Analysis with Orthofinder

## Loading packages

#if installation is need, uncomment the following  
#if (!require("DESeq2")) install.packages("DESeq2"); library(DESeq2)  
#if (!require("apeglm")) install.packages("apeglm"); library(apeglm)  
#if (!require("ggplot2")) install.packages("ggplot2"); library(ggplot2)  
#if (!require("pheatmap")) install.packages("pheatmap"); library(pheatmap)  
#if (!require("ggVennDiagram")) install.packages("ggVennDiagram"); library(ggVennDiagram)  
  
library(DESeq2)

## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##   
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':  
##   
## IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':  
##   
## anyDuplicated, append, as.data.frame, basename, cbind, colnames,  
## dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,  
## grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,  
## order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,  
## rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,  
## union, unique, unsplit, which.max, which.min

##   
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':  
##   
## expand.grid, I, unname

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##   
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':  
##   
## colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,  
## colCounts, colCummaxs, colCummins, colCumprods, colCumsums,  
## colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,  
## colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,  
## colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,  
## colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,  
## colWeightedMeans, colWeightedMedians, colWeightedSds,  
## colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,  
## rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,  
## rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,  
## rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,  
## rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,  
## rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,  
## rowWeightedMads, rowWeightedMeans, rowWeightedMedians,  
## rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

## Welcome to Bioconductor  
##   
## Vignettes contain introductory material; view with  
## 'browseVignettes()'. To cite Bioconductor, see  
## 'citation("Biobase")', and for packages 'citation("pkgname")'.

##   
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':  
##   
## rowMedians

## The following objects are masked from 'package:matrixStats':  
##   
## anyMissing, rowMedians

library(apeglm)  
library(ggplot2)  
library(pheatmap)  
library(ggVennDiagram)

## Loading data

data <- read.csv("ortho\_count.csv", header=T, row.names = 1)  
info <- read.table("orthoRef.txt", header = T, sep ="\t")

## Running DESeq and editing data

de <- DESeqDataSetFromMatrix(data, info, ~diet)

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in  
## design formula are characters, converting to factors

keep <- rowSums(counts(de)) >= 10  
de <- de[keep,]  
  
deSeqData <- DESeq(de)

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## -- note: fitType='parametric', but the dispersion trend was not well captured by the  
## function: y = a/x + b, and a local regression fit was automatically substituted.  
## specify fitType='local' or 'mean' to avoid this message next time.

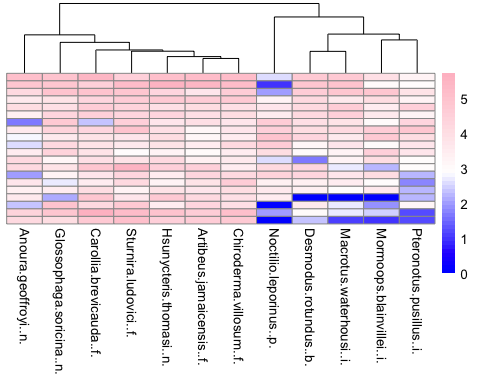
## final dispersion estimates

## fitting model and testing

#export normalized read count  
normCounts <- counts(deSeqData, normalized = T)  
#write.csv(normCounts, "normal.ortho\_count.csv")  
  
#p value less than .05 is d.e.   
result <- results(deSeqData, alpha = 0.05)  
  
#summary(res)  
  
# order based on p adjusted value   
resOrdered <- result[order(result$padj),]  
#write.csv(resOrdered, "deSeq.order.ortho.csv")

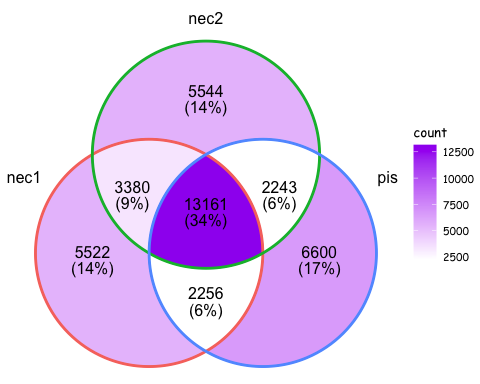
## Plotting the data

normCount <- read.csv("normal.ortho\_count.csv", row.names = 1)  
  
sigCounts <- normCount[1:20,]   
  
pheatmap(log2(sigCounts + 1), color=colorRampPalette(c("blue", "white", "pink"))(50),treeheight\_row= 0, show\_rownames = F, )

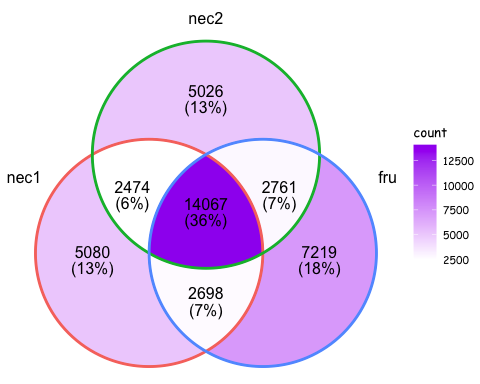


## Using ggVennDiagram to show overlapping orthogroups

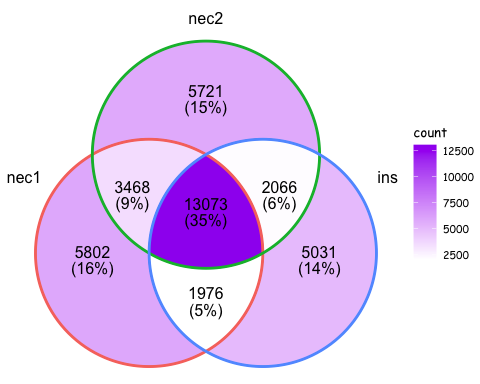
#reading in the files   
  
#nerctarivores  
ange <- readLines("ange01.csv")  
loth<- readLines("loth01.csv")  
glso<- readLines("glso01.csv")  
  
#frugivores  
arja <- readLines("arja01.csv")  
cabr <- readLines("cabr01.csv")  
chvi<- readLines("chvi01.csv")  
stlu<- readLines("stlu01.csv")  
  
#blood feeding  
dero<- readLines("dero01.csv")  
  
#insectivores  
mawa<- readLines("mawa01.csv")  
mobl<- readLines("mobl01.csv")  
ptpa<- readLines("ptpa01.csv")  
  
#piscivore  
nole<- readLines("nole01.csv")  
  
  
# nectarivores vs piscivore   
nec\_pis <- list(nec1 = ange, nec2= glso, pis = nole )  
  
ggVennDiagram(nec\_pis[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



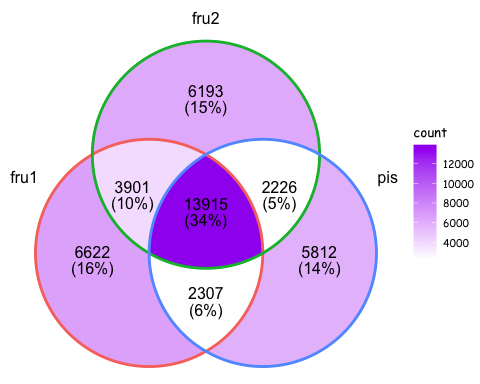
# nectarivores vs frugivore   
nec\_fru <- list(nec1 = ange, nec2= glso, fru = arja )  
  
ggVennDiagram(nec\_fru[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



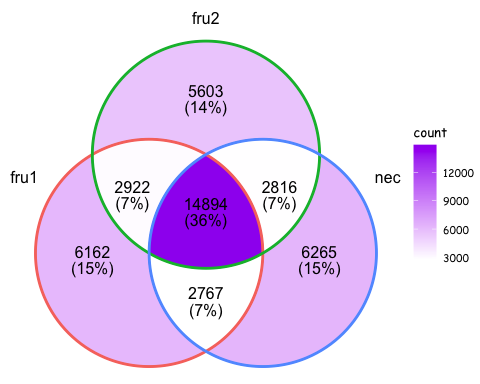
# nectarivores vs insectivore   
nec\_ins <- list(nec1 = ange, nec2= glso, ins = ptpa )  
  
ggVennDiagram(nec\_ins[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



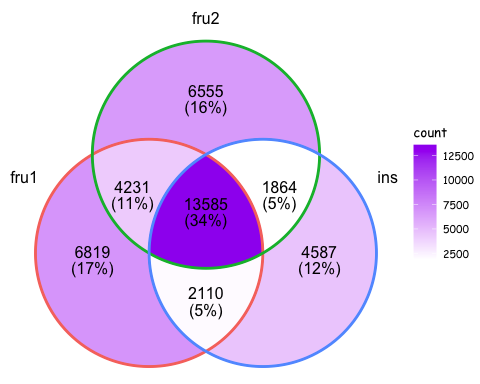
# frugivores vs piscivore  
fru\_pis <- list(fru1 = arja, fru2= stlu, pis = nole )  
  
ggVennDiagram(fru\_pis[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



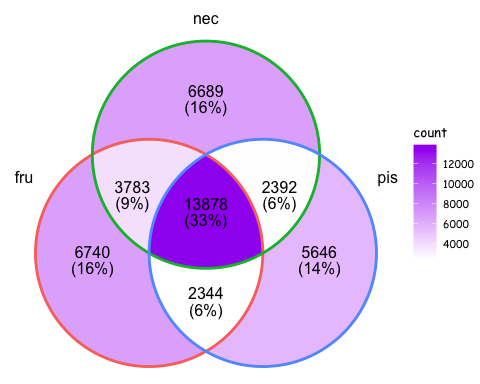
# frugivores vs nectarivore  
fru\_nec <- list(fru1 = arja, fru2= stlu, nec = loth )  
  
ggVennDiagram(fru\_nec[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



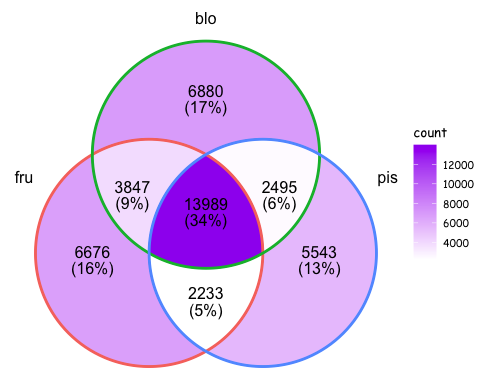
# frugivores vs insectivore  
fru\_ins <- list(fru1 = arja, fru2= stlu, ins = ptpa )  
  
ggVennDiagram(fru\_ins[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



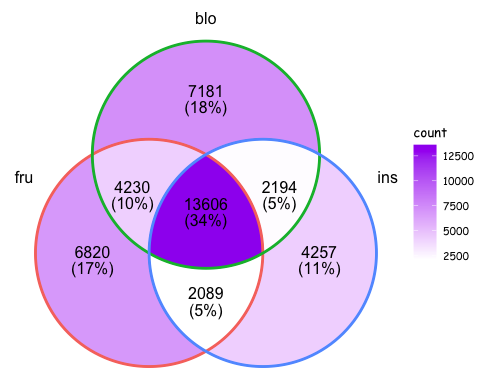
# frugivore vs nectarivore vs piscivore  
fru\_pis\_nec <- list(fru = arja, nec = loth, pis= nole )  
  
ggVennDiagram(fru\_pis\_nec[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



# frugivore vs blood feeding vs piscivore  
fru\_pis\_blo <- list(fru = arja, blo = dero, pis= nole )  
  
ggVennDiagram(fru\_pis\_blo[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



# frugivore vs insectivore vs piscivore  
fru\_ins\_blo <- list(fru = arja, blo = dero, ins = ptpa)  
  
ggVennDiagram(fru\_ins\_blo[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))



# frugivore vs nectarivore vs blood feeding  
fru\_blo\_nec <- list(fru = arja, nec = loth, blo= dero )  
  
ggVennDiagram(fru\_blo\_nec[1:4],label\_alpha = 0, stroke\_size = 0.1) +  
 ggplot2::scale\_fill\_gradient(low="white",high = "purple") +   
 theme(text = element\_text(size=10, family="Comic Sans MS"))

