RNA-seq sputum sample with PCs and clusters

Carla Casanova

6/6/2022

```
library(dplyr)
# Fir filterbyExpression function
library(edgeR)
# plotRLE
library(EDASeq)
library(tidyverse)
library(SummarizedExperiment)
library(ggplot2)
library(RColorBrewer)
# For aheatmap function
library(NMF)
# For automatic selection of variables to adjust the model
library(leaps)
library(sva)
library(EnhancedVolcano)
library(factoextra)
library(FactoMineR)
library(clusterProfiler)
```

Loading data

```
# load('/Users/carlacasanovasuarez/Documents/Prácticas Radiomics/Radiomic
# features/Results_rfeatures/radar_L1_Norm_adjusted_min_PCA_final.rda')

load("/Users/carlacasanovasuarez/Documents/Master Bioinformatics UAB/Prácticas Radiomics/Radiomic featu

rdr_L1_final
```

```
## class: SummarizedExperiment
## dim: 101 1773
## metadata(1): extractor
## assays(2): values adjusted_min_int_center
## rownames(101): Elongation.original Flatness.original ...
##   Complexity.original Strength.original
## rowData names(4): feature_name image_type feature_description
##   feature_type
## colnames(1773): 23140000005 23140000006 ... 26658003992 26658003997
## colData names(201): filename sample_id ... change.fv1 D_SUBJID
```

```
load("/Users/carlacasanovasuarez/Documents/Master Bioinformatics UAB/Prácticas Radiomics/Transcriptomics

sputum_eset
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 54675 features, 140 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: 30442 85931 ... 12897 (140 total)
##   varLabels: D_SUBJID AGE ... GROUP (83 total)
##   varMetadata: labelDescription
## featureData
##   featureNames: 1 2 ... 3 (54675 total)
##   fvarLabels: PROBEID ENSEMBL ... GENETYPE (9 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: hgu133plus2
```

Checking patients having both: transcriptomic and image data

First of all, remember that patients of `rdr` object are represented with `SUBJID` while the `Expression set` object has `D_SUBJID`. Now, let's check which patients with image data have also transcriptomic data:

```
# Change patient ID in rdr object
colnames(rdr_L1_final) <- colData(rdr_L1_final)$D_SUBJID

# Are all the patients having proper ID conversion? Check for NaN
table(is.na(colnames(rdr_L1_final)))
```

```
##
## FALSE
##  1773
```

```
# Save patients names for each object
patients_eset <- colnames(sputum_eset)
patients_rdr <- colnames(rdr_L1_final)

# See patients with both: radiomic features and transcriptomic data. The result
# are the D_SUBJID for common patients Put IDs as character, otherwise it will
# take IDs as number of row when filtering
common_transcrip <- as.character(intersect(patients_rdr, patients_eset))
```

So, let's filter and check both objects to have the same patients:

```
# Filter patients that are common in both objects
rdr_L1_filt <- rdr_L1_final[, common_transcrip]
sputum_eset_filt <- sputum_eset[, common_transcrip]

# Check that you have the same amount of patients in both
dim(rdr_L1_filt)
```

```
## [1] 101 125
```

```
dim(sputum_eset_filt)
```

```
## Features  Samples
##    54675      125
```

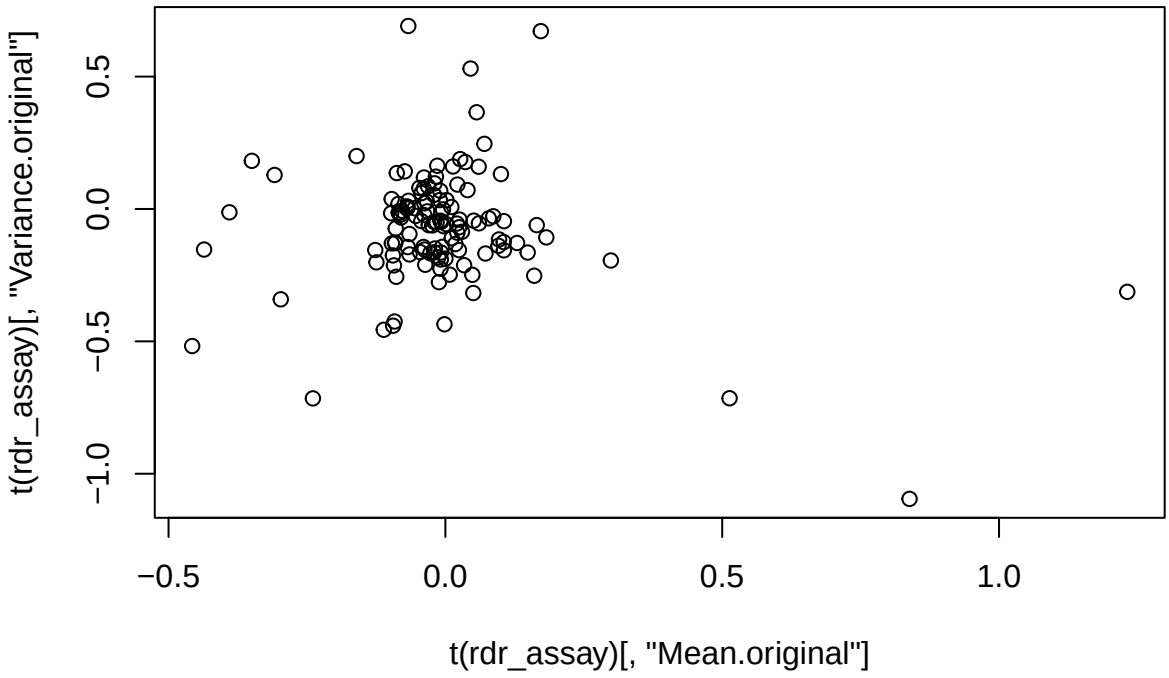
```
# Check that patients are the same and equally ordered
identical(colnames(rdr_L1_filt), colnames(sputum_eset_filt))
```

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

Storing data

Store `colData` and `assay` from `rdr` object:

```
# Store clinical data for patients with radiomic features
rdr_colData <- as.data.frame(colData(rdr_L1_filt)) %>%
    relocate(D_SUBJID, .after = SUBJID)

# Store adjusted and scaled radiomic features
rdr_assay <- assay(rdr_L1_filt, 2)

# Check if radiomic features are the adjusted ones
plot(t(rdr_assay)[, "Mean.original"], t(rdr_assay)[, "Variance.original"])
```



Check that radiomic features are correctly scaled (mean = 0 and sd = 1):

```
# Mean
rowMeans(rdr_assay)
```

```
##                            Elongation.original
##                                  -2.156530e-02
##                              Flatness.original
##                                   1.566650e-01
##                         LeastAxisLength.original
##                                   1.251880e-01
##                         MajorAxisLength.original
##                                  -1.755357e-02
##                 Maximum2DDiameterColumn.original
##                                  -4.916887e-02
##                    Maximum2DDiameterRow.original
##                                   1.672510e-02
##                  Maximum2DDiameterSlice.original
```

```
##                                5.945569e-02
##                   Maximum3DDiameter.original
##                                -2.258093e-02
##                         MeshVolume.original
##                                -1.584766e-02
##                     MinorAxisLength.original
##                                -3.770423e-02
##                          Sphericity.original
##                                -1.715871e-01
##                       SurfaceArea.original
##                                5.605257e-02
##                  SurfaceVolumeRatio.original
##                                1.102105e-01
##                         VoxelVolume.original
##                                -1.583153e-02
##                       10Percentile.original
##                                1.354099e-03
##                       90Percentile.original
##                                -1.759259e-02
##                             Energy.original
##                                -4.247035e-02
##                            Entropy.original
##                                2.615047e-01
##                  InterquartileRange.original
##                                -4.221462e-02
##                           Kurtosis.original
##                                -7.918538e-02
##                            Maximum.original
##                                -3.274319e-02
##               MeanAbsoluteDeviation.original
##                                -5.237051e-02
##                               Mean.original
##                                -5.617889e-03
##                             Median.original
##                                -1.914665e-03
##                            Minimum.original
##                                -2.499702e-17
##                              Range.original
##                                -2.288310e-02
##         RobustMeanAbsoluteDeviation.original
##                                -4.489109e-02
##                    RootMeanSquared.original
##                                -7.342097e-02
##                           Skewness.original
##                                -8.302687e-02
##                       TotalEnergy.original
##                                -4.247035e-02
##                        Uniformity.original
##                                -2.553301e-01
##                           Variance.original
##                                -7.058942e-02
##                     Autocorrelation.original
##                                -4.773641e-02
##                    ClusterProminence.original
```

```
##                                 2.970637e-01
##                        ClusterShade.original
##                                 9.473455e-02
##                      ClusterTendency.original
##                                 2.698746e-01
##                            Contrast.original
##                                 1.914294e-01
##                         Correlation.original
##                                 3.790664e-02
##                   DifferenceAverage.original
##                                 1.914294e-01
##                   DifferenceEntropy.original
##                                 2.247047e-01
##                  DifferenceVariance.original
##                                 2.122828e-01
##                                  Id.original
##                                -1.914294e-01
##                                 Idm.original
##                                -1.914294e-01
##                                Idmn.original
##                                -1.914294e-01
##                                 Idn.original
##                                -1.914294e-01
##                                Imc1.original
##                                -1.131492e-01
##                                Imc2.original
##                                 2.878436e-01
##                      InverseVariance.original
##                                 1.914294e-01
##                         JointAverage.original
##                                -4.011648e-02
##                          JointEnergy.original
##                                -2.383427e-01
##                         JointEntropy.original
##                                 2.438866e-01
##                                 MCC.original
##                                 3.791481e-02
##                  MaximumProbability.original
##                                -2.189520e-01
##                            SumAverage.original
##                                -4.011648e-02
##                            SumEntropy.original
##                                 2.533335e-01
##                            SumSquares.original
##                                 2.482110e-01
##               GrayLevelNonUniformity.original
##                                 1.825393e-01
##     GrayLevelNonUniformityNormalized.original
##                                -1.940287e-01
##                    GrayLevelVariance.original
##                                 1.940287e-01
##             HighGrayLevelRunEmphasis.original
##                                -8.081454e-03
##                       LongRunEmphasis.original
```

```
##                                  -2.263255e-01
##         LongRunHighGrayLevelEmphasis.original
##                                  -1.613383e-01
##          LongRunLowGrayLevelEmphasis.original
##                                  -2.028831e-01
##              LowGrayLevelRunEmphasis.original
##                                   8.081454e-03
##                            RunEntropy.original
##                                  -2.199644e-01
##               RunLengthNonUniformity.original
##                                   1.513838e-01
##     RunLengthNonUniformityNormalized.original
##                                   1.914701e-01
##                         RunPercentage.original
##                                   1.928448e-01
##                           RunVariance.original
##                                  -2.145780e-01
##                      ShortRunEmphasis.original
##                                   2.084097e-01
##        ShortRunHighGrayLevelEmphasis.original
##                                   1.880902e-01
##         ShortRunLowGrayLevelEmphasis.original
##                                   1.197142e-01
##            HighGrayLevelZoneEmphasis.original
##                                   9.259274e-02
##                     LargeAreaEmphasis.original
##                                  -1.007597e-01
##       LargeAreaHighGrayLevelEmphasis.original
##                                  -1.203463e-01
##        LargeAreaLowGrayLevelEmphasis.original
##                                  -8.711726e-02
##             LowGrayLevelZoneEmphasis.original
##                                  -9.259274e-02
##                SizeZoneNonUniformity.original
##                                   1.762171e-03
##      SizeZoneNonUniformityNormalized.original
##                                  -6.411542e-02
##                    SmallAreaEmphasis.original
##                                   1.481321e-01
##       SmallAreaHighGrayLevelEmphasis.original
##                                   1.922800e-01
##        SmallAreaLowGrayLevelEmphasis.original
##                                  -1.798917e-02
##                           ZoneEntropy.original
##                                   8.351342e-02
##                         ZonePercentage.original
##                                  -3.447834e-02
##                          ZoneVariance.original
##                                  -1.221448e-01
##                    DependenceEntropy.original
##                                   2.358296e-01
##              DependenceNonUniformity.original
##                                  -1.948360e-01
##    DependenceNonUniformityNormalized.original
```

```
##                                 -2.250917e-01
##                   DependenceVariance.original
##                                  2.371995e-01
##                HighGrayLevelEmphasis.original
##                                 -3.812955e-02
##             LargeDependenceEmphasis.original
##                                 -1.954696e-01
## LargeDependenceHighGrayLevelEmphasis.original
##                                 -9.815096e-02
##  LargeDependenceLowGrayLevelEmphasis.original
##                                 -1.422526e-02
##                 LowGrayLevelEmphasis.original
##                                  3.812955e-02
##             SmallDependenceEmphasis.original
##                                  8.584284e-02
## SmallDependenceHighGrayLevelEmphasis.original
##                                  1.910997e-01
##  SmallDependenceLowGrayLevelEmphasis.original
##                                  2.521942e-02
##                               Busyness.original
##                                  -1.757089e-02
##                             Coarseness.original
##                                  -1.429322e-01
##                             Complexity.original
##                                  1.928846e-01
##                               Strength.original
##                                  -4.868507e-02
```

```
# Sd
rowSds(rdr_assay)
```

```
##                             Elongation.original
##                                    9.707006e-01
##                               Flatness.original
##                                    1.070414e+00
##                       LeastAxisLength.original
##                                    1.026254e+00
##                       MajorAxisLength.original
##                                    9.566020e-01
##                Maximum2DDiameterColumn.original
##                                    9.446964e-01
##                   Maximum2DDiameterRow.original
##                                    9.316520e-01
##                 Maximum2DDiameterSlice.original
##                                    9.963886e-01
##                      Maximum3DDiameter.original
##                                    9.598362e-01
##                             MeshVolume.original
##                                    9.635952e-01
##                       MinorAxisLength.original
##                                    9.221252e-01
##                             Sphericity.original
##                                    1.262051e+00
##                            SurfaceArea.original
##                                    1.037106e+00
```

```
##                   SurfaceVolumeRatio.original
##                                  1.178687e+00
##                         VoxelVolume.original
##                                  9.636134e-01
##                       10Percentile.original
##                                  1.620134e-01
##                       90Percentile.original
##                                  2.601059e-01
##                            Energy.original
##                               3.468056e-01
##                           Entropy.original
##                               1.233727e+00
##                  InterquartileRange.original
##                                  3.555514e-01
##                          Kurtosis.original
##                               9.747746e-01
##                           Maximum.original
##                               6.786196e-01
##                MeanAbsoluteDeviation.original
##                                  2.609273e-01
##                             Mean.original
##                               1.805349e-01
##                           Median.original
##                               1.703771e-01
##                          Minimum.original
##                               1.030106e-16
##                            Range.original
##                               4.742640e-01
##          RobustMeanAbsoluteDeviation.original
##                                  3.410841e-01
##                    RootMeanSquared.original
##                                  2.955809e-01
##                          Skewness.original
##                               9.563223e-01
##                       TotalEnergy.original
##                                  3.468056e-01
##                        Uniformity.original
##                                  1.231839e+00
##                          Variance.original
##                               2.218469e-01
##                     Autocorrelation.original
##                                  6.682197e-01
##                   ClusterProminence.original
##                                  1.245496e+00
##                       ClusterShade.original
##                                  1.035368e+00
##                     ClusterTendency.original
##                                  1.254930e+00
##                          Contrast.original
##                                  1.137487e+00
##                       Correlation.original
##                                  9.597100e-01
##                    DifferenceAverage.original
##                                  1.137487e+00
```

```
##                     DifferenceEntropy.original
##                                   1.170479e+00
##                    DifferenceVariance.original
##                                   1.156551e+00
##                                     Id.original
##                                   1.137487e+00
##                                    Idm.original
##                                   1.137487e+00
##                                   Idmn.original
##                                   1.137487e+00
##                                    Idn.original
##                                   1.137487e+00
##                                   Imc1.original
##                                   4.695869e-01
##                                   Imc2.original
##                                   1.096457e+00
##                       InverseVariance.original
##                                   1.137487e+00
##                          JointAverage.original
##                                   6.531167e-01
##                           JointEnergy.original
##                                   1.201501e+00
##                          JointEntropy.original
##                                   1.206736e+00
##                                    MCC.original
##                                   9.598022e-01
##                    MaximumProbability.original
##                                   1.196900e+00
##                             SumAverage.original
##                                   6.531167e-01
##                             SumEntropy.original
##                                   1.218795e+00
##                             SumSquares.original
##                                   1.220746e+00
##                GrayLevelNonUniformity.original
##                                   1.134556e+00
##     GrayLevelNonUniformityNormalized.original
##                                   9.535484e-01
##                     GrayLevelVariance.original
##                                   9.535484e-01
##              HighGrayLevelRunEmphasis.original
##                                   9.622660e-01
##                       LongRunEmphasis.original
##                                   1.056849e+00
##          LongRunHighGrayLevelEmphasis.original
##                                   1.009583e+00
##           LongRunLowGrayLevelEmphasis.original
##                                   1.020514e+00
##               LowGrayLevelRunEmphasis.original
##                                   9.622660e-01
##                             RunEntropy.original
##                                   1.149065e+00
##                RunLengthNonUniformity.original
##                                   1.116724e+00
```

```
##     RunLengthNonUniformityNormalized.original
##                                  1.124636e+00
##                         RunPercentage.original
##                                  1.136904e+00
##                           RunVariance.original
##                                  1.166746e+00
##                       ShortRunEmphasis.original
##                                  1.082163e+00
##          ShortRunHighGrayLevelEmphasis.original
##                                  8.015270e-01
##           ShortRunLowGrayLevelEmphasis.original
##                                  9.706545e-01
##              HighGrayLevelZoneEmphasis.original
##                                  5.373320e-01
##                     LargeAreaEmphasis.original
##                                  1.099895e+00
##         LargeAreaHighGrayLevelEmphasis.original
##                                  9.623139e-01
##          LargeAreaLowGrayLevelEmphasis.original
##                                  1.116648e+00
##               LowGrayLevelZoneEmphasis.original
##                                  5.373320e-01
##                  SizeZoneNonUniformity.original
##                                  9.351906e-01
##        SizeZoneNonUniformityNormalized.original
##                                  8.503077e-01
##                     SmallAreaEmphasis.original
##                                  9.715657e-01
##         SmallAreaHighGrayLevelEmphasis.original
##                                  7.618756e-01
##          SmallAreaLowGrayLevelEmphasis.original
##                                  7.376781e-01
##                            ZoneEntropy.original
##                                  7.923056e-01
##                         ZonePercentage.original
##                                  9.657033e-01
##                           ZoneVariance.original
##                                  8.971247e-01
##                     DependenceEntropy.original
##                                  1.168398e+00
##                DependenceNonUniformity.original
##                                  1.083133e+00
##      DependenceNonUniformityNormalized.original
##                                  1.130122e+00
##                    DependenceVariance.original
##                                  1.172274e+00
##                  HighGrayLevelEmphasis.original
##                                  6.517226e-01
##                LargeDependenceEmphasis.original
##                                  1.139273e+00
## LargeDependenceHighGrayLevelEmphasis.original
##                                  8.069877e-01
##  LargeDependenceLowGrayLevelEmphasis.original
##                                  6.057517e-01
```

```
##                 LowGrayLevelEmphasis.original
##                                     6.517226e-01
##              SmallDependenceEmphasis.original
##                                     1.002413e+00
## SmallDependenceHighGrayLevelEmphasis.original
##                                     1.078358e+00
##  SmallDependenceLowGrayLevelEmphasis.original
##                                     9.905405e-01
##                               Busyness.original
##                                     3.085053e-01
##                             Coarseness.original
##                                     9.948089e-01
##                             Complexity.original
##                                     1.140754e+00
##                               Strength.original
##                                     1.180427e+00
```

Storing **phenotype**, **expression** and **annotation** data from `Expression set` object:

```
# Transcriptomic data
counts <- exprs(sputum_eset_filt)

# Phenotipic data Retrieves the subjects' phenotypes in an AnnotatedDataFrame
# object
pheno_blood <- phenoData(sputum_eset_filt)
# pData() converts the phenotipic data to a data frame
phenoDataFrame <- pData(sputum_eset_filt)

# Annotation data
annotation.sputum <- fData(sputum_eset_filt)
```

Exploring variables

```
# Check groups in expression set
table(phenoDataFrame$GROUP)
```

```
##
## Moderate
##      125
```

```
table(is.na(phenoDataFrame$GROUP))
```

```
##
## FALSE
##   125
```

```
table(phenoDataFrame$GOLDCD)
```

```
##
##  2  3  4
## 61 52 12
```

```
table(is.na(phenoDataFrame$GOLDCD))
```

```
##
## FALSE
##   125
```

```
# Check groups in rdr object
table(rdr_colData$GOLDCD)
```

```
##
##  2  3  4
## 61 52 11
```

```
table(is.na(rdr_colData$GOLDCD))
```

```
##
## FALSE  TRUE
##   124     1
```

Dictionary for GOLD classification:

- 0=GOLD 0: Control (FEV1 $>$ 80% FEV1/FVC $>$ 0.7)
- 1=GOLD 1: (FEV1 $>$ 80% FEV1/FVC $<$ 0.7)
- 2=GOLD 2: (50% $<$ FEV1 $<$ 80% FEV1/FVC $<$ 0.7)
- 3=GOLD 3: (30% $<$ FEV1 $<$ 50% FEV1/FVC $<$ 0.7)
- 4=GOLD 4: (FEV1 $<$ 30% FEV1/FVC $<$ 0.7)

Let's transform three stages into a binary variable (moderate, severe):

```
# Group by tumor stages
stage <- phenoDataFrame$GOLDCD
ids.moderate <- grep(paste(2, sep = "|"), stage)

ids.severe <- grep(paste(3, 4, sep = "|"), stage)

phenoDataFrame$GROUP_bin <- rep(NA, ncol(sputum_eset_filt))
phenoDataFrame$GROUP_bin[ids.moderate] <- "Moderate"
phenoDataFrame$GROUP_bin[ids.severe] <- "Severe"
phenoDataFrame$GROUP_bin <- as.factor(phenoDataFrame$GROUP_bin)
```

Exploring gene counts

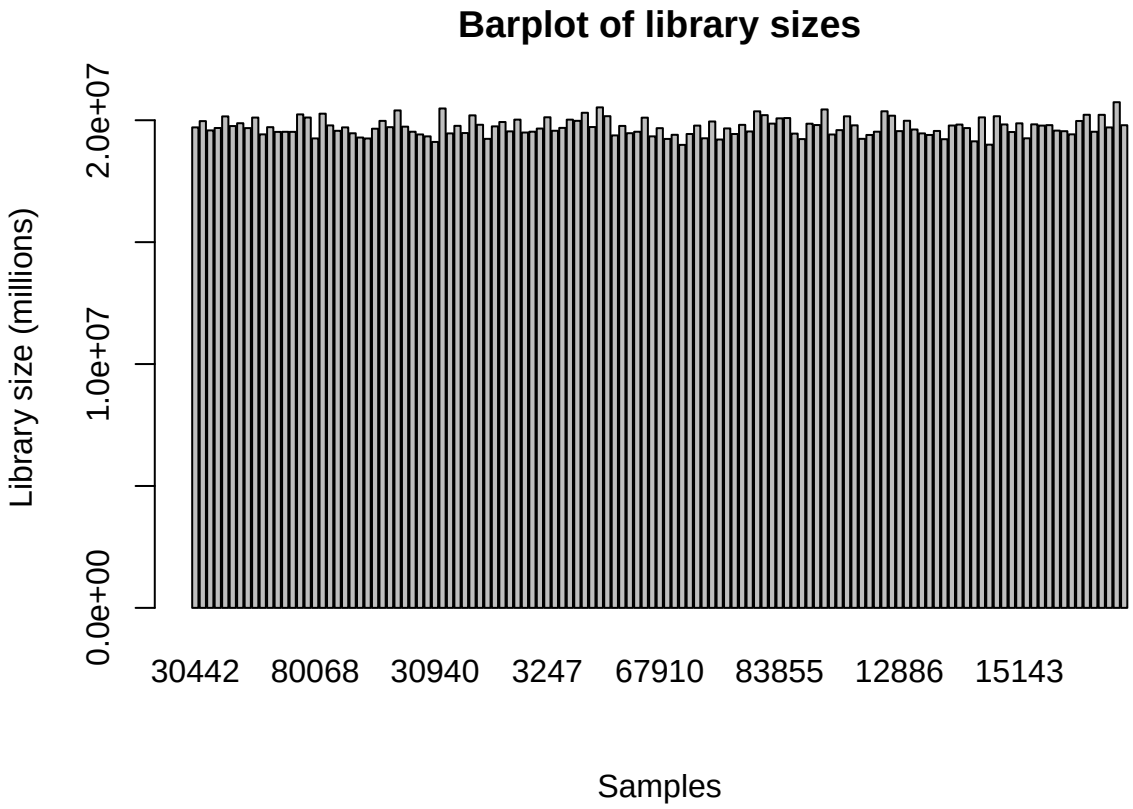
First, analyze **library sizes** between samples:

```
# Check that the sum of each column is not equal to 10^6 since values have RPKM
# normalization
colSums(counts, na.rm = TRUE)
```

```
##    30442    85931     4428    41420    15405    26383    55856    92849
## 19706718 19963650 19583022 19681032 20156789 19760854 19876753 19677396
##    40819    77811    14804    51797    55255    29240    95106    87587
## 20109294 19420063 19715320 19520146 19524860 19523141 20239253 20111108
##    80068    73795    10788    47780    66276     3269    58758    42518
## 19254293 20268985 19787441 19566115 19706560 19464859 19289606 19251357
##    98007    90488    45977    82970     1466    38458    75451    12443
## 19651455 19973233 19715459 20401671 19736282 19528826 19416495 19338181
##    30940     4925    15902    71391    74976    93472    85953    41442
## 19109799 20482398 19458917 19769844 19474433 20199927 19813618 19236087
##    96931    26404    85352     3848    14826    92270    47758     3247
## 19748548 19926680 19543077 20025003 19495217 19529157 19659007 20121532
##    40240    51217    88210    91669    39639    46556    83549    50015
## 19568171 19685322 20023485 19975672 20306060 19720955 20526897 20165591
##     5504    24000    60993    90466    82948    38437    67910    23399
```

```
## 19376885 19766381 19468447 19527306 20105871 19338203 19674709 19232600
##    30317    48813    96783     3701    22197    44152    88062     6558
## 19402503 18988954 19435662 19779772 19259010 19947957 19209132 19664389
##    60845    48064    66561    85057    51523    24907    83855    94832
## 19435145 19810948 19539597 20364857 20208068 19861150 20077212 20089836
##    98291    21607    87472    62364    99357    73342    32743    25224
## 19451245 19226375 19857970 19802496 20442877 19414913 19593420 20160333
##    43720    62217    36202    39660    83412    75893    12886    60856
## 19791153 19236189 19395744 19529910 20366807 20185198 19553787 19978327
##    82811     1307    12285    67774    41759    60255    70484    25973
## 19619724 19459397 19396308 19559880 19222063 19782827 19822278 19678680
##    44469    99958    36950    55446    15744    74691    15143    89128
## 19135127 20116964 18997699 20163810 19827416 19516010 19873285 19258061
##    40557    51534     2963    72888    53157     1127    23082    41578
## 19831552 19780019 19798093 19574370 19550240 19421511 19968746 20223520
##    97067    85489    26098     3542    77527
## 19527481 20222104 19704910 20739475 19795543
```

```r
# Number of reads in that sample (library size)
lib.size <- colSums(counts, na.rm = TRUE)

barplot(lib.size, names = colnames(counts))
mtext(side = 1, text = "Samples", line = 4)
mtext(side = 2, text = "Library size (millions)", line = 3)
title("Barplot of library sizes")
```



Gene selection

Now, let's select genes for the analysis using `filterByExpr` from `edgeR` package (based on minimum library size) and

```
# Remove data with NA values
counts.ok <- counts[complete.cases(counts), ]
dim(counts.ok)
```

```
## [1] 54672   125
```

```
# Create groups
group <- phenoDataFrame$GOLDCD
group <- factor(group)

# Are there negative counts?
table(apply(counts.ok, 1, function(x) {
    any(x < 0)
}))
```

```
##
## FALSE  TRUE
## 48565  6107
```

```
# If yes, then filter those genes
counts.ok <- counts.ok[!apply(counts.ok, 1, function(x) {
    any(x < 0)
}), ]

# Use edgeR package to create a DGEList object with gene matrix to filter genes
y = DGEList(counts = counts.ok, group = group)

# Check rows of clinical data and DGEList object before adding more variables
identical(rownames(y$samples), rownames(phenoDataFrame))
```

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

```
identical(rownames(y), rownames(counts.ok))
```

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

```
y$samples$Stage <- phenoDataFrame$GROUP_bin
y$samples$Cough <- phenoDataFrame$COUGH
y$samples$Sex <- phenoDataFrame$SEX

# Filter genes
keep <- filterByExpr(y)
table(keep)
```

```
## keep
## FALSE  TRUE
##    40 48525
```

```
y <- y[keep, , keep.lib.sizes = FALSE]
counts.ok <- counts.ok[rownames(y), ]

dim(counts.ok)
```

```
## [1] 48525   125
```

In this case, all the 40 genes were removed for the analysis.

```
# Filter data frame with features information by removing genes that are not
# desired for the current analysis
identical(rownames(counts.ok), rownames(annotation.sputum))
```

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

```
filter_rows <- intersect(rownames(counts.ok), rownames(annotation.sputum))
annotation.sputum.ok <- annotation.sputum[filter_rows, ]
identical(rownames(counts.ok), annotation.sputum.ok$PROBEID)
```

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

```
rownames(annotation.sputum.ok) <- annotation.sputum.ok$PROBEID
```

Save objects for future analyses using only sputum samples:

```
save(rdr_assay, file = "/Users/carlacasanovasuarez/Documents/Master Bioinformatics UAB/Prácticas Radiom
```

```
save(sputum_eset_filt, file = "/Users/carlacasanovasuarez/Documents/Master Bioinformatics UAB/Prácticas
```

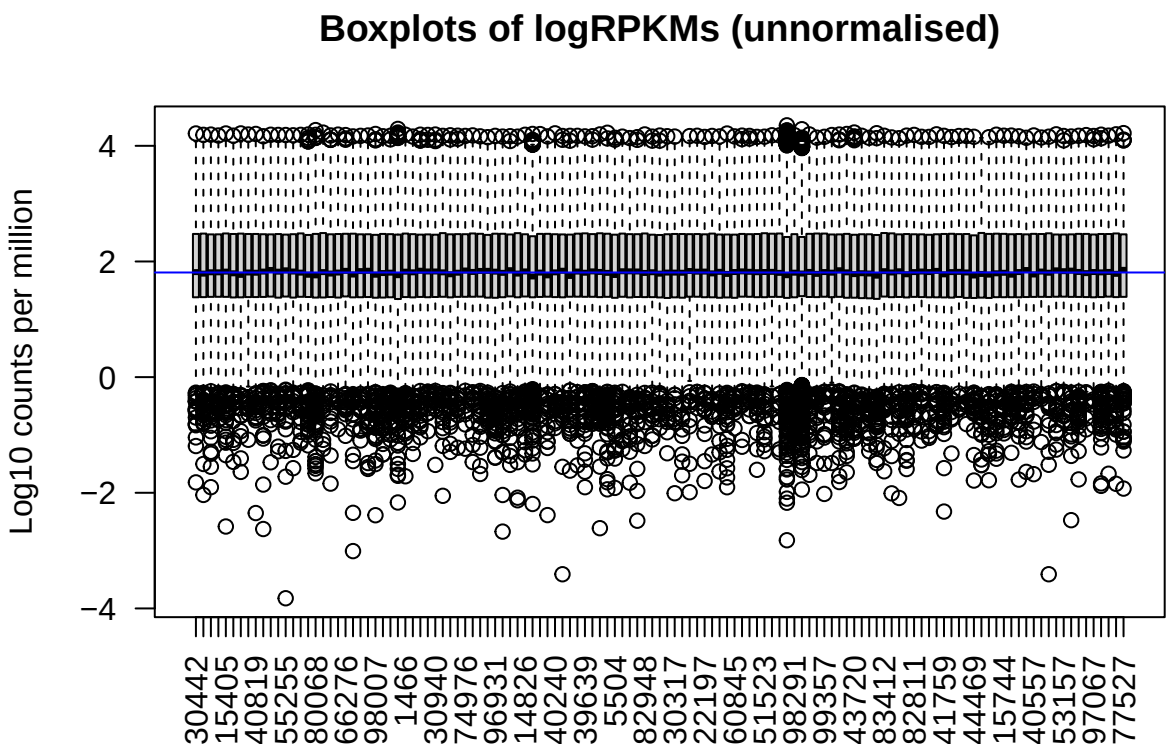
```
save(counts.ok, file = "/Users/carlacasanovasuarez/Documents/Master Bioinformatics UAB/Prácticas Radiom
```

```
save(phenoDataFrame, file = "/Users/carlacasanovasuarez/Documents/Master Bioinformatics UAB/Prácticas Ra
```

```
save(annotation.sputum.ok, file = "/Users/carlacasanovasuarez/Documents/Master Bioinformatics UAB/Práct
```

From the boxplots we see that overall the density distributions of raw log-intensities are not identical but still not very different. If a sample is really far above or below the blue horizontal line we may need to investigate that sample further.

```
# logsignal values
counts.log <- log10(counts.ok)

boxplot(counts.log, xlab = "", ylab = "Log10 counts per million", las = 2)
abline(h = median(counts.log, na.rm = TRUE), col = "blue")
title("Boxplots of logRPKMs (unnormalised)")
```



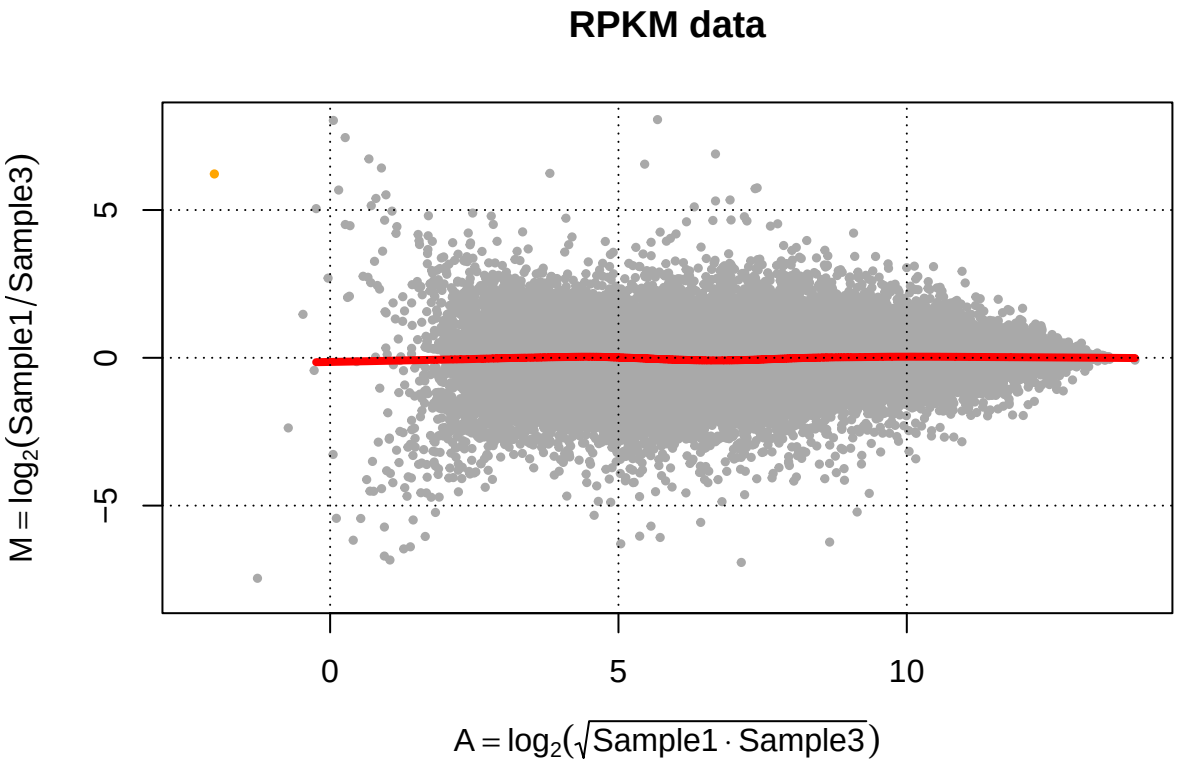
Check normalization

First of all, should the following should be considered. TMM normalizes the library sizes to produce **effective library sizes**. **CPM** values are counts normalized by the effective library sizes. **RPKM** values are counts normalized by effective library sizes and by gene/feature length (Biostars). Hence, RPKM has been computed using by using **normalization factors** that correct the library size-scaled values for the compositional component. This here is what the **Trimmed Mean of M-values (TMM)** does. So, in practce steps when using `edgeR` include:

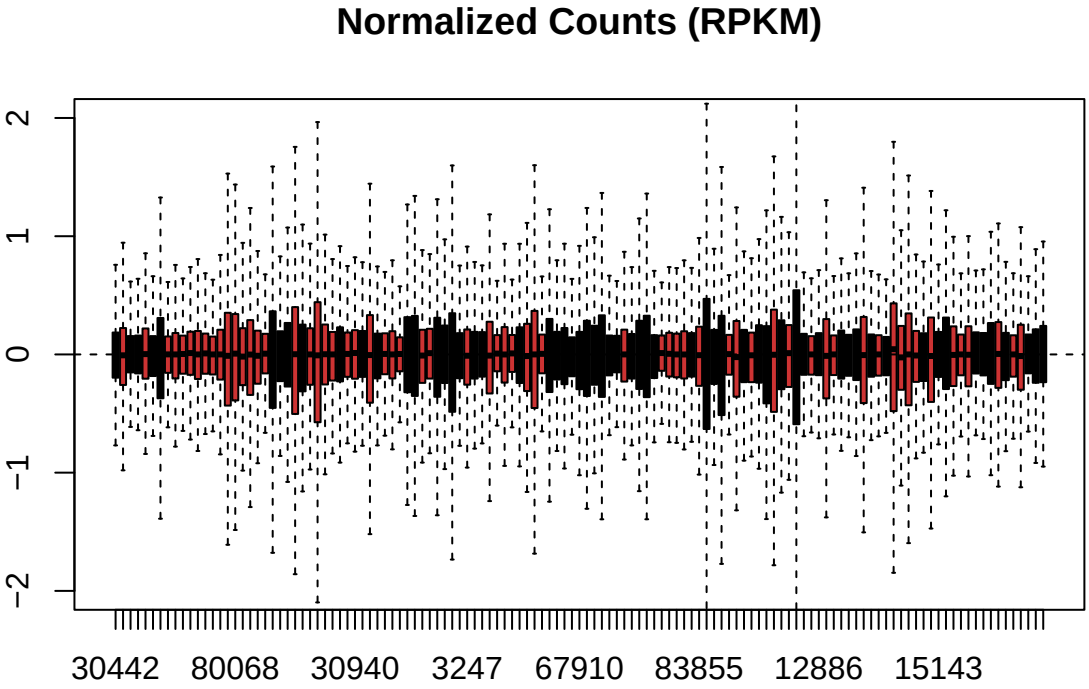
```
#/ make the DGEList:
y <- DGEList(...)

#/ calculate TMM normalization factors:
y <- calcNormFactors(y)

#/ get the normalized counts:
cpms <- cpm(y, log=FALSE)
```

Let's check **RPKM** counts with a MA plot in order to see if data has been properly normalized:

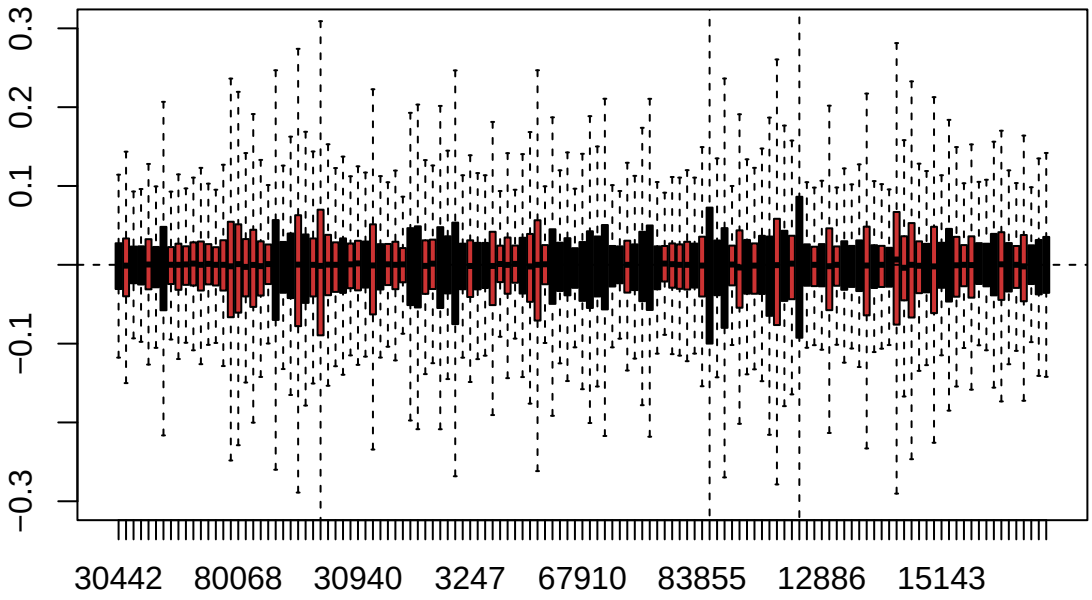
```
maPlot(counts.ok[, 1], counts.ok[, 2], pch = 19, cex = 0.5, ylim = c(-8, 8), allCol = "darkgray",
    lowess = TRUE, xlab = expression(A == log[2](sqrt(Sample1 %.% Sample3))), ylab = expression(M ==
        log[2](Sample1/Sample3)))
grid(col = "black")
title("RPKM data")
```




```
color = ifelse(phenoDataFrame$GROUP_bin == "Moderate", "black", "brown3")

plotRLE(counts.ok, outline = FALSE, ylim = c(-2, 2), col = color, main = "Normalized Counts (RPKM)")
```




```
color = ifelse(phenoDataFrame$GROUP_bin == "Moderate", "black", "brown3")

plotRLE(counts.log, outline = FALSE, ylim = c(-0.3, 0.3), col = color, main = "Log-transformed Counts (
```




```
# Test normality of RPKM values
shap.RPKM <- shapiro.test(counts.ok[1, ])

# Test normality of log values
shap.log <- shapiro.test(counts.log[1, ])

# Check results
shap.log
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts.log[1, ]
## W = 0.91542, p-value = 8.544e-07
```

```
shap.RPKM
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts.ok[1, ]
## W = 0.43034, p-value < 2.2e-16
```

Multidimensional analysis of gene expression with PCA

```
# Compute the variance of each gene across samples
V <- apply(counts.ok, 1, var)
# sort the results by variance in decreasing order and select the top 500 genes
selectedGenes <- names(V[order(V, decreasing = TRUE)][1:500])

# Perform PCA pcaResults1 <- prcomp(t(counts.log[selectedGenes,]),scale = TRUE)
pcaResults2 <- prcomp(t(counts.ok[selectedGenes, ]), scale = TRUE)
```

See explained varibility for each principal component:

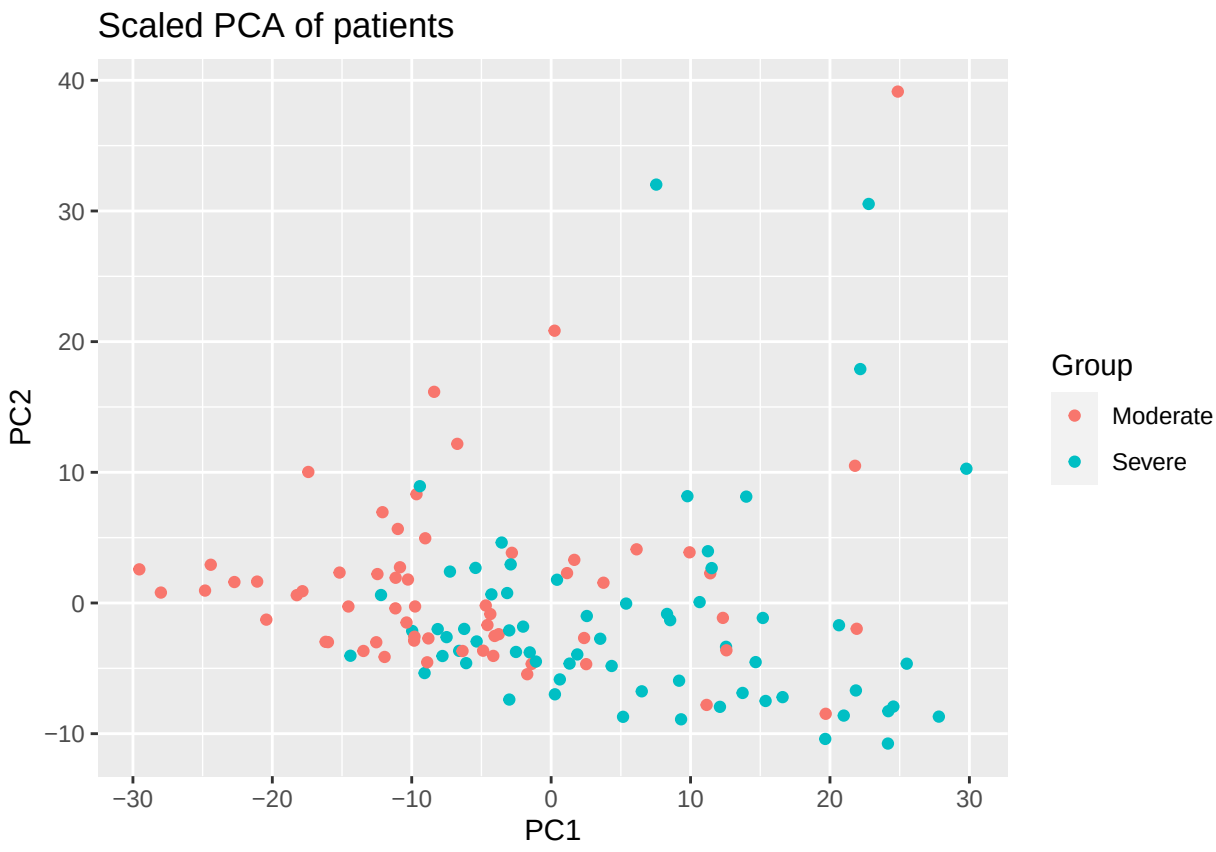
```
summary(pcaResults2)
```

```
## Importance of components:
##                            PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     13.2689 7.7209 6.41793 5.21684 5.11165 4.44868 4.01208
## Proportion of Variance  0.3521 0.1192 0.08238 0.05443 0.05226 0.03958 0.03219
## Cumulative Proportion   0.3521 0.4713 0.55373 0.60816 0.66042 0.70000 0.73219
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     3.5778 3.34345 2.83928 2.58313 2.36103 2.29845 2.12906
## Proportion of Variance 0.0256 0.02236 0.01612 0.01335 0.01115 0.01057 0.00907
## Cumulative Proportion  0.7578 0.78015 0.79628 0.80962 0.82077 0.83134 0.84040
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     2.07709 1.93281 1.89210 1.81798 1.75193 1.67442 1.57469
## Proportion of Variance 0.00863 0.00747 0.00716 0.00661 0.00614 0.00561 0.00496
## Cumulative Proportion  0.84903 0.85650 0.86366 0.87027 0.87641 0.88202 0.88698
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     1.53104 1.51776 1.46271 1.45115 1.38999 1.33368 1.27238
## Proportion of Variance 0.00469 0.00461 0.00428 0.00421 0.00386 0.00356 0.00324
## Cumulative Proportion  0.89167 0.89627 0.90055 0.90476 0.90863 0.91219 0.91542
##                           PC29    PC30    PC31    PC32    PC33    PC34    PC35
## Standard deviation     1.26204 1.23830 1.21595 1.15135 1.13802 1.10333 1.08376
## Proportion of Variance 0.00319 0.00307 0.00296 0.00265 0.00259 0.00243 0.00235
## Cumulative Proportion  0.91861 0.92168 0.92463 0.92728 0.92987 0.93231 0.93466
##                           PC36    PC37    PC38    PC39    PC40   PC41    PC42
## Standard deviation     1.05486 1.02759 1.01698 1.00624 0.97667 0.9734 0.95690
## Proportion of Variance 0.00223 0.00211 0.00207 0.00203 0.00191 0.0019 0.00183
## Cumulative Proportion  0.93688 0.93900 0.94106 0.94309 0.94500 0.9469 0.94872
##                           PC43    PC44    PC45    PC46    PC47    PC48   PC49
## Standard deviation     0.93458 0.90512 0.88959 0.87399 0.85859 0.84339 0.8374
## Proportion of Variance 0.00175 0.00164 0.00158 0.00153 0.00147 0.00142 0.0014
## Cumulative Proportion  0.95047 0.95211 0.95369 0.95522 0.95669 0.95812 0.9595
##                           PC50    PC51   PC52    PC53    PC54    PC55    PC56
## Standard deviation     0.81688 0.78757 0.7742 0.76268 0.75407 0.74401 0.73939
## Proportion of Variance 0.00133 0.00124 0.0012 0.00116 0.00114 0.00111 0.00109
## Cumulative Proportion  0.96085 0.96209 0.9633 0.96446 0.96559 0.96670 0.96779
##                           PC57    PC58    PC59    PC60    PC61    PC62   PC63
## Standard deviation     0.72997 0.72730 0.69877 0.68923 0.68449 0.67291 0.6699
## Proportion of Variance 0.00107 0.00106 0.00098 0.00095 0.00094 0.00091 0.0009
## Cumulative Proportion  0.96886 0.96992 0.97089 0.97184 0.97278 0.97369 0.9746
##                           PC64    PC65    PC66   PC67    PC68    PC69    PC70
## Standard deviation     0.66471 0.65181 0.64435 0.6312 0.62725 0.61062 0.60806
## Proportion of Variance 0.00088 0.00085 0.00083 0.0008 0.00079 0.00075 0.00074
## Cumulative Proportion  0.97547 0.97632 0.97715 0.9779 0.97873 0.97948 0.98022
##                           PC71    PC72    PC73    PC74    PC75    PC76   PC77
## Standard deviation     0.59712 0.59669 0.58073 0.57675 0.56896 0.56597 0.5461
## Proportion of Variance 0.00071 0.00071 0.00067 0.00067 0.00065 0.00064 0.0006
## Cumulative Proportion  0.98093 0.98164 0.98232 0.98298 0.98363 0.98427 0.9849
##                           PC78    PC79    PC80    PC81    PC82    PC83   PC84
## Standard deviation     0.54056 0.53693 0.52945 0.52407 0.51234 0.50355 0.4999
## Proportion of Variance 0.00058 0.00058 0.00056 0.00055 0.00052 0.00051 0.0005
## Cumulative Proportion  0.98545 0.98603 0.98659 0.98714 0.98766 0.98817 0.9887
##                           PC85    PC86    PC87    PC88    PC89    PC90    PC91
## Standard deviation     0.49482 0.48763 0.48240 0.47698 0.47171 0.46609 0.45089
```

```
## Proportion of Variance 0.00049 0.00048 0.00047 0.00046 0.00045 0.00043 0.00041
## Cumulative Proportion  0.98916 0.98963 0.99010 0.99055 0.99100 0.99143 0.99184
##                           PC92    PC93    PC94    PC95    PC96    PC97    PC98
## Standard deviation     0.4485 0.44368 0.43793 0.43042 0.42429 0.41939 0.41762
## Proportion of Variance 0.0004 0.00039 0.00038 0.00037 0.00036 0.00035 0.00035
## Cumulative Proportion  0.9922 0.99264 0.99302 0.99339 0.99375 0.99410 0.99445
##                           PC99   PC100  PC101   PC102   PC103   PC104   PC105
## Standard deviation     0.41283 0.39528 0.3854 0.38106 0.37642 0.37021 0.36689
## Proportion of Variance 0.00034 0.00031 0.0003 0.00029 0.00028 0.00027 0.00027
## Cumulative Proportion  0.99479 0.99510 0.9954 0.99569 0.99597 0.99625 0.99652
##                           PC106   PC107   PC108   PC109   PC110   PC111  PC112
## Standard deviation     0.36505 0.35221 0.33876 0.33509 0.33209 0.32492 0.3184
## Proportion of Variance 0.00027 0.00025 0.00023 0.00022 0.00022 0.00021 0.0002
## Cumulative Proportion  0.99678 0.99703 0.99726 0.99749 0.99771 0.99792 0.9981
##                          PC113   PC114   PC115   PC116   PC117   PC118   PC119
## Standard deviation     0.3143 0.30974 0.30629 0.29301 0.28724 0.28024 0.27664
## Proportion of Variance 0.0002 0.00019 0.00019 0.00017 0.00017 0.00016 0.00015
## Cumulative Proportion  0.9983 0.99851 0.99870 0.99887 0.99904 0.99919 0.99935
##                           PC120   PC121   PC122   PC123   PC124     PC125
## Standard deviation     0.26388 0.26200 0.25894 0.25411 0.23958 5.893e-15
## Proportion of Variance 0.00014 0.00014 0.00013 0.00013 0.00011 0.000e+00
## Cumulative Proportion  0.99948 0.99962 0.99976 0.99989 1.00000 1.000e+00
```

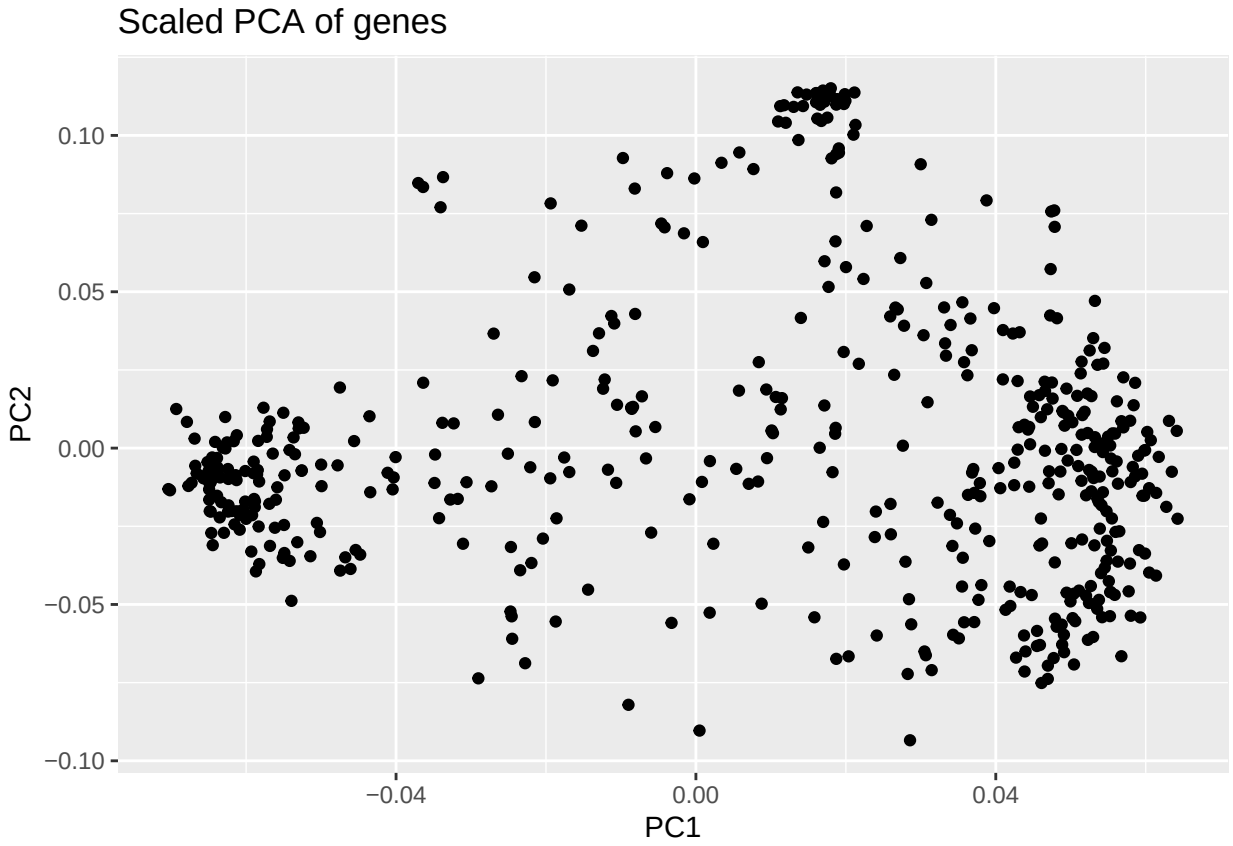
PCA for patients:

```
qplot(pcaResults2$x[, 1], pcaResults2$x[, 2], xlab = "PC1", ylab = "PC2", main = "Scaled PCA of patients
    colour = phenoDataFrame$GROUP_bin) + labs(colour = "Group")
```

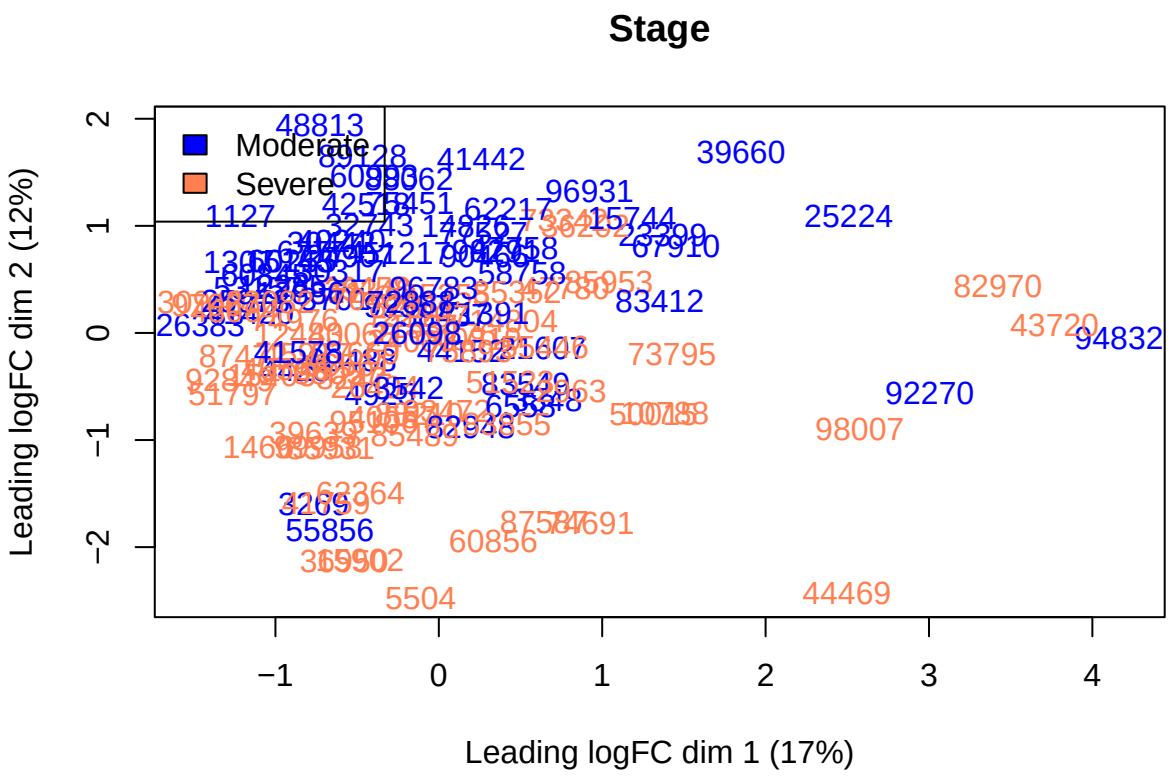


PCA for genes:

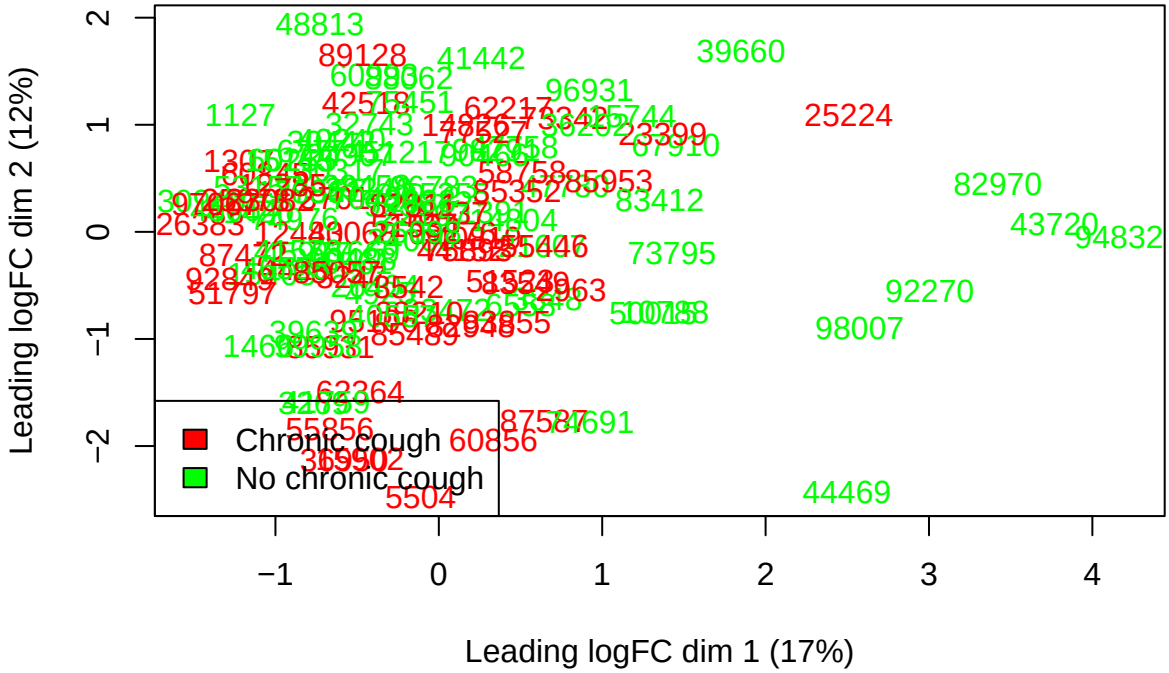
```
qplot(pcaResults2$rotation[, 1], pcaResults2$rotation[, 2], xlab = "PC1", ylab = "PC2",
    main = "Scaled PCA of genes")
```



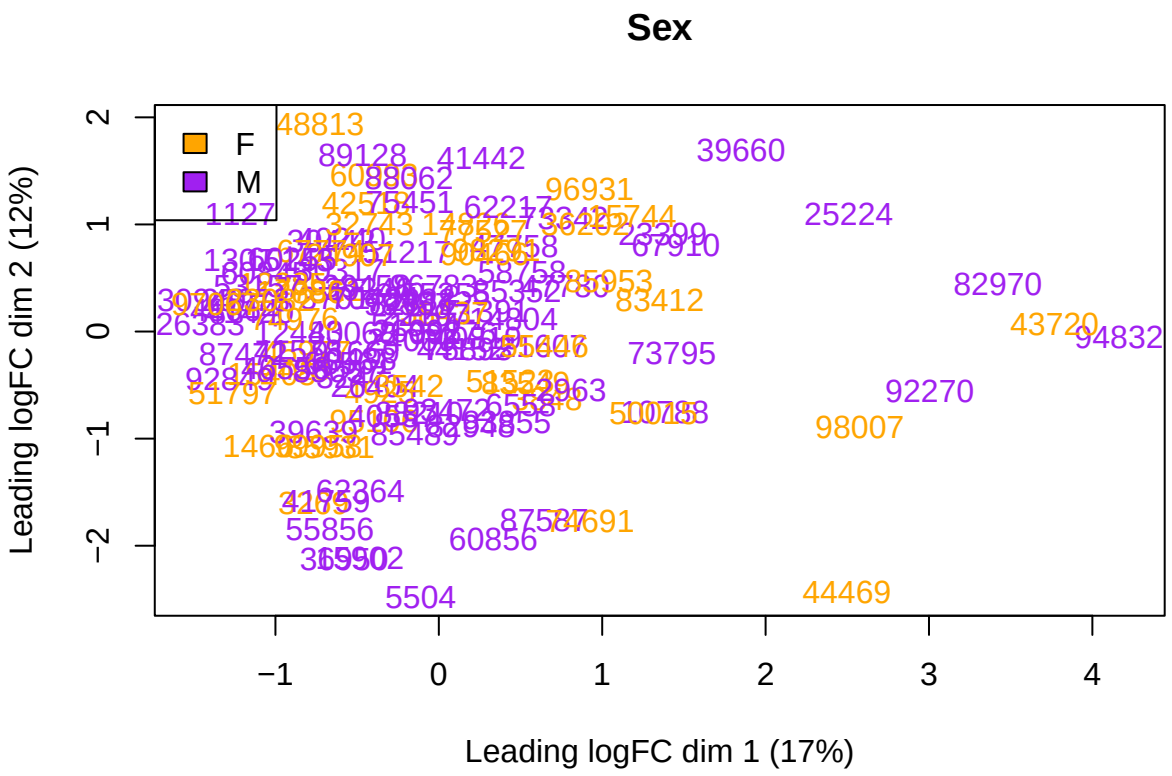

```
color.gr <- ifelse(phenoDataFrame$GROUP_bin == "Moderate", "blue", "coral")
color.cough <- ifelse(phenoDataFrame$COUGH == "Chronic cough", "red", "green")
color.sex <- ifelse(phenoDataFrame$SEX == "F", "orange", "purple")

plotMDS(y, col = color.gr)
legend("topleft", fill = c("blue", "coral"), legend = levels(phenoDataFrame$GROUP_bin))
title("Stage")
```




```
plotMDS(y, col = color.cough)
legend("bottomleft", fill = c("red", "green"), legend = levels(as.factor(phenoDataFrame$COUGH)))
title("Cough")
```



```
plotMDS(y, col = color.sex)
legend("topleft", fill = c("orange", "purple"), legend = levels(as.factor(phenoDataFrame$SEX)))
title("Sex")
```



As can be seen in the previous analysis, in this case patients are slightly stratified by COPD stage. Nevertheless, `Moderate` and `Severe` seem to be close stages since patients are almost mixed.

Nevertheless, this does not mean that counts are not properly normalized. The **mean-difference plots** show average expression (mean: x-axis) against log-fold-changes (difference: y-axis). Imagine that counts have been normalized just **by library size, but not for composition bias**. Which is translated to the fact that TMM normalization factors have been not computed. Counts would be displayed asymmetrically over or undo the horizontal line (one side bigger than the another).

Because our `DGEList` object contains the normalisation factors (if computed with TMM), if we redo these plots using `y` object, we should see the composition bias problem has been solved. See in the following example as data is similarly distributed across the horizontal line:

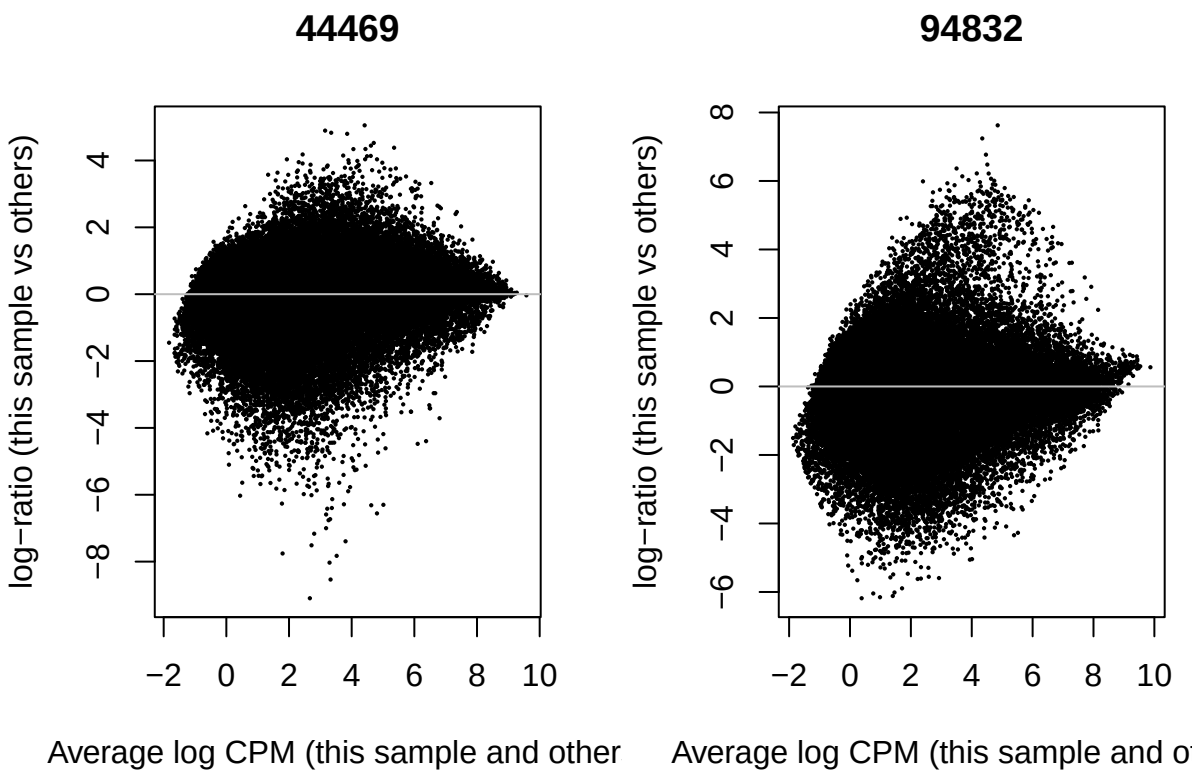
```
# Look column index for patients of interest
grep("44469", colnames(counts.ok))
```

```
## [1] 105
```

```
grep("94832", colnames(counts.ok))
```

```
## [1] 80
```

```
par(mfrow = c(1, 2))
plotMD(y, column = 105)
abline(h = 0, col = "grey")
plotMD(y, column = 80)
abline(h = 0, col = "grey")
```



In this case, if we use `log-transformed` counts, scale would be smaller and data will be seen more compacted:

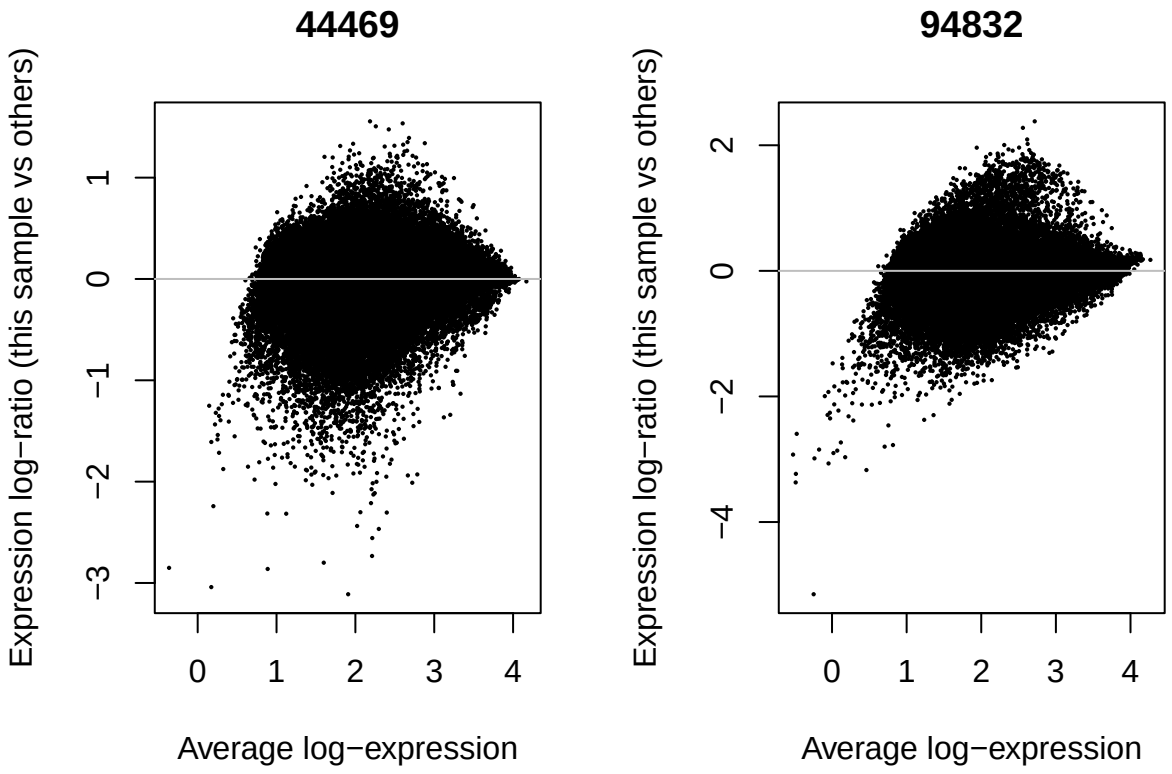
```
# Look column index for patients of interest
grep("44469", colnames(counts.log))
```

```
## [1] 105
```

```
grep("94832", colnames(counts.log))
```

```
## [1] 80
```

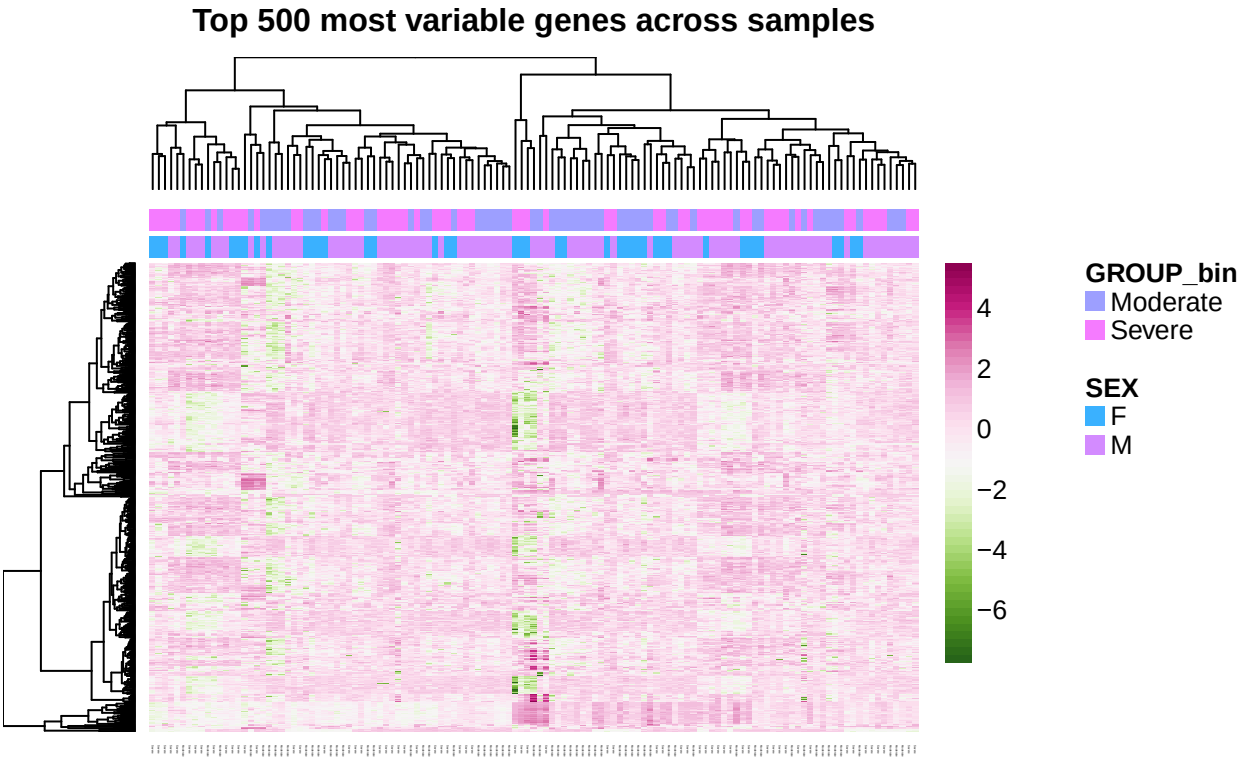
```
par(mfrow = c(1, 2))
plotMD(counts.log, column = 105)
abline(h = 0, col = "grey")
plotMD(counts.log, column = 80)
abline(h = 0, col = "grey")
```



Multidimensional analysis Hierarchical Clustering

```
# You can use those function either to watch or get info of the palette
# display.brewer.all() brewer.pal.info

# Select a palette
mypalette <- brewer.pal(11, "PiYG")
# Extend colors
morecols <- colorRampPalette(mypalette)

# Select more variable genes (previously computed for PCA analysis)
highly_variable_lcpm <- counts.log[selectedGenes, ]
```

```
# pdf('heatmap_sputum.pdf')

# Plot heatmap
aheatmap(highly_variable_lcpm, col = rev(morecols(50)), main = "Top 500 most variable genes across samp
    annCol = phenoDataFrame[, c("GROUP_bin", "SEX")], labCol = phenoDataFrame[, "GROUP_bin"],
    scale = "row")
```




```
# dev.off()
```

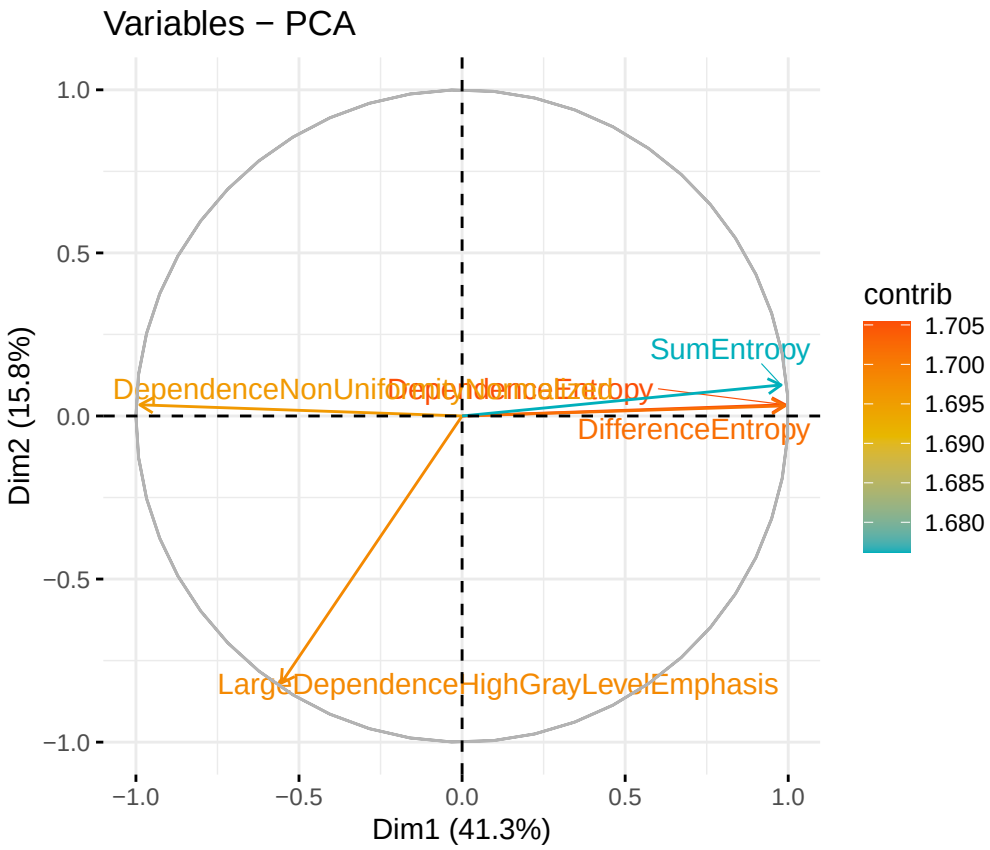
PCA for radiomic features of 125 patients

Compute pca for radiomic features object with 125 patients:

```
# Prepare data
rdr_pca_filt <- rdr_assay

# Change labels to visualize better
rownames(rdr_pca_filt) <- substr(rownames(rdr_pca_filt), start = 1, stop = nchar(rownames(rdr_pca_filt))
    9)

# Store 15 first dimensions
res.pca <- PCA(t(rdr_pca_filt), ncp = 15, graph = FALSE)
```

Top variables contribution:

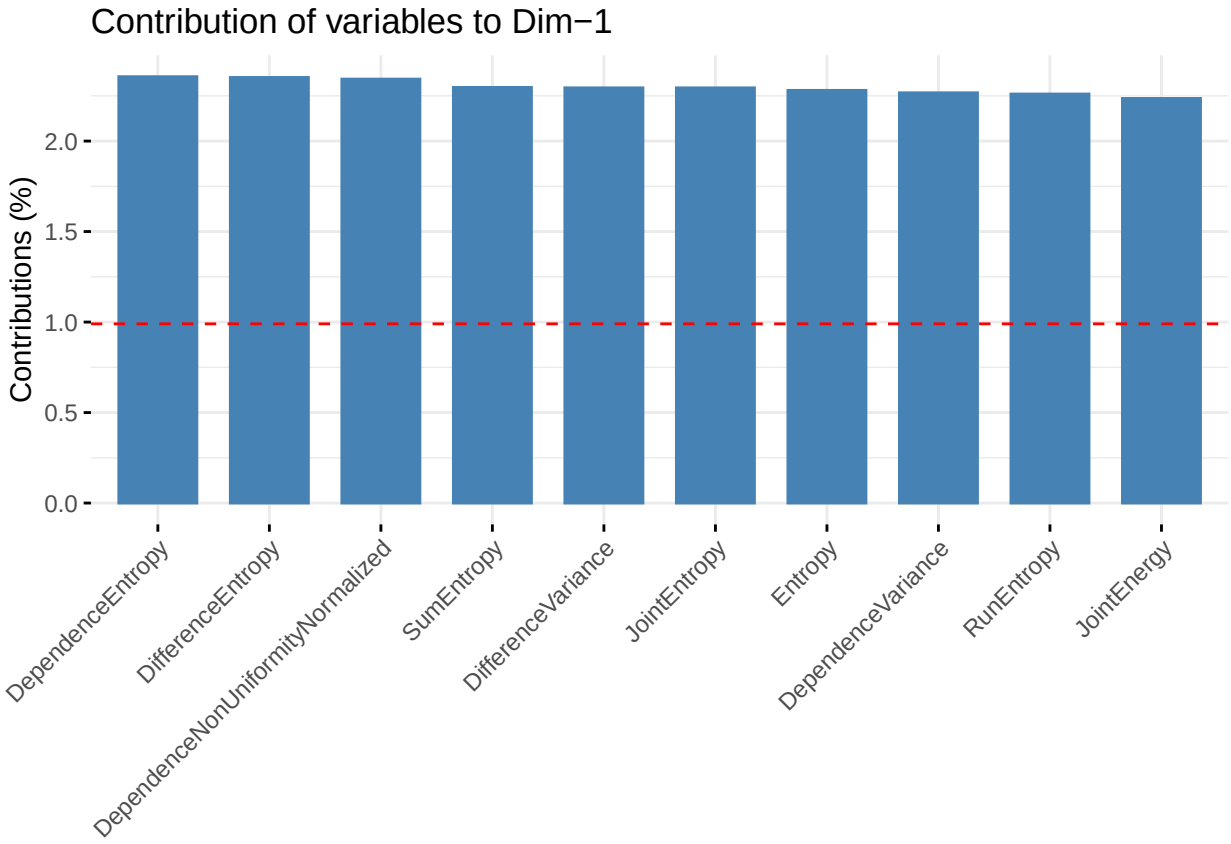
```
# pdf('Contribution variables PCA patients.pdf')

# Graph of variables: default plot
fviz_pca_var(res.pca, col.var = "contrib", gradient.cols = c("#00AFBB", "#E7B800",
    "#FC4E07"), select.var = list(contrib = 5), repel = TRUE)  # Avoid text overlapping
```



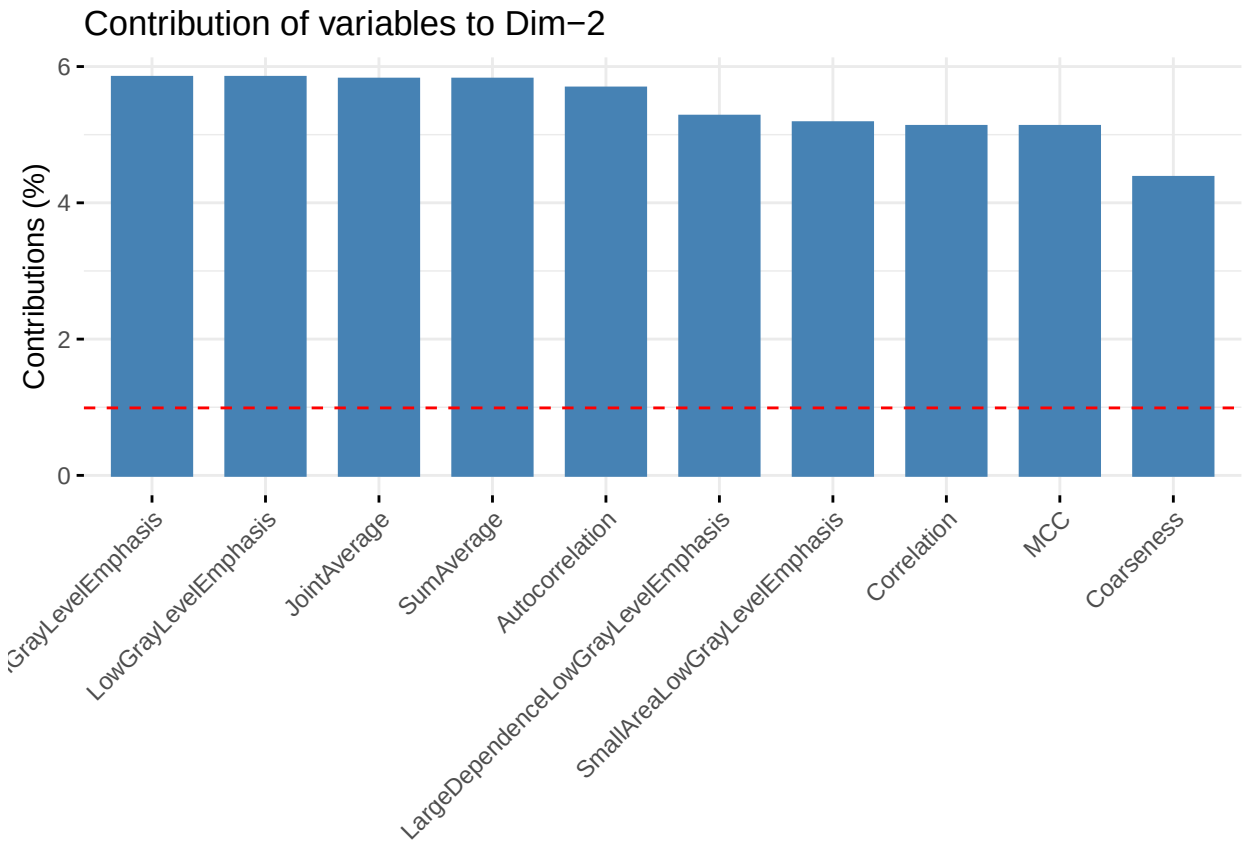

```
# dev.off()
```

See which features are contributing the most per dimension:

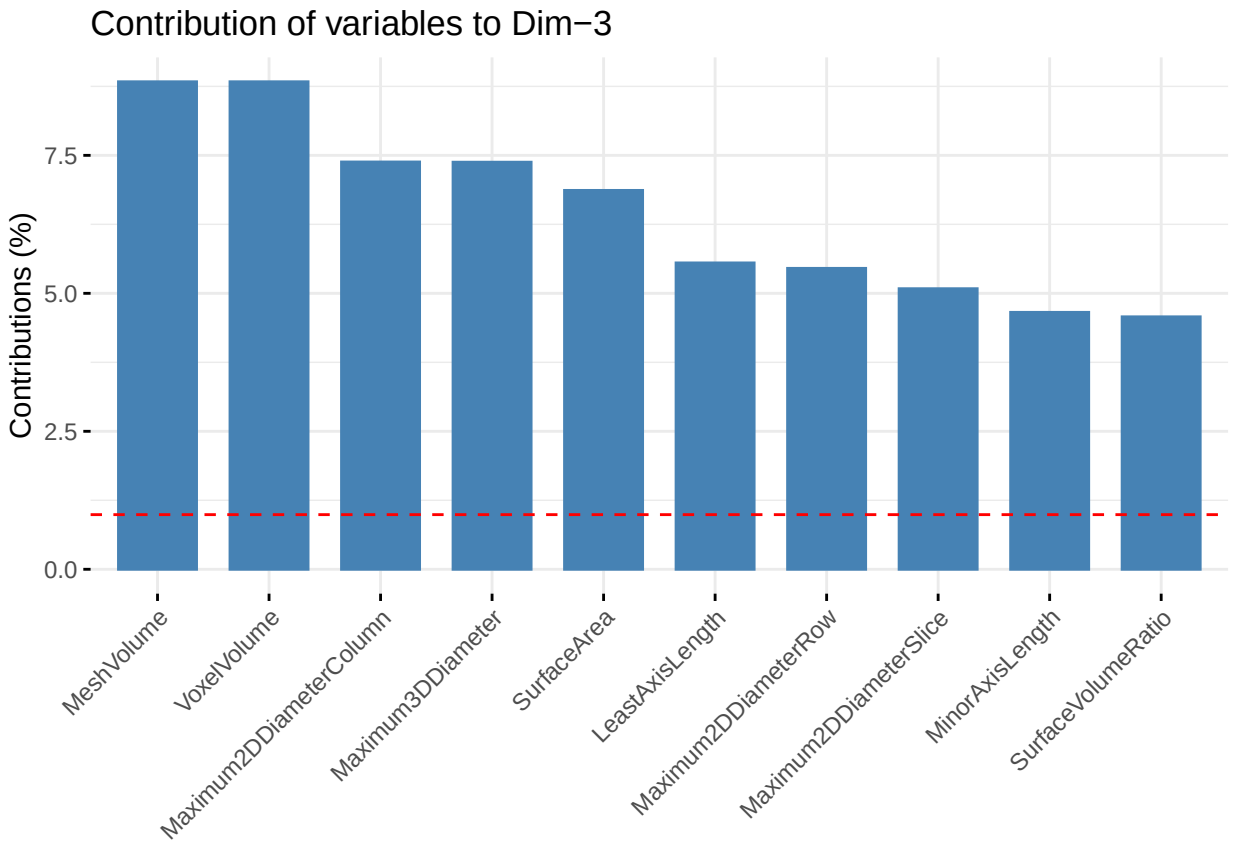
```
# pdf('Contribution variables PC1-3.pdf')

# Contributions of variables to PC1
fviz_contrib(res.pca, choice = "var", axes = 1, top = 10)
```




```
# Contributions of variables to PC2
fviz_contrib(res.pca, choice = "var", axes = 2, top = 10)
```



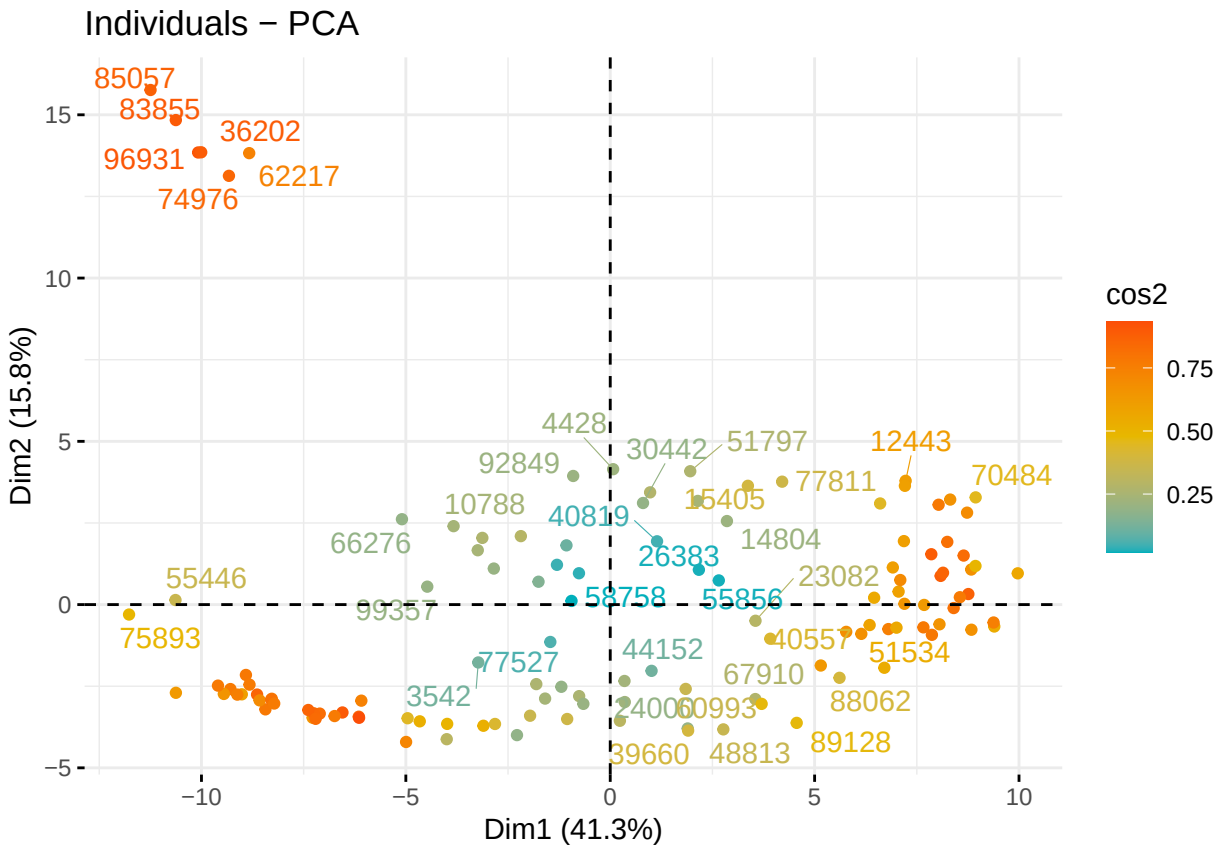

```
# Contributions of variables to PC3
fviz_contrib(res.pca, choice = "var", axes = 3, top = 10)
```




```
# dev.off()
```

See individuals that are contributing the most to the analysis:

```
# pdf('Contribution individuals PCA.pdf')

# Graph of individuals 1. Use repel = TRUE to avoid overplotting 2. Control
# automatically the color of individuals using the cos2 cos2 = the quality of
# the individuals on the factor map Use points only 3. Use gradient color
fviz_pca_ind(res.pca, col.ind = "cos2", gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
    repel = TRUE  # Avoid text overlapping (slow if many points)
)
```




```
# dev.off()
```

Selection of variables

Prepare a data frame with possible covariates and PCs:

```
# residual_volume = phenoDataFrame$RV, SVC = phenoDataFrame$SVC, TLC =
# phenoDataFrame$TLC, FRC = phenoDataFrame$FRC, Low_percentile =
# phenoDataFrame$LOW15PCT, PC2_rdr = rdr_colData$PC2, PC3_rdr =
# rdr_colData$PC3, PC4_rdr = rdr_colData$PC4, PC5_rdr = rdr_colData$PC5,
# PC6_rdr = rdr_colData$PC6, PC7_rdr = rdr_colData$PC7, PC8_rdr =
# rdr_colData$PC8, PC9_rdr = rdr_colData$PC9, PC10_rdr = rdr_colData$PC10,


# Store values for for individuals from PCA analysis of radiomic features
pca_ind <- get_pca_ind(res.pca)
pca_values <- pca_ind$coord

# Create a data frame with clinical variables to asses and PCs
variables_interest <- data.frame(Sex = as.factor(phenoDataFrame$SEX), Age = phenoDataFrame$AGE,
    Dwalk = phenoDataFrame$DWALK, BMI = phenoDataFrame$BMI, Smoke_history = phenoDataFrame$SUSMHS,
    Years_smoke = phenoDataFrame$SUSMYR, Cough = as.factor(phenoDataFrame$COUGH),
    ex_first_year = phenoDataFrame$Y1EXBS, Group = as.factor(phenoDataFrame$GROUP_bin),
    N_cigarrete_day = phenoDataFrame$SUCGSMDY, Cr_bronchitis = as.factor(phenoDataFrame$CBRONCH),
    Cr_wheezeng = as.factor(phenoDataFrame$ATS3EG), history_asthma = as.factor(phenoDataFrame$ATS5G),
    fume_expose = as.factor(phenoDataFrame$ATS6C), dusty_expose = as.factor(phenoDataFrame$ATS6B),
    phlegm = as.factor(phenoDataFrame$PHLEGM), heart_failure = as.factor(phenoDataFrame$ATS8F),
```

```
    stroke = as.factor(phenoDataFrame$ATS8E), diabetes = as.factor(phenoDataFrame$ATS8L),
    osteoporosis = as.factor(phenoDataFrame$ATS8H), FEVVCPD = phenoDataFrame$FEVVCPD,
    FEV1PSPC = phenoDataFrame$FEV1PSPC, TLC = phenoDataFrame$TLC, FRC = phenoDataFrame$FRC,
    Low_percentile = phenoDataFrame$LOW15PCT, PC1_rdr = pca_values[, 1], PC2_rdr = pca_values[,
        2], PC3_rdr = pca_values[, 3], PC4_rdr = pca_values[, 4], PC5_rdr = pca_values[,
        5], PC6_rdr = pca_values[, 6], PC7_rdr = pca_values[, 7], PC8_rdr = pca_values[,
        8], PC9_rdr = pca_values[, 9], PC10_rdr = pca_values[, 10], PC11_rdr = pca_values[,
        11], PC12_rdr = pca_values[, 12], PC13_rdr = pca_values[, 13], PC14_rdr = pca_values[,
        14], PC15_rdr = pca_values[, 15], row.names = rownames(phenoDataFrame))

# Code for removing columns with some NA
variables_interest <- variables_interest[, apply(variables_interest, 2, function(x) !any(is.na(x)))]

# count unique values for each variable
sapply(lapply(variables_interest, unique), length)
```

```
##             Sex             Age           Dwalk             BMI   Smoke_history
##               2              25              98             124               1
##     Years_smoke           Cough           Group N_cigarrete_day   Cr_bronchitis
##              37               2               2              18               2
##     Cr_wheezeng  history_asthma     fume_expose    dusty_expose          phlegm
##               2               2               2               2               2
##   heart_failure          stroke        diabetes    osteoporosis         FEVVCPD
##               2               2               2               2              47
##        FEV1PSPC         PC1_rdr         PC2_rdr         PC3_rdr         PC4_rdr
##             115             125             125             125             125
##         PC5_rdr         PC6_rdr         PC7_rdr         PC8_rdr         PC9_rdr
##             125             125             125             125             125
##        PC10_rdr        PC11_rdr        PC12_rdr        PC13_rdr        PC14_rdr
##             125             125             125             125             125
##        PC15_rdr
##             125
```

```
# display unique values for each variable
lapply(variables_interest[c("Smoke_history", "Sex", "Cough")], unique)
```

```
## $Smoke_history
## [1] "Former smoker"
##
## $Sex
## [1] M F
## Levels: F M
##
## $Cough
## [1] No chronic cough Chronic cough
## Levels: Chronic cough No chronic cough
```

```
# Remove variables with just one factor
variables_interest[, "Smoke_history"] <- NULL

# Select counts higher than 10
countData <- counts.ok[rowSums(counts.ok) > 10, ]
```

Perform automatic selection by using **forward stepwise regression**. Each gene counts represents a model, therefore is necessary to check which variables are present in models with the best adjustment (highest

R-squares):

```
# Perform automatic stepwise regression: forward
mod.fow <- stats::step(lm(t(countData) ~ ., data = variables_interest), trace = FALSE,
    direction = "forward")

# Store coefficients of the model in a long list (each gene is a model since
# we're looking counts)
models.results <- summary(mod.fow)

# Check and store genes with the highest R-squared values (better adjust)
models.r.sq <- sapply(models.results, function(x) {
    x$r.squared > 0.6
})
table(models.r.sq)
```

```
## models.r.sq
## FALSE  TRUE
## 48492    33
```

```
genes.mfit.high <- models.results[models.r.sq]

# Retrieve significant p.values (< 0.05) for each variable from summary of lm
# to most adjusted genes and check which variables are significant
models.p.values <- sapply(genes.mfit.high, function(x) {
    x$coefficients[, 4] < 0.05
})
head(models.p.values)
```

```
##             Response 1557124_at Response 200789_at Response 201909_at
## (Intercept)               FALSE               TRUE              FALSE
## SexM                      FALSE              FALSE               TRUE
## Age                       FALSE               TRUE              FALSE
## Dwalk                     FALSE              FALSE              FALSE
## BMI                       FALSE              FALSE              FALSE
## Years_smoke               FALSE              FALSE               TRUE
##             Response 203179_at Response 203576_at Response 203980_at
## (Intercept)               TRUE               TRUE              FALSE
## SexM                     FALSE              FALSE              FALSE
## Age                       TRUE               TRUE              FALSE
## Dwalk                    FALSE              FALSE              FALSE
## BMI                       TRUE              FALSE              FALSE
## Years_smoke              FALSE              FALSE              FALSE
##             Response 204409_s_at Response 204410_at Response 205000_at
## (Intercept)                FALSE               FALSE              FALSE
## SexM                        TRUE                TRUE               TRUE
## Age                        FALSE               FALSE              FALSE
## Dwalk                       TRUE                TRUE              FALSE
## BMI                        FALSE               FALSE              FALSE
## Years_smoke                FALSE               FALSE              FALSE
##             Response 205001_s_at Response 206279_at Response 208690_s_at
## (Intercept)                FALSE               FALSE                TRUE
## SexM                        TRUE                TRUE               FALSE
## Age                        FALSE               FALSE                TRUE
## Dwalk                      FALSE               FALSE               FALSE
```

```
## BMI                         FALSE              FALSE                FALSE
## Years_smoke                 FALSE              FALSE                FALSE
##             Response 210322_x_at Response 211149_at Response 214131_at
## (Intercept)                FALSE              FALSE               FALSE
## SexM                        TRUE               TRUE                TRUE
## Age                        FALSE              FALSE                TRUE
## Dwalk                      FALSE              FALSE               FALSE
## BMI                         TRUE              FALSE               FALSE
## Years_smoke                 TRUE              FALSE               FALSE
##             Response 214983_at Response 215909_x_at Response 217892_s_at
## (Intercept)              FALSE                TRUE                 TRUE
## SexM                      TRUE               FALSE                FALSE
## Age                      FALSE                TRUE                 TRUE
## Dwalk                    FALSE                TRUE                FALSE
## BMI                      FALSE               FALSE                FALSE
## Years_smoke              FALSE               FALSE                FALSE
##             Response 221728_x_at Response 223645_s_at Response 223646_s_at
## (Intercept)                TRUE                FALSE                FALSE
## SexM                       TRUE                 TRUE                 TRUE
## Age                       FALSE                 TRUE                FALSE
## Dwalk                     FALSE                FALSE                FALSE
## BMI                        TRUE                FALSE                FALSE
## Years_smoke               FALSE                FALSE                FALSE
##             Response 224588_at Response 227024_s_at Response 227671_at
## (Intercept)               TRUE                TRUE               FALSE
## SexM                      TRUE               FALSE                TRUE
## Age                      FALSE                TRUE               FALSE
## Dwalk                    FALSE               FALSE               FALSE
## BMI                      FALSE               FALSE               FALSE
## Years_smoke              FALSE               FALSE               FALSE
##             Response 228492_at Response 230760_at Response 231592_at
## (Intercept)              FALSE              FALSE               TRUE
## SexM                      TRUE               TRUE               TRUE
## Age                      FALSE              FALSE              FALSE
## Dwalk                    FALSE              FALSE              FALSE
## BMI                      FALSE              FALSE              FALSE
## Years_smoke              FALSE              FALSE              FALSE
##             Response 232618_at Response 234992_x_at Response 235446_at
## (Intercept)              FALSE                TRUE               FALSE
## SexM                      TRUE                TRUE                TRUE
## Age                       TRUE               FALSE               FALSE
## Dwalk                    FALSE               FALSE               FALSE
## BMI                      FALSE               FALSE               FALSE
## Years_smoke              FALSE               FALSE               FALSE
##             Response 235978_at Response 236694_at Response 238035_at
## (Intercept)              FALSE              FALSE               TRUE
## SexM                     FALSE               TRUE              FALSE
## Age                       TRUE              FALSE              FALSE
## Dwalk                    FALSE              FALSE              FALSE
## BMI                      FALSE              FALSE              FALSE
## Years_smoke              FALSE              FALSE               TRUE
```

```
# Check how many variables have at least one significant p.value
variables.sig <- apply(models.p.values, 1, function(x) {
```

```
    any(x == TRUE)
})
table(variables.sig)
```

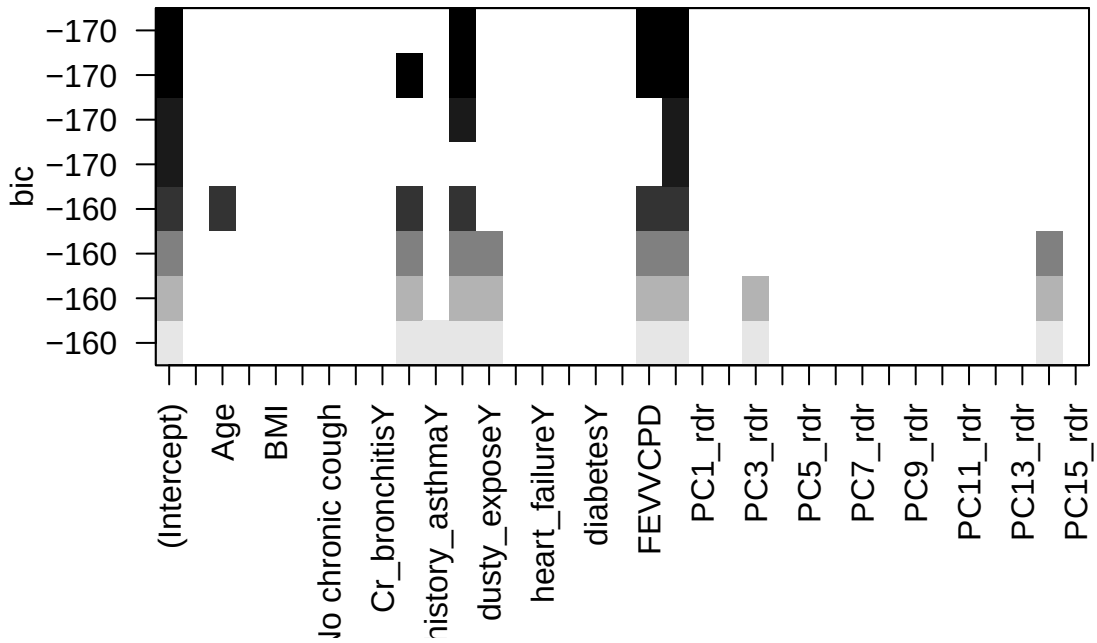
```
## variables.sig
## FALSE  TRUE
##     2    34
```

```
not.sig.variables <- rownames(models.p.values[!variables.sig, ])
not.sig.variables
```

```
## [1] "PC6_rdr"  "PC11_rdr"
```

See variables that better explain differences between stages automatically (optional):

```
plot(regsubsets(Group ~ ., data = variables_interest, method = "exhaustive", nbest = 1))
```



Functions for assessing linear models:

```
# RSS se ajusta mejor si es más bajo
rss <- function(fitted, actual) {
    sum((fitted - actual)^2)
}

# RMSE se ajusta mejor si es más bajo
rmse <- function(fitted, actual) {
    sqrt(mean((fitted - actual)^2))
}
```

Let's start exploring manually variables:

```
mod.s <- lm(t(countData) ~ Sex, data = variables_interest)
mod.sa <- lm(t(countData) ~ Sex + Age, data = variables_interest)
mod.a <- lm(t(countData) ~ Age, data = variables_interest)
mod.g <- lm(t(countData) ~ Group, data = variables_interest)
mod.sag <- lm(t(countData) ~ Group + Sex + Age, data = variables_interest)
mod.dwalk <- lm(t(countData) ~ Dwalk, data = variables_interest)
```

```
mod.dsag <- lm(t(countData) ~ Group + Sex + Age + Dwalk, data = variables_interest)
mod.dag <- lm(t(countData) ~ Group + Age + Dwalk, data = variables_interest)
mod.dsagf <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC, data = variables_interest)
mod.dsagffd <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose, data = variables_interest)
mod.dsagffda <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma, data = variables_interest)
mod.pc1n2 <- lm(t(countData) ~ PC1_rdr + PC2_rdr, data = variables_interest)
mod.dsagffdapc1n2 <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma + PC1_rdr + PC2_rdr, data = variables_interest)
# Without Cough
mod.dsagffdapc1n2 <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma + PC1_rdr + PC2_rdr, data = variables_interest)
# Without Dwalk and FEV1PSPC but Cough
mod.sagfdacpc1n2 <- lm(t(countData) ~ Group + Sex + Age + fume_expose + dusty_expose +
    history_asthma + Cough + PC1_rdr + PC2_rdr, data = variables_interest)
# ALL + 1 and 2 PC
mod.dsagffdacpc1n2 <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma + Cough + PC1_rdr + PC2_rdr, data = variables_interest)
# All + PCs
mod.dsagffdacpc1.8 <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma + Cough + PC1_rdr + PC2_rdr + PC3_rdr + PC4_rdr +
    PC5_rdr + PC6_rdr + PC7_rdr + PC8_rdr, data = variables_interest)
# All + PCs + BMI + Cr_wheezeng
mod.dsagffdacBWpc1.8 <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC +
    fume_expose + dusty_expose + history_asthma + Cough + BMI + Cr_wheezeng + PC1_rdr +
    PC2_rdr + PC3_rdr + PC4_rdr + PC5_rdr + PC6_rdr + PC7_rdr + PC8_rdr, data = variables_interest)

rmse(fitted(mod.s), t(countData))
```

```
## [1] 254.2539
```

```
rmse(fitted(mod.sa), t(countData))
```

```
## [1] 252.6789
```

```
rmse(fitted(mod.a), t(countData))
```

```
## [1] 254.0286
```

```
rmse(fitted(mod.g), t(countData))
```

```
## [1] 244.71
```

```
rmse(fitted(mod.sag), t(countData))
```

```
## [1] 240.8153
```

```
rmse(fitted(mod.dwalk), t(countData))
```

```
## [1] 252.8118
```

```
rmse(fitted(mod.dsag), t(countData))
```

```
## [1] 239.7644
```

```
rmse(fitted(mod.dag), t(countData))
```

```
## [1] 241.4093
```

```
rmse(fitted(mod.dsagf), t(countData))
```

```
## [1] 238.4131
```

```
rmse(fitted(mod.dsagffd), t(countData))
```

```
## [1] 235.1352
```

```
rmse(fitted(mod.dsagffda), t(countData))
```

```
## [1] 233.4724
```

```
rmse(fitted(mod.pc1n2), t(countData))
```

```
## [1] 253.0372
```

```
rmse(fitted(mod.dsagffdapc1n2), t(countData))
```

```
## [1] 230.938
```

```
rmse(fitted(mod.sagfdacpc1n2), t(countData))
```

```
## [1] 231.9872
```

```
rmse(fitted(mod.dsagffdacpc1n2), t(countData))
```

```
## [1] 229.7521
```

```
rmse(fitted(mod.dsagffdacpc1.8), t(countData))
```

```
## [1] 221.9395
```

```
# Best model
rmse(fitted(mod.dsagffdacBWpc1.8), t(countData))
```

```
## [1] 220.3828
```

Surrogate variable analysis

In this case, variables that want to be tested lately have to be protected from `sva` analysis in order to preserve the variability related with this known covariates. The null-model `mod0` can have all the desired variables of the model, but the one to **contrast** lately:

```
# Create a model with tumor stage as covariate and a null model
mod1 <- model.matrix(~Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose + dusty_expose +
    history_asthma + Cough + BMI + Cr_wheezeng + PC1_rdr + PC2_rdr + PC3_rdr + PC4_rdr +
    PC5_rdr + PC6_rdr + PC7_rdr + PC8_rdr + PC9_rdr + PC10_rdr + PC11_rdr + PC12_rdr +
    PC13_rdr + PC14_rdr + PC15_rdr, data = variables_interest)
mod0 <- model.matrix(~Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose + dusty_expose +
    history_asthma + Cough + BMI + Cr_wheezeng, data = variables_interest)

# This estimates the required number of surrogate variables countData <-
# counts.ok[rowSums(counts.ok) > 10, ]
res <- svaseq(countData, mod1, mod0)
```

```
## Number of significant surrogate variables is:  17
## Iteration (out of 5 ):1  2  3  4  5
```

```
# Number of estimated surrogate variables
res$n.sv
```

```
## [1] 17
```

```
# Check surrogated variables
head(res$sv)
```

```
##              [,1]        [,2]        [,3]         [,4]         [,5]        [,6]
## [1,]  0.080174193 -0.01962393  0.04433164 -0.005233658  0.026427018  0.03407429
## [2,] -0.074225555 -0.09975531 -0.00580933  0.042401062 -0.054391321 -0.06330214
## [3,]  0.017140297 -0.06799412  0.02349279 -0.038344709  0.012863424 -0.07784118
## [4,]  0.037922399 -0.06041488  0.03025702 -0.014384378  0.041716273  0.01261646
## [5,] -0.002935307 -0.07629789  0.06179889 -0.034181759 -0.001493857 -0.15509415
## [6,]  0.034441680 -0.06080802  0.02598731  0.019899230 -0.045440364 -0.04778736
##             [,7]         [,8]         [,9]       [,10]       [,11]        [,12]
## [1,] -0.10081835  0.112998425 -0.085864015  0.02212467 -0.12081837  0.022051140
## [2,]  0.05401323 -0.003326599 -0.049954485 -0.03886582 -0.04774171  0.000827258
## [3,] -0.02898089 -0.116886381  0.002134143 -0.04732038 -0.04129926  0.042580166
## [4,] -0.07136432  0.026289927 -0.073089247 -0.02548334 -0.12627437  0.066715587
## [5,]  0.04898925 -0.089368107 -0.121330815  0.02522723  0.03882326 -0.033416425
## [6,] -0.12036950 -0.020614945 -0.041746569  0.06250516 -0.05752680 -0.050494619
##            [,13]       [,14]       [,15]        [,16]       [,17]
## [1,] -0.04012976 0.06906810 -0.01271244  0.008333964 -0.02248712
## [2,] -0.04573386 0.10833858 -0.01111121 -0.016097548 -0.06128763
## [3,]  0.01862796 0.11435517  0.02587727 -0.054561740 -0.05389601
## [4,] -0.03912730 0.08751979 -0.01642437  0.024186697 -0.06425140
## [5,] -0.05998001 0.04222731 -0.05655177 -0.066419196 -0.08372127
## [6,] -0.11379232 0.11579049  0.02824012  0.032433819 -0.07492556
```

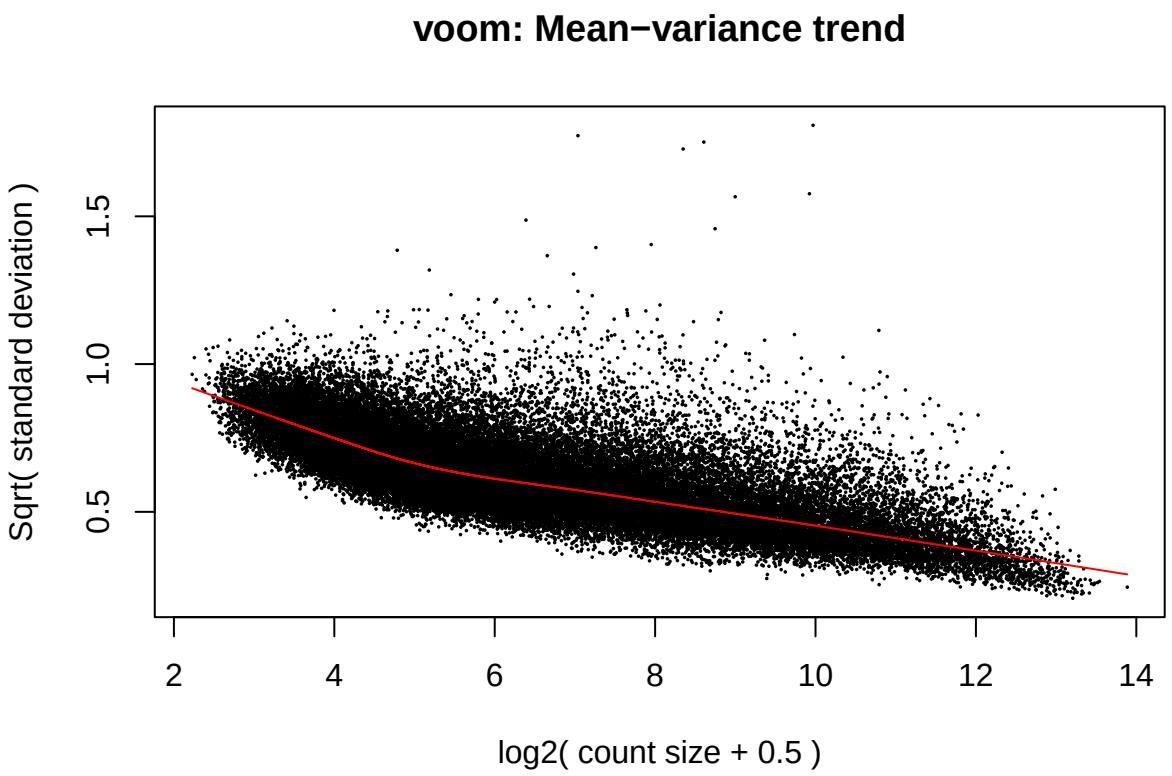
```
# Add SVs to column data information
variables_interest_sva <- DataFrame(variables_interest, res$sv)
```

Differential expression analysis

First we need to create a **design matrix** for the groups (see the excellent limma user guide for more information on design matrices). There are many different ways to set up your design matrix, and **it is dictated by what comparisons you would like to test**.

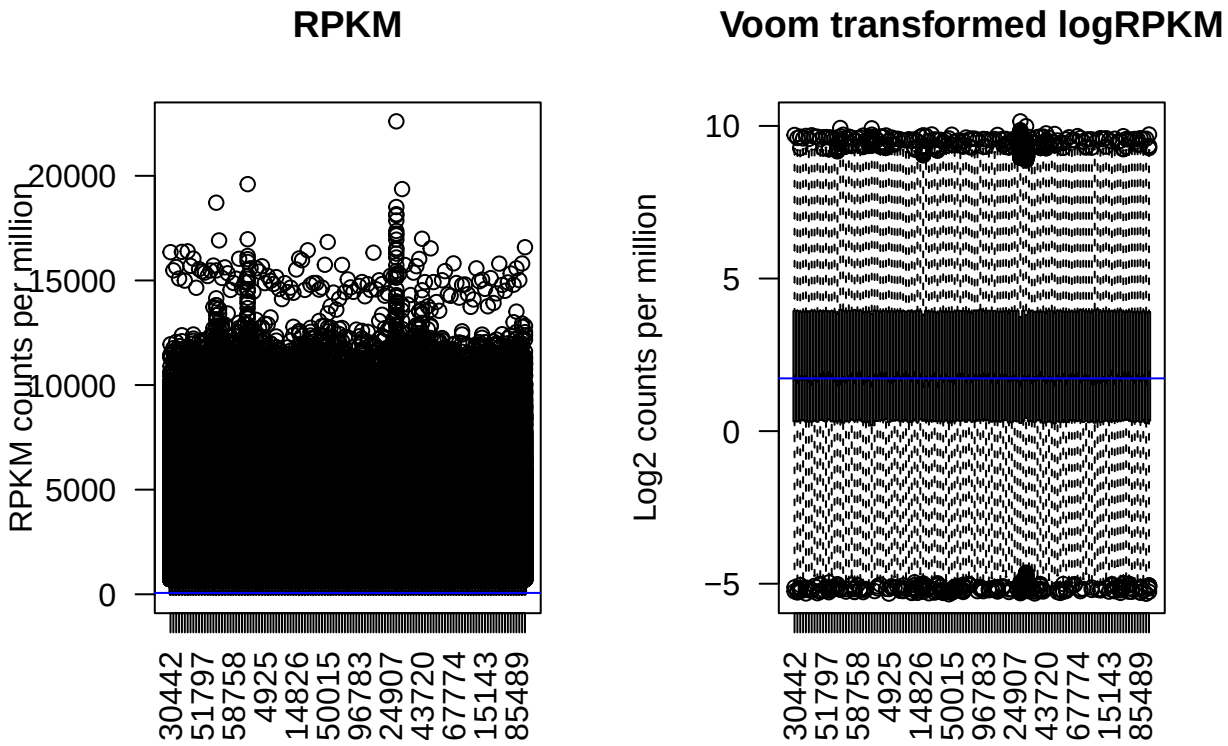
Voom will automatically adjust the library sizes using the `norm.factors` if calculated. We can add `plot=TRUE` to generate a plot of the **mean-variance trend**. This plot can also tell us if there are any genes that look really variable in our data, and if we've filtered the low counts adequately.

```
# Normalize data with voom and add tumor stage as covariate in design
design <- model.matrix(~Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose + dusty_expose +
    history_asthma + Cough + BMI + Cr_wheezeng + PC1_rdr + PC2_rdr + PC3_rdr + PC4_rdr +
    PC5_rdr + PC6_rdr + PC7_rdr + PC8_rdr + PC9_rdr + PC10_rdr + PC11_rdr + PC12_rdr +
    PC13_rdr + PC14_rdr + PC15_rdr + V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 +
    V10 + V11 + V12 + V13 + V14 + V15 + V16 + V17, data = variables_interest_sva)
v <- voom(counts.ok, design = design, plot = TRUE)
```



We can repeat the box plots for the normalised data to compare to before normalisation. The expression values in `v$E` are already **log2 values** so we don't need to log-transform.

```
par(mfrow = c(1, 2))
boxplot(counts.ok, xlab = "", ylab = "RPKM counts per million", las = 2, main = "RPKM")
## Let's add a blue horizontal line that corresponds to the median logCPM
abline(h = median(counts.ok), col = "blue")

boxplot(v$E, xlab = "", ylab = "Log2 counts per million", las = 2, main = "Voom transformed logRPKM")
## Let's add a blue horizontal line that corresponds to the median logCPM
abline(h = median(v$E), col = "blue")
```



`lmFit` estimates group means according to the design matrix, as well as gene-wise variances.

```
fit <- lmFit(v, design)
fit <- eBayes(fit)
```

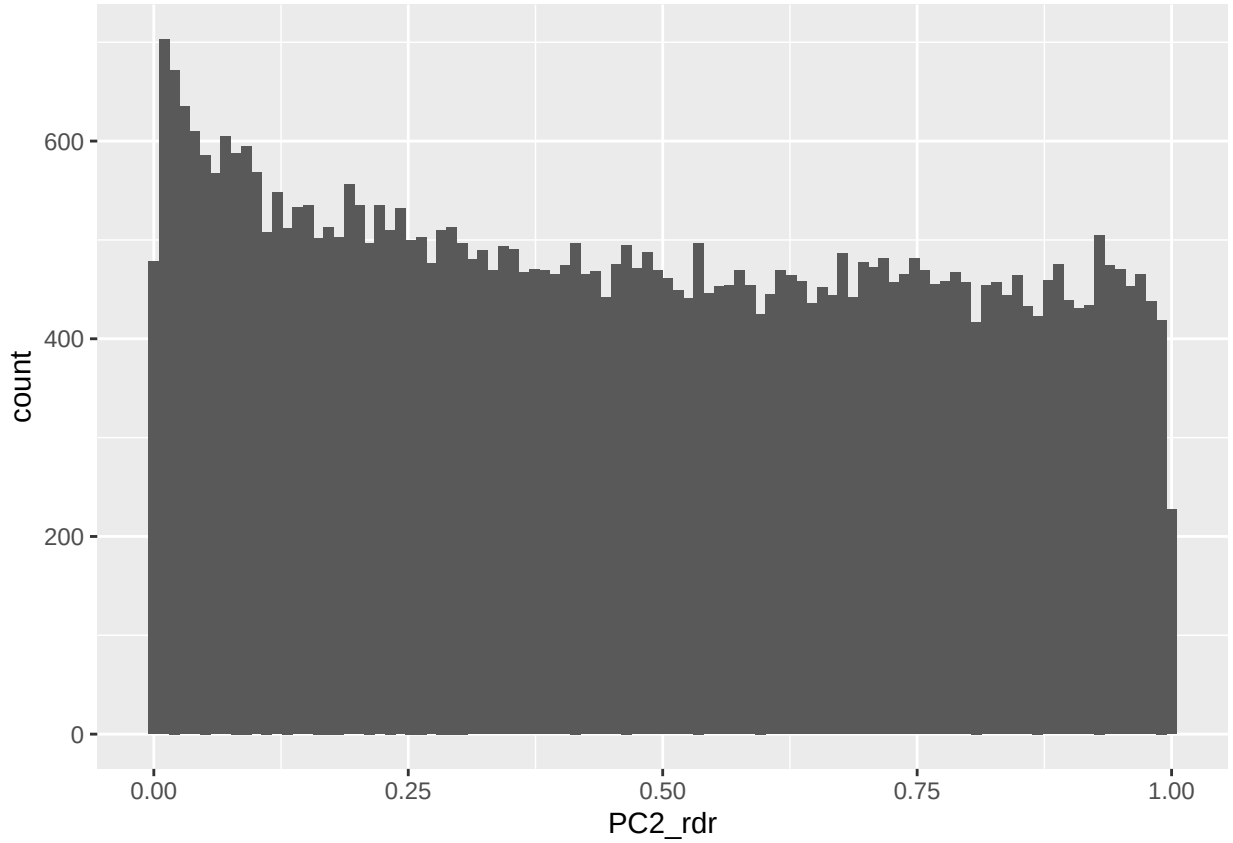
Check variables which are significantly correlated with genes DE:

```
summa.fit <- decideTests(fit)
summary(summa.fit)
```

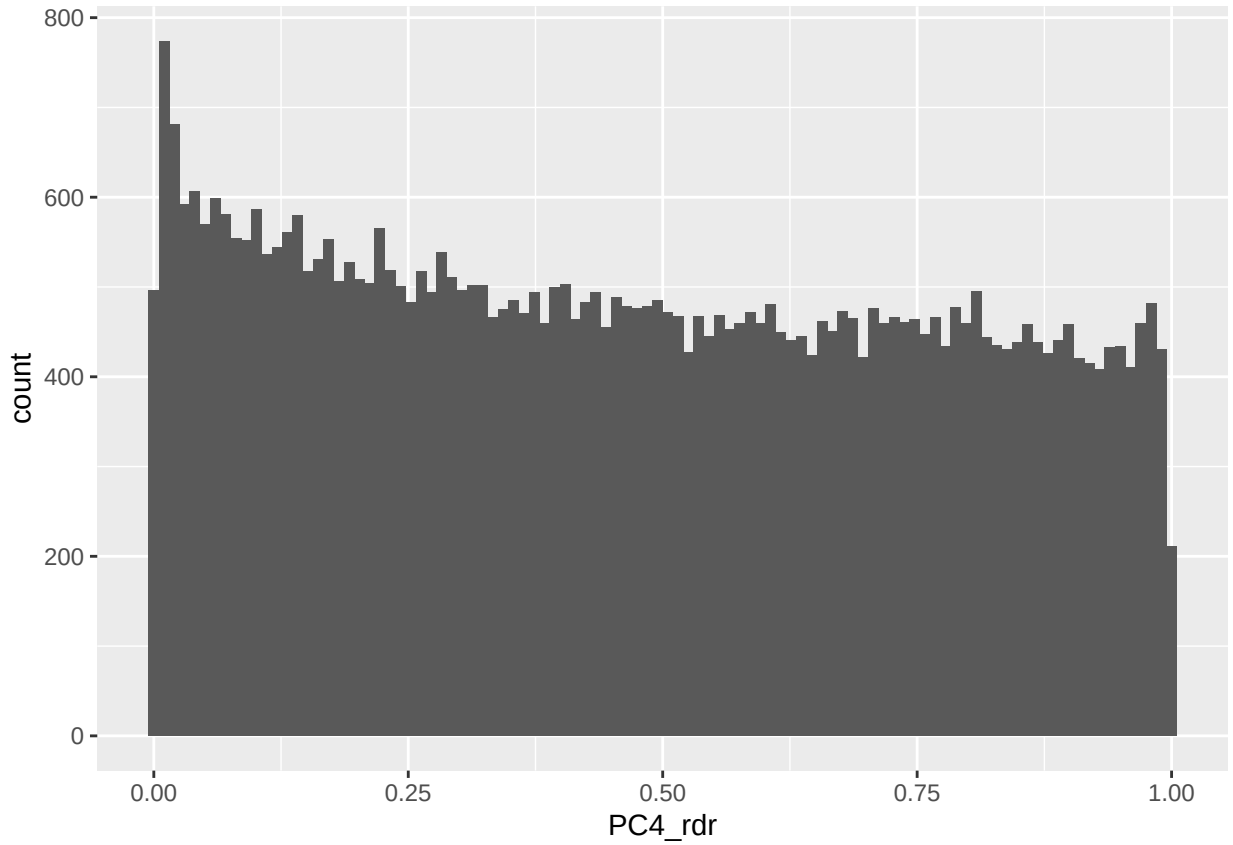
```
##        (Intercept) GroupSevere  SexM   Age Dwalk FEV1PSPC fume_exposeY
## Down           378           0    36     0     0        0           0
## NotSig       24520       48525 48462 48525 48523    48525       48525
## Up           23627           0    27     0     2        0           0
##        dusty_exposeY history_asthmaY CoughNo chronic cough   BMI Cr_wheezengY
## Down               0               0                    0     0            0
## NotSig         48525           48525                48525 48525        48525
## Up                 0               0                    0     0            0
##        PC1_rdr PC2_rdr PC3_rdr PC4_rdr PC5_rdr PC6_rdr PC7_rdr PC8_rdr PC9_rdr
## Down         0       0       0       2       0       0       0       0       0
## NotSig   48525   48522   48525   48522   48525   48525   48525   48524   48524
## Up           0       3       0       1       0       0       0       1       1
##        PC10_rdr PC11_rdr PC12_rdr PC13_rdr PC14_rdr PC15_rdr    V1    V2    V3
## Down          0        0        0        0        0        0 18073 15295 11956
## NotSig    48525    48525    48525    48525    48525    48525 13607 21757 25482
## Up            0        0        0        0        0        0 16845 11473 11087
##           V4    V5    V6    V7    V8    V9   V10   V11   V12   V13   V14   V15
## Down   11067  9107  9665  5745  5921  7031  7047  7677  4890  4340  4922  3617
## NotSig 27876 29577 30436 35358 36279 34481 34227 34492 37861 38651 39054 40802
## Up      9582  9841  8424  7422  6325  7013  7251  6356  5774  5534  4549  4106
##           V16   V17
## Down     3828  2160
```

```
## NotSig 40767 43358
## Up      3930  3007
```

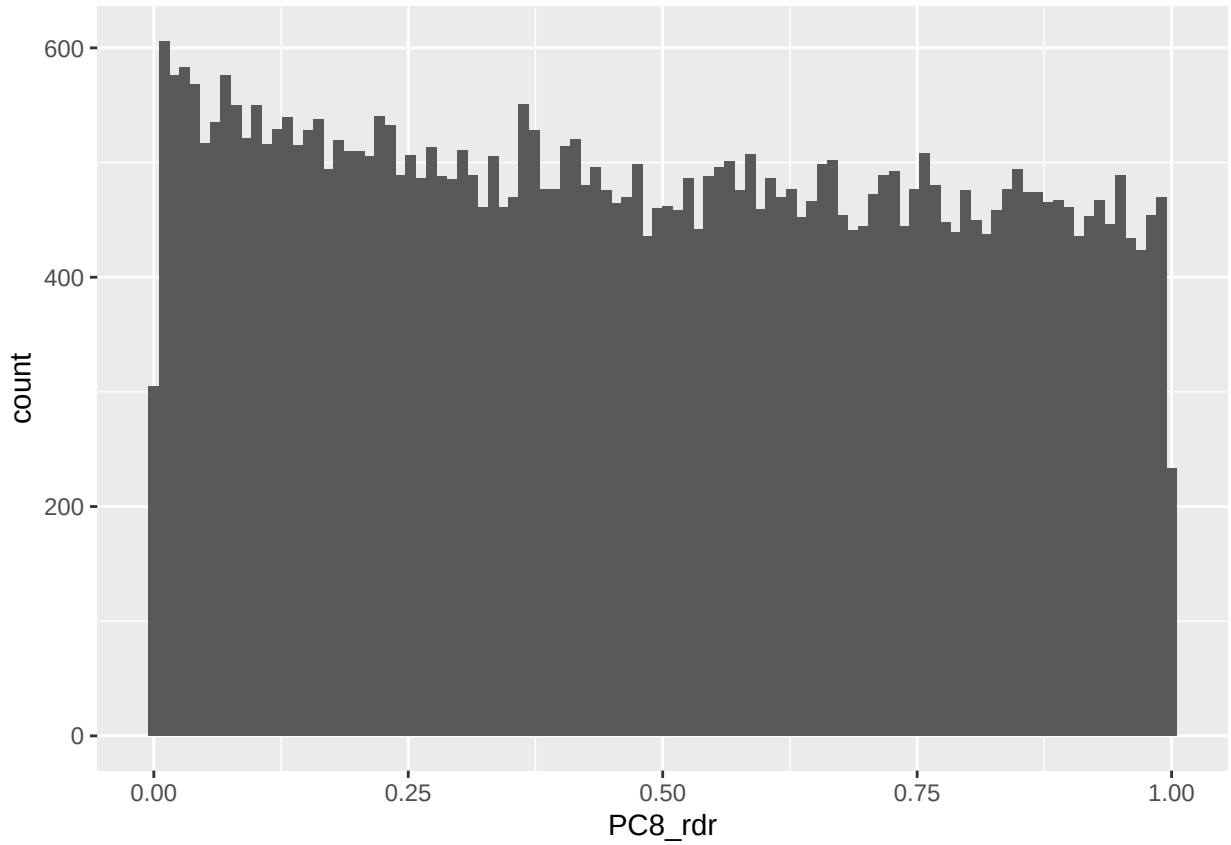
```r
# Get p-values computed by limma
p.val.voom <- as.data.frame(fit$p.value)

# P-value distribution of results computed by limma
ggplot(data = p.val.voom, aes(x = PC2_rdr)) + geom_histogram(bins = 100)
```



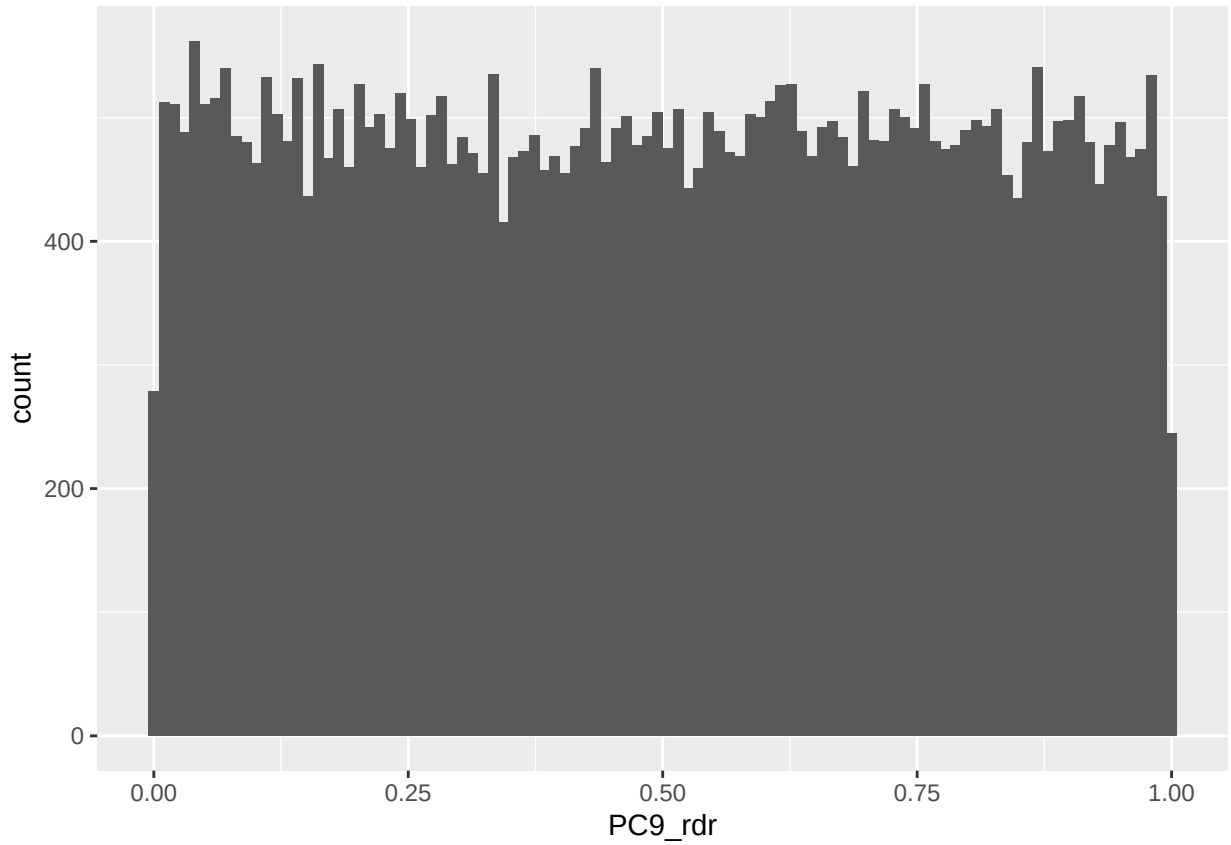

```r
# P-value distribution of results computed by limma
ggplot(data = p.val.voom, aes(x = PC4_rdr)) + geom_histogram(bins = 100)
```



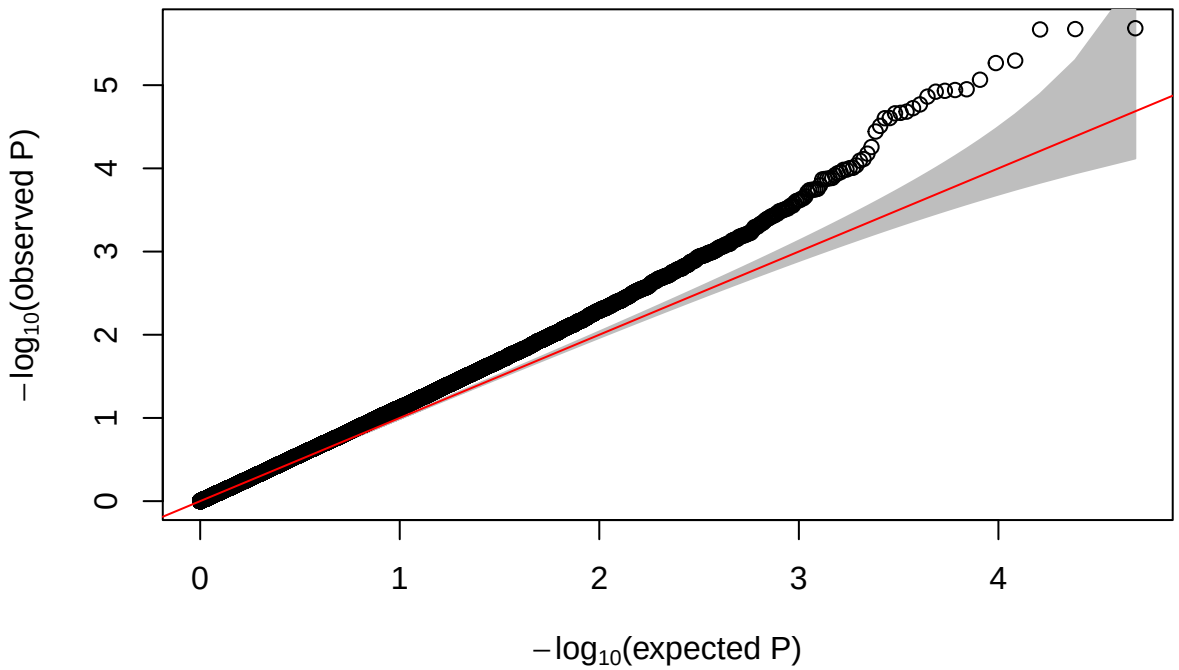

```
# P-value distribution of results computed by limma
ggplot(data = p.val.voom, aes(x = PC8_rdr)) + geom_histogram(bins = 100)
```



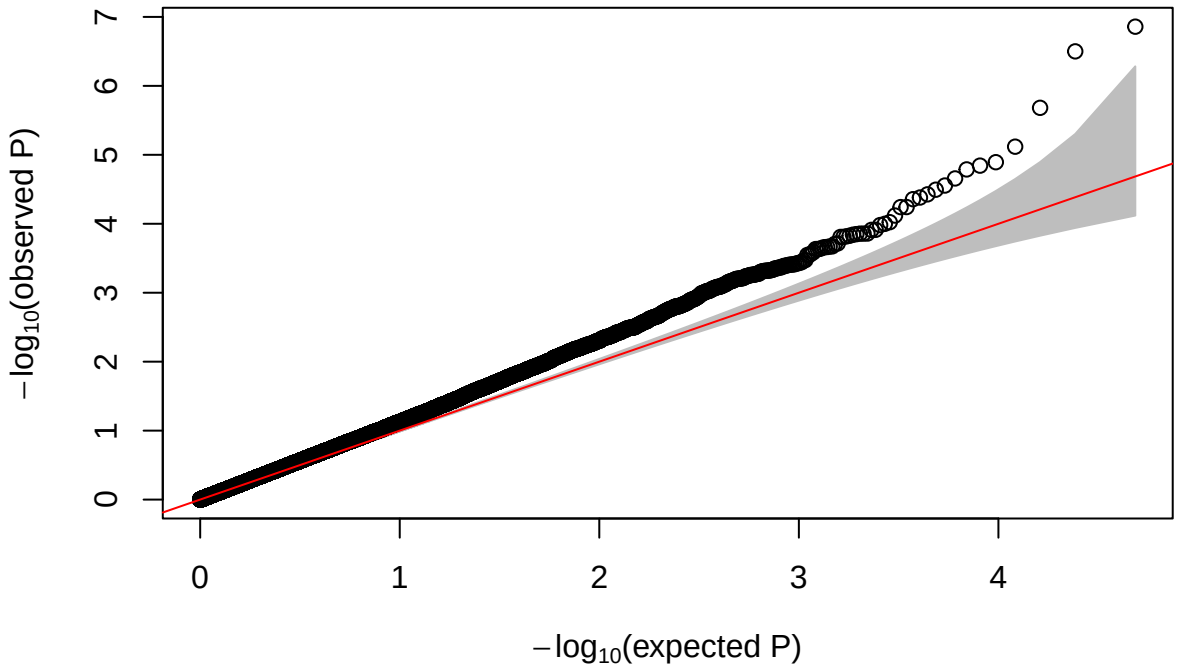

```
# P-value distribution of results computed by limma
ggplot(data = p.val.voom, aes(x = PC9_rdr)) + geom_histogram(bins = 100)
```



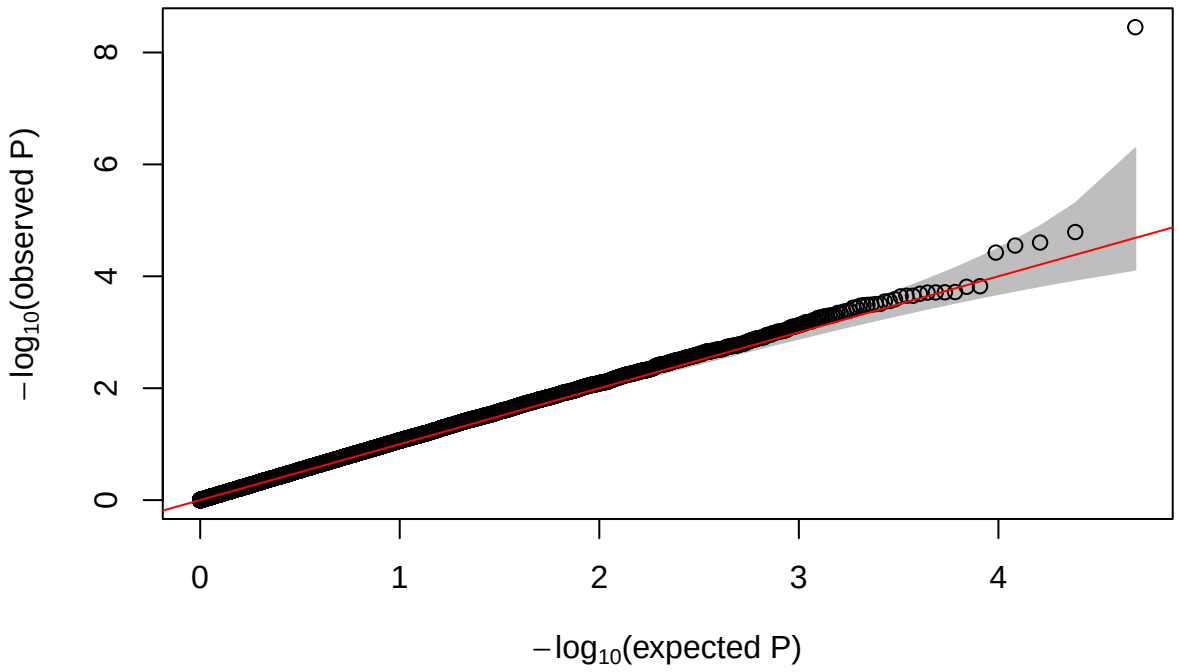

```
# QQplot plot for p-values computed by limma
GWASTools::qqPlot(p.val.voom$PC2_rdr)
```



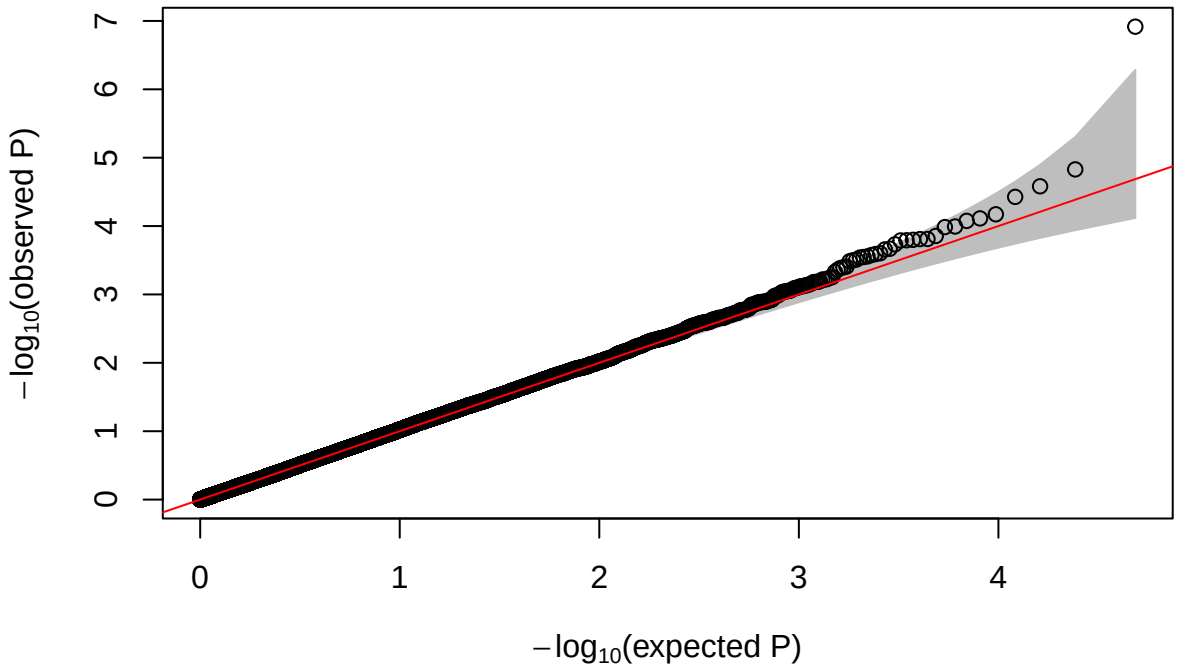

```
# QQplot plot for p-values computed by limma
GWASTools::qqPlot(p.val.voom$PC4_rdr)
```




```
# QQplot plot for p-values computed by limma
GWASTools::qqPlot(p.val.voom$PC8_rdr)
```




```
# QQplot plot for p-values computed by limma
GWASTools::qqPlot(p.val.voom$PC9_rdr)
```



Studying differential expression of one effect

Group Since we are interested in differences between groups, we need to specify which comparisons we want to test. The comparison of interest can be specified using the `makeContrasts` function.

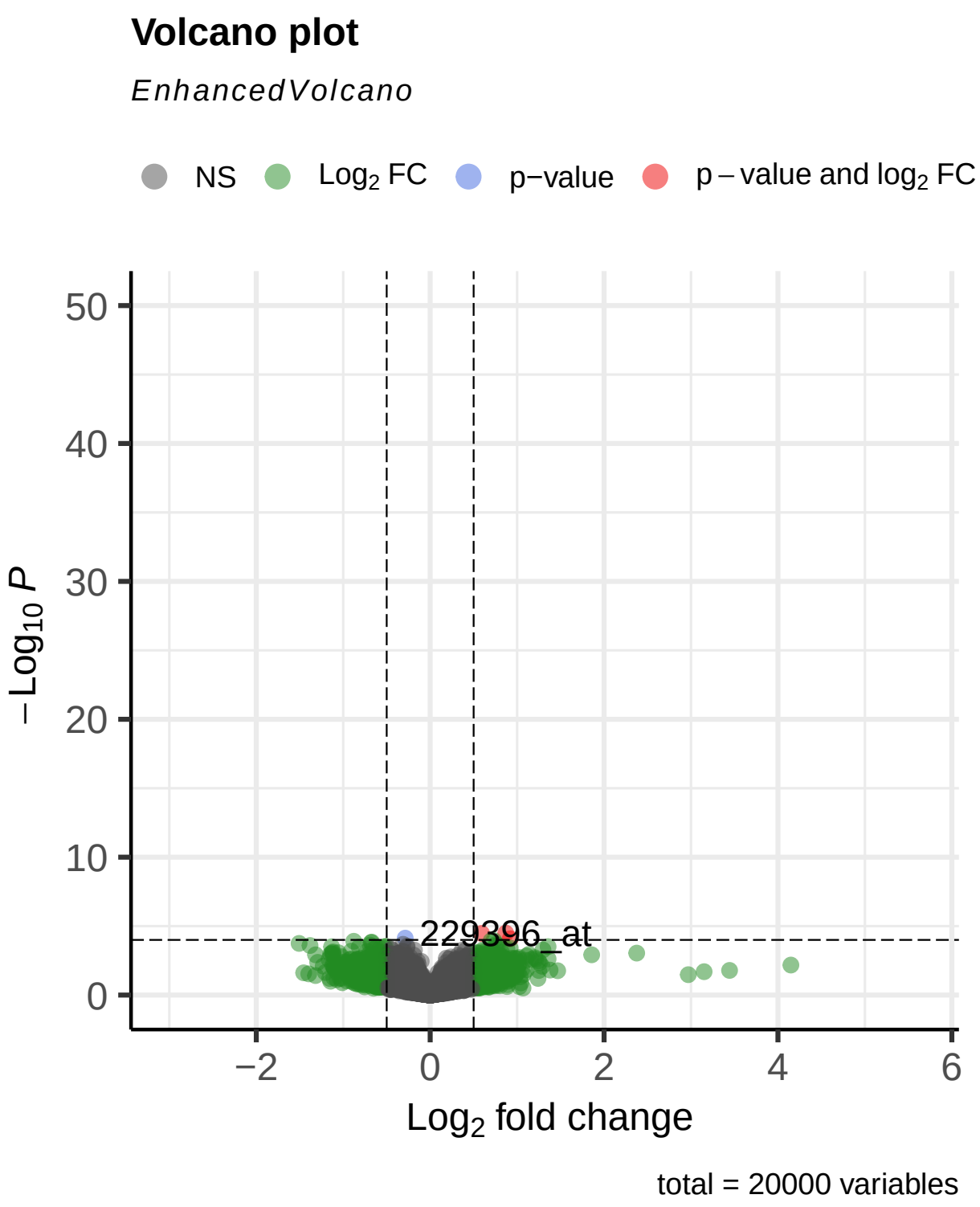
```
# You can check the index of your coefficient
grep("GroupSevere", colnames(fit$coefficients))
```

```
## [1] 2
```

```
groupSevere <- topTable(fit, number = 20000, coef = 2)
groupSevere <- groupSevere[order(groupSevere$P.Value), ]
head(groupSevere)
```

```
##                 logFC  AveExpr         t      P.Value adj.P.Val           B
## 229396_at    0.8698642 1.886358  4.384494 3.232540e-05 0.7246843  0.05557569
## 218174_s_at  0.5917341 2.827671  4.371683 3.391962e-05 0.7246843  0.54771918
## 208634_s_at -0.2870751 6.630862 -4.160397 7.417582e-05 0.7246843  1.29525557
## 233762_at    0.9208280 1.232435  4.128616 8.327943e-05 0.7246843 -0.71142512
## 224444_s_at  0.7328126 2.917643  4.047253 1.117431e-04 0.7246843 -0.19629286
## 222750_s_at  0.7007386 4.753901  4.038339 1.153770e-04 0.7246843  0.45397091
```

```
EnhancedVolcano(groupSevere, lab = rownames(groupSevere), x = "logFC", y = "P.Value",
    ylim = c(0, 50), pCutoff = 1e-04, FCcutoff = 0.5, pointSize = 3, labSize = 6)
```

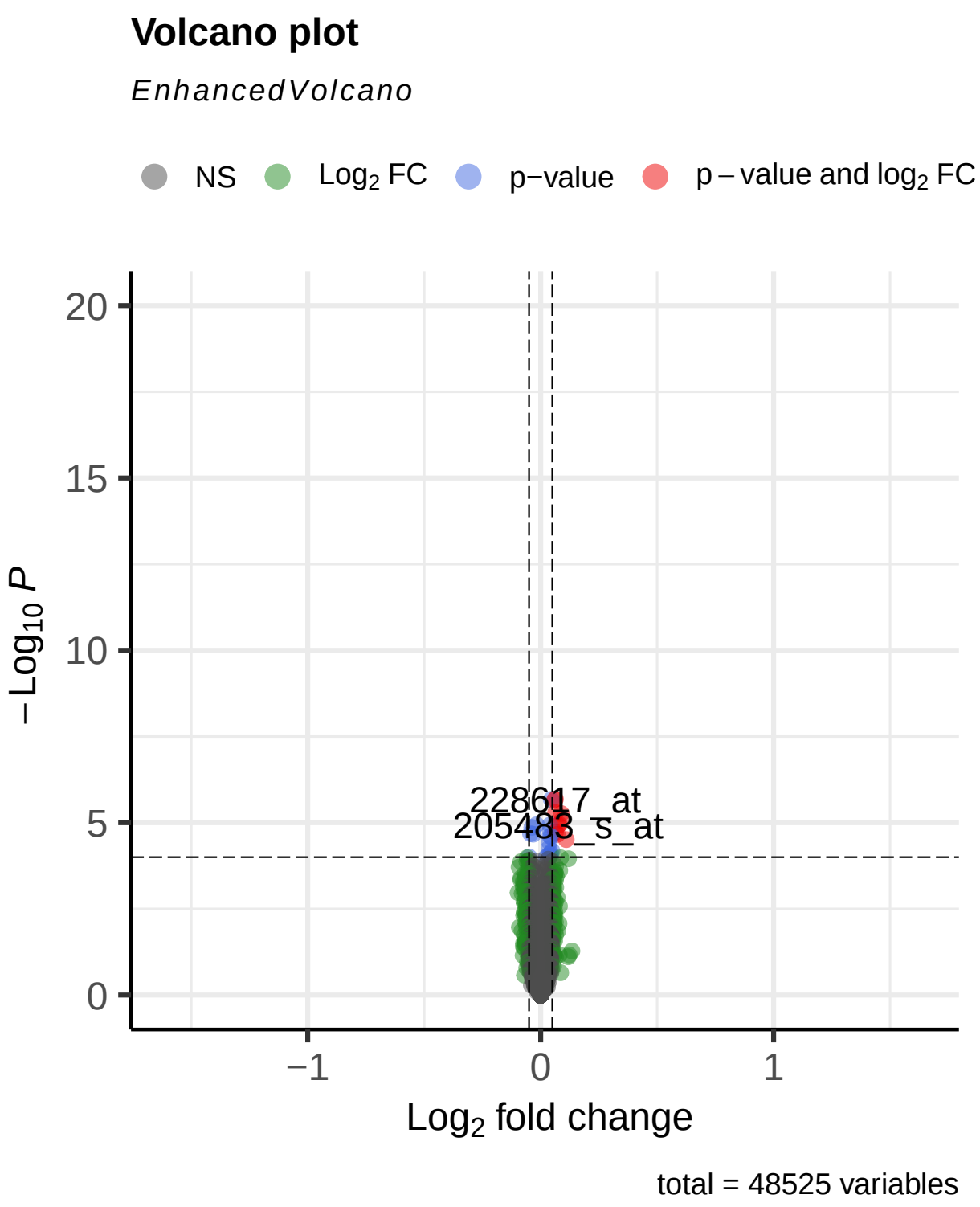



```
groupPC2 <- topTable(fit, number = dim(counts.ok)[1], coef = "PC2_rdr")
groupPC2 <- groupPC2[order(groupPC2$P.Value), ]
head(groupPC2)
```

PC2_rdr

```
##                  logFC    AveExpr        t      P.Value  adj.P.Val        B
## 228617_at   0.06321121  5.7549490 5.086274 2.068628e-06 0.03461886 3.760562
## 230036_at   0.04538590  5.6260651 5.080576 2.117056e-06 0.03461886 3.792488
## 206133_at   0.06040584  5.0826216 5.077890 2.140270e-06 0.03461886 3.844858
## 214453_s_at 0.06432246  5.8042486 4.863309 5.067604e-06 0.05261153 2.873517
## 241358_at   0.08818751 -0.3638065 4.846326 5.421075e-06 0.05261153 3.665487
## 214059_at   0.09101803  3.6797030 4.729274 8.599713e-06 0.05813230 2.699275
```

```
EnhancedVolcano(groupPC2, lab = rownames(groupPC2), x = "logFC", y = "P.Value", ylim = c(0,
    20), pCutoff = 1e-04, FCcutoff = 0.05, pointSize = 3, labSize = 6)
```

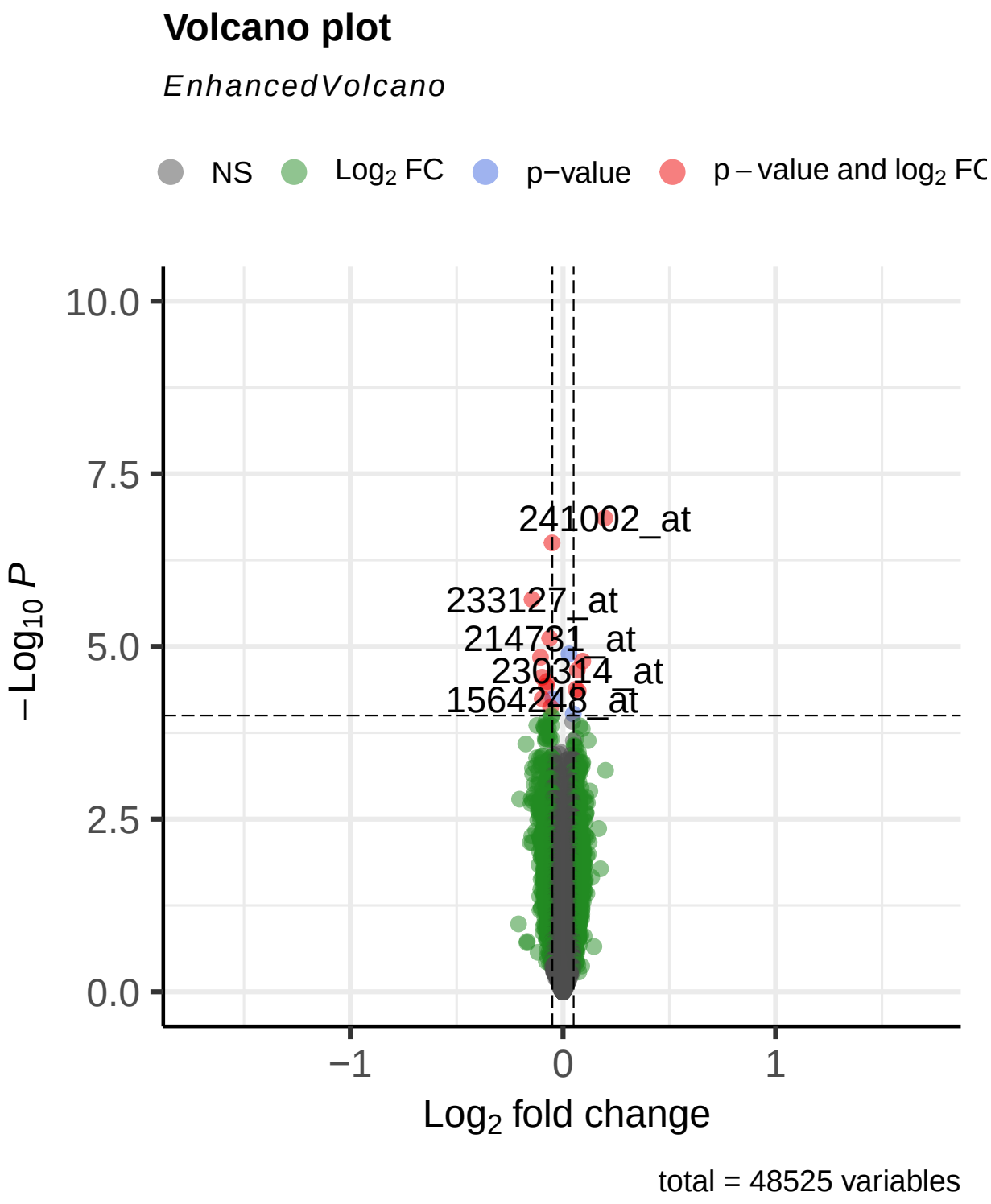



```
groupPC4 <- topTable(fit, number = dim(counts.ok)[1], coef = "PC4_rdr")
groupPC4 <- groupPC4[order(groupPC4$P.Value), ]
head(groupPC4)
```

PC4_rdr

```
##                   logFC    AveExpr         t      P.Value   adj.P.Val        B
## 241002_at    0.19630407 -0.5028479  5.733518 1.385642e-07 0.006723828 7.103723
## 215200_x_at -0.05131442  4.3461624 -5.539584 3.160883e-07 0.007669093 6.327408
## 233127_at   -0.14548039  1.0540609 -5.084710 2.081816e-06 0.033673377 4.721170
## 214731_at   -0.06244183  3.5839259 -4.760093 7.620169e-06 0.092442175 3.341647
## 227244_s_at  0.02838944  5.7003926  4.627257 1.279559e-05 0.112857225 2.597018
## 1564424_at  -0.10592510  1.9773490 -4.597659 1.434682e-05 0.112857225 2.870256
```

```
EnhancedVolcano(groupPC4, lab = rownames(groupPC4), x = "logFC", y = "P.Value", ylim = c(0,
    10), pCutoff = 1e-04, FCcutoff = 0.05, pointSize = 3, labSize = 6)
```



Enrichment

Store gene names of the data set with genes DE:

```
# Store genes with p.adjusted lesser than 0.01 (highly significant)
mask <- groupPC4$adj.P.Val < 0.01 & !is.na(groupPC4$adj.P.Val)
deGenes <- rownames(groupPC4[mask, ])
```

```
# Check differentially expressed genes
head(deGenes)
```

```
## [1] "241002_at"   "215200_x_at"
```

```
length(deGenes)
```

```
## [1] 2
```

Do we have information about these genes?

```
annotation.sputum.ok[deGenes, ]
```

```
## # A tibble: 2 x 9
##   PROBEID     ENSEMBL ENTREZID SYMBOL GENENAME GO    EVIDENCE ONTOLOGY GENETYPE
##   <chr>       <chr>   <chr>    <chr>  <chr>    <chr> <chr>    <chr>    <chr>
## 1 241002_at   ""      ""       ""     ""       ""    ""       ""       ""
## 2 215200_x_at ""      ""       ""     ""       ""    ""       ""       ""
```

Trying another one:

```
# Store genes with p.adjusted lesser than 0.01 (highly significant)
mask <- groupPC2$adj.P.Val < 0.04 & !is.na(groupPC2$adj.P.Val)
deGenes <- rownames(groupPC2[mask, ])

# Check differentially expressed genes
head(deGenes)
```

```
## [1] "228617_at" "230036_at" "206133_at"
```

```
length(deGenes)
```

```
## [1] 3
```

```
# Check info
annotation.sputum.ok[deGenes, ]
```

```
## # A tibble: 3 x 9
##   PROBEID   ENSEMBL    ENTREZID SYMBOL GENENAME GO    EVIDENCE ONTOLOGY GENETYPE
##   <chr>     <chr>      <chr>    <chr>  <chr>    <chr> <chr>    <chr>    <chr>
## 1 228617_at ENSG00000~ 54739    XAF1   XIAP as~ GO:0~ IDA|IEA~ BP|CC|MF protein~
## 2 230036_at ENSG00000~ 219285   SAMD9L sterile~ GO:0~ IBA|IEA~ CC|MF    protein~
## 3 206133_at ENSG00000~ 54739    XAF1   XIAP as~ GO:0~ IDA|IEA~ BP|CC|MF protein~
```

Store gene names of all the data set:

```
geneUniverse <- rownames(groupPC2[!is.na(groupPC2$P.Value), ])
length(geneUniverse)
```

```
## [1] 48525
```

Retrieve `Entrez ID` from the annotations of genes:

```
library(annotate)
library(hgu133plus2.db)

# Change row names of annotation data
rownames(annotation.sputum.ok) <- annotation.sputum.ok$PROBEID

# Retrieve ENTREZ names
```

```
deGenes.entrez <- unlist(mget(deGenes, envir = hgu133plus2ENTREZID, ifnotfound = NA))
geneUniverse.entrez <- unlist(mget(geneUniverse, envir = hgu133plus2ENTREZID, ifnotfound = NA))
```

GO enrichment analysis:

```
library(org.Hs.eg.db)

ans.go <- enrichGO(gene = deGenes.entrez, ont = "MF", OrgDb = "org.Hs.eg.db", readable = TRUE,
    pvalueCutoff = 0.05)

# See results
tab.go <- as.data.frame(ans.go)
tab.go <- subset(tab.go, Count > 5)
tab.go[1:5, 1:6]
```

KEGG enrichment analysis:

```
ans.kegg <- enrichKEGG(gene = deGenes.entrez, organism = "hsa", pvalueCutoff = 0.05)

# See results
tab.kegg <- as.data.frame(ans.kegg)
tab.kegg <- subset(tab.kegg, Count > 5)
tab.kegg[1:5, 1:6]
```

Code to install `disgenet2r`:

```
library(devtools)
install_url("https://cran.r-project.org/src/contrib/Archive/SPARQL/SPARQL_1.16.tar.gz")
install_bitbucket("ibi_group/disgenet2r")
```

DisGeNet analysis. Get API:

```
library(disgenet2r)

# Get API
disgenet_api_key <- get_disgenet_api_key(email = "carlacasanovasuarez@gmail.com",
    password = "mypassword1234", verbose = TRUE)

Sys.setenv(DISGENET_API_KEY = disgenet_api_key)
```

Start enrichment:

```
# Retrieve gene 2 disease
gq <- gene2disease(gene = deGenes.entrez, vocabulary = "ENTREZ", database = "CURATED",
    score = c(0.1, 1))
gq@qresult
```

```
##    protein_class_name gene_symbol disease_type  source geneid protein_class
## 1                  NA      SAMD9L      disease CURATED 219285            NA
## 2                  NA        XAF1      disease CURATED  54739            NA
## 3                  NA        XAF1      disease CURATED  54739            NA
## 4                  NA        XAF1        group CURATED  54739            NA
## 5                  NA        XAF1      disease CURATED  54739            NA
## 6                  NA        XAF1        group CURATED  54739            NA
## 7                  NA        XAF1        group CURATED  54739            NA
## 8                  NA        XAF1      disease CURATED  54739            NA
## 9                  NA        XAF1      disease CURATED  54739            NA
```

```
## 10                NA        XAF1      disease CURATED  54739            NA
## 11                NA        XAF1      disease CURATED  54739            NA
## 12                NA        XAF1      disease CURATED  54739            NA
##    uniprotid disease_class gene_dsi   gene_pli score ei year_initial gene_dpi
## 1     Q8IVG5   C23;C10;C15    0.700 5.5651e-15  0.72  1         2016    0.500
## 2     Q6GPH4       C06;C04    0.587 6.8151e-12  0.35  1         2005    0.769
## 3     Q6GPH4       C06;C04    0.587 6.8151e-12  0.35  1         2006    0.769
## 4     Q6GPH4       C06;C04    0.587 6.8151e-12  0.33  1         2005    0.769
## 5     Q6GPH4       C06;C04    0.587 6.8151e-12  0.33  1         2007    0.769
## 6     Q6GPH4       C06;C04    0.587 6.8151e-12  0.31  1         2006    0.769
## 7     Q6GPH4       C06;C04    0.587 6.8151e-12  0.30  1         2007    0.769
## 8     Q6GPH4           C04    0.587 6.8151e-12  0.30  1         2017    0.769
## 9     Q6GPH4       C01;C08    0.587 6.8151e-12  0.30  1         2013    0.769
## 10    Q6GPH4           C04    0.587 6.8151e-12  0.30  1         2017    0.769
## 11    Q6GPH4           C04    0.587 6.8151e-12  0.30  1         2017    0.769
## 12    Q6GPH4       C06;C04    0.587 6.8151e-12  0.30  1         2006    0.769
##    disease_semantic_type                      disease_name year_final
## 1    Disease or Syndrome          Myelocerebellar Disorder       2019
## 2     Neoplastic Process          Malignant tumor of colon       2010
## 3     Neoplastic Process     Malignant neoplasm of stomach       2015
## 4     Neoplastic Process                Colonic Neoplasms       2017
## 5     Neoplastic Process              Colorectal Carcinoma       2019
## 6     Neoplastic Process                 Stomach Neoplasms       2006
## 7     Neoplastic Process              Colorectal Neoplasms       2007
## 8     Neoplastic Process                             Glioma       2017
## 9    Disease or Syndrome                          Influenza       2013
## 10    Neoplastic Process                     mixed gliomas       2017
## 11    Neoplastic Process                   Malignant Glioma       2017
## 12    Neoplastic Process Hereditary Diffuse Gastric Cancer       2006
##                                                                                                                  disease_
## 1     Pathological Conditions, Signs and Symptoms;    Nervous System Diseases;    Hemic and Lymphati
## 2                                                                                  Digestive System Diseases;
## 3                                                                                  Digestive System Diseases;
## 4                                                                                  Digestive System Diseases;
## 5                                                                                  Digestive System Diseases;
## 6                                                                                  Digestive System Diseases;
## 7                                                                                  Digestive System Diseases;
## 8
## 9                                                                               Infections;    Respiratory Trac
## 10
## 11
## 12                                                                                 Digestive System Diseases;
##    diseaseid el
## 1   C1327919 NA
## 2   C0007102 NA
## 3   C0024623 NA
## 4   C0009375 NA
## 5   C0009402 NA
## 6   C0038356 NA
## 7   C0009404 NA
## 8   C0017638 NA
## 9   C0021400 NA
## 10  C0259783 NA
## 11  C0555198 NA
```

```
## 12  C1708349 NA
```

```
res_enrich <- disease_enrichment(entities = deGenes.entrez, vocabulary = "ENTREZ",
    database = "CURATED")

res.table.disgenet <- res_enrich@qresult
head(res.table.disgenet)
```

```
##          ID              Description  source Ratio  BgRatio       pvalue
## 11 C1327919 Myelocerebellar Disorder CURATED   1/2   1/9703 0.0002061431
## 5  C0017638                   Glioma CURATED   1/2  87/9703 0.0178549515
## 6  C0021400                Influenza CURATED   1/2  52/9703 0.0106912622
## 9  C0259783            mixed gliomas CURATED   1/2  70/9703 0.0143786965
## 10 C0555198         Malignant Glioma CURATED   1/2  70/9703 0.0143786965
## 1  C0007102 Malignant tumor of colon CURATED   1/2 159/9703 0.0325098299
##             FDR disease_class
## 11 0.002473717   C10;C15;C23
## 5  0.042851884           C04
## 6  0.042851884       C01;C08
## 9  0.042851884           C04
## 10 0.042851884           C04
## 1  0.055731137       C04;C06
##                                                                                  disease_class_name
## 11 Nervous System Diseases;Hemic and Lymphatic Diseases;Pathological Conditions, Signs and Symptoms
## 5                                                                                        Neoplasms
## 6                                                          Infections;Respiratory Tract Diseases
## 9                                                                                        Neoplasms
## 10                                                                                       Neoplasms
## 1                                                          Neoplasms;Digestive System Diseases
##    disease_semantic_type shared_geneid shared_symbol Count  gg
## 11   Disease or Syndrome        219285        SAMD9L     1 0.5
## 5     Neoplastic Process         54739          XAF1     1 0.5
## 6    Disease or Syndrome         54739          XAF1     1 0.5
## 9     Neoplastic Process         54739          XAF1     1 0.5
## 10    Neoplastic Process         54739          XAF1     1 0.5
## 1     Neoplastic Process         54739          XAF1     1 0.5
```

MSigDb enrichment analysis (group C7 inmunologic signature):

```
# Load terms for genes coded by ENTREZ of the gene set C7
c3.tf <- read.gmt("/Users/carlacasanovasuarez/Downloads/c7.all.v7.5.1.entrez.gmt.txt")

# Enrichment
ans.tf <- enricher(deGenes.entrez, TERM2GENE = c3.tf)
tab.tf <- as.data.frame(ans.tf@result)
# tab.tf<- subset(tab.tf, Count>5)
tab.tf[1:5, 1:5]
```

```
##
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NON_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEI
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NAIVE_NOT_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS_
## HOWARD_T_CELL_INACT_MONOV_INFLUENZA_A_INDONESIA_05_2005_H5N1_AGE_18_49YO_1DY_UP
## QUEREC_PBMC_YF_17D_VACCINE_AGE_18_45YO_3DY_UP
##
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NON_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEI
```

```
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NAIVE_NOT_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS_
## HOWARD_T_CELL_INACT_MONOV_INFLUENZA_A_INDONESIA_05_2005_H5N1_AGE_18_49YO_1DY_UP
## QUEREC_PBMC_YF_17D_VACCINE_AGE_18_45YO_3DY_UP
##
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NON_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEI
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NAIVE_NOT_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS_
## HOWARD_T_CELL_INACT_MONOV_INFLUENZA_A_INDONESIA_05_2005_H5N1_AGE_18_49YO_1DY_UP
## QUEREC_PBMC_YF_17D_VACCINE_AGE_18_45YO_3DY_UP
##
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NON_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEI
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NAIVE_NOT_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS_
## HOWARD_T_CELL_INACT_MONOV_INFLUENZA_A_INDONESIA_05_2005_H5N1_AGE_18_49YO_1DY_UP
## QUEREC_PBMC_YF_17D_VACCINE_AGE_18_45YO_3DY_UP
##
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NON_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEI
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_RESPONDERS_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS
## ERWIN_COHEN_PBMC_TC_83_AGE_18_45YO_NAIVE_NOT_PREVIOUSLY_IMMUNIZED_24HR_DEG_CANONICAL_PATHWAY_MEMBERS_
## HOWARD_T_CELL_INACT_MONOV_INFLUENZA_A_INDONESIA_05_2005_H5N1_AGE_18_49YO_1DY_UP
## QUEREC_PBMC_YF_17D_VACCINE_AGE_18_45YO_3DY_UP
```

DE using cluster

```r
hclustCarlos <- read.delim("/Users/carlacasanovasuarez/Downloads/consensus_class_k20.csv",
    sep = ",")
```

```r
# Change radiomic features' names to be equal than rdr object
hclustCarlos.s <- hclustCarlos[, 1:2]
hclustCarlos.s[22, 1] <- "10Percentile.original"
hclustCarlos.s[23, 1] <- "90Percentile.original"

# Create a table where clusters are columns and radiomic features belonging to
# that cluster in rows
df.wide <- pivot_wider(hclustCarlos.s, names_from = x, values_from = X)

# Select clusters with more than 2 radiomic features
clust.interest <- apply(df.wide, 2, function(x) {
    length(unlist(x)) > 2
})
df.wide.ok <- df.wide[, clust.interest]

# Create a list with significative clusters and an empty data frame
clusters.rf <- colnames(df.wide.ok)
dd <- data.frame(patients = colnames(rdr_assay), stringsAsFactors = FALSE)

# Iterate over the data frame with cluster information for computing the mean
# of all the radiomic features belonging to each cluster. Radiomic features
# must be scaled (to allow comparisions)
for (i in 1:length(clusters.rf)) {
    dd[, i + 1] <- cbind(apply(rdr_assay[unlist(df.wide.ok[, i]), ], 2, mean))
}
```

```r
# Change row and col names
rownames(dd) <- dd$patients
dd$patients <- NULL
colnames(dd) <- clusters.rf

# Check if patients from the resulted data frame are ordered as pheno data
identical(rownames(dd), rownames(phenoDataFrame))
```

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

```r
head(dd)
```

```
##                1          3           5            6          7          8
## 30442 -0.4577654  0.2811301 -0.12341086 -0.008482849 -0.5032941 0.11594643
## 85931  0.1616485 -0.5909636 -0.01480877 -0.014427366 -0.4615389 0.19497340
## 4428   1.5255348 -0.2227635 -0.58845249 -0.085338421 -0.4099109 0.09194633
## 41420 -0.6135493  0.5970924  0.05063508  0.040143273 -0.4493625 0.42350746
## 15405 -0.3961501 -0.8120081  0.16243595  0.012258889 -0.4073348 0.55083648
## 26383  0.2726520 -0.8067956 -0.66252031  0.657049555 -1.2037555 0.31375535
##               10         11          12
## 30442 -0.10396808 -0.6456234 0.319859683
## 85931 -0.15681280 -0.6657433 0.324997394
## 4428  -0.09861258 -0.8110482 0.361222649
## 41420 -0.25437218 -0.5750900 0.264233920
## 15405 -0.44711160 -0.6956447 0.308133981
## 26383 -0.27744080 -0.1125143 0.006748047
```

```r
variables_interest <- data.frame(Sex = as.factor(phenoDataFrame$SEX), Age = phenoDataFrame$AGE,
    Dwalk = phenoDataFrame$DWALK, BMI = phenoDataFrame$BMI, Smoke_history = phenoDataFrame$SUSMHS,
    Years_smoke = phenoDataFrame$SUSMYR, Cough = as.factor(phenoDataFrame$COUGH),
    ex_first_year = phenoDataFrame$Y1EXBS, Group = as.factor(phenoDataFrame$GROUP_bin),
    N_cigarrete_day = phenoDataFrame$SUCGSMDY, Cr_bronchitis = as.factor(phenoDataFrame$CBRONCH),
    Cr_wheezeng = as.factor(phenoDataFrame$ATS3EG), history_asthma = as.factor(phenoDataFrame$ATS5G),
    fume_expose = as.factor(phenoDataFrame$ATS6C), dusty_expose = as.factor(phenoDataFrame$ATS6B),
    phlegm = as.factor(phenoDataFrame$PHLEGM), heart_failure = as.factor(phenoDataFrame$ATS8F),
    stroke = as.factor(phenoDataFrame$ATS8E), diabetes = as.factor(phenoDataFrame$ATS8L),
    osteoporosis = as.factor(phenoDataFrame$ATS8H), FEVVCPD = phenoDataFrame$FEVVCPD,
    FEV1PSPC = phenoDataFrame$FEV1PSPC, TLC = phenoDataFrame$TLC, FRC = phenoDataFrame$FRC,
    Low_percentile = phenoDataFrame$LOW15PCT, clust1 = dd$`1`, clust3 = dd$`3`, clust5 = dd$`5`,
    clust6 = dd$`6`, clust7 = dd$`7`, clust8 = dd$`8`, clust10 = dd$`10`, clust11 = dd$`11`,
    clust12 = dd$`12`, row.names = rownames(phenoDataFrame))

# Code for removing columns with some NA
variables_interest <- variables_interest[, apply(variables_interest, 2, function(x) !any(is.na(x)))]

## Select counts
countData <- counts.ok[rowSums(counts.ok) > 10, ]

# count unique values for each variable
sapply(lapply(variables_interest, unique), length)
```

```
##             Sex             Age           Dwalk             BMI   Smoke_history
##               2              25              98             124               1
##     Years_smoke           Cough           Group N_cigarrete_day   Cr_bronchitis
##              37               2               2              18               2
```

```
##     Cr_wheezeng  history_asthma     fume_expose    dusty_expose          phlegm
##               2               2               2               2               2
##   heart_failure          stroke        diabetes    osteoporosis         FEVVCPD
##               2               2               2               2              47
##        FEV1PSPC          clust1          clust3          clust5          clust6
##             115             125             125             125             125
##          clust7          clust8         clust10         clust11         clust12
##             125             125             125             125             125
```

```
# Remove variables with just one factor
variables_interest[, "Smoke_history"] <- NULL
```

Functions for assessing linear models:

```
# RSS se ajusta mejor si es más bajo
rss <- function(fitted, actual) {
    sum((fitted - actual)^2)
}

# RMSE se ajusta mejor si es más bajo
rmse <- function(fitted, actual) {
    sqrt(mean((fitted - actual)^2))
}
```

Let's start exploring manually variables:

```
mod.s <- lm(t(countData) ~ Sex, data = variables_interest)
mod.sa <- lm(t(countData) ~ Sex + Age, data = variables_interest)
mod.a <- lm(t(countData) ~ Age, data = variables_interest)
mod.g <- lm(t(countData) ~ Group, data = variables_interest)
mod.sag <- lm(t(countData) ~ Group + Sex + Age, data = variables_interest)
mod.dwalk <- lm(t(countData) ~ Dwalk, data = variables_interest)
mod.dsag <- lm(t(countData) ~ Group + Sex + Age + Dwalk, data = variables_interest)
mod.dag <- lm(t(countData) ~ Group + Age + Dwalk, data = variables_interest)
mod.dsagf <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC, data = variables_interest)
mod.dsagffd <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose, data = variables_interest)
mod.dsagffda <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma, data = variables_interest)
# Trying just radiomic features
mod.clust <- lm(t(countData) ~ clust1 + clust3 + clust5 + clust6 + clust7 + clust8 +
    clust10 + clust11 + clust12, data = variables_interest)
# Adding the rest + radiomic features: without Cough
mod.dsagffdaclust <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma + clust1 + clust3 + clust5 + clust6 + clust7 +
    clust8 + clust10 + clust11 + clust12, data = variables_interest)
# Without Dwalk and FEV1PSPC but Cough
mod.sagfdacclust <- lm(t(countData) ~ Group + Sex + Age + fume_expose + dusty_expose +
    history_asthma + Cough + clust1 + clust3 + clust5 + clust6 + clust7 + clust8 +
    clust10 + clust11 + clust12, data = variables_interest)
# ALL previous + clusters
mod.dsagffdacclust <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose +
    dusty_expose + history_asthma + Cough + clust1 + clust3 + clust5 + clust6 + clust7 +
    clust8 + clust10 + clust11 + clust12, data = variables_interest)
# All + PCs + BMI + Cr_wheezeng
mod.dsagffdacBWclust <- lm(t(countData) ~ Group + Sex + Age + Dwalk + FEV1PSPC +
```

```
    fume_expose + dusty_expose + history_asthma + Cough + BMI + Cr_wheezeng + clust1 +
    clust3 + clust5 + clust6 + clust7 + clust8 + clust10 + clust11 + clust12, data = variables_interest)

rmse(fitted(mod.s), t(countData))
```

```
## [1] 254.2539
```

```
rmse(fitted(mod.sa), t(countData))
```

```
## [1] 252.6789
```

```
rmse(fitted(mod.a), t(countData))
```

```
## [1] 254.0286
```

```
rmse(fitted(mod.g), t(countData))
```

```
## [1] 244.71
```

```
rmse(fitted(mod.sag), t(countData))
```

```
## [1] 240.8153
```

```
rmse(fitted(mod.dwalk), t(countData))
```

```
## [1] 252.8118
```

```
rmse(fitted(mod.dsag), t(countData))
```

```
## [1] 239.7644
```

```
rmse(fitted(mod.dag), t(countData))
```

```
## [1] 241.4093
```

```
rmse(fitted(mod.dsagf), t(countData))
```

```
## [1] 238.4131
```

```
rmse(fitted(mod.dsagffd), t(countData))
```

```
## [1] 235.1352
```

```
rmse(fitted(mod.dsagffda), t(countData))
```

```
## [1] 233.4724
```

```
rmse(fitted(mod.clust), t(countData))
```

```
## [1] 244.0044
```

```
rmse(fitted(mod.dsagffdaclust), t(countData))
```

```
## [1] 221.4942
```

```
rmse(fitted(mod.sagfdacclust), t(countData))
```

```
## [1] 222.9751
```

```
rmse(fitted(mod.dsagffdacclust), t(countData))
```

```
## [1] 220.5864
```

```
# Best model
rmse(fitted(mod.dsagffdacBWclust), t(countData))
```

```
## [1] 219.3151
```

Perform SVA analysis:

```
# Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose + dusty_expose +
# history_asthma +

# Create a model with tumor stage as covariate and a null model
mod1 <- model.matrix(~Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose + dusty_expose +
    history_asthma + Cough + BMI + Cr_wheezeng + clust1 + clust3 + clust5 + clust6 +
    clust7 + clust8 + clust10 + clust11 + clust12, data = variables_interest)
mod0 <- model.matrix(~Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose + dusty_expose +
    history_asthma + Cough + BMI + Cr_wheezeng, data = variables_interest)

# This estimates the required number of surrogate variables countData <-
# counts.ok[rowSums(counts.ok) > 10, ]
res <- svaseq(countData, mod1, mod0)
```

```
## Number of significant surrogate variables is:  18
## Iteration (out of 5 ):1  2  3  4  5
```

```
# Number of estimated surrogate variables
res$n.sv
```

```
## [1] 18
```

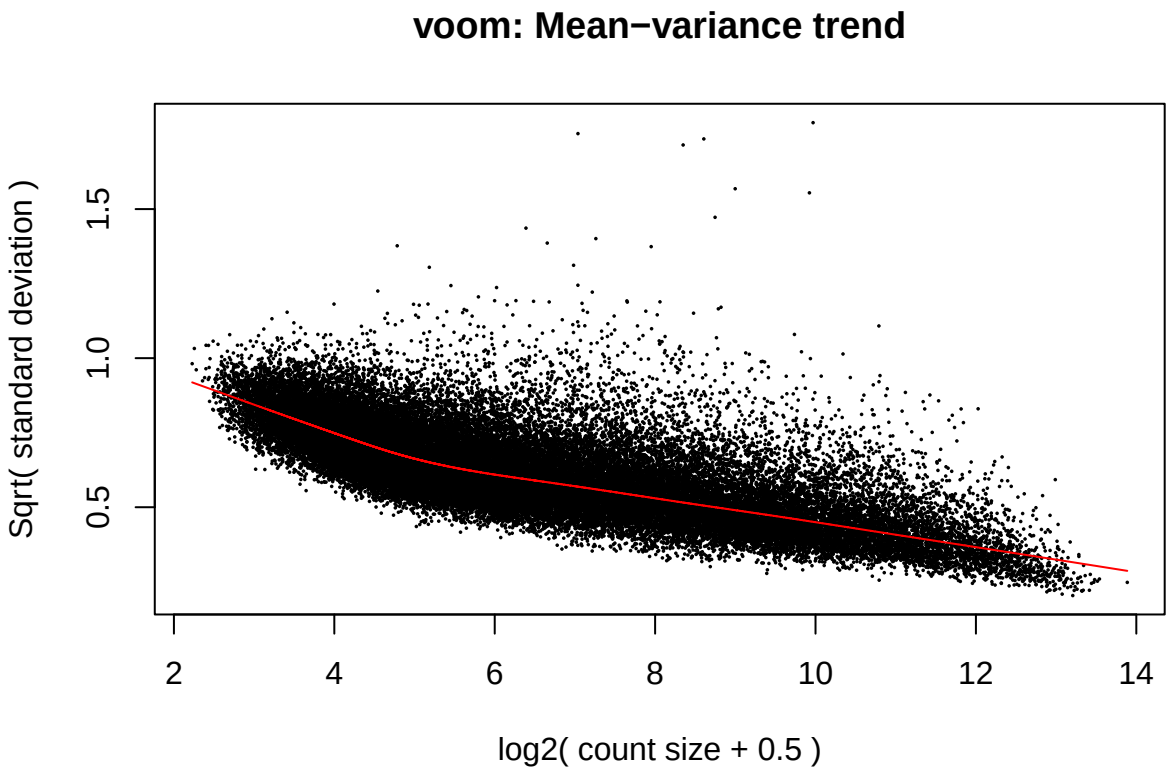
```
# Check surrogated variables
head(res$sv)
```

```
##              [,1]        [,2]         [,3]         [,4]        [,5]
## [1,]  0.079811938 -0.02011154  0.043700171 -0.004090236  0.01986034
## [2,] -0.074504467 -0.09926864 -0.004085109  0.039324523 -0.04767339
## [3,]  0.017160030 -0.06677548  0.023390022 -0.035927749  0.01639599
## [4,]  0.038023186 -0.06047352  0.030791514 -0.012879302  0.03774572
## [5,] -0.002116841 -0.07751261  0.059228565 -0.031603322  0.01036891
## [6,]  0.035247002 -0.06070390  0.025595457  0.021628782 -0.04910716
##              [,6]        [,7]        [,8]        [,9]        [,10]       [,11]
## [1,]  0.027929249 -0.10790312  0.10143986 -0.08267387  0.037134207 -0.12305413
## [2,] -0.061295861  0.06160970 -0.00960097 -0.05055522 -0.031511689 -0.05524857
## [3,] -0.077150389 -0.02247582 -0.12218631  0.01245616 -0.040154393 -0.04707940
## [4,]  0.008125893 -0.07555953  0.01479009 -0.06517737 -0.004949119 -0.13137972
## [5,] -0.151925041  0.05673998 -0.09633060 -0.12095975  0.024253989  0.04116172
## [6,] -0.055722737 -0.11491303 -0.02082525 -0.03551992  0.063668461 -0.05395586
##             [,12]       [,13]      [,14]        [,15]        [,16]       [,17]
## [1,]  0.030693155 -0.03647763 0.06634999 -0.009531323  0.002633611 -0.02411800
## [2,]  0.007754208 -0.04004598 0.11228305 -0.005118779 -0.023078745 -0.07587110
## [3,]  0.043637149  0.02563045 0.11065032  0.032365485 -0.057274969 -0.05984782
## [4,]  0.073630242 -0.03493240 0.08568404 -0.015430967  0.013248313 -0.07288268
## [5,] -0.027997807 -0.05683361 0.05332355 -0.049631606 -0.076522318 -0.07609520
## [6,] -0.044096734 -0.10708397 0.11474504  0.035463093  0.026676125 -0.08061301
##             [,18]
## [1,]  0.03953956
## [2,] -0.08083412
## [3,] -0.10305893
## [4,]  0.08243837
## [5,] -0.11542691
## [6,]  0.03328825
```

```
# Add SVs to column data information
variables_interest_sva <- DataFrame(variables_interest, res$sv)
```

Perform DE analysis with voom:

```
# Normalize data with voom and add tumor stage as covariate in design
design <- model.matrix(~Group + Sex + Age + Dwalk + FEV1PSPC + fume_expose + dusty_expose +
    history_asthma + Cough + BMI + Cr_wheezeng + clust1 + clust3 + clust5 + clust6 +
    clust7 + clust8 + clust10 + clust11 + clust12 + V1 + V2 + V3 + V4 + V5 + V6 +
    V7 + V8 + V9 + V10 + V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18, data = variables_interest_sva)
v <- voom(countData, design = design, plot = TRUE)
```




```
fit <- lmFit(v, design)
fit <- eBayes(fit)
```

```
summa.fit <- decideTests(fit)
summary(summa.fit)
```

```
##        (Intercept) GroupSevere  SexM   Age Dwalk FEV1PSPC fume_exposeY
## Down           427           0    44     0     0        0           0
## NotSig       23676       48525 48450 48525 48525    48525       48525
## Up           24422           0    31     0     0        0           0
##        dusty_exposeY history_asthmaY CoughNo chronic cough   BMI Cr_wheezengY
## Down               0               0                    0     0           0
## NotSig         48525           48525                48525 48524       48525
## Up                 0               0                    0     1           0
##        clust1 clust3 clust5 clust6 clust7 clust8 clust10 clust11 clust12    V1
## Down        0      0      0      0      0      0       0       0       0 18307
## NotSig  48525  48525  48525  48525  48525  48525   48525   48524   48524 13236
## Up          0      0      0      0      0      0       0       1       1 16982
##           V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V12   V13
```

```
## Down   15909 13411 12384  9502  9900  6144  6941  7473  7506  7658  5206  4697
## NotSig 20504 22921 25173 28507 30245 34622 34145 33812 33017 34622 37244 37810
## Up     12112 12193 10968 10516  8380  7759  7439  7240  8002  6245  6075  6018
##           V14   V15   V16   V17   V18
## Down     5248  3660  4104  2381  2619
## NotSig 38174 40714 40216 42860 43875
## Up      5103  4151  4205  3284  2031
```