Assignment 4.3, Comparative transcriptomics, Advanced Bioinformatics for NGS

Stefano Pellegrini

11/5/2020

```
load('inputData.Rdata')
source("shared_functions.R")
library("reshape")
library("VennDetail")
```

1) Try to include also the two queenless ant species in colony effect correction, and report the similarity matrix (Heatmap + Hclustering that we used in the classroom) and PCA result.

```
# Section 2: Using Combat to normalize batch effect.
sampleTable$caste[which(sampleTable$caste == 'Minor_worker')] = 'Worker'
normal_ant = which(sampleTable$species %in%
                     c("Aech",'Mpha',"Lhum",'Sinv',"Lnig", "Cbir", "Dqua"))
ortholog_exp.ant = ortholog_counts[,normal_ant]
sampleTable.ant = sampleTable[normal_ant,]
ortholog_exp.ant = ortholog_exp.ant[!apply(ortholog_exp.ant, 1, anyNA),]
ortholog_exp.ant.norm = log2(normalize.quantiles(ortholog_exp.ant)+1)
colnames(ortholog_exp.ant.norm) = colnames(ortholog_exp.ant)
rownames(ortholog_exp.ant.norm) = rownames(ortholog_exp.ant)
ortholog_exp.ant.norm =
  ortholog_exp.ant.norm[apply(ortholog_exp.ant.norm, 1,
                              FUN = function(x) return(var(x, na.rm = T) > 0)),]

# Removing colony we also remove the effect of the species
batch = droplevels(sampleTable.ant$colony)
modcombat = model.matrix(~1, data = sampleTable.ant)

# Empirical based method to removed the effect of species identity
combat.ortholog_exp.ant = ComBat(dat=ortholog_exp.ant.norm, batch=batch, mod=modcombat,
                                 mean.only = F, par.prior=TRUE, prior.plots=FALSE)
```

```
## Found 318 genes with uniform expression within a single batch (all zeros); these will not be adjuste
```

```
# Plot heatmap
sampleDists.combat = as.dist(1 - cor(combat.ortholog_exp.ant,method = 's'))
heatmap <- pheatmap(sampleDists.combat,annotation_col = sampleTable.ant[,c(1:3)],
                    annotation_colors = ann_colors,
                    color = colors)
ggsave("heatmap.png", heatmap, width = 8, height = 6, dpi = 300)
heatmap
```

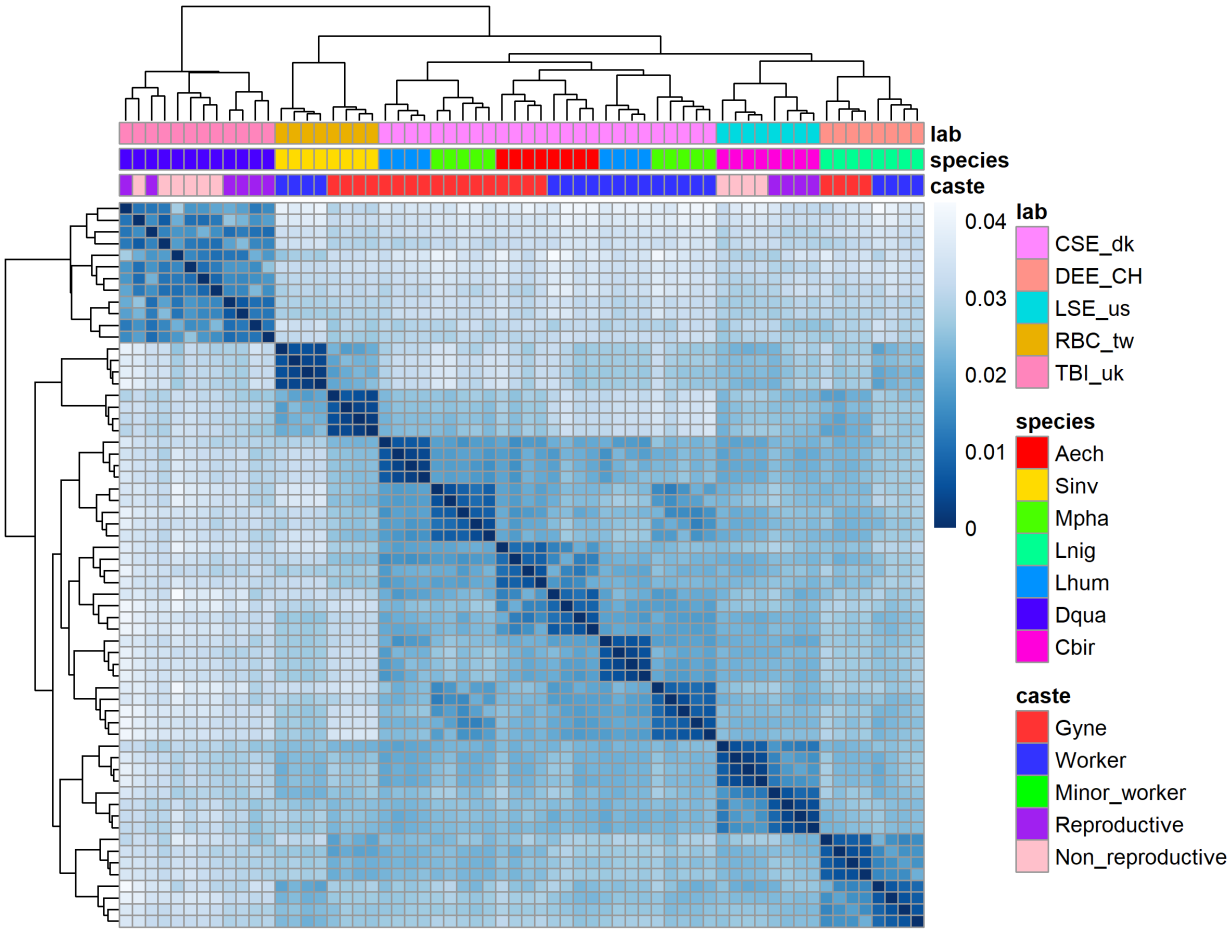
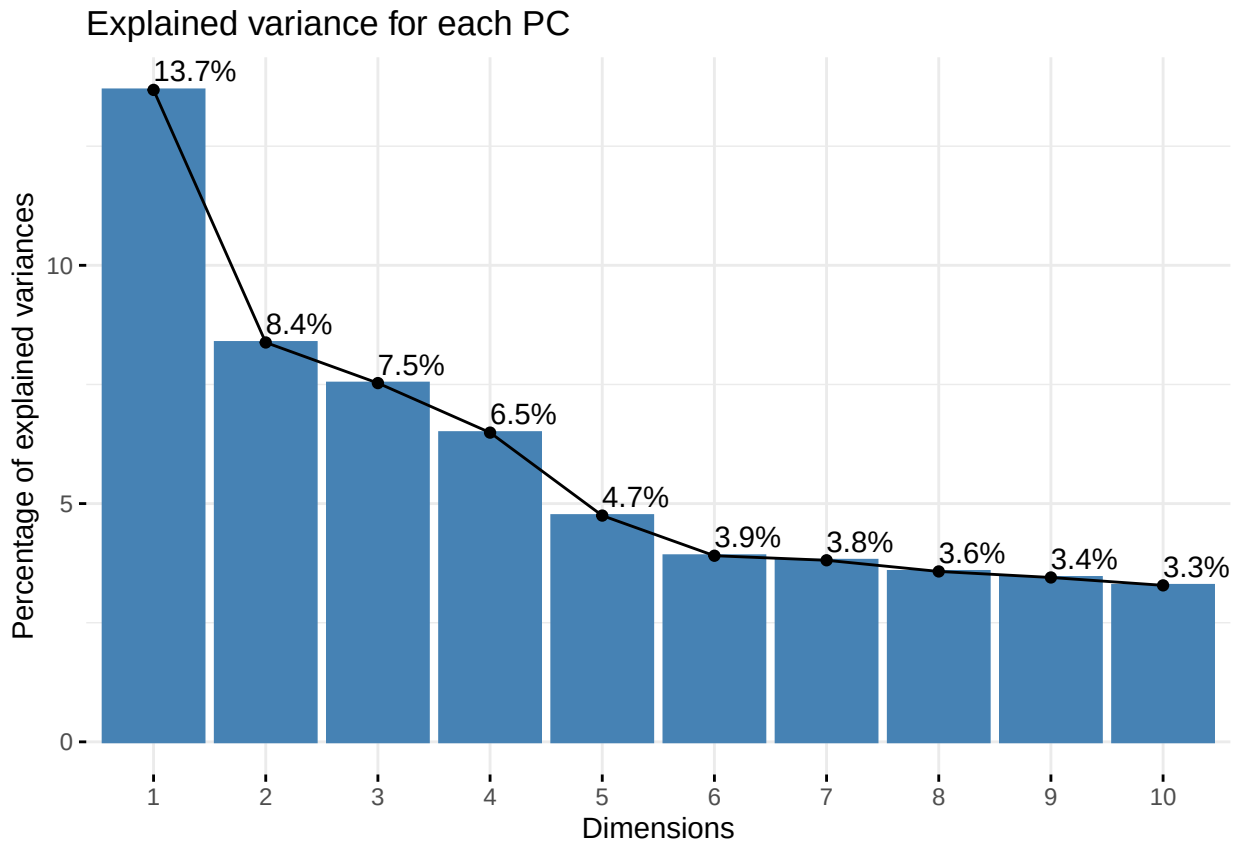


Figure 1: heatmap

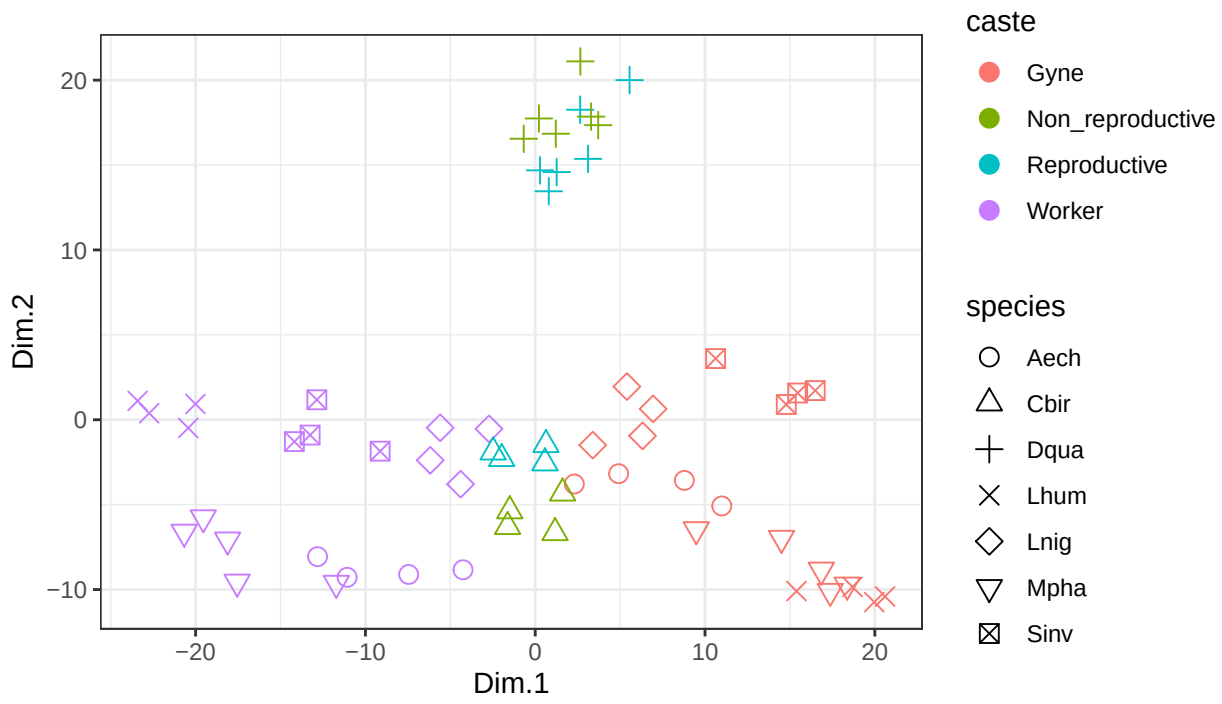
In the heatmap we can see that the samples cluster by caste, species and lab. That was expected because, even if we partially removed the effect of the lab and species by removing the colony effect, these effects are still present in the caste itself. In fact, the expression level between the caste classes can be different across different species.

```
var.gene = order(apply(combat.ortholog_exp.ant,1,var),decreasing = T)[c(1:1000)]
ortholog_exp.combat.pca <- PCA(t(combat.ortholog_exp.ant[var.gene,]),ncp = 4, graph = FALSE)

# Take a look at the amount of variations explained by each PC.
fviz_eig(ortholog_exp.combat.pca, addlabels = TRUE,main = 'Explained variance for each PC')
```




```
pca.combat.var = ortholog_exp.combat.pca$eig
pca.combat.data = cbind(ortholog_exp.combat.pca$ind$coord,sampleTable.ant)
ggplot(pca.combat.data, aes(x = Dim.1, y = Dim.2, color = caste, shape = species)) +
  geom_point(size=3) +
  coord_fixed() +
  scale_shape_manual(values = seq(1,8)) + theme_bw()
```



It is possible to observe that the effect of the caste is captured by the first principal component (13.7% of captured variance), where the worker caste clustered on the left side of the plot, the gyne on the right side, and non-reproductive and reproductive castes are clustered in the center. The second principal component (8.4% captured variance) seemed to capture the species effect. In this regard, it is interesting to note that the queenless species *Dqua* formed one cluster isolated from the other queenless species *Cbir*, which clustered together with the other species.

```
# Section 4: Identification of caste differentially expressed genes
normal_ant = which(sampleTable$species %in%
                     c("Aech",'Mpha',"Lhum",'Sinv',"Lnig")) # Use only 5 species
ortholog_counts.ant = ortholog_counts[,normal_ant]
ortholog_counts.ant = ortholog_counts.ant[!apply(ortholog_counts.ant, 1, anyNA),]
ortholog_counts.ant.norm = matrix(as.integer(ortholog_counts.ant),
                                  ncol = dim(ortholog_counts.ant)[2],
                                  dimnames = list(rownames(ortholog_counts.ant),
                                                  colnames(ortholog_counts.ant)))

target_species = 'Aech'
dds <- DESeqDataSetFromMatrix(
  ortholog_counts.ant.norm[,which(sampleTable.ant$species %in% target_species)],
  sampleTable.ant[which(sampleTable.ant$species %in% target_species),],
  ~ caste)

dds = DESeq(dds)
res.aech = results(dds, contrast = c("caste",c("Gyne",'Worker')),alpha = 0.05)
summary(res.aech)
```

```
##
## out of 6540 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 257, 3.9%
## LFC < 0 (down)     : 66, 1%
## outliers [1]       : 0, 0%
## low counts [2]     : 127, 1.9%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

2.1) Can you report the number of overlapping DEGs in two, three, four, and all five typical ant species, i.e. Aech Mpha Lhum Lnig and Sinv, similar to the last slide of the class lecture?

```
# Perform an independent DESeq analysis for each target species
deseq_target_species <- function(target){
  dds <- DESeqDataSetFromMatrix(
    ortholog_counts.ant.norm[,which(sampleTable.ant$species %in% target)],
    sampleTable.ant[which(sampleTable.ant$species %in% target),],
    ~ caste)
  dds = DESeq(dds)
  res <- results(dds, contrast = c("caste",c("Gyne",'Worker')), alpha = 0.05)
  return(res)
}

target_species <- c("Aech",'Mpha',"Lhum",'Sinv',"Lnig")
res_species <- sapply(target_species, function(x) deseq_target_species(x))

# Divide between up and downregulated genes
get_direction <- function(deseq_res, direction = "up"){
  if (direction == "up"){
    return(deseq_res[which(deseq_res$log2FoldChange > 0),])
  }else{
    return(deseq_res[which(deseq_res$log2FoldChange < 0),])
  }
}

res_species_up <- sapply(res_species, function(x) get_direction(x, direction = "up"))
res_species_down <- sapply(res_species, function(x) get_direction(x, direction = "down"))

# Get differentially expressed genes
get_degs <- function(deseq_res){
  return(rownames(deseq_res[which(deseq_res$padj < 0.05),]))
}

# Get overlaps over combinations of species
get_overlaps <- function(n_species, list_deseq_res){
  overlaps <- list()
  combinations <- combn(1:5, n_species)
  for (col in seq(ncol(combinations))){
    comb = combinations[,col]
    degs_list = lapply(comb, function(x) get_degs(list_deseq_res[[x]]))
```

```
    species = paste(target_species[comb], collapse = ".")
    overlaps[[species]] = length(Reduce(intersect, degs_list))
  }
  # Convert to df and add the column size
  size = as.data.frame(rep(n_species, ncol(combinations)))
  df = t(data.frame(overlaps))
  df = cbind(df, size)
  colnames(df) = c("Overlaps", "Size")
  return(df)
}

overlaps_up <- lapply(1:5, function(x) get_overlaps(x, res_species_up))
overlaps_down <- lapply(1:5, function(x) get_overlaps(x, res_species_down))

# Convert each list to a single dataframe
overlaps_up <- rownames_to_column(do.call(rbind.data.frame, overlaps_up), var = "Species")
overlaps_down <- rownames_to_column(do.call(rbind.data.frame, overlaps_down),
                                    var = "Species")

# Make a single dataframe for both up and down regulated genes
overlaps <- overlaps_up %>%
  mutate(Overlaps_down = overlaps_down$Overlaps) %>%
  relocate(Size, .after = Species)
colnames(overlaps)[3] = c("Overlaps_up")
```

The numbers of common up and down regulated genes (p-value adjusted threshold 0.05) in the different combinations of species, are shown in the following table, boxplot and Venn diagram.

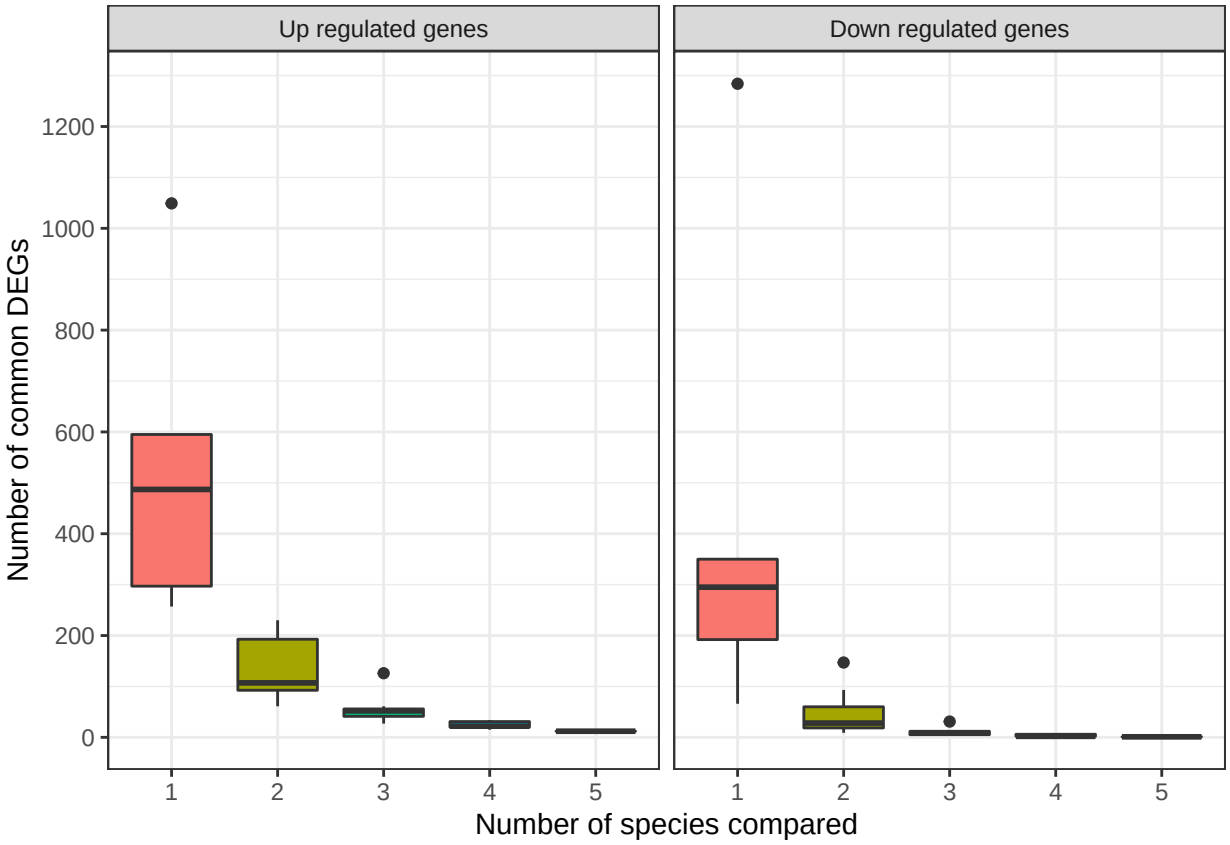
```
# Table
overlaps
```

```
##                    Species Size Overlaps_up Overlaps_down
## 1                     Aech    1         257            66
## 2                     Mpha    1         487           350
## 3                     Lhum    1        1049          1284
## 4                     Sinv    1         595           295
## 5                     Lnig    1         297           192
## 6                Aech.Mpha    2          98            16
## 7                Aech.Lhum    2          91            22
## 8                Aech.Sinv    2         116            18
## 9                Aech.Lnig    2          61             9
## 10               Mpha.Lhum    2         230           147
## 11               Mpha.Sinv    2         219            67
## 12               Mpha.Lnig    2          80            20
## 13               Lhum.Sinv    2         218            93
## 14               Lhum.Lnig    2          97            39
## 15               Sinv.Lnig    2         117            34
## 16           Aech.Mpha.Lhum    3          50            11
## 17           Aech.Mpha.Sinv    3          61             9
## 18           Aech.Mpha.Lnig    3          27             3
## 19           Aech.Lhum.Sinv    3          54             9
## 20           Aech.Lhum.Lnig    3          30             3
## 21           Aech.Sinv.Lnig    3          40             4
## 22           Mpha.Lhum.Sinv    3         126            31
```

```
## 23           Mpha.Lhum.Lnig    3          45             9
## 24           Mpha.Sinv.Lnig    3          55            14
## 25           Lhum.Sinv.Lnig    3          56            12
## 26      Aech.Mpha.Lhum.Sinv    4          31             6
## 27      Aech.Mpha.Lhum.Lnig    4          15             2
## 28      Aech.Mpha.Sinv.Lnig    4          20             2
## 29      Aech.Lhum.Sinv.Lnig    4          22             1
## 30      Mpha.Lhum.Sinv.Lnig    4          34             8
## 31 Aech.Mpha.Lhum.Sinv.Lnig    5          12             1
```

```r
# Boxplot
colnames(overlaps)[c(3, 4)] = c("Up regulated genes",
                                "Down regulated genes")
overlaps_plot <- melt(overlaps, id.vars = c("Species", "Size"),
                      measure.vars = c("Up regulated genes",
                                       "Down regulated genes"))

overlaps_plot %>% ggplot(aes(y = value, x = as.factor(Size), fill = as.factor(Size))) +
  geom_boxplot() +
  xlab("Number of species compared") + ylab("Number of common DEGs") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  facet_grid(~variable) + theme_bw() + theme(legend.position = "none")
```



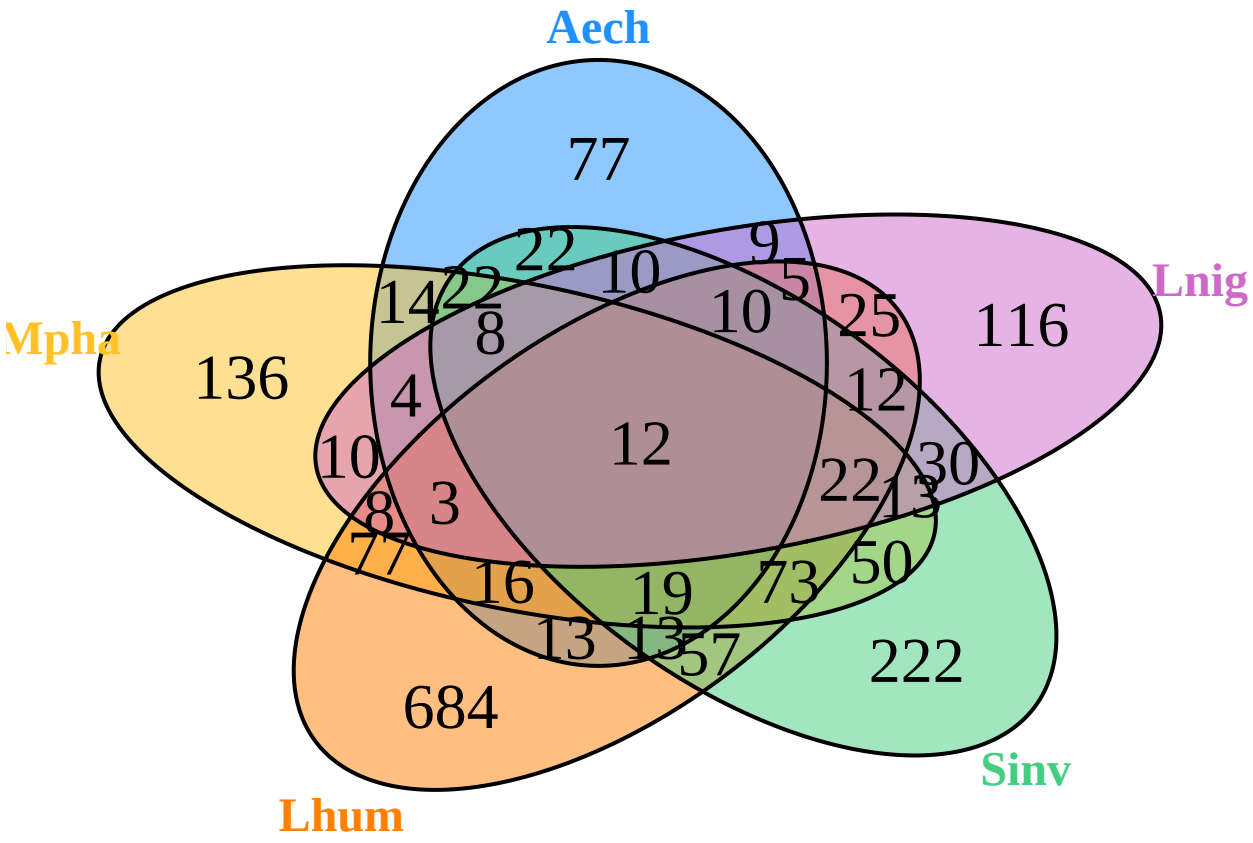

```r
# Venn diagram
degs_up <- sapply(res_species_up, function(x) get_degs(x))
degs_down <- sapply(res_species_down, function(x) get_degs(x))
```

```
venn_up <- venndetail(list(Aech = degs_up$Aech, Mpha = degs_up$Mpha,
                           Lhum = degs_up$Lhum, Sinv = degs_up$Sinv,
                           Lnig = degs_up$Lnig))

venn_down <- venndetail(list(Aech = degs_down$Aech, Mpha = degs_down$Mpha,
                             Lhum = degs_down$Lhum, Sinv = degs_down$Sinv,
                             Lnig = degs_down$Lnig))
```

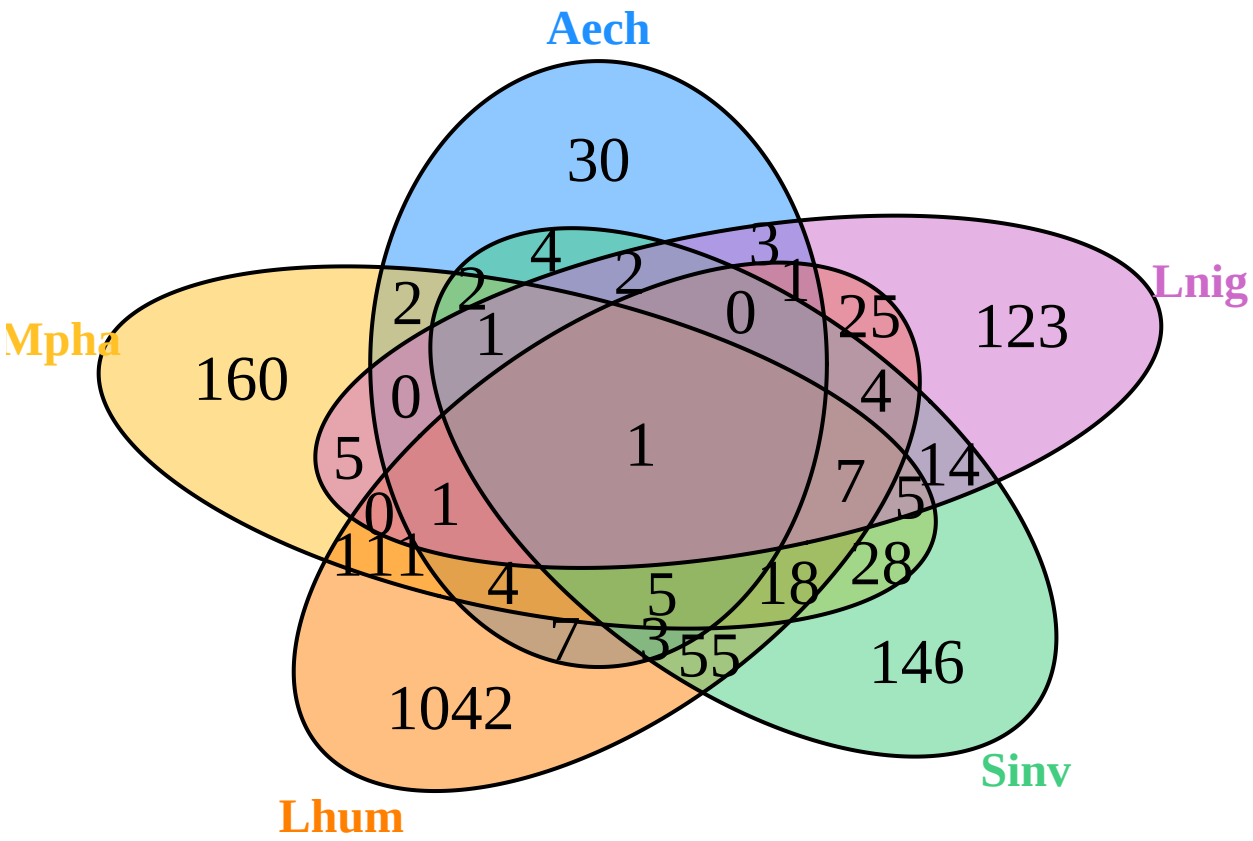
Venn diagram up regulated genes:

```
plot(venn_up)
```



Venn diagram down regulated genes:

```
plot(venn_down)
```



2.2) And how many DEGs have you found if you use the model: Exp ~ Caste + Species? Is this number different from the number of overlapping DEGs? Why?

```
# Interaction model
target_species <- c("Aech",'Mpha',"Lhum",'Sinv',"Lnig")
dds <- DESeqDataSetFromMatrix(
  ortholog_counts.ant.norm[,which(sampleTable.ant$species %in% target_species)],
  sampleTable.ant[which(sampleTable.ant$species %in% target_species),],
  ~ caste + species)
dds = DESeq(dds)
result_interaction = results(dds, contrast = c("caste",c("Gyne",'Worker')), alpha = 0.05)
summary(result_interaction)
```

```
##
## out of 6562 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 743, 11%
## LFC < 0 (down)     : 746, 11%
## outliers [1]       : 8, 0.12%
## low counts [2]     : 0, 0%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
up_deg_inter <- get_direction(result_interaction, direction = "up")
down_deg_inter <- get_direction(result_interaction, direction = "down")
```

```r
up_deg_inter <- get_degs(up_deg_inter)
down_deg_inter <- get_degs(down_deg_inter)

# Get overlapping gene names in all 5 species (previous task result)
get_overlaps_names <- function(n_species, list_deseq_res){
  overlaps <- list()
  combinations <- combn(1:5, n_species)
  for (col in seq(ncol(combinations))){
    comb = combinations[,col]
    degs_list = lapply(comb, function(x) get_degs(list_deseq_res[[x]]))
    species = paste(target_species[comb], collapse = ".")
    overlaps[[species]] = Reduce(intersect, degs_list)
  }
  return(overlaps)
}

up_deg_comb <- get_overlaps_names(5, res_species_up)
down_deg_comb <- get_overlaps_names(5, res_species_down)

# Check which DEGs obtained by model 1 is also present in the model 2
length(up_deg_comb$Aech.Mpha.Lhum.Sinv.Lnig %in% up_deg_inter)
```

```
## [1] 12
```

```r
length(down_deg_comb$Aech.Mpha.Lhum.Sinv.Lnig %in% down_deg_inter)
```

```
## [1] 1
```

In both models we tested the number of differentially expressed genes between gyne and worker castes using DESeq2. In model 2 (~ Caste + Species) I obtained 743 up-regulate and 746 down-regulated genes (p-value adjusted threshold 0.05). While in the model 1, where we inferred the DEGs in each species and than we extracted the ones overlapping in the 5 species (result of task 2.1), I obtained only 12 up-regulated and 1 down-regulated genes. Furthermore all DEGs genes identified by model 1 were also identified by model 2. The larger amount of DEGs obtained by model 2 is explained by its larger detection power. In fact, while model 1 consider only the differentially expressed genes between gyne and worker castes that are shared across the 5 species, model 2 takes into account the species effect into the linear model, which considerably increase the detection power for DEGs.