# Descriptive analysis

Sleiman Bassim, PhD

July 23, 2018

## Contents

Loaded functions.

↰ Project started Dec 10 2017, updated July 23, 2018

```
#source("/media/Data/Dropbox/humanR/01funcs.R")
rm(list=ls())
#setwd("/media/Data/Dropbox/humanR/PD/")
#setwd("~/Dropbox/humanR/PD/")
###load("PD.Rdata", .GlobalEnv)
#lsos(pat="")
```

Loaded packages.

```
pkgs <- c('gdata','lattice','latticeExtra',
          'ggplot2', 'dplyr', 'tidyr', 'RColorBrewer','igraph',
          'DescTools', 'scales', 'brotools', 'Hmisc', 'finalfit',
          'plyr', 'paletteer')
lapply(pkgs, require, character.only = TRUE)
```

# 1 Exploratory Data Analysis

Data is from patients with Lymphoma tumors, either undergone or not a Rituximab CHOP treatment. Some patients show relapse after treatment. Tumors migrate though nodal (lymphnodes) or extranodal tissues. Tumors involve two different subtypes of cells of origin, ABC or GCB. **The first aim is to find correlation genes that respond differently to treatment, nodal transmission, and cell subtypes.**

↰OR: Odds ratio. HR: Hazard ratio

```
#read.table("data/phenodata", sep = "\t", header = T) %>%
#    dplyr::select(SAMPLE_ID, Timepoint,
#    GROUP, SITE, Score, Prediction, ABClikelihood) %>%
#    brotools::describe()

print_summary_table <- function(features, dependent, df, execute = TRUE) {
    if ( execute == TRUE ) {
        x <- df %>%
            summary_factorlist(dependent, features, p=FALSE, add_dependent_label=TRUE)
        ## print latex table
        Hmisc::latex(x, file = "", booktabs = TRUE, title = "")
    } else {
        cat("LaTeX summary table printed\n")
    }
}

dfs <- read.table("data/phenodata", sep = "\t", header = T)
print_summary_table(features= c("Score", "ABClikelihood", "GROUP"),
                    dependent= c("Prediction"),
                    df = dfs,
                    execute = F)

LaTeX summary table printed
```

| | Dependent: Prediction | | ABC | GCB | U |
|---|---|---|---|---|---|
| 10 | Score | Mean (SD) | 3156.3 (475.5) | 506.4 (721.1) | 2162.8 (143.6) |
| 1 | ABClikelihood | Mean (SD) | 1 (0) | 0 (0) | 0.5 (0.4) |
| 2 | GROUP | CNS DIAGNOSIS | 4 (33.3) | 6 (50.0) | 2 (16.7) |
| 3 | | CNS RELAPSE CHOP or EQUIVALENT | 6 (60.0) | 3 (30.0) | 1 (10.0) |
| 4 | | CNS RELAPSE RCHOP | 17 (44.7) | 13 (34.2) | 8 (21.1) |
| 5 | | NO RELAPSE | 27 (28.1) | 52 (54.2) | 17 (17.7) |
| 6 | | NORMAL ABC CONTROL | 2 (100.0) | 0 (0.0) | 0 (0.0) |
| 7 | | NORMAL GCB CONTROL | 0 (0.0) | 4 (100.0) | 0 (0.0) |
| 8 | | SYTEMIC RELAPSE NO CNS | 31 (48.4) | 25 (39.1) | 8 (12.5) |
| 9 | | TESTICULAR NO CNS RELAPSE | 9 (75.0) | 0 (0.0) | 3 (25.0) |

## 1.1 Data reformating

In the first steps of the analysis, the samples will be labeled (supervised) into the following categories (based on patients diagnosis).

```
metadata <- read.table("data/phenodata", sep = "\t", header = T) %>%
```

```r
    dplyr::select(SAMPLE_ID, Timepoint, GROUP, SITE, Score, Prediction, ABClikelihood) %>%
    filter(Timepoint != "T2") %>%
    mutate(Groups = case_when(GROUP %in% c("CNS_RELAPSE_RCHOP",
                                            "CNS_RELAPSE_CHOPorEQUIVALENT",
                                            "CNS_DIAGNOSIS") ~ "CNS",
                              GROUP %in% c("TESTICULAR_NO_CNS_RELAPSE", "NO_RELAPSE") ~ "NOREL",
                              GROUP == "SYTEMIC_RELAPSE_NO_CNS" ~ "SYST",
                              TRUE ~ "CTRL")) %>%
    mutate(ABClassify = case_when(ABClikelihood >= .9 ~ "ABC",
                                  ABClikelihood <= .1 ~ "GCB",
                                  TRUE ~ "U")) %>%
    mutate(ABCScore = case_when(Score > 2412 ~ "ABC",
                                Score <= 1900 ~ "GCB",
#                                Score == NA ~ "NA",
                                TRUE ~ "U")) %>%
    mutate(Nodes = case_when(SITE == "LN" ~ "LN",
                             SITE == "TO" ~ "LN",
                             SITE == "SP" ~ "LN",
                             TRUE ~ "EN")) %>%
    mutate(Lymphnodes = case_when(Nodes == "LN" ~ 1, TRUE ~ 0))

# make sure all samples preserve their ID
metadata$Groups <- as.factor(metadata$Groups)
metadata$ABClassify <- as.factor(metadata$ABClassify)
metadata$ABCScore <- as.factor(metadata$ABCScore)
metadata$Nodes <- as.factor(metadata$Nodes)
metadata$Lymphnodes <- as.factor(metadata$Lymphnodes)
#brotools::describe(metadata)
print_summary_table(c("ABCScore", "ABClassify", "GROUP"), c("Nodes"), metadata, execute = F)

LaTeX summary table printed
```

| | Dependent: Nodes | | EN | LN |
|---|---|---|---|---|
| 4 | ABCScore | ABC | 34 (37.0) | 58 (63.0) |
| 5 | | GCB | 36 (35.0) | 67 (65.0) |
| 6 | | U | 16 (39.0) | 25 (61.0) |
| 1 | ABClassify | ABC | 37 (35.9) | 66 (64.1) |
| 2 | | GCB | 38 (32.5) | 79 (67.5) |
| 3 | | U | 11 (68.8) | 5 (31.2) |
| 7 | GROUP | CNS DIAGNOSIS | 7 (63.6) | 4 (36.4) |
| 8 | | CNS RELAPSE CHOP or EQUIVALENT | 5 (62.5) | 3 (37.5) |
| 9 | | CNS RELAPSE RCHOP | 20 (51.3) | 19 (48.7) |
| 10 | | NO RELAPSE | 30 (31.2) | 66 (68.8) |
| 11 | | NORMAL ABC CONTROL | 2 (NA) | 0 (0.0) |
| 12 | | NORMAL GCB CONTROL | 0 (0.0) | 4 (100.0) |
| 13 | | SYTEMIC RELAPSE NO CNS | 10 (15.6) | 54 (84.4) |
| 14 | | TESTICULAR NO CNS RELAPSE | 12 (100.0) | 0 (0.0) |

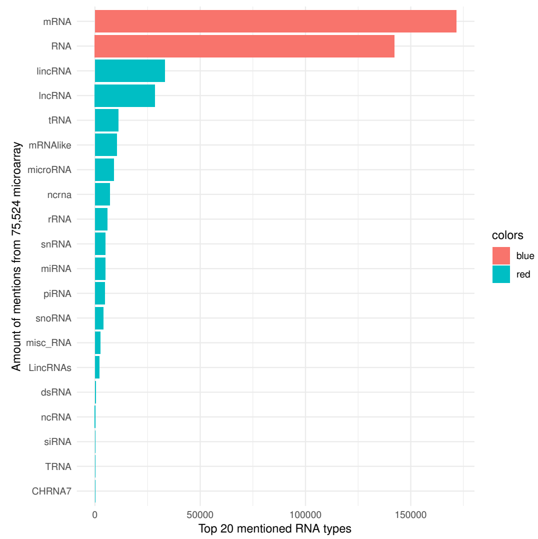
### 1.1.1 On-array content of non-coding RNA

One microarray contains 70,524 probes, of which 12,969 genes do not contain any of the terms related to non-coding RNAs. The graph shows in red that the coding genes are categorized as mRNA or RNA. The non-coding RNAs are in blue and there is over 600 different mentions among the probes (large intergenic non-coding RNAs, lincRNAs make up most of the long ncRNAs). Even though they are less *mentioned* in gene annotations (smaller bars) they do however cover 70% of the annotated probes.

↰ Two or more mentions (patterns) can be recognized in one gene

```r
colors <- c(rep("blue",2), rep("red",18))
```

```r
read.table("./data/rna.patterns.annotated.array.txt", header = F) %>%
    mutate(percent = (V1/5 / (sum(V1)/5)) * 100) %>%
    slice(1:20) %>%
    ggplot(aes(x = reorder(V2, percent),
               y = V1,
               fill = colors)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    coord_flip() +
    theme_minimal() +
    labs(x = "Amount of mentions from 75,524 microarray",
         y = "Top 20 mentioned RNA types")
```




### 1.1.2 Regression analyses to quantify diagnosis connections

Logistic regression of binomial factoring between nodal/extranodal diagnosis and patients labels for cell-of-origin classification and CNS relapse or systemic relapse. Regression model summary with odds ratio with 95% confidence interval to quantify how much nodal and extranodal diagnosis is associated with the cell-of-origin ABC or GCB nature in DLBCL patients with CNS, systemic or no relapse.

```r
fit_summary_table <- function(features, dependent, df, method, execute = TRUE) {
    if ( execute == TRUE ) {
        if ( method == "glm" || method == "cox" ) {
            x <- df %>%
                finalfit(dependent, features)
        } else if ( execute == "glmer" ) {
            x <- df %>%
                finalfit(dependent, features,
                         mixed, random_effect)
        }
        ## print latex table
        Hmisc::latex(x, file = "", booktabs = TRUE, title = "")
    } else {
        cat("LaTeX summary table printed\n")
    }
}


fit_summary_table(features= c("ABCScore", "ABClassify", "GROUP"),
                  dependent= c("Nodes"),
                  df = metadata,
                  method = "glm",
                  execute = F)

LaTeX summary table printed
```

Mixed effects multilevel logistic regression model fit to find connections between patients (CNS relapse,

| | Dependent: Nodes | | EN | LN | OR (univariable) | OR (multivariable) |
|---|---|---|---|---|---|---|
| 4 | ABCScore | ABC | 34 (39.5) | 58 (38.7) | - | - |
| 5 | | GCB | 36 (41.9) | 67 (44.7) | 1.09 (0.61-1.96, p=0.771) | 0.44 (0.06-3.23, p=0.408) |
| 6 | | U | 16 (18.6) | 25 (16.7) | 0.92 (0.43-1.97, p=0.820) | 0.96 (0.25-4.74, p=0.952) |
| 1 | ABClassify | ABC | 37 (43.0) | 66 (44.0) | - | - |
| 2 | | GCB | 38 (44.2) | 79 (52.7) | 1.17 (0.67-2.04, p=0.591) | 1.61 (0.24-10.98, p=0.615) |
| 3 | | U | 11 (12.8) | 5 (3.3) | 0.25 (0.08-0.76, p=0.018) | 0.52 (0.07-2.97, p=0.473) |
| 7 | GROUP | CNS DIAGNOSIS | 7 (8.1) | 4 (2.7) | - | - |
| 8 | | CNS RELAPSE CHOP or EQUIVALENT | 5 (5.8) | 3 (2.0) | 1.05 (0.15-7.08, p=0.960) | 0.97 (0.13-6.76, p=0.979) |
| 9 | | CNS RELAPSE RCHOP | 20 (23.3) | 19 (12.7) | 1.66 (0.43-7.21, p=0.470) | 1.71 (0.42-7.73, p=0.461) |
| 10 | | NO RELAPSE | 30 (34.9) | 66 (44.0) | 3.85 (1.08-15.64, p=0.042) | 3.40 (0.91-14.2, p=0.074) |
| 11 | | NORMAL ABC CONTROL | 2 (2.3) | 0 (0.0) | 0.00 (NA-NA, p=0.995) | 0.00 (NA-NA, p=0.995) |
| 12 | | NORMAL GCB CONTROL | 0 (0.0) | 4 (2.7) | 74.56 (0.00-NA, p=0.993) | 79.25 (0.00-NA, p=0.993) |
| 13 | | SYTEMIC RELAPSE NO CNS | 10 (11.6) | 54 (36.0) | 9.45 (2.42-NA, p=0.002) | 8.07 (1.98-NA, p=0.004) |
| 14 | | TESTICULAR NO CNS RELAPSE | 12 (14.0) | 0 (0.0) | 0.00 (0.00-NA, p=0.988) | 0.00 (0.00-NA, p=0.988) |

systemic, and no relapse) and cell-of-origin predictions (ABC, GCB likelihoods), while considering nodal and extranodal involvement in the relapse (diagnosed tissue sites with cancer invasion).

```
mixed = c("GROUP")
random_effect = c("SITE")
fit_summary_table(features= c("Prediction", "GROUP"),
                  dependent= c("Nodes"),
                  df = metadata,
                  method = "glmer",
                  execute = F)

LaTeX summary table printed
```

| | Dependent: Nodes | | EN | LN | OR (univariable) | OR (multilevel) |
|---|---|---|---|---|---|---|
| 9 | Prediction | ABC | 34 (40.5) | 58 (38.7) | - | - |
| 10 | | GCB | 36 (42.9) | 67 (44.7) | 1.09 (0.61-1.96, p=0.771) | - |
| 11 | | U | 14 (16.7) | 25 (16.7) | 1.05 (0.48-2.32, p=0.908) | - |
| 1 | GROUP | CNS DIAGNOSIS | 7 (8.1) | 4 (2.7) | - | - |
| 2 | | CNS RELAPSE CHOP or EQUIVALENT | 5 (5.8) | 3 (2.0) | 1.05 (0.15-7.08, p=0.960) | 0.38 (0.00-NA, p=0.989) |
| 3 | | CNS RELAPSE RCHOP | 20 (23.3) | 19 (12.7) | 1.66 (0.43-7.21, p=0.470) | 0.49 (0.00-NA, p=0.988) |
| 4 | | NO RELAPSE | 30 (34.9) | 66 (44.0) | 3.85 (1.08-15.64, p=0.042) | 1.70 (0.00-NA, p=0.989) |
| 5 | | NORMAL ABC CONTROL | 2 (2.3) | 0 (0.0) | 0.00 (NA, p=0.995) | 0.00 (0.00-Inf, p=1.000) |
| 6 | | NORMAL GCB CONTROL | 0 (0.0) | 4 (2.7) | NA (0.00-NA, p=0.993) | 285412.87 (0.00-Inf, p=0.999) |
| 7 | | SYTEMIC RELAPSE NO CNS | 10 (11.6) | 54 (36.0) | 9.45 (2.42-42.22, p=0.002) | 1.76 (0.00-NA, p=0.989) |
| 8 | | TESTICULAR NO CNS RELAPSE | 12 (14.0) | 0 (0.0) | 0.00 (0.00-NA, p=0.988) | 0.00 (0.00-Inf, p=1.000) |

## 1.2 Featured data and groups of sample cases

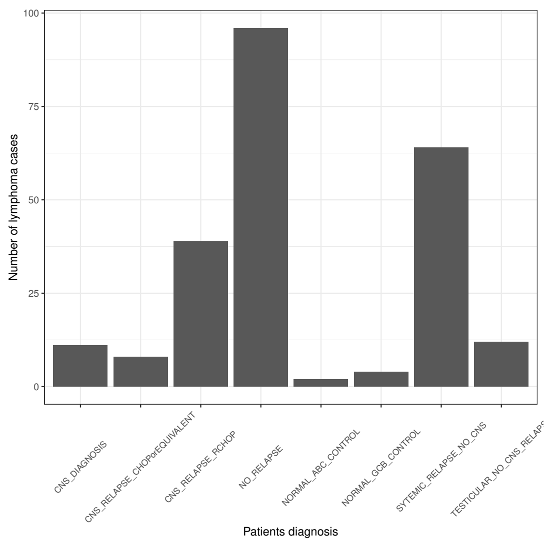
Difference in cases being indexed based on their *cell-of-origin* association subtypes using either of the following features: prediction, ABClassify, ABCScore.

```
metadata %>%
    select(Prediction, ABClassify, ABCScore) %>%
    summary

 Prediction ABClassify ABCScore
 ABC : 92   ABC:103    ABC: 92
 GCB :103   GCB:117    GCB:103
 U   : 39   U  : 16    U  : 41
 NA's:  2
```

Distribution of samples with different treatments.

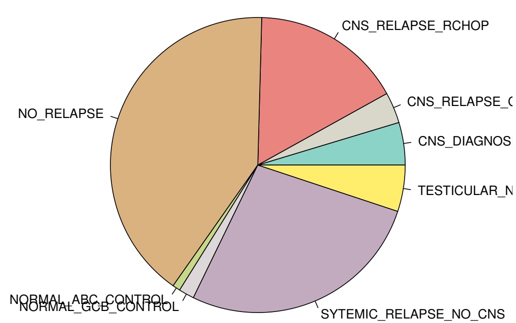
```
metadata %>%
```

```
    select(GROUP) %>%
    ggplot(aes(x = GROUP)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of lymphoma cases",
         x = "Patients diagnosis") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
                                     size = 8))

Warning:  Ignoring unknown parameters:  binwidth, bins, pad
```




Or as a pie chart.

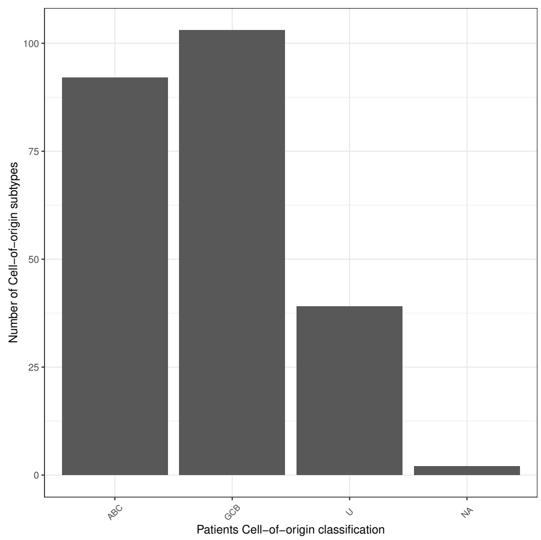
```
palette.pies <- brewer.pal(12, name = "Set3")
palette.pies.adj <- colorRampPalette(palette.pies)(length(unique(metadata$GROUP)))
pie(table(metadata$GROUP), col=palette.pies.adj)
```




Distribution of samples with different cells of origin subtypes.

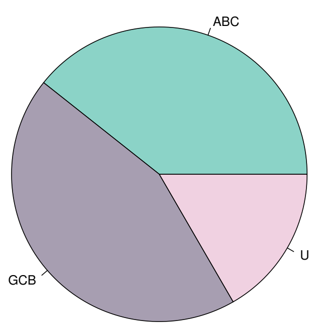
```
metadata %>%
```

```
    select(Prediction) %>%
    ggplot(aes(x = Prediction)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of Cell-of-origin subtypes",
         x = "Patients Cell-of-origin classification") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
                                     size = 8))

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




Or as pie chart.

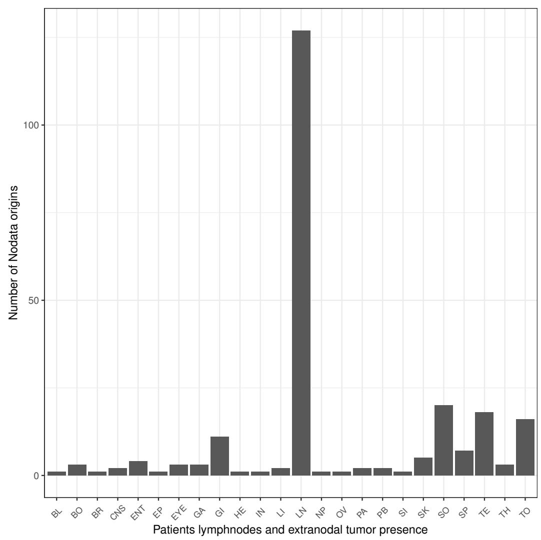
```
palette.pies <- brewer.pal(12, name = "Set3")
palette.pies.adj <- colorRampPalette(palette.pies)(length(unique(metadata$Prediction)))
pie(table(metadata$Prediction), col=palette.pies.adj)
```




Distribution of samples with different lymphnodes and extranodal cancer metastasis.

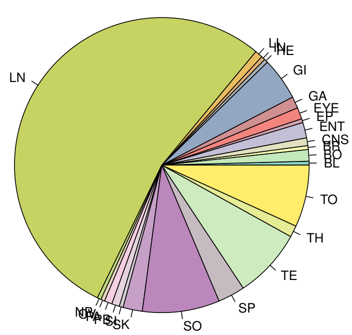
```
par(mfrow=c(2,2))
```

```r
metadata %>%
     select(SITE) %>%
     ggplot(aes(x = SITE)) +
     geom_histogram(stat = "count") +
     labs(y = "Number of Nodata origins",
          x = "Patients lymphnodes and extranodal tumor presence") +
     theme_bw() +
     theme(axis.text.x = element_text(vjust = .5,
                                      angle = 45,
                                      size = 8))

Warning:  Ignoring unknown parameters:  binwidth, bins, pad
```




Or as a pie chart.

```r
palette.pies <- brewer.pal(12, name = "Set3")
palette.pies.adj <- colorRampPalette(palette.pies)(length(unique(metadata$SITE)))
pie(table(metadata$SITE), col=palette.pies.adj)
```




## 2 Differential expression of microarray Affymetrix data

Genes have been fitted in a model that is based on an Empirical Bayes approach. Ranking of the genes determine if they are statistically significant. Bonferroni correction is used to control the false discovery rate (FDR). Moderated t-statistics, FDR, and fold change (log2) are implemented to reduce selection of false positives.

- **adjpval** is the adjusted P-value to control the FDR using Bonferroni correction. **Genes selected here based on their adjpval are also greater than or equal to the bstat threshold**.

- **avgex** is the average expression the ordinary arithmetic average of the log2-expression values for the probe, across all arrays. **Genes selected here based on their avgex are also greater than or equal to the bstat threshold**.

- **bstat** is the moderated t-statistics using an Empirical Bayes approach generating B-statistics scores.

```
expression <- read.table("data/summary.full.390387.txt", sep = "\t", header = T) %>%
    select(Design, Model, Bthreshold, adjPval, Category, Parameter, Transcripts) %>%
    filter(Category == "total")
summary(expression)

              Design                              Model
 CNSvsNOREL    : 12   systemicRelapse              : 36
 CNSvsNOREL_ABC: 12   systemicRelapseCOOprediction:108
 CNSvsNOREL_EN : 12   systemicRelapseNodes         :108
 CNSvsNOREL_GCB: 12
 CNSvsNOREL_LN : 12
 CNSvsSYST     : 12
 (Other)       :180
   Bthreshold        adjPval     Category     Parameter
 Min.   :-4.00   Min.   :0.1   total:252   adjpval:84
 1st Qu.:-2.50   1st Qu.:0.1               avgex  :84
 Median :-1.00   Median :0.1               bval   :84
 Mean   :-1.25   Mean   :0.1
 3rd Qu.: 0.25   3rd Qu.:0.1
 Max.   : 1.00   Max.   :0.1

  Transcripts
 Min.   :   0
 1st Qu.:   0
 Median :   6
 Mean   : 373
 3rd Qu.: 102
 Max.   :6141
```

Number of transcripts when comparing B-statistics scores, which represent confidence in selecting each significantly expressed gene.

```
aggregate( Transcripts ~ Bthreshold, data=expression, FUN=range)

  Bthreshold Transcripts.1 Transcripts.2
1         -4             0          6141
2         -2             0           820
3          0             0           103
4          1             0            44
```

Number of transcripts when samples are classed into groups, which are based on clinical data (e.g., cell-of-origin, CNS relapse, and nodal/extranodal tumor transmission).

```
aggregate( Transcripts ~ Model, data=expression, FUN=range)

                         Model Transcripts.1 Transcripts.2
1              systemicRelapse             0          6141
2 systemicRelapseCOOprediction             0          5563
3         systemicRelapseNodes             0          3995
```

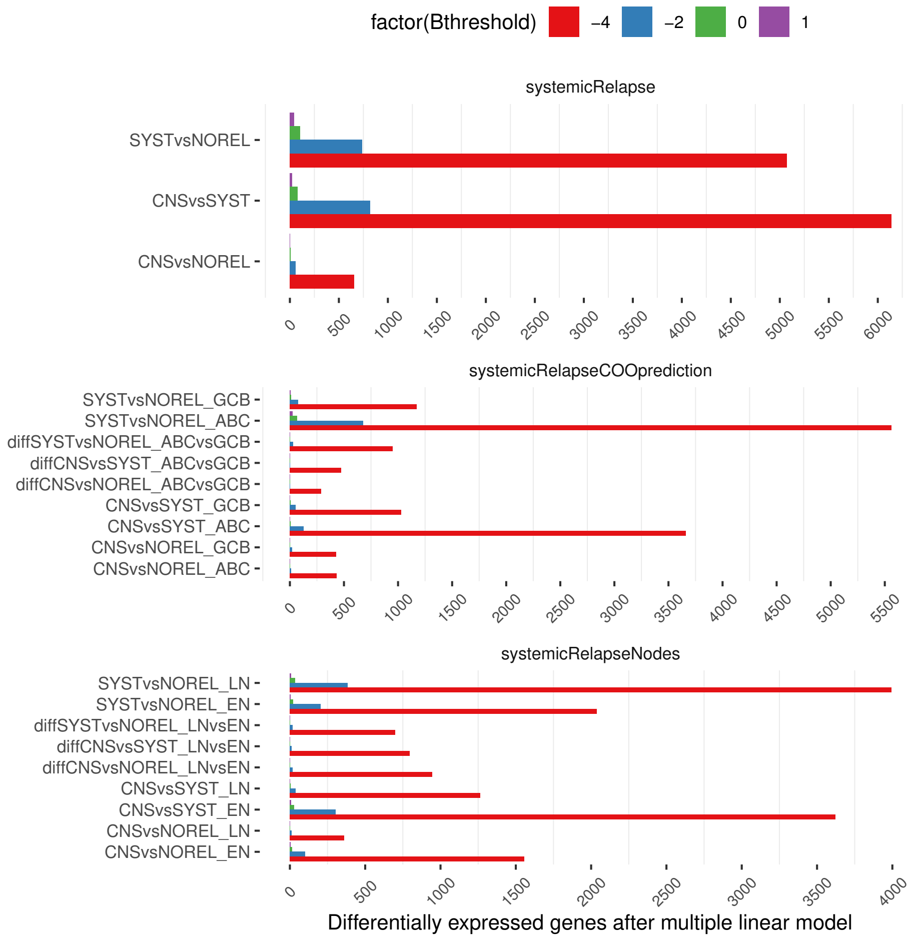
Number of transcripts found when comparing different sample cases indexed based on their clinical data.

```
aggregate( Transcripts ~ Design, data=expression, FUN=range)
```

```
                    Design Transcripts.1 Transcripts.2
1                CNSvsNOREL             0           658
2            CNSvsNOREL_ABC             0           434
3             CNSvsNOREL_EN             0          1557
4            CNSvsNOREL_GCB             0           427
5             CNSvsNOREL_LN             0           361
6                 CNSvsSYST            24          6141
7             CNSvsSYST_ABC             0          3661
8              CNSvsSYST_EN             1          3621
9             CNSvsSYST_GCB             2          1029
10             CNSvsSYST_LN             0          1264
11  diffCNSvsNOREL_ABCvsGCB             0           289
12    diffCNSvsNOREL_LNvsEN             0           944
13   diffCNSvsSYST_ABCvsGCB             0           473
14     diffCNSvsSYST_LNvsEN             0           796
15 diffSYSTvsNOREL_ABCvsGCB             0           952
16   diffSYSTvsNOREL_LNvsEN             0           700
17              SYSTvsNOREL            44          5071
18          SYSTvsNOREL_ABC            27          5563
19           SYSTvsNOREL_EN             1          2039
20          SYSTvsNOREL_GCB             0          1174
21           SYSTvsNOREL_LN             0          3995
```

Number of genes that respond to treatment, cell subtypes, and nodal transmission.

```
expression %>%
    ggplot(aes(
        x = Design,
        y = Transcripts,
        fill = factor(Bthreshold))) +
    theme_bw() +
    geom_bar(stat = "identity",
             position = "dodge") +
    coord_flip() +
    facet_wrap( ~ Model,
              ncol = 1,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    labs(x = "",
         y = "Differentially expressed genes after multiple linear model") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    scale_y_continuous(breaks = pretty(expression$Transcripts, n = 10)) +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
                                     size = 8))
```




## 2.1 Cleaning and removing non-essential genes

Subsetting the data by reducing the number of gene profiles improves interpretation and reduces noise. It is well established that many machine learning models used for classification can be sensitive to high number of *irrelevant* genes, others like support vector machines and random forests are less so (Statnikov 2008).

↰ $\sigma^2$ is the average of the squared differences from the $\mu$

Each array contains probes of 75,523 functional and non-functional RNAs. Either ncRNA, mRNA, and non annotated genes. More than 53.32% of the probes are non-coding. For interpretation purpose, ncRNAs profiles were discarded before fitting the expressions. In addition, the variation from the mean of each transcript was assessed and the spread of expression were all used to discard top and bottom variants. Individual genes that vary widely from the mean of the array were removed thus reducing the spread of the expression across profiles. Transcripts with potential biased high expressions were thus flagged and discarded thus improving correlation of other transcripts. Subsetting was done after normalization of all datasets, all arrays. This would reduce technical errors appearing significant when comparing arrays between each others. Data was transformed (standardization protocol) before calculating means and variances. This helps a better signal recovery from a large dataset with potential expression bias.

↰ Each array correspond to a DLBCL patient's case

### 2.1.1 Variance optimization for each array

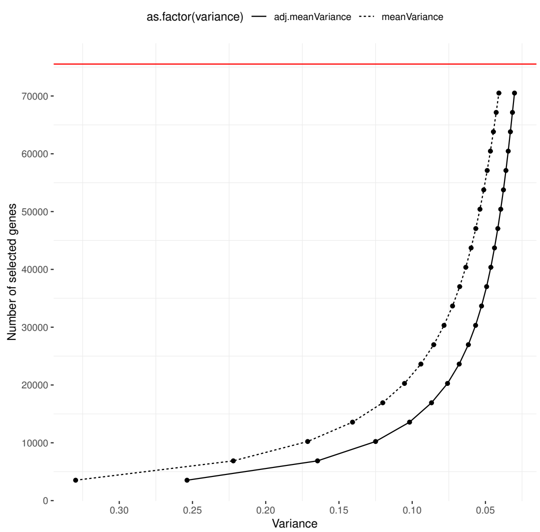
Full probe list accounting for 75,523 genes (red horizontal line). The full line represents the variance after being adjusted by iteratively discarding top/low variant expression profiles. The dotted line represent the original variance before discarding genes.

The graph below shows that by discarding highly variant expressions and selecting only the top 1613 genes for example, the mean variance of the whole array (0.27) is higher than a ranked subset of 10,811 (0.09). Ideally, the reduction of the data is on both, the mean variance and mean standard deviation of the whole array.

↰ The smaller the variance, the better

```
read.table("./data/summary.139102.adjusted.means.subsetting.txt", header = T) %>%
```

```r
    select(dimension, meanVariance, adj.meanVariance) %>%
    gather("variance", "count", 2:3) %>%
    ggplot(aes(x = count,
               y = dimension)) +
    theme_bw() +
    geom_line(aes(linetype = as.factor(variance))) +
    geom_point() +
    scale_x_continuous(trans = "reverse",
                       breaks = scales::pretty_breaks(n = 10)) +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
    geom_hline(aes(yintercept = 75523), colour = "red") +
    labs(y = "Number of selected genes",
         x = "Variance") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```



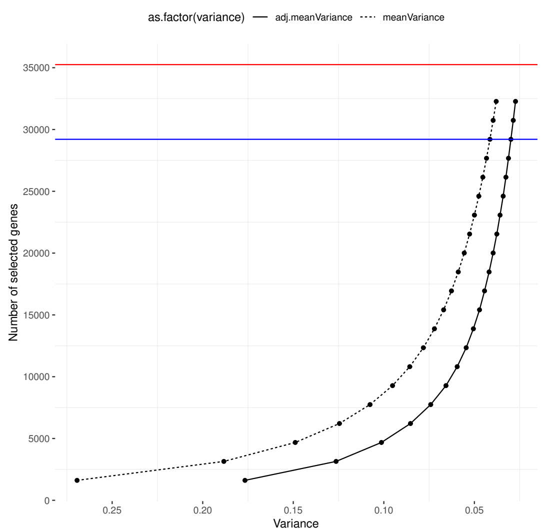

Same plot description as above however we removed ncRNA which account for 53.32% of the probes. The total number of transcripts is now 35,253 (46%, red horizontal line). The blue horizontal line represents the threshold that was selected for subsequent analysis.

↰ 29,207 genes were selected for clustering and nets

By discarding 1198 transcripts from the 35,253 the top outliers with high variance are not included in the clustering process. More rare expression signals will get distinguished. Also, the size of the dataset was reduced to 29,207 by removing transcripts with little deviation from the mean of each array. The total number of transcripts by array was kept above 25k to increase the sizes of the clusters (modules and networks) in later analyses. For example, network analysis on 20k transcripts generated network sizes between 200 and 500. At 29k networks have a total size over 700 nodes.

```r
read.table("./data/summary.149317.adjusted.means.subsetting.txt", header = T) %>%
```

```
  select(dimension, meanVariance, adj.meanVariance) %>%
  gather("variance", "count", 2:3) %>%
  ggplot(aes(x = count,
             y = dimension)) +
  theme_bw() +
  geom_line(aes(linetype = as.factor(variance))) +
  geom_point() +
  scale_x_continuous(trans = "reverse",
                     breaks = scales::pretty_breaks(n = 8)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  geom_hline(aes(yintercept = 35253), color = "red") +
  geom_hline(aes(yintercept = 29207), color = "blue") +
  labs(y = "Number of selected genes",
       x = "Variance") +
  theme(legend.position = "top",
        strip.background = element_rect(linetype = "blank",
                                         fill = "white"),
        panel.border = element_rect(linetype = "blank",
                                    fill = NA),
        panel.grid.major = element_line(linetype = "blank"))
```




### 2.1.2 Standard deviation optimization for each array

The spread of the gene expression scores is dependent on their variance, their deviation from each array's mean (population mean). By removing potentially noisy expressions we are reducing the spread of the arrays numbers, hence improving recognition of rare gene regulations. Below, the plot shows how the standard deviation, **spread** of the data is getting smaller the more we discard genes with high and low variance.

↰ Best if small spread between 2 SDs

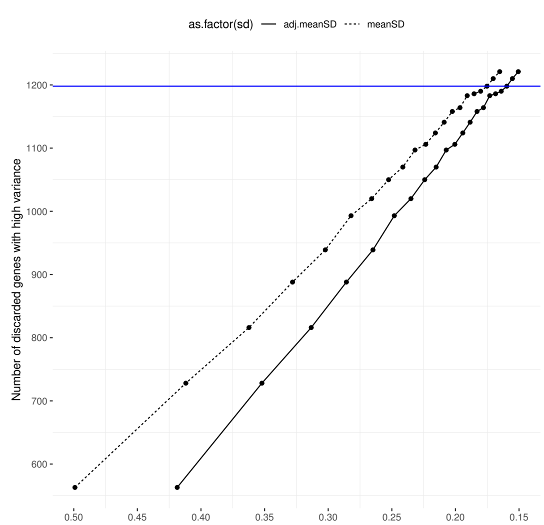
All array probes with all RNAs.

```
pvals <- read.table("./data/summary.139102.adjusted.means.subsetting.txt", header = T) %>%
```

```
    select(discarded, meanSD, adj.meanSD) %>%
    gather("sd", "count", 2:3) %>%
    ggplot(aes(x = count,
               y = discarded)) +
    theme_bw() +
    geom_line(aes(linetype = as.factor(sd))) +
    geom_point() +
    scale_x_continuous(trans = "reverse",
                       breaks = scales::pretty_breaks(n = 8)) +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
    labs(y = "Number of discarded genes with high variance",
         x = "Standard deviation") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```

Without the ncRNAs. Blue horizontal line is the threshold that was selected for later analysis.

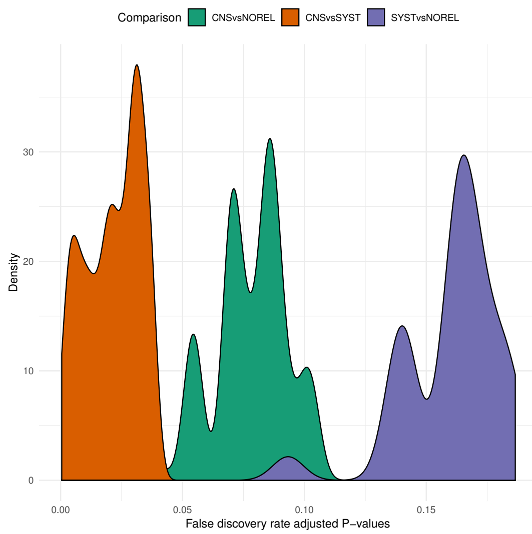
```
read.table("./data/summary.149317.adjusted.means.subsetting.txt", header = T) %>%
    select(discarded, meanSD, adj.meanSD) %>%
    gather("sd", "count", 2:3) %>%
    ggplot(aes(x = count,
               y = discarded)) +
    theme_bw() +
    geom_line(aes(linetype = as.factor(sd))) +
    geom_point() +
    geom_hline(aes(yintercept = 1198), colour = "blue") +
    scale_x_continuous(trans = "reverse",
                       breaks = scales::pretty_breaks(n = 8)) +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
    labs(y = "Number of discarded genes with high variance",
         x = "Standard deviation") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




## 2.2 Distribution of p-values in pairwise comparisons

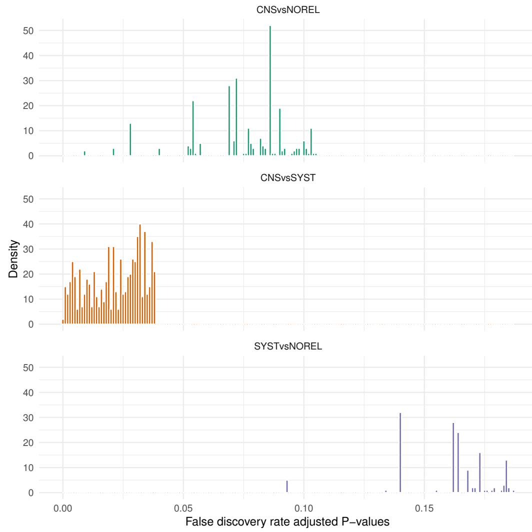
Distribution of p-values in pairwise comparisons between patients recognized as CNS relapse, systemic, and did not relapse.

```r
pvals <- read.table("./data/summary.lmfit.fdrAdjpval.txt", header = TRUE, fill = TRUE)
pvals %>%
    filter(Contrast == "systemicRelapse") %>%
    ggplot(aes(x = FDRadjPval,
               fill = Comparison)) +
    geom_density() +
    theme_minimal() +
    theme(legend.position = "top") +
    scale_fill_brewer(palette = "Dark2") +
    labs(x = "False discovery rate adjusted P-values",
         y = "Density")
```



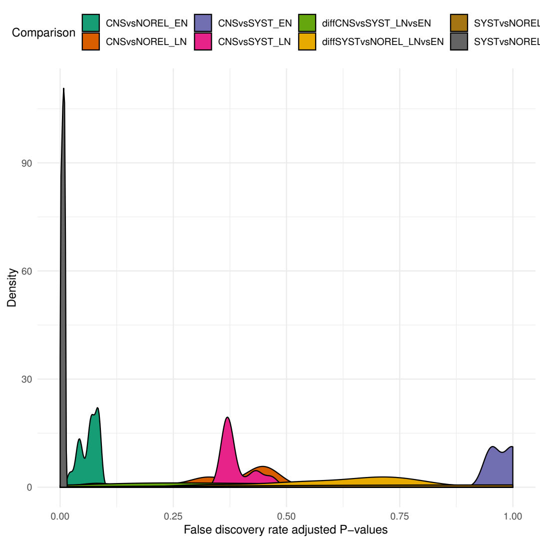

Distribution of p-values in pairwise comparisons between patients recognized as CNS relapse, systemic, and did not relapse.

```r
pvals %>%
    filter(Contrast == "systemicRelapse") %>%
    ggplot(aes(x = FDRadjPval,
               fill = Comparison)) +
    geom_histogram(binwidth = 0.001,
                   col=I("white")) +
    theme_minimal() +
    facet_wrap( ~ Comparison,
               ncol = 1) +
    theme(legend.position = "none") +
    scale_fill_brewer(palette = "Dark2") +
    labs(x = "False discovery rate adjusted P-values",
         y = "Density")
```




Distribution of p-values in pairwise comparisons between patients recognized with lymph nodes and extranodal implication.

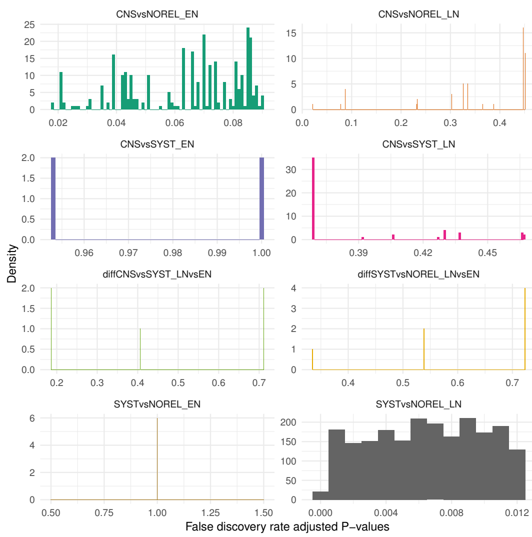
```r
pvals %>%
    filter(Contrast == c("systemicRelapseNodes")) %>%
    ggplot(aes(x = FDRadjPval,
               fill = Comparison)) +
    geom_density() +
    theme_minimal() +
    theme(legend.position = "top") +
    scale_fill_brewer(palette = "Dark2") +
    labs(x = "False discovery rate adjusted P-values",
         y = "Density")
```




Distribution of p-values in pairwise comparisons between patients recognized with lymph nodes and extranodal implication.

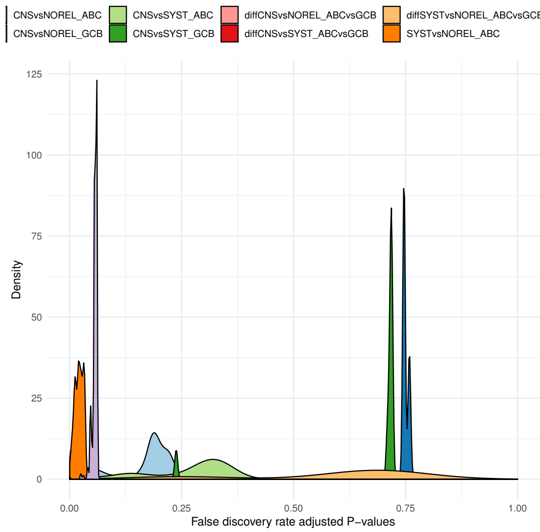
```r
pvals %>%
```

```r
    filter(Contrast == "systemicRelapseNodes") %>%
    ggplot(aes(x = FDRadjPval,
               fill = Comparison)) +
    geom_histogram(binwidth = 0.001) +
    theme_minimal() +
    facet_wrap( ~ Comparison,
               ncol = 2, scales = "free") +
    theme(legend.position = "none") +
    scale_fill_brewer(palette = "Dark2") +
    labs(x = "False discovery rate adjusted P-values",
         y = "Density")
```




Distribution of p-values in pairwise comparisons between patients recognized with cell-of-origin based on ABC or GCB.

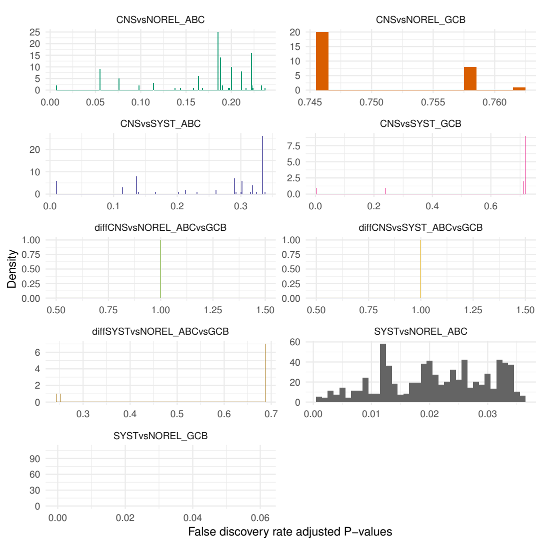
```r
pvals %>%
    filter(Contrast == c("systemicRelapseCOOprediction")) %>%
    ggplot(aes(x = FDRadjPval,
               fill = Comparison)) +
    geom_density() +
    theme_minimal() +
    theme(legend.position = "top") +
    scale_fill_brewer(palette = "Paired") +
    labs(x = "False discovery rate adjusted P-values",
         y = "Density")

Warning:  Groups with fewer than two data points have been dropped.
Warning:  Groups with fewer than two data points have been dropped.
```




Distribution of p-values in pairwise comparisons between patients recognized with cell-of-origin based on ABC or GCB.

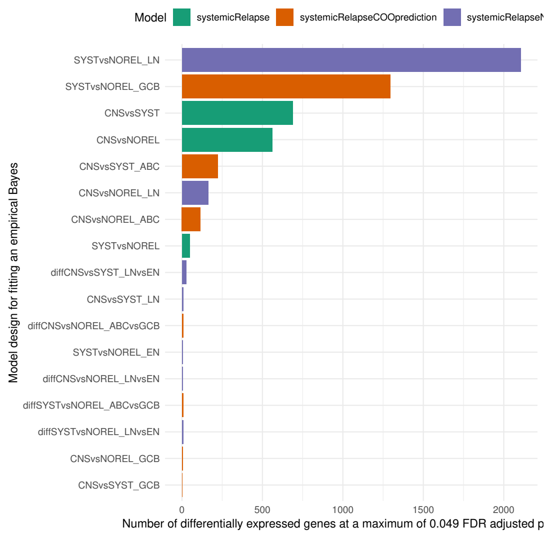
```
pvals %>%
    filter(Contrast == "systemicRelapseCOOprediction") %>%
    ggplot(aes(x = FDRadjPval,
               fill = Comparison)) +
    geom_histogram(binwidth = 0.001) +
    theme_minimal() +
    facet_wrap( ~ Comparison,
               ncol = 2, scales = "free") +
    theme(legend.position = "none") +
    scale_fill_brewer(palette = "Dark2") +
    labs(x = "False discovery rate adjusted P-values",
         y = "Density")
```

```
Warning in RColorBrewer::brewer.pal(n, pal):  n too large, allowed maximum for
palette Dark2 is 8
Returning the palette you asked for with that many colors
```




```
read.table("./data/summary.lmfit.bval.txt", header = TRUE, fill = TRUE) %>%
```

```r
    filter(Model == c("systemicRelapse","systemicRelapseNodes","systemicRelapseCOOprediction")) %>%
    ggplot(aes(x = reorder(Design, Transcripts),
               y = Transcripts,
               fill = Model)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    coord_flip() +
    theme_minimal() +
    theme(legend.position = "top") +
    scale_fill_brewer(palette = "Dark2") +
    labs(y = "Number of differentially expressed genes at a maximum of 0.049 FDR adjusted p-value",
         x = "Model design for fitting an empirical Bayes")
```




## 3 Clustering and network analyses

The number of clusters and modules per networks are assigned by designing first a similarity matrix between differentially expressed gene for any two conditions (eg., relapse vs no relapse patient cases). An adjacency matrix is then constructed by weighting the previously inferred measures. The data is transformed to increase the correlation coefficient therefore improving detection of strong correlated patterns. (Example of the strength of data transformation and correlation, visit the following online page).

Overfitting is a source of bias.

- **MaxEdgesPerGene**, maximum number of correlations per genes
- **NbNodes**, number of genes found for each edge connection bracket
- **Normalization**, method that focuses on creating complete clusters. We tested methods ranging from Complete clustering, Average, and Ward. Each method is detailed here. Only Complete clustering was retained. All other methods overfitted the data.
- **Correlation**, finding ranges from linear to non-linear trends. We tested Pearson and Spearman correlation.
- **Standardization**, data transformation method. We tested transformation by Hellinger, Standardize, Range, and Logarithmic scaling. Each method is detailed here.
- **MaxGenePerModule**, how many genes assigned by cluster (module)
- **SimilaritySize**, number of initial differentially expressed genes
- **EdgeThreshold**, parameter to limit the weight of the edges
- **CorrelationPower**, power transformation of the data

Effect of correlation methods is seen on module content

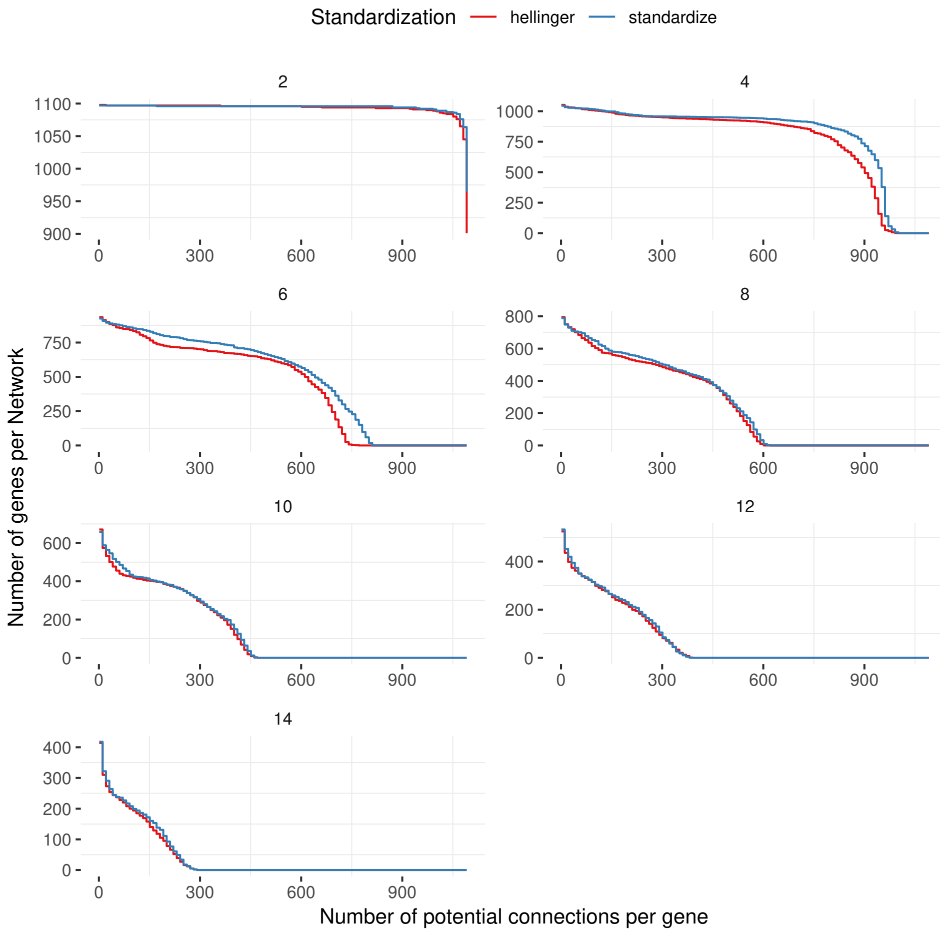
```r
ns <- read.table("./data/networks.summary.104795.txt", header = T)
```

```r
summary(ns)
```

```
 MaxEdgesPerGene     NbNodes       Normalization     Correlation
 Min.   :   1     Min.   :   0   complete:4620   spearman:4620
 1st Qu.: 271     1st Qu.:   0
 Median : 546     Median : 244
 Mean   : 546     Mean   : 406
 3rd Qu.: 821     3rd Qu.: 862
 Max.   :1091     Max.   :1098
    Standardization MaxGenesPerModule SimilaritySize EdgeThreshold
 hellinger  :2310   Min.   :26        Min.   :1099   Min.   :0.5
 standardize:2310   1st Qu.:36        1st Qu.:1099   1st Qu.:0.5
                    Median :55        Median :1099   Median :0.5
                    Mean   :57        Mean   :1099   Mean   :0.5
                    3rd Qu.:79        3rd Qu.:1099   3rd Qu.:0.5
                    Max.   :91        Max.   :1099   Max.   :0.5
 CorrelationPower
 Min.   : 2
 1st Qu.: 4
 Median : 8
 Mean   : 8
 3rd Qu.:12
 Max.   :14
```

Difference between methods used for network inference. Are we able to generate convergence of the output of all iterations across all methods? ↰Test graphs

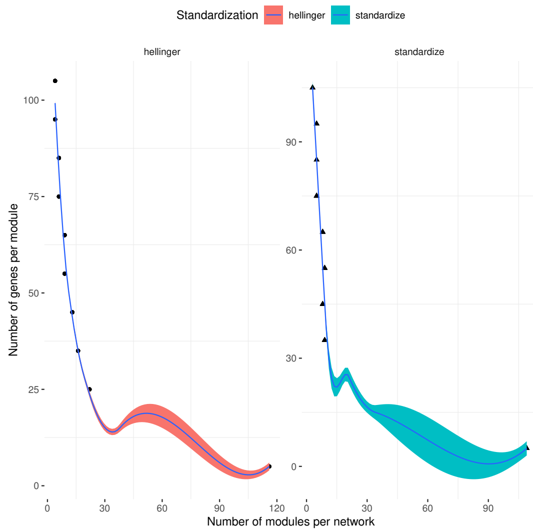
```r
ns %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module. Each module contains differing number of nodes based on their correlation strength. Each cluster contains at least one module. Each network contains at least one cluster. One module can be assigned to nodes that belong to more than one cluster. The Lowess curves show if the trend in the data is linear or not. The wave around Lowess curves represents the level of confidence of the data points (the narrower the interval the better, less variability = more accuracy).

Points=iterations. With less iterations comes high variability of the curve

```
read.table("./data/modules.summary.104795.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.1 Network analysis for Spearman-related correlations (relaxed)

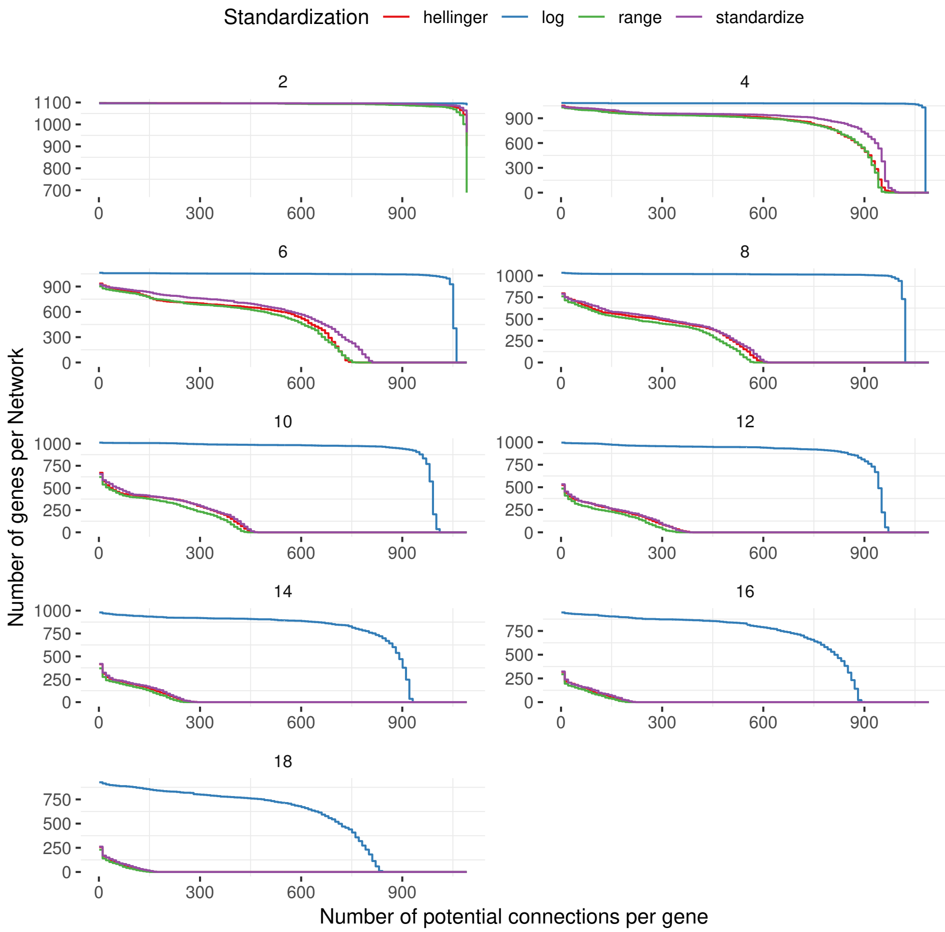
Thresholds based on the Empirical Bayes approach to rank genes and determine if a gene is significantly expressed by shrinking the sample variance Smyth 2004. Limma implementation.

- **Average Expression**: 5
- **Adjusted P-value**: equal or less than 0.045
- **Log Fold Change**: 1
- **B-statisitcs**: 1.5

#### 3.1.1 Nodal versus extra-nodal lymphoma

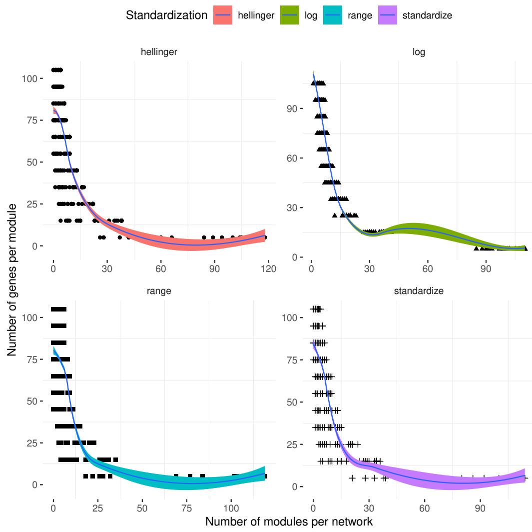
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
read.table("./data/networks.summary.104859.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

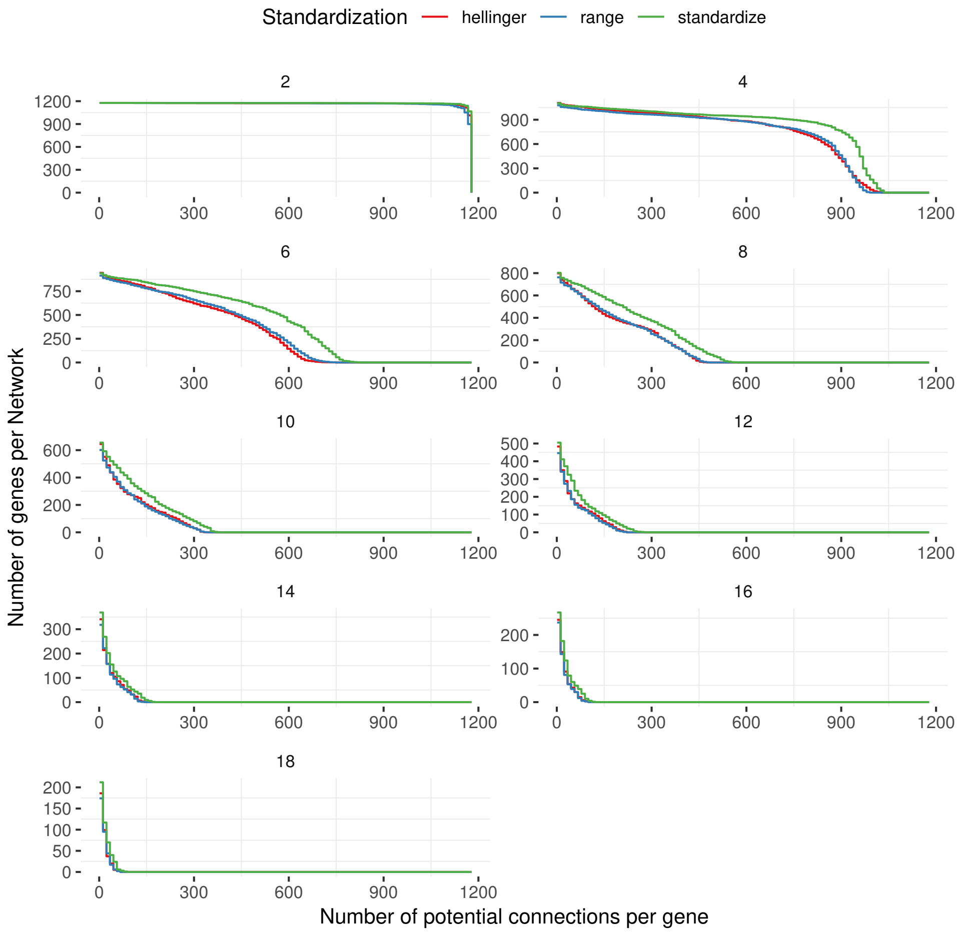
```
read.table("./data/modules.summary.104859.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.1.2 Relapsed versus no CNS relapsed cases

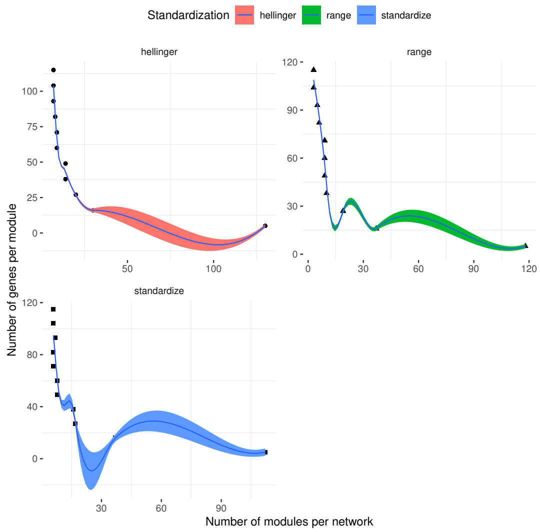
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
read.table("./data/networks.summary.114018.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

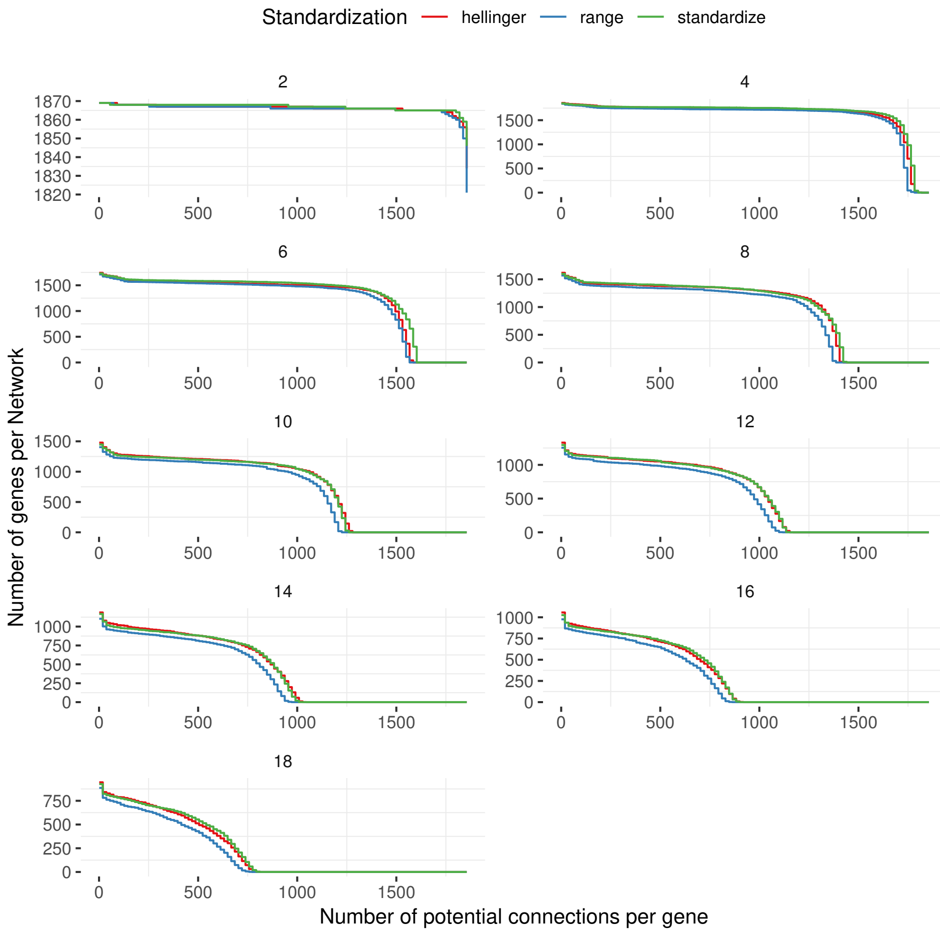
```r
read.table("./data/modules.summary.114018.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.1.3 Lymphoma cases classified by Cell-of-origin subtypes

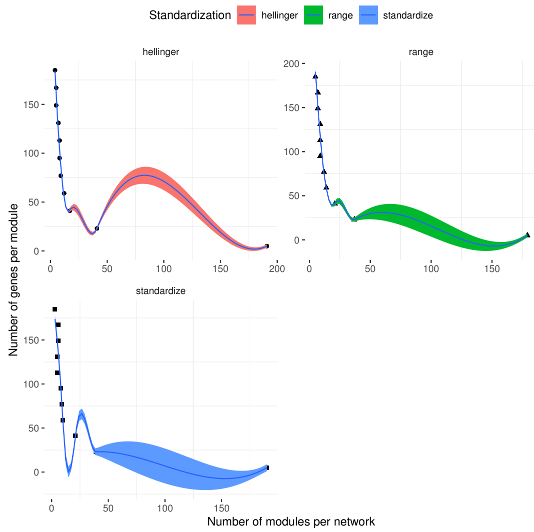
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
read.table("./data/networks.summary.114017.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

```
read.table("./data/modules.summary.114017.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




## 3.2 Network analysis for Pearson-related correlations (relaxed)

Thresholds based on the Empirical Bayes approach to rank genes and determine if a gene is significantly expressed. Limma implementation.

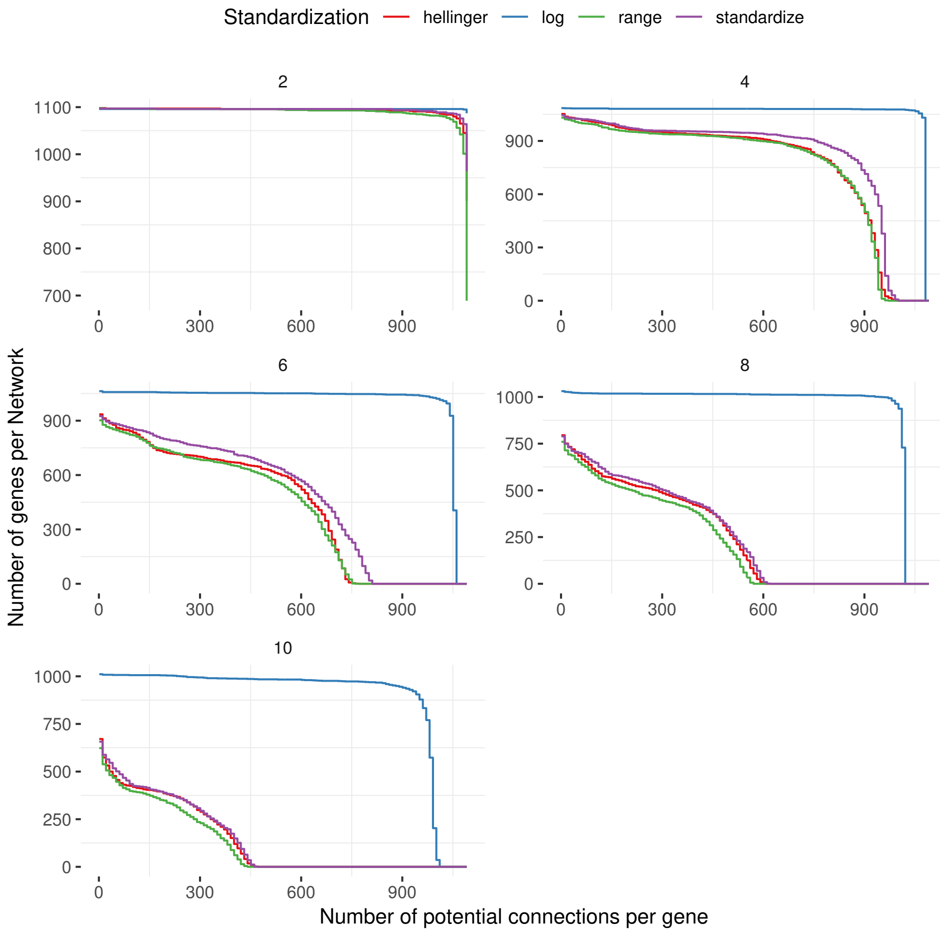
> With pearson, we can only raise the data to power 10. All are discarded after 10.

- **Average Expression**: 5
- **Adjusted P-value**: equal or less than 0.045
- **Log Fold Change**: 1
- **B-statisitcs**: 1.5

### 3.2.1 Nodal versus extra-nodal lymphoma

Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

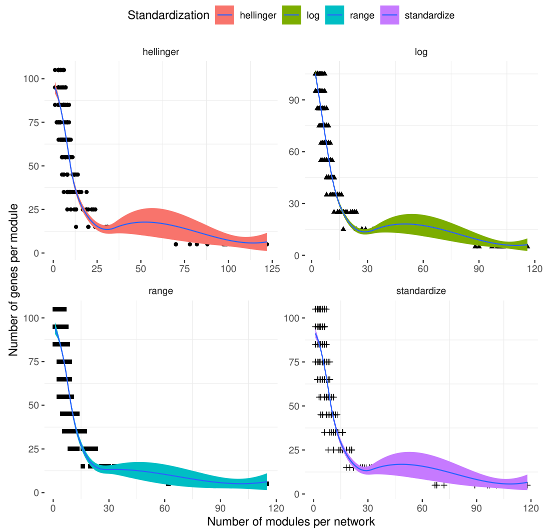
```
read.table("./data/networks.summary.104862.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

```
read.table("./data/modules.summary.104862.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```

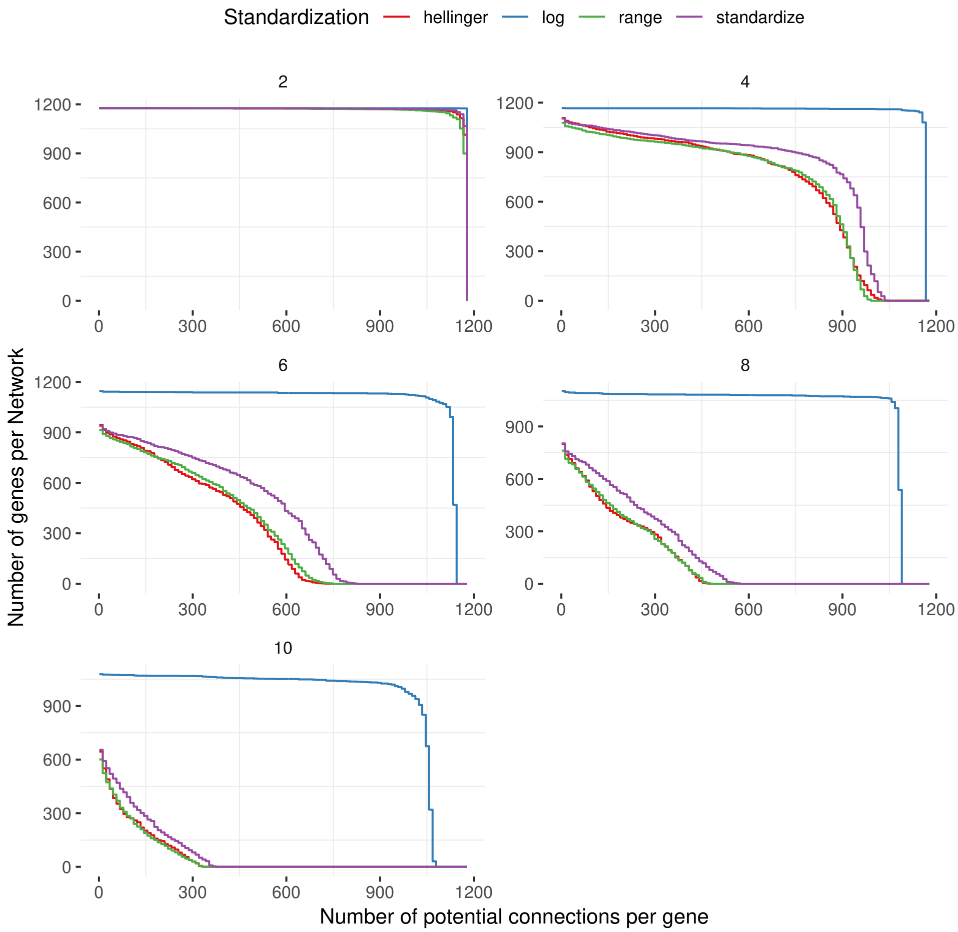
↰Since Lowess ranks by confidence, Log transformation seems the best, ie, low variability. For this, Log is removed from further tests.




### 3.2.2 Relapsed versus no CNS relapsed cases

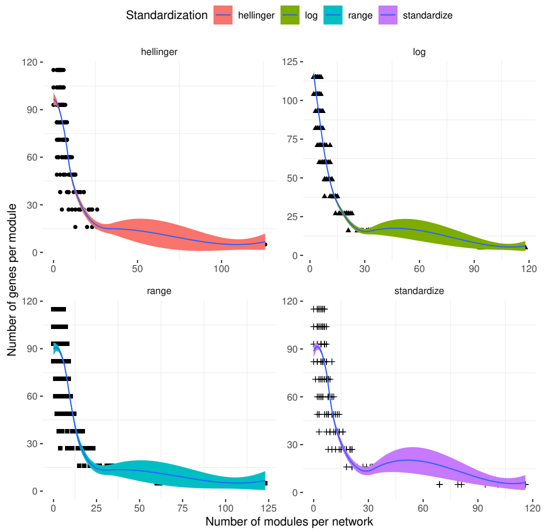
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
read.table("./data/networks.summary.104863.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

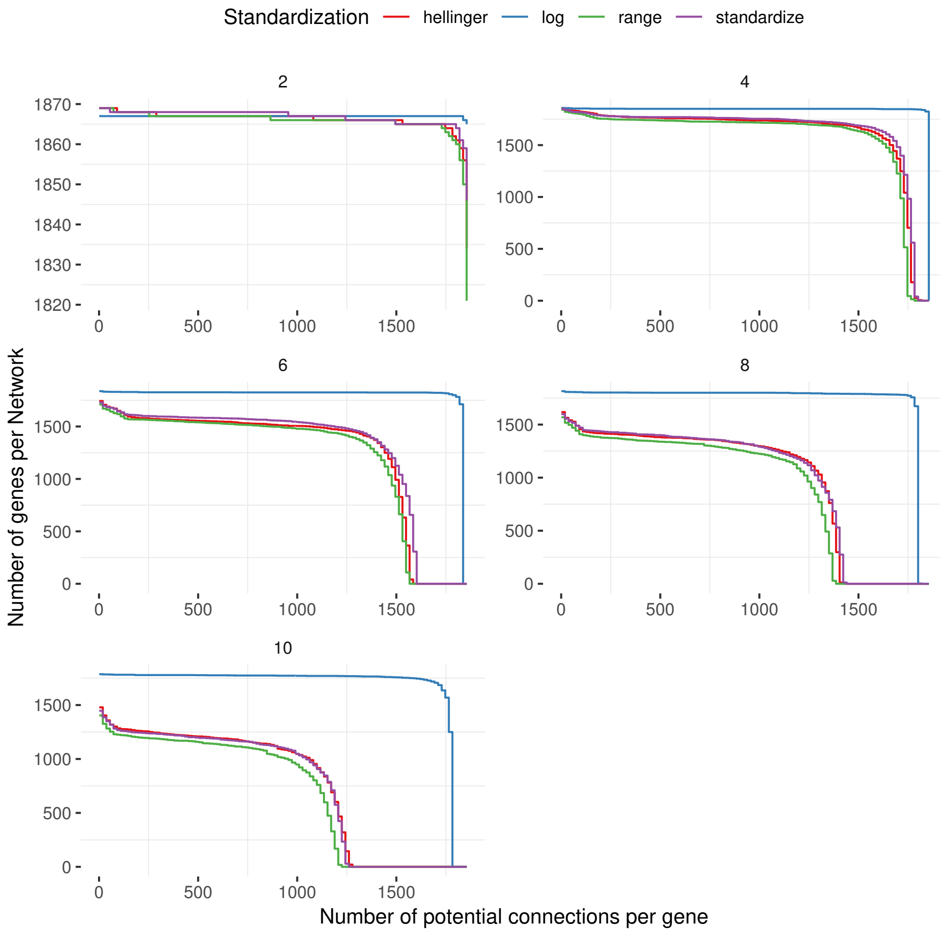
```
read.table("./data/modules.summary.104863.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.2.3 Lymphoma cases classified by Cell-of-origin subtypes

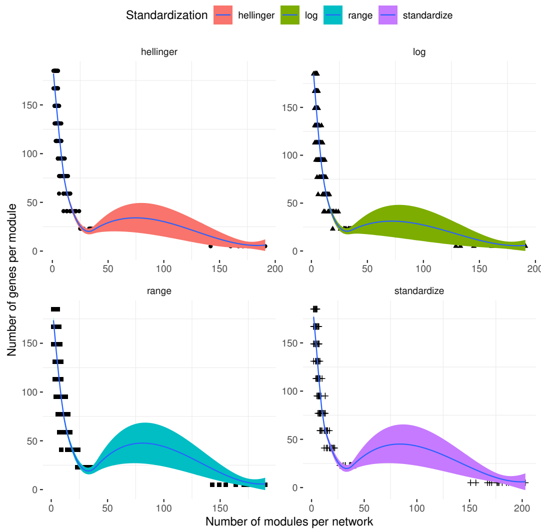
Genetic networks from differentially expressed genes selected by comparing sample cases with cell of origin classification based on ABC or GCB subtypes.

```
read.table("./data/networks.summary.104864.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

```
read.table("./data/modules.summary.104864.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.3 Network analysis for Spearman-related correlations (stringent)

Thresholds based on the Empirical Bayes approach to rank genes and determine if a gene is significantly expressed. Limma implementation.

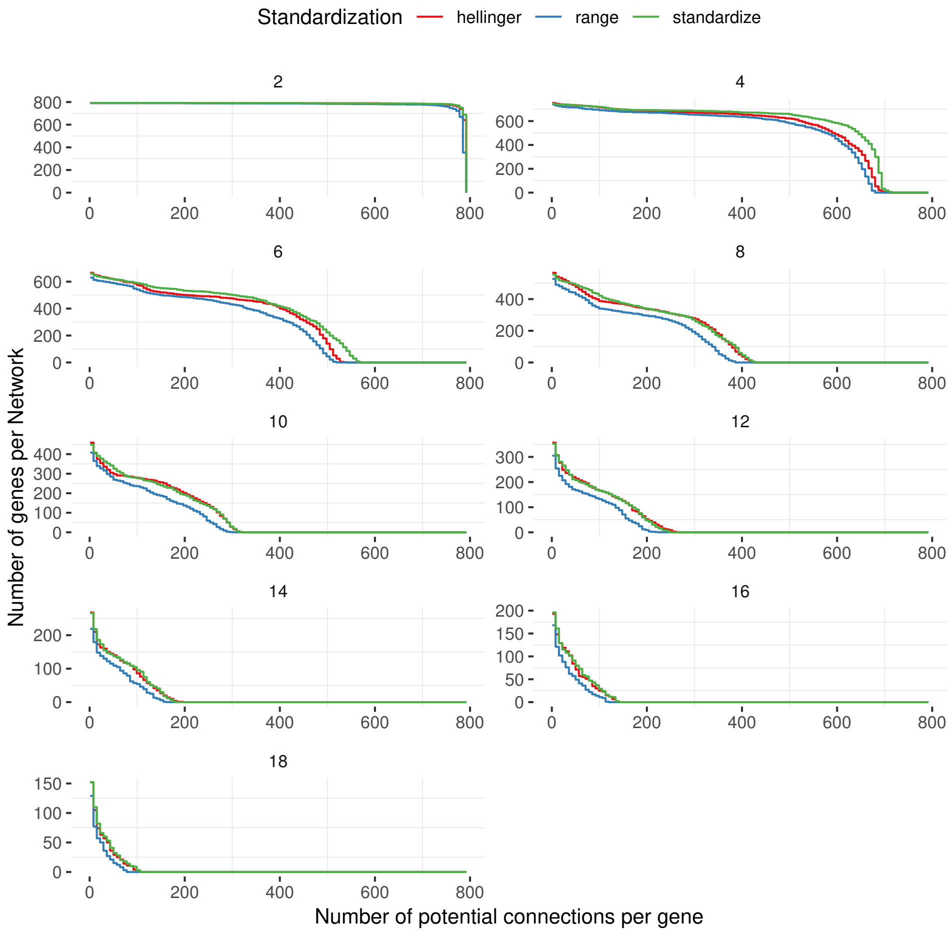
Same analysis with more stringent parameters

- **Average Expression**: 10
- **Adjusted P-value**: equal or less than 0.030
- **Log Fold Change**: 1
- **B-statisitcs**: 2

#### 3.3.1 Nodal versus extra-nodal lymphoma

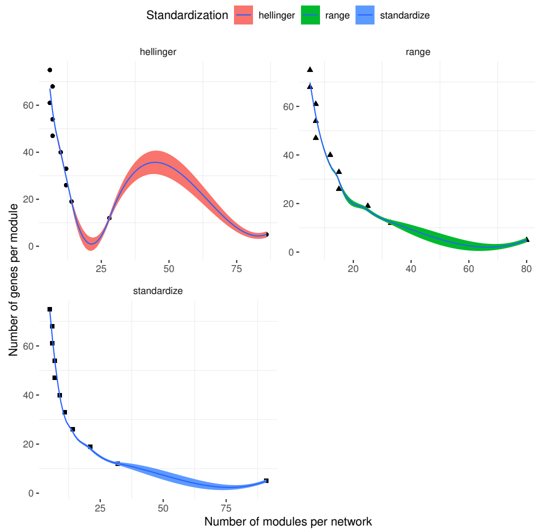
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
read.table("./data/networks.summary.119759.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

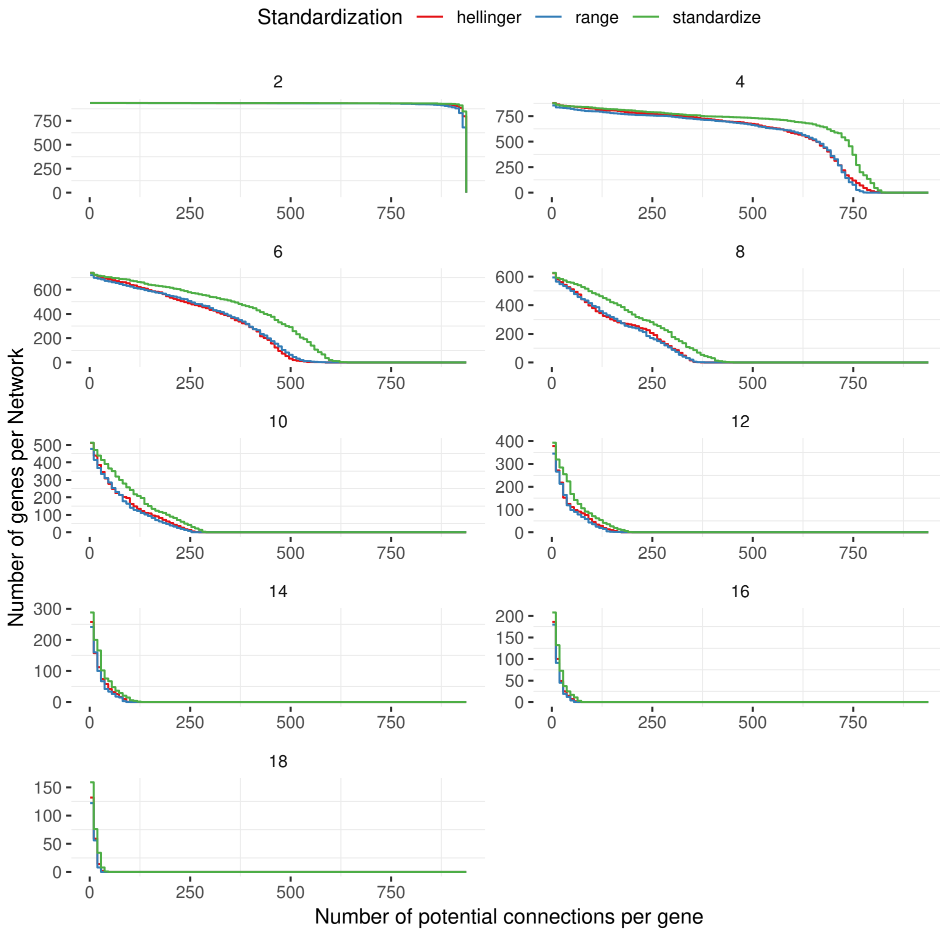
```r
read.table("./data/modules.summary.119759.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
                ncol = 2,
                scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.3.2 Relapsed versus no CNS relapsed cases

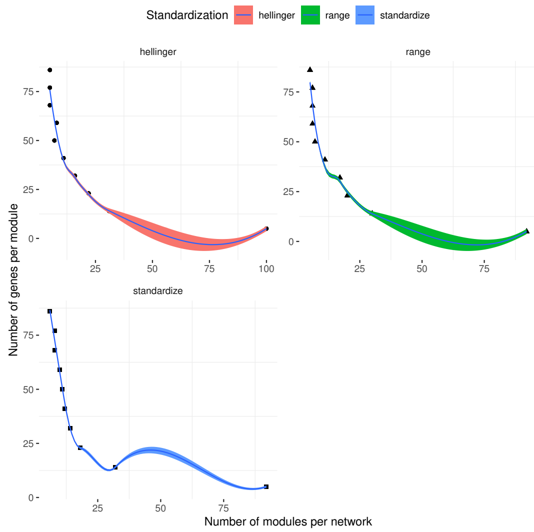
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
read.table("./data/networks.summary.119760.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

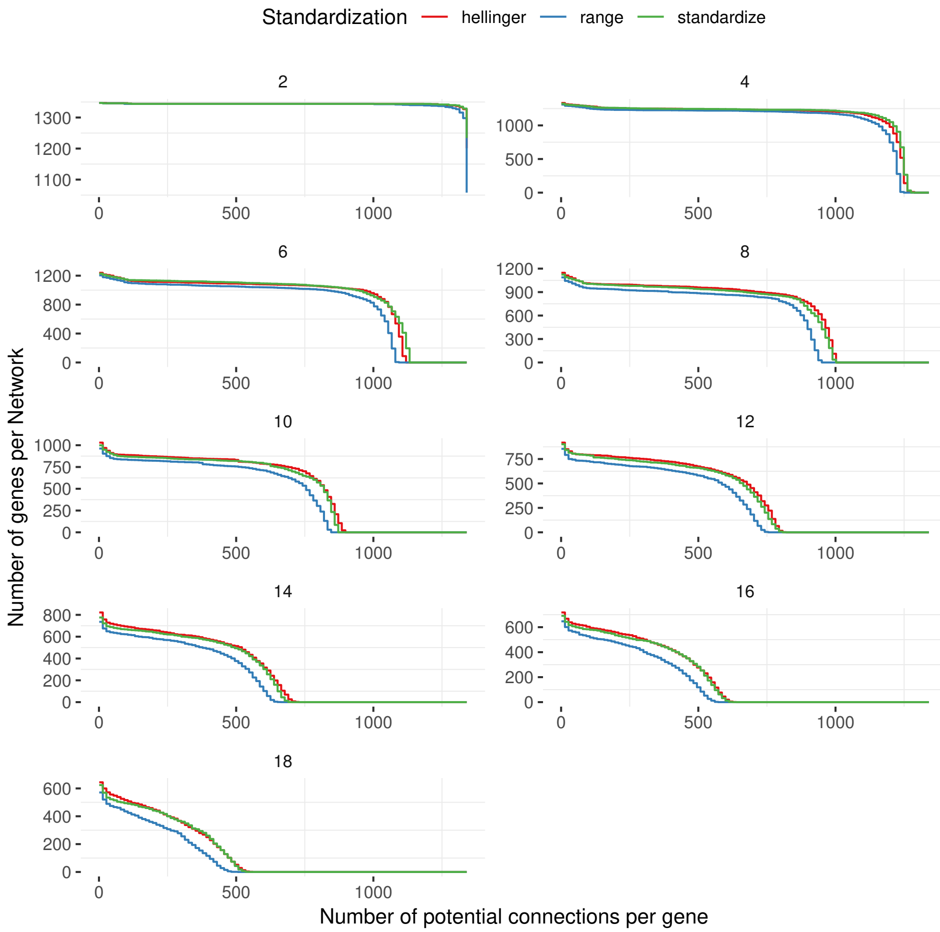
```
read.table("./data/modules.summary.119760.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.3.3 Lymphoma cases classified by Cell-of-origin subtypes

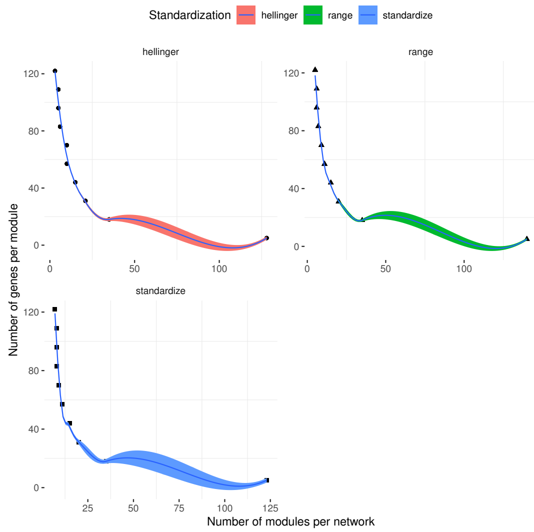
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
read.table("./data/networks.summary.119758.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

```
read.table("./data/modules.summary.119758.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.4 Network analysis for Pearson-related correlations (stringent)

Thresholds based on the Empirical Bayes approach to rank genes and determine if a gene is significantly expressed. Limma implementation.

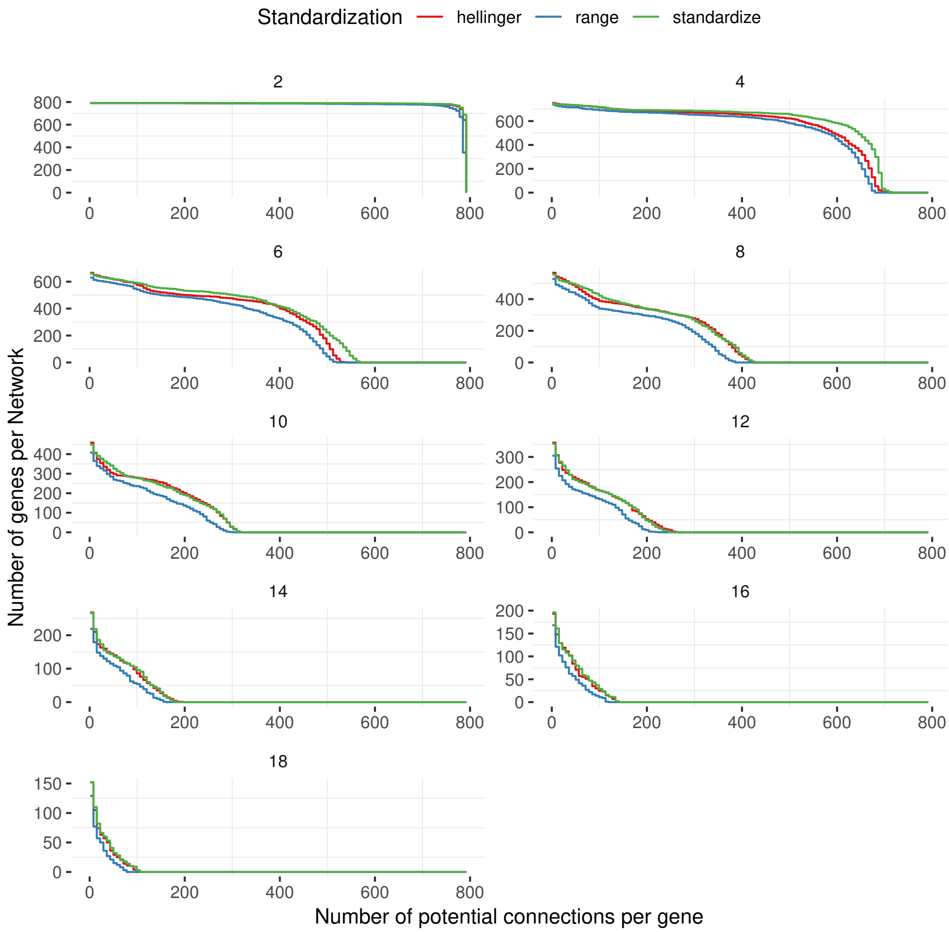
Same analysis with more stringent parameters

- **Average Expression**: 10
- **Adjusted P-value**: equal or less than 0.030
- **Log Fold Change**: 1
- **B-statisitcs**: 2

#### 3.4.1 Nodal versus extra-nodal lymphoma

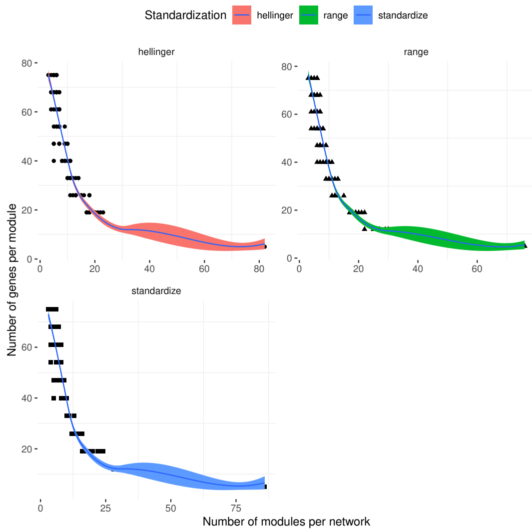
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
read.table("./data/networks.summary.119755.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

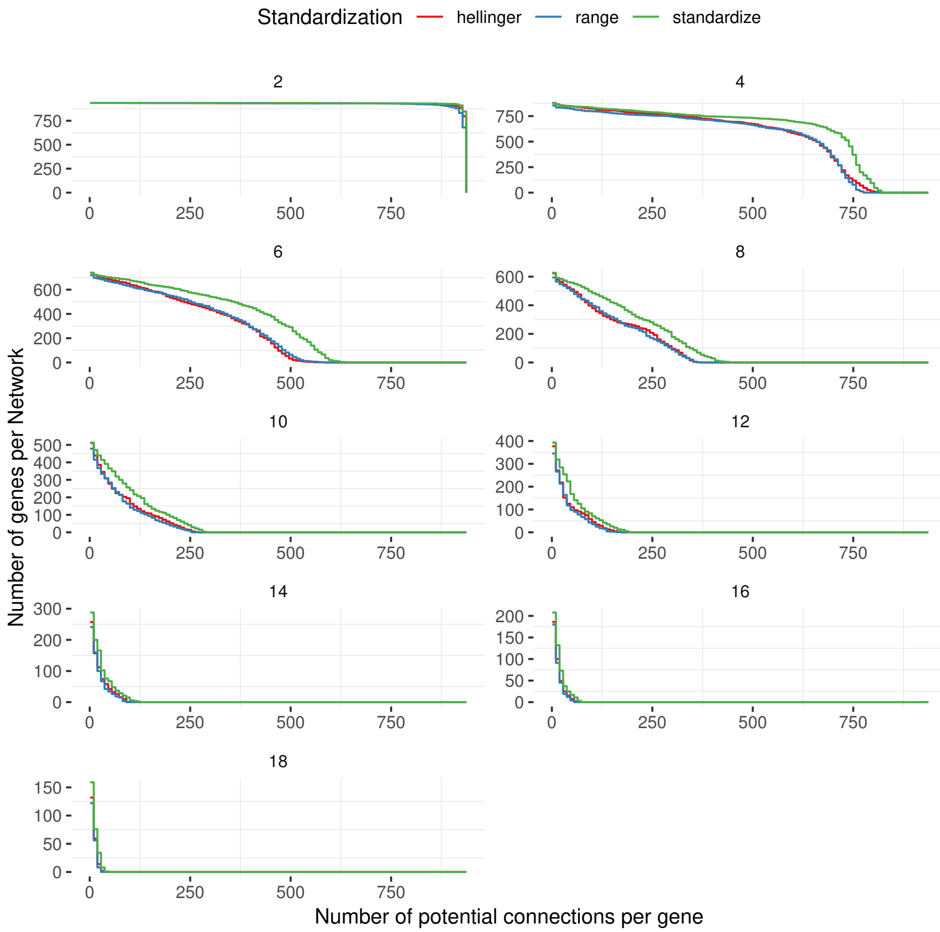
```
read.table("./data/modules.summary.119755.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.4.2 Relapsed versus no CNS relapsed cases

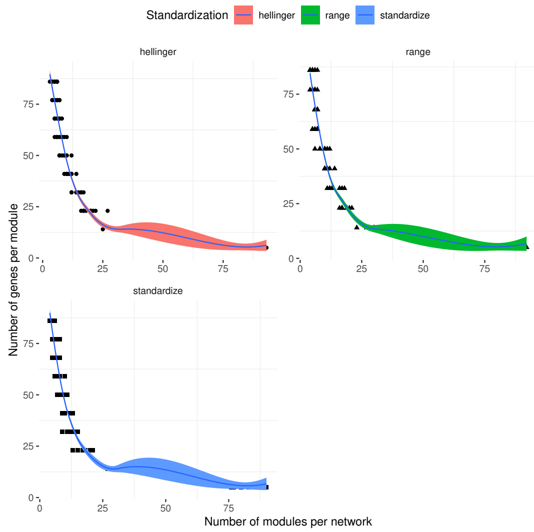
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
read.table("./data/networks.summary.119754.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

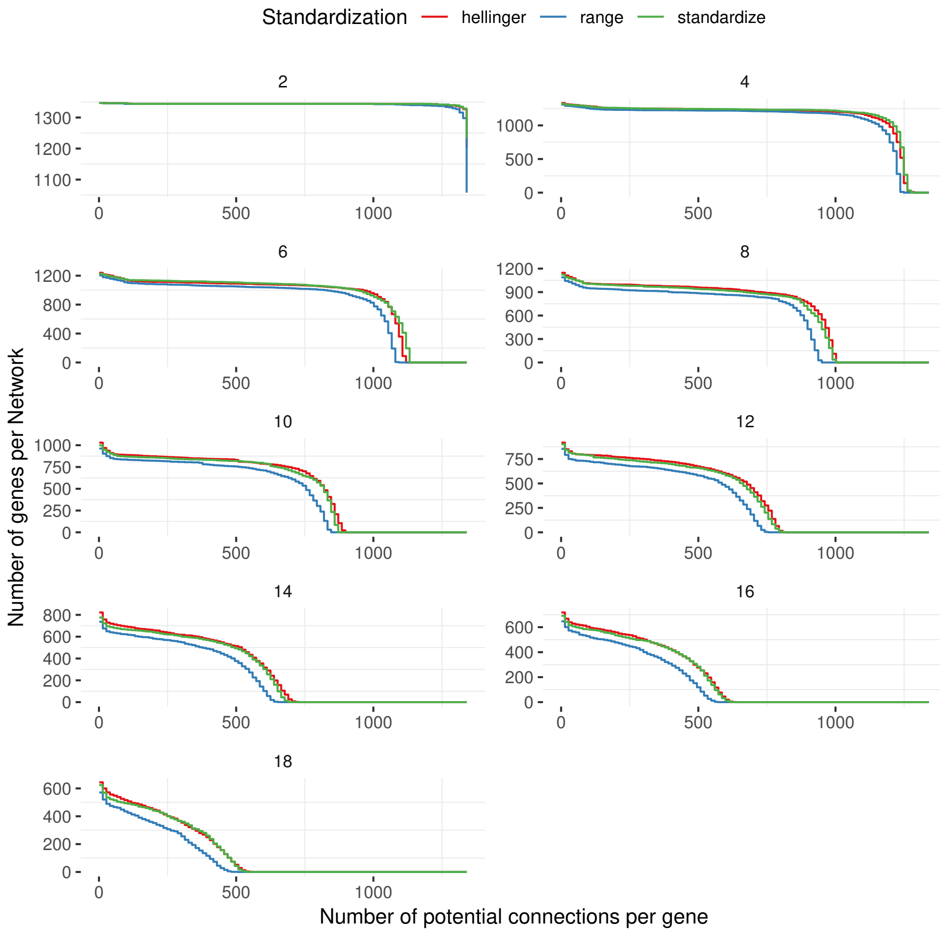
```
read.table("./data/modules.summary.119754.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




### 3.4.3 Lymphoma cases classified by Cell-of-origin subtypes

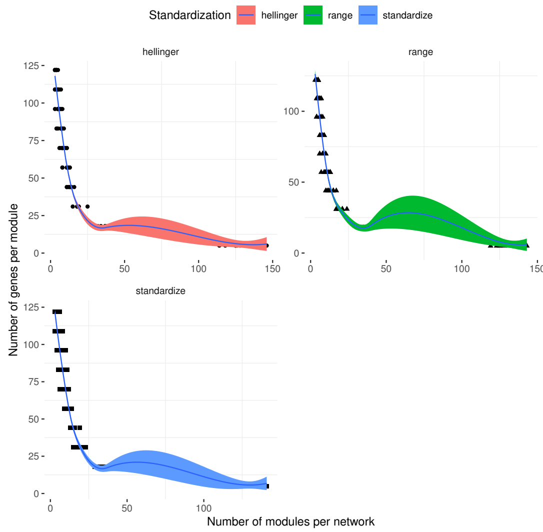
Genetic networks from differentially expressed genes selected by comparing sample cases with cell of origin classification based on ABC or GCB subtypes.

```
read.table("./data/networks.summary.119757.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of potential connections per gene",
         y = "Number of genes per Network") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                           fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank"))
```




Showing the number of modules per network and the number of genes per module.

```
read.table("./data/modules.summary.119757.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "Number of modules per network",
         y = "Number of genes per module") +
    facet_wrap( ~ Standardization,
               ncol = 2,
               scales = "free") +
    theme(legend.position = "top",
          strip.background = element_rect(linetype = "blank",
                                          fill = "white"),
          panel.border = element_rect(linetype = "blank",
                                      fill = NA),
          panel.grid.major = element_line(linetype = "blank")) +
    geom_smooth(method = 'loess', size = .5, level = 0.5, alpha=1)
```




# 4 Machine Learning

Machine learning models were used for classification of patients cases into systemic relapse of DLBCL, CNS relapse or no relapse. Data are gene expression from Affymetrix arrays of 240 patients with a form of DLBCL. Subsets of the whole number of microarray probes will be used for classification.

## 4.1 Regularization

Least absolute shrinkage and selection operator (LASSO) was used for dimension reduction. Gene expression profiles were extracted from networks with significant connectivity. Subset selection using lasso, penalizes genes based on coefficient estimates, to increase accuracy of classification. Briefly, cases are assigned to either diagnosis category, systemic relapse (SYST), CNS relapse (CNS), and no relapse (NOREL). During each iteration, a prediction is made to assign a category. Then a probability is calculated for having an accuracy performance for that iteration. A single iteration has a different random seed, which generates a different set of lambda coefficients for adjusting the lasso penalty. The best lambda across a grid of coefficients with the best accuracy classification is then selected based on accuracy. Adjusting the lambda score also adjusts the subset of genes used for the classification. For one best lambda there is one subset of significantly expressed genes and each gene has a different probability. For one best lambda there is one mean probability registered for that subset of genes.

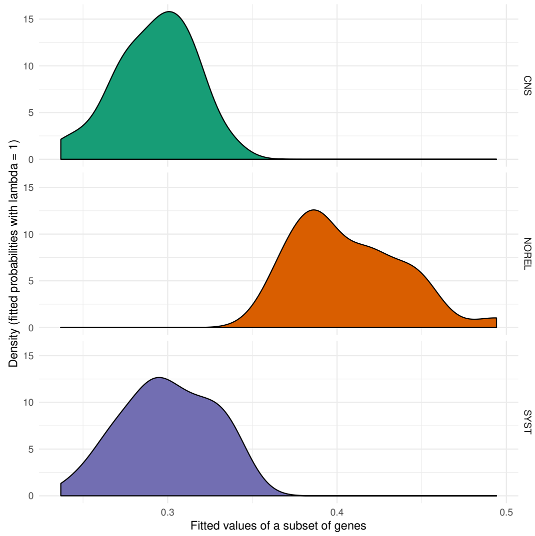
↰If a subset has 50 genes, the reported probabilities are the mean of each gene individual probability to predict all patient cases

### 4.1.1 Uncertainty estimation for selected genes from expression networks

The chart below shows how many iterations (dots) were executed for each sample group before selecting a subset of genes through regularization.
The probabilities are the fitted values of either a multinomial or binomial model at the best lambda ($\lambda$), shrinking parameter determined by tuning and cross-validation resampling. When predictions were made with lasso, the least squares were penalized. Lasso zero out coefficient estimates thus reducing the data. The fitted values are compared to the outcome to follow the proportion of variance "explained" by the model and the proportion of variance "not explained".

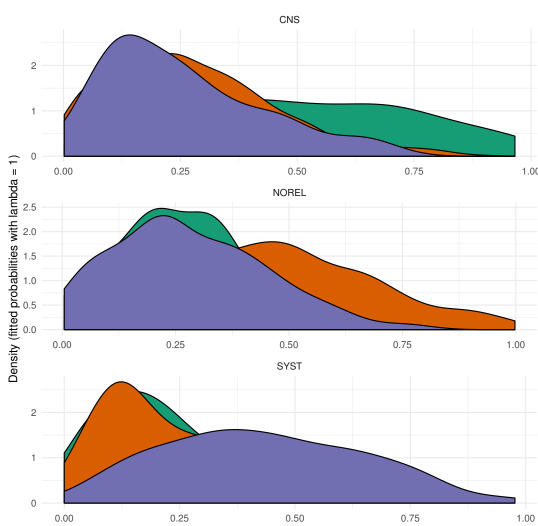
Peaks in density represents variance fitted at best $\lambda$ between sample groups. Probabilities are compared to the residuals of the data, the outcome is the fitted values. As long as the peaks differ between groups, then the prediction is possible between samples. There is an overlay between CNS and SYST groups, which indicates the presence of some bias in differentiating between them.

```
read.table("./data/logSummary.lambda.iterations30.multinomial.probabilities.txt",
           row.names = 1, header = T) %>%
    ggplot(aes(x = probabilityScore,
               fill = class)) +
    geom_density() +
    theme_minimal() +
    facet_grid(class ~ .) +
    theme(legend.position = "none") +
    scale_fill_brewer(palette = "Dark2") +
    labs(x = "Fitted values of a subset of genes",
         y = "Density (fitted probabilities with lambda = 1)")
```




Density plot summarizing the distribution of how many times the CNS/NOREL/SYST sample classes were correctly predicted against other classes using the list of genes that supposedly represent patients with each type of prognosis.

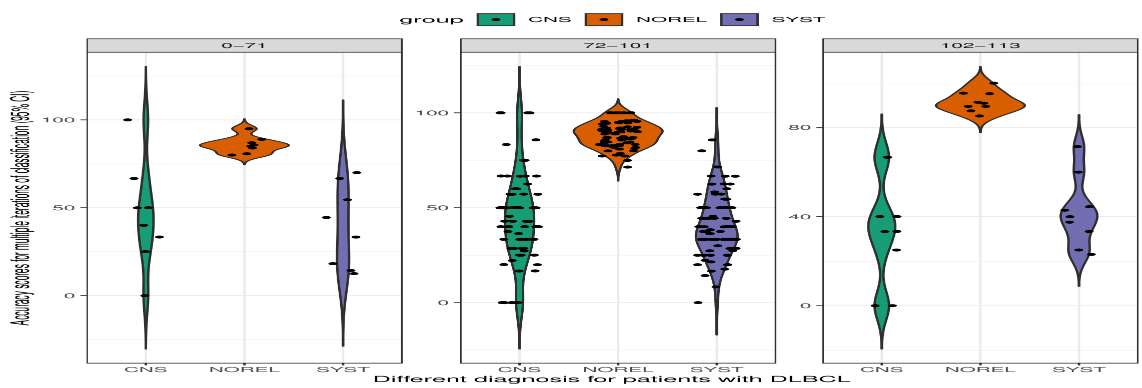
```
read.table("./data/logSummary.lambda.iterations30.multinomial.densities.txt",
           row.names = 1, header = T) %>%
    filter(classes == c("CNS", "SYST", "NOREL")) %>%
    ggplot(aes(x = probabilities,
               fill = classes)) +
    geom_density() +
    theme_minimal() +
    facet_wrap( ~ genes,
               ncol = 1,
               scales = "free") +
    theme(legend.position = "none") +
    scale_fill_brewer(palette = "Dark2") +
    labs(x = "Probability of selected genes to predict different patient classes",
         y = "Density (fitted probabilities with lambda = 1)")
```




Plot showing the accuracy of assigning a patient to its correct class (or diagnosis) based on lambda calculation for lasso regularization. Each facet represents an accuracy for multiple iterations with a specific number of genes.

```
df <- read.table("./data/logSummary.lambda.iterations30.multinomial.accuracies.txt",
```

```
                    row.names = 1, header = T)
mir <- min(df$regNgenes)
mar <- max(df$regNgenes)
q1 <- floor((mir+mar)/2.5)
q2 <- floor((mir+mar)/1.75)
df$grouped <- cut(df$regNgenes, c(0, q1, q2, mar))
levels(df$grouped) <- c(paste0(0,"-",q1),
                       paste0(q1+1,"-",q2),
                       paste0(q2+1,"-",mar))
df %>%
    ggplot(aes(x = group,
               y = accuracy,
               fill = group)) +
    geom_violin(trim = FALSE) +
    geom_jitter(shape=16, position=position_jitter(0.2)) +
    scale_fill_brewer(palette = "Dark2") +
    theme_bw() +
    labs(x = "Different diagnosis for patients with DLBCL",
         y = "Accuracy scores for multiple iterations of classification (95% CI)") +
    facet_wrap( ~ grouped,
               ncol = 3,
               scales = "free") +
    theme(legend.position = "top")
```




### 4.2 Selected machine and deep learning models

Selection of learners was based on historical efficiency of such algorithms. Also, the training of classifiers starts from simple logistic regression, naive bayes, nearest neighbors, going to more flexible models, such as trees and deep neural nets that require more hyperparameter tuning. This strategy lets assess with less bias the overfitting issues that may arise.

### 4.3 Available and tuned hyperparameters for each model

### 4.4 Machine learning performance benchmarks

Please follow up on performance metrics for classification problem by reading Sokolova 2009. ↰ Link to metrics definitions

- **Sensitivity**, is how many true cases are correctly classified to their expected class. Or **recall**, is the fraction of events where we correctly declared $i$ form all cases where the true of state of the world is $i$. $TP/(TP+FN)$
- **Specificity**, is how many wrong cases are correctly classified elsewhere. $TN/(TN+FP)$
- **Precision**, is the fraction of events where we correctly declared $i$ out of all instances where the algorithm declared $i$. $TP/(TP+FP)$
- **Accuracy**, is an overall measure that assesses the predictive model by comparing predicted classes to observed expected classes. Scores can also be shown as the area under the receiver operator curve (AUROC, 95% CI). $(TN+TP)/(TP+TN+FP+FN)$

#### 4.4.1 Creating the baseline of models performance

Machine learning models were trained only without tuning for hyperparameter optimization. Metrics generated show the raw performance of each model. ↰ Precision and recall are best for multi class learning

For this type of nominal data, classification models (not regression) are used, see Section /refsubsec:models. The performance metrics for this type of models are an accuracy score and kappa, which

**Table 1: Machine learning models**

| # | Classifiers trained | R package* | Parameters† tuned | Abbreviation |
|---|---|---|---|---|
| 1 | Naive bayes | naivebayes | laplace, usekernel, adjust | naive_bayes |
| 2 | Weighted k-Nearest Neighbors | kknn | kmax, distance, kernel | kknn |
| 3 | Penalized multinomial regression | nnet | decay | multinom |
| 4 | Random forest | randomForest | mtry | rf |
| 5 | Regularized random forest | RRF | mtry, coefReg, coefImp | RRF |
| 6 | Linear discriminant analysis (LDA) | MASS | dimen | lda2 |
| 7 | Localized LDA | klaR | k | loclda |
| 8 | Flexible discriminant analysis (FDA) | mda | degree, nprune | fda |
| 9 | Bagged FDA | mda | degree, nprune | bagFDA |
| 10 | Bagged FDA using gCV pruning | earth | degree | bagFDAGCV |
| 11 | Penalized discriminant analysis | mda | lambda | pda |
| 12 | Partial least squares | pls | ncomp | kernelpls |
| 13 | Support vector machines (SVM) with linear kernel | kernlab | C | svmLinear |
| 14 | L2 regularized SVM (dual) with linear kernel | LiblineaR | cost, loss | svmLinear3 |
| 15 | SVM with polynomial kernel | kernlab | degree, scale, C | svmPoly |
| 16 | SVM with radial basis function kernel | kernlab | sigma, C | svmRadialSigma |
| 17 | Neural network (NN) | nnet | size, decay | nnet |
| 18 | NN with feature extraction | nnet | size, decay | pcaNNet |
| 19 | Monotone multi-layer perceptron NN | monmlp | hidden1, n.ensemble | monmlp |
| 20 | Stacked autoencoder deep NN | deepnet | layer1, layer2, layer3, hidden_dropout, visible_dropout | dnn |
| 21 | Boosted logistic regression | caTools | nIter | LogitBoost |
| 22 | Stochastic gradient boosting | gbm | n.trees, interaction.depth, shrinkage, n.minobsinnode | gbm |
| 23 | Multilayer perceptron network by stochastic gradient descent | FCNN4R | size, l2reg, lambda, learn_rate, momentum, gamma, minibatchsz, repeats | mlpSGD |

* The version of each package is shared under section 4.5. Links are forwarded to the CRAN page (except those imported from Tensorflow and H2O) of each package for assessment of version, vignettes, advanced functionality, and description. † Parameters are crucial to optimize for accuracy (95% CI). Similar models have different parameters. Multi-layered neural networks are used for deep learning. In some instances, only the layer 1 is used. For such instances the classifier is considered a neural network.

takes into account the possibility of the agreement occurring by chance (the kappa score however reflects the adequate agreement). Standard error (SE in red) bars for the kappa significance per model reproducible across 10 cross-validation each repeated 5 times. Minimum and maximum accuracy thresholds are held at 95% confidence intervals.

↰ Kappa is Cohen's (unweighted) Kappa statistic averaged across the resampling results

Load standard error and deviation equations.

```
summary_SE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                      conf.interval=.95, .drop=TRUE) {

    length2 <- function (x, na.rm=FALSE) {
        if (na.rm) sum(!is.na(x))
        else       length(x)
    }
    datac <- ddply(data, groupvars, .drop=.drop,
                   .fun= function(xx, col, na.rm) {
                       c( N    = length2(xx[,col], na.rm=na.rm),
                         mean = mean   (xx[,col], na.rm=na.rm),
                         sd   = sd     (xx[,col], na.rm=na.rm)
                         )
                   },
                   measurevar,
                   na.rm
                   )
    datac <- rename(datac, c("mean"=measurevar))
    # Calculate standard error of the mean
    datac$se <- datac$sd / sqrt(datac$N)
    # Confidence interval multiplier for standard error
    # Calculate t-statistic for confidence interval:
    ciMult <- qt(conf.interval/2 + .5, datac$N-1)
    datac$ci <- datac$se * ciMult
    return(datac)
}
```

Metrics for classification performance without tuning for hyperparameter optimization. Quick comparison of statistical learning on the DLBCL data.

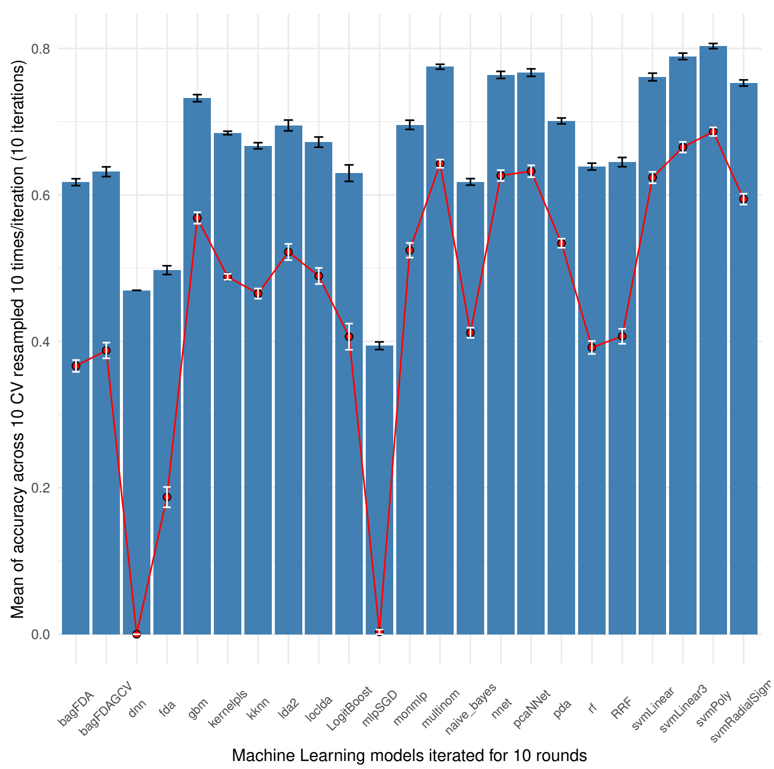
↰ Accuracy, is the true prediction rate averaged over cross-validation iterations

**Table 2: Hyperparameters for each classifier.**

| Classifier trained | Hyperparameter | Scenario A* | Scenario B** | Scenario C*** |
|---|---|---|---|---|
| Naive bayes | laplace | | | |
| | usekernel | | | |
| | adjust | | | |
| Weighted k-Nearest Neighbors | kmax | | | |
| | distance | | | |
| | kernel | | | |
| Penalized multinomial regression | decay | | | |
| Random forest | mtry | | | |
| Regularized random forest | mtry | | | |
| | coefReg | | | |
| | coefImp | | | |
| Linear discriminant analysis (LDA) | dimen | | | |
| Localized LDA | k | | | |
| Flexible discriminant analysis (FDA) | degree | | | |
| | nprune | | | |
| Bagged FDA | degree | | | |
| | nprune | | | |
| Bagged FDA using gCV pruning | degree | | | |
| Penalized discriminant analysis | lambda | | | |
| Partial least squares | ncomp | | | |
| Support vector machines (SVM) with linear kernel | C | 0.1 | | |
| L2 regularized SVM (dual) with linear kernel | cost | | | |
| | loss | | | |
| SVM with polynomial kernel | degree | | | |
| | scale | | | |
| | C | | | |
| SVM with radial basis function kernel | sigma | | | |
| | C | | | |
| Neural network (NN) | size | | | |
| | decay | | | |
| NN with feature extraction | size | | | |
| | decay | | | |
| Monotone multi-layer perceptron NN | hidden1 | | | |
| | n.ensemble | | | |
| Stacked autoencoder deep NN | layer1 | | | |
| | layer2 | | | |
| | layer3 | | | |
| | hidden_dropout | | | |
| | visible_dropout | | | |
| Boosted logistic regression | nIter | | | |
| Stochastic gradient boosting | n.trees | | | |
| | interaction.depth | | | |
| | shrinkage | | | |
| | n.minobsinnode | | | |
| Multilayer perceptron network by stochastic gradient descent | size | | | |
| | l2reg | | | |
| | lambda | | | |
| | learn_rate | | | |
| | momentum | | | |
| | gamma | | | |
| | minibatchsz | | | |
| | repeats | | | |

*Comparing samples with patients grouped by either having a central nervous system relapse (CNS), systemic relapse (SYST), no relapse (NOREL) or controls. **Comparing samples with patients grouped by either having an activated B-cell (ABC) or germinal center B-cell (GCB) diagnosis. ***Comparing samples with patients grouped by either having an tumor occurrence in lymph nodes (LN) in contrast to extra-nodal (EN) presence. The tuning parameters were iterated across a random selection grid for best tuning. Grid configurations are found on Github.

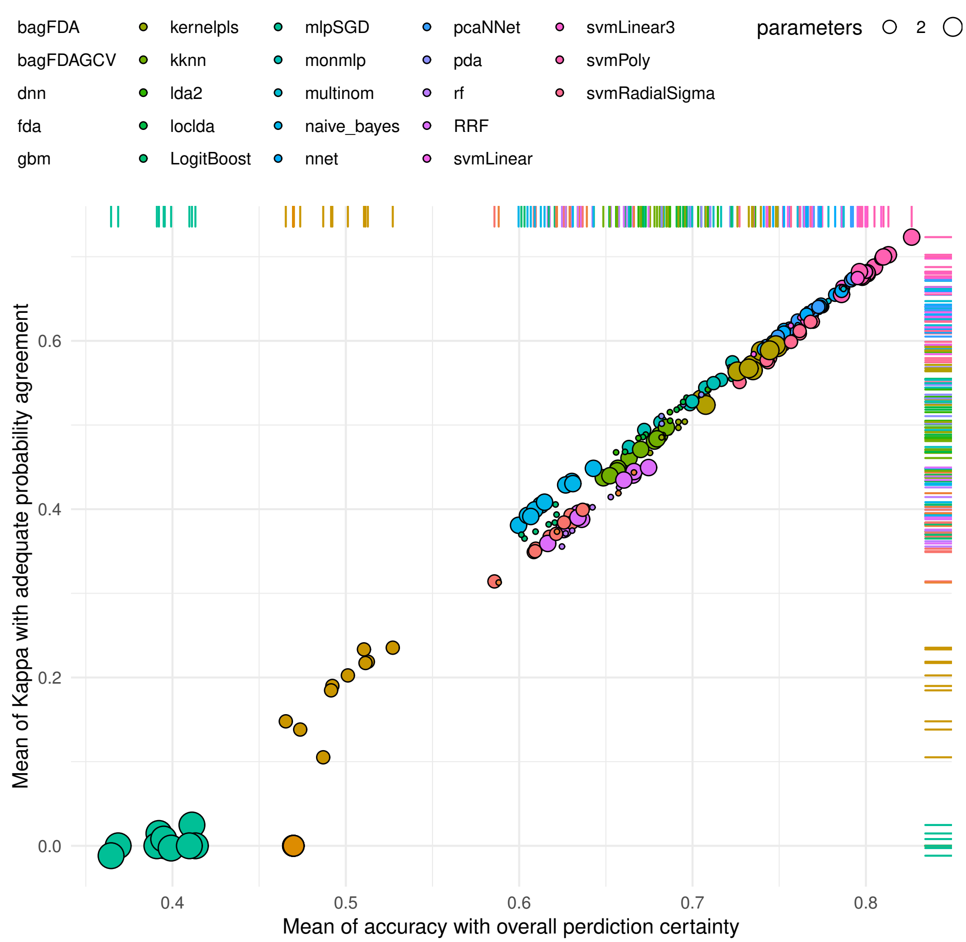
```
accuracy <- read.table("./data/log.Accuracy.performance1.multianalysis.seed14899815.409567.txt", header
kappa <- read.table("./data/log.Kappa.performance1.multianalysis.seed14899815.409567.txt", header = TRU

accuracy.se <- summary_SE(accuracy, measurevar = "Mean", groupvars = "model")
kappa.se <- summary_SE(kappa, measurevar = "Mean", groupvars = "model")

accuracy.se %>%
    ggplot(aes(x = model,
               y = Mean)) +
    geom_bar(position=position_dodge(),
             stat="identity",
             fill = "steelblue") +
    geom_errorbar(data = accuracy.se,
                  aes(ymin=Mean-se,
                      ymax=Mean+se),
                  width=.3,
                  position=position_dodge(.9)) +
    geom_line(data = kappa.se,
              aes(x = as.numeric(model),
                  y = Mean),
              color = "red") +
    geom_point(data = kappa.se,
               size=2, shape=21, fill="red") +
    geom_errorbar(data = kappa.se,
                  aes(ymin=Mean-se,
                      ymax=Mean+se),
                  width=.25,
                  position=position_dodge(.9),
                  color = "white") +
    theme_minimal() +
    ylab("Mean of accuracy across 10 CV resampled 10 times/iteration (10 iterations)") +
    xlab("Machine Learning models iterated for 10 rounds") +
    theme(legend.position = "top") +
    guides(fill=guide_legend(title="Number of parameters per model")) +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
                                     size = 8))
```




360

Kappa (vertical axis) and accuracy (horizontal axis) calculated from the performance tests of machine learning models. The higher kappa is the stronger agreement for a prediction and classification. Although, this chart shows little correlation between prediction accuracy and outcome of each model, its important to highlight the number of parameters tune for each classifier. Because, based on a recent paper from Google in Nature, Rajkomar 2018, deep learning models with many layers and hyperparameters perform well enough as a regularized logistic regression model (Supplementary text). In the paper however, little was mentioned about this discrepancy in the published results. Lastly, it remains important to address the value of the parameters that are manipulated and the amount of time spent training the model based on the tuning process.

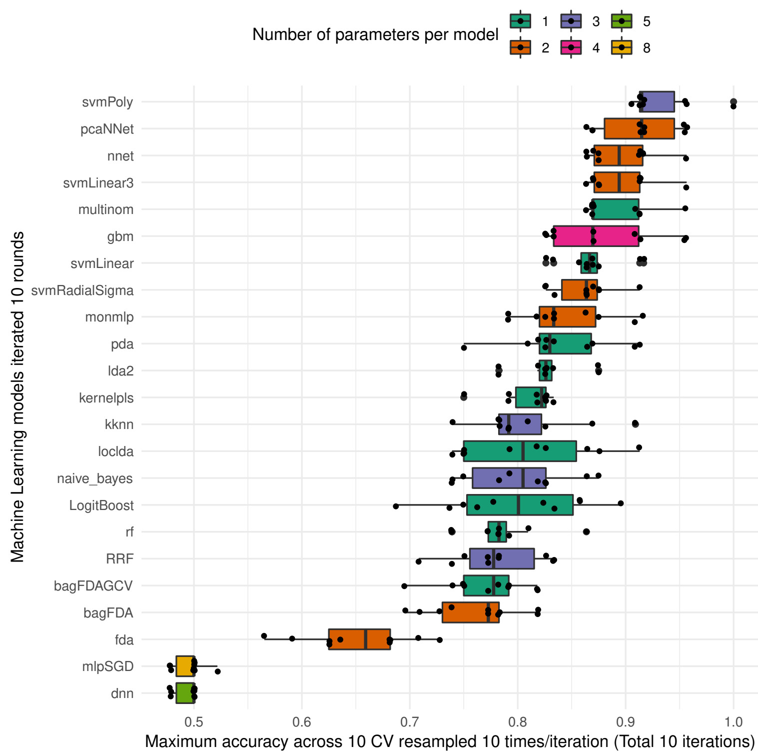
```
data.frame(accuracy, kappa) %>%
    ggplot(aes(x = Mean,
               y = Mean.1,
               fill = model)) +
    geom_point(aes(size=parameters), shape=21) +
    geom_rug(aes(stat = "identity",
                 color = as.character(model)),
             sides = "tr",
             show.legend = F) +
    theme_minimal() +
    ylab("Mean of Kappa with adequate probability agreement") +
    xlab("Mean of accuracy with overall perdiction certainty") +
    theme(legend.position = "top")

Warning:  Ignoring unknown aesthetics:  stat
```




Maximum accuracy (95% CI) registered for machine learning fitting for all classification groups. Iterations for reproducibility were executed 10 times.

```
accuracy %>%
```

```r
ggplot(aes(x = reorder(model, Max.),
           y = Max.,
           fill = as.character(parameters))) +
  geom_boxplot() +
  geom_jitter(shape=16, position=position_jitter(0.2), cex = 1.5) +
  coord_flip() +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  ylab("Maximum accuracy across 10 CV resampled 10 times/iteration (Total 10 iterations)") +
  xlab("Machine Learning models iterated 10 rounds") +
  theme(legend.position = "top") +
  guides(fill=guide_legend(title="Number of parameters per model"))
```




#### 4.4.2 Models performance with hyperparameter tuning

Compared to the baseline, the parameters available for each learner should increase its performance at predicting each expected outcome. Tuning these hyperparameters will leverage the results with increased accuracy.

#### 4.4.3 Prediction accuracy of each classifier at optimal parameters

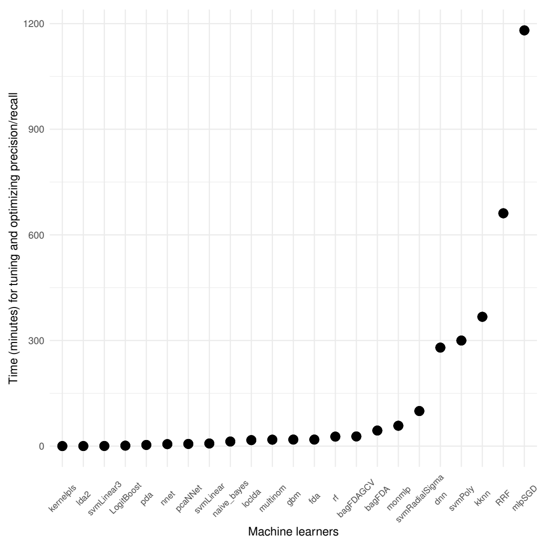
Final report of the actual accuracy (95% CI) for each machine learning model from comparing predicted values and expected outcomes.

↰ Data are retrieved from Confusion Matrix

How long (seconds) a statistical learner requires to optimize the hyperparameters and gets the highest significant accuracy on expected data.

```r
df <- read.table("./data/log.performance3.full.hyperTuning.seed14899815.409567.txt", header=T)
```

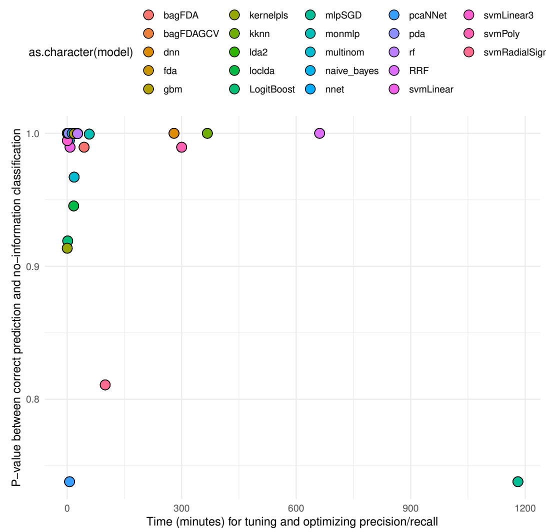
```r
df$group <- gsub("[0-9]", "", df$group)

df %>%
    ggplot(aes(x = reorder(model, durationMinutes),
               y = durationMinutes)) +
    geom_point(position = position_dodge(),
               stat = "identity",
               size=4, shape=16) +
    theme_minimal() +
    xlab("Machine learners") +
    ylab("Time (minutes) for tuning and optimizing precision/recall") +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
                                     size = 8))

Warning:  Width not defined.  Set with `position_dodge(width = ?)`
```




How is time training a model deliver on the significance of its accuracy? The p-value evaluates whether the overall accuracy rate is greater than the rate of the largest class. Proportions between classes (if one group of samples is larger than an other) is also considered in the hypothesis testing.

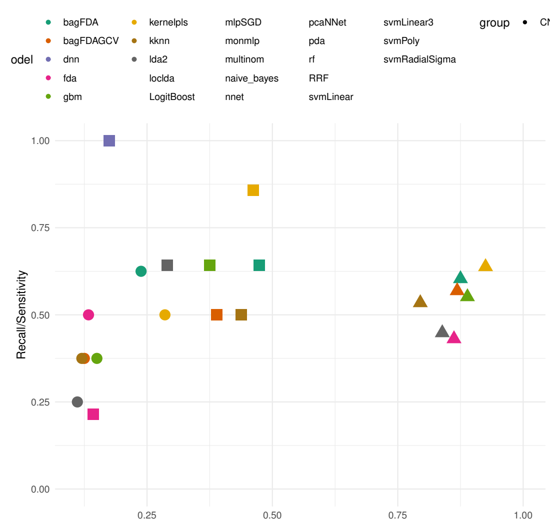
↰ Link documentation Section 17.2

```r
df %>%
    ggplot(aes(y = accuracyPval,
               x= durationMinutes,
               fill = as.character(model))) +
    geom_point(size = 4,
               shape = 21) +
    theme_minimal() +
    xlab("Time (minutes) for tuning and optimizing precision/recall") +
    ylab("P-value between correct prediction and no-information classification") +
    theme(legend.position = "top")
```




Precision versus recall across all sample groups for a multi-class classification. ↰True/False Positives/Negatives

```
df %>%
    ggplot(aes(x = Precision,
               y = Recall,
               group = as.character(group),
               na.rm = T)) +
    geom_point(aes(size = 4,
               shape = group,
               color = model)) +
    theme_minimal() +
    xlab("Precision of how many correct classes were called") +
    ylab("Recall/Sensitivity") +
    scale_color_brewer(palette = "Dark2") +
    theme(legend.position = "top")

Warning in RColorBrewer::brewer.pal(n, pal):  n too large, allowed maximum for
palette Dark2 is 8
Returning the palette you asked for with that many colors
Warning:  Removed 47 rows containing missing values (geom_point).
```

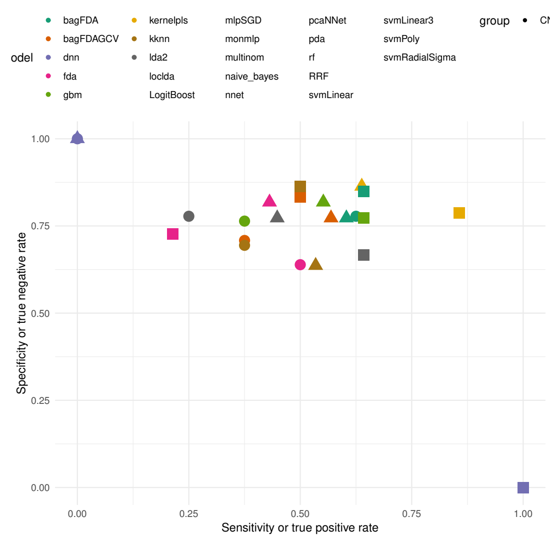



Specificity and sensitivity across all sample groups.

```
df %>%
```

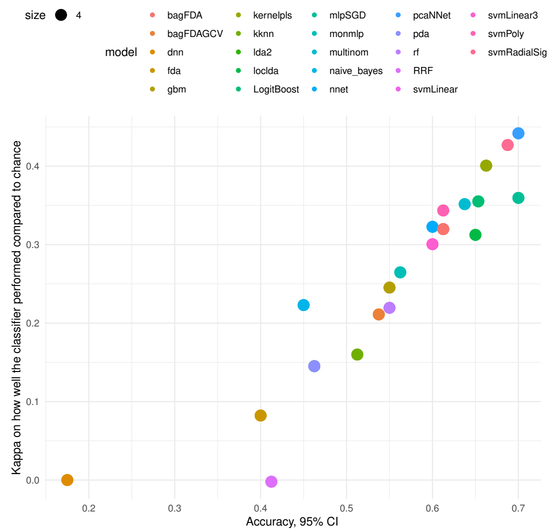
```r
    ggplot(aes(x = Sensitivity,
               y = Specificity,
               group = as.character(group))) +
    geom_point(aes(size = 4,
               shape = group,
               color = model)) +
    theme_minimal() +
    xlab("Sensitivity or true positive rate") +
    ylab("Specificity or true negative rate") +
    scale_color_brewer(palette = "Dark2") +
    theme(legend.position = "top")
```

```
Warning in RColorBrewer::brewer.pal(n, pal):  n too large, allowed maximum for
palette Dark2 is 8
Returning the palette you asked for with that many colors
Warning:  Removed 45 rows containing missing values (geom_point).
```



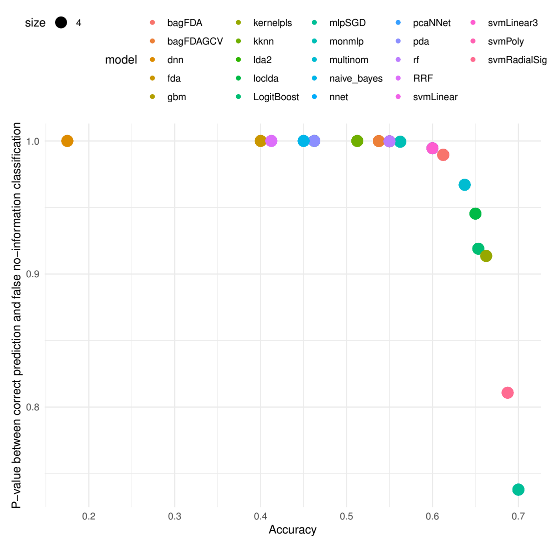

Accuracy and Kappa across all sample groups.

```r
df %>%
    ggplot(aes(x = accuracy,
               y = kappa)) +
    geom_point(aes(size = 4,
               color = model)) +
    theme_minimal() +
    xlab("Accuracy, 95% CI") +
    ylab("Kappa on how well the classifier performed compared to chance") +
    theme(legend.position = "top")
```



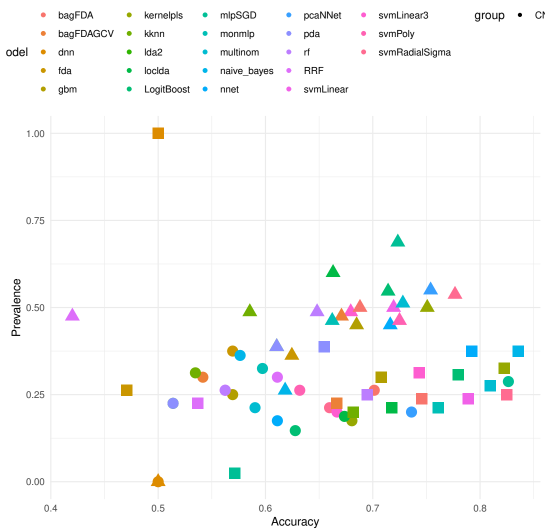

Accuracy versus the p-value of each classification. The p-value is a hypothesis test between predicting expected samples and the probability that the classification is biased by disproportionate class sizes (one group of samples is larger than an other).

```
df %>%
    ggplot(aes(x = accuracy,
               y = accuracyPval)) +
    geom_point(aes(size = 4,
               color = model)) +
    theme_minimal() +
    xlab("Accuracy") +
    ylab("P-value between correct prediction and false no-information classification") +
    theme(legend.position = "top")
```




Prevalence of cases for each classifier. Were the classes perfectly balanced? A positive predictive score is similar to precision while accounting for disproportionality of the classes.

```
df %>%
    ggplot(aes(x = Balanced.Accuracy,
               y = Detection.Prevalence,
               group = as.character(group))) +
    geom_point(aes(size = 4,
               color = model,
               shape = group)) +
    theme_minimal() +
    xlab("Accuracy") +
    ylab("Prevalence") +
    theme(legend.position = "top")
```




## 4.5 Version of machine learning models

↰ Version of R packages used for their algorithmic implementation of machine learning models

# 5 System Information

The version number of R and packages loaded for generating the vignette were:

```
###save(list=ls(pattern=".*|.*"),file="PD.Rdata")
```

```r
sessionInfo()
```

```
R version 3.4.4 (2018-03-15)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: elementary OS 0.4.1 Loki

Matrix products: default
BLAS: /usr/lib/libblas/libblas.so.3.6.0
LAPACK: /usr/lib/lapack/liblapack.so.3.6.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
 [1] bindrcpp_0.2        paletteer_0.1.0     plyr_1.8.4
 [4] finalfit_0.7.4      Hmisc_4.1-1         Formula_1.2-3
 [7] survival_2.42-3     brotools_0.2        scales_0.5.0
[10] DescTools_0.99.23   igraph_1.1.2        tidyr_0.8.0
[13] dplyr_0.7.4         ggplot2_3.0.0       latticeExtra_0.6-28
[16] RColorBrewer_1.1-2  lattice_0.20-35     gdata_2.18.0
[19] knitr_1.20

loaded via a namespace (and not attached):
 [1] gtools_3.5.0      tidyselect_0.2.4  reshape2_1.4.3
 [4] purrr_0.2.4       splines_3.4.4     colorspace_1.3-2
 [7] expm_0.999-2      htmltools_0.3.6   base64enc_0.1-3
[10] rlang_0.2.0       pillar_1.1.0      foreign_0.8-70
[13] glue_1.2.0        withr_2.0.0       bindr_0.1
[16] stringr_1.3.1     munsell_0.4.3     gtable_0.2.0
[19] htmlwidgets_1.2   mvtnorm_1.0-7     evaluate_0.10.1
[22] labeling_0.3      manipulate_1.0.1  htmlTable_1.11.2
[25] highr_0.6         Rcpp_0.12.16      acepack_1.4.1
[28] backports_1.1.1   checkmate_1.8.5   gridExtra_2.3
[31] digest_0.6.12     stringi_1.2.2     grid_3.4.4
[34] tools_3.4.4       magrittr_1.5      lazyeval_0.2.1
[37] tibble_1.4.2      cluster_2.0.7-1   pkgconfig_2.0.1
[40] MASS_7.3-47       Matrix_1.2-11     data.table_1.11.2
[43] rstudioapi_0.7    assertthat_0.2.0  R6_2.2.2
[46] boot_1.3-20       rpart_4.1-13      nnet_7.3-12
[49] compiler_3.4.4
```