# aryl_hydrocarbon

Isabella Hofstede

2/10/2021

### loading packages

```
## Bioconductor version '3.11' is out-of-date; the current release version '3.14'
##   is available with R version '4.1'; see https://bioconductor.org/install
```

```
## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
##     CRAN: https://cloud.r-project.org
```

```
## Bioconductor version 3.11 (BiocManager 1.30.16), R 4.0.4 (2021-02-15)
```

```
## Warning: package(s) not installed when version(s) same as current; use `force = TRUE` to
##   re-install: 'edgeR'
```

```
## Installation paths not writeable, unable to update packages
##   path: /opt/GnuR
##   packages:
##     BBmisc, BH, BatchJobs, BayesFactor, Brobdingnag, CVST, Cairo, DBI, DBItest,
##     DEoptimR, DT, GLMMadaptive, HSAUR3, Hmisc, IRdisplay, Kendall, MALDIquant,
##     MASS, MCMCglmm, Matrix, MuMIn, R.utils, RCurl, RJSONIO, RMariaDB, RMySQL,
##     RODBC, RPostgreSQL, RPostgres, RSQLite, RSiena, RandomFields,
##     RandomFieldsUtils, Rcmdr, RcmdrMisc, Rcpp, RcppArmadillo, RcppCCTZ,
##     RcppParallel, Rcsdp, Rdpack, RhpcBLASctl, Rmpfr, TH.data, TMB, TSP, TTR,
##     V8, VGAM, XML, ade4, adegraphics, akima, alphahull, animation, ape,
##     aplpack, argparse, arm, ashr, av, backports, basefun, bayesplot,
##     bayestestR, bench, bio3d, bookdown, brew, brio, brms, broom, bslib,
##     candisc, car, carData, cli, clipr, clubSandwich, coin, colorspace,
##     colourpicker, commonmark, coneproj, conquer, corpcor, cpp11, crayon,
##     credentials, crosstalk, crul, cubature, data.table, datawizard, deSolve,
##     deldir, dendextend, desc, devtools, diffobj, digest, distributional, doMC,
##     doParallel, downlit, dplyr, dtplyr, e1071, effects, effectsize, emmeans,
##     evaluate, evd, exact2x2, exactextractr, fansi, fastICA, fda, ff, filehash,
##     flextable, foreach, fs, future, gam, gamlss, gamlss.data, gamlss.dist,
##     gdtools, gee, geepack, generics, geojsonsf, geosphere, gert, git2r, glmnet,
##     glue, gmailr, gmp, goftest, googleVis, gower, gsl, gstat, heplots, hms,
##     htmlTable, htmlwidgets, httpuv, igraph, influenceR, insight, ipred, irlba,
##     iterators, jose, jqr, jsonlite, jsonvalidate, knitr, ks, lavaan, leaflet,
##     lfe, libcoin, lifecycle, linprog, lme4, lmtest, loo, lubridate, lwgeom,
##     magic, magrittr, manipulateWidget, mapproj, maps, maptools, mathjaxr,
##     matrixStats, mclust, memoise, mice, microbenchmark, mime, misc3d, mlt,
##     mockery, modelbased, moonBook, msm, multcomp, mvtnorm, nanotime, ncdf4,
##     nlme, nloptr, odbc, officer, openssl, openxlsx, optimx, optparse, osmdata,
```

```
##     pander, parallelly, parameters, parsedate, pbapply, pder, pdftools,
##     performance, perm, permute, pillar, pkgbuild, pkgdown, pkgload, plm,
##     plotly, polycor, posterior, pracma, progressr, psych, psychTools,
##     psychotools, qtl, quantreg, rJava, ragg, randtoolbox, raster, rasterVis,
##     rbibutils, rcmdcheck, readr, recipes, remotes, repr, reticulate, rex,
##     rgdal, rgeos, rgl, rio, rjson, rlang, rmarkdown, rncl, rngWELL, rngtools,
##     robust, robustbase, rootSolve, rrcov, rsconnect, rsm, rstan, rsvg, rtdists,
##     rugarch, rvest, s2, sass, satellite, scico, see, segmented, sem, seriation,
##     servr, sessioninfo, sets, sf, sfsmisc, shiny, shinyjs, shinystan,
##     shinytest, showtext, sjmisc, skewt, slackr, slam, sm, snow, sp, spData,
##     spacetime, spam, spatstat, spatstat.core, spatstat.data, spatstat.geom,
##     spatstat.linnet, spatstat.sparse, spatstat.utils, spdep, speedglm,
##     splines2, stargazer, stars, stringdist, stringi, styler, subplex,
##     survPresmooth, svglite, sysfonts, systemfonts, terra, tesseract, testthat,
##     texreg, textshaping, tibble, tidyr, tidyselect, tinytex, tis, tkrplot,
##     tmap, tram, trtf, tseries, tufte, tzdb, udunits2, units, unix, usethis,
##     uuid, vars, vcd, vdiffr, viridis, vroom, waldo, webp, withr, wk, xfun,
##     xml2, yaml, ztable
##   path: /usr/lib/R/library
##   packages:
##     KernSmooth, boot, class, cluster, foreign, lattice, mgcv, nnet, rpart,
##     spatial, survival
##   path: /usr/lib/R/site-library
##   packages:
##     littler
```

##loading data

```
my_data <- read.table('./GSE47944_ps_tt_gel_raw.txt', header = TRUE, row.names = 1)
```

##summary

```
summary(my_data)
```

```
##  Sample_PDM_K1_1    Sample_PDM_K1_2    Sample_PDM_K1_3    Sample_PDM_K1_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0
##  Median :     2.0   Median :     1.0   Median :     2.0   Median :     1
##  Mean   :   490.2   Mean   :   427.6   Mean   :   335.1   Mean   :   349
##  3rd Qu.:   237.0   3rd Qu.:   152.0   3rd Qu.:   118.0   3rd Qu.:   120
##  Max.   :243455.0   Max.   :123795.0   Max.   :136972.0   Max.   :108562
##  Sample_PDM_K1_5    Sample_PDM_K1_6    Sample_PDM_K1_7    Sample_PDM_K1_8
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0
##  Median :     2.0   Median :     1.0   Median :     1.0   Median :     2
##  Mean   :   521.9   Mean   :   491.3   Mean   :   447.8   Mean   :   528
##  3rd Qu.:   248.0   3rd Qu.:   181.0   3rd Qu.:   163.0   3rd Qu.:   216
##  Max.   :591016.0   Max.   :150175.0   Max.   :153667.0   Max.   :188082
##  Sample_PDM_K2_1  Sample_PDM_K2_2  Sample_PDM_K2_3    Sample_PDM_K2_4
##  Min.   :     0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0   1st Qu.:     0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     1   Median :     2   Median :     2.0   Median :     2.0
##  Mean   :   324   Mean   :   539   Mean   :   530.2   Mean   :   502.1
##  3rd Qu.:   146   3rd Qu.:   204   3rd Qu.:   202.0   3rd Qu.:   176.0
##  Max.   :337013   Max.   :180125   Max.   :314830.0   Max.   :186504.0
##  Sample_PDM_K2_5    Sample_PDM_K2_6    Sample_PDM_K2_7    Sample_PDM_K2_8
```

```
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :    0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:    0.0
##  Median :     3.0   Median :     2.0   Median :     2.0   Median :    2.0
##  Mean   :   445.9   Mean   :   447.8   Mean   :   416.1   Mean   :  360.8
##  3rd Qu.:   236.0   3rd Qu.:   188.0   3rd Qu.:   174.0   3rd Qu.:  188.0
##  Max.   :396182.0   Max.   :165907.0   Max.   :152862.0   Max.   :73993.0
##  Sample_PDM_K3_1    Sample_PDM_K3_2    Sample_PDM_K3_3    Sample_PDM_K3_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     1.0   Median :     2.0   Median :     2.0   Median :     1.0
##  Mean   :   355.3   Mean   :   409.4   Mean   :   396.4   Mean   :   425.8
##  3rd Qu.:   168.8   3rd Qu.:   148.0   3rd Qu.:   159.0   3rd Qu.:   150.0
##  Max.   :347263.0   Max.   :145586.0   Max.   :129595.0   Max.   :144429.0
##  Sample_PDM_K3_5    Sample_PDM_K3_6  Sample_PDM_K3_7    Sample_PDM_K3_8
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     1.0   Median :     1   Median :     1.0   Median :     1.0
##  Mean   :   320.5   Mean   :   464   Mean   :   392.2   Mean   :   475.9
##  3rd Qu.:   146.0   3rd Qu.:   183   3rd Qu.:   160.0   3rd Qu.:   196.0
##  Max.   :499095.0   Max.   :295604   Max.   :206067.0   Max.   :371514.0
##   Sample_K4.1        Sample_K4.2        Sample_K4.3      Sample_K4.4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0   1st Qu.:     0.0
##  Median :     4.0   Median :     2.0   Median :     4   Median :     3.0
##  Mean   :   693.7   Mean   :   664.8   Mean   :  1000   Mean   :   712.6
##  3rd Qu.:   357.0   3rd Qu.:   256.8   3rd Qu.:   319   3rd Qu.:   345.0
##  Max.   :303103.0   Max.   :181111.0   Max.   :354277   Max.   :137348.0
##   Sample_K4.5        Sample_K4.6      Sample_K4.7        Sample_K4.8
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     5.0   Median :     5   Median :     3.0   Median :     4.0
##  Mean   :   744.8   Mean   :  1135   Mean   :   826.9   Mean   :   984.7
##  3rd Qu.:   401.0   3rd Qu.:   517   3rd Qu.:   372.0   3rd Qu.:   455.0
##  Max.   :714692.0   Max.   :359225   Max.   :265554.0   Max.   :304043.0
##   Sample_K5.1        Sample_K5.2        Sample_K5.3        Sample_K5.4
##  Min.   :      0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:      0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :      5.0   Median :     3.0   Median :     3.0   Median :     3.0
##  Mean   :    999.3   Mean   :   763.5   Mean   :   580.9   Mean   :   814.4
##  3rd Qu.:    490.0   3rd Qu.:   250.0   3rd Qu.:   240.0   3rd Qu.:   311.0
##  Max.   :1467069.0   Max.   :371382.0   Max.   :236320.0   Max.   :286124.0
##   Sample_K5_5        Sample_K5_6        Sample_K5_7        Sample_K5_8
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     2.0   Median :     1.0   Median :     2.0   Median :     1.0
##  Mean   :   591.3   Mean   :   462.6   Mean   :   772.9   Mean   :   506.2
##  3rd Qu.:   263.0   3rd Qu.:   187.0   3rd Qu.:   302.0   3rd Qu.:   206.0
##  Max.   :722125.0   Max.   :247148.0   Max.   :465601.0   Max.   :183027.0
##   Sample_K6_1        Sample_K6_2        Sample_K6_3        Sample_K6_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     2.0   Median :     2.0   Median :     1.0   Median :     2.0
##  Mean   :   568.8   Mean   :   564.6   Mean   :   435.1   Mean   :   483.1
##  3rd Qu.:   210.0   3rd Qu.:   182.0   3rd Qu.:   152.0   3rd Qu.:   166.0
```

```
##  Max.   :888166.0   Max.   :187337.0   Max.   :390415.0   Max.   :296735.0
##   Sample_K6_5        Sample_K6_6        Sample_K6_7        Sample_K6_8
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     2.0   Median :     3.0   Median :     2.0   Median :     2.0
##  Mean   :   579.2   Mean   :   888.6   Mean   :   445.8   Mean   :   500.8
##  3rd Qu.:   257.0   3rd Qu.:   379.0   3rd Qu.:   201.0   3rd Qu.:   219.0
##  Max.   :826889.0   Max.   :267107.0   Max.   :121829.0   Max.   :220719.0
##   Sample_K8_1        Sample_K8_2        Sample_K8_3        Sample_K8_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     1.0   Median :     3.0   Median :     2.0   Median :     2.0
##  Mean   :   337.4   Mean   :   807.9   Mean   :   570.2   Mean   :   598.4
##  3rd Qu.:   132.0   3rd Qu.:   277.0   3rd Qu.:   155.0   3rd Qu.:   164.0
##  Max.   :484824.0   Max.   :338668.0   Max.   :293372.0   Max.   :292273.0
##   Sample_K8_5        Sample_K8_6        Sample_K8_7        Sample_K8_8
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0
##  Median :     3.0   Median :     2.0   Median :     3.0   Median :     2
##  Mean   :   514.4   Mean   :   845.8   Mean   :   923.9   Mean   :   403
##  3rd Qu.:   248.0   3rd Qu.:   318.0   3rd Qu.:   365.8   3rd Qu.:   176
##  Max.   :641604.0   Max.   :404971.0   Max.   :327274.0   Max.   :113009
##   Sample_K9_1       Sample_K9_2       Sample_K9_3        Sample_K9_4
##  Min.   :     0    Min.   :     0    Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0    1st Qu.:     0    1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     3    Median :     2    Median :     3.0   Median :     2.0
##  Mean   :   659    Mean   :   560    Mean   :   616.6   Mean   :   648.8
##  3rd Qu.:   322    3rd Qu.:   216    3rd Qu.:   212.0   3rd Qu.:   212.0
##  Max.   :663837    Max.   :233609    Max.   :234700.0   Max.   :238487.0
##   Sample_K9_5        Sample_K9_6        Sample_K9_7        Sample_K9_8
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     3.0   Median :     3.0   Median :     1.0   Median :     2.0
##  Mean   :   598.8   Mean   :   793.9   Mean   :   340.3   Mean   :   610.7
##  3rd Qu.:   276.8   3rd Qu.:   297.0   3rd Qu.:   131.0   3rd Qu.:   222.0
##  Max.   :719420.0   Max.   :286619.0   Max.   :115618.0   Max.   :471901.0
##  Sample_N248_1      Sample_N248_2      Sample_N248_3      Sample_N248_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     2.0   Median :     2.0   Median :     1.0   Median :     2.0
##  Mean   :   441.1   Mean   :   544.1   Mean   :   477.8   Mean   :   805.1
##  3rd Qu.:   202.0   3rd Qu.:   218.0   3rd Qu.:   195.0   3rd Qu.:   330.8
##  Max.   :564081.0   Max.   :477181.0   Max.   :316705.0   Max.   :621046.0
##  Sample_N250_1      Sample_N250_2      Sample_N250_3      Sample_N250_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :    0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:    0.0
##  Median :     2.0   Median :     2.0   Median :     2.0   Median :    1.0
##  Mean   :   335.5   Mean   :   769.9   Mean   :   523.9   Mean   :  216.3
##  3rd Qu.:   168.0   3rd Qu.:   299.0   3rd Qu.:   205.0   3rd Qu.:   76.0
##  Max.   :371949.0   Max.   :229900.0   Max.   :134029.0   Max.   :65612.0
##  Sample_N252_1      Sample_N252_2      Sample_N252_3      SSample_N252_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     2.0   Median :     3.0   Median :     2.0   Median :     3.0
```

```
##  Mean   :   444.4   Mean   :   863.6   Mean   :   480.4   Mean   :   780.6
##  3rd Qu.:   227.0   3rd Qu.:   349.0   3rd Qu.:   198.0   3rd Qu.:   361.0
##  Max.   :416633.0   Max.   :221136.0   Max.   :134461.0   Max.   :121592.0
##  Sample_N253_1      Sample_N253_2      Sample_N253_3      Sample_N253_4
##  Min.   :     0.0   Min.   :     0.0   Min.   :    0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:    0.0   1st Qu.:     0.0
##  Median :     2.0   Median :     2.0   Median :    1.0   Median :     1.0
##  Mean   :   457.4   Mean   :   527.5   Mean   :  297.3   Mean   :   363.5
##  3rd Qu.:   198.0   3rd Qu.:   193.0   3rd Qu.:  111.0   3rd Qu.:   102.0
##  Max.   :592796.0   Max.   :192895.0   Max.   :76085.0   Max.   :117744.0
##  Sample_N254_1    Sample_N254_2      Sample_N254_3      Sample_N254_4
##  Min.   :     0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     0   1st Qu.:     0.0   1st Qu.:     0.0   1st Qu.:     0.0
##  Median :     3   Median :     1.0   Median :     1.0   Median :     2.0
##  Mean   :   712   Mean   :   554.5   Mean   :   407.9   Mean   :   473.6
##  3rd Qu.:   338   3rd Qu.:   190.0   3rd Qu.:   157.0   3rd Qu.:   192.0
##  Max.   :930461   Max.   :227714.0   Max.   :189010.0   Max.   :321427.0
```

```
# de data sample naamgeving is opgebouwd uit samplenaam_patientnummer.
# Dus het getal op het eind geeft aan bij welke patient het sample hoort
```

##Data preperation

```
# 64 sick samples
sick_data <- my_data[1:64]
#20 healthy samples
healthy_data <- my_data[65: 84]

library(GEOquery)
```

```
## Loading required package: Biobase
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     Filter, Find, Map, Position, Reduce, anyDuplicated, append,
##     as.data.frame, basename, cbind, colnames, dirname, do.call,
##     duplicated, eval, evalq, get, grep, grepl, intersect, is.unsorted,
##     lapply, mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##     pmin.int, rank, rbind, rownames, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min
```

```
## Welcome to Bioconductor
```

```
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Setting options('download.file.method.GEOquery'='auto')

## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

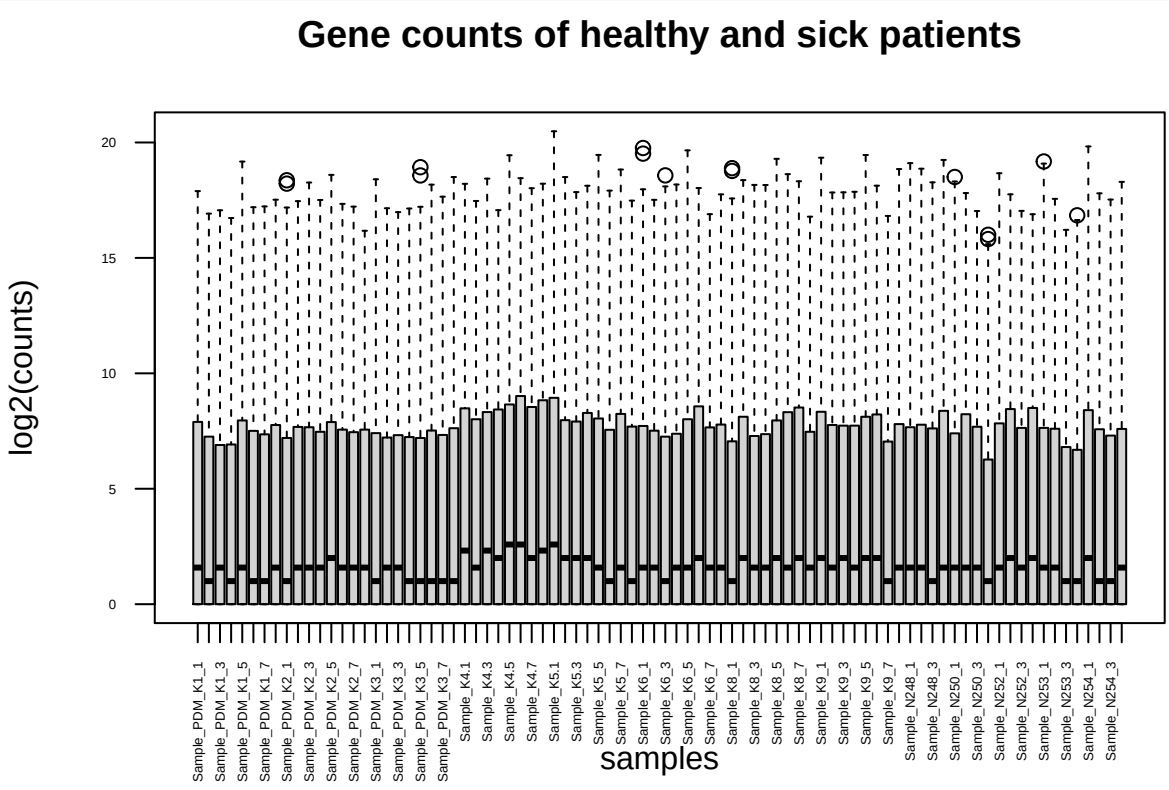
```
classification <- getGEO(filename="GSE47944_series_matrix.txt")
```

```
## Rows: 0 Columns: 85

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (85): ID_REF, GSM1162982, GSM1162983, GSM1162984, GSM1162985, GSM1162986...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## File stored at:

## /tmp/RtmpZyWZa5/GPL11154.soft
```

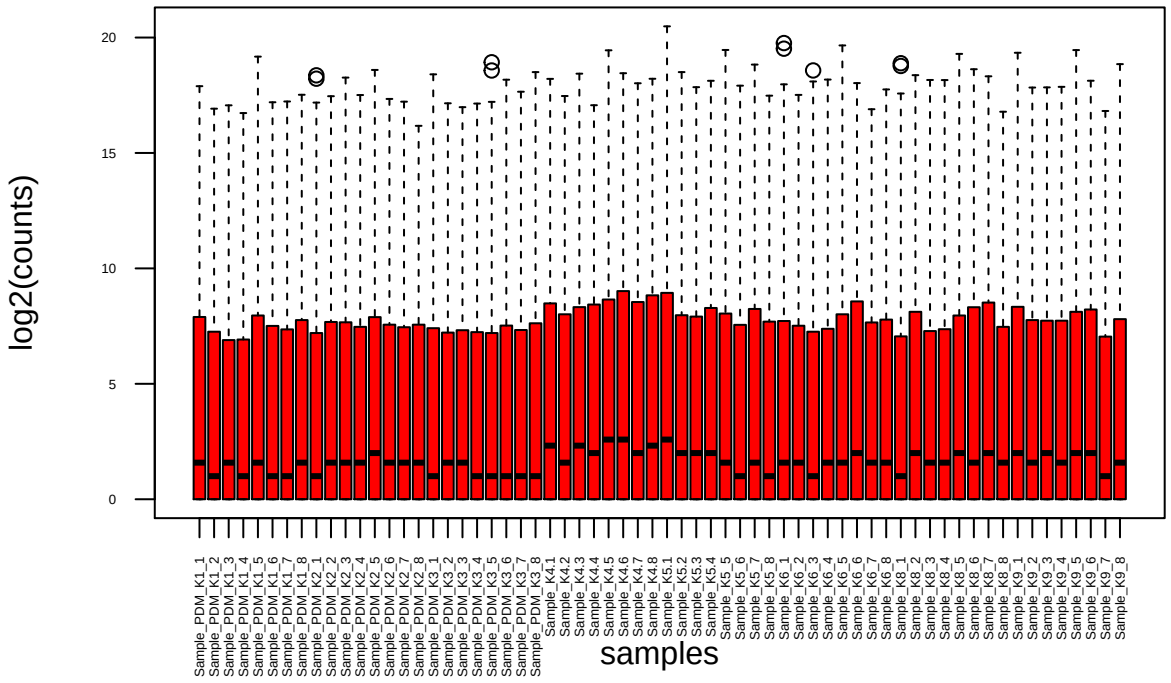
```
phenodata <- classification@phenoData@data
```

##Boxplot Analyses

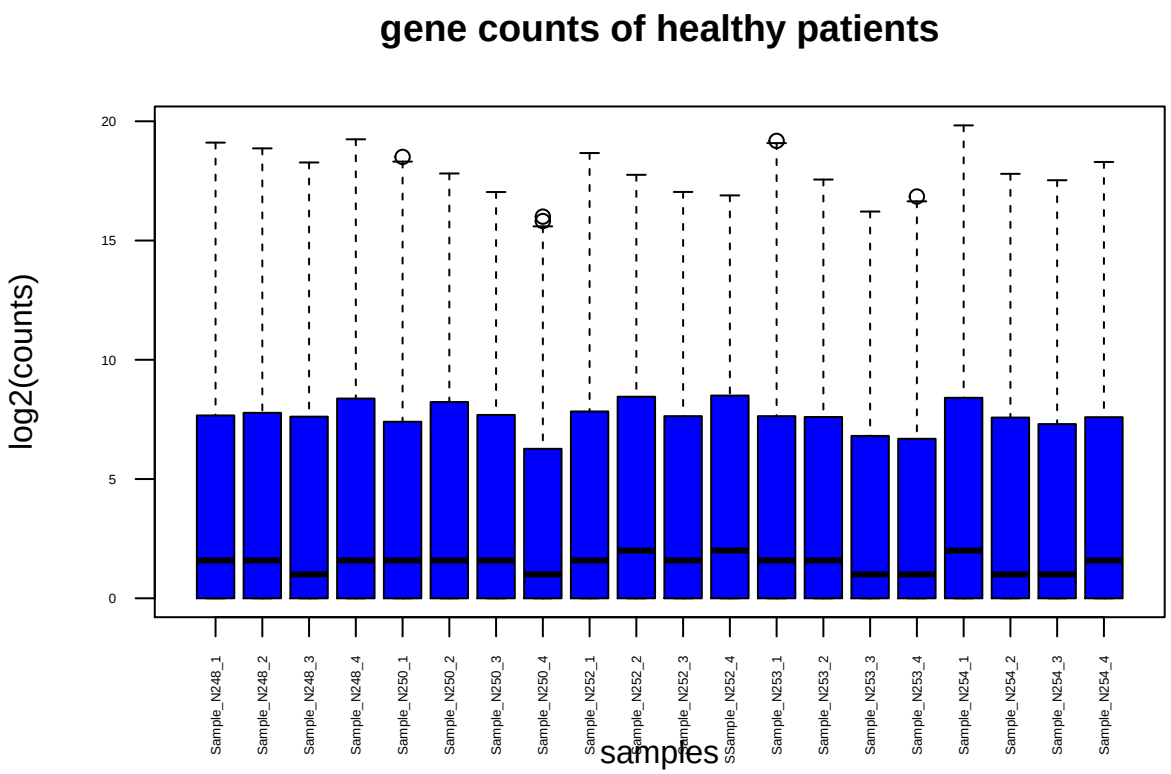
```
boxplot(log2(my_data + 1), data=my_data, xlab="samples", ylab="log2(counts)",
    main="Gene counts of healthy and sick patients", las=2, par(cex.axis=0.45))
```




```
boxplot(log2(sick_data + 1), data=my_data, xlab="samples", ylab="log2(counts)",
    main="gene counts of sick patients", col="red", las=2)
```




```
boxplot(log2(healthy_data + 1), data=my_data, xlab="samples", ylab="log2(counts)",
    main="gene counts of healthy patients", col="blue", las=2)
```

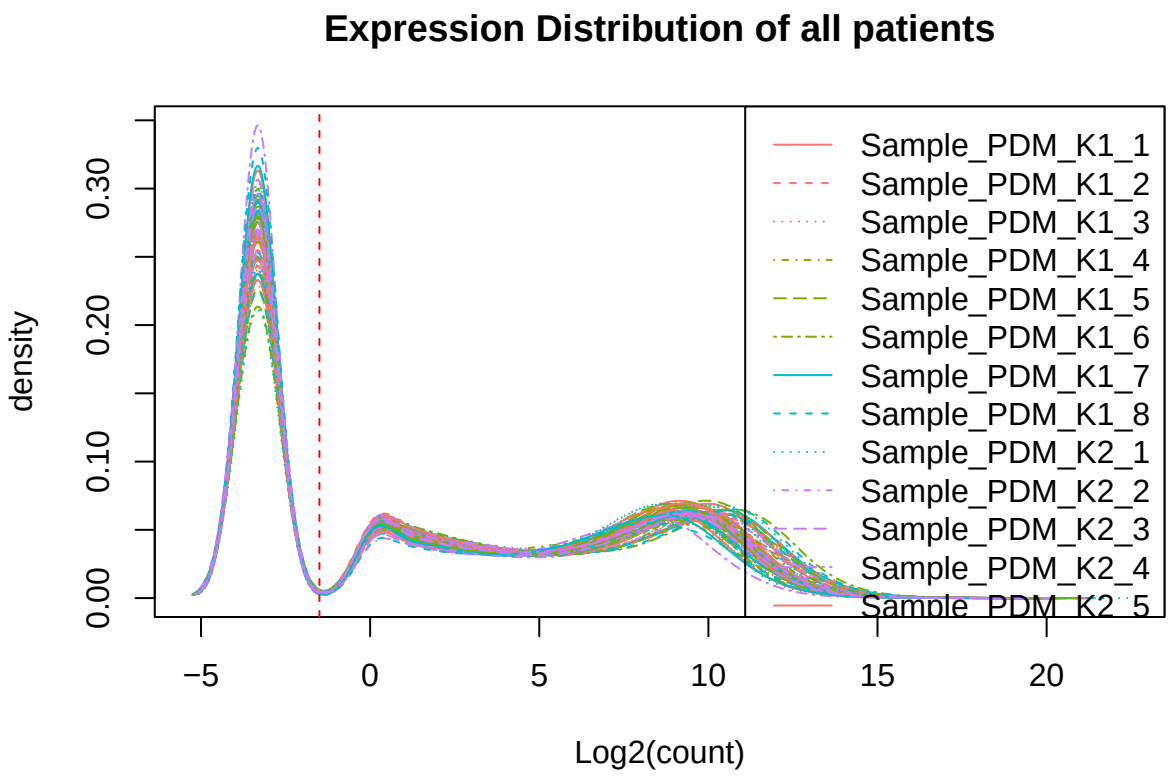



##density plot analysis

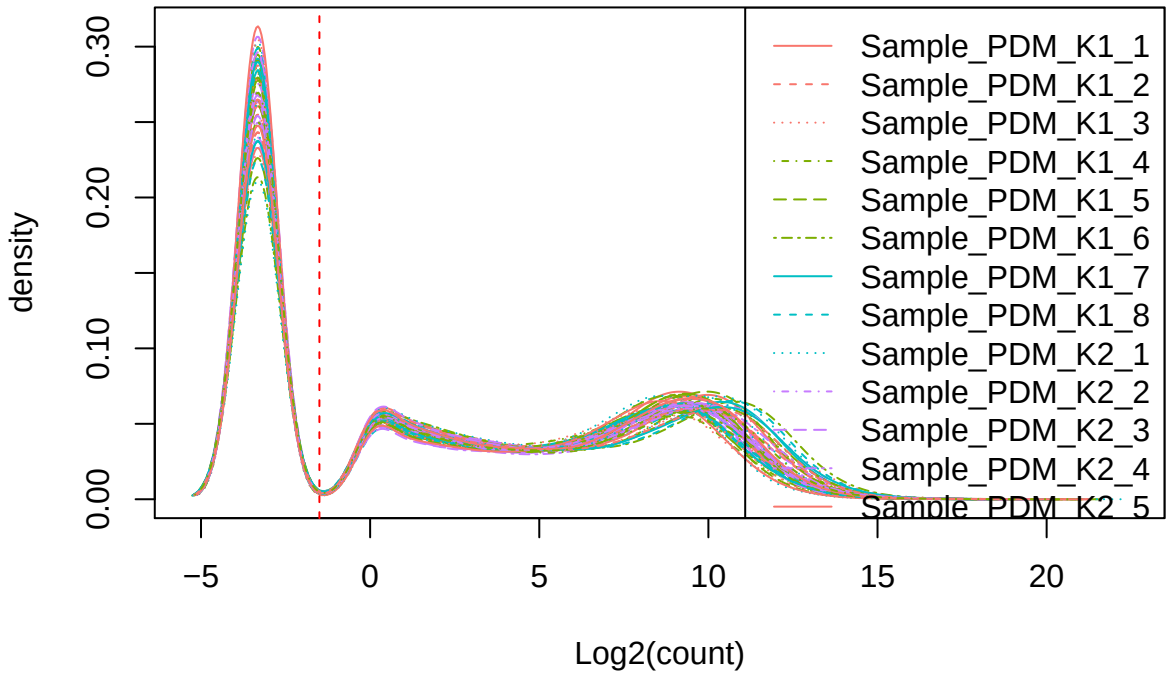
```
## The affy library has a density plotting function
library(affy)

## Create a list of 4 colors to use which are the same used throughout
#this chapter
library(scales)
myColors <- hue_pal()(4)
```

##density plot of all sick patients

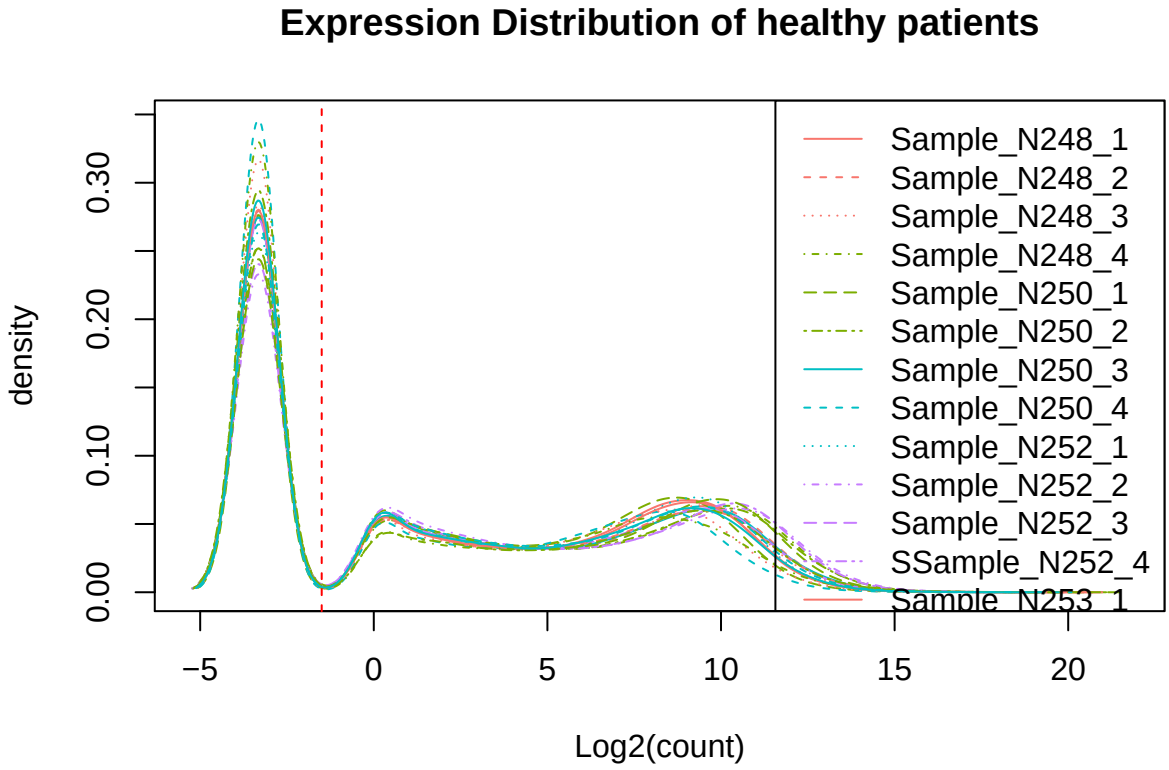
```
## Plot the log2-transformed data with a 0.1 pseudocount of the whole dataset
plotDensity(log2(my_data + 0.1), col=rep(myColors, each=3),
            lty=c(1:ncol(my_data)), xlab='Log2(count)',
            main='Expression Distribution of all patients')

## Add a legend and vertical line
legend('topright', names(my_data), lty=c(1:ncol(my_data)),
       col=rep(myColors, each=3))
abline(v=-1.5, lwd=1, col='red', lty=2)
```




```
## Plot the log2-transformed data with a 0.1 pseudocount of the sick patients
plotDensity(log2(sick_data + 0.1), col=rep(myColors, each=3),
            lty=c(1:ncol(sick_data)), xlab='Log2(count)',
            main='Expression Distribution of sick patients')

## Add a legend and vertical line
legend('topright', names(sick_data), lty=c(1:ncol(my_data)),
       col=rep(myColors, each=3))
abline(v=-1.5, lwd=1, col='red', lty=2)
```

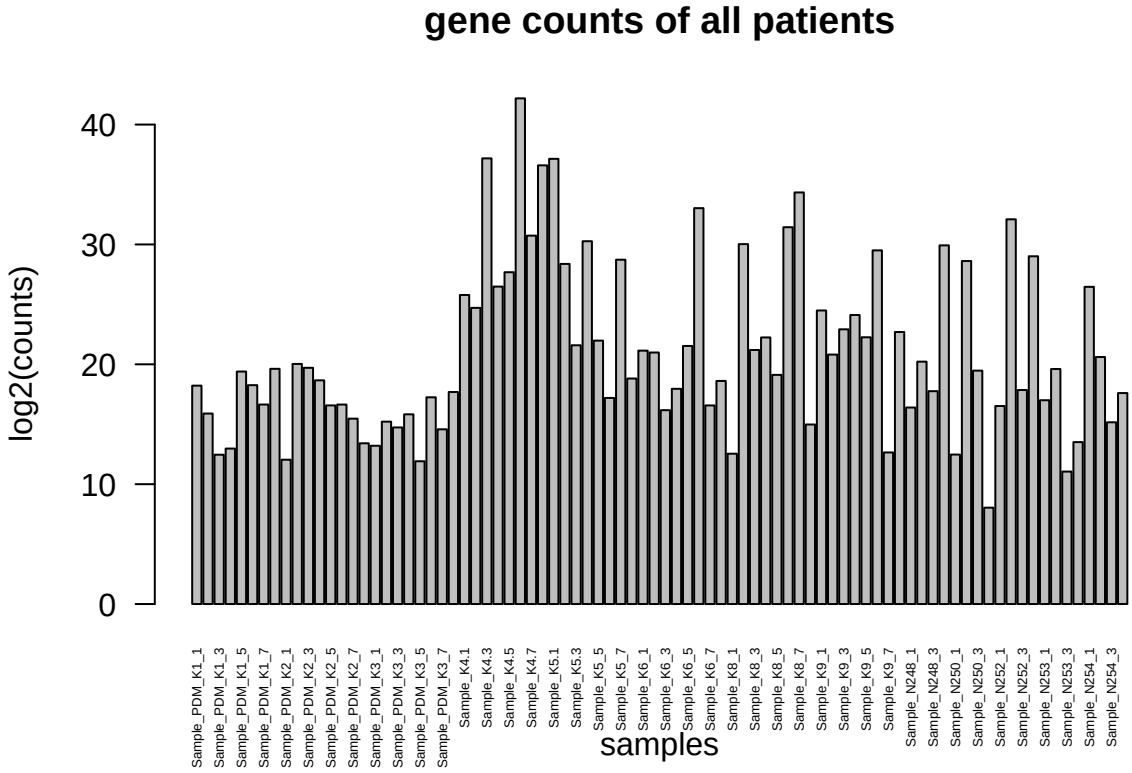



```
## Plot the log2-transformed data with a 0.1 pseudocount of healthy patients
plotDensity(log2(healthy_data + 0.1), col=rep(myColors, each=3),
            lty=c(1:ncol(healthy_data)), xlab='Log2(count)',
            main='Expression Distribution of healthy patients')

## Add a legend and vertical line
legend('topright', names(healthy_data), lty=c(1:ncol(my_data)),
       col=rep(myColors, each=3))
abline(v=-1.5, lwd=1, col='red', lty=2)
```

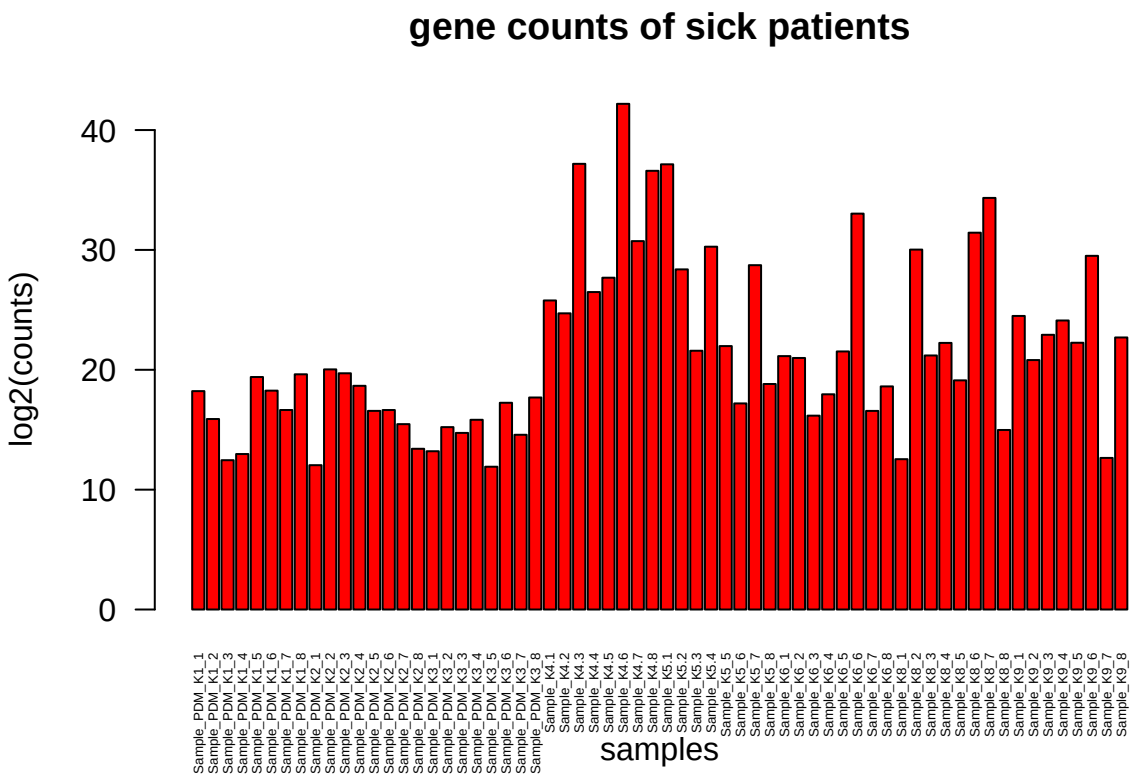



##Barplot analysis

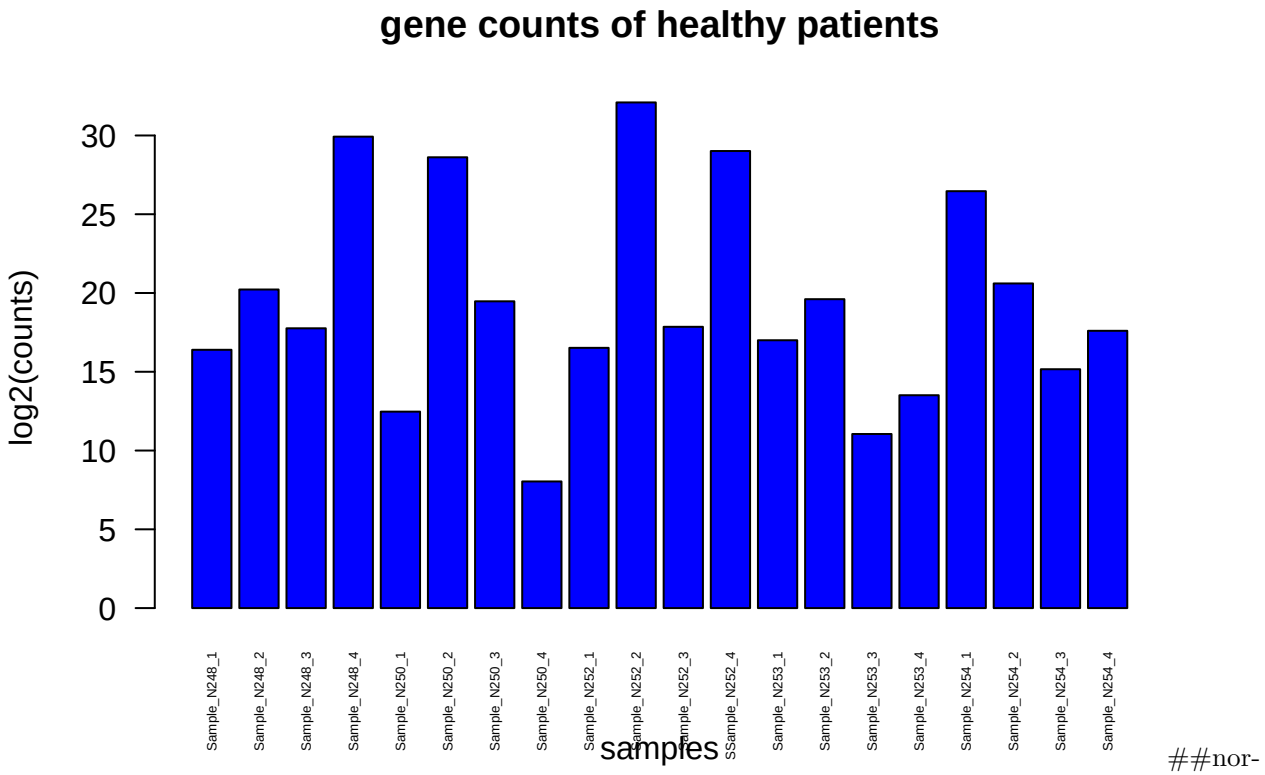
```
barplot(colSums(my_data) / 1e6, xlab="samples", ylab="log2(counts)",
    main="gene counts of all patients", las=2, cex.names= 0.45)
```



```
barplot(colSums(sick_data) / 1e6, xlab="samples", ylab="log2(counts)",
    main="gene counts of sick patients", col="red", las=2, cex.names= 0.45)
```




```
barplot(colSums(healthy_data) / 1e6, xlab="samples", ylab="log2(counts)",
    main="gene counts of healthy patients", col="blue", las=2, cex.names= 0.45)
```




##nor-malisation

```
# Load the library
library('DESeq2')
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:base':
##
##     expand.grid
```

```
## Loading required package: IRanges
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: GenomeInfoDb
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: DelayedArray
```

```
## Loading required package: matrixStats
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
```

```
##
## Attaching package: 'DelayedArray'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
```

```
## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum
```

```
# DESeq2 will construct a SummarizedExperiment object and combine this
# into a 'DESeqDataSet' object. The 'design' argument usually indicates the
# experimental design using the condition(s) names as a 'factor', for now
#we use just '~ 1'
(ddsMat <- DESeqDataSetFromMatrix(countData = my_data,
                                  colData = data.frame(samples = names(my_data))
                                  , design = ~ 1))
```

```
## class: DESeqDataSet
## dim: 37166 84
## metadata(1): version
## assays(1): counts
## rownames(37166): ENSG00000237223|SULT1C2P1 ENSG00000176903|PNMA1 ...
##   ENSG00000261716|RP11-196G18.22 ENSG00000141934|PPAP2C
## rowData names(0):
## colnames(84): Sample_PDM_K1_1 Sample_PDM_K1_2 ... Sample_N254_3
##   Sample_N254_4
## colData names(1): samples
```

##Normalization

```
# Perform normalization
rld.dds <- vst(ddsMat)
# 'Extract' normalized values
rld <- assay(rld.dds)
```

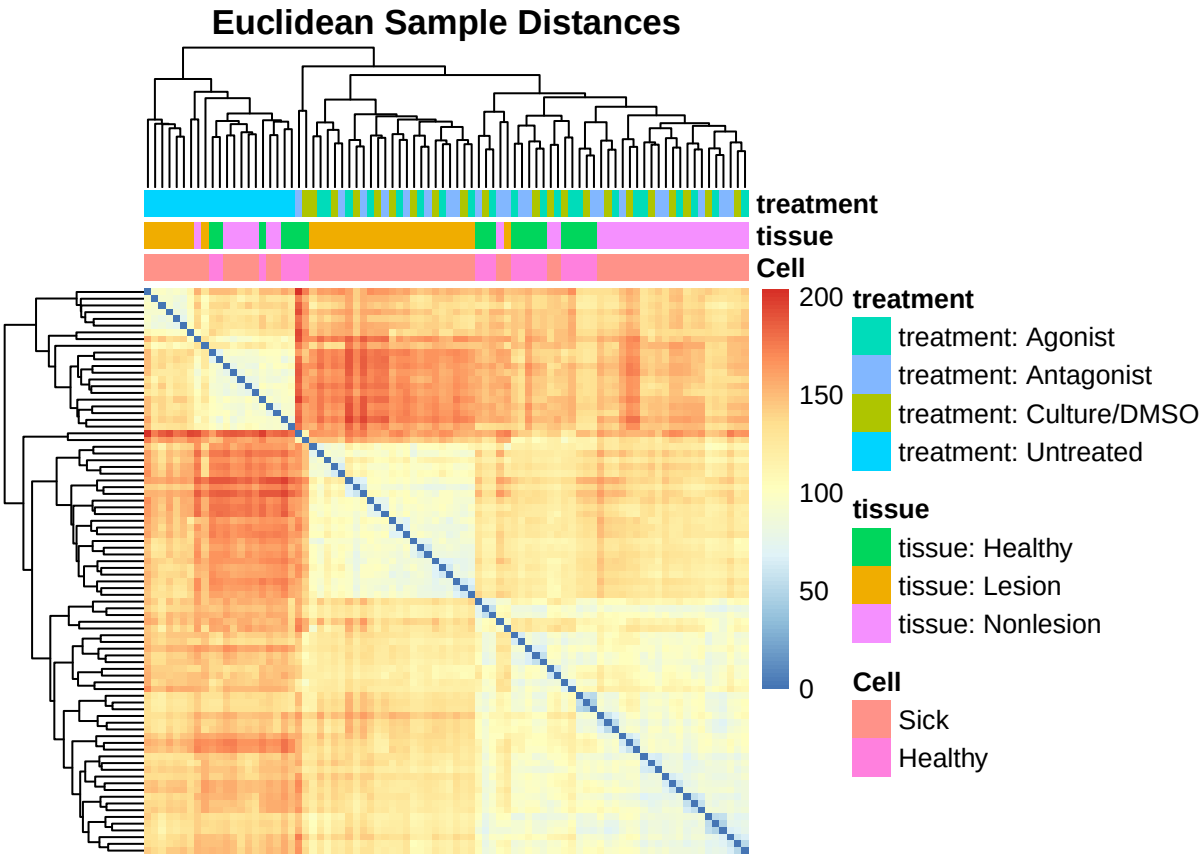
##distance calculation

```
# Calculate basic distance metric (using euclidean distance, see '?dist')
sampledists <- dist( t( rld ))
```

##heatmap

```
# We use the 'pheatmap' library (install with install.packages('pheatmap'))
library(pheatmap)

# Convert the 'dist' object into a matrix for creating a heatmap
sampleDistMatrix <- as.matrix(sampledists)
#The annotation is an extra layer that will be plotted above the heatmap columns
# indicating the cell type
annotation <- data.frame(Cell = factor(c(rep(1, 64), rep(2, 20)),
                                       labels = c("Sick", "Healthy")))

annotation$tissue <- phenodata$characteristics_ch1.5
annotation$treatment <- phenodata$characteristics_ch1.6
#Set the rownames of the annotation dataframe to the sample names (required)
rownames(annotation) <- names(my_data)

pheatmap(sampleDistMatrix, show_colnames = FALSE,
         annotation_col = annotation,
```

```
         clustering_distance_rows = sampledists,
         clustering_distance_cols = sampledists,
         show_rownames = F,
         main = "Euclidean Sample Distances")
```



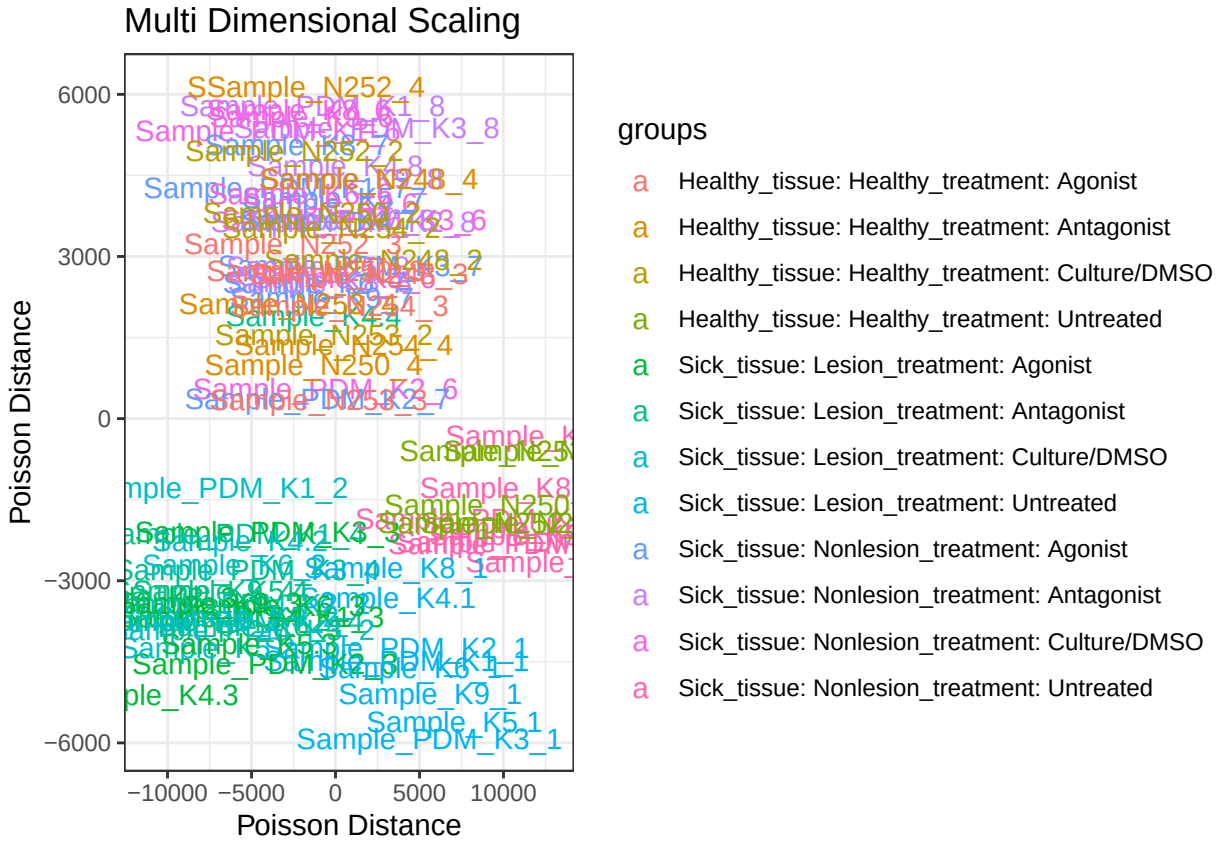

##multi-dimentional scaling

```
library('PoiClaClu')
# Use the raw (not r-log transformed!) counts
dds <- assay(ddsMat)
poisd <- PoissonDistance( t(dds) )
# Extract the matrix with distances
samplePoisDistMatrix <- as.matrix(poisd$dd)
# Calculate the MDS and get the X- and Y-coordinates
mdsPoisData <- data.frame( cmdscale(samplePoisDistMatrix) )

# And set some better readable names for the columns
names(mdsPoisData) <- c('x_coord', 'y_coord')
```

```
library(ggplot2)
# Separate the annotation factor (as the variable name is used as label)
groups <- factor(c(rep(1, 64), rep(2, 20)),
                                    labels = c("Sick", "Healthy"))
groups <- as.factor(paste(annotation$Cell, phenodata$characteristics_ch1.5, phenodata$characteristics_ch
coldata <- names(my_data)

levels(groups)
```

```
##  [1] "Healthy_tissue: Healthy_treatment: Agonist"
##  [2] "Healthy_tissue: Healthy_treatment: Antagonist"
##  [3] "Healthy_tissue: Healthy_treatment: Culture/DMSO"
##  [4] "Healthy_tissue: Healthy_treatment: Untreated"
##  [5] "Sick_tissue: Lesion_treatment: Agonist"
##  [6] "Sick_tissue: Lesion_treatment: Antagonist"
##  [7] "Sick_tissue: Lesion_treatment: Culture/DMSO"
##  [8] "Sick_tissue: Lesion_treatment: Untreated"
##  [9] "Sick_tissue: Nonlesion_treatment: Agonist"
## [10] "Sick_tissue: Nonlesion_treatment: Antagonist"
## [11] "Sick_tissue: Nonlesion_treatment: Culture/DMSO"
## [12] "Sick_tissue: Nonlesion_treatment: Untreated"
```

```
# Create the plot using ggplot
ggplot(mdsPoisData, aes(x_coord, y_coord, color = groups, label = coldata)) +
  geom_text(size = 4) +
  ggtitle('Multi Dimensional Scaling') +
  labs(x = "Poisson Distance", y = "Poisson Distance") +
  theme_bw()
```




There is a clear clustering of multiple samples.For further analysis the groups that will be compared are the different types of treatment from one type of tissue. That means antagonist vs agonist treatment type of healthy tissue.

## chapter 4 pre-processing data

```
# Perform a naive FPM normalization
# Note: log transformation includes a pseudocount of 1
```

```
my_data.fpm <- log2( (my_data/ (colSums(my_data) / 1e6)) + 1 )

#removing the low counts
library(edgeR)
```

```
## Loading required package: limma
```

```
##
## Attaching package: 'limma'
```

```
## The following object is masked from 'package:DESeq2':
##
##     plotMA
```

```
## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
```

```
keep.exprs <- filterByExpr(my_data.fpm, min.count=10)
```

```
## No group or design set. Assuming all samples belong to one group.
```

```
filt1 <- my_data.fpm[keep.exprs,]
dim(filt1)
```

```
## [1] 31 84
```

### The fold change value

### Links

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE47944