5. PCA genes - Lab

Fay

2022-10-08

Load libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(stringr)
library(FactoMineR)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(ggpubr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(pheatmap)
```

Load data

```
hm <- read.csv("output_data/imputed_MICE.csv")
```

vectors for selecting

```
Gene_lab   <- c("IFNy", "CXCR3", "IL.6", "IL.13", "IL.10",
                "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                "TICAM1", "TNF") #"IL.12", "IRG6")

#add a suffix to represent changes in data file
Gene_lab_imp <- paste(Gene_lab, "imp", sep = "_")

Genes_wild   <- c("IFNy", "CXCR3", "IL.6", "IL.13", "IL.10",
                  "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF") #, "IL.12", "IRG6")

Genes_wild_imp <- paste(Genes_wild, "imp", sep = "_")

Facs_lab <- c("Position", "CD4", "Treg", "Div_Treg", "Treg17", "Th1",
                    "Div_Th1", "Th17", "Div_Th17", "CD8", "Act_CD8",
                    "Div_Act_CD8", "IFNy_CD4", "IFNy_CD8","Treg_prop",
                    "IL17A_CD4")

Facs_lab_imp <- paste(Facs_lab, "imp", sep = "_")

Facs_wild <- c( "Treg", "CD4", "Treg17", "Th1", "Th17", "CD8",
                     "Act_CD8", "IFNy_CD4", "IL17A_CD4", "IFNy_CD8")

Facs_wild_imp <- paste(Facs_wild, "imp", sep = "_")
```

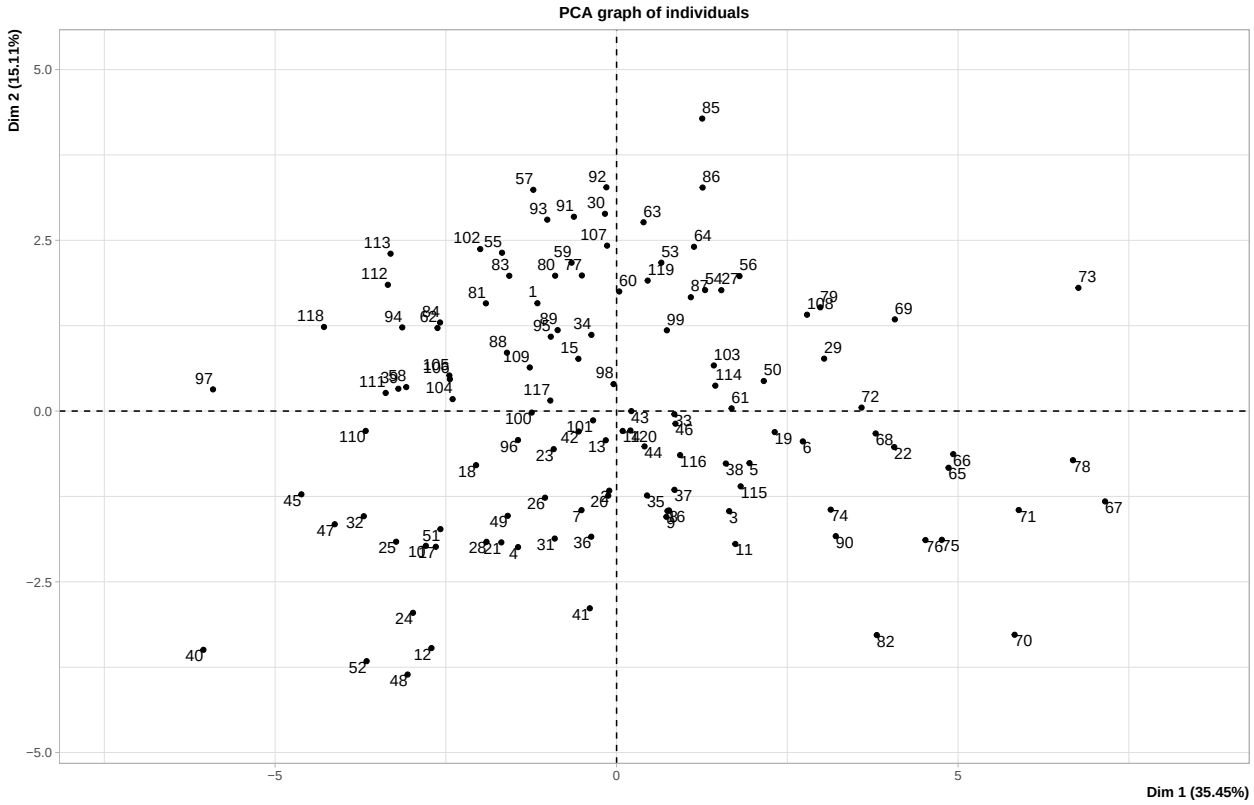
PCA on the lab genes -*imputed*

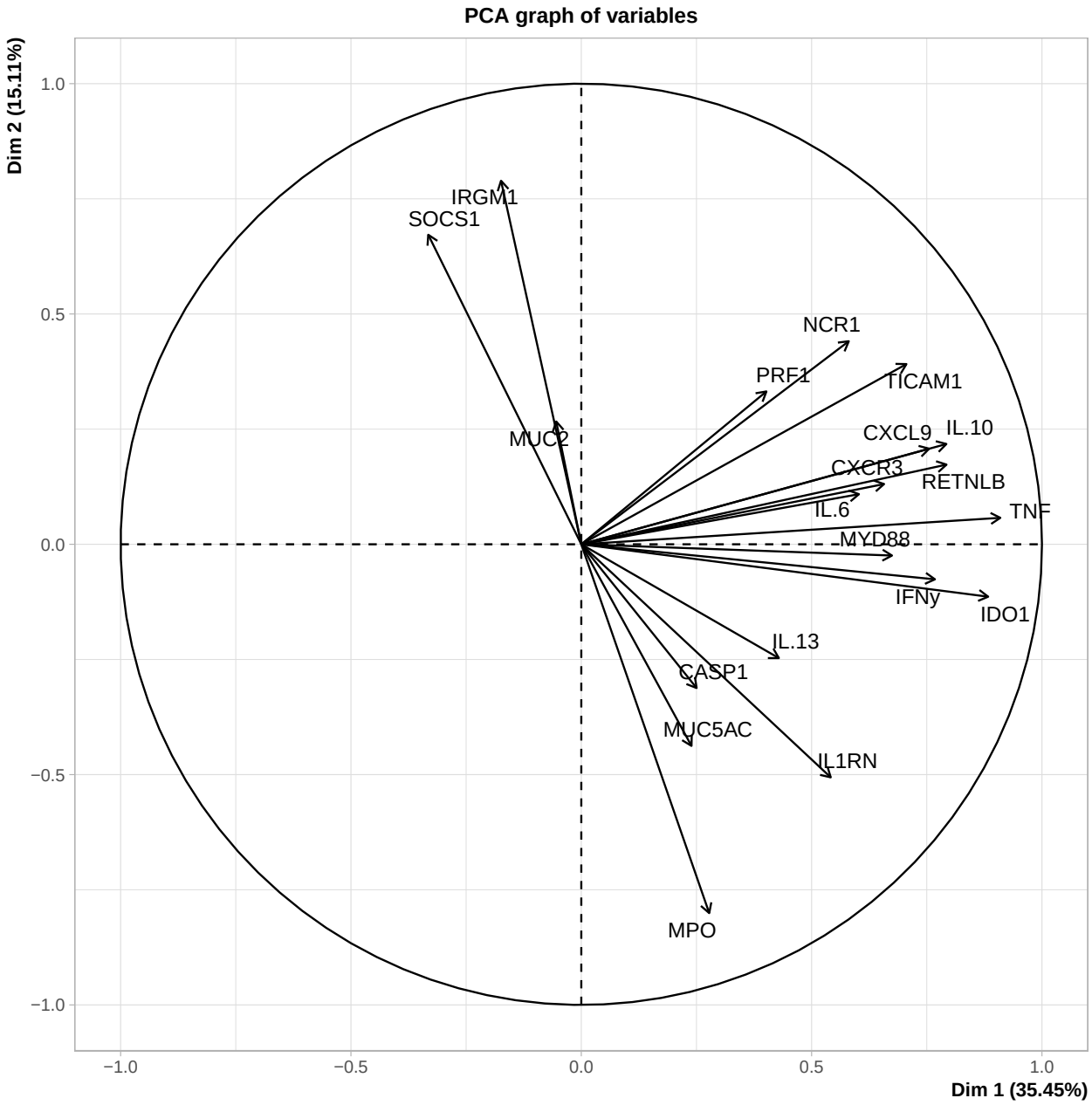
```
#select the genes and lab muce
lab <- hm %>%
  dplyr::filter(origin == "Lab", Position == "mLN") #selecting for mln to avoid
# duplicates

lab <- unique(lab)
```

```
genes <- lab %>%
  dplyr::select(Gene_lab_imp) %>%
  rename_with(~str_remove(., '_imp'))
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(Gene_lab_imp)` instead of `Gene_lab_imp` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
genes <- unique(genes)

# we can now run a normal pca on the complete data set
res.pca <- PCA(genes)
```





Caution: When imputing data, the percentages of inertia associated with the first dimensions will be overestimated.

Another problem: the imputed data are, when the pca is performed considered like real observations. But they are estimations!!

Visualizing uncertainty due to issing data:

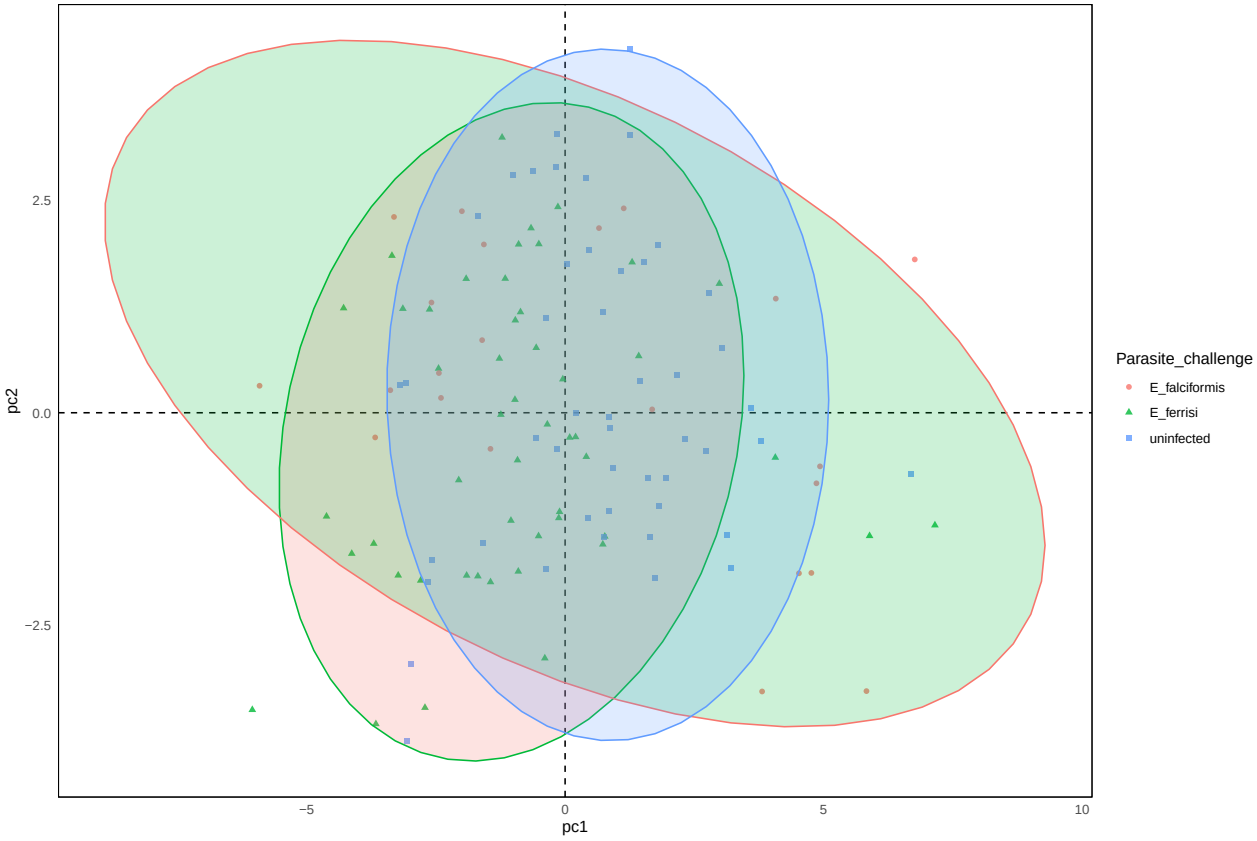
–> mulrimple imputation: generate several plausible values for each missing data point

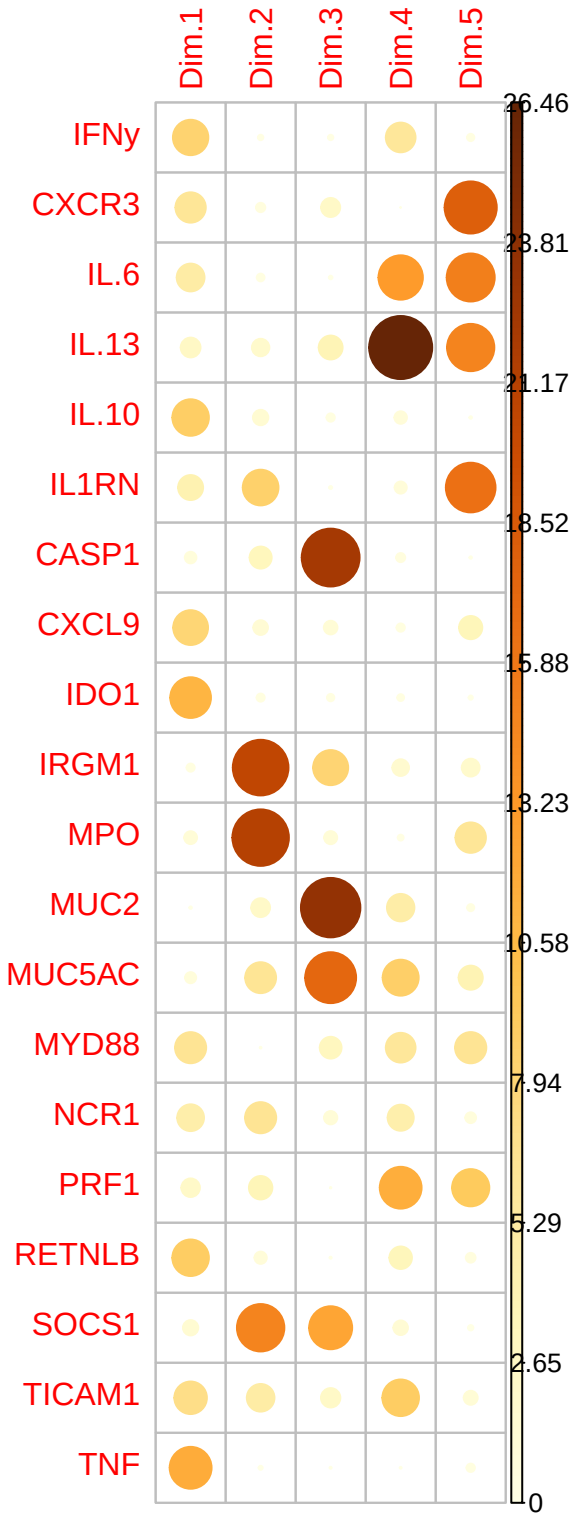
We here visualize the variability, that is uncertainty on the plane defined by two pca axes.

Biplot of the imputed gene pca

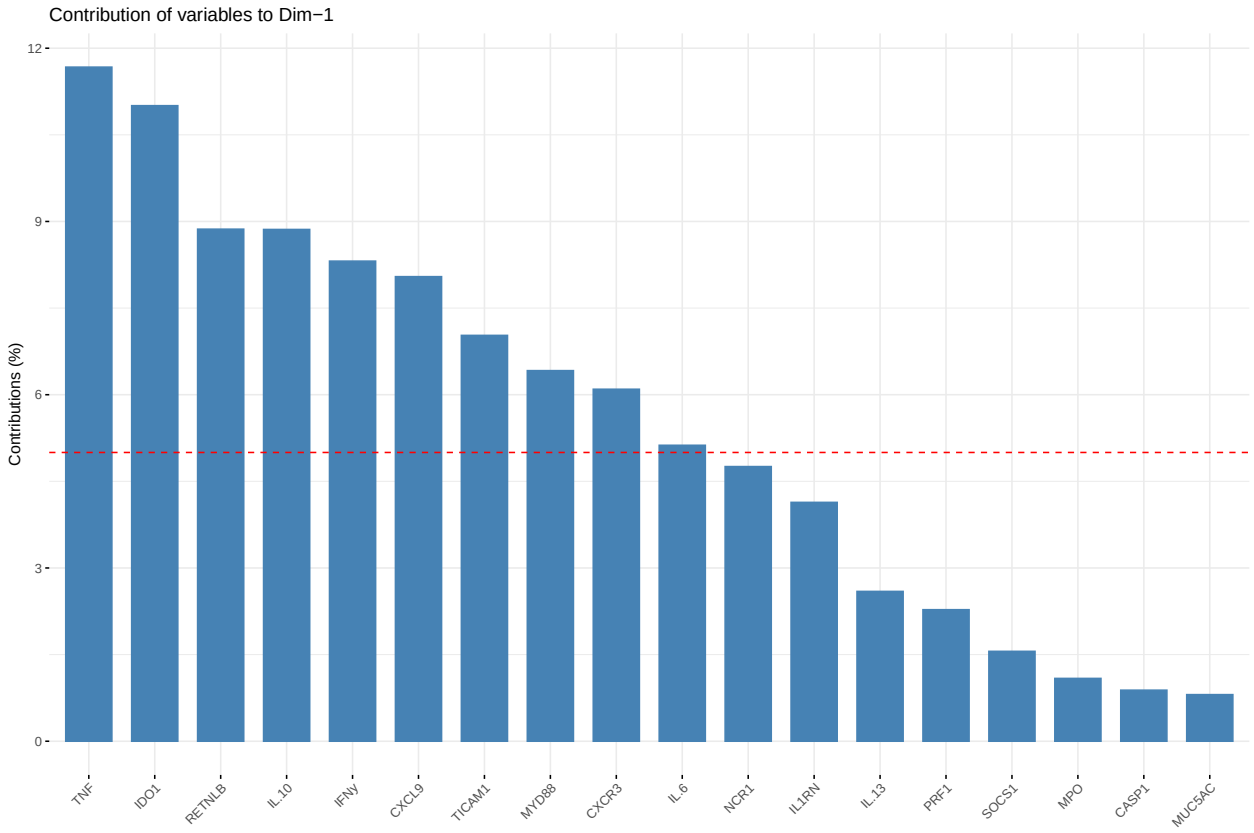
```
#Now we can make our initial plot of the PCA.
lab %>%
  ggplot(aes(x = pc1, y = pc2,
```

```
            color = Parasite_challenge,
            shape = Parasite_challenge)) +
geom_hline(yintercept = 0, lty = 2) +
geom_vline(xintercept = 0, lty = 2) +
geom_point(alpha = 0.8) +
stat_ellipse(geom="polygon",
             aes(fill = challenge_infection),
             alpha = 0.2, show.legend = FALSE,
             level = 0.95) +
theme_minimal() +
theme(panel.grid = element_blank(),
      panel.border = element_rect(fill= "transparent"))
```

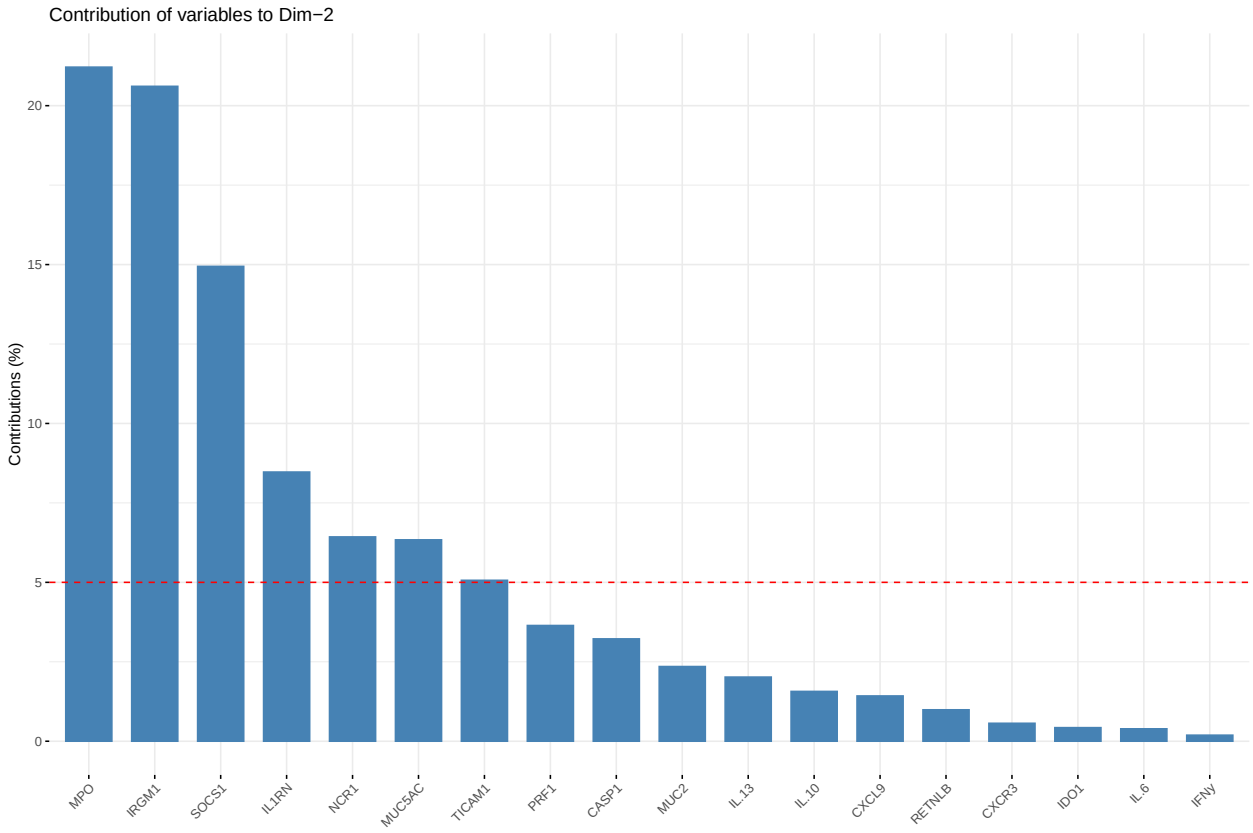


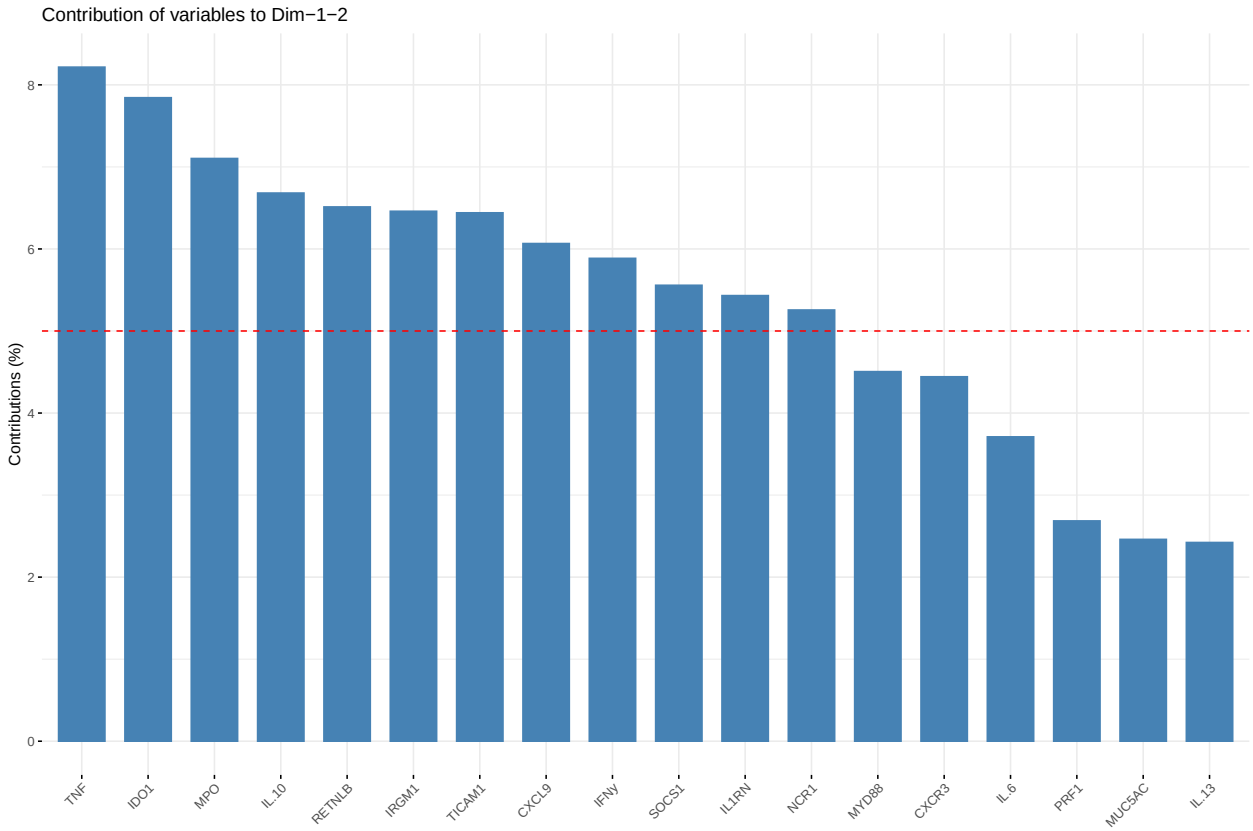


The function fviz_contrib() [factoextra package] can be used to draw a bar plot of variable contributions. If your data contains many variables, you can decide to show only the top contributing variables. The R code below shows the top 10 variables contributing to the principal components:




```
# Contributions of variables to PC2
fviz_contrib(res.pca, choice = "var", axes = 2, top = 18)
```



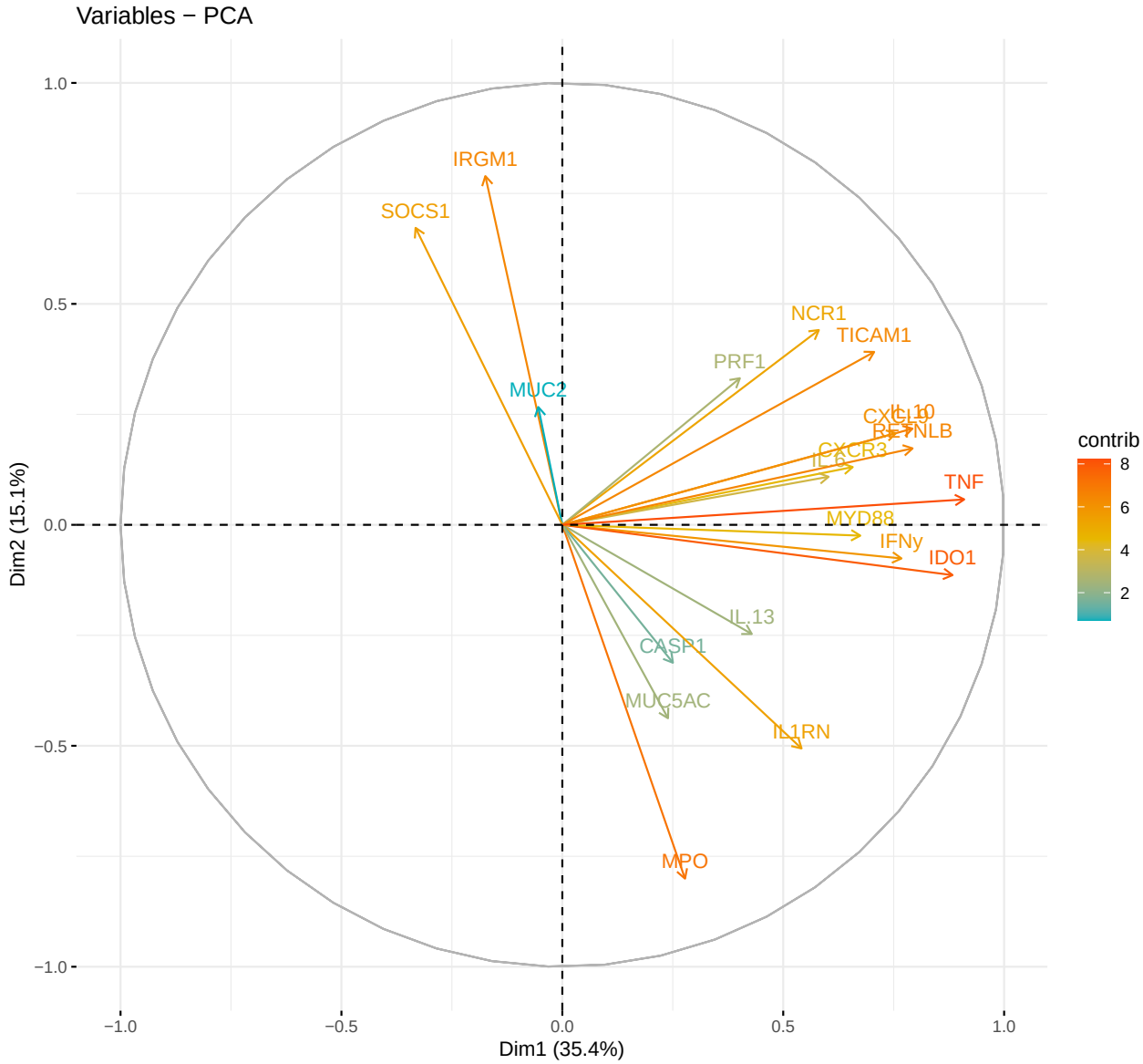


The red dashed line on the graph above indicates the expected average contribution. If the contribution of the variables were uniform, the expected value would be 1/length(variables) = 1/10 = 10%. For a given

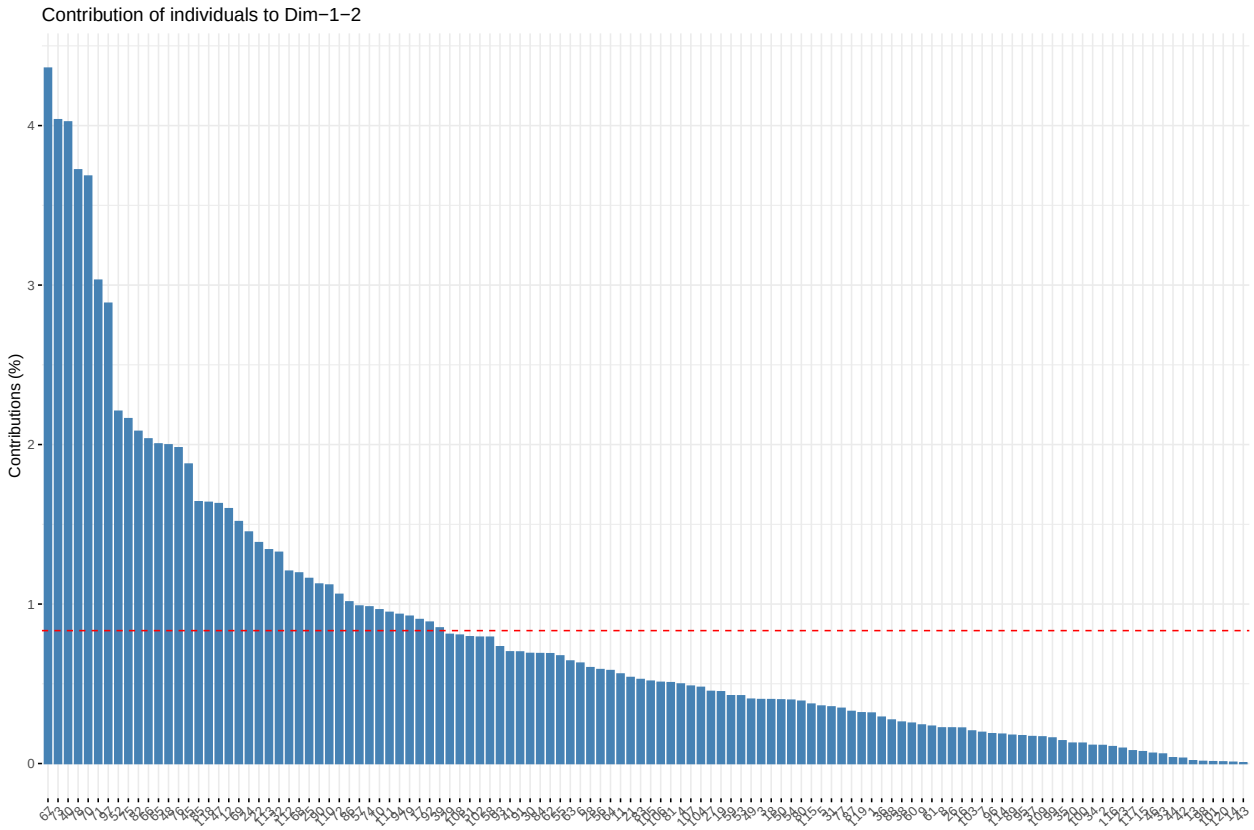
component, a variable with a contribution larger than this cutoff could be considered as important in contributing to the component.

Note that, the total contribution of a given variable, on explaining the variations retained by two principal components, say PC1 and PC2, is calculated as contrib = [(C1 * Eig1) + (C2 * Eig2)]/(Eig1 + Eig2), where

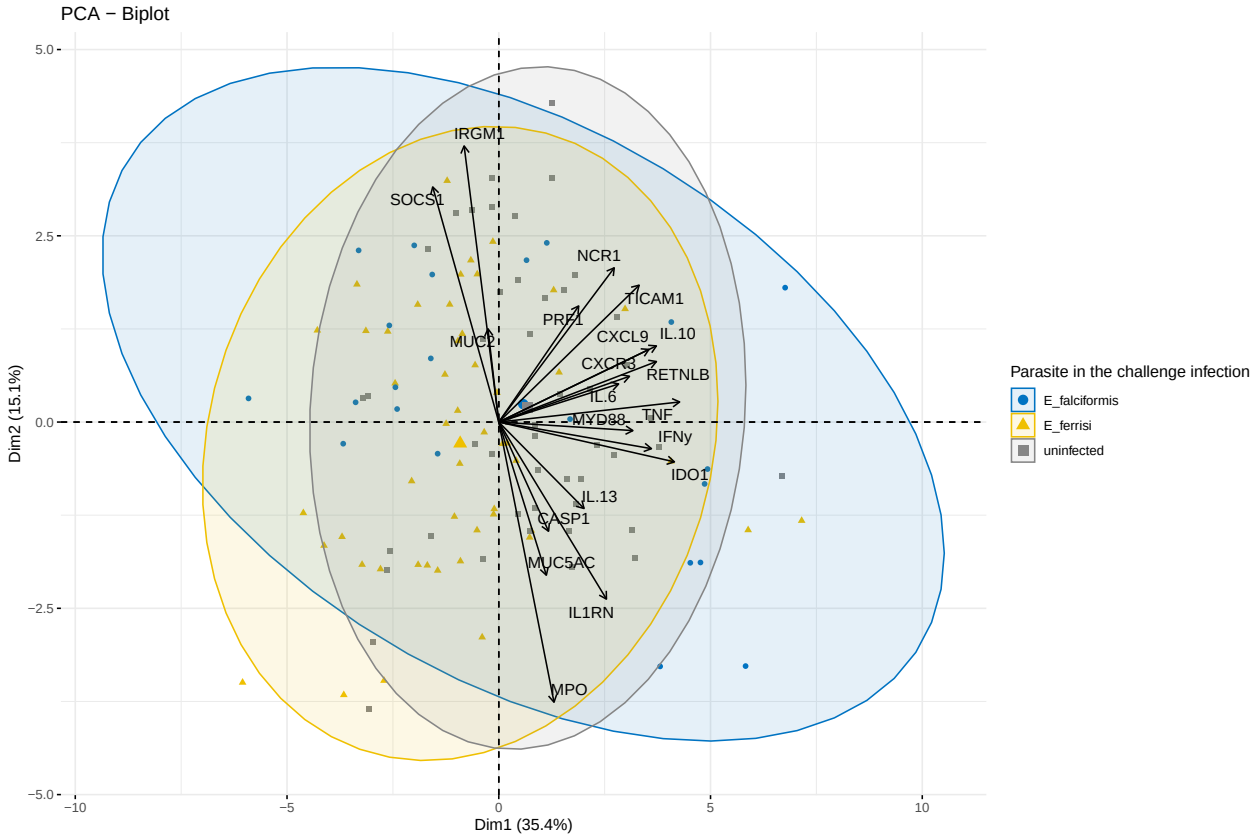
C1 and C2 are the contributions of the variable on PC1 and PC2, respectively Eig1 and Eig2 are the eigenvalues of PC1 and PC2, respectively. Recall that eigenvalues measure the amount of variation retained by each PC. In this case, the expected average contribution (cutoff) is calculated as follow: As mentioned above, if the contributions of the 10 variables were uniform, the expected average contribution on a given PC would be 1/10 = 10%. The expected average contribution of a variable for PC1 and PC2 is : [(10* Eig1) + (10 * Eig2)]/(Eig1 + Eig2)



To visualize the contribution of individuals to the first two principal components:



PCA + Biplot combination



In the following example, we want to color both individuals and variables by groups. The trick is to use pointshape = 21 for individual points. This particular point shape can be filled by a color using the argument fill.ind. The border line color of individual points is set to "black" using col.ind. To color variable by groups, the argument col.var will be used.

Linear models:

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + Parasite_challenge, data = lab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.184  -3.163   0.203   3.654  14.256
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   85.6497     1.1139  76.893  < 2e-16 ***
## pc1                            0.1901     0.1861   1.022  0.30897
## pc2                           -0.8691     0.2755  -3.155  0.00205 **
## Parasite_challengeE_ferrisi    5.9792     1.3653   4.379 2.64e-05 ***
## Parasite_challengeuninfected  10.4528     1.3364   7.822 2.76e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.19 on 115 degrees of freedom
## Multiple R-squared:  0.393,  Adjusted R-squared:  0.3719
## F-statistic: 18.62 on 4 and 115 DF,  p-value: 8.023e-12
```

```
## [1] 742.6357
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + Parasite_challenge + hybrid_status,
##     data = lab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.185  -3.428   0.688   3.330  13.957
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       86.0011     1.3618  63.153  < 2e-16 ***
## pc1                                0.1940     0.2270   0.855    0.395
## pc2                               -0.7580     0.3004  -2.524    0.013 *
## Parasite_challengeE_ferrisi        5.7997     1.4210   4.081 8.51e-05 ***
## Parasite_challengeuninfected      10.0550     1.4328   7.018 1.95e-10 ***
## hybrid_statusF0 M. m. musculus    -0.9542     1.4240  -0.670    0.504
## hybrid_statusF1 hybrid             1.6744     1.6017   1.045    0.298
## hybrid_statusF1 M. m. domesticus  -2.0496     2.0622  -0.994    0.322
## hybrid_statusF1 M. m. musculus     1.0148     2.4035   0.422    0.674
## hybrid_statusother                -0.3595     1.4562  -0.247    0.805
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.216 on 110 degrees of freedom
```

```
## Multiple R-squared:  0.4135, Adjusted R-squared:  0.3655
## F-statistic: 8.615 on 9 and 110 DF,  p-value: 1.095e-09
```

```
## [1] 748.5313
```

Try instead: LLR test (likelihood ration) (LM4 package)?

https://www.rdocumentation.org/packages/lmtest/versions/0.9-38/topics/lrtest

In this way you compare each model, with the different variables usesd to predict.

Another way is to compare the AIC. (function : step)

```
weight_lm3 <- lm(max_WL ~ pc1 + pc2 + hybrid_status, data = lab)
weight_no_pc1 <- lm(max_WL ~ pc2 + hybrid_status, data = lab)
weight_no_pc2 <- lm(max_WL ~ pc1  + hybrid_status, data = lab)
weight_no_hybrid <- lm(max_WL ~ pc1 + pc2, data = lab)
lrtest(weight_lm3, weight_no_pc1)
```

```
## Likelihood ratio test
##
## Model 1: max_WL ~ pc1 + pc2 + hybrid_status
## Model 2: max_WL ~ pc2 + hybrid_status
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -385.89
## 2   8 -387.77 -1 3.7675    0.05226 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(weight_lm3, weight_no_pc2)
```

```
## Likelihood ratio test
##
## Model 1: max_WL ~ pc1 + pc2 + hybrid_status
## Model 2: max_WL ~ pc1 + hybrid_status
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -385.89
## 2   8 -386.85 -1 1.9188      0.166
```

```
lrtest(weight_lm3, weight_no_hybrid)
```

```
## Likelihood ratio test
##
## Model 1: max_WL ~ pc1 + pc2 + hybrid_status
## Model 2: max_WL ~ pc1 + pc2
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -385.89
## 2   4 -391.36 -5 10.941    0.05256 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + hybrid_status, data = lab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.414  -3.377   1.069   4.635  10.270
##
```

```
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      92.4051     1.0055  91.899   <2e-16 ***
## pc1                               0.4892     0.2588   1.890   0.0613 .
## pc2                              -0.4754     0.3539  -1.344   0.1818
## hybrid_statusF0 M. m. musculus   -0.8463     1.7017  -0.497   0.6199
## hybrid_statusF1 hybrid            3.7607     1.8839   1.996   0.0483 *
## hybrid_statusF1 M. m. domesticus -0.8354     2.4476  -0.341   0.7335
## hybrid_statusF1 M. m. musculus    3.3948     2.8480   1.192   0.2358
## hybrid_statusother               -2.5870     1.7010  -1.521   0.1311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.242 on 112 degrees of freedom
## Multiple R-squared:  0.1448, Adjusted R-squared:  0.09136
## F-statistic: 2.709 on 7 and 112 DF,  p-value: 0.01242
```

```
## [1] 789.7811
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + infection_history, data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3015  -3.4827   0.3916   2.9820  14.2944
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           90.03404    1.90753  47.199  < 2e-16
## pc1                                    0.07033    0.18085   0.389  0.69813
## pc2                                   -0.74707    0.28452  -2.626  0.00989
## infection_historyfalciformis_ferrisi   1.80628    2.26538   0.797  0.42698
## infection_historyfalciformis_uninfected  6.61142    2.31505   2.856  0.00514
## infection_historyferrisi_falciformis  -7.59036    2.51968  -3.012  0.00322
## infection_historyferrisi_ferrisi       2.98608    2.24216   1.332  0.18571
## infection_historyferrisi_uninfected    5.45674    2.12696   2.566  0.01166
## infection_historyuninfected            7.46027    2.56194   2.912  0.00436
## infection_historyuninfected_falciformis -4.54416    2.77132  -1.640  0.10395
## infection_historyuninfected_ferrisi   -2.79520    2.54972  -1.096  0.27537
##
## (Intercept)                           ***
## pc1
## pc2                                   **
## infection_historyfalciformis_ferrisi
## infection_historyfalciformis_uninfected **
## infection_historyferrisi_falciformis  **
## infection_historyferrisi_ferrisi
## infection_historyferrisi_uninfected   *
## infection_historyuninfected           **
## infection_historyuninfected_falciformis
## infection_historyuninfected_ferrisi
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.921 on 109 degrees of freedom
## Multiple R-squared:  0.4827, Adjusted R-squared:  0.4353
## F-statistic: 10.17 on 10 and 109 DF,  p-value: 6.437e-12
```

```
## [1] 735.4449
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2, data = lab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.220  -3.524   1.443   5.192  10.951
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  92.3222     0.5835 158.217   <2e-16 ***
## pc1           0.2736     0.2192   1.248   0.2143
## pc2          -0.8445     0.3357  -2.516   0.0132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.392 on 117 degrees of freedom
## Multiple R-squared:  0.06317,    Adjusted R-squared:  0.04715
## F-statistic: 3.944 on 2 and 117 DF,  p-value: 0.02199
```

```
##                    df      AIC
## weight_lm           6 742.6357
## weight_lm_exp_only  4 790.7223
```

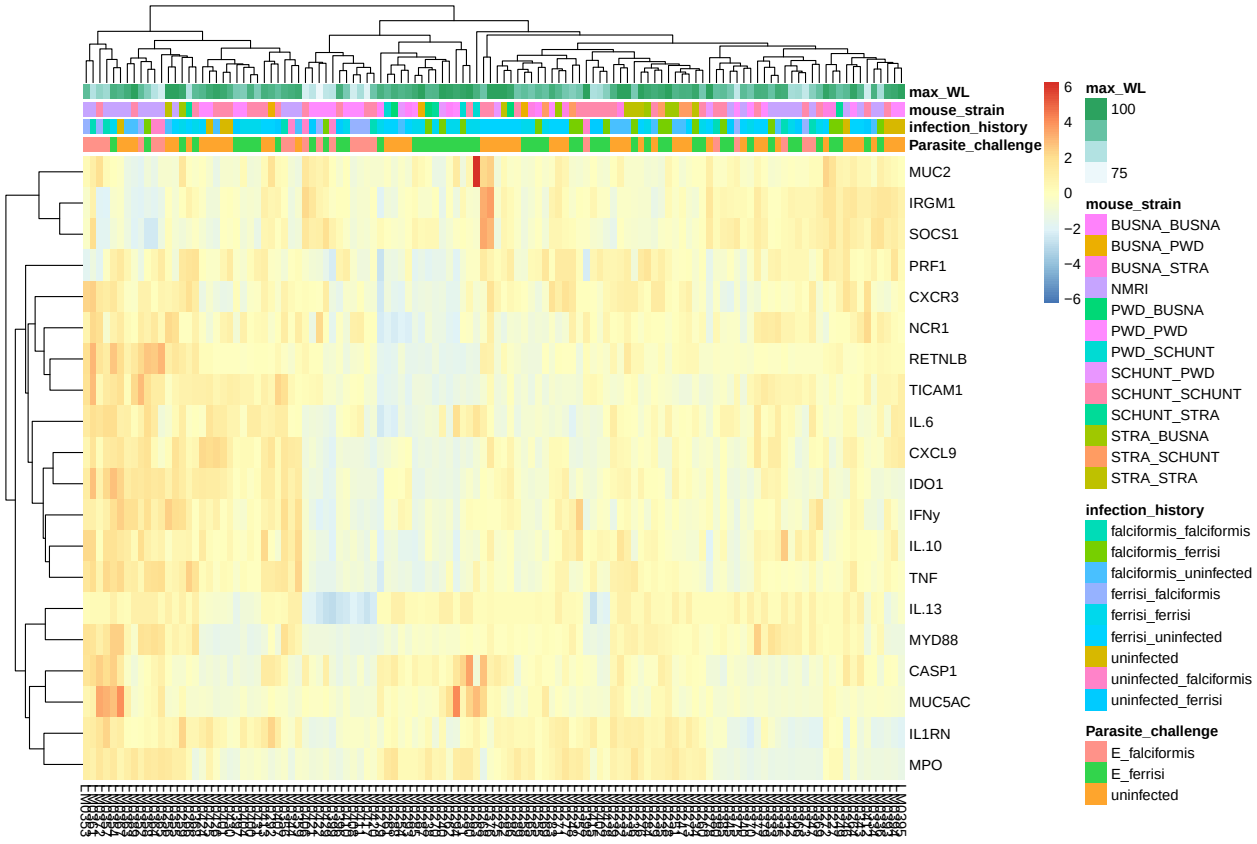
repeating the heatmap on the now imputed data

```
lab <- unique(lab)

gene <- lab %>%
  dplyr::select(c(Mouse_ID, Gene_lab_imp)) %>%
  rename_with(~str_remove(., '_imp'))

gene <- unique(gene)

 # turn the data frame into a matrix and transpose it. We want to have each cell
 # type as a row name
 gene <- t(as.matrix(gene))

 # turn the first row into column names
 gene %>%
     row_to_names(row_number = 1) -> heatmap_data

 heatmap_data <- as.data.frame(heatmap_data)

 table(rowSums(is.na(heatmap_data)) == nrow(heatmap_data))
```

```
##
## FALSE
##    20
```

```r
# turn the columns to numeric other wise the heatmap function will not work
 heatmap_data[] <- lapply(heatmap_data, function(x) as.numeric(as.character(x)))

 # remove columns with only NAs
 heatmap_data <- Filter(function(x)!all(is.na(x)), heatmap_data)

 #remove rows with only Nas
 heatmap_data <-  heatmap_data[, colSums(is.na(heatmap_data)) !=
                                 nrow(heatmap_data)]


#Prepare the annotation data frame
annotation_df <- as_tibble(lab) %>%
    dplyr::select(c("Mouse_ID", "Parasite_challenge", "infection_history",
                    "mouse_strain", "max_WL"))

annotation_df <- unique(annotation_df)

annotation_df <- as.data.frame(annotation_df)

 ### Prepare the annotation columns for the heatmap
rownames(annotation_df) <- annotation_df$Mouse_ID

# Match the row names to the heatmap data frame
rownames(annotation_df) <- colnames(heatmap_data)

#remove the unecessary column
annotation_df <- annotation_df %>% dplyr::select(-Mouse_ID, )
```

Heatmap on gene expression data:


max_WL
mouse_strain
infection_history
Parasite_challenge
MUC2
IRGM1
SOCS1
PRF1
CXCR3
NCR1
RETNLB
TICAM1
IL.6
CXCL9
IDO1
IFNy
IL.10
TNF
IL.13
MYD88
CASP1
MUC5AC
IL1RN
MPO
6
4
2
0
−2
−4
−6
max_WL
100
75
mouse_strain
BUSNA_BUSNA
BUSNA_PWD
BUSNA_STRA
NMRI
PWD_BUSNA
PWD_PWD
PWD_SCHUNT
SCHUNT_PWD
SCHUNT_SCHUNT
SCHUNT_STRA
STRA_BUSNA
STRA_SCHUNT
STRA_STRA
infection_history
falciformis_falciformis
falciformis_ferrisi
falciformis_uninfected
ferrisi_falciformis
ferrisi_ferrisi
ferrisi_uninfected
uninfected
uninfected_falciformis
uninfected_ferrisi
Parasite_challenge
E_falciformis
E_ferrisi
uninfected