metadata001356$Organism)
metadata001356 #checks the changes
write_tsv(metadata001356, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR00135
6.metadata.complete.tsv") #rewrites the tsv file with added information

metadata011598$Organism<-ifelse(grepl("NV2", metadata011598$Entry), "Nematostella vec
tensis", metadata011598$Organism)
metadata011598$Organism<-ifelse(grepl("AAUR2", metadata011598$Entry), "Aurelia sp.",
metadata011598$Organism) #metadata011598
write_tsv(metadata011598, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR01159
8.metadata.complete.tsv")

metadata036910$Organism<-ifelse(grepl("NV2", metadata036910$Entry), "Nematostella vec
tensis", metadata036910$Organism)
metadata036910$Organism<-ifelse(grepl("AAUR2", metadata036910$Entry), "Aurelia sp.",
metadata036910$Organism)
#metadata036910
write_tsv(metadata036910, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR03691
0.metadata.complete.tsv")

metadata046360$Organism<-ifelse(grepl("NV2", metadata046360$Entry), "Nematostella vec
tensis", metadata046360$Organism)
metadata046360$Organism<-ifelse(grepl("AAUR2", metadata046360$Entry), "Aurelia sp.",
metadata046360$Organism)
#metadata046360
write_tsv(metadata046360, file="/scratch/molevo/janicek/Lab2/results/iqtree/IPR04636
0.metadata.complete.tsv")

#to match EntryName in tsv file to the iqtree - takes out :'s and replaces with _'s
metadata001356$Entry2<-gsub(":","_", metadata001356$EntryName)
metadata011598$Entry2<-gsub(":","_", metadata011598$EntryName)
metadata036910$Entry2<-gsub(":","_", metadata036910$EntryName)
metadata046360$Entry2<-gsub(":","_", metadata046360$EntryName)

#load data
tree001356<-read.tree(iqTreeFile001356)
annot001356<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR001356.metadat
a.complete.tsv")

tree011598<-read.tree(iqTreeFile011598)
annot011598<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR011598.metadat
a.complete.tsv")
```

```
tree036910<-read.tree(iqTreeFile036910)
annot036910<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR036910.metadat
a.complete.tsv")

tree046360<-read.tree(iqTreeFile046360)
annot046360<-read_tsv("/scratch/molevo/janicek/Lab2/results/iqtree/IPR046360.metadat
a.complete.tsv")
```

Fixing and merging data fields within the trees

This is performed to take out spaces and make the names match

Hide

```
#001356
tree001356$tip.label<-gsub("_[0-9]+?.+$", "", gsub("^.+?\\|", "", tree001356$tip.label)) #removed numbers after _, removed everything before the |
write.tree(tree001356, file="PF001356.iqtree.renamed.nwk") #created a new file with the fixed names
newAnnot<-annot001356%>%unite("fullEntry", c(Entry, EntryName), sep="|", remove=TRUE) #merges Entry and EntryName into one column called fullEntry
newAnnot$fullEntry<-gsub(":.+$", "", newAnnot$fullEntry) #removes everything after the : in fullEntry column
newAnnot<-newAnnot[, c(1:6)] #removes column 7 which was Entry2
newAnnot$fullEntry<-ifelse(grepl("NV2", newAnnot$fullEntry), gsub("\\|.+$", "", newAnnot$fullEntry), newAnnot$fullEntry) #takes all NV2 and removes anything after the | in fullEntry column
newAnnot$fullEntry<-ifelse(grepl("AAUR2", newAnnot$fullEntry), gsub("\\|.+$", "", newAnnot$fullEntry), newAnnot$fullEntry) #takes all AAUR2 and removes anything after the | in fullEntry column
annot001356$GeneName<-gsub(" .+$", "", newAnnot$GeneName) #removes any spaces in GeneName column
annot001356$GeneName<-gsub("c\\(", "", newAnnot$GeneName) #removes c's from GeneName column
newAnnot$GeneName<-gsub("c\\(", "", newAnnot$GeneName) #removes the c's from the newAnnot GeneName column
newAnnot$GeneName<-gsub(" .+$", "", newAnnot$GeneName) #removes the empty spaces
newAnnot$GeneName<-gsub(",", "", newAnnot$GeneName) #removes the "s from GeneName

#011598
tree011598$tip.label<-gsub("_[0-9]+?.+$", "", gsub("^.+?\\|", "", tree011598$tip.label))
write.tree(tree011598, file="PF011598.iqtree.renamed.nwk")
newAnnot011598<-annot011598%>%unite("fullEntry", c(Entry, EntryName), sep="|", remove=TRUE)
newAnnot011598$fullEntry<-gsub(":.+$", "", newAnnot011598$fullEntry)
newAnnot011598<-newAnnot011598[, c(1:6)]
newAnnot011598$fullEntry<-ifelse(grepl("NV2", newAnnot011598$fullEntry), gsub("\\|.+$", "", newAnnot011598$fullEntry), newAnnot011598$fullEntry)
newAnnot011598$fullEntry<-ifelse(grepl("AAUR2", newAnnot011598$fullEntry), gsub("\\|.+$", "", newAnnot011598$fullEntry), newAnnot011598$fullEntry)
annot011598$GeneName<-gsub(" .+$", "", newAnnot011598$GeneName)
annot011598$GeneName<-gsub("c\\(", "", newAnnot011598$GeneName)
newAnnot011598$GeneName<-gsub("c\\(", "", newAnnot011598$GeneName)
newAnnot011598$GeneName<-gsub(" .+$", "", newAnnot011598$GeneName)
newAnnot011598$GeneName<-gsub(",", "", newAnnot011598$GeneName)

#036910
tree036910$tip.label<-gsub("_[0-9]+?.+$", "", gsub("^.+?\\|", "", tree036910$tip.label))
write.tree(tree036910, file="PF036910.iqtree.renamed.nwk")
newAnnot036910<-annot036910%>%unite("fullEntry", c(Entry, EntryName), sep="|", remove=TRUE)
newAnnot036910$fullEntry<-gsub(":.+$", "", newAnnot036910$fullEntry)
newAnnot036910<-newAnnot036910[, c(1:6)]
```

```
newAnnot036910$fullEntry<-ifelse(grepl("NV2", newAnnot036910$fullEntry), gsub
("\\|.+$", "", newAnnot036910$fullEntry), newAnnot036910$fullEntry)
newAnnot036910$fullEntry<-ifelse(grepl("AAUR2", newAnnot036910$fullEntry), gsub
("\\|.+$", "", newAnnot036910$fullEntry), newAnnot036910$fullEntry)
annot036910$GeneName<-gsub(" .+$", "", newAnnot036910$GeneName)
annot036910$GeneName<-gsub("c\\(", "", newAnnot036910$GeneName)
newAnnot036910$GeneName<-gsub("c\\(", "", newAnnot036910$GeneName)
newAnnot036910$GeneName<-gsub(" .+$", "", newAnnot036910$GeneName)
newAnnot036910$GeneName<-gsub(",", "", newAnnot036910$GeneName)

#046360
tree046360$tip.label<-gsub("_[0-9]+?.+$", "", gsub("^.+?\\|", "", tree046360$tip.labe
l))
write.tree(tree046360, file="PF046360.iqtree.renamed.nwk")
newAnnot046360<-annot046360%>%unite("fullEntry", c(Entry, EntryName), sep="|", remove
=TRUE)
newAnnot046360$fullEntry<-gsub(":.+$", "", newAnnot046360$fullEntry)
newAnnot046360<-newAnnot046360[, c(1:6)]
newAnnot046360$fullEntry<-ifelse(grepl("NV2", newAnnot046360$fullEntry), gsub
("\\|.+$", "", newAnnot046360$fullEntry), newAnnot046360$fullEntry)
newAnnot046360$fullEntry<-ifelse(grepl("AAUR2", newAnnot046360$fullEntry), gsub
("\\|.+$", "", newAnnot046360$fullEntry), newAnnot046360$fullEntry)
annot046360$GeneName<-gsub(" .+$", "", newAnnot046360$GeneName)
annot046360$GeneName<-gsub("c\\(", "", newAnnot046360$GeneName)
newAnnot046360$GeneName<-gsub("c\\(", "", newAnnot046360$GeneName)
newAnnot046360$GeneName<-gsub(" .+$", "", newAnnot046360$GeneName)
newAnnot046360$GeneName<-gsub(",", "", newAnnot046360$GeneName)
```

Distance data from Matrices

Hide

```
#PF001356 matrix
matrix001356 <- read.table("/scratch/molevo/janicek/Lab2/results/iqtree/PF001356.iqtr
ee.mldist", header = FALSE, sep = "", skip=1) #reads the file for the matrix created
in iqtree
matrix001356$V1 <- gsub("^.+\\|", "", gsub("_[0-9]+-[0-9]+$", "", matrix001356$V1))
head(matrix001356) #double checks that the columns are there
human<-grepl("HUMAN", matrix001356$V1) #selects the human genes from the matrix and s
kips the 1st column; creates a True or False table
summary(human) #shows the number of True and False in the matrix for human
nema<-grepl("NV2", matrix001356$V1) #selects the human genes from the matrix and skip
s the 1st column; creates a True or False table
summary(nema) #shows the number of True and False in the matrix for nematostella
aur<-grepl("AAUR2", matrix001356$V1)#selects the human genes from the matrix and skip
s the 1st column; creates a True or False table
summary(aur) #shows the number of True and False in the matrix for aurelia
mat001356<-matrix001356[, c(2:ncol(matrix001356))] #creates a matrix that takes out t
he 1st column
dim(mat001356[nema,human]) #gives the dimensions for nematostella and human genes in
the matrix selected
dim(mat001356[aur,human])
image(as.matrix(mat001356[nema,human])) #shows the heat map of the matrix for nematos
tella and human; lighter colors are closer related
image(as.matrix(mat001356[aur,human]))
apply(mat001356[nema,human], 1, which.min) #gives the min distances for nematostella
to human genes; for each nematostella it gives the closest human gene and where it's
found
matrix001356$V1[nema] #shows the names of all the nematostella genes in matrix
matrix001356$V1[human] #shows the names of all the human genes in matrix
tibble(nv2=matrix001356$V1[nema], human=matrix001356$V1[human][apply(mat001356[nema,h
uman], 1, which.min)]) #creates a table of the genes names from nematostella that cor
respond to the human gene names
nv2_hum<-tibble(nv2=matrix001356$V1[nema], human=matrix001356$V1[human][apply(mat0013
56[nema,human], 1, which.min)]) #creates a file name to use to save the data
write_tsv(nv2_hum, file="PF001356.nv2_human.tsv") #saves it as a tsv file
apply(mat001356[aur,human], 1, which.min) #gives the min distances for aurelia to hum
an genes; for each aurelia it gives the closest human gene and where it's found
aur_hum<-tibble(aurelia=matrix001356$V1[aur], human=matrix001356$V1[human][apply(mat0
01356[aur,human], 1, which.min)]) #creates a file name to use to save the data
aur_hum #shows the table of the genes names from aurelia that correspond to the human
gene names
write_tsv(aur_hum, file="PF001356.aur_human.tsv") #saves it as a tsv file

#PF011598 matrix
matrix011598 <- read.table("/scratch/molevo/janicek/Lab2/results/iqtree/PF011598.iqtr
ee.mldist", header = FALSE, sep = "", skip=1)
matrix011598$V1 <- gsub("^.+\\|", "", gsub("_[0-9]+-[0-9]+$", "", matrix011598$V1))
head(matrix011598)
human<-grepl("HUMAN", matrix011598$V1)
summary(human)
nema<-grepl("NV2", matrix011598$V1)
summary(nema)
```

```
aur<-grepl("AAUR2", matrix011598$V1)
summary(aur)
mat011598<-matrix011598[, c(2:ncol(matrix011598))]
dim(mat011598[nema,human])
dim(mat011598[aur,human])
image(as.matrix(mat011598[nema,human]))
image(as.matrix(mat011598[aur,human]))
apply(mat011598[nema,human], 1, which.min)
#matrix011598$V1[nema] #don't have to run this, but checks that the names are there
#matrix011598$V1[human] #don't have to run this, but checks that the names are there
#matrix011598$V1[aur] #don't have to run this, but checks that the names are there
tibble(nv2=matrix011598$V1[nema], human=matrix011598$V1[human][apply(mat011598[nema,h
uman], 1, which.min)])
nv2_hum<-tibble(nv2=matrix011598$V1[nema], human=matrix011598$V1[human][apply(mat0115
98[nema,human], 1, which.min)])
write_tsv(nv2_hum, file="PF011598.nv2_human.tsv")
apply(mat011598[aur,human], 1, which.min)
aur_hum<-tibble(aurelia=matrix011598$V1[aur], human=matrix011598$V1[human][apply(mat0
11598[aur,human], 1, which.min)])
aur_hum
write_tsv(aur_hum, file="PF011598.aur_human.tsv")

#PF036910 matrix
matrix036910 <- read.table("/scratch/molevo/janicek/Lab2/results/iqtree/PF036910.iqtr
ee.mldist", header = FALSE, sep = "", skip=1)
matrix036910$V1 <- gsub("^.+\\|", "", gsub("_[0-9]+-[0-9]+$", "", matrix036910$V1))
head(matrix036910)
human<-grepl("HUMAN", matrix036910$V1)
summary(human)
nema<-grepl("NV2", matrix036910$V1)
summary(nema)
aur<-grepl("AAUR2", matrix036910$V1)
summary(aur)
mat036910<-matrix036910[, c(2:ncol(matrix036910))]
dim(mat036910[nema,human])
dim(mat036910[aur,human])
image(as.matrix(mat036910[nema,human]))
image(as.matrix(mat036910[aur,human]))
apply(mat036910[nema,human], 1, which.min)
#matrix036910$V1[nema] #don't have to run this, but checks that the names are there
#matrix036910$V1[human] #don't have to run this, but checks that the names are there
#matrix036910$V1[aur] #don't have to run this, but checks that the names are there
tibble(nv2=matrix036910$V1[nema], human=matrix036910$V1[human][apply(mat036910[nema,h
uman], 1, which.min)])
nv2_hum<-tibble(nv2=matrix036910$V1[nema], human=matrix036910$V1[human][apply(mat0369
10[nema,human], 1, which.min)])
write_tsv(nv2_hum, file="PF036910.nv2_human.tsv")
apply(mat036910[aur,human], 1, which.min)
aur_hum<-tibble(aurelia=matrix036910$V1[aur], human=matrix036910$V1[human][apply(mat0
36910[aur,human], 1, which.min)])
aur_hum
```

```
write_tsv(aur_hum, file="PF036910.aur_human.tsv")

#PF046360 matrix
matrix046360 <- read.table("/scratch/molevo/janicek/Lab2/results/iqtree/PF046360.iqtr
ee.mldist", header = FALSE, sep = "", skip=1)
matrix046360$V1 <- gsub("^.+\\|", "", gsub("_[0-9]+-[0-9]+$", "", matrix046360$V1))
head(matrix046360)

#generate geneIds vector
geneIds<-matrix046360$V1
gsub("^.+?\\|", "", geneIds) #removes the "tr" or "sp" before the | in the geneIds
gsub("_[0-9]+-[0-9]+$", "", gsub("^.+?\\|", "", geneIds)) #removes numbers at the end
s of geneIds
geneIds<-gsub("_[0-9]+-[0-9]+$", "", gsub("^.+?\\|", "", geneIds)) #makes the command
s actually write

#select positioins of human, nv and aur proteins in geneIds vector
human<-grepl("HUMAN", geneIds)
summary(human)
nema<-grepl("NV2", geneIds)
summary(nema)
aur<-grepl("AAUR2", geneIds)
summary(aur)

#check that the selections are working
mat046360<-matrix046360[, c(2:ncol(matrix046360))]
dim(mat046360[nema,human])
dim(mat046360[aur,human])
#apply(mat046360[nema,human], 1, which.min)
#geneIds[nema] #don't have to run this, but checks that the names are there
#geneIds[human] #don't have to run this, but checks that the names are there
#geneIds[aur] #don't have to run this, but checks that the names are there
#tibble(nv2=geneIds[nema], human=geneIds[human][apply(mat046360[nema,human], 1, whic
h.min)])
nv2_hum<-tibble(nv2=geneIds[nema], human=geneIds[human][apply(mat046360[nema,human],
1, which.min)])
nv2_hum
write_tsv(nv2_hum, file="PF046360.nv2_human.tsv")
#apply(mat046360[aur,human], 1, which.min)
aur_hum<-tibble(aurelia=geneIds[aur], human=geneIds[human][apply(mat046360[aur,huma
n], 1, which.min)])
aur_hum
write_tsv(aur_hum, file="PF046360.aur_human.tsv")

#can change the code to do for all species so that a heat map can be visualized for a
ny two species together, the above code is for nematostella against human and aurelia
against human.
```

Heat Maps

A distance matrix or similarity matrix is created from iqtree. These matrices show the distance measure

between species for each TF family protein. The lighter the color, the closer the distance. Perhaps suggesting how closely related or orthologous they may be.

Hide

```
library(gridExtra)
library(heatmaply)

#IPR001356 -Homebox (revised Nema/Human)
geneIds <- matrix001356$V1
nema <- grepl("NV2", matrix001356$V1)
human <- grepl("HUMAN", matrix001356$V1)

heatmap1 <- heatmaply(
  normalize(as.matrix(mat001356[nema, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[nema],
  labCol = geneIds[human],
  main = "Homeobox",
  fontsize_row = 5,
  fontsize_col = 5,
  legend_title = "Gene Distance",
  xlab = "",
  ylab = "",
  showlegend = TRUE
)
print(heatmap1)

#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#IPR001356
#geneIds<-matrix001356$V1
#nema<- grepl("NV2", matrix001356$V1)
#human<- grepl("HUMAN", matrix001356$V1)
#heatmap1<-heatmap(as.matrix(mat001356[nema,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[nema], labCol=geneIds[human],
        #main="Homeobox",
       # )

#IPR001356 - Homeobox (revised Aurelia/Human)
geneIds <- matrix001356$V1
aur <- grepl("AUR", matrix001356$V1)
human <- grepl("HUMAN", matrix001356$V1)

heatmap2 <- heatmaply(
  normalize(as.matrix(mat001356[aur, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[aur],
  labCol = geneIds[human],
  main = "Homeobox",
  fontsize_row = 5,
  fontsize_col = 5,
```

```
  legend_title = "Gene Distance",
  xlab = "",
  ylab = "",
  showlegend = TRUE
)
print(heatmap2)

#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#geneIds<-matrix001356$V1
#aur<- grepl("AUR", matrix001356$V1)
#human<- grepl("HUMAN", matrix001356$V1)
#heatmap2<-heatmap(as.matrix(mat001356[aur,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[aur], labCol=geneIds[human],
        #main="Homeobox",
        #)

#IPR011598 - bHLH (revised Nema/Human)
geneIds <- matrix011598$V1
nema <- grepl("NV2", matrix011598$V1)
human <- grepl("HUMAN", matrix011598$V1)

heatmap3 <- heatmaply(
  normalize(as.matrix(mat011598[nema, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[nema],
  labCol = geneIds[human],
  main = "bHLH",
  fontsize_row = 5,
  fontsize_col = 5,
  legend_title = "Gene Distance",
  xlab = "",
  ylab = "",
  showlegend = TRUE
)
print(heatmap3)

#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#IPR011598
#geneIds<-matrix011598$V1
#nema<- grepl("NV2", matrix011598$V1)
#human<- grepl("HUMAN", matrix011598$V1)
#heatmap3<-heatmap(as.matrix(mat011598[nema,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[nema], labCol=geneIds[human],
        #main="bHLH",
```

```
        #)

#IPR011598 - bHLH (revised Aurelia/Human)
geneIds <- matrix011598$V1
aur <- grepl("AUR", matrix011598$V1)
human <- grepl("HUMAN", matrix011598$V1)

heatmap4 <- heatmaply(
  normalize(as.matrix(mat011598[aur, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[aur],
  labCol = geneIds[human],
  main = "bHLH",
  fontsize_row = 5,
  fontsize_col = 5,
  legend_title = "Gene Distance",
  xlab = "",
  ylab = "",
  showlegend = TRUE
)
print(heatmap4)
#IPR011598
#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#geneIds<-matrix011598$V1
#aur<- grepl("AUR", matrix011598$V1)
#human<- grepl("HUMAN", matrix011598$V1)
#heatmap4<-heatmap(as.matrix(mat011598[aur,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[aur], labCol=geneIds[human],
        #main="bHLH",
        #)

#IPR036910 - HMG (revised Nema/Human)
geneIds <- matrix036910$V1
nema <- grepl("NV2", matrix036910$V1)
human <- grepl("HUMAN", matrix036910$V1)

heatmap5 <- heatmaply(
  normalize(as.matrix(mat036910[nema, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[nema],
  labCol = geneIds[human],
  main = "HMG",
  fontsize_row = 8,
  fontsize_col = 5,
  legend_title = "Gene Distance",
  xlab = "",
```

```
  ylab = "",
  showlegend = TRUE
)
print(heatmap5)

#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#IPR036910
#geneIds<-matrix036910$V1
#nema<- grepl("NV2", matrix036910$V1)
#human<- grepl("HUMAN", matrix036910$V1)
#heatmap5<-heatmap(as.matrix(mat036910[nema,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[nema], labCol=geneIds[human],
        #main="HMG",
        #)

#IPR036910 - HMG (revised Aurelia/Human)
geneIds <- matrix036910$V1
aur <- grepl("AUR", matrix036910$V1)
human <- grepl("HUMAN", matrix036910$V1)

heatmap6 <- heatmaply(
  normalize(as.matrix(mat036910[aur, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[aur],
  labCol = geneIds[human],
  main = "HMG",
  fontsize_row = 8,
  fontsize_col = 5,
  legend_title = "Gene Distance",
  xlab = "",
  ylab = "",
  showlegend = TRUE
)
print(heatmap6)

#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#IPR036910
#geneIds<-matrix036910$V1
#aur<- grepl("AUR", matrix036910$V1)
#human<- grepl("HUMAN", matrix036910$V1)
#heatmap6<-heatmap(as.matrix(mat036910[aur,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[aur], labCol=geneIds[human],
        #main="HMG",
        #)
```

```
#IPR046360 - TBox (revised - Nema/Human)
geneIds <- matrix046360$V1
nema <- grepl("NV2", matrix046360$V1)
human <- grepl("HUMAN", matrix046360$V1)

heatmap7 <- heatmaply(
  normalize(as.matrix(mat046360[nema, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[nema],
  labCol = geneIds[human],
  main = "Tbox",
  fontsize_row = 6,
  fontsize_col = 6,
  legend_title = "Gene Distance",
  xlab = "",
  ylab = "",
  showlegend = TRUE
)
print(heatmap7)

#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#IPR046360
#geneIds<-matrix046360$V1
#nema<- grepl("NV2", matrix046360$V1)
#human<- grepl("HUMAN", matrix046360$V1)
#heatmap7<-heatmap(as.matrix(mat046360[nema,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[nema], labCol=geneIds[human],
        #main="Tbox",
        #)

#IPR046360 - TBox (revised - Aurelia/Human)
geneIds <- matrix046360$V1
aur <- grepl("AUR", matrix046360$V1)
human <- grepl("HUMAN", matrix046360$V1)

heatmap8 <- heatmaply(
  normalize(as.matrix(mat046360[aur, human])),
  col = colorRampPalette(c("yellow", "red"))(100),
  scale = "none",
  labRow = geneIds[aur],
  labCol = geneIds[human],
  main = "Tbox",
  fontsize_row = 6,
  fontsize_col = 6,
  legend_title = "Gene Distance",
  xlab = "",
```

```
  ylab = "",
  showlegend = TRUE
)
print(heatmap8)

#heatmap can be seen more clearly using this code, but could not produce a legend to
go with it
#IPR046360
#geneIds<-matrix046360$V1
#aur<- grepl("AUR", matrix046360$V1)
#human<- grepl("HUMAN", matrix046360$V1)
#heatmap8<-heatmap(as.matrix(mat046360[aur,human]),
        #Rowv=NA, Colv=NA,
        #col = colorRampPalette(c("yellow", "red"))(100),
        #scale="none", labRow=geneIds[aur], labCol=geneIds[human],
        #main="Tbox",
        #)

#only heatmaps 7 and 8 are legible in R due to the large number of genes found in the
other TF families.
```

Plot phylogenetic tree data

circular, non-rooted phylogenetic trees for each Transcription factor family for the seven species. These trees are quite large, with the exception of the Tbox tree, and further analysis with software programs that allow for interaction on specific branches is needed to visualize in detail.

Hide

```
library(tidyverse)
library(treeio)
library(tidytree)
library(ggtree)
library(ggsci)
library(ggstar)
library(ggplot2)
library(ggstance)
library(ape)
library(ggtreeExtra)
library(dbplyr)

# Trees for IPR001356 with Human names and highlighting Nematostella and Aurelia
ggtree(tree001356, layout="circular", branch.length="none", size=0.1) +
     geom_fruit(data=newAnnot, geom=geom_star, offset=0.01, size=0.3, starstroke=0, a
es(y=fullEntry, fill=Organism)) +
     geom_fruit(data=newAnnot%>%filter(grepl("Aur",Organism) | grepl("Human", Organis
m) | grepl("Nema", Organism)), geom=geom_text, size=0.3, offset=0.1, aes(y=fullEntry,
label=GeneName)) +
     scale_fill_d3() +
     theme(plot.margin = unit(c(1,1,1,1), "cm"),
           plot.background = element_rect(fill = "white"),
           legend.position = "bottom",
           legend.box = "horizontal",
           legend.margin = margin(t=0, r=0, b=0, l=0),
           legend.spacing.x = unit(0.2, "cm"),
           legend.text=element_text(size=6))


# Trees for IPR011598 with Human names and highlighting Nematostella and Aurelia
ggtree(tree011598, layout="circular", branch.length="none", size=0.1) +
     geom_fruit(data=newAnnot011598, geom=geom_star, offset=0.01, size=0.5, starstrok
e=0.1, aes(y=fullEntry, fill=Organism)) +
     geom_fruit(data=newAnnot011598%>%filter(grepl("Aur",Organism) | grepl("Human", O
rganism) | grepl("Nema", Organism)), geom=geom_text, size=0.35, offset=0.1, aes(y=ful
lEntry, label=GeneName)) +
     scale_fill_d3() +
     theme(plot.margin = unit(c(1,1,1,1), "cm"),
           plot.background = element_rect(fill = "white"),
           legend.position = "bottom",
           legend.box = "horizontal",
           legend.margin = margin(t=0, r=0, b=0, l=0),
           legend.spacing.x = unit(0.2, "cm"),
           legend.text=element_text(size=6))


# Trees for IPR036910 with Human names and highlighting Nematostella and Aurelia
ggtree(tree036910, layout="circular", branch.length="none", size=0.1) +
     geom_fruit(data=newAnnot036910, geom=geom_star, offset=0.01, size=0.5, starstrok
e=0.1, aes(y=fullEntry, fill=Organism)) +
     geom_fruit(data=newAnnot036910%>%filter(grepl("Aur",Organism) | grepl("Human", O
```

```
rganism) | grepl("Nema", Organism)), geom=geom_text, size=0.5, offset=0.1, aes(y=full
Entry, label=GeneName)) +
     scale_fill_d3() +
     theme(plot.margin = unit(c(1,1,1,1), "cm"),
           plot.background = element_rect(fill = "white"),
           legend.position = "bottom",
           legend.box = "horizontal",
           legend.margin = margin(t=0, r=0, b=0, l=0),
           legend.spacing.x = unit(0.2, "cm"),
           legend.text=element_text(size=6))


# Trees for IPR046360 with Human names and highlighting Nematostella and Aurelia
ggtree(tree046360, layout="circular", branch.length="none", size=0.1) +
     geom_fruit(data=newAnnot046360, geom=geom_star, offset=0.01, size=0.5, starstrok
e=0.1, aes(y=fullEntry, fill=Organism)) +
     geom_fruit(data=newAnnot046360%>%filter(grepl("Aur",Organism) | grepl("Human", O
rganism) | grepl("Nema", Organism)), geom=geom_text, size=0.5, offset=0.1, aes(y=full
Entry, label=GeneName)) +
     scale_fill_d3() +
     theme(plot.margin = unit(c(1,1,1,1), "cm"),
           plot.background = element_rect(fill = "white"),
           legend.position = "bottom",
           legend.box = "horizontal",
           legend.margin = margin(t=0, r=0, b=0, l=0),
           legend.spacing.x = unit(0.2, "cm"),
           legend.text=element_text(size=6))


#trees saved as svg file in /scratch/molevo/janicek/Lab2/results/iqtree so they can b
e seen in greater detail
```