Gene_expression_analysis

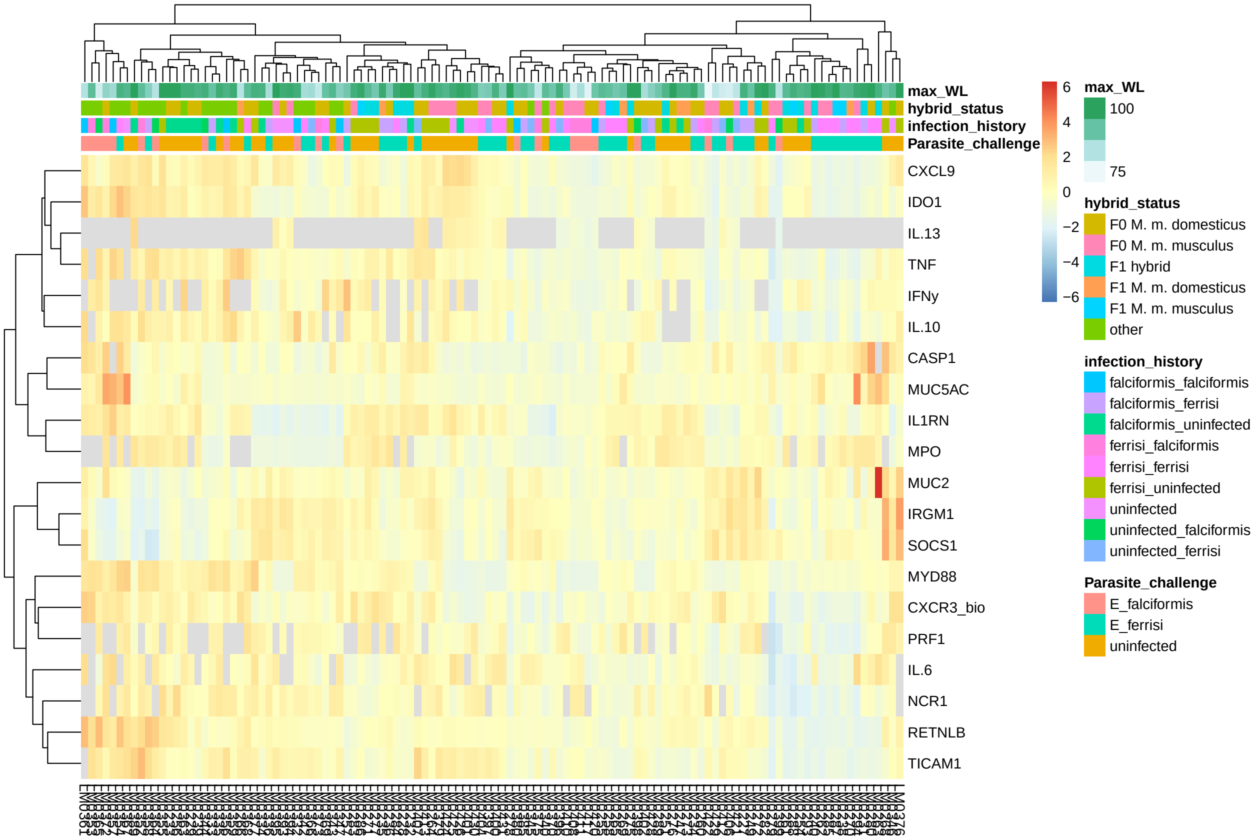
Fay

2022-05-18

1. Gene expression in the laboratory infections - Heatmap

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

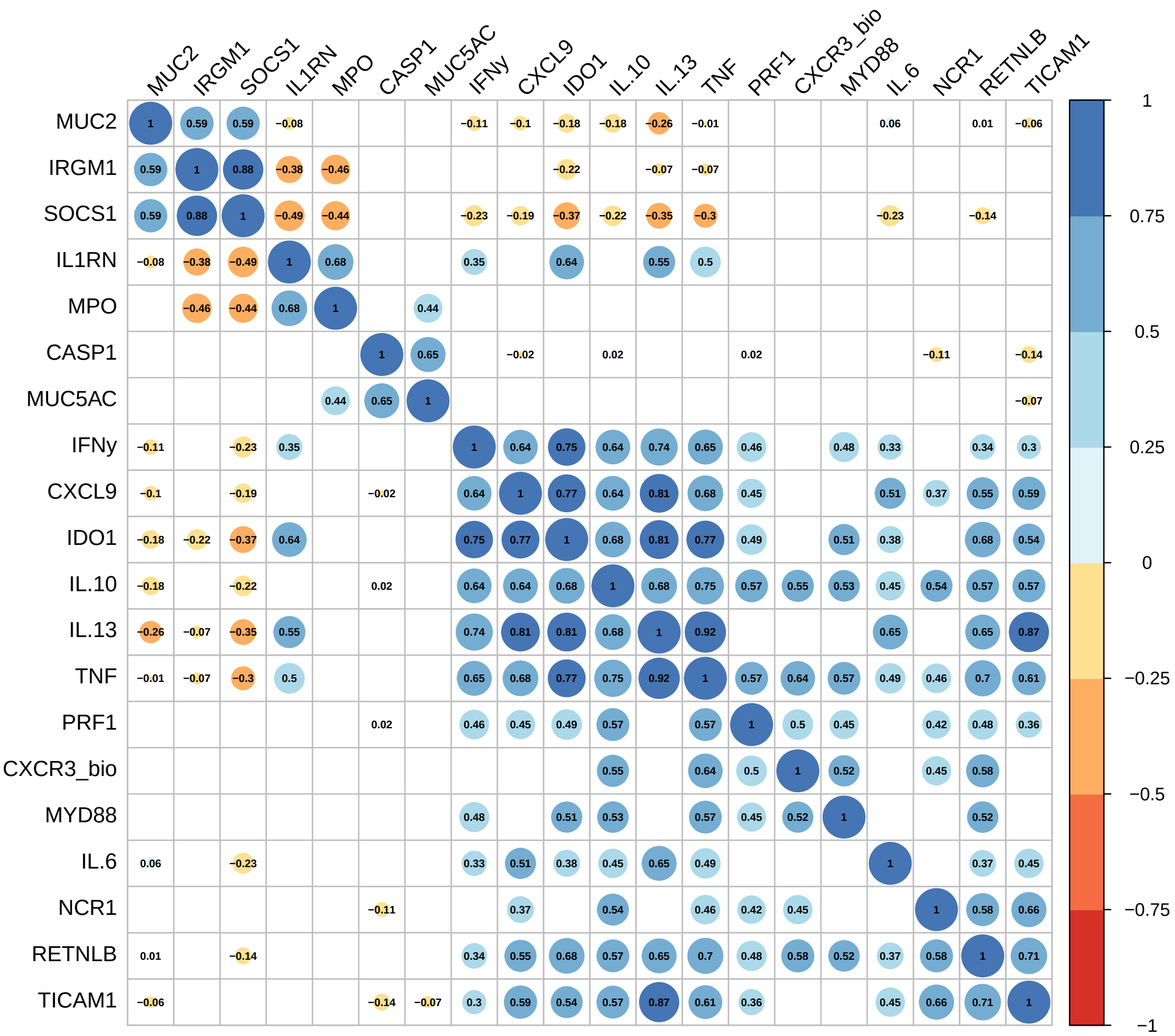
Heatmap on gene expression data:

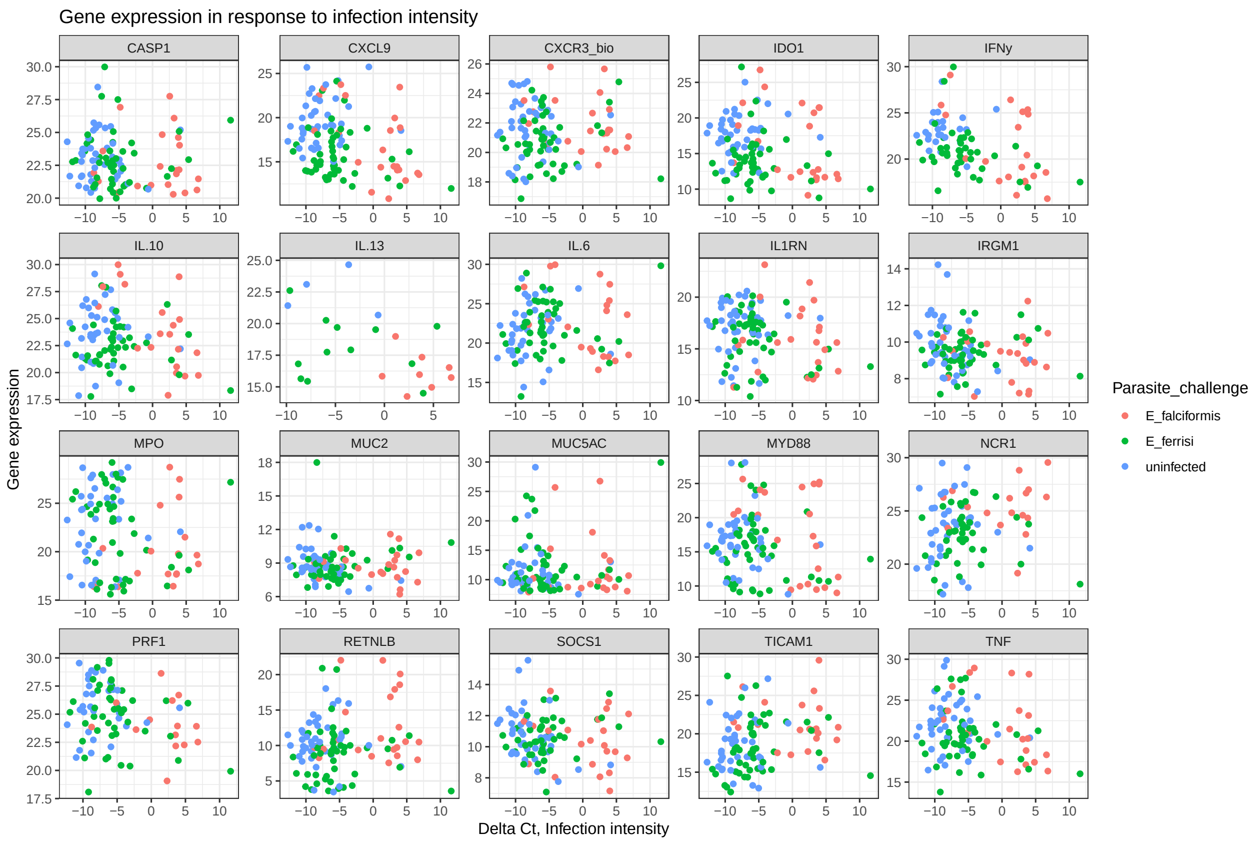


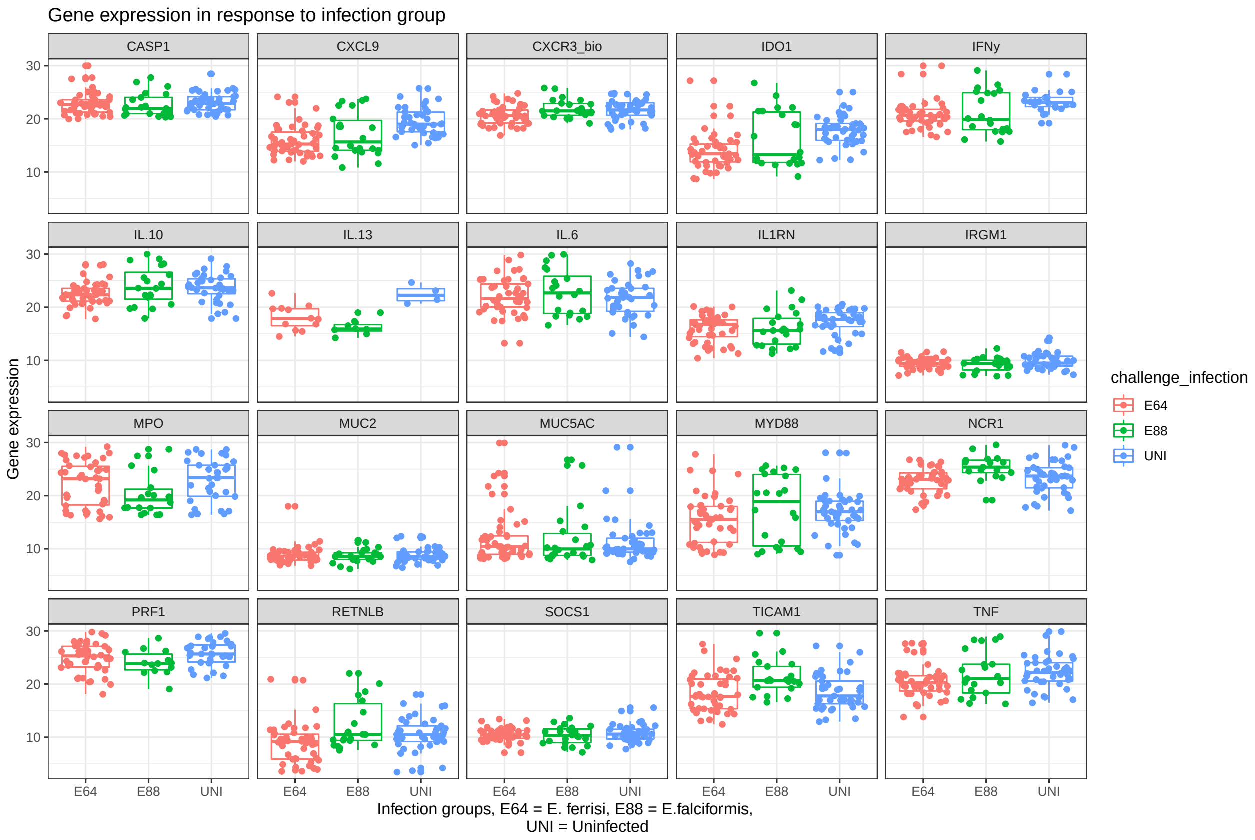
2. Correlations between the genes

Corrplot of correlations

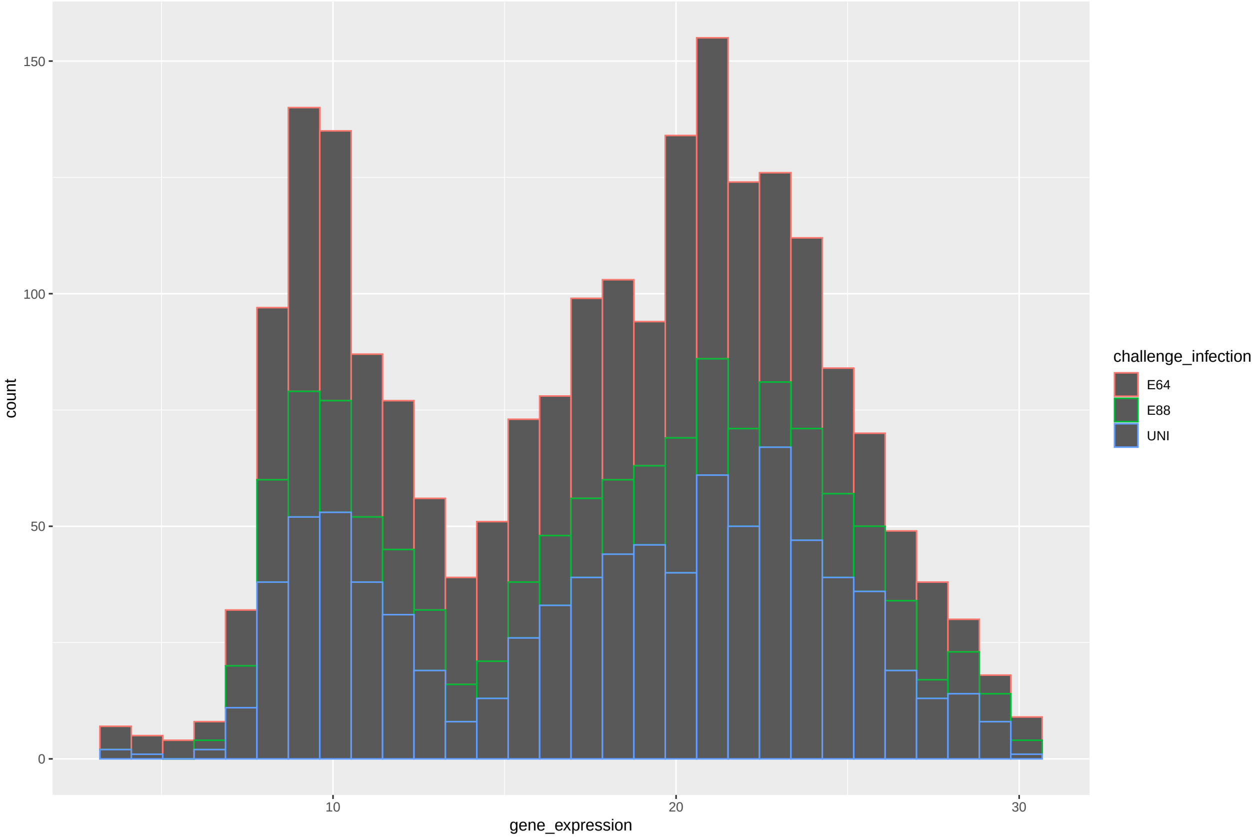
Here is a corrplot of the correlations between the genes. I am using the non-normalized genes


MUC2 IRGM1 SOCS1 IL1RN MPO CASP1 MUC5AC IFNy CXCL9 IDO1 IL.10 IL.13 TNF PRF1 CXCR3_bio MYD88 IL.6 NCR1 RETNLB TICAM1
MUC2 1 0.59 0.59 −0.08 −0.11 −0.1 −0.18 −0.18 −0.26 −0.01 0.06 0.01 −0.06
IRGM1 0.59 1 0.88 −0.38 −0.46 −0.22 −0.07 −0.07
SOCS1 0.59 0.88 1 −0.49 −0.44 −0.23 −0.19 −0.37 −0.22 −0.35 −0.3 −0.23 −0.14
IL1RN −0.08 −0.38 −0.49 1 0.68 0.35 0.64 0.55 0.5
MPO −0.46 −0.44 0.68 1 0.44
CASP1 1 0.65 −0.02 0.02 0.02 −0.11 −0.14
MUC5AC 0.44 0.65 1 −0.07
IFNy −0.11 −0.23 0.35 1 0.64 0.75 0.64 0.74 0.65 0.46 0.48 0.33 0.34 0.3
CXCL9 −0.1 −0.19 −0.02 0.64 1 0.77 0.64 0.81 0.68 0.45 0.51 0.37 0.55 0.59
IDO1 −0.18 −0.22 −0.37 0.64 0.75 0.77 1 0.68 0.81 0.77 0.49 0.51 0.38 0.68 0.54
IL.10 −0.18 −0.22 0.02 0.64 0.64 0.68 1 0.68 0.75 0.57 0.55 0.53 0.45 0.54 0.57 0.57
IL.13 −0.26 −0.07 −0.35 0.55 0.74 0.81 0.81 0.68 1 0.92 0.65 0.65 0.87
TNF −0.01 −0.07 −0.3 0.5 0.65 0.68 0.77 0.75 0.92 1 0.57 0.64 0.57 0.49 0.46 0.7 0.61
PRF1 0.02 0.46 0.45 0.49 0.57 0.57 1 0.5 0.45 0.42 0.48 0.36
CXCR3_bio 0.55 0.64 0.5 1 0.52 0.45 0.58
MYD88 0.48 0.51 0.53 0.57 0.45 0.52 1 0.52
IL.6 0.06 −0.23 0.33 0.51 0.38 0.45 0.65 0.49 1 0.37 0.45
NCR1 −0.11 0.37 0.54 0.46 0.42 0.45 1 0.58 0.66
RETNLB 0.01 −0.14 0.34 0.55 0.68 0.57 0.65 0.7 0.48 0.58 0.52 0.37 0.58 1 0.71
TICAM1 −0.06 −0.14 −0.07 0.3 0.59 0.54 0.57 0.87 0.61 0.36 0.45 0.66 0.71 1
1
0.75
0.5
0.25
0
−0.25
−0.5
−0.75
−1

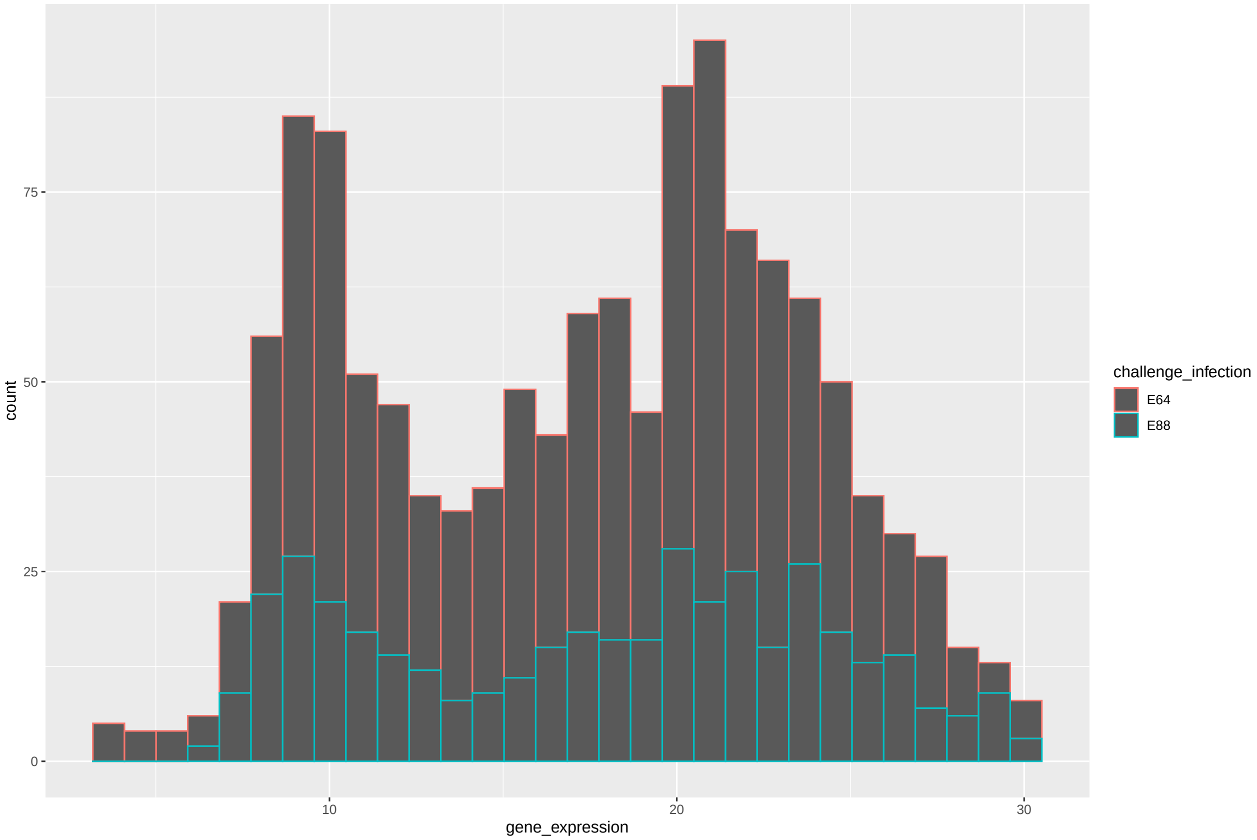




```
## Warning: Ignoring unknown parameters: echo
## Warning: Removed 186 rows containing non-finite values (stat_bin).
```



```
## Warning: Removed 97 rows containing non-finite values (stat_bin).
```

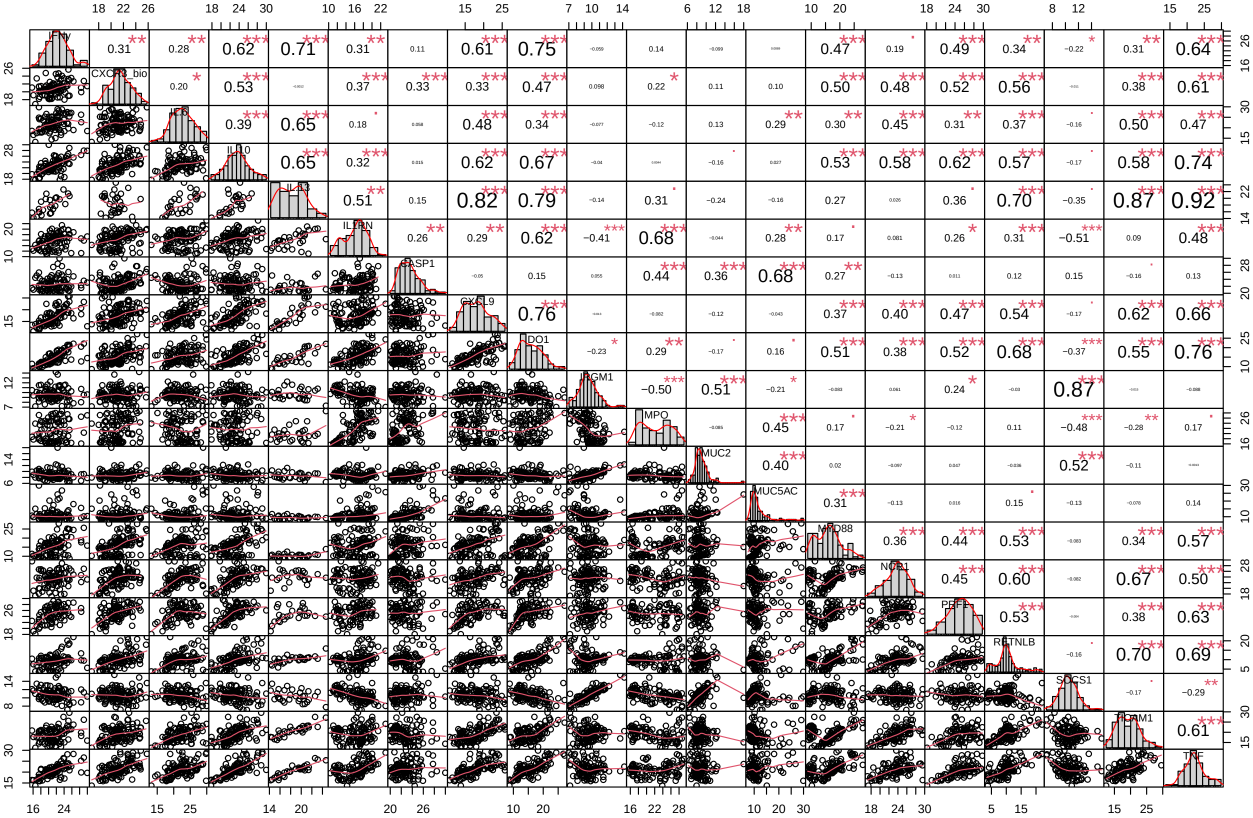


It is possible to compute a pca with missing data using the package missMDA. The missMDA package is dedicated to missing values in exploratory multivariate data analysis: single imputation/multiple imputation, etc.

Following the tutorial of the package author: Francois Husson: https://www.youtube.com/watch?v=OOM8_FH6_8o

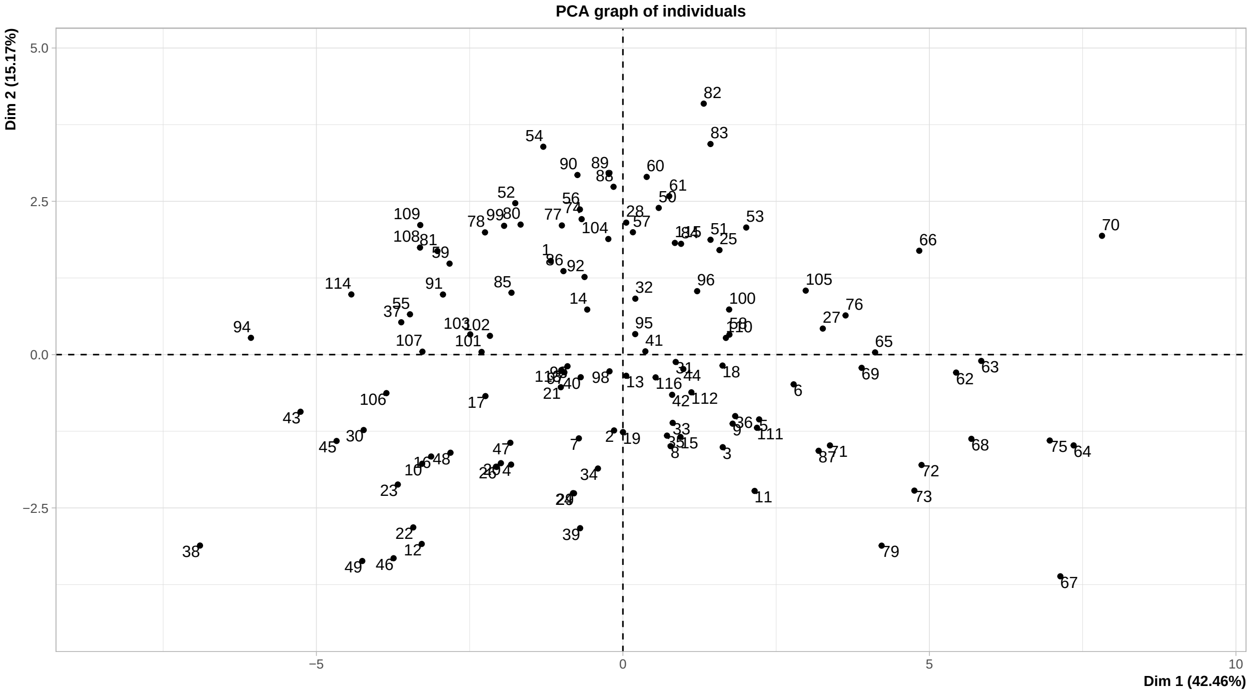
3. PCA

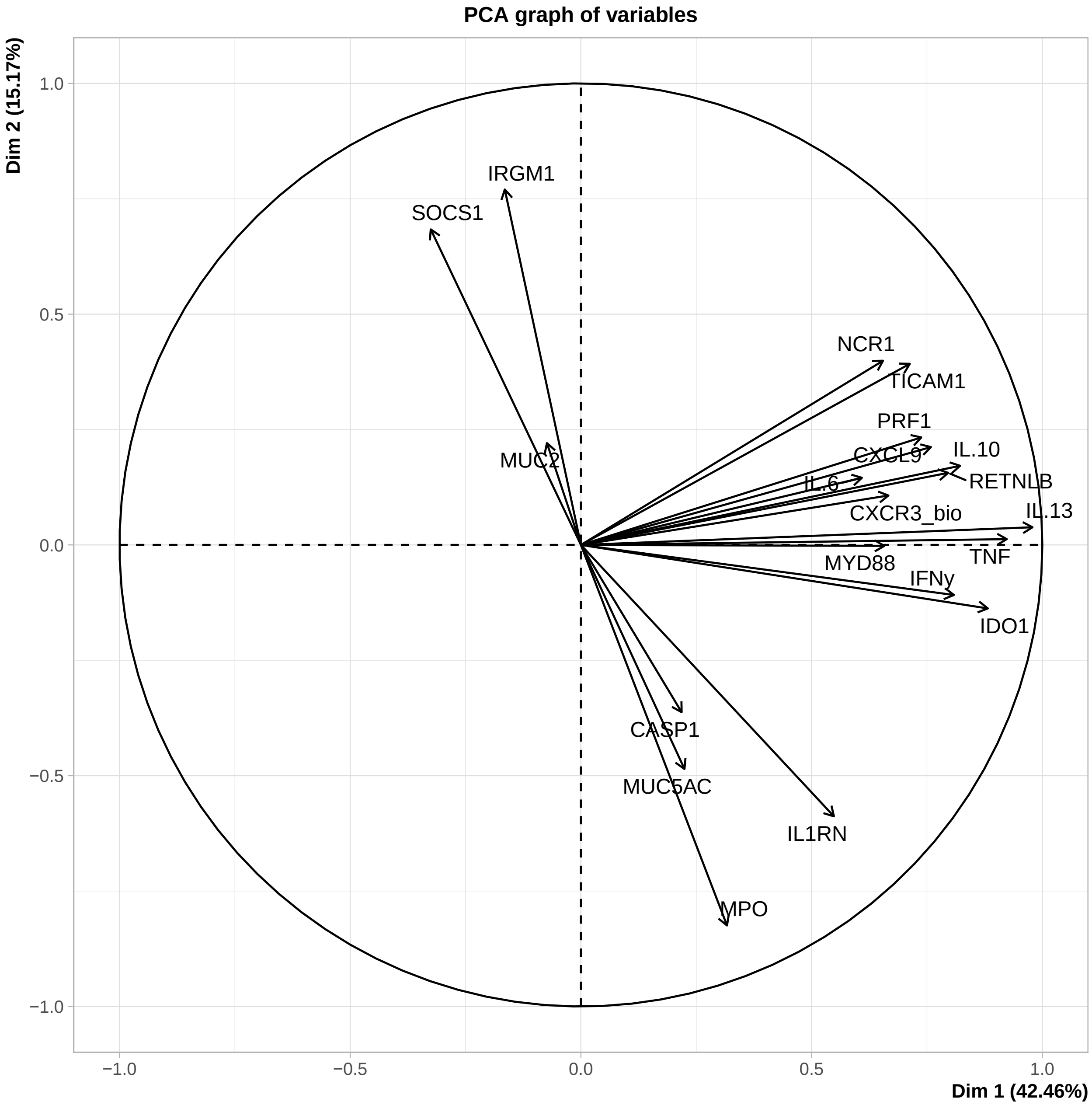
Handling missing data in a pca: Bad methods: removing individuals with missing data or replacing missing data with the mean (default setting in many packages).



We will now continue by using an iterative pca to impute missing data A. Initialization: impute using the mean B. Step lampda: # a. do pca on imputed data table S dimensions retained # b. missing data imputed using pca # c. means (and standard deviations) updated C. Iterate the estimation and imputation steps (until convergence) (convergence: the act of converging and especially moving toward union or uniformity)

Overfitting is a common problem due to believing too much in links between variables. –> regularized iterative PCA (This version is what is being implented in missMDA) This is a way of taking less risk when imputing the missing data. The algorithm estimates the missing data values with values that have no influence on the PCA results, i.e., no influence on the coordinates of the individuals or variables.


PCA graph of individuals
Dim 2 (15.17%)
Dim 1 (42.46%)
5.0
2.5
0.0
−2.5
−5
0
5
10
82
83
54
90
89
88
60
61
52
56
50
109
99
80
77
74
28
57
53
78
104
51
25
108
81
59
1
86
92
66
70
114
91
85
96
105
32
14
100
76
55
37
103
102
95
27
94
107
101
41
65
13
98
40
21
116
44
18
69
62
63
6
106
17
42
112
43
36
5
111
9
33
30
45
7
2
19
35
15
8
3
87
71
68
75
64
47
48
16
10
28
34
72
23
11
73
29
22
39
12
38
49
46
79
67



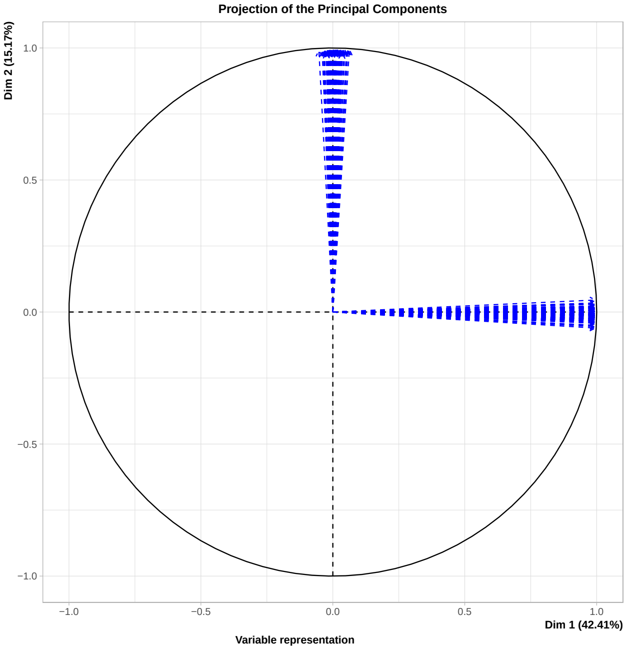
Caution: When imputing data, the percentages of inertia associated with the first dimensions will be overestimated.

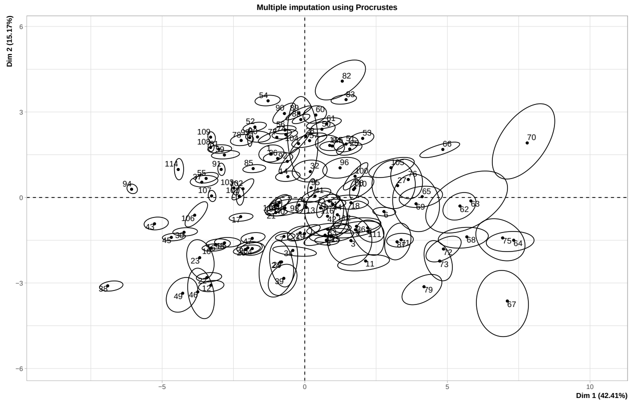
Another problem: the imputed data are, when the pca is performed considered like real observations. But they are estimations!!

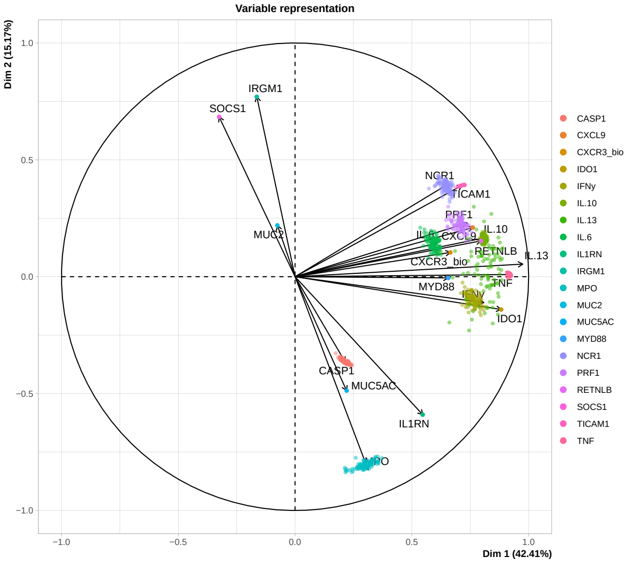
Visualizing uncertainty due to issing data:

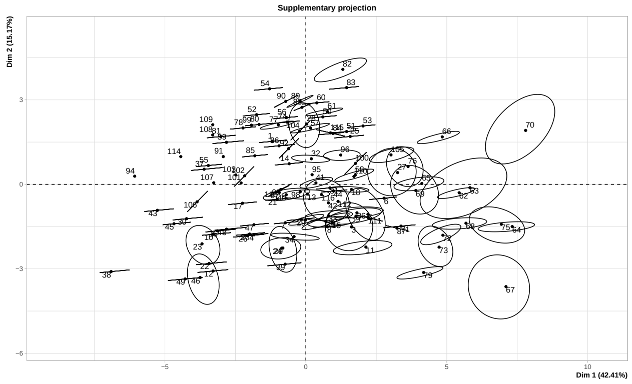
–> mulrimple imputation: generate several plausible values for each missing data point

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

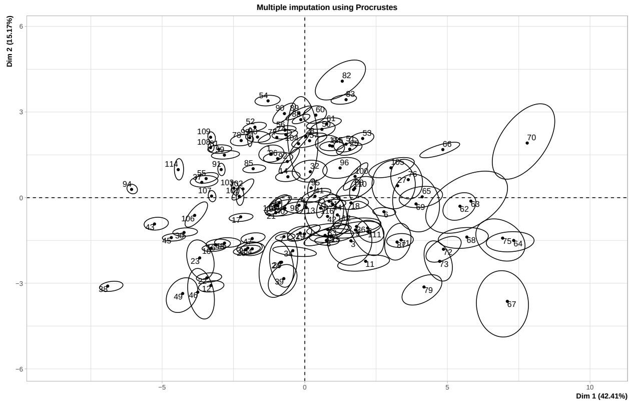






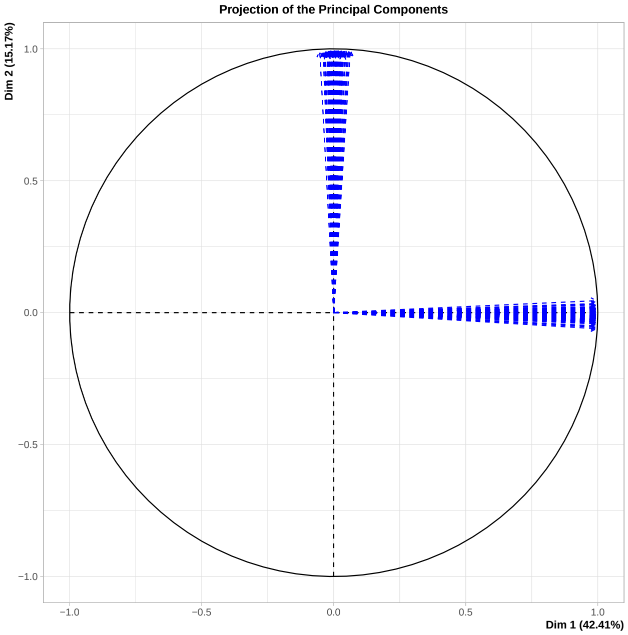


```
## $PlotIndProc
```

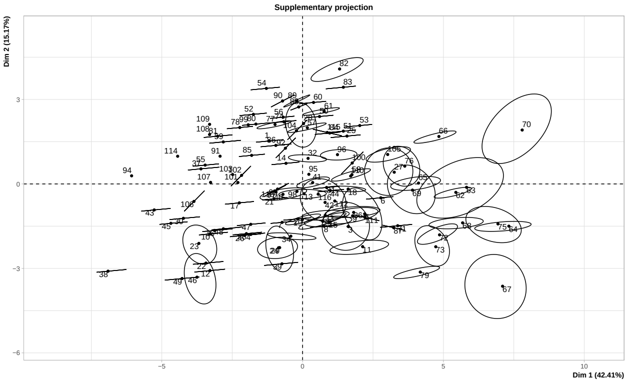


```
##
```

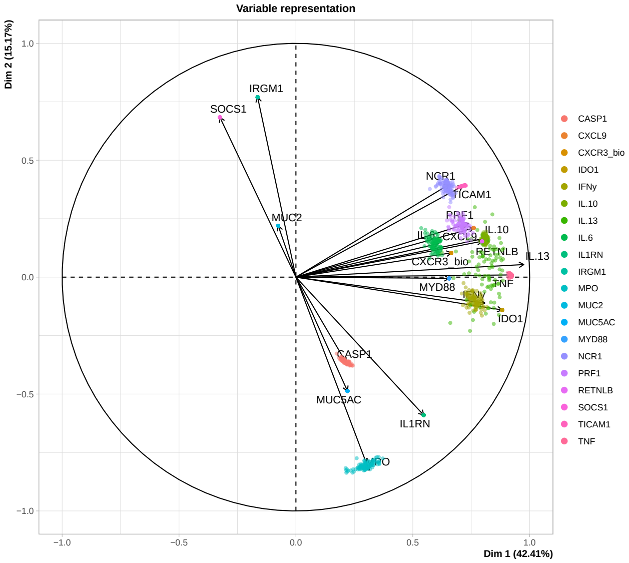
```
## $PlotDim
```



```
##
## $PlotIndSupp
```



```
##
## $PlotVar
```



```
## Help on topic 'plot' was found in the following packages:
##
##   Package               Library
##   base                  /usr/lib/R/library
##   graphics              /usr/lib/R/library
##
##
## Using the first match ...
```

Individuals lying on the axis have no missing data, but individuals that far away have many missing data. big ellipse = big uncertainty tight elipse (line) = low uncertainty

Variable representation: Poins tight together)look like one) - have no missing variables –> low uncertainty Points spread – > higher variability – > higher uncertainty

High uncertainty–> we should interpret the result with care

The individuals with many missing data values make the axes move, and thus the positions of all individuals

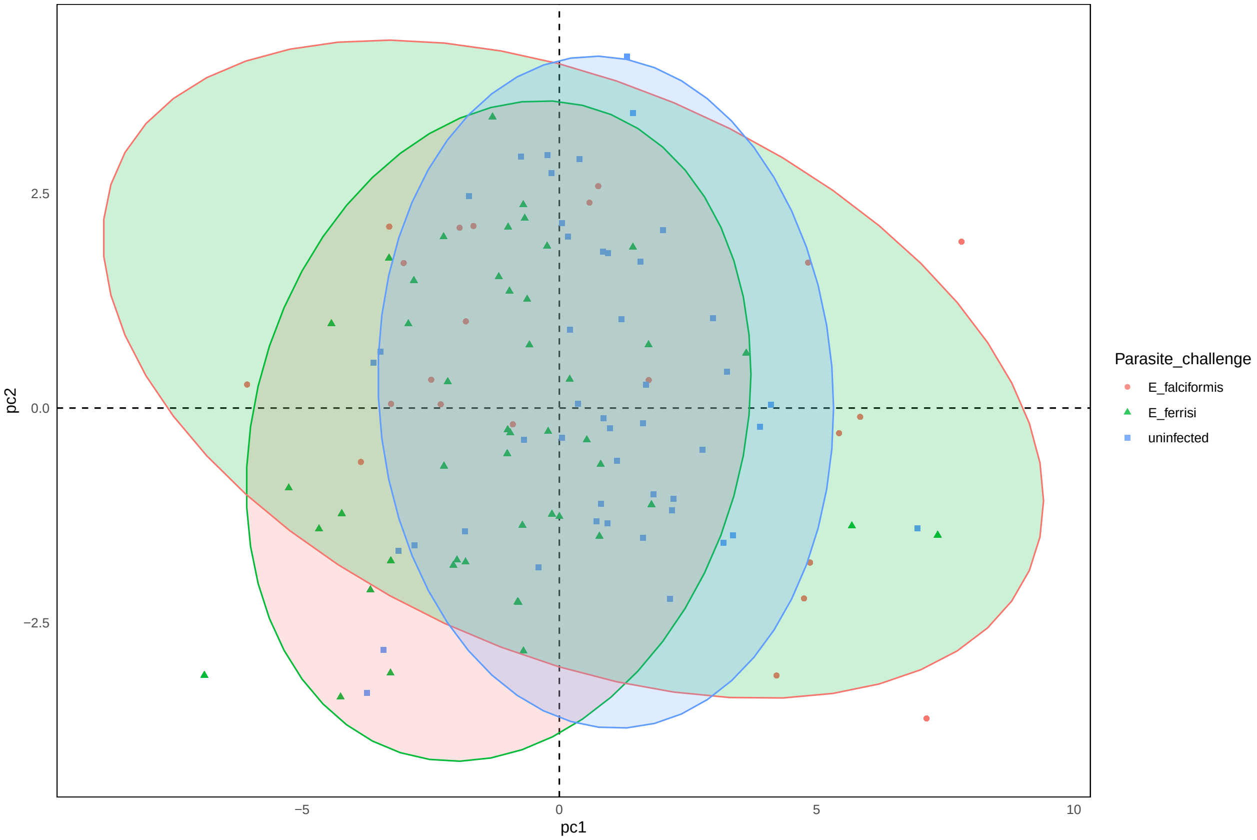
Therefore in the last plots every individual is getting an eclipse as they are as well influenced by the missing data of the others.

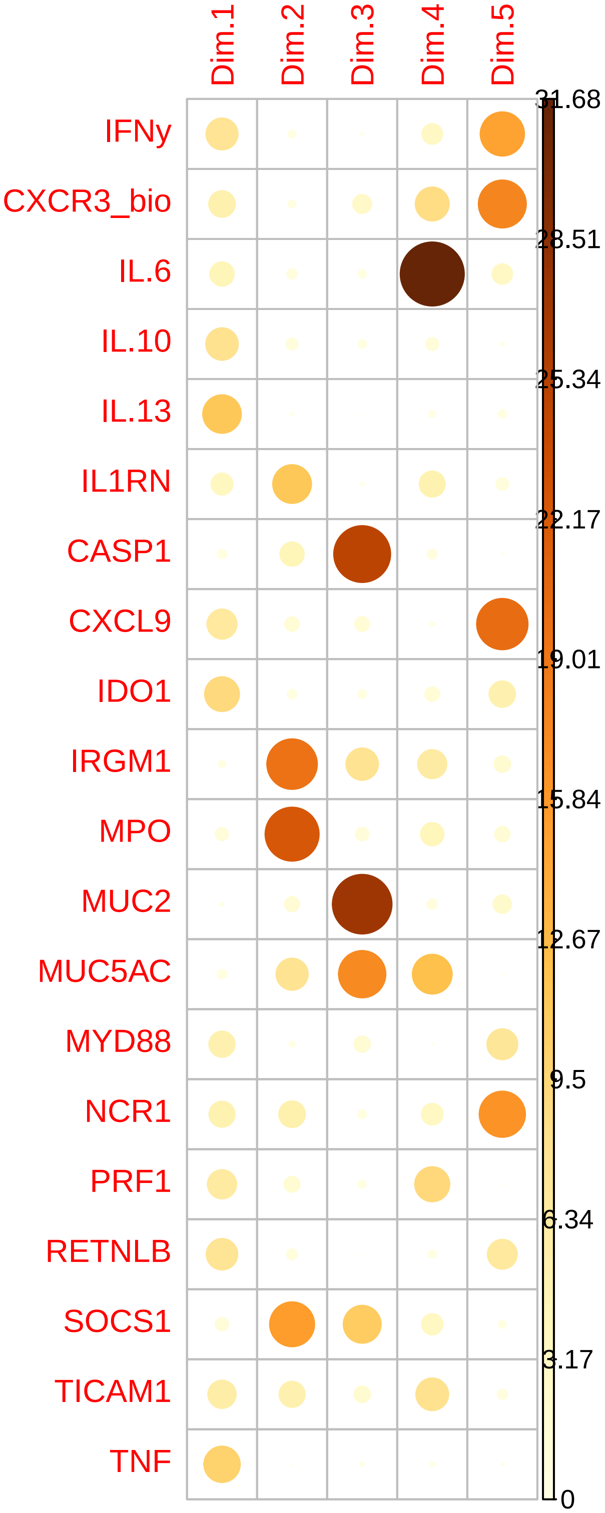
THe plot with the dimensions shows the projections of the pca dimensions of each imputed table on the pca plane obtained using the original imputed data table

As all of the arrows are close to either the first or second axes, this means that the axes are stable with respect to the set of imputed tables –> we don't have evidence of instability here.

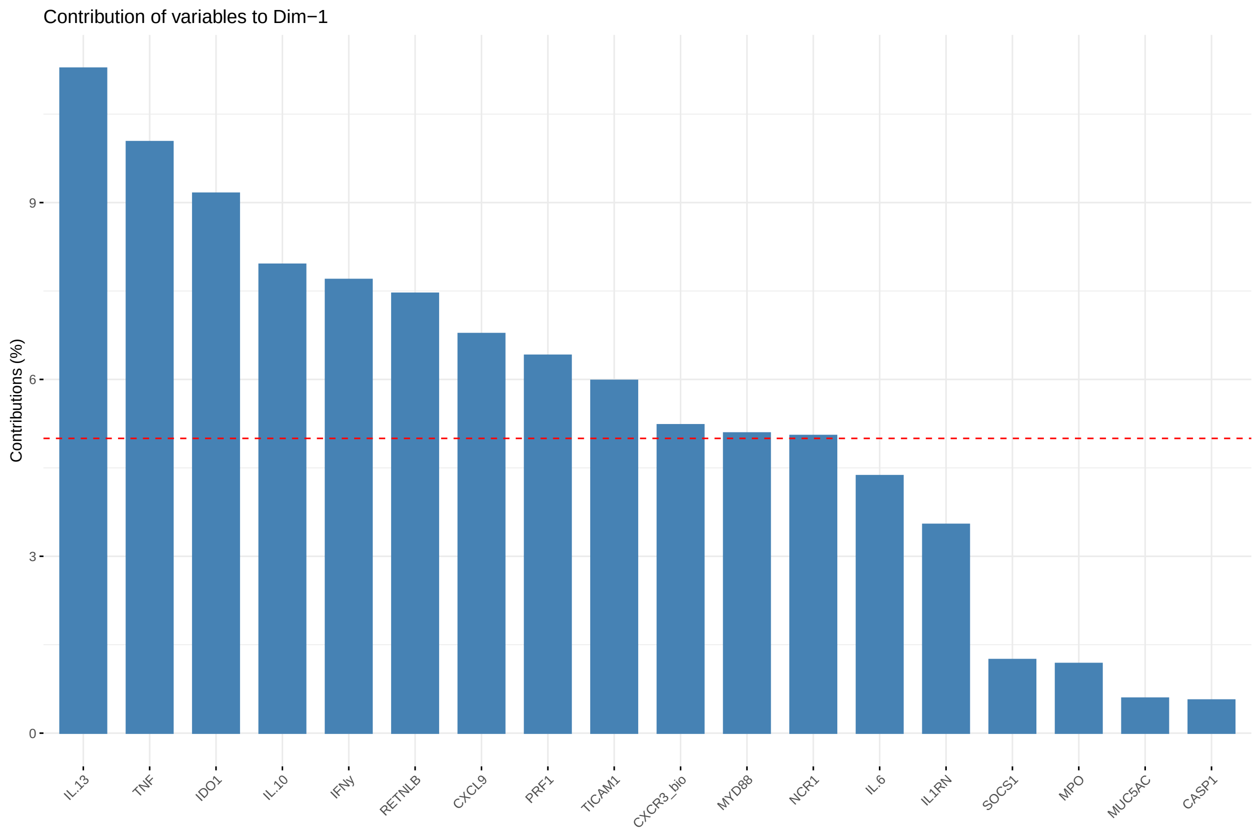
Biplot of the imputed gene pca

```r
#Now we can make our initial plot of the PCA.
imputed_expr %>%
  pivot_longer(cols = 12:31, names_to = "Gene", values_to = "gene_expression")  %>%
  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"))
```

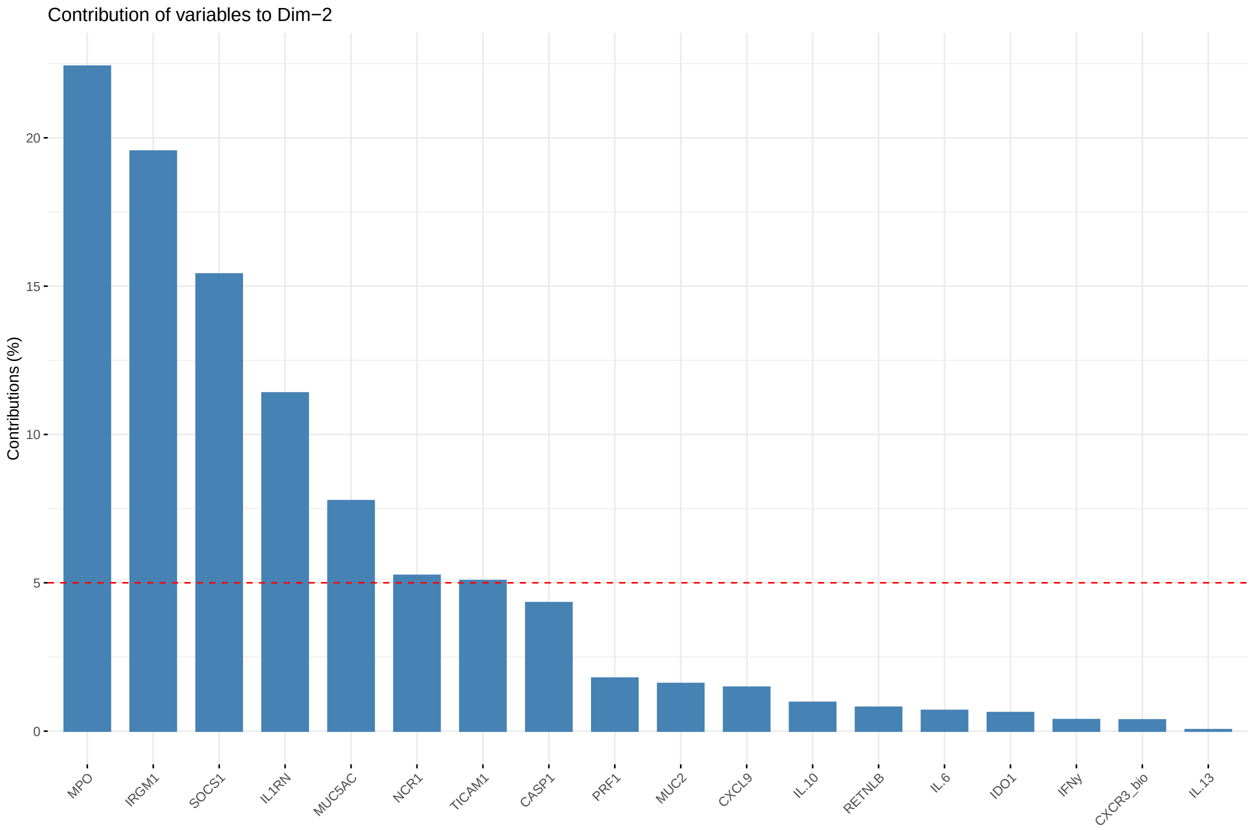


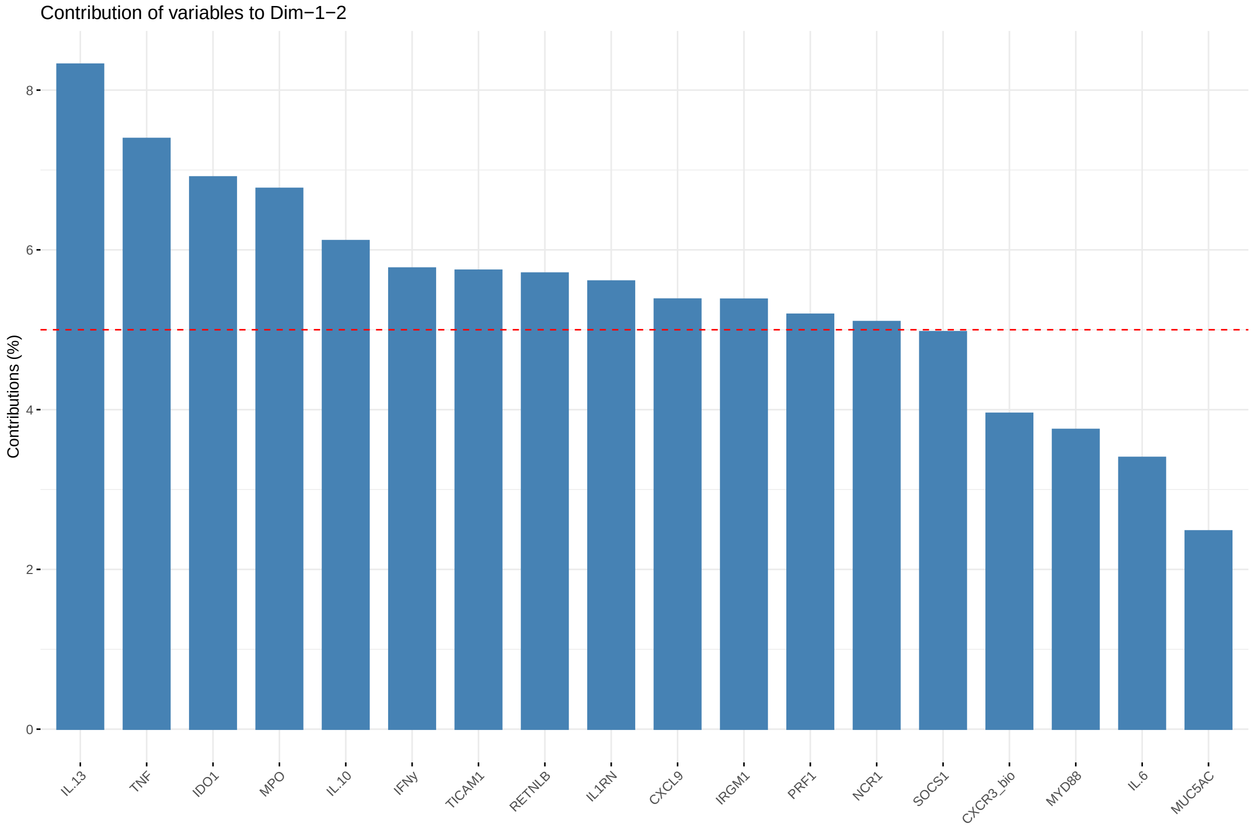


The function fviz_contrib() [factoextra package] can be used to draw a bar plot of variable contributions. If your data contains many variables, you can decide to show only the top contributing variables. The R code below shows the top 10 variables contributing to the principal components:



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

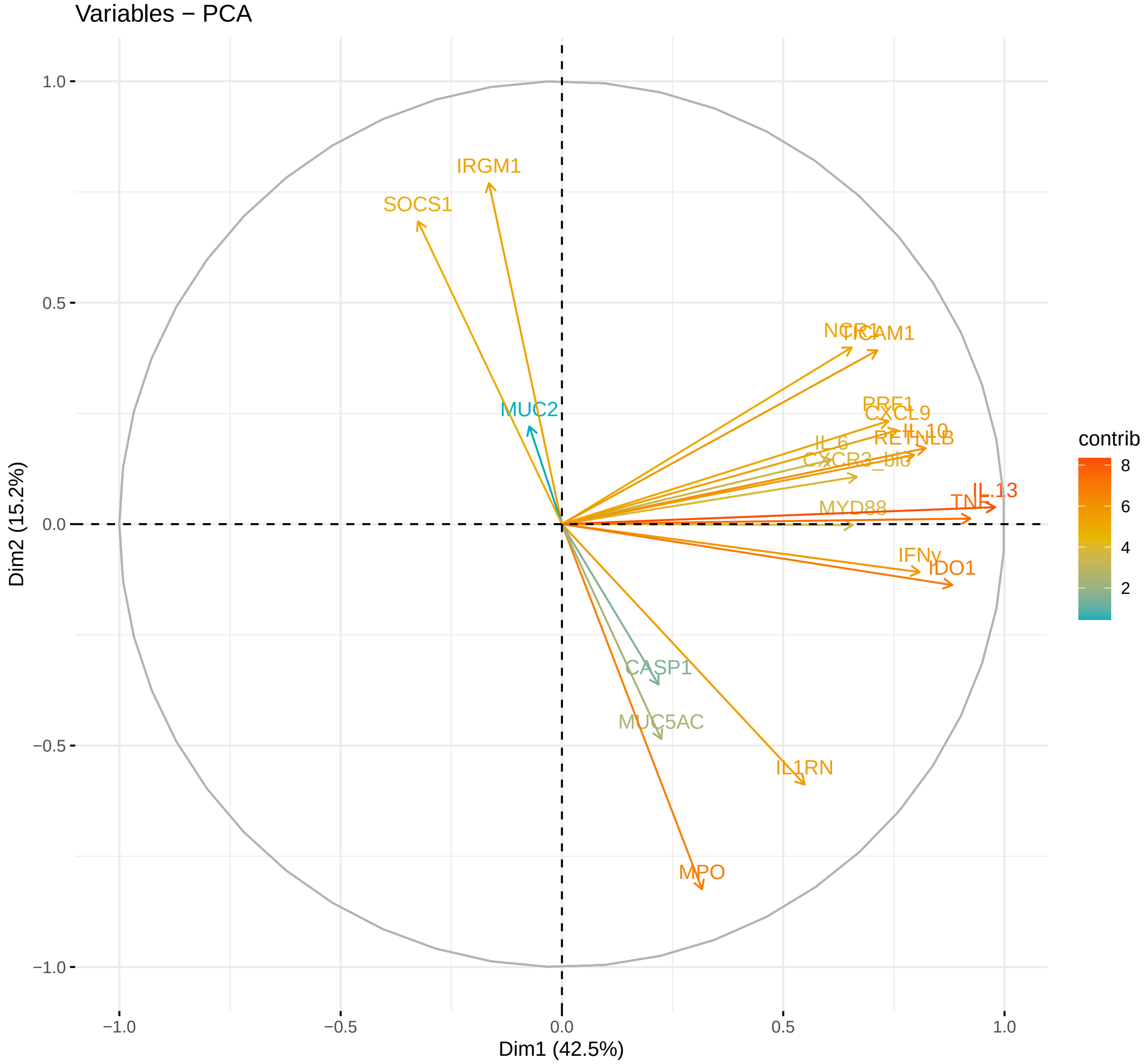




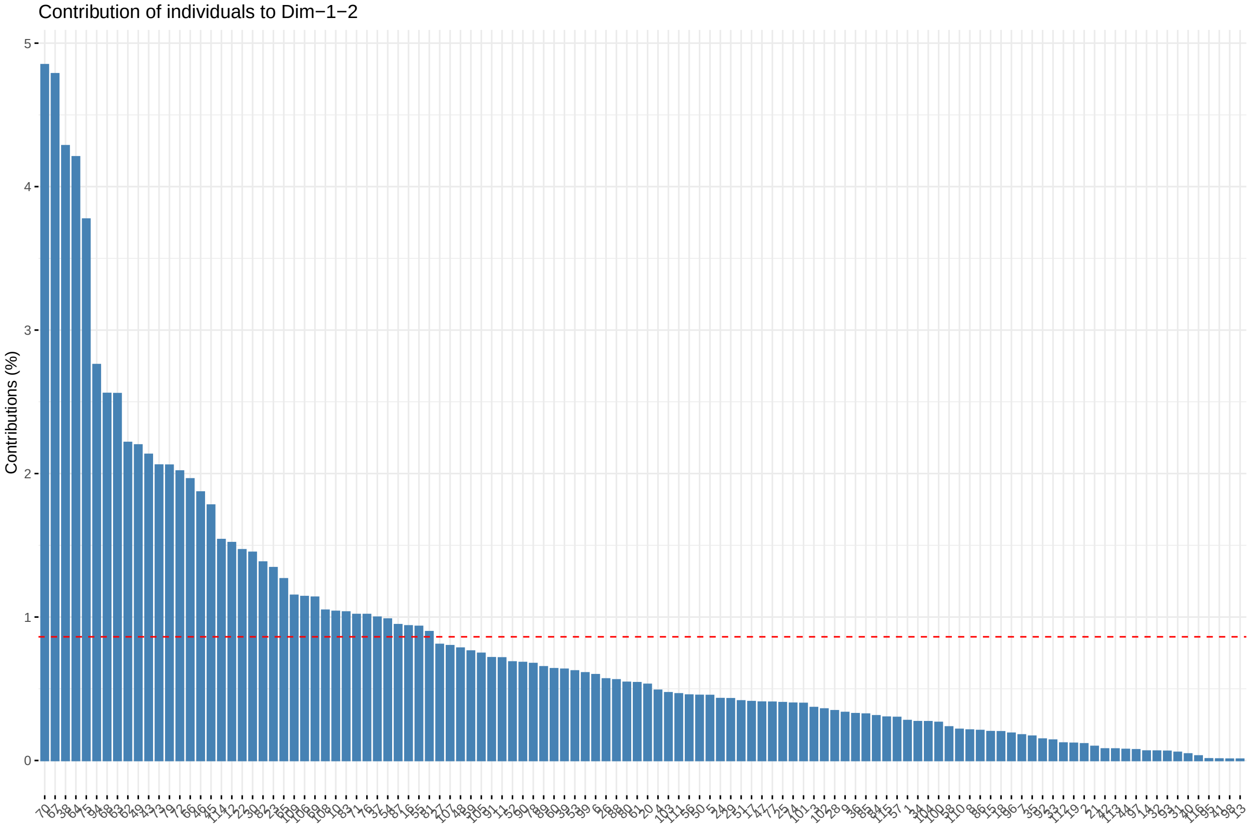
The red dashed line on the graph above indicates the expected average contribution. If the contribution of the variables were uniform, the expected value would be 1/length(variables) = 1/10 = 10%. For a given component, a variable with a contribution larger than this cutoff could be considered as important 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

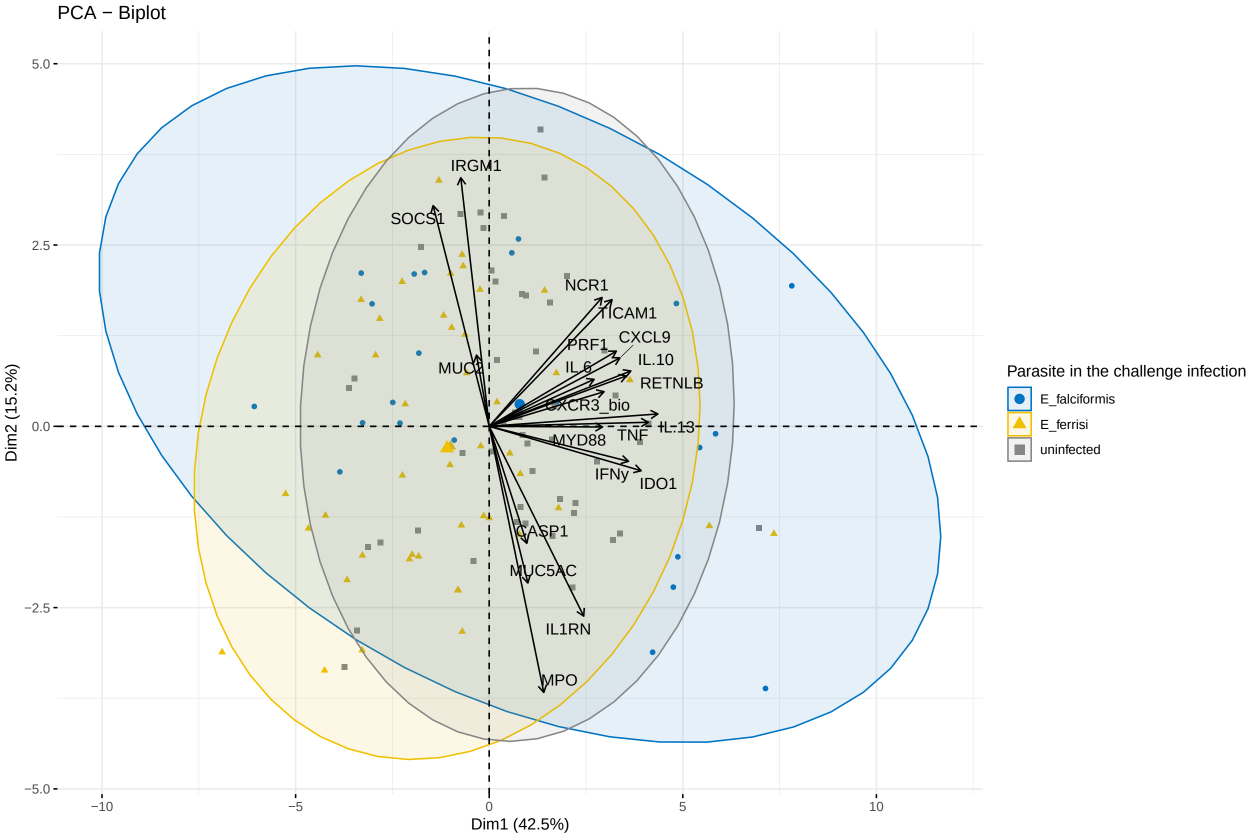
C1 and C2 are the contributions of the variable on PC1 and PC2, respectively Eig1 and Eig2 are the eigenvalues of PC1 and PC2, respectively. Recall that eigenvalues measure the amount of variation retained by each PC. In this case, the expected average contribution (cutoff) is calculated as follow: As 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 = imputed_expr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4014  -3.0944   0.1175   3.5262  14.3050
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   85.6719     1.1323  75.664  < 2e-16 ***
## pc1                            0.1272     0.1762   0.722   0.4718
## pc2                           -0.7278     0.2834  -2.568   0.0116 *
## Parasite_challengeE_ferrisi    6.0895     1.4092   4.321 3.40e-05 ***
## Parasite_challengeuninfected  10.4534     1.3576   7.700 6.09e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.253 on 111 degrees of freedom
## Multiple R-squared:  0.3818, Adjusted R-squared:  0.3596
## F-statistic: 17.14 on 4 and 111 DF,  p-value: 5.657e-11
```

```
## [1] 720.9133
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + Parasite_challenge + hybrid_status,
##     data = imputed_expr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7141  -3.5997   0.4672   3.5380  13.9501
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       86.0601     1.3838  62.193  < 2e-16 ***
## pc1                                0.1364     0.2199   0.620   0.5364
## pc2                               -0.5959     0.3144  -1.896   0.0607 .
## Parasite_challengeE_ferrisi        5.9059     1.4538   4.062 9.34e-05 ***
## Parasite_challengeuninfected      10.0684     1.4516   6.936 3.30e-10 ***
## hybrid_statusF0 M. m. musculus    -1.1985     1.4579  -0.822   0.4129
## hybrid_statusF1 hybrid             1.4620     1.6821   0.869   0.3867
## hybrid_statusF1 M. m. domesticus  -1.7765     2.2126  -0.803   0.4238
## hybrid_statusF1 M. m. musculus     1.7684     2.6843   0.659   0.5115
## hybrid_statusother                -0.3591     1.4859  -0.242   0.8095
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.288 on 106 degrees of freedom
## Multiple R-squared:  0.4017, Adjusted R-squared:  0.3509
## F-statistic: 7.907 on 9 and 106 DF,  p-value: 7.337e-09

## [1] 727.1249

##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + hybrid_status, data = imputed_expr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.942  -3.138   0.991   4.739   9.868
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      92.5229     1.0319  89.664   <2e-16 ***
## pc1                               0.3934     0.2496   1.576   0.1179
## pc2                              -0.3243     0.3701  -0.876   0.3827
## hybrid_statusF0 M. m. musculus    -1.1490     1.7436  -0.659   0.5113
## hybrid_statusF1 hybrid            3.7568     1.9749   1.902   0.0598 .
## hybrid_statusF1 M. m. domesticus  -0.3187     2.6314  -0.121   0.9038
## hybrid_statusF1 M. m. musculus     3.9912     3.1916   1.251   0.2138
## hybrid_statusother               -2.5944     1.7376  -1.493   0.1383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.332 on 108 degrees of freedom
## Multiple R-squared:  0.1259, Adjusted R-squared:  0.06928
## F-statistic: 2.223 on 7 and 108 DF,  p-value: 0.03774

## [1] 767.095

##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + infection_history, data = imputed_expr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4909  -3.4963   0.2167   3.0235  14.3776
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             90.04699    1.95916  45.962  < 2e-16
## pc1                                      0.02434    0.17334   0.140  0.88862
## pc2                                     -0.61060    0.29415  -2.076  0.04035
## infection_historyfalciformis_ferrisi     2.11290    2.34416   0.901  0.36947
## infection_historyfalciformis_uninfected  6.75252    2.36949   2.850  0.00527
## infection_historyferrisi_falciformis    -7.57150    2.57368  -2.942  0.00401
## infection_historyferrisi_ferrisi         2.81356    2.33435   1.205  0.23080
## infection_historyferrisi_uninfected      5.45339    2.19195   2.488  0.01442
## infection_historyuninfected              7.17664    2.60969   2.750  0.00702
## infection_historyuninfected_falciformis -4.51846    2.83429  -1.594  0.11389
## infection_historyuninfected_ferrisi     -2.60534    2.67011  -0.976  0.33144
##
## (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: 5.021 on 105 degrees of freedom
## Multiple R-squared:  0.4656, Adjusted R-squared:  0.4147 
## F-statistic: 9.149 on 10 and 105 DF,  p-value: 1.007e-10
```

```
## [1] 716.0163
```

```
## 
## Call:
## lm(formula = max_WL ~ pc1 + pc2, data = g)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -16.913  -3.236   1.379   5.127  10.471 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  92.3746     0.6006 153.811   <2e-16 ***
## pc1           0.1702     0.2061   0.826   0.4107    
## pc2          -0.7501     0.3448  -2.175   0.0317 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.468 on 113 degrees of freedom
## Multiple R-squared:  0.04572,    Adjusted R-squared:  0.02883 
## F-statistic: 2.707 on 2 and 113 DF,  p-value: 0.07108
```

```
##                    df      AIC
## weight_lm           6 720.9133
## weight_lm_exp_only  4 767.2808
```

repeating the heatmap on the now imputed data

```
gene <- imputed_expr %>% dplyr::select(c(EH_ID, all_of(Genes)))

 # 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))

 #switch the matrix back to a data frame format
 gene <- as.data.frame(gene)

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


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

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

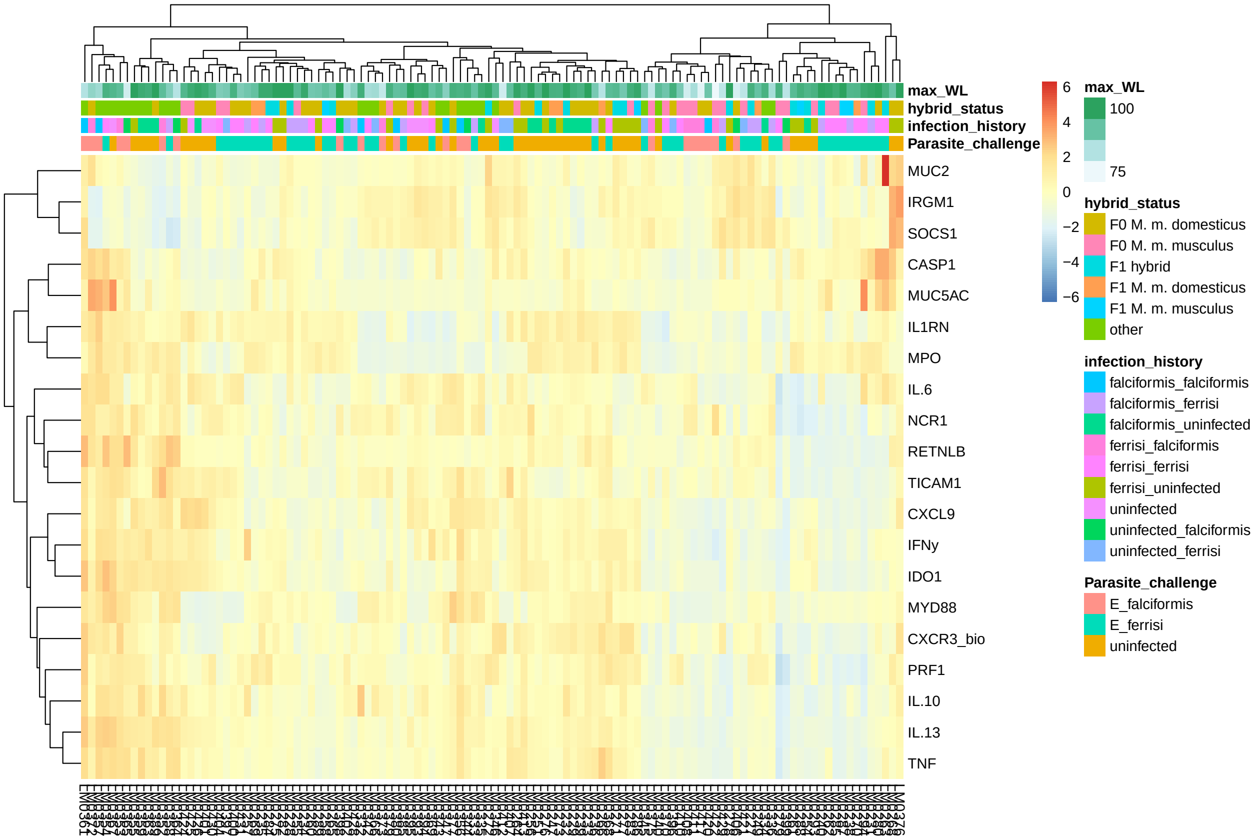
```
# turn the columns to numeric other wise the heatmap function will not work
heatmap_data[] <- lapply(heatmap_data, function(x) as.numeric(as.character(x)))

# 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)]

rownames(annotation_df) <- colnames(heatmap_data)
```

Heatmap on gene expression data:


max_WL
hybrid_status
infection_history
Parasite_challenge
MUC2
IRGM1
SOCS1
CASP1
MUC5AC
IL1RN
MPO
IL.6
NCR1
RETNLB
TICAM1
CXCL9
IFNy
IDO1
MYD88
CXCR3_bio
PRF1
IL.10
IL.13
TNF
max_WL
hybrid_status
F0 M. m. domesticus
F0 M. m. musculus
F1 hybrid
F1 M. m. domesticus
F1 M. m. musculus
other
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