PCA analysis

Marc Galland

2021-07-27

Contents

1 Data import

Two datasets:
1. 793 metabolites' peak area detected by LC-MS from 23 different plants from 4 different genotypes.
2. Sample to genotype correspondence.

1.1 Peaks

```
peaks <- read.csv("../genotype_and_peak_data.csv",
                  stringsAsFactors = F)
# The first five rows and five columns of the `peaks` dataframe.
knitr::kable(peaks[1:5,1:5])
```

metabolite_1	metabolite_2	metabolite_3	metabolite_4	metabolite_5
4.22e-05	0.0003852	0.0007782	0.0014183	0.0028457
2.66e-05	0.0008488	0.0008215	0.0006727	0.0025750
5.69e-05	0.0011296	0.0007667	0.0001023	0.0024471
5.61e-05	0.0011317	0.0007291	0.0008141	0.0023510
8.59e-05	0.0017653	0.0000578	0.0000805	0.0042726

1.2 Sample to genotype

```
sample_info <- read.csv("../sample_genotype_phenotype.csv",
                        stringsAsFactors = F)
knitr::kable(sample_info)
```

sample	genotype	phenotype
IL27_1	IL1927	resistant
IL27_2	IL1927	resistant
IL27_4	IL1927	resistant
IL27_5	IL1927	resistant
IL27_6	IL1927	resistant
IL28_1	IL1928	sensitive
IL28_2	IL1928	sensitive
IL28_3	IL1928	sensitive
IL28_4	IL1928	sensitive
IL28_5	IL1928	sensitive
IL28_6	IL1928	sensitive
IL55_1	KG1955	sensitive
IL55_2	KG1955	sensitive
IL55_3	KG1955	sensitive
IL55_4	KG1955	sensitive
IL55_5	KG1955	sensitive
IL55_6	KG1955	sensitive
s_ch_1	LA1840	resistant
s_ch_2	LA1840	resistant
s_ch_3	LA1840	resistant
s_ch_4	LA1840	resistant
s_ch_5	LA1840	resistant
s_ch_6	LA1840	resistant

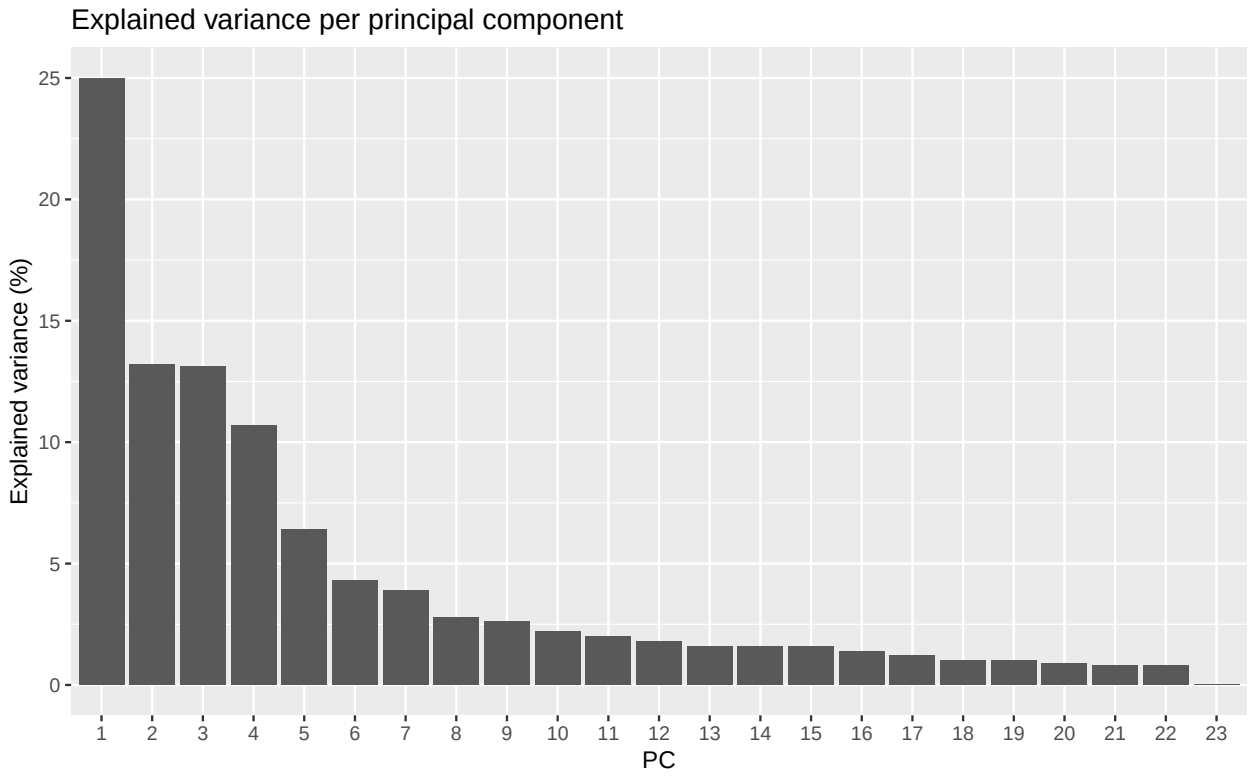
2 PCA analysis (all samples)

The PCA analysis is computed with the `mypca` function that returns: - sample scores - variable loadings - percentage of explained variance by each principal component (PC)

```
pca_results <- mypca(peaks, center = TRUE, scale = TRUE)
```

2.1 Scree plot: variance explained

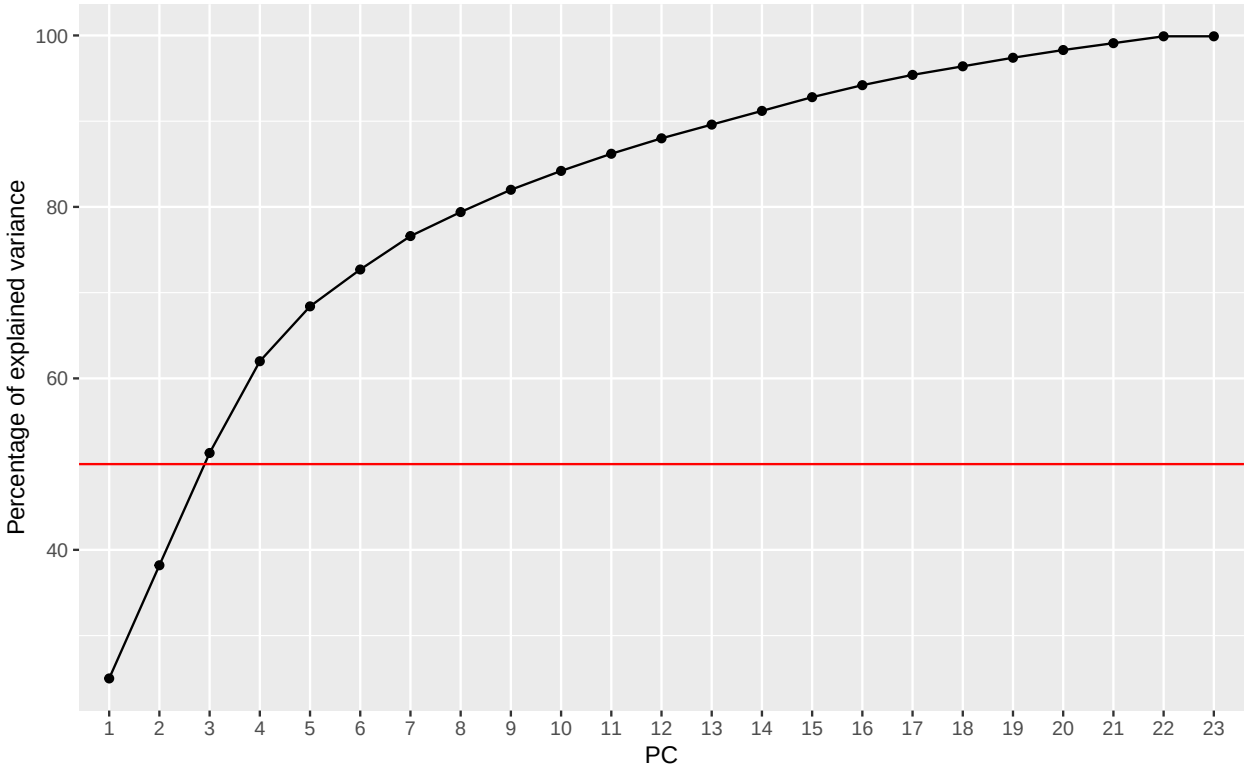
```
df_explained_variance <- data.frame(
  exp_var = pca_results$explained_var$exp_var
  ) %>%
  rownames_to_column("PC") %>%
  mutate(PC = factor(PC,levels = unique(PC)))

scree_plot <-
  ggplot(df_explained_variance, aes(x = PC, y = exp_var)) +
  ylab('Explained variance (%)') +
  ggtitle('Explained variance per principal component') +
  geom_bar(stat = "identity")
scree_plot
```



The variance explained by PC1 is around 25%.
PC2 and PC3 explain almost exactly the same variance (around 13%).

```
df_explained_variance %>%
  mutate(cumulated_variance = cumsum(exp_var)) %>%
  ggplot(mapping = aes(x = PC, y = cumulated_variance)) +
  geom_point() +
```

```
  geom_line(group = 1) +
  labs(y = "Percentage of explained variance") +
  geom_hline(yintercept = 50, color = "red")
```

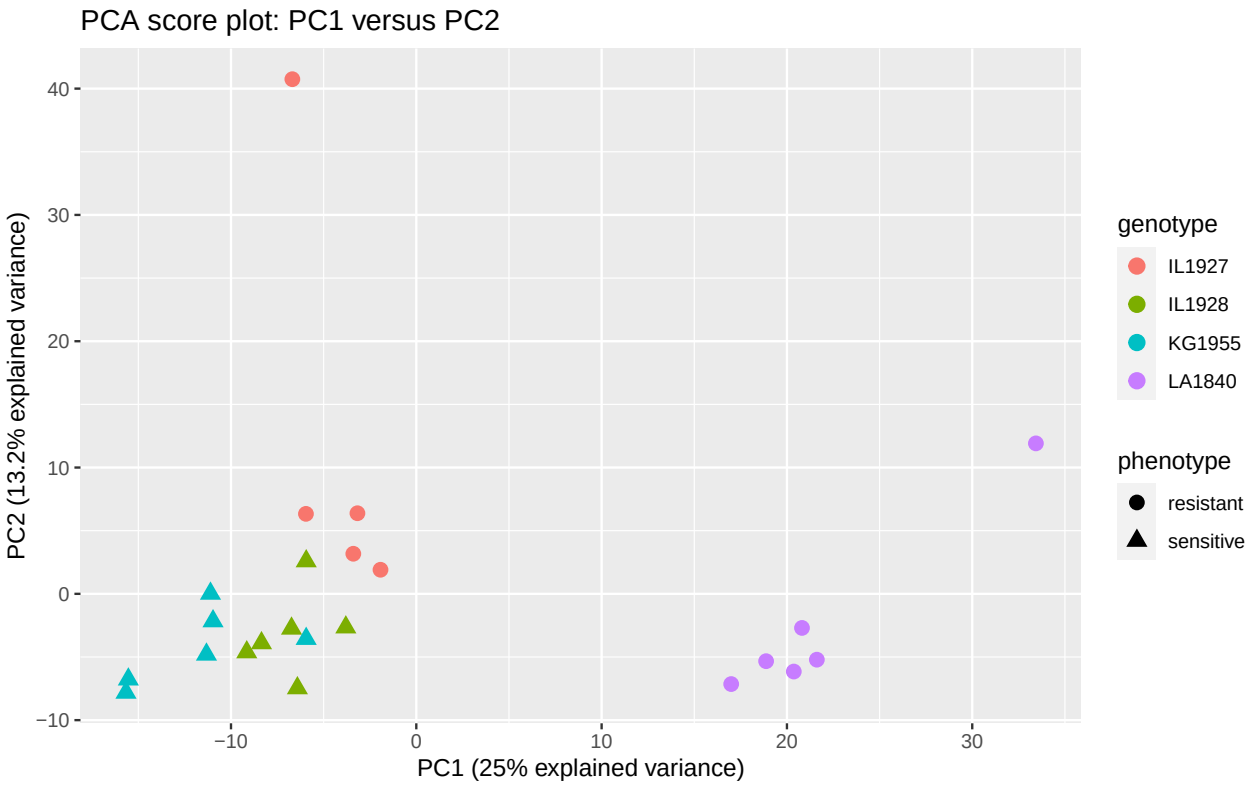


The 3 first PC explain roughly 50% of the total variance.

2.2 Samples score plot

Can we distinguish the resistant genotypes from the sensitive ones on the first 3 PCs?

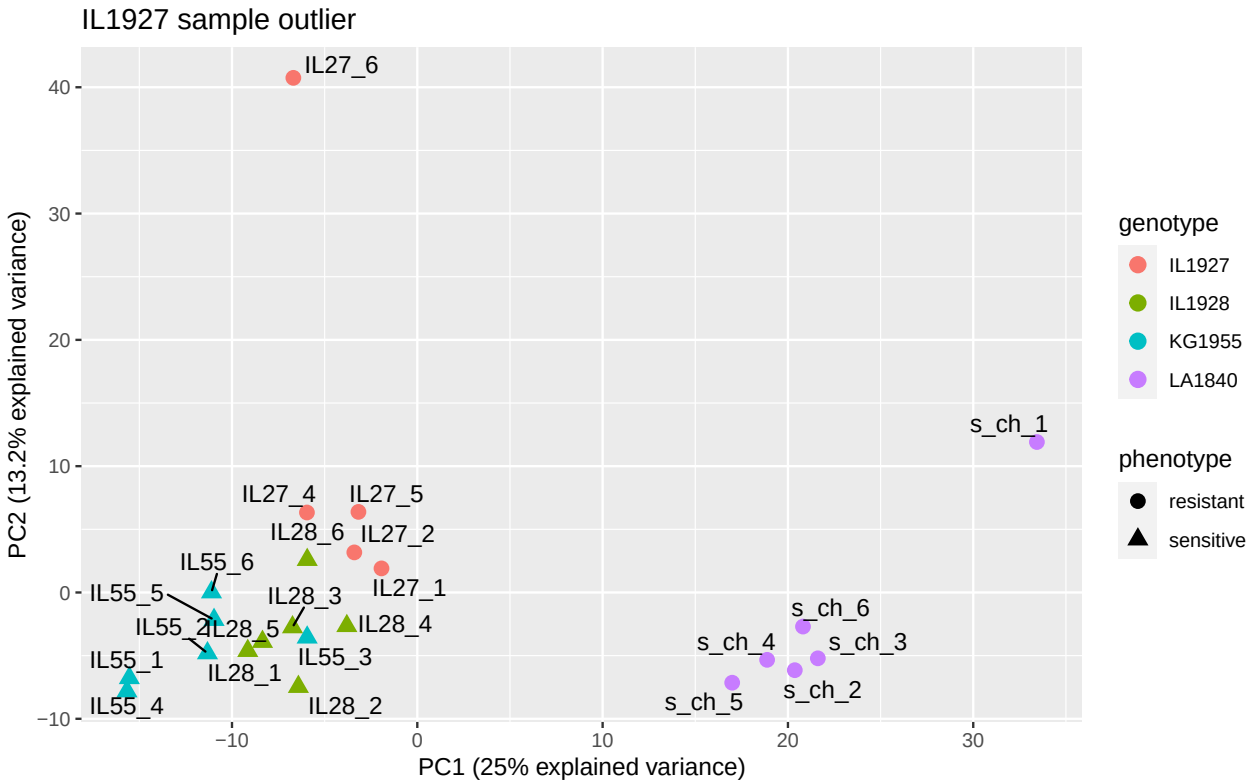
2.2.1 PC1 versus PC2



Obviously, S. chm LA1840 can easily be separated on PC1 from the rest of the genotypes.
The other genotypes are all clustered on PC1 suggesting that PC1 is not related to sensitivity or resistance perhaps.

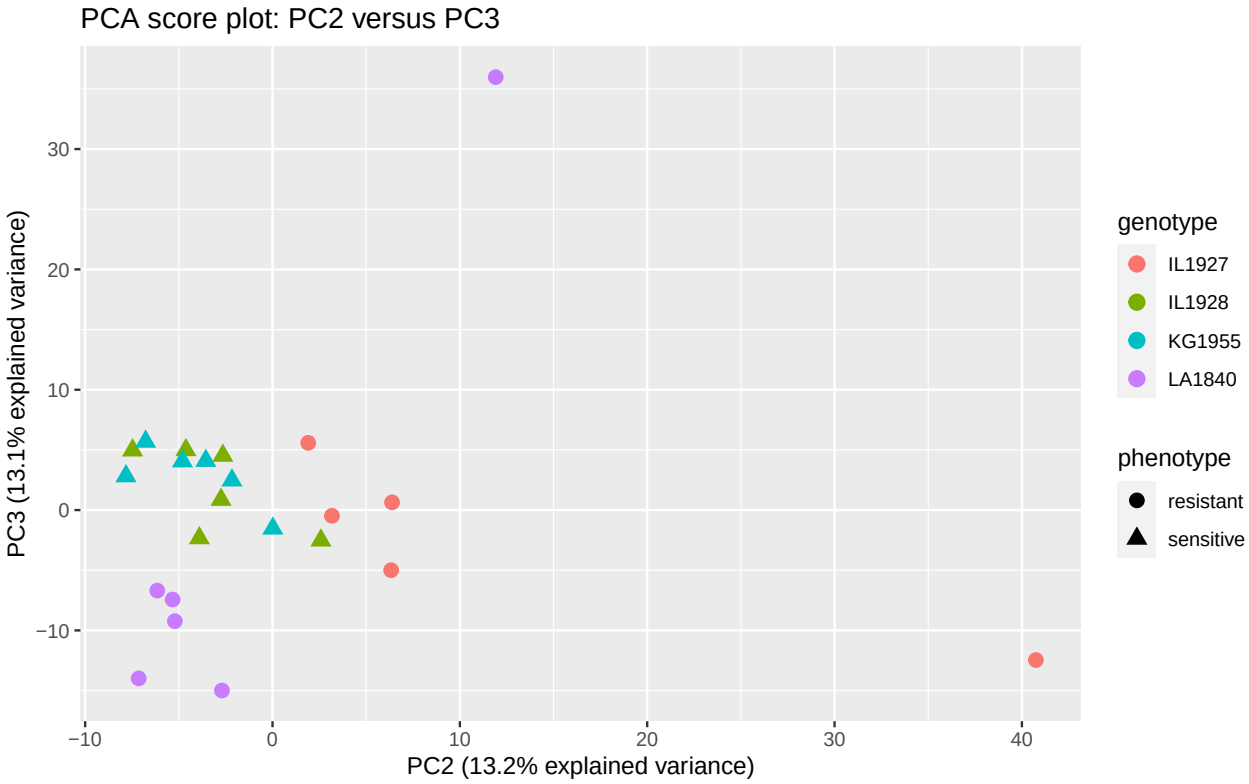
Identifying the outlier on PC2

```
pc1_vs_pc2 +
  ggrepel::geom_text_repel(aes(x = PC1, y = PC2, label = sample)) +
  ggtitle("IL1927 sample outlier ")
```

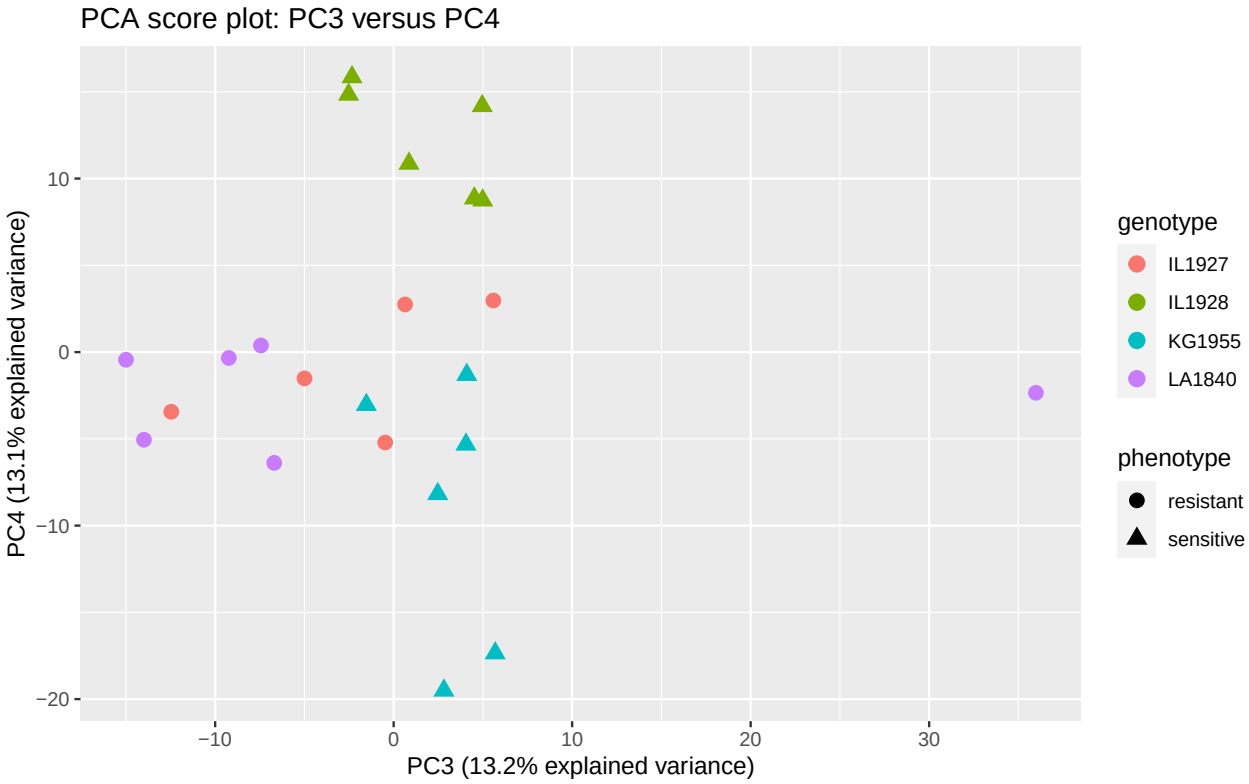


Is it possible to separate the resistant from the sensitive genotypes?
Let's explore the different PCs.

2.2.2 PC2 versus PC3

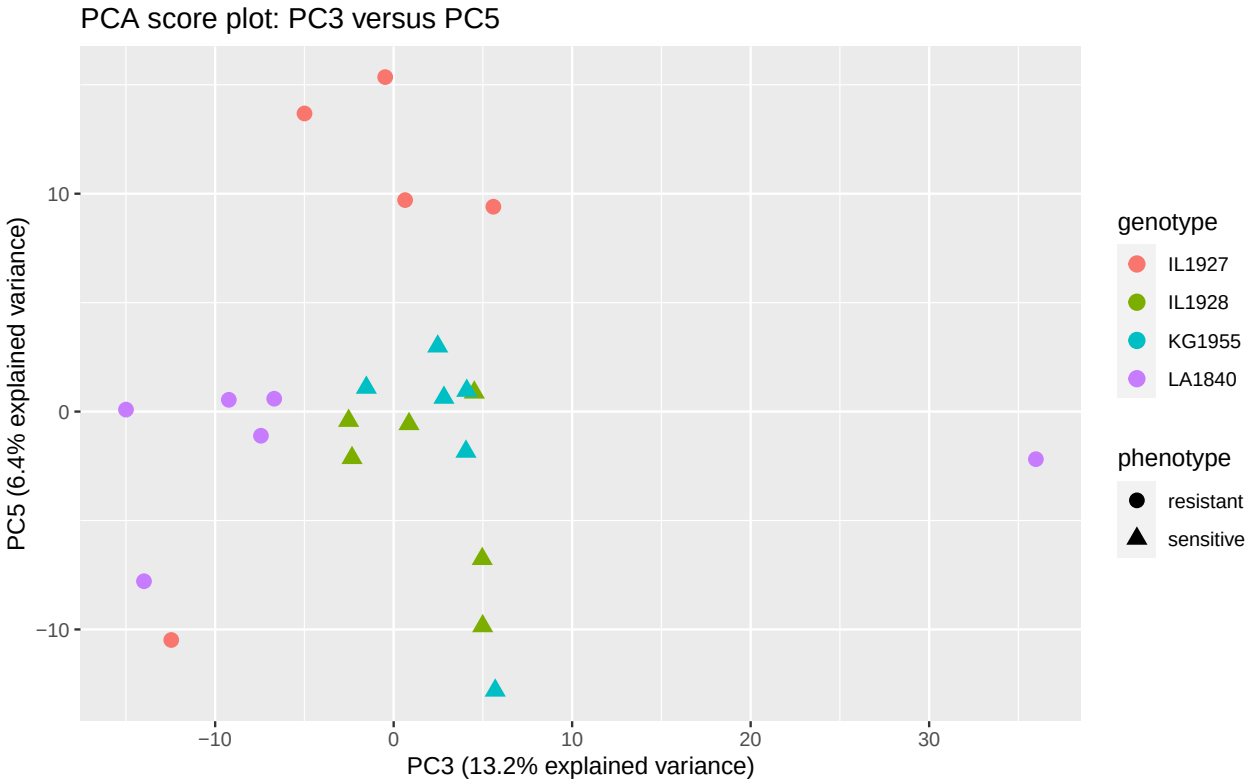


2.2.3 PC3 versus PC4

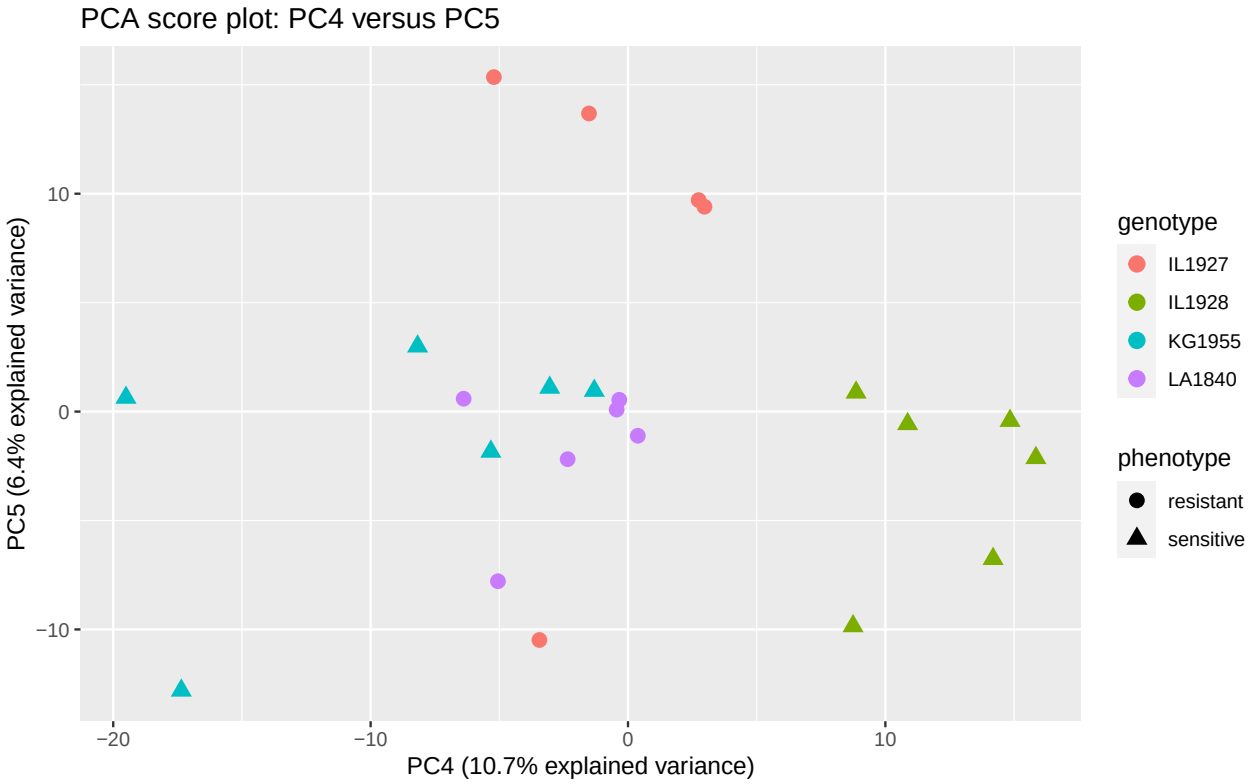


Not super clear but perhaps PC3 slightly separates sensitive from resistant.

2.2.4 PC3 versus PC5



2.2.5 PC4 versus PC5



Still not very clear and PC5 explains only 6% of the total variance so will stop here.

3 PCA analysis (IL27_6 removed)

3.1 Outlier removal

From the previous analysis, two individuals appeared different from the rest of their corresponding genotype:
- IL27_6
- s_ch_1

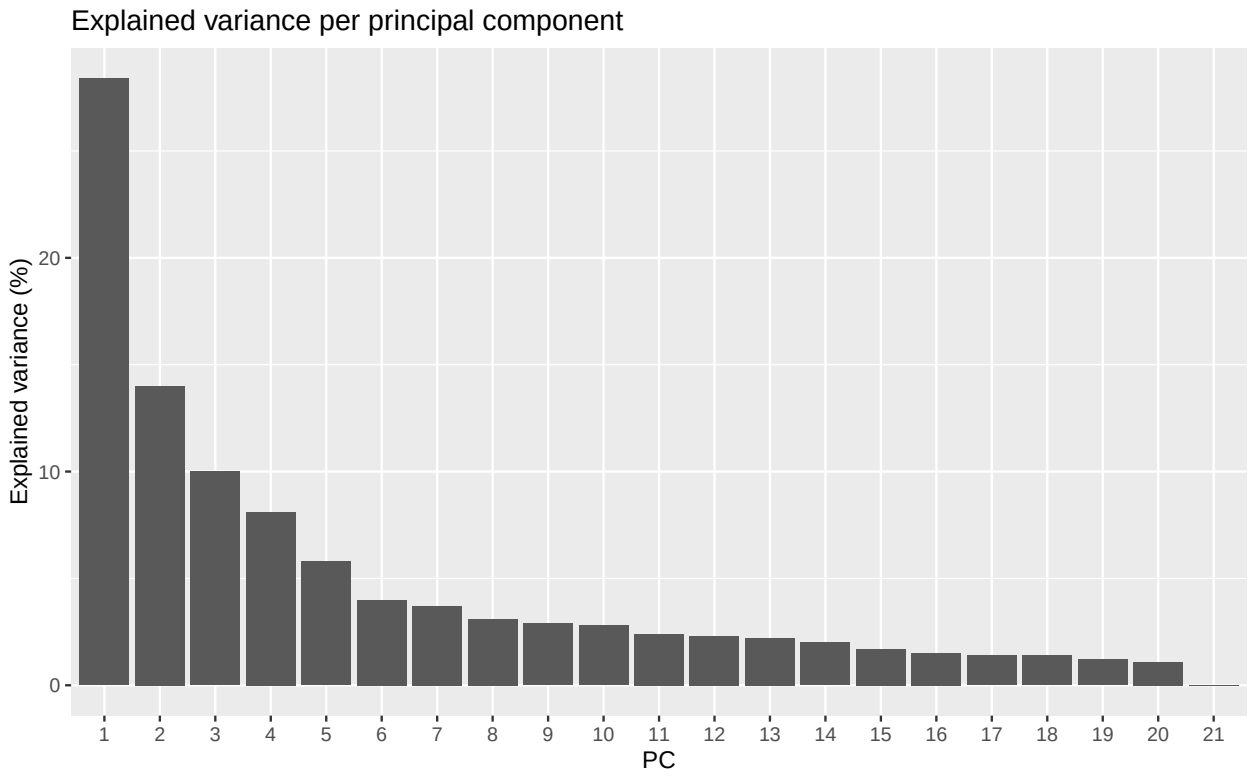
Since IL27_6 appears very different from the other IL1927 samples, it will be removed before the PCA analysis is done. Same for s_ch_1.

	sample	genotype	phenotype
1	IL27_1	IL1927	resistant
2	IL27_2	IL1927	resistant
3	IL27_4	IL1927	resistant
4	IL27_5	IL1927	resistant
6	IL28_1	IL1928	sensitive
7	IL28_2	IL1928	sensitive
8	IL28_3	IL1928	sensitive
9	IL28_4	IL1928	sensitive
10	IL28_5	IL1928	sensitive
11	IL28_6	IL1928	sensitive
12	IL55_1	KG1955	sensitive
13	IL55_2	KG1955	sensitive
14	IL55_3	KG1955	sensitive
15	IL55_4	KG1955	sensitive
16	IL55_5	KG1955	sensitive
17	IL55_6	KG1955	sensitive
19	s_ch_2	LA1840	resistant
20	s_ch_3	LA1840	resistant
21	s_ch_4	LA1840	resistant
22	s_ch_5	LA1840	resistant
23	s_ch_6	LA1840	resistant

The original `peaks` dataframe has 23 rows and the filtered `peaks_wo_outlier` dataframe has now 21 rows. Same for the samples

3.2 Scree plot (wo outlier)

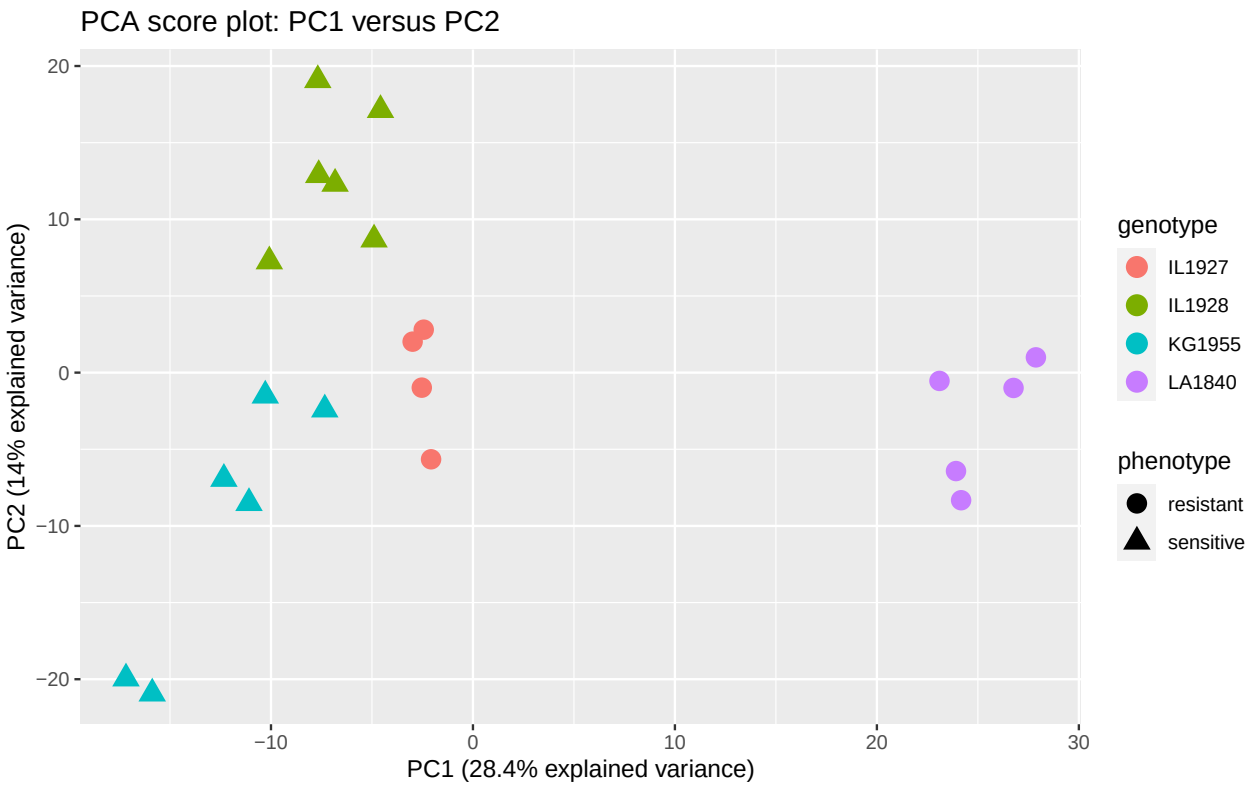
```
pca_results2 <- mypca(peaks_wo_outliers, center = TRUE, scale = TRUE)

df_explained_variance2 <- data.frame(
  exp_var = pca_results2$explained_var$exp_var
  ) %>%
  rownames_to_column("PC") %>%
  mutate(PC = factor(PC,levels = unique(PC)))

scree_plot2 <-
  ggplot(df_explained_variance2, aes(x = PC, y = exp_var)) +
  ylab('Explained variance (%)') +
  ggtitle('Explained variance per principal component') +
  geom_bar(stat = "identity")
scree_plot2
```




```
ggsave("scree_plot.png")
```

3.3 Samples score plot (wo outlier)

3.3.1 PC1 versus PC2



This plot seems to show a much clearer picture with S. chm LA1840 clearly separable from the other genotypes on PC1 (as before).

But now, it seems that IL1927 (resistant) is also more intermediate between LA1840 (resistant) and the other sensitive genotypes (KG1955 “elite line”, IL1927).

3.4 Loadings (wo outliers)

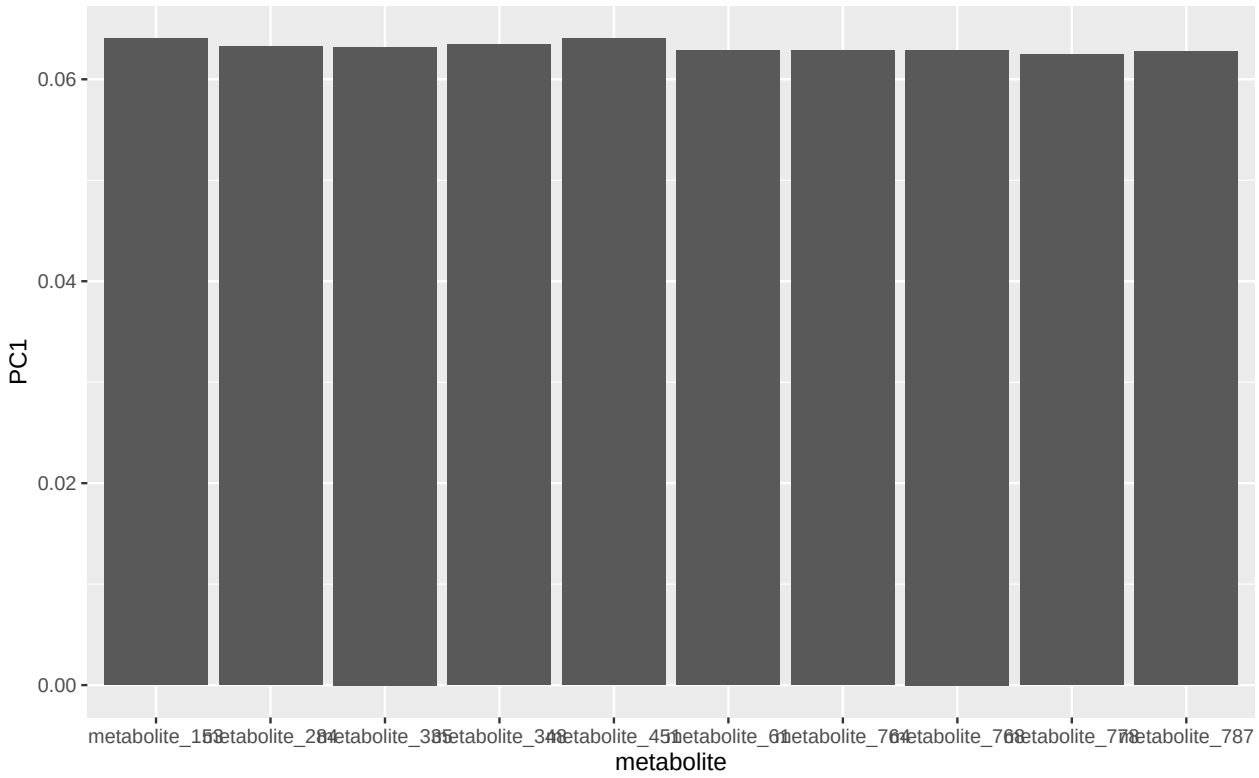
Since PC1 is now more related to our phenotype of interest (resistance/sensitivity), we can extract the metabolites with the highest loadings on PC1.

```
loadings <- pca_results2$loadings

loadings_long <- loadings %>%
  rownames_to_column("metabolite") %>%
  select(metabolite, PC1) %>%
  mutate(abs_PC1 = abs(PC1)) %>%
  arrange(desc(abs_PC1))
kable(head(loadings_long, n = 10))
```

metabolite	PC1	abs_PC1
metabolite_451	0.0640494	0.0640494
metabolite_153	0.0640091	0.0640091
metabolite_348	0.0634267	0.0634267
metabolite_284	0.0632358	0.0632358
metabolite_335	0.0631951	0.0631951
metabolite_768	0.0628917	0.0628917
metabolite_61	0.0628762	0.0628762
metabolite_764	0.0628682	0.0628682
metabolite_787	0.0627310	0.0627310
metabolite_778	0.0624825	0.0624825

We can also visualise it as a barplot.

```
loadings_long %>%
  arrange(desc(abs_PC1)) %>%
  top_n(10) %>%
  ggplot(., aes(x = metabolite, y = PC1)) +
  geom_bar(stat = "identity")
```


0.06
0.04
0.02
0.00
PC1
metabolite_158
metabolite_284
metabolite_385
metabolite_348
metabolite_451
metabolite_61
metabolite_764
metabolite_768
metabolite_778
metabolite_787
metabolite