Exam - NCI60

CODE

Florence Mehl

April 04, 2023

NCI60 dataset

This dataset is a selection of data from a publicly available repository of the National Cancer Institute, i.e. the NCI-60 dataset, which includes gene expression analysis as well as data from metabolomics and proteomics experiments. It provides experimental data obtained from 60 human cancer cell lines derived from nine tissue origins, such as breast, colon, lung, ovary, blood and skin. These cell lines constitute key in vitro models for cancer research and they are used for extensive anti-cancer drug screening.

R.H. Shoemaker, The NCI60 human tumour cell line anticancer drug screen, Nat. Rev. Cancer 6 (2006) 813–823.

Data import and formatting

Some variables with near zero variance are removed from the metabolomics dataset.

HIDE

```
load("../data/NCI60_custom.RData")
genes <- NCI60_custom$transcripto$data
metabo <- NCI60_custom$metabo$data
proteo <- NCI60_custom$proteo$data

# remove near zero variance features in metabolomics data
metaboNZV <- caret::nearZeroVar(metabo, saveMetrics = TRUE)
metabo <- metabo[, metaboNZV[,"zeroVar"]==FALSE]

data <- list(genes = genes,
             metabo = metabo,
             proteo = proteo)

metadata <- data.frame(cell.line = NCI60_custom$metabo$splMD$Cell.line, origin = NCI60_cust
om$metabo$splMD$Origin)
rownames(metadata) <- NCI60_custom$metabo$splMD$splID
```

Single block multivariate analyses

Principal Component Analysis

Principal Component Analysis is run on each centered and scaled dataset.

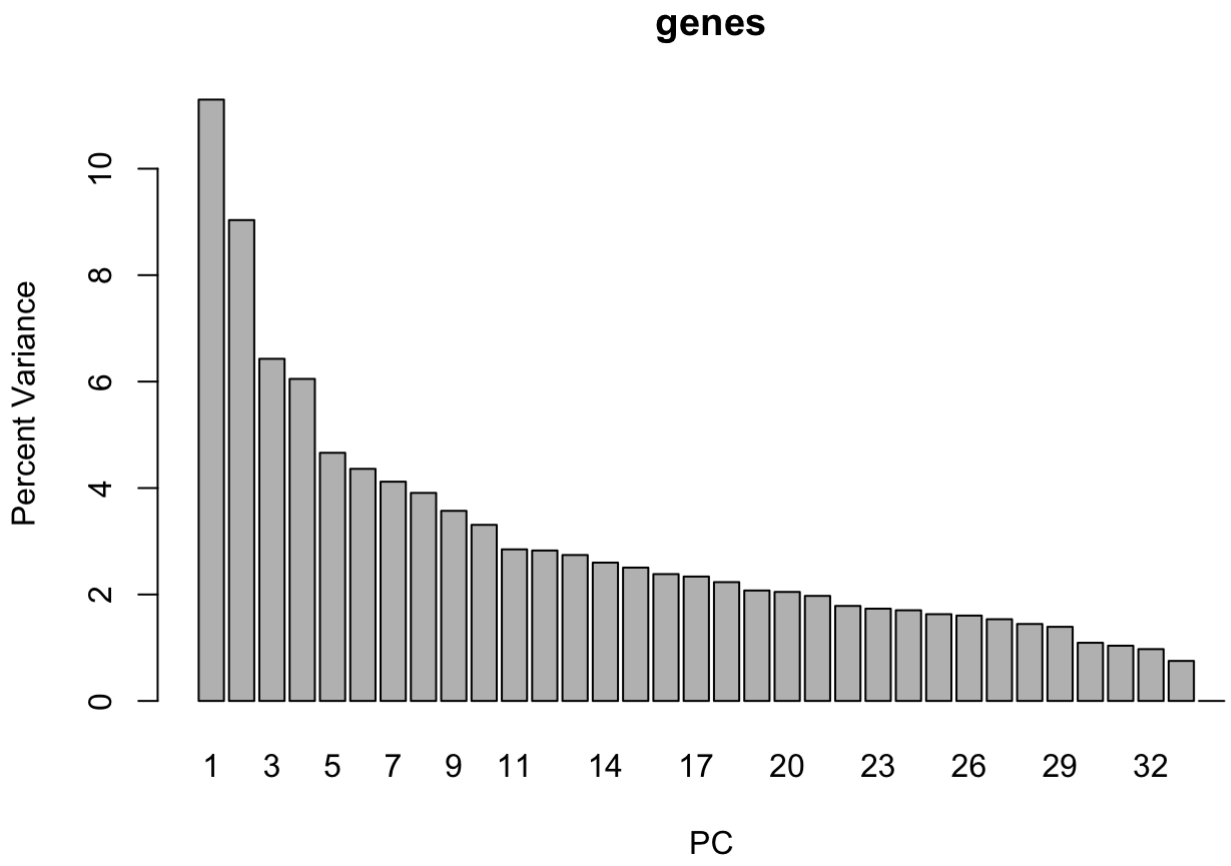
HIDE

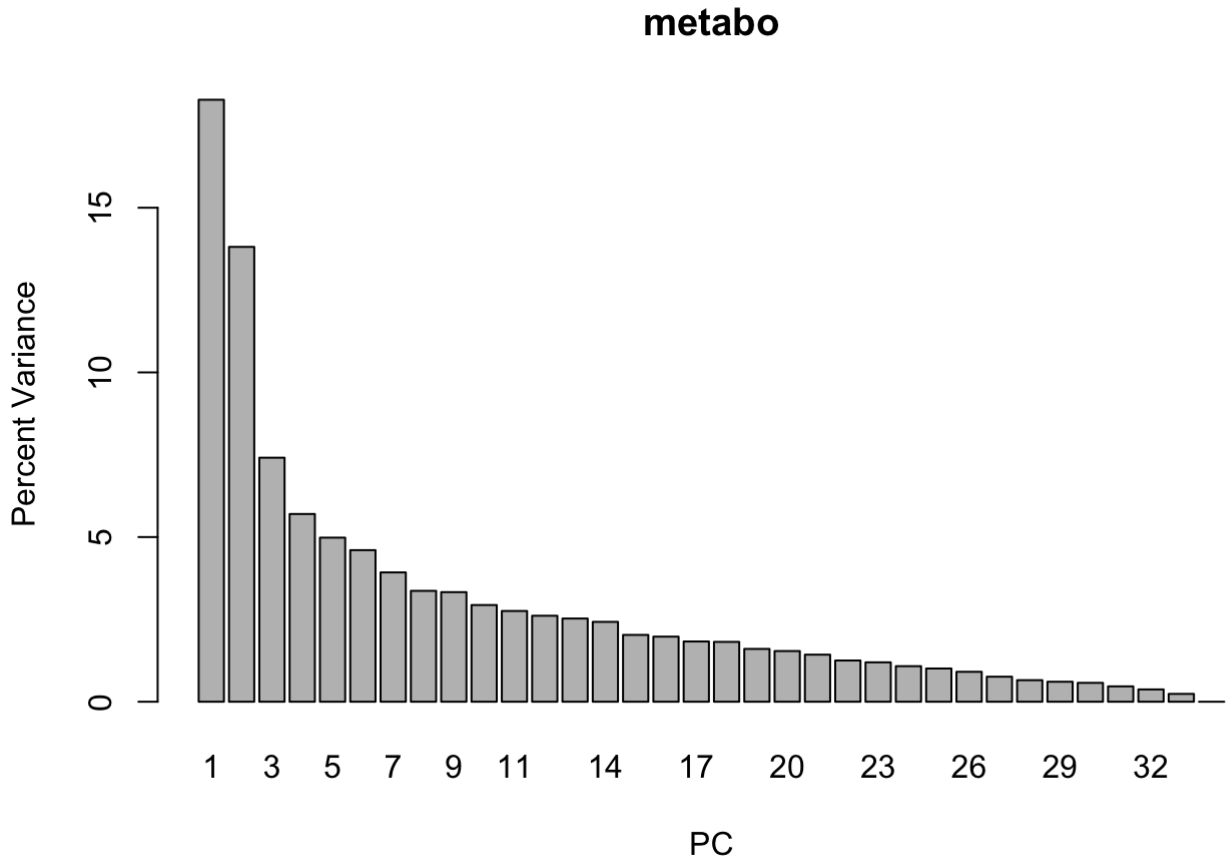
```
pca.res <- lapply(data, function(x) {
  prcomp(x, center=TRUE, scale.=TRUE)
})
```

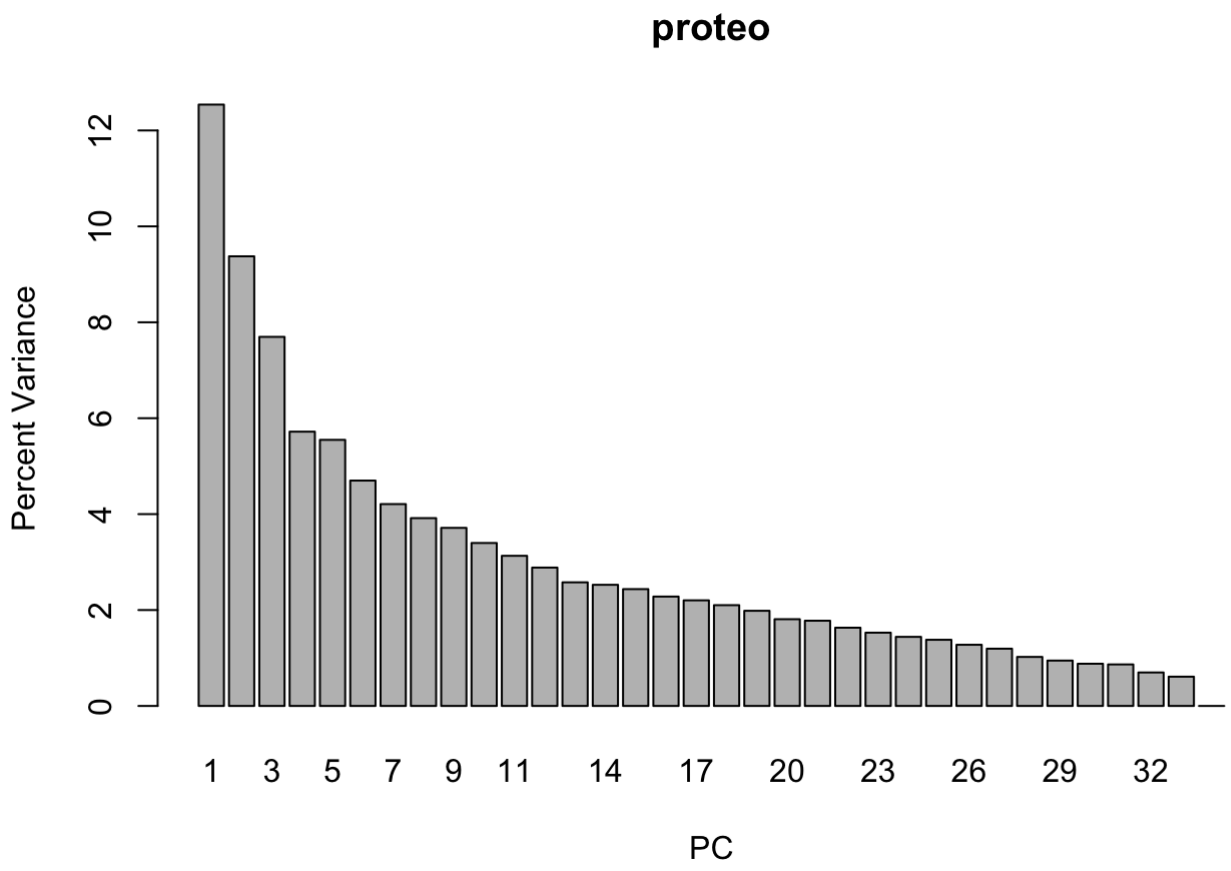
Scree plot on variance percentage.

HIDE

```
varPercent <- lapply(pca.res, function(x) {x$sdev^2/sum(x$sdev^2) * 100})

for(omic in c("genes", "metabo", "proteo")){
  barplot(varPercent[[omic]], xlab='PC', ylab='Percent Variance', names.arg=1:length(varPer
cent[[omic]]), main=omic)
}
```


genes
Percent Variance
0
2
4
6
8
10
1
3
5
7
9
11
14
17
20
23
26
29
32
PC

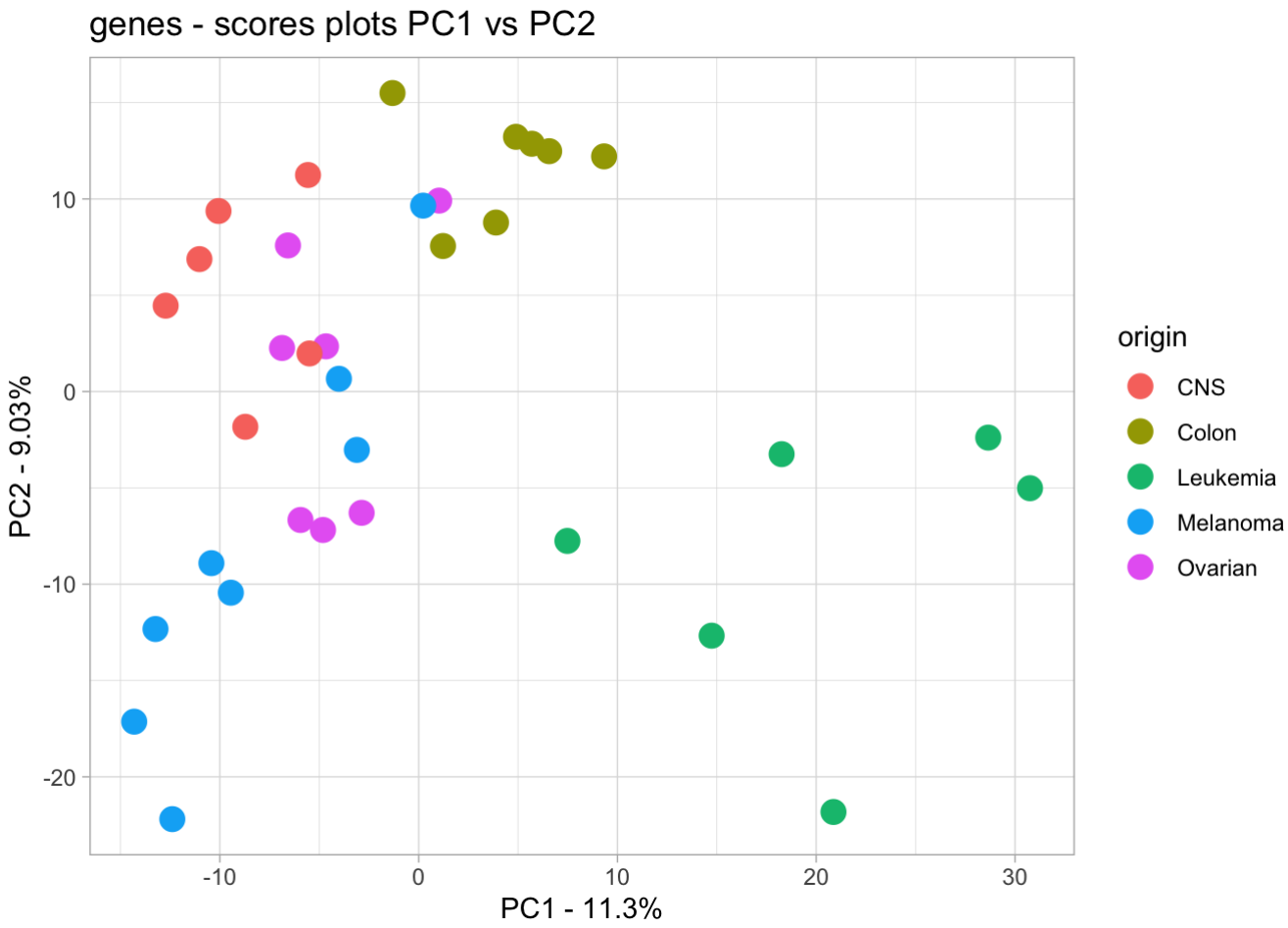

metabo
Percent Variance
0
5
10
15
1
3
5
7
9
11
14
17
20
23
26
29
32
PC

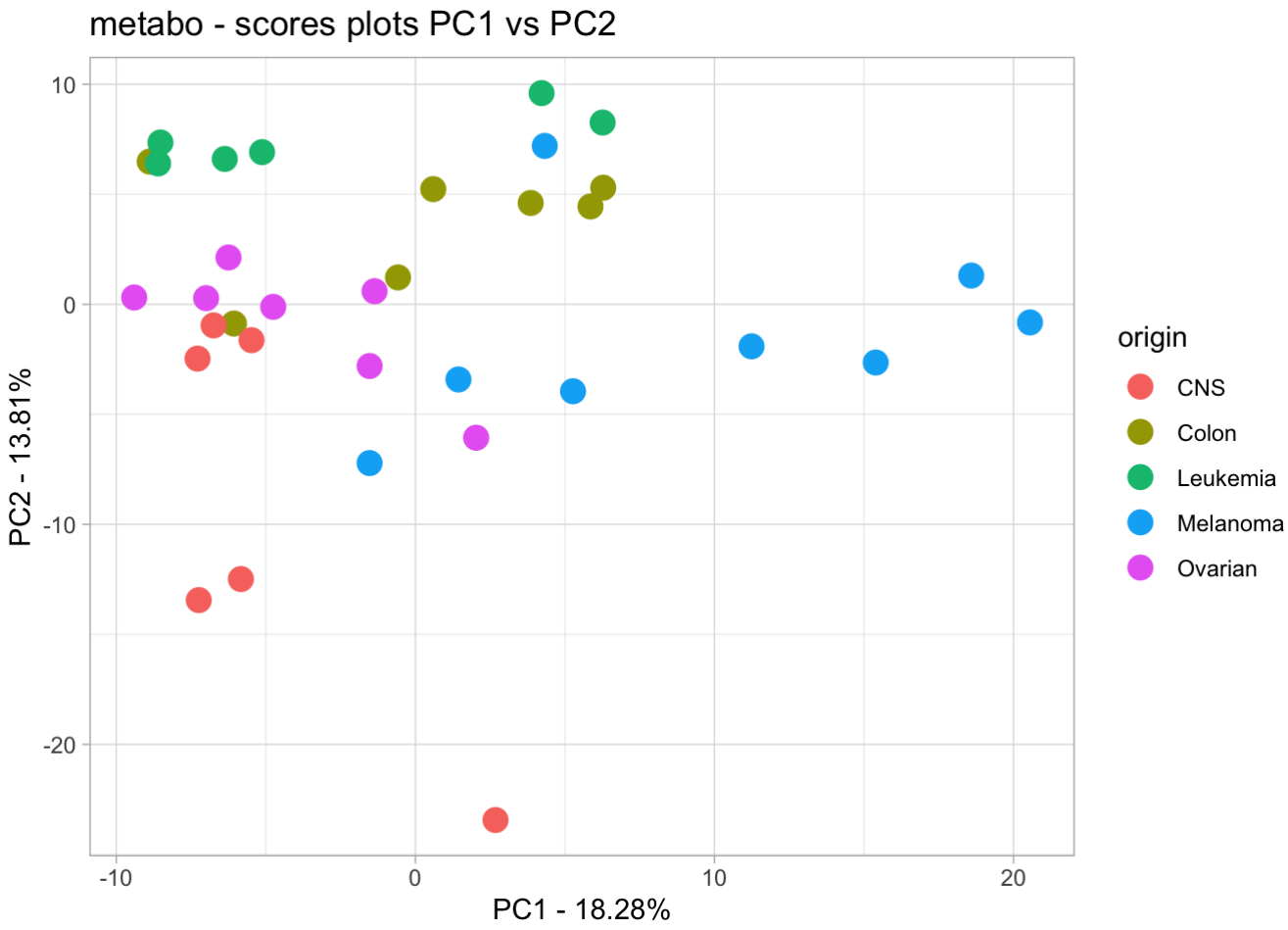


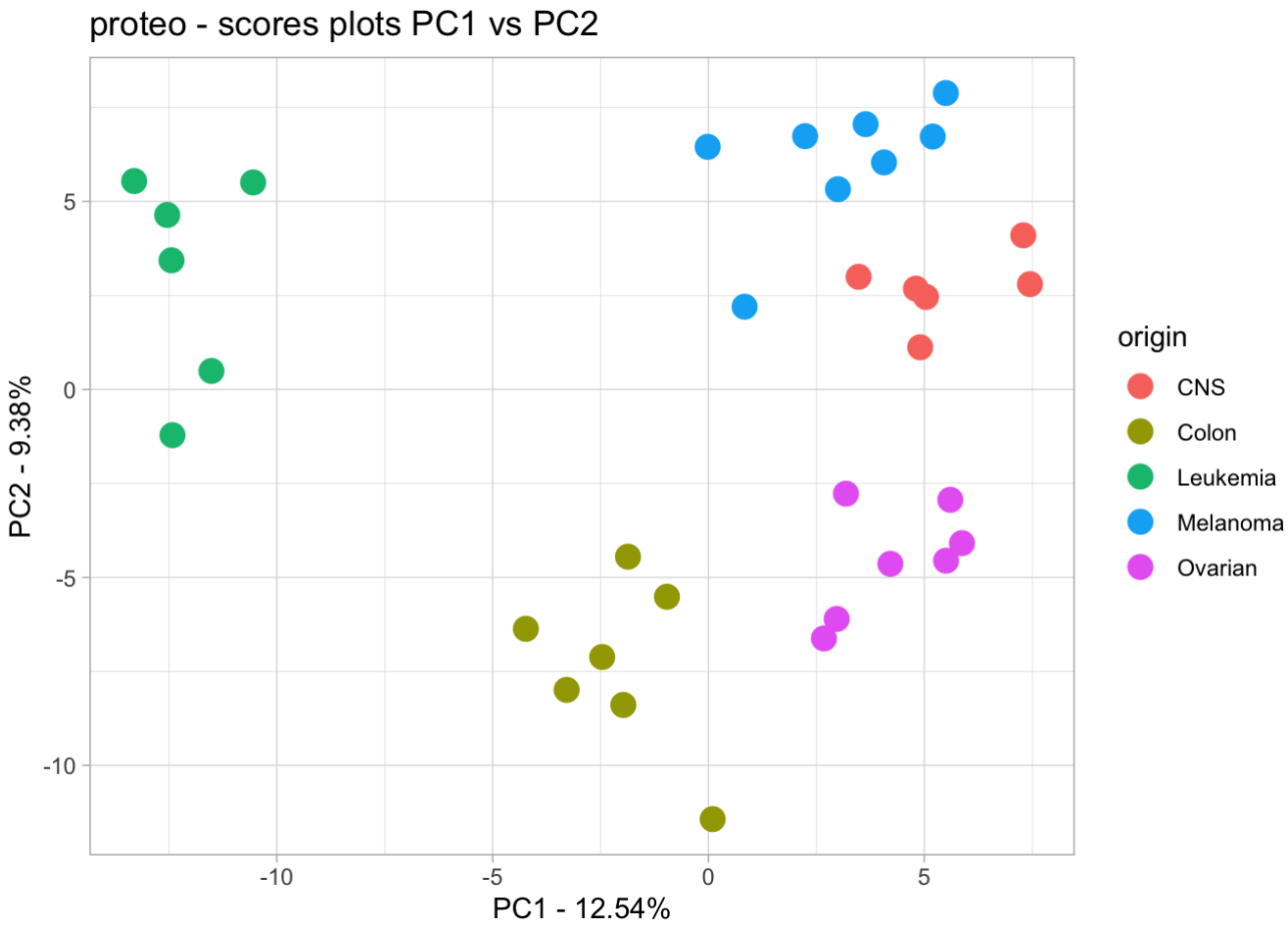
Score plot: plot of sample distribution.

HIDE

```
for(omic in c("genes", "metabo", "proteo")){
  scores <- data.frame(metadata, pca.res[[omic]]$x)
  p <- ggplot(scores, aes(x=PC1, y=PC2, col=origin)) +
  geom_point(size=4) +
  labs(x=paste0("PC1 - ", round(varPercent[[omic]][1], digits=2), "%"),
       y=paste0("PC2 - ", round(varPercent[[omic]][2], digits=2), "%"),
       title = paste0(omic, " - scores plots PC1 vs PC2")) +
  theme_light()
  print(p)
}
```


genes - scores plots PC1 vs PC2
PC2 - 9.03%
PC1 - 11.3%
10
0
-10
-20
-10
0
10
20
30
origin
CNS
Colon
Leukemia
Melanoma
Ovarian


metabo - scores plots PC1 vs PC2
PC2 - 13.81%
PC1 - 18.28%
10
0
-10
-20
-10
0
10
20
origin
CNS
Colon
Leukemia
Melanoma
Ovarian

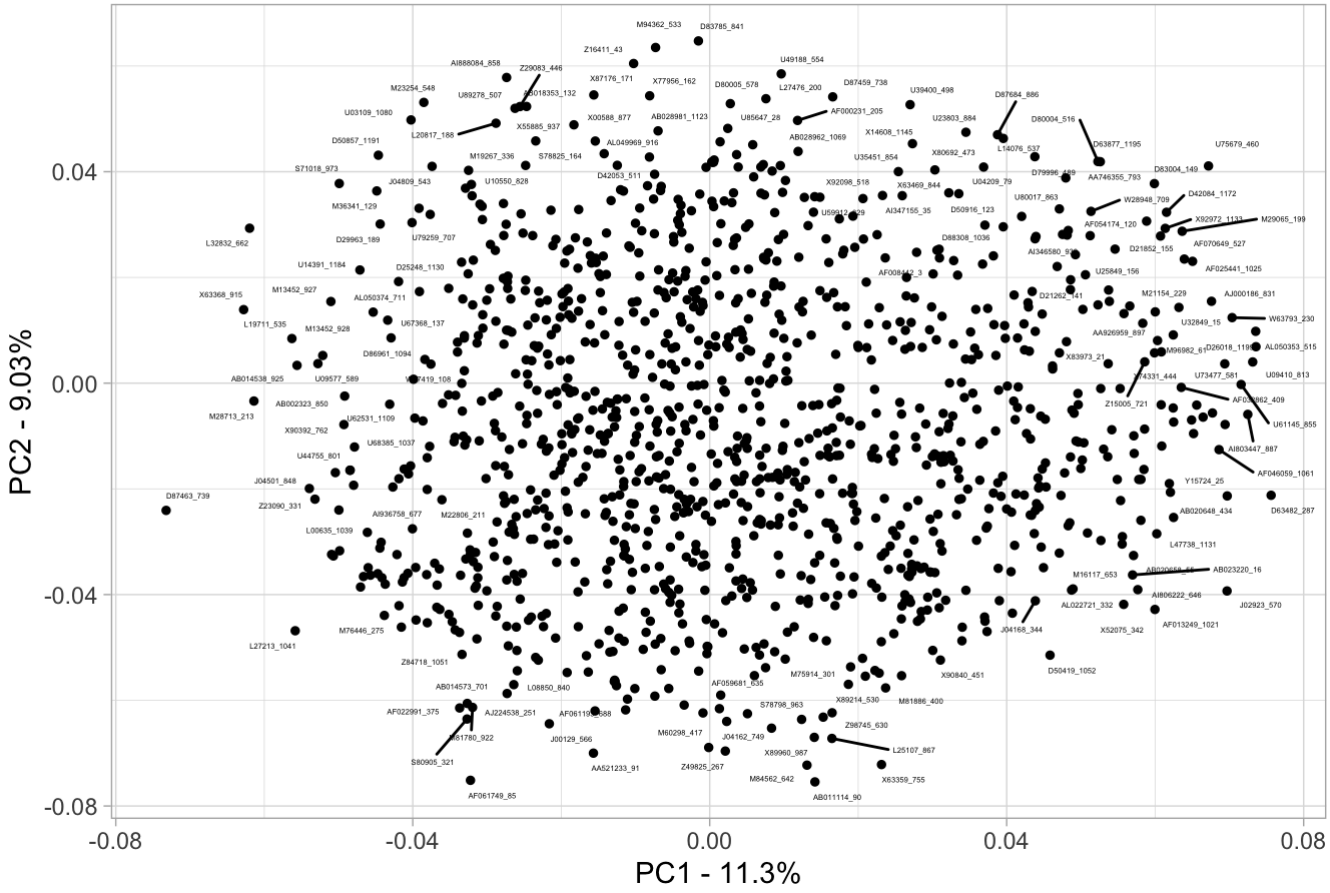


Loadings plot: plot of contributions of variables to principal components

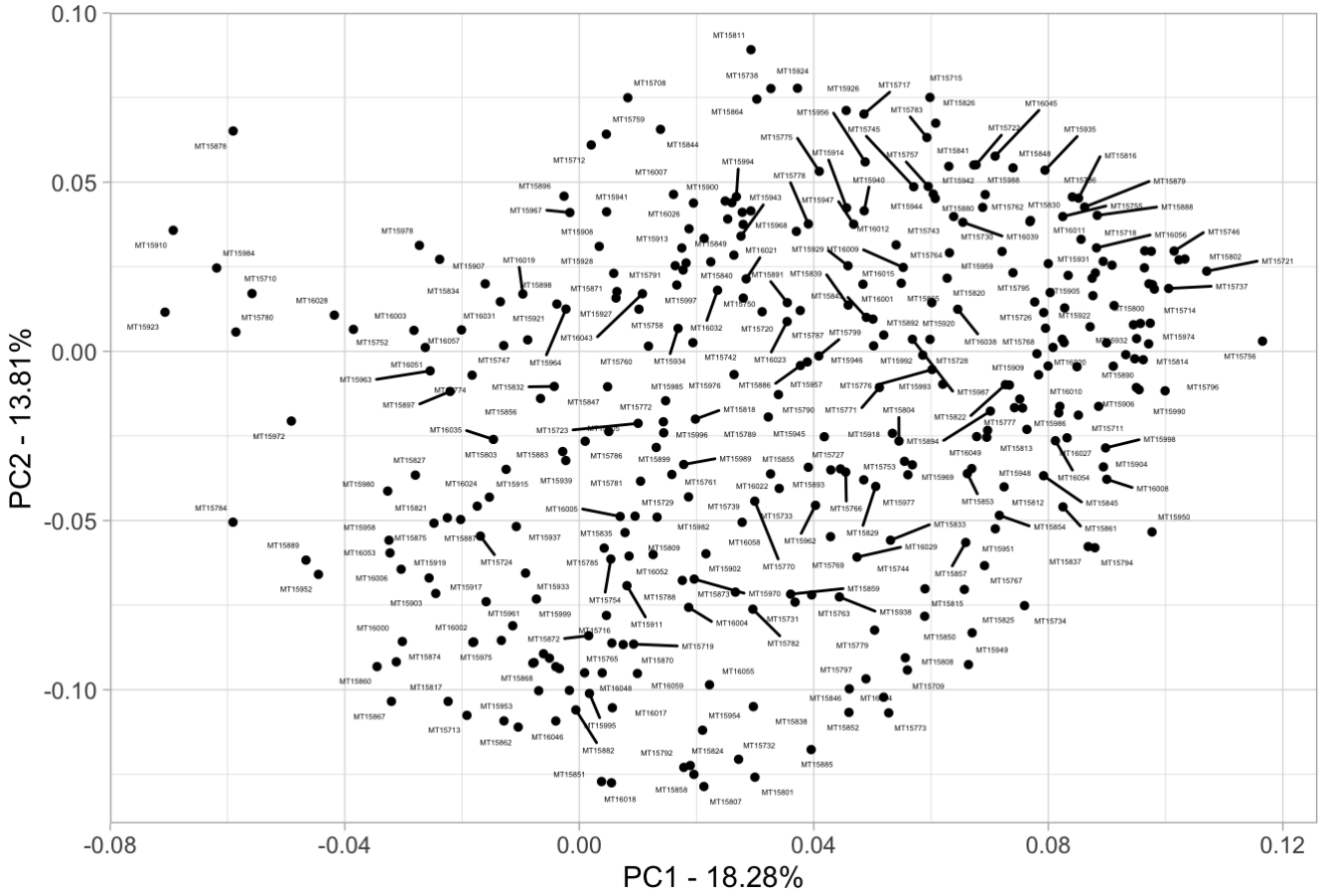
HIDE

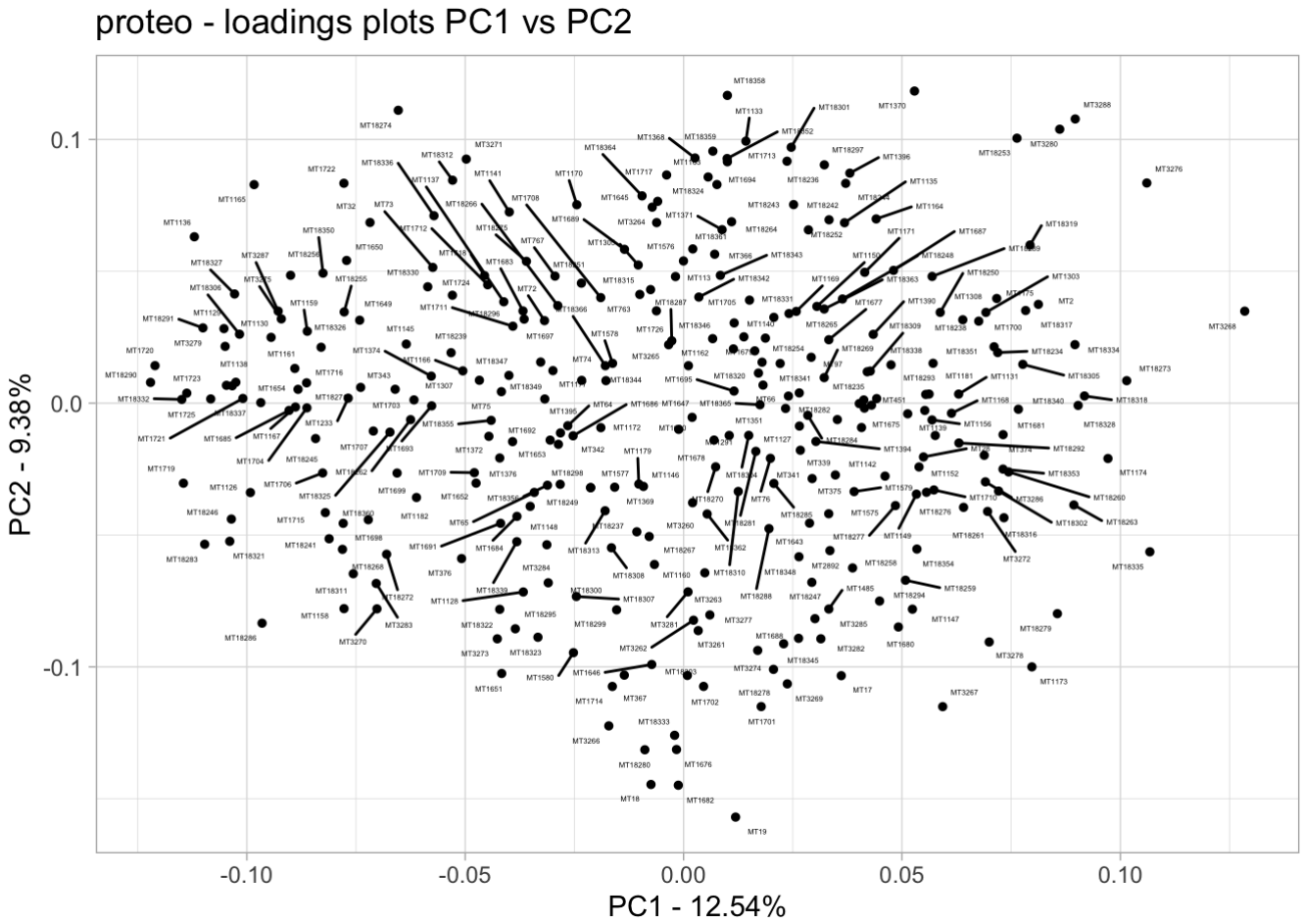
```
for(omic in c("genes", "metabo", "proteo")){
  loadings <- data.frame(pca.res[[omic]]$rotation)
  loadings$variables <- rownames(loadings)
  p <- ggplot(loadings, aes(x=PC1, y=PC2, label=variables)) +
  geom_point(size=1) +
  geom_text_repel(size=1) +
  labs(x=paste0("PC1 - ", round(varPercent[[omic]][1], digits=2), "%"),
       y=paste0("PC2 - ", round(varPercent[[omic]][2], digits=2), "%"),
       title = paste0(omic, " - loadings plots PC1 vs PC2")) +
  theme_light()
  print(p)
}
```

genes - loadings plots PC1 vs PC2



metabo - loadings plots PC1 vs PC2





Partial Least Square Discriminant Analysis

A Partial Least Square Discriminant Analysis is done on colon and ovarian tissues based on each omic data.

Prepare data and run model to discriminate between Colon and Ovarian tissues

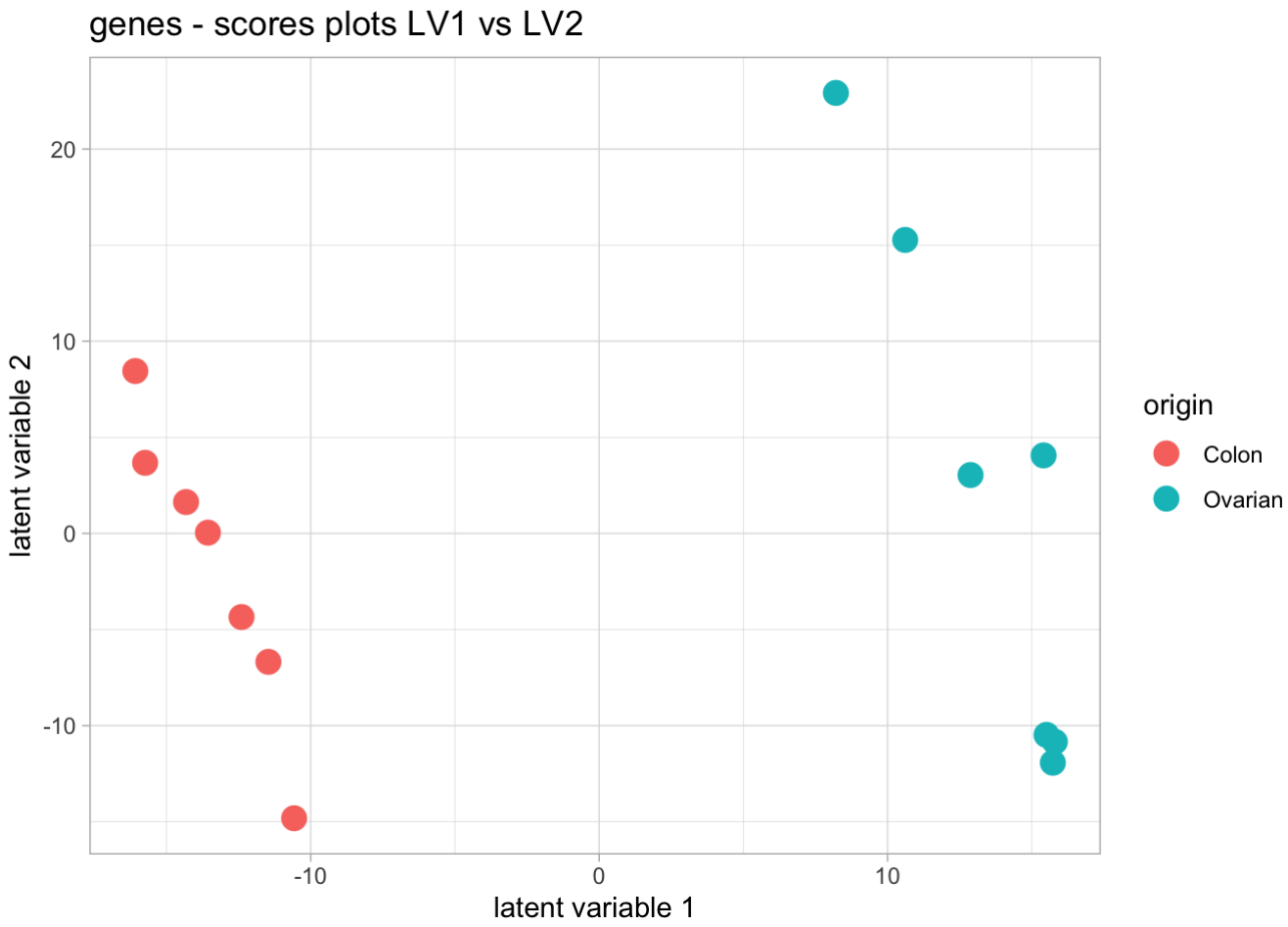
HIDE

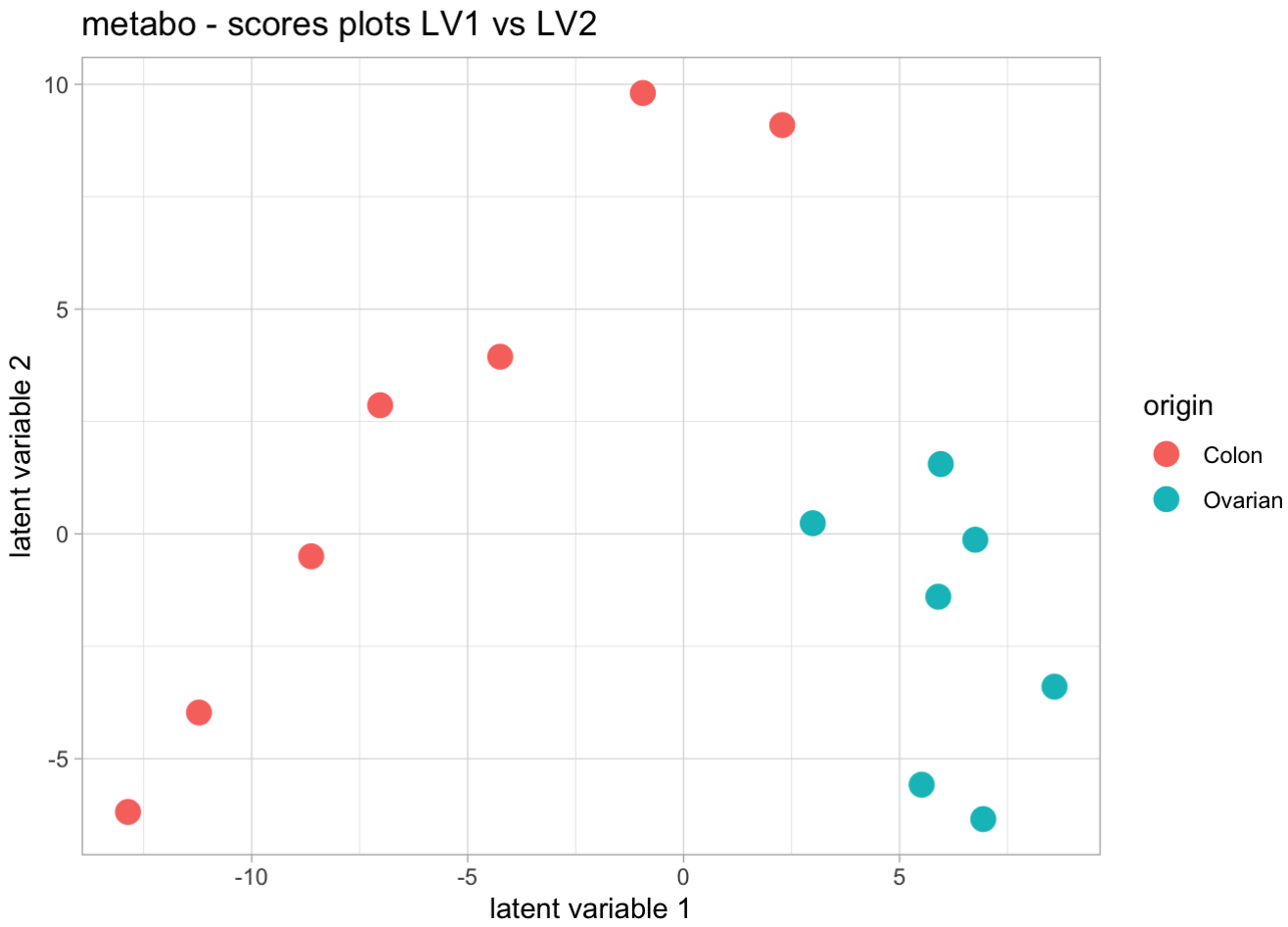
```
data.pls <- list(genes = genes[metadata$origin %in% c("Colon", "Ovarian"),],
                 metabo = metabo[metadata$origin %in% c("Colon", "Ovarian"),],
                 proteo = proteo[metadata$origin %in% c("Colon", "Ovarian"),])

metadata.pls <- metadata[metadata$origin %in% c("Colon", "Ovarian"),]

Y <- data.frame("Colon" = ifelse(metadata.pls$origin == "Colon", 1, 0),
                "Ovarian" = ifelse(metadata.pls$origin == "Ovarian", 1, 0))

pls.res <- lapply(data.pls, function(x) {
  pls(X=x, Y=Y, ncomp=2, scale=TRUE, mode="regression")
})
```

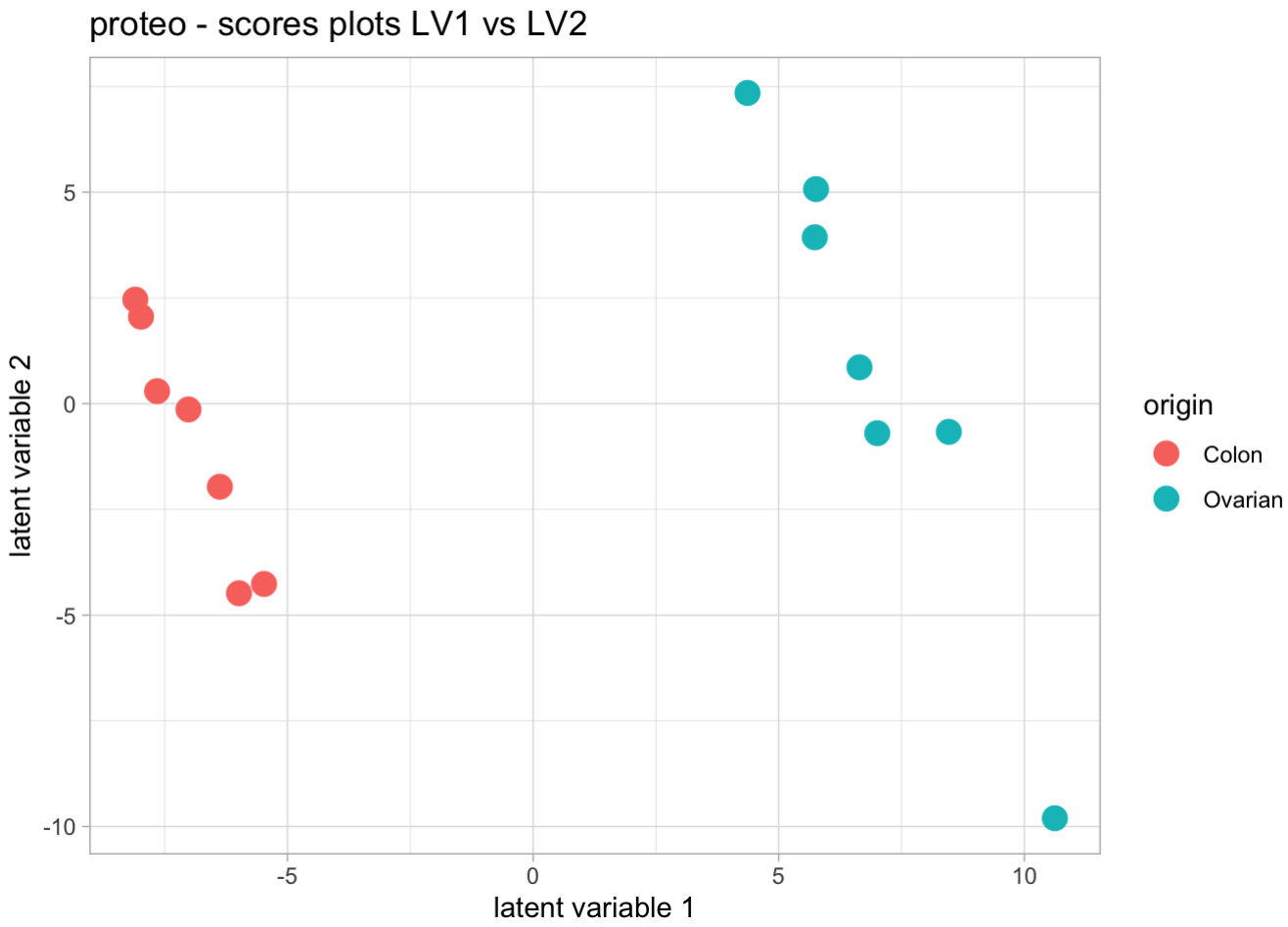
Score plot: plot of sample distribution.

HIDE

```
for(omic in c("genes", "metabo", "proteo")){
  scores <- data.frame(metadata.pls, pls.res[[omic]]$variates$X)
  p <- ggplot(scores, aes(x=comp1, y=comp2, col=origin)) +
  geom_point(size=4) +
  labs(x="latent variable 1",
       y="latent variable 2",
       title = paste0(omic, " - scores plots LV1 vs LV2")) +
  theme_light()
  print(p)
}
```


genes - scores plots LV1 vs LV2
latent variable 2
20
10
0
-10
-10
0
10
latent variable 1
origin
Colon
Ovarian

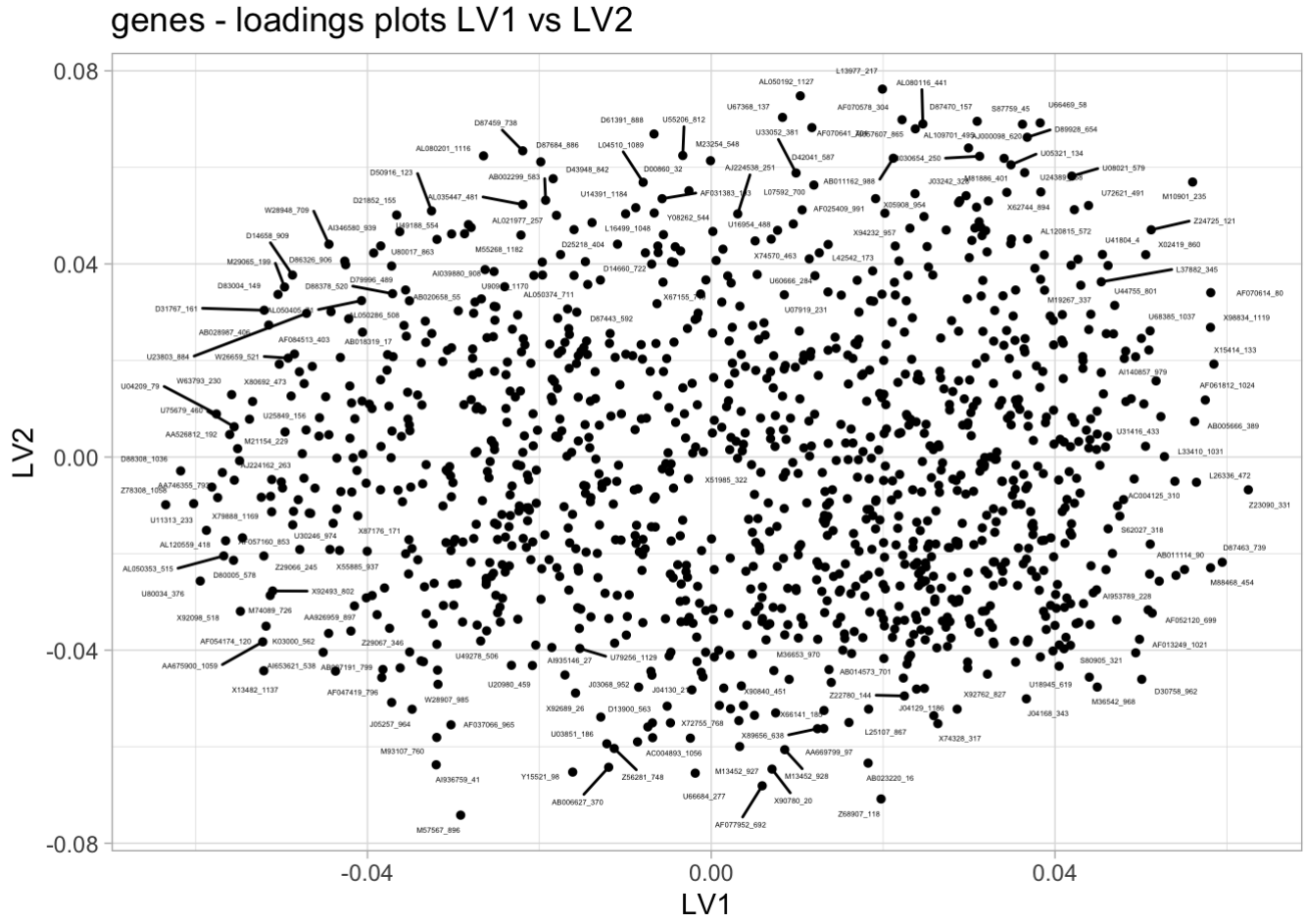

metabo - scores plots LV1 vs LV2
latent variable 2
10
5
0
-5
-10
-5
0
5
latent variable 1
origin
Colon
Ovarian

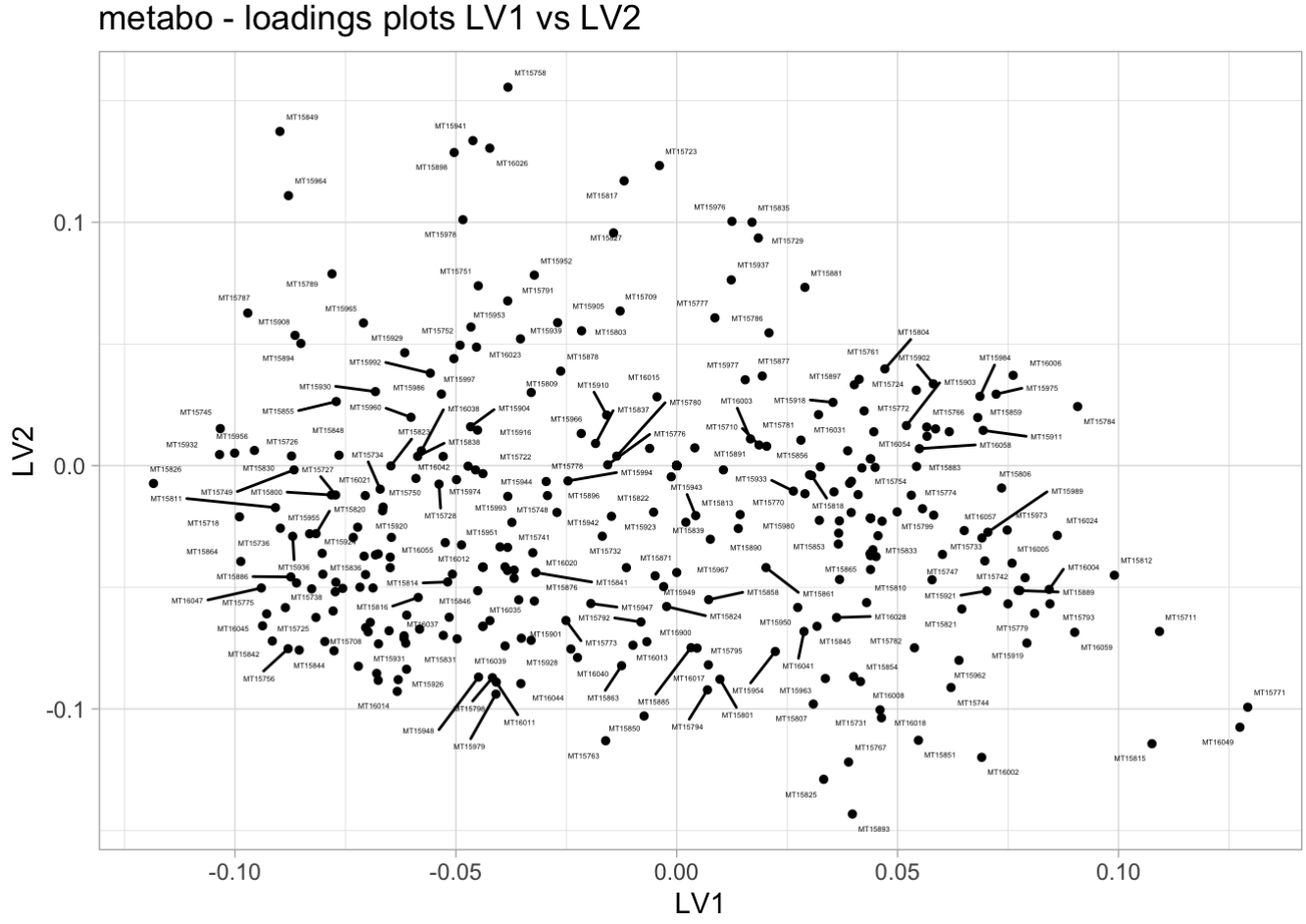


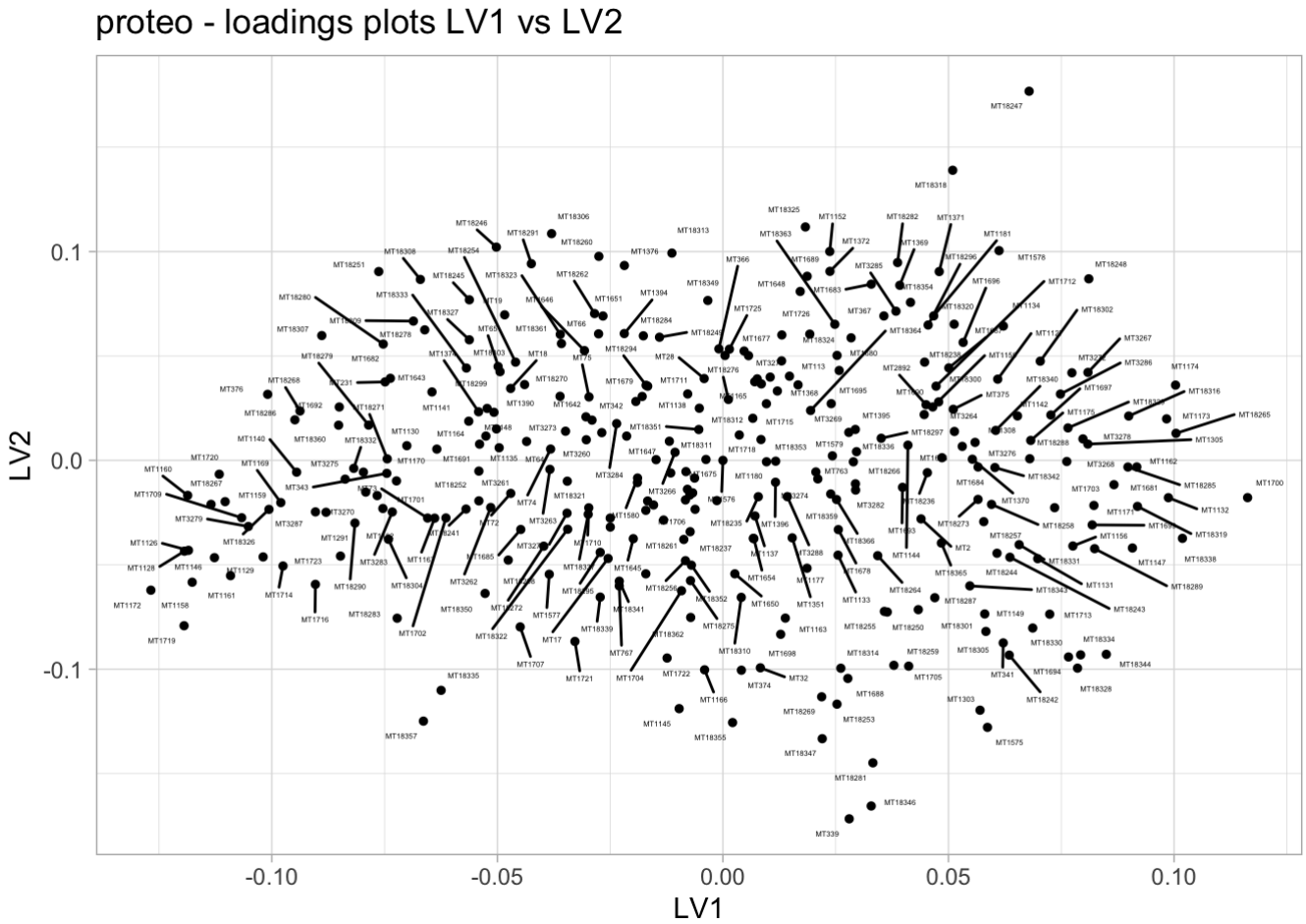
Loadings plot: plot of contributions of variables to principal components

HIDE

```
for(omic in c("genes", "metabo", "proteo")){
  loadings <- data.frame(pls.res[[omic]]$loadings$X)
  loadings$variables <- rownames(loadings)
  p <- ggplot(loadings, aes(x=comp1, y=comp2, label=variables)) +
  geom_point(size=1) +
  geom_text_repel(size=1) +
  labs(x=paste0("LV1"),
       y=paste0("LV2"),
       title = paste0(omic, " - loadings plots LV1 vs LV2")) +
  theme_light()
  print(p)
}
```


genes - loadings plots LV1 vs LV2
LV2
LV1
0.08
0.04
0.00
-0.04
-0.08
-0.04
0.00
0.04


metabo - loadings plots LV1 vs LV2
LV2
LV1
0.1
0.0
-0.1
-0.10
-0.05
0.00
0.05
0.10



Integrative analysis of complete dataset

Unsupervised analysis with CCSWA

prepare data and run model

HIDE

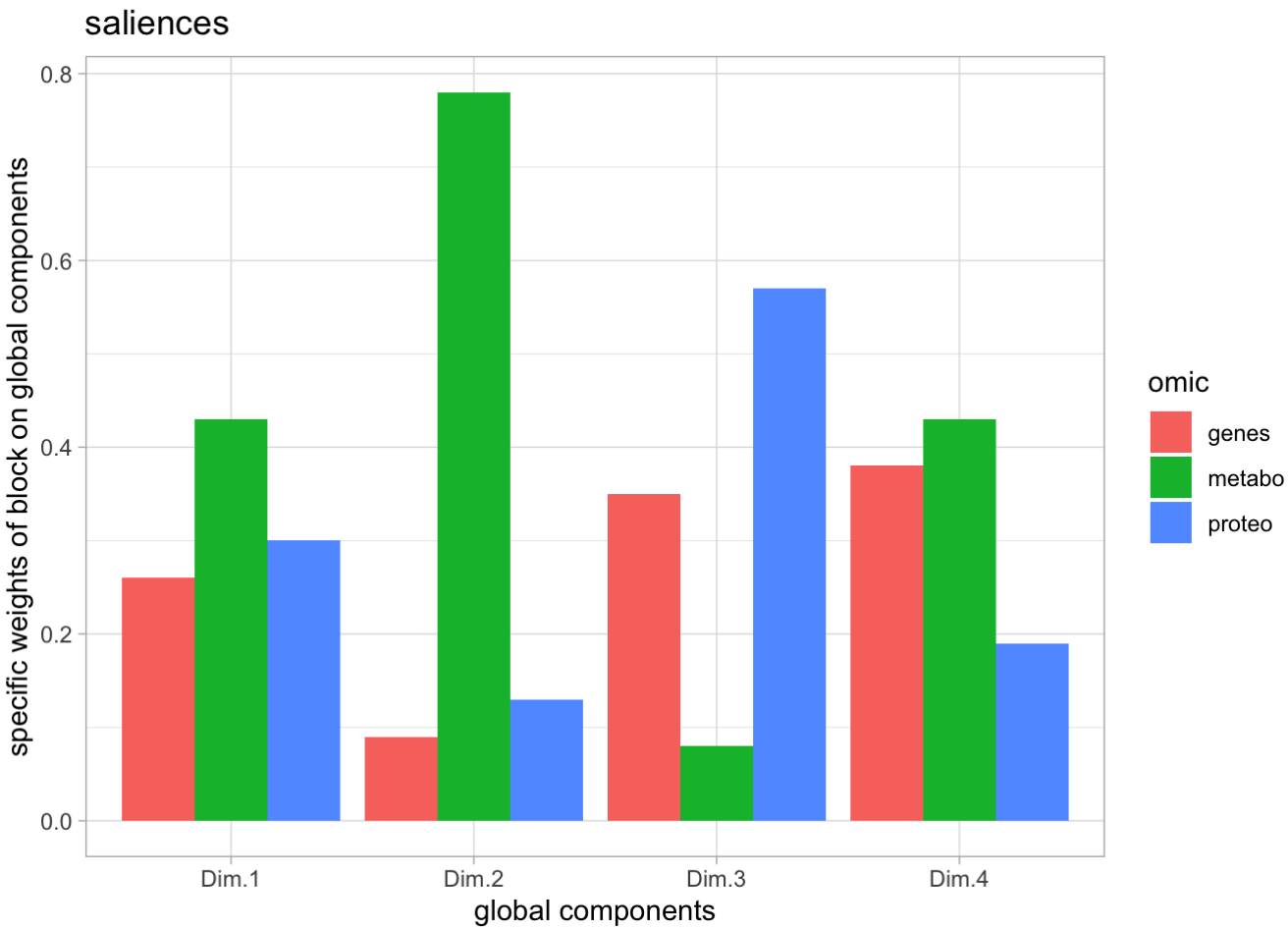
```
# prepare dataset
ComDim_data <- cbind.data.frame(scale(genes), scale(metabo), scale(proteo))
n_group <- c(dim(genes)[[2]], dim(metabo)[[2]], dim(proteo)[[2]])

# run analysis
ComDim_res <- ComDim(X = ComDim_data, group = n_group, plotgraph = F)
```

plot saliences and block contributions

All three omics data contribute to the first dimension, the second dimension is driven by the metabolomics data and the third one by genes and proteomics data.

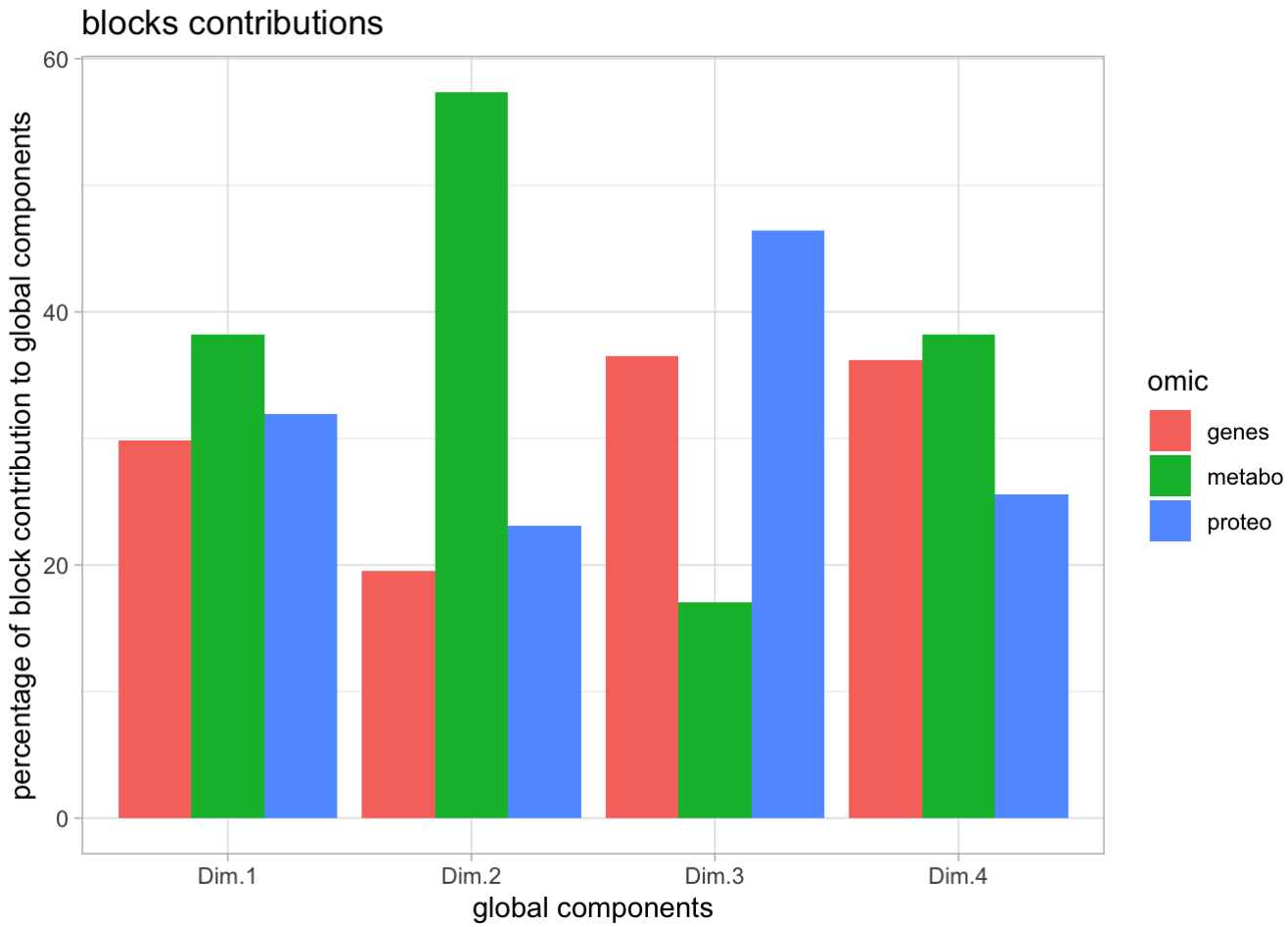
HIDE

```
# saliences

saliences <- ComDim_res$saliences
rownames(saliences) <- c("genes", "metabo", "proteo")
saliences <- as.data.frame(t(saliences[,1:4]))
saliences$Dim <- rownames(saliences)
saliences <- melt(saliences)

ggplot(saliences, aes(x=Dim, y=value, fill=variable)) +
  geom_bar(stat = "identity", position=position_dodge()) +
  theme_light() +
  labs(x = "global components", y = "specific weights of block on global components", fill
= "omic",
       title = "saliences")
```



HIDE

```
# block contributions

contributions <- ComDim_res$contrib
rownames(contributions) <- c("genes", "metabo", "proteo")
contributions <- as.data.frame(t(contributions[,1:4]))
contributions$Dim <- rownames(contributions)
contributions <- melt(contributions)

ggplot(contributions, aes(x=Dim, y=value, fill=variable)) +
  geom_bar(stat = "identity", position=position_dodge()) +
  theme_light() +
  labs(x = "global components", y = "percentage of block contribution to global component
s", fill = "omic",
       title = "blocks contributions")
```

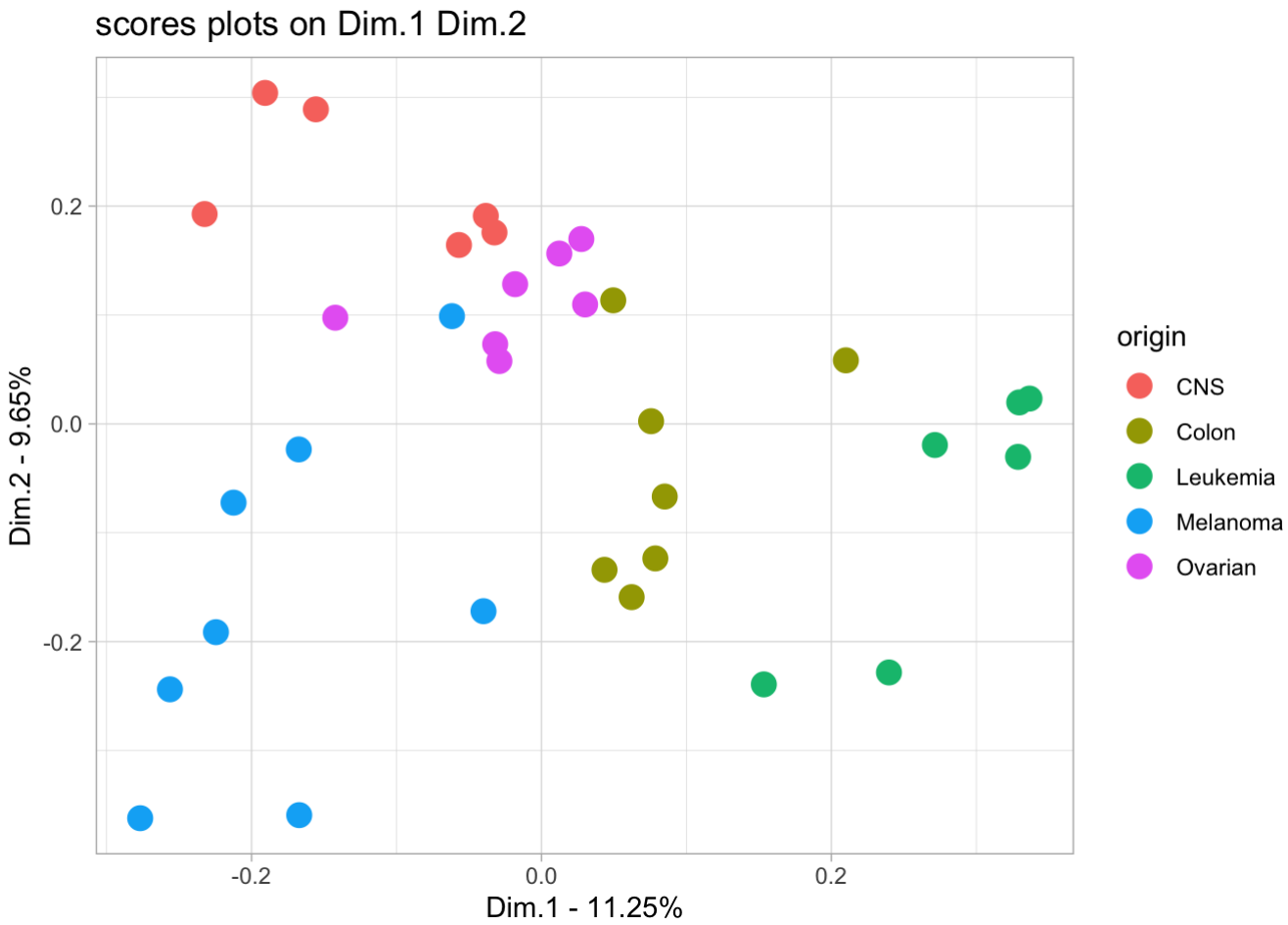


Scores plot

Scores plots on the first 4 dimensions show that samples from the different origins cluster naturally based on genes, metabolomics and proteomics data.

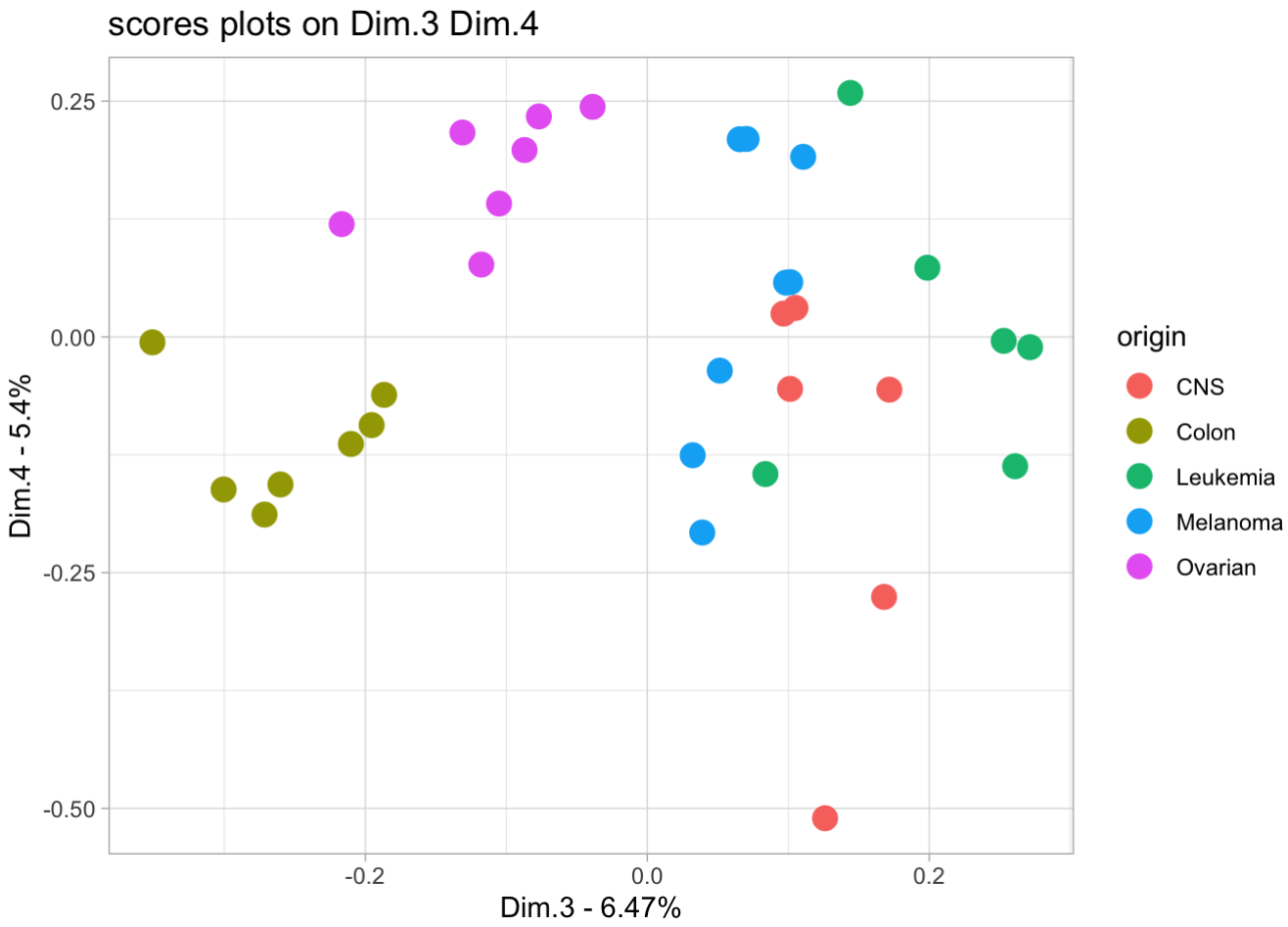
HIDE

```
scores <- data.frame(metadata, ComDim_res$T)

ggplot(scores, aes(x=Dim.1, y=Dim.2, col=origin)) +
  geom_point(size=4) +
  labs(x=paste0("Dim.1 - ", ComDim_res$cumexplained[1,"%explX"], "%"),
       y=paste0("Dim.2 - ", ComDim_res$cumexplained[2,"%explX"], "%"),
       title = "scores plots on Dim.1 Dim.2") +
  theme_light()
```



HIDE

```
ggplot(scores, aes(x=Dim.3, y=Dim.4, col=origin)) +
  geom_point(size=4) +
  labs(x=paste0("Dim.3 - ", ComDim_res$cumexplained[3,"%explX"], "%"),
       y=paste0("Dim.4 - ", ComDim_res$cumexplained[4,"%explX"], "%"),
       title = "scores plots on Dim.3 Dim.4") +
theme_light()
```

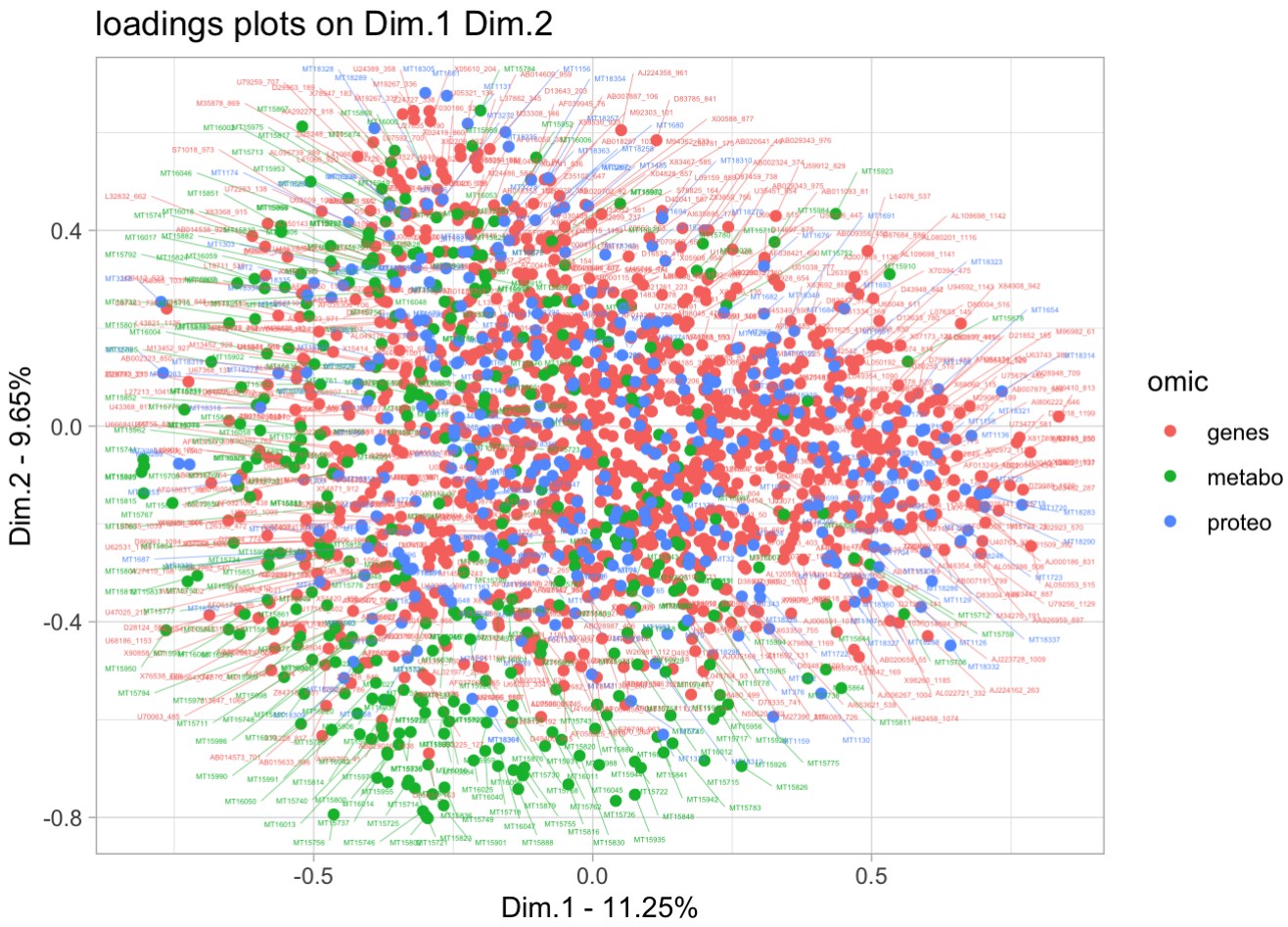


Loadings plot

Here are the loadings plots on the 4 dimensions.

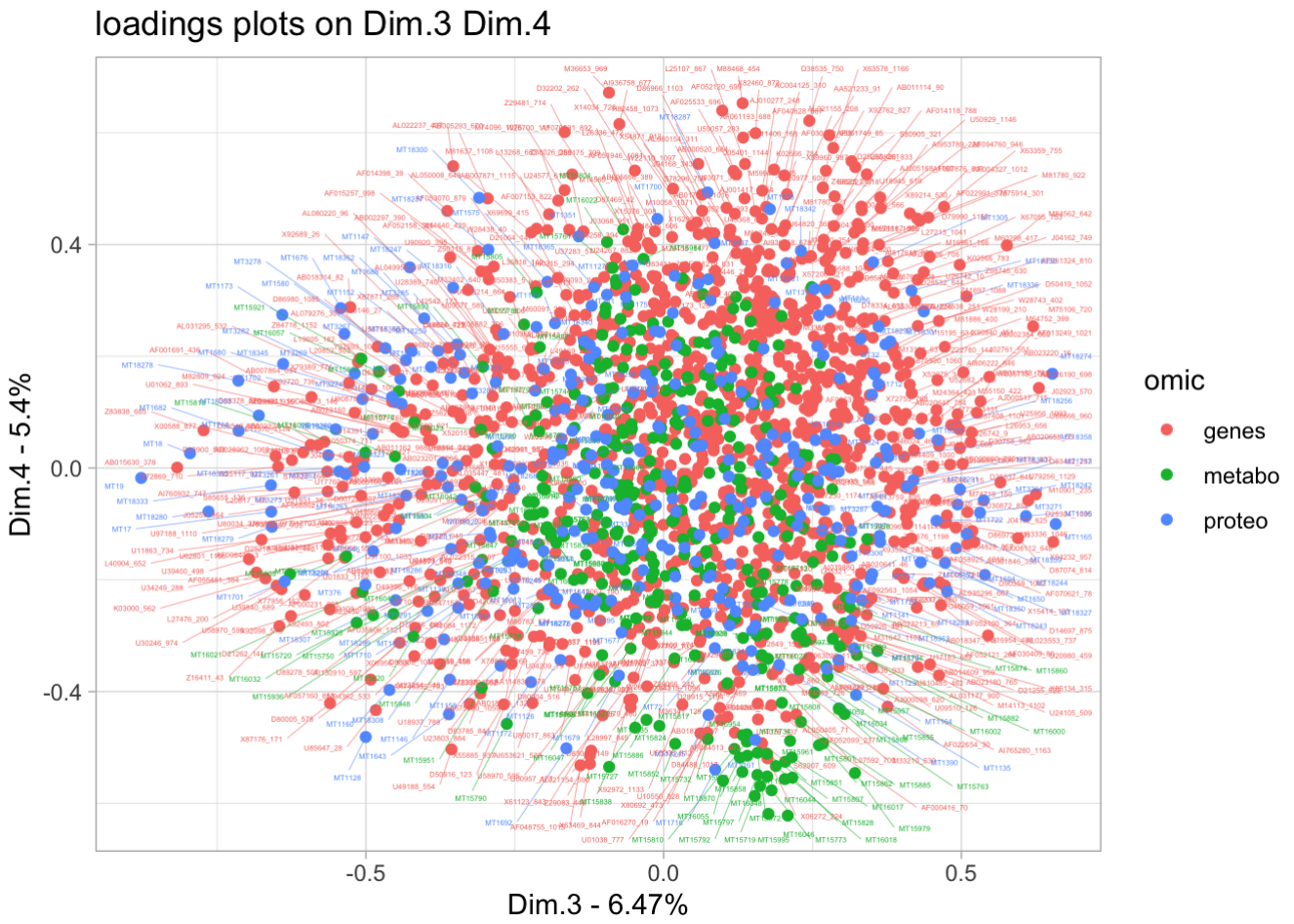
HIDE

```
loadings <- data.frame(ComDim_res$globalcor)
loadings$omic <- c(rep("genes", dim(genes)[[2]]), rep("metabo", dim(metabo)[[2]]), rep("pro
teo", dim(proteo)[[2]]))
loadings$variable <- rownames(loadings)

ggplot(loadings, aes(x=X1, y=X2, col=omic, label=variable)) +
  geom_point(siue = 2) +
  geom_text_repel(size = 1, max.overlaps = 50, segment.size=0.1) +
  labs(x=paste0("Dim.1 - ", ComDim_res$cumexplained[1,"%explX"], "%"),
       y=paste0("Dim.2 - ", ComDim_res$cumexplained[2,"%explX"], "%"),
       title = "loadings plots on Dim.1 Dim.2") +
  theme_light()
```



HIDE

```
ggplot(loadings, aes(x=X3, y=X4, col=omic, label=variable)) +
  geom_point(siue = 2) +
  geom_text_repel(size = 1, max.overlaps = 50, segment.size=0.1) +
  labs(x=paste0("Dim.3 - ", ComDim_res$cumexplained[3,"%explX"], "%"),
       y=paste0("Dim.4 - ", ComDim_res$cumexplained[4,"%explX"], "%"),
       title = "loadings plots on Dim.3 Dim.4") +
  theme_light()
```



Supervised analysis with block.pls

Discriminate samples form colon and ovarian tissues

Run block.plsda analysis with block.plsda() from mixomics package

HIDE

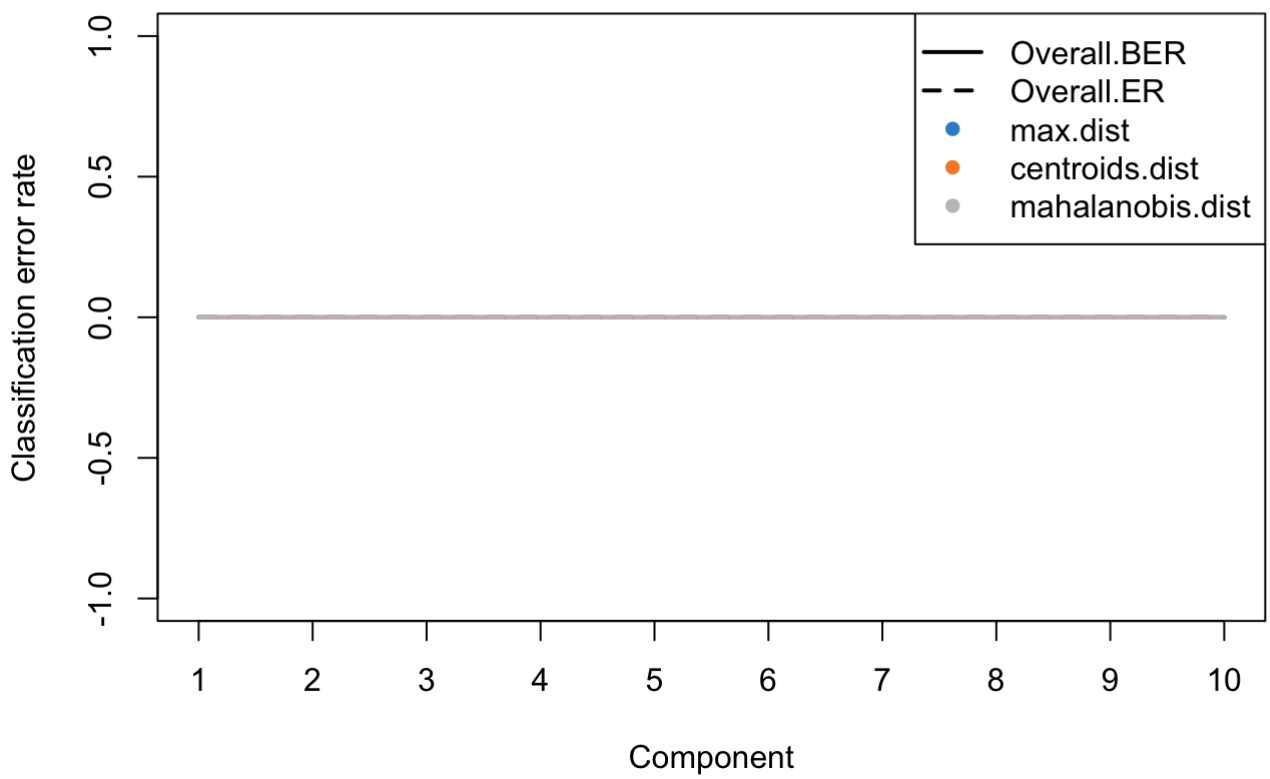
```
# prepare data
blockPLS_data <- lapply(data.pls, scale)
origin <- as.factor(metadata$origin[metadata$origin %in% c("Colon", "Ovarian")])

# run analysis
blockPLS_res <- block.plsda(X = blockPLS_data, Y = origin, design = "full", all.outputs =
T, ncomp = 10, near.zero.var = T)
```

Choose the optimal number of latent variables

Run perf() plot the results with plot() Run the analysis with optimal number of latent variables In this analysis one latent variable is sufficient to discriminate between these two tissue origins. We still compute a model with two latent variables to be able to plot scores and loadings on 2D graphs.

HIDE

```
blockPLS_perf <- perf(blockPLS_res, validation = 'Mfold', folds = 7, nrepeat = 1, auc = TRU
E, cpus=2, progressBar = FALSE)

plot(blockPLS_perf)
```



HIDE

```
blockPLS_res <- block.plsda(X = blockPLS_data, Y = origin, design = "full", all.outputs =
T, ncomp = 2, near.zero.var = T)
```

Permutation test

A permutation test run with DIABLO.test() from RVAideMemoire package shows that the true model is statistically different from randome ones.

HIDE

```
blockPLS_permtest <- DIABLO.test(blockPLS_res, progress = FALSE)
blockPLS_permtest
```

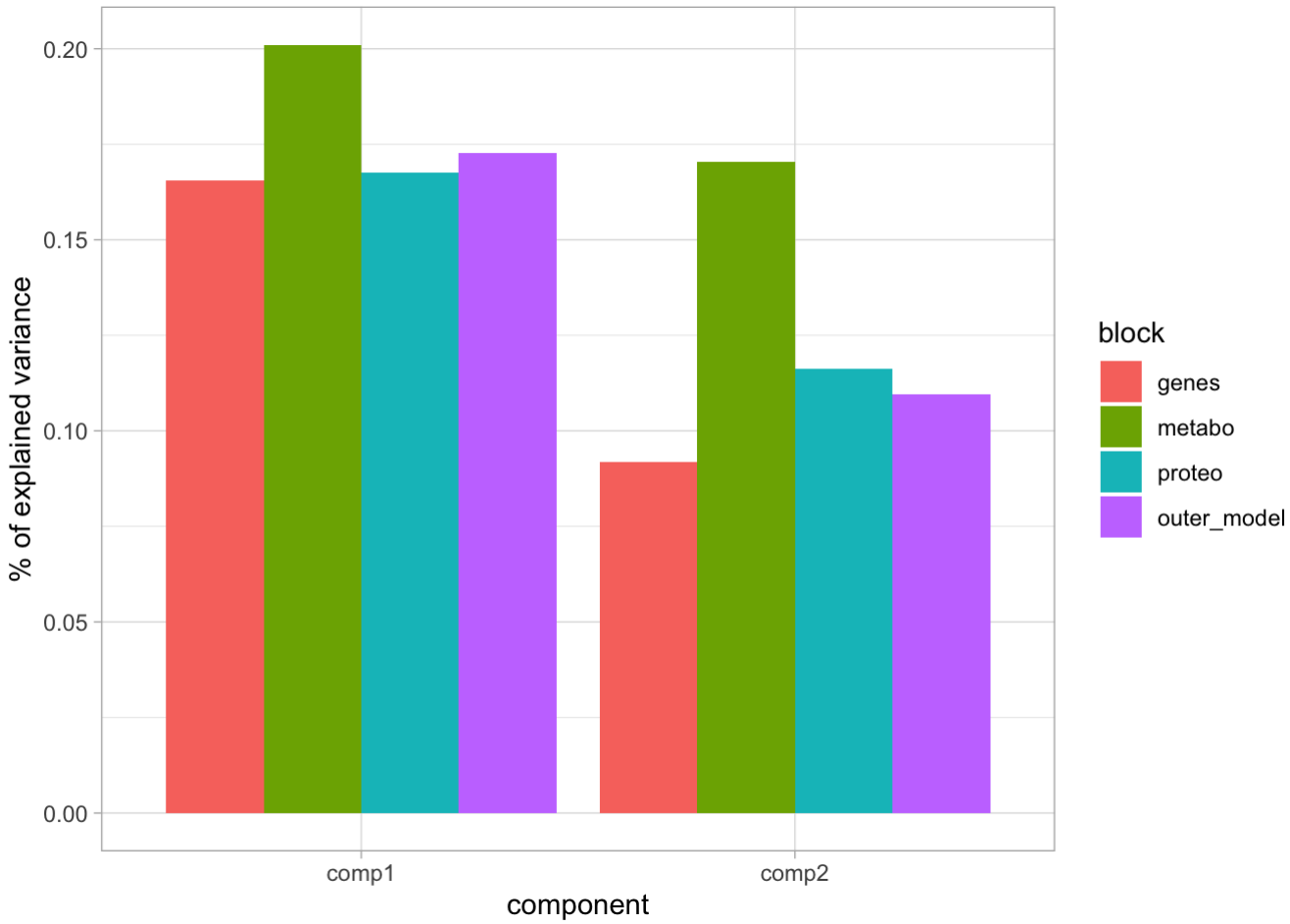
```
##
##  Permutation test based on cross-validation
##
## data:  blockPLS_res
## DIABLO (2 components)
## 999 permutations
## CER = 0.042857, p-value = 0.001
```

Variance explained for each block by each latent variable and globally

Almost the same proportion of each omics data is used to build the first latent variable. Metabolites contributes slightly more.

HIDE

```
blockPLS_expl <- do.call("rbind",blockPLS_res$AVE$AVE_X[1:3])
blockPLS_expl <- rbind(blockPLS_expl, blockPLS_res$AVE[["AVE_outer"]])
rownames(blockPLS_expl)[4] <- "outer_model"
blockPLS_expl <- melt(blockPLS_expl)
colnames(blockPLS_expl) <- c("block", "comp", "value")

ggplot(blockPLS_expl, aes(x=comp, y=value, fill=block)) +
  geom_bar(stat="identity", position=position_dodge()) +
  labs(x="component",
       y="% of explained variance") +
  theme_light()
```

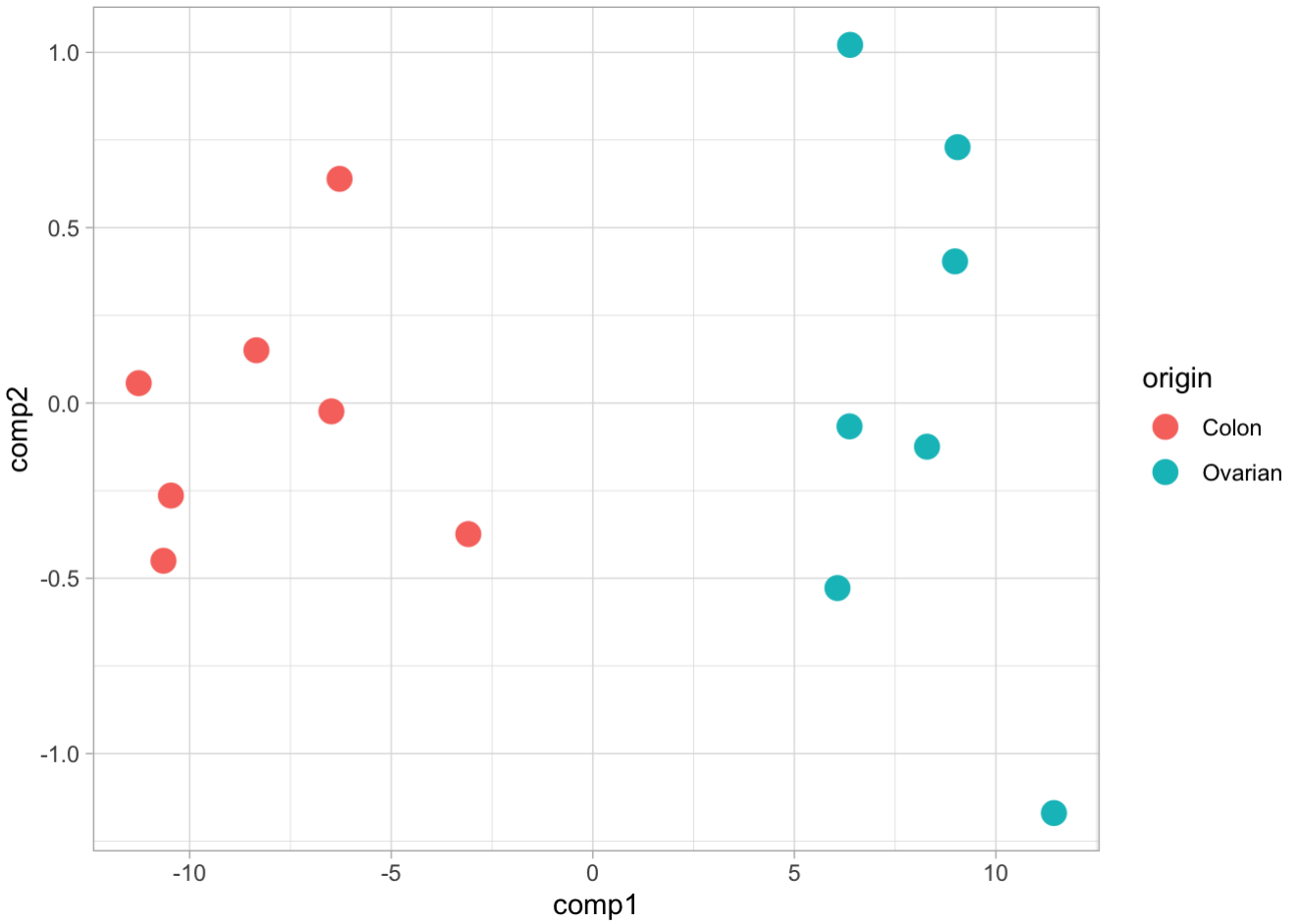


Scores plot

As aimed, the first latent variable discriminates between the two tissues origins, the second latent variable shows orthogonal intra group variation.

HIDE

```
blockPLS_variates.weighted <- blockPLS_res$variates[c("genes", "metabo", "proteo")]
for(omic in c("genes", "metabo", "proteo")){
  for(comp in c("comp1", "comp2")){
      blockPLS_variates.weighted[[omic]][,comp] <- blockPLS_variates.weighted[[omic]][,com
p] * blockPLS_res$weights[omic, comp]
  }
}
blockPLS_scores.weighted <- abind(blockPLS_variates.weighted[c("genes", "metabo", "prote
o")], along = 3)
blockPLS_scores.weighted <- apply(blockPLS_scores.weighted, c(1,2), mean)

blockPLS_scores.weighted <- data.frame(metadata.pls, blockPLS_scores.weighted)

ggplot(blockPLS_scores.weighted, aes(x=comp1, y=comp2, col=origin)) +
  geom_point(size=4) +
  theme_light()
```

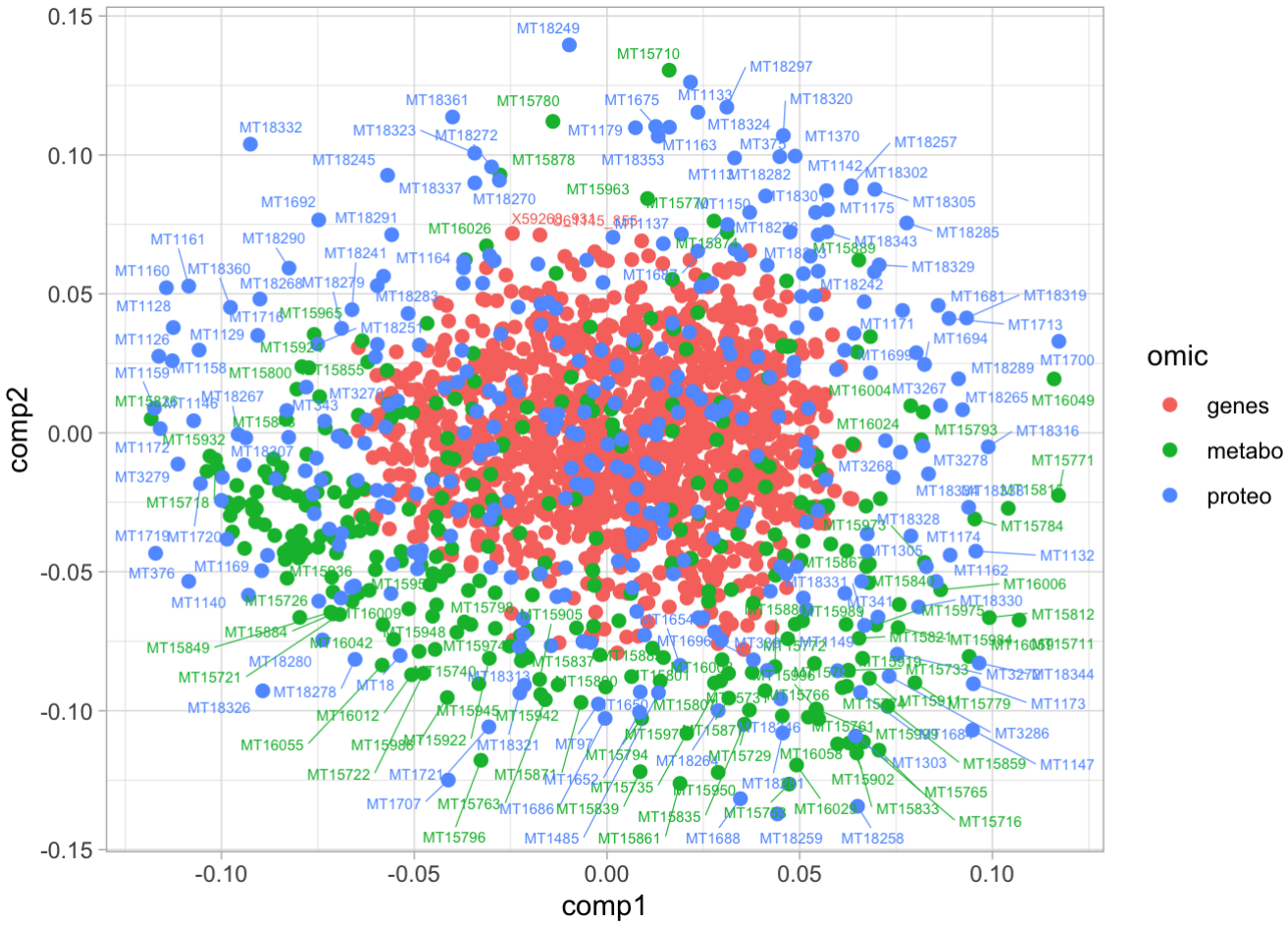


Loadings plot

Loadings are plotted on the 2 first latent variables, a barplot visualisation for each dataset could be used to be able to find the most discriminant variables in each dataset.

HIDE

```
blockPLS_loadings_genes <- blockPLS_res$loadings$genes
blockPLS_loadings_metabo <- blockPLS_res$loadings$metabo
blockPLS_loadings_proteo <- blockPLS_res$loadings$proteo
blockPLS_loadings <- rbind.data.frame(blockPLS_loadings_genes, blockPLS_loadings_metabo, bl
ockPLS_loadings_proteo)
blockPLS_loadings$omic <- c(rep("genes", dim(blockPLS_loadings_genes)[[1]]), rep("metabo",
dim(blockPLS_loadings_metabo)[[1]]), rep("proteo", dim(blockPLS_loadings_proteo)[[1]]))
blockPLS_loadings$variable <- rownames(blockPLS_loadings)

ggplot(blockPLS_loadings, aes(x=comp1, y=comp2, col=omic, label=variable)) +
  geom_point(size=2) +
  geom_text_repel(size=2, max.overlaps = 25, segment.size=0.2) +
  theme_light()
```


comp2
comp1
0.15
0.10
0.05
0.00
-0.05
-0.10
-0.15
-0.10
-0.05
0.00
0.05
0.10
omic
genes
metabo
proteo