Descriptive analysis

Sleiman Bassim, PhD

May 31, 2018

Contents

Loaded functions.

↰ Project started Dec 10 2017, updated May 31, 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')
lapply(pkgs, require, character.only = TRUE)
```

1 Data structure

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) %>%
```

```
    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 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.

```
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
```

	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)

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

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 %>%
```

	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)

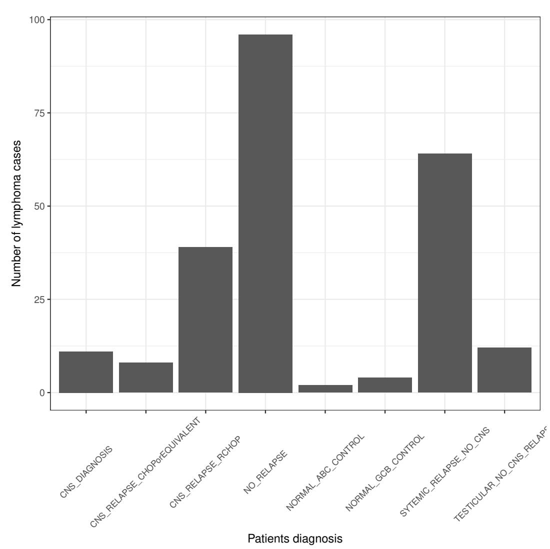
```
    select(Prediction, ABClassify, ABCScore) %>%
    summary
```

```
 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.

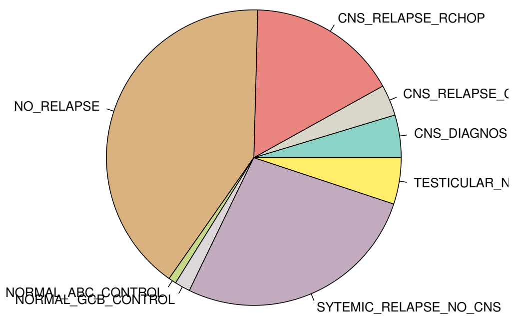
```
metadata %>%
    select(GROUP) %>%
    ggplot(aes(x = GROUP)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of lymphoma cases",
         x = "Patients diagnosis") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
  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

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



Or as a pie chart.

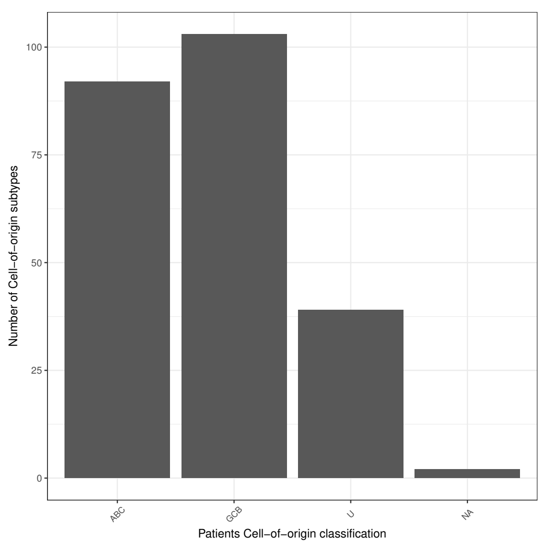
```
palette.pies <- brewer.pal(12, name = "Set3")
palette.pies.adj <- colorRampPalette(palette.pies)(length(unique(metadata$GROUP)))
pie(table(metadata$GROUP), col=palette.pies.adj)
```



Distribution of samples with different cells of origin subtypes.

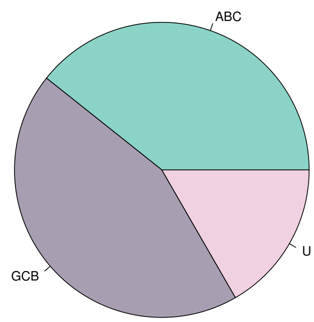
```
metadata %>%
    select(Prediction) %>%
    ggplot(aes(x = Prediction)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of Cell-of-origin subtypes",
         x = "Patients Cell-of-origin classification") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
            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

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



Or as pie chart.

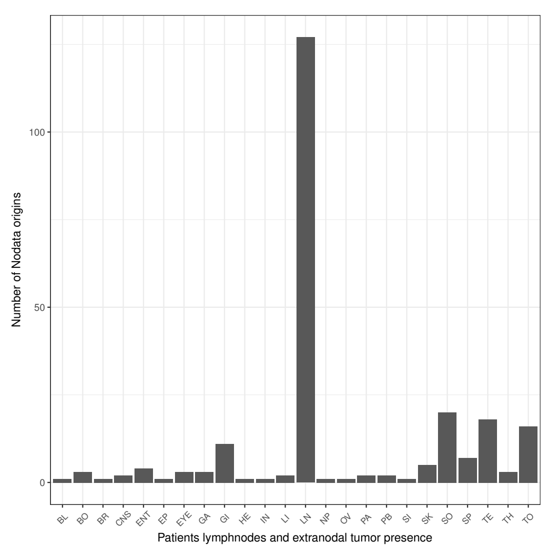
```
palette.pies <- brewer.pal(12, name = "Set3")
palette.pies.adj <- colorRampPalette(palette.pies)(length(unique(metadata$Prediction)))
pie(table(metadata$Prediction), col=palette.pies.adj)
```



Distribution of samples with different lymphnodes and extranodal cancer metastasis.

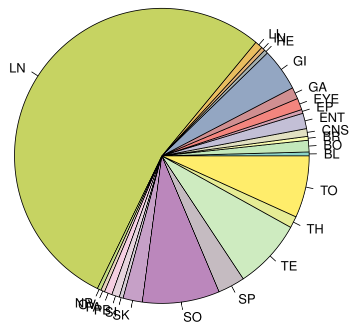
```
par(mfrow=c(2,2))
metadata %>%
    select(SITE) %>%
    ggplot(aes(x = SITE)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of Nodata origins",
         x = "Patients lymphnodes and extranodal tumor presence") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
                                     size = 8))

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



Or as a pie chart.

```
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.90800.txt", sep = "\t", header = T) %>%
    select(Design, Model, Bthreshold, adjPval, Category, Parameter, Transcripts) %>%
    filter(Category == "total")
summary(expression)

                     Design                                Model
 CNSvsNOREL_ABC         : 54   systemicRelapse               : 54
 CNSvsNOREL_GCB         : 54   systemicRelapseCOOclasses     :162
 CNSvsSYST_ABC          : 54   systemicRelapseCOOprediction:162
 CNSvsSYST_GCB          : 54   systemicRelapseCOOscores      :162
 diffCNSvsNOREL_ABCvsGCB: 54   systemicRelapseNodes          :162
 diffCNSvsSYST_ABCvsGCB : 54
 (Other)                :378
   Bthreshold        adjPval        Category     Parameter
 Min.   :-2.00   Min.   :0.049   down :  0   adjpval:234
 1st Qu.:-1.00   1st Qu.:0.049   total:702   avgex  :234
 Median : 0.25   Median :0.049   up   :  0   bval   :234
 Mean   : 0.00   Mean   :0.049
 3rd Qu.: 1.00   3rd Qu.:0.049
 Max.   : 1.50   Max.   :0.049

  Transcripts
 Min.   :    0
 1st Qu.:    2
 Median :   46
 Mean   :  580
 3rd Qu.:  463
 Max.   :10578
```

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

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

  Bthreshold Transcripts.1 Transcripts.2
1       -2.0             0         10578
2       -1.0             0          6448
3        0.0             0          3618
4        0.5             0          2688
5        1.0             0          1976
6        1.5             0          1429
```

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).

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

                         Model Transcripts.1 Transcripts.2
1               systemicRelapse             0          4938
2     systemicRelapseCOOclasses             0         10578
3 systemicRelapseCOOprediction             0         10578
4      systemicRelapseCOOscores             0         10578
5          systemicRelapseNodes             0          6609
```

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

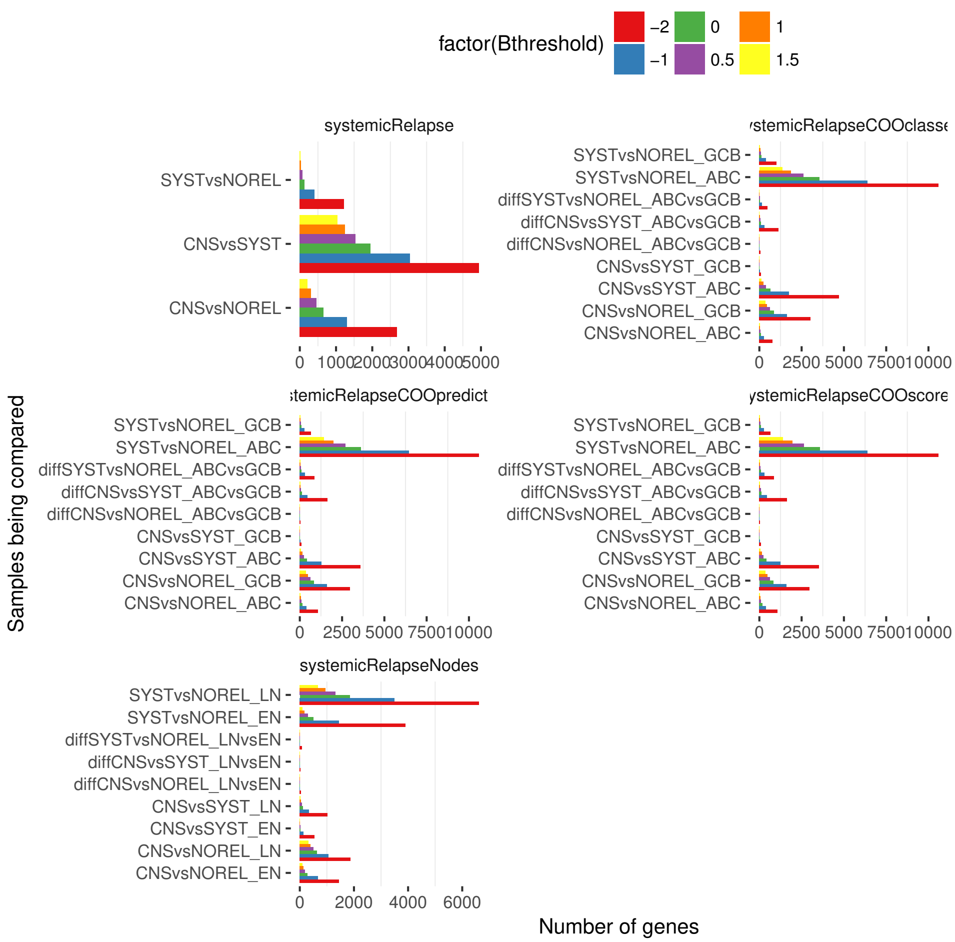
```r
aggregate( Transcripts ~ Design, data=expression, FUN=range)

                      Design Transcripts.1 Transcripts.2
1                 CNSvsNOREL           116          2678
2             CNSvsNOREL_ABC             2          1082
3              CNSvsNOREL_EN            51          1442
4             CNSvsNOREL_GCB           130          3019
5              CNSvsNOREL_LN           125          1873
6                  CNSvsSYST           441          4938
7              CNSvsSYST_ABC             2          4691
8               CNSvsSYST_EN             3           547
9              CNSvsSYST_GCB             0            98
10              CNSvsSYST_LN             0          1014
11  diffCNSvsNOREL_ABCvsGCB             0            58
12    diffCNSvsNOREL_LNvsEN             0            37
13   diffCNSvsSYST_ABCvsGCB             1          1640
14     diffCNSvsSYST_LNvsEN             0            23
15 diffSYSTvsNOREL_ABCvsGCB             0           868
16   diffSYSTvsNOREL_LNvsEN             0            85
17               SYSTvsNOREL             0          1214
18           SYSTvsNOREL_ABC           704         10578
19            SYSTvsNOREL_EN            35          3907
20           SYSTvsNOREL_GCB             2           994
21            SYSTvsNOREL_LN           295          6609
```

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

```r
expression %>%
```

```r
ggplot(aes(
    x = Design,
    y = Transcripts,
    fill = factor(Bthreshold))) +
theme_bw() +
geom_bar(stat = "identity",
         position = "dodge") +
coord_flip() +
facet_wrap( ~ Model,
          ncol = 2,
           scales = "free") +
scale_fill_brewer(type = "qual", palette = 6) +
labs(x = "Samples being compared",
     y = "Number of genes") +
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 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).

↰ σ^2 is the average of the squared differences from the μ

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

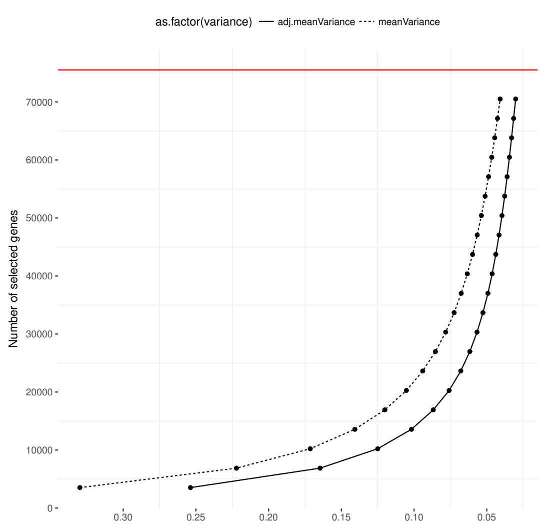
2.1.1 Variance optimization for each array

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) %>%
    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"))
```

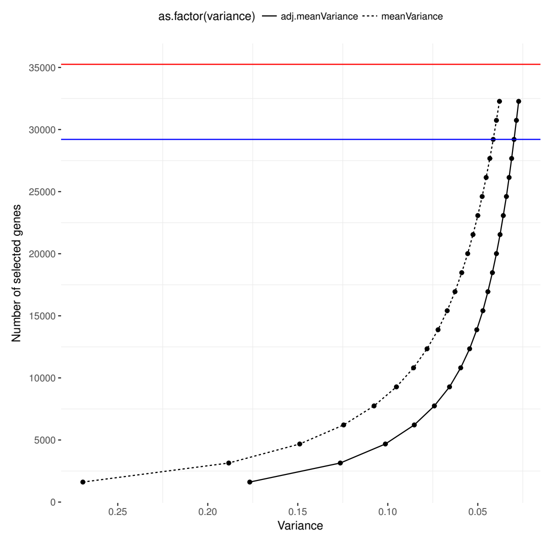


Same plot description as above however we removed ncRNA which account for 53.32% of the probes. The total number of transcripts is now 35,253 (46%, red horizontal line). The blue horizontal line represents the threshold that was selected for subsequent analysis.

↰ 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

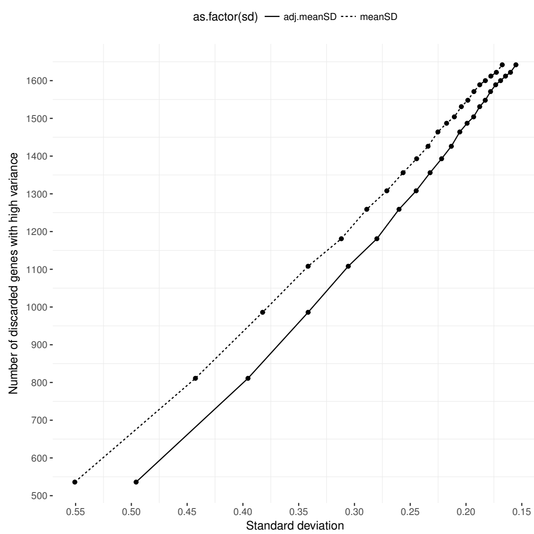
The spread of the gene expression scores is dependent on their variance, their deviation from each array's mean (population mean). By removing potentially noisy expressions we are reducing the spread of the arrays numbers, hence improving recognition of rare gene regulations. Below, the plot shows how 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.

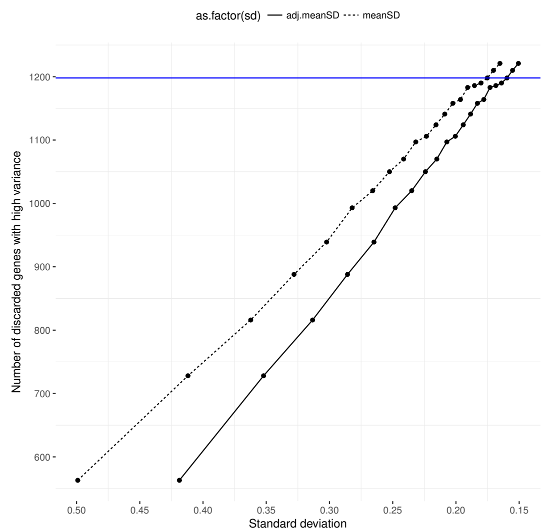
```r
read.table("./data/summary.139102.adjusted.means.subsetting.txt", header = T) %>%
```

```
    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"))
```



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

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

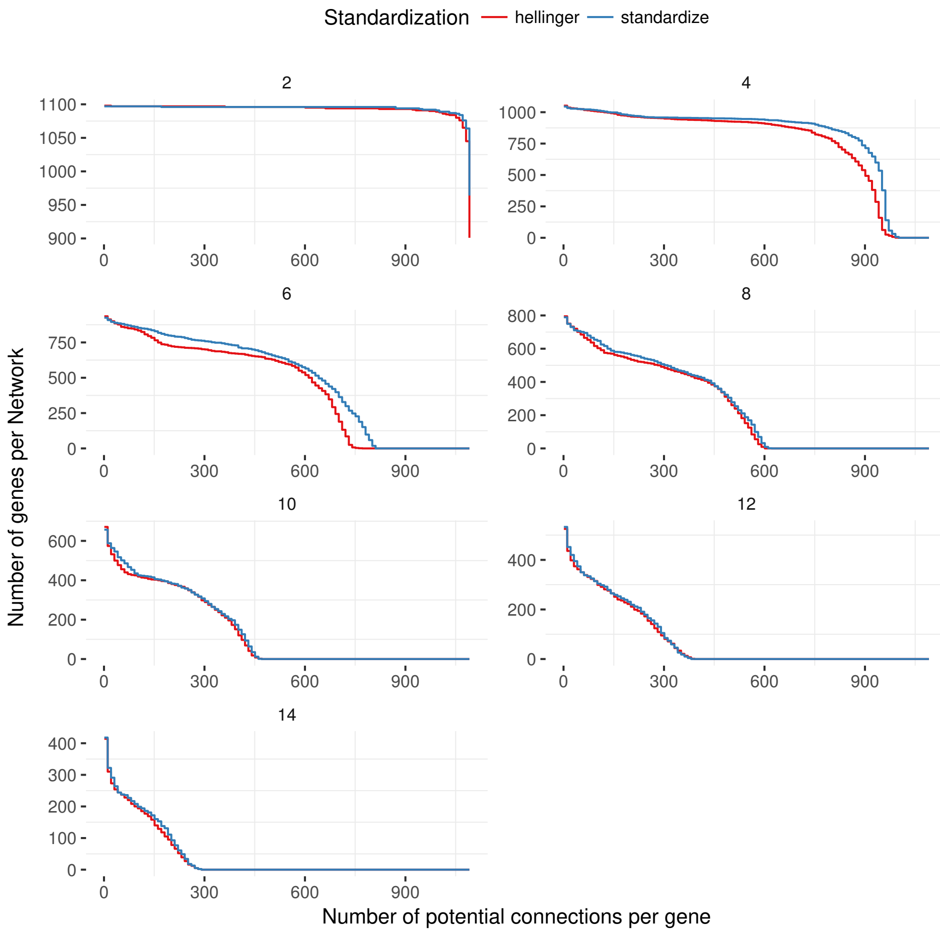
↰Effect of correlation methods is seen on module content

```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

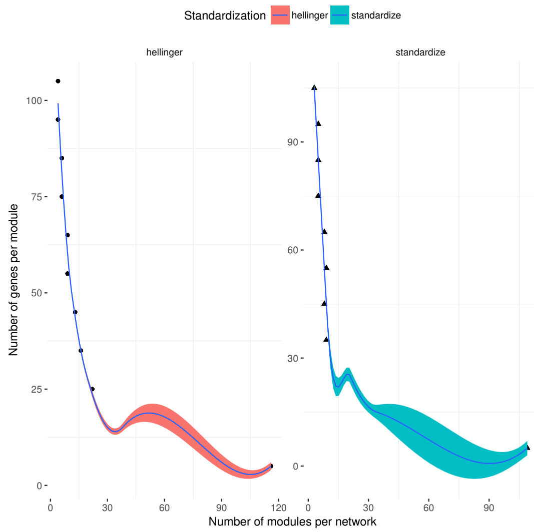
```r
ns %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free") +
    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)

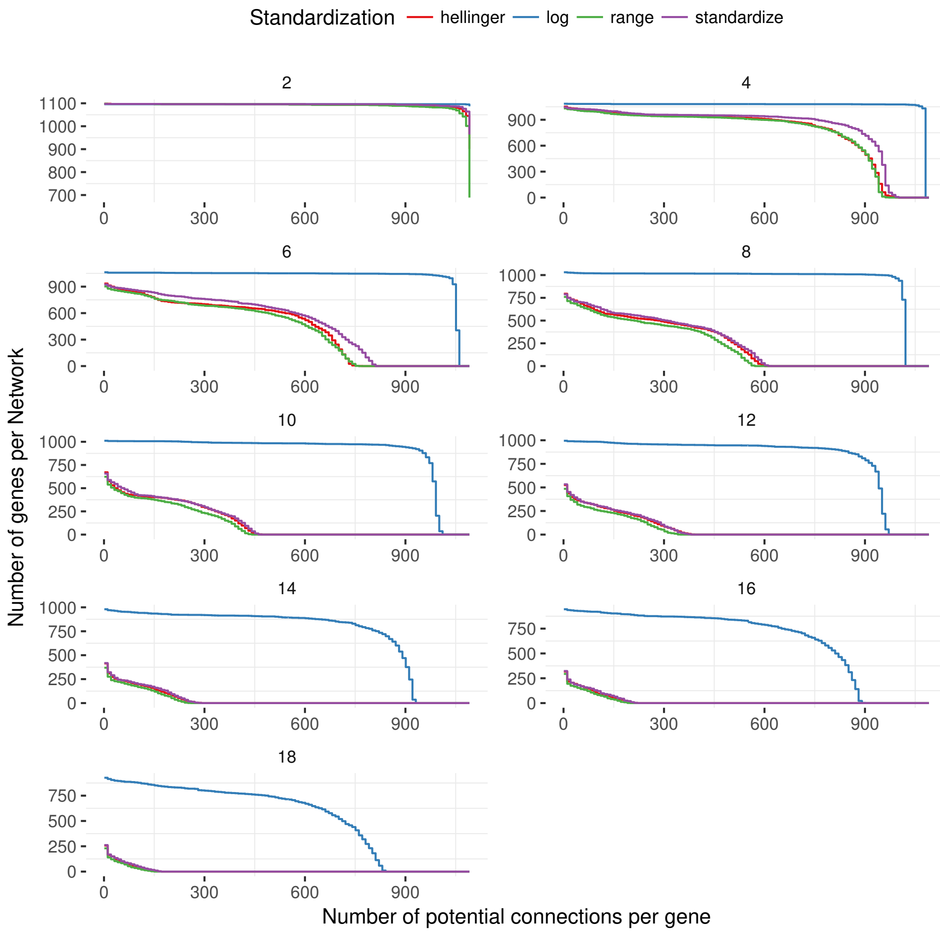
Thresholds based on the Empirical Bayes approach to rank genes and determine if a gene is significantly expressed. Limma implementation.

- **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

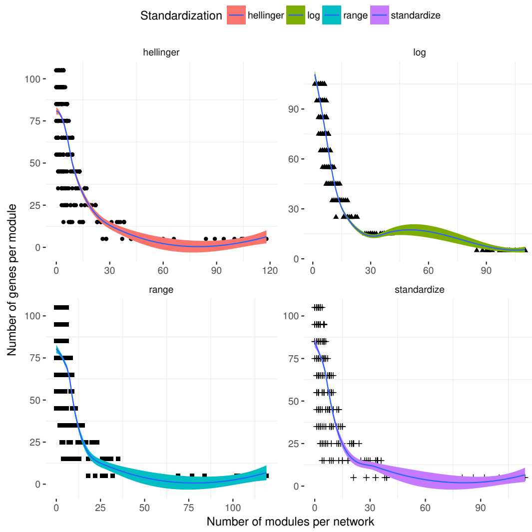
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
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.

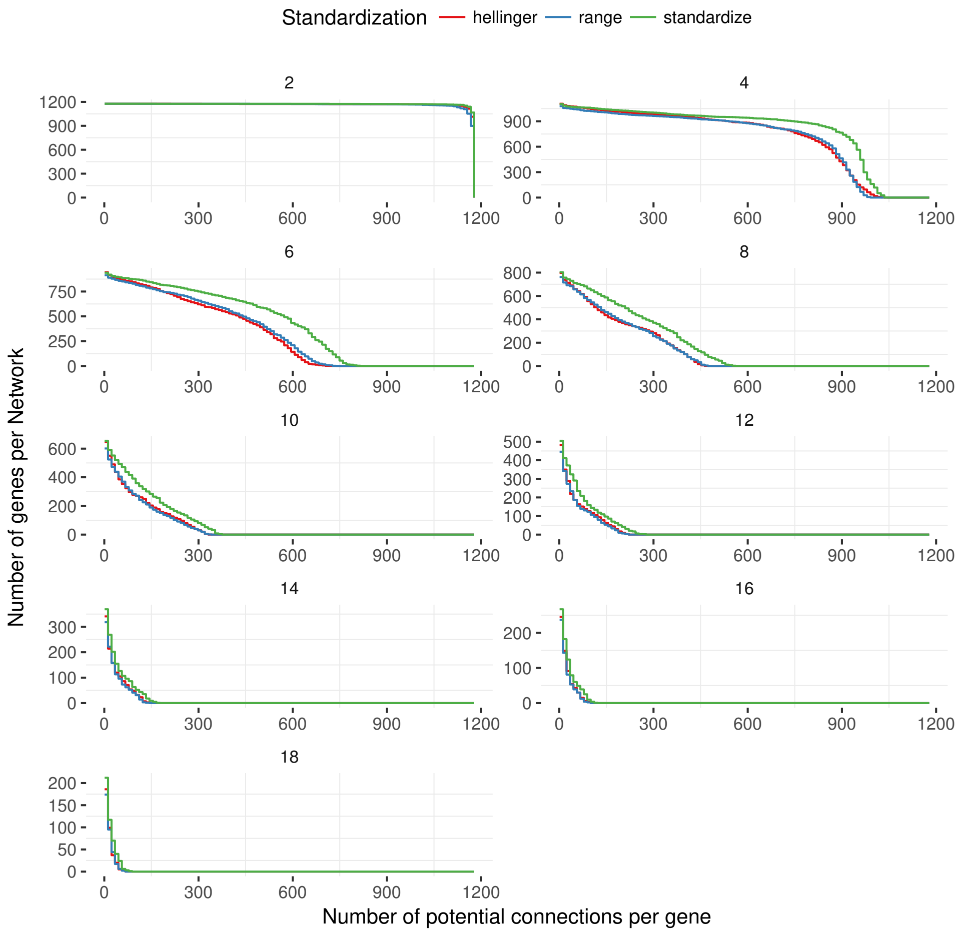
```
read.table("./data/modules.summary.104859.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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

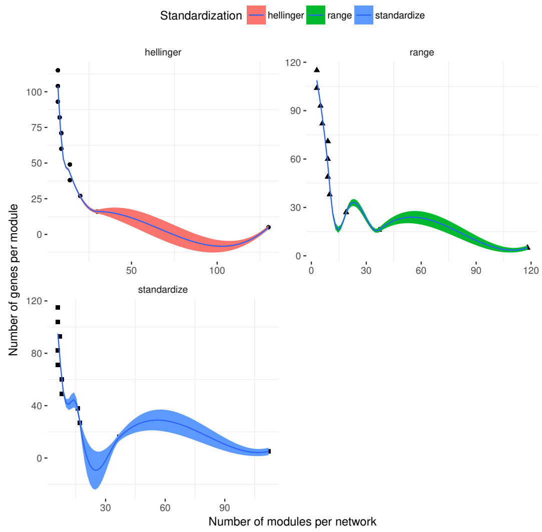
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

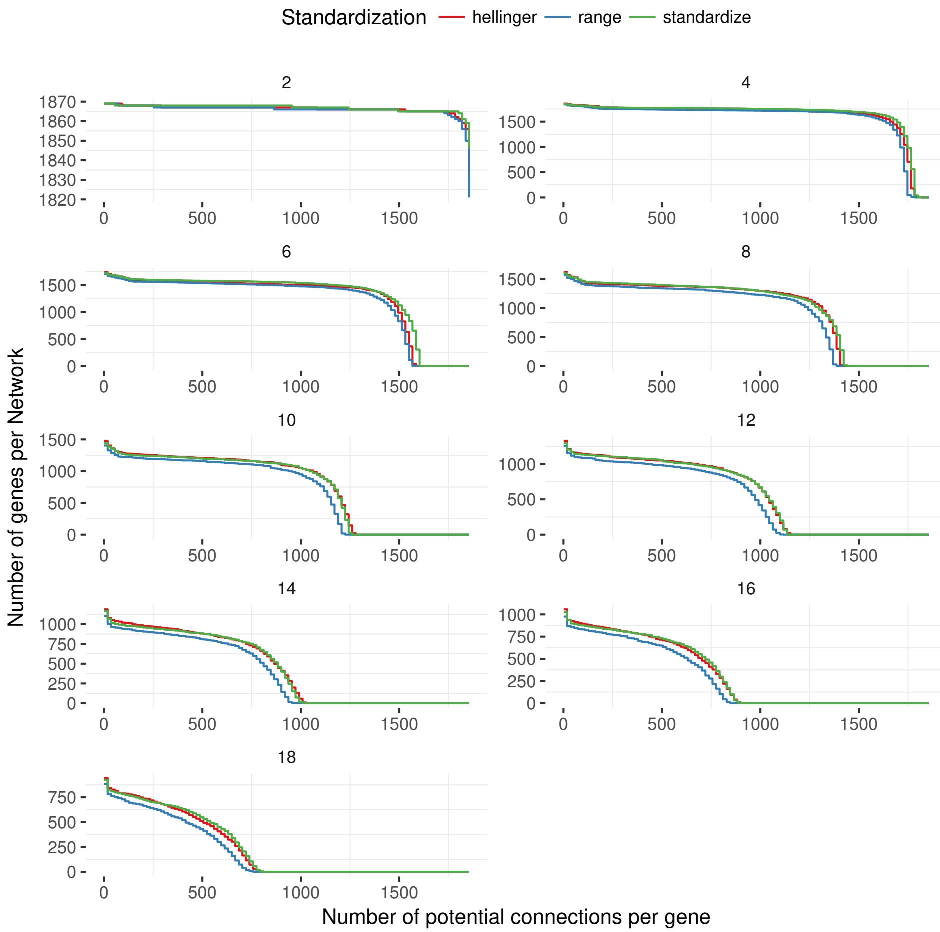
```
read.table("./data/modules.summary.114018.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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

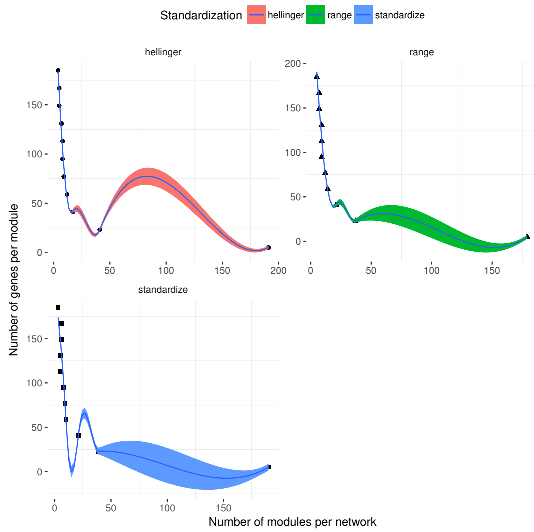
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

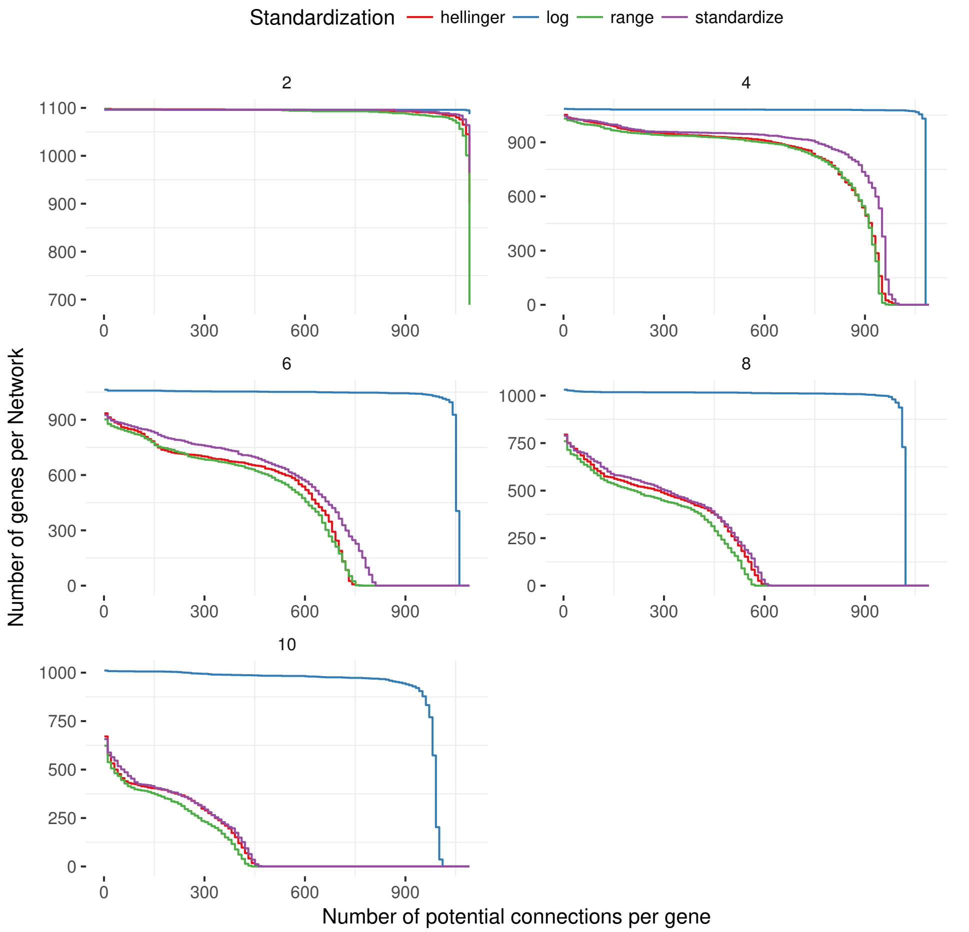
With pearson, we can only raise the data to power 10. All are discarded after 10.

- **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.

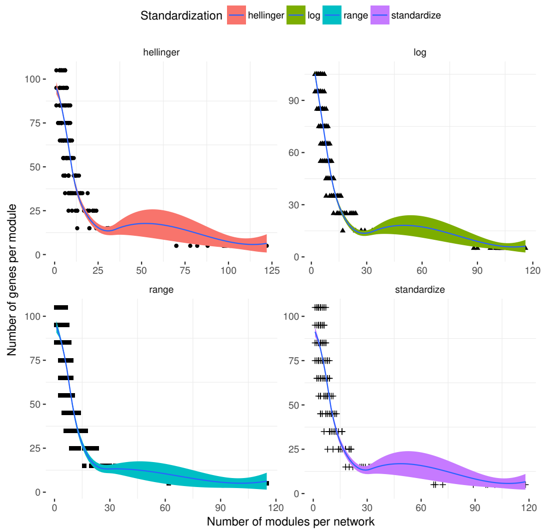
```
read.table("./data/networks.summary.104862.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               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.

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

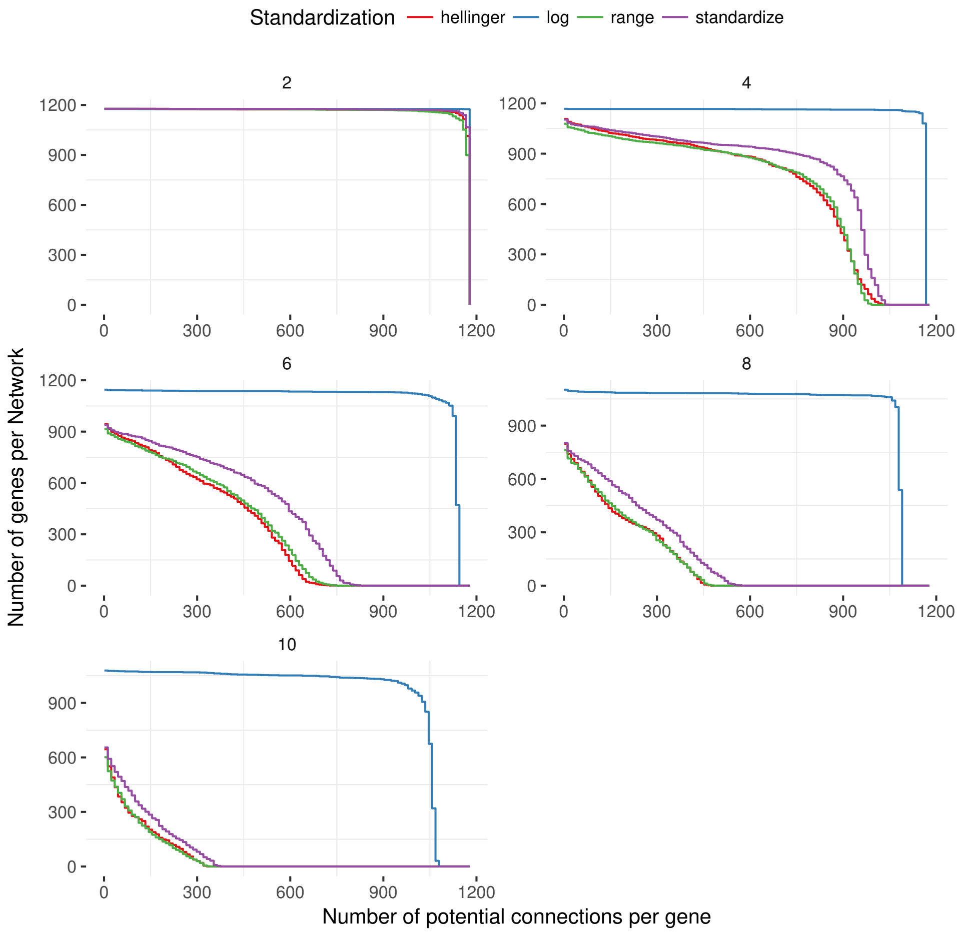
```
read.table("./data/modules.summary.104862.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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)
```



3.2.2 Relapsed versus no CNS relapsed cases

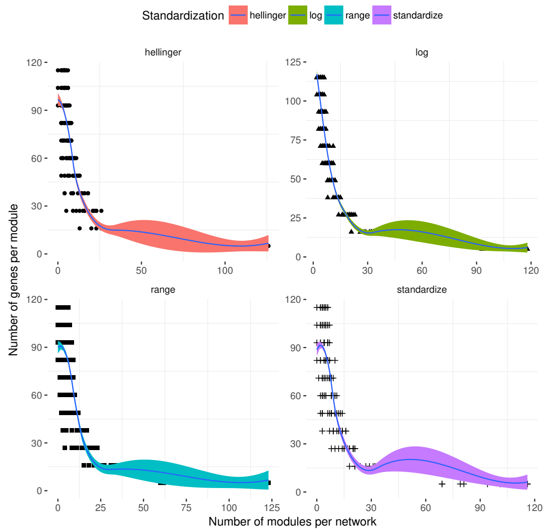
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

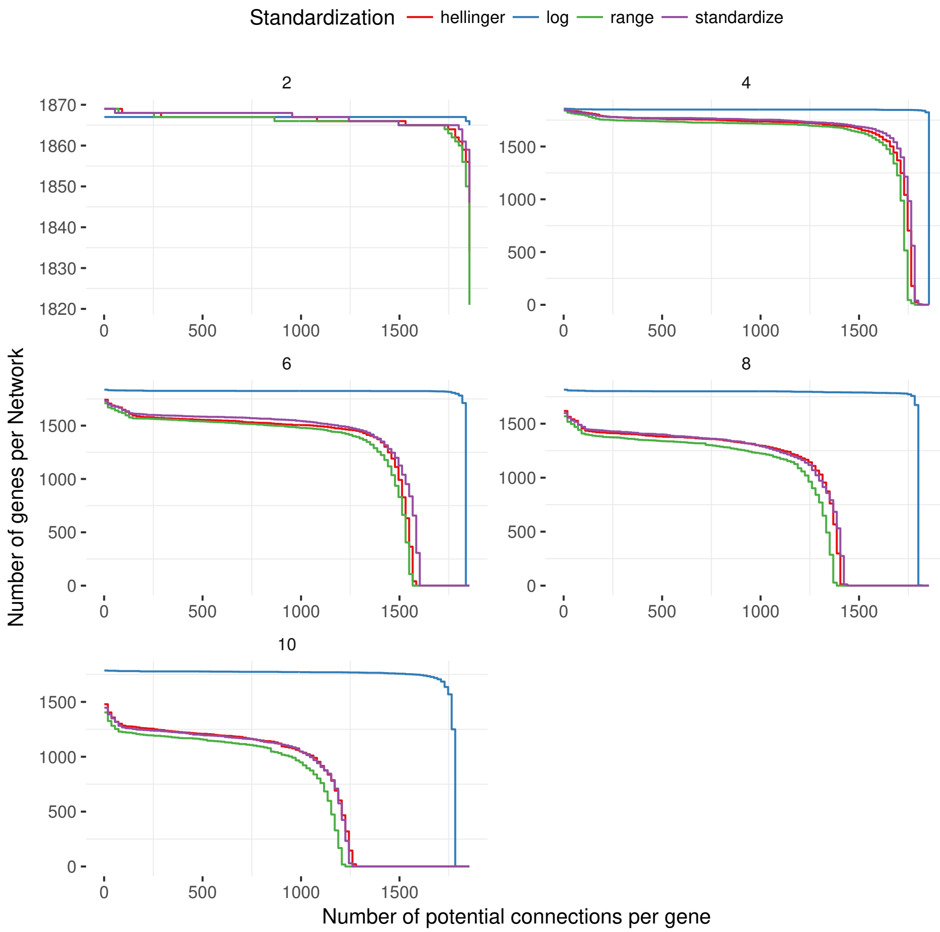
```
read.table("./data/modules.summary.104863.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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

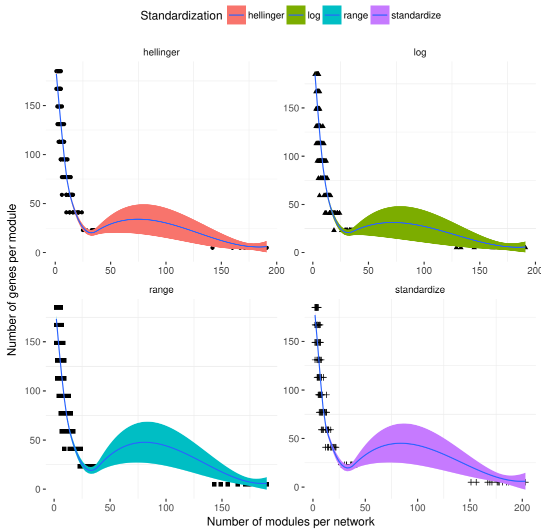
Genetic networks from differentially expressed genes selected by comparing sample cases with cell of origin classification based on ABC or GCB subtypes.

```
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.

```r
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.

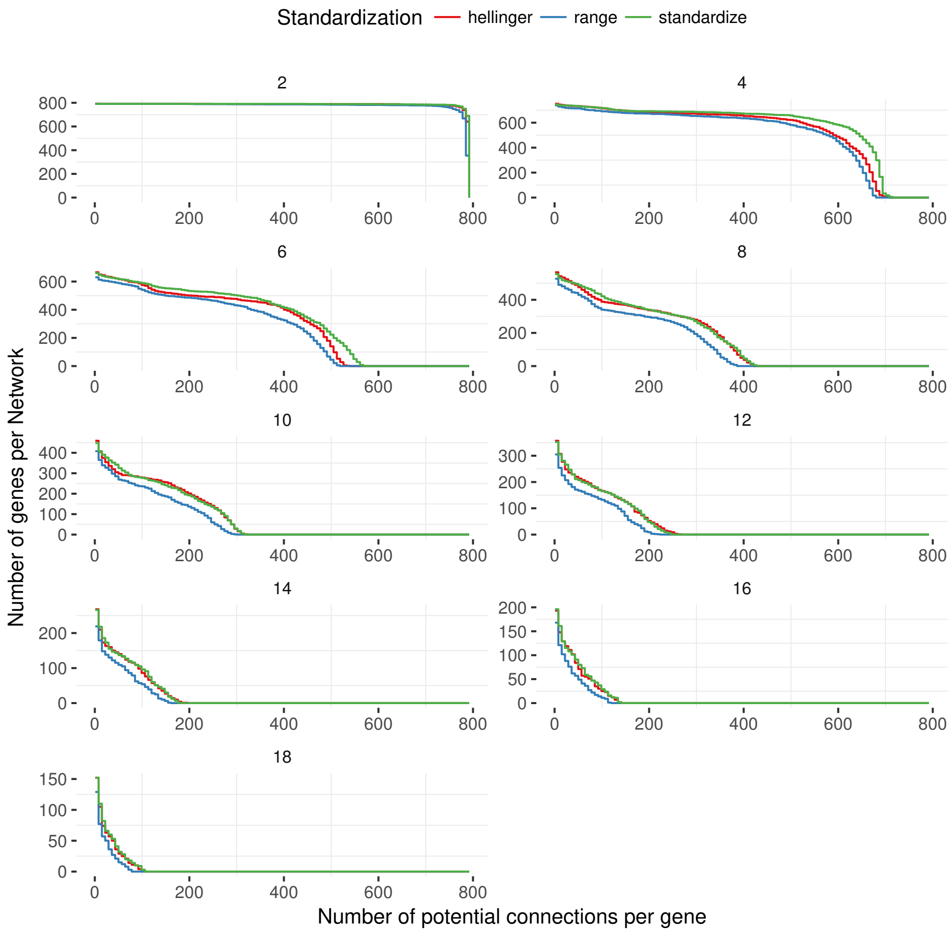
↰Same analysis with more stringent parameters

- **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

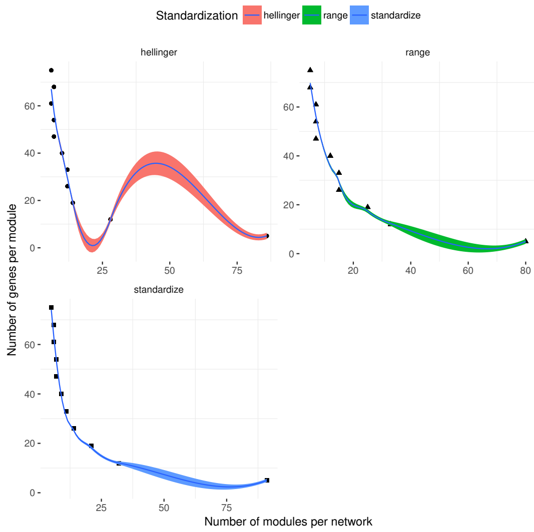
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
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.

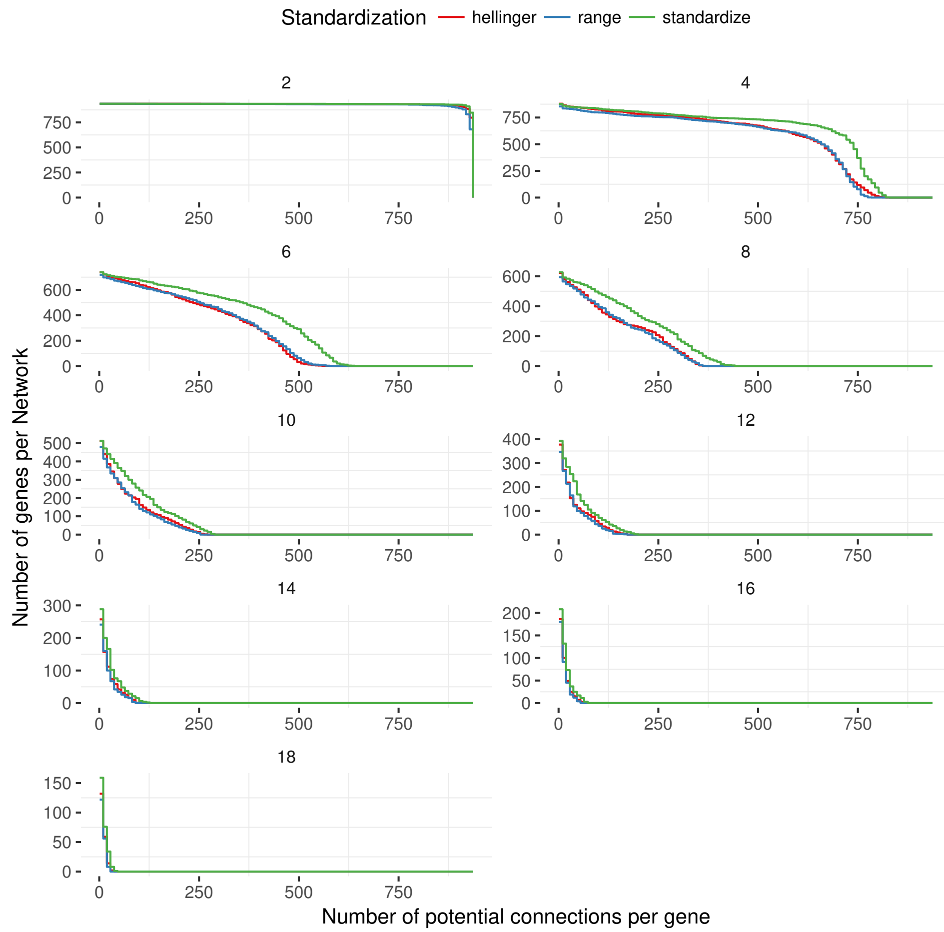
```
read.table("./data/modules.summary.119759.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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

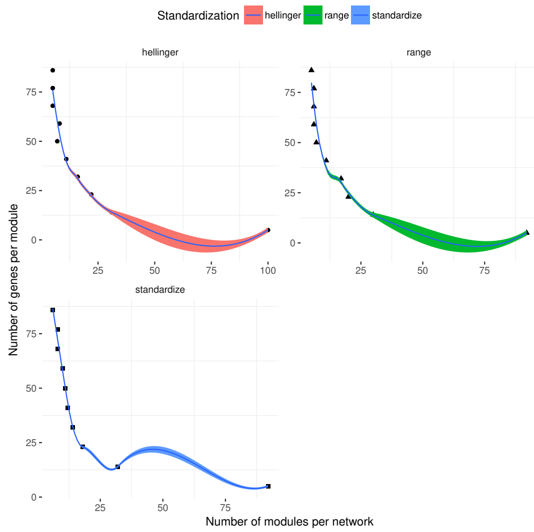
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

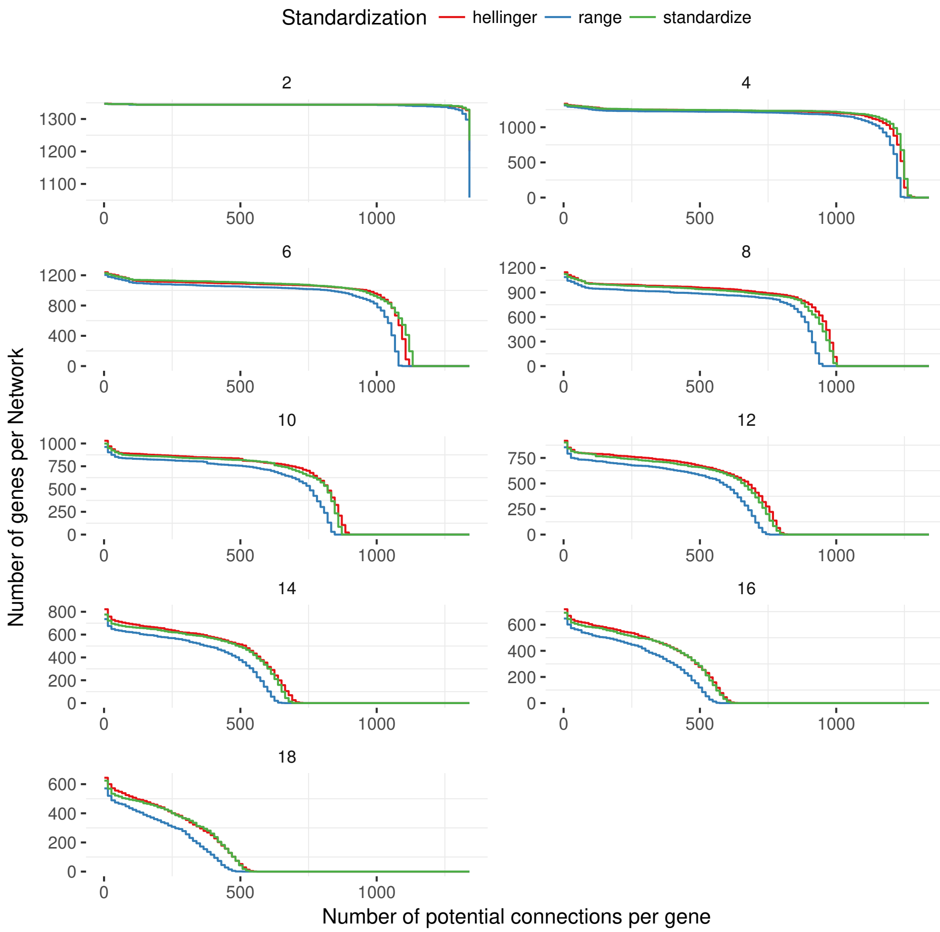
```
read.table("./data/modules.summary.119760.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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

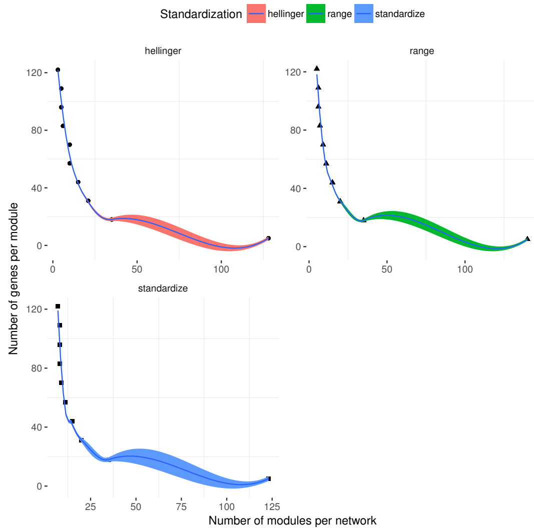
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

```r
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.

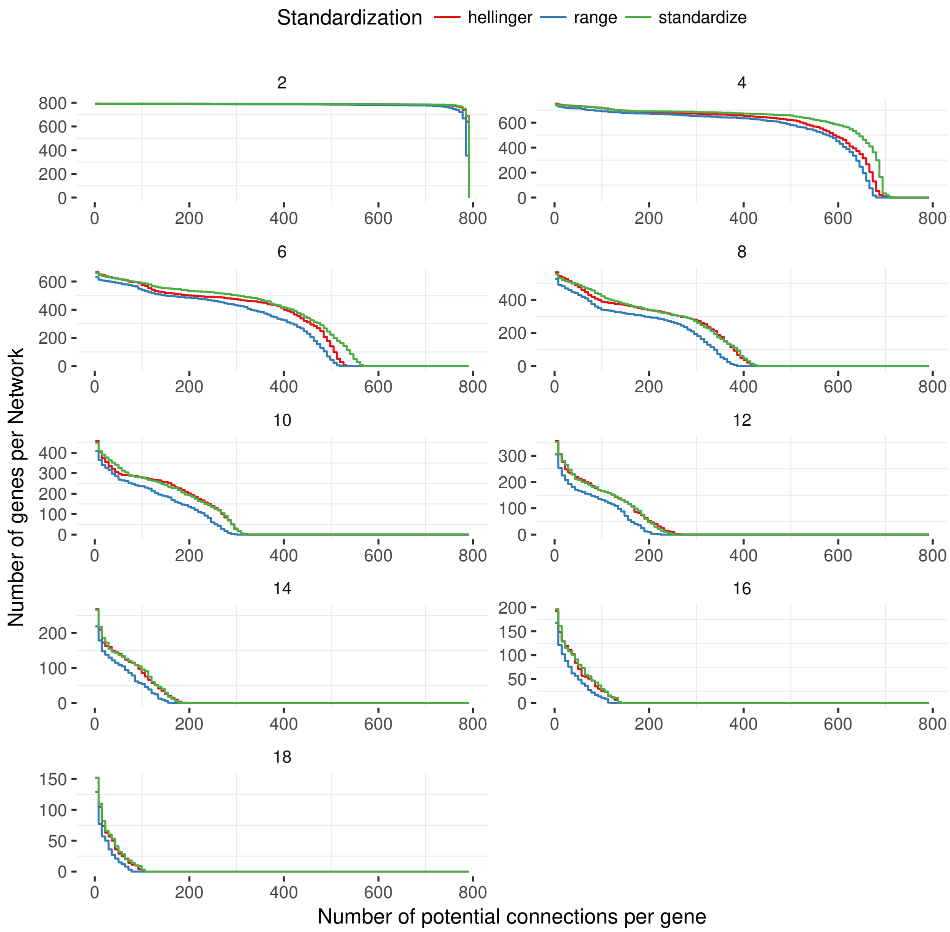
Same analysis with more stringent parameters

- **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

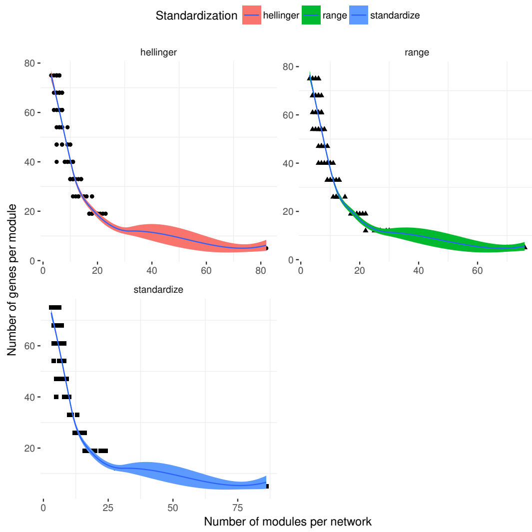
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```r
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.

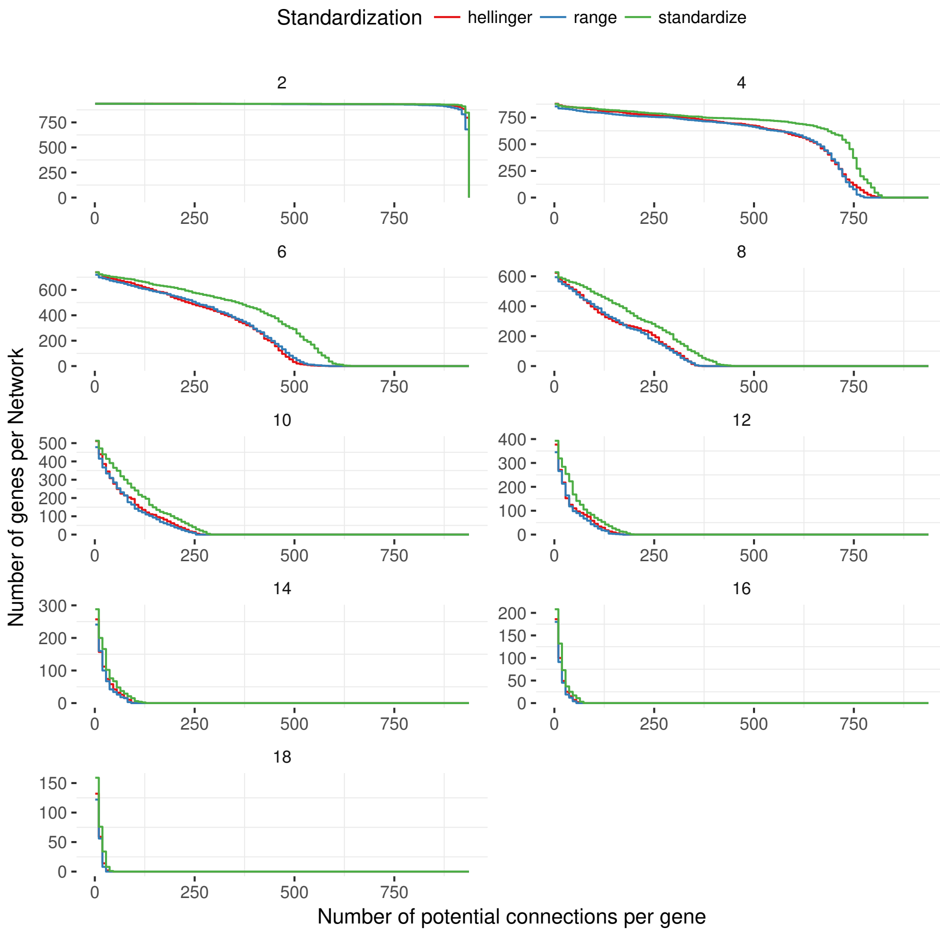
```
read.table("./data/modules.summary.119755.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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

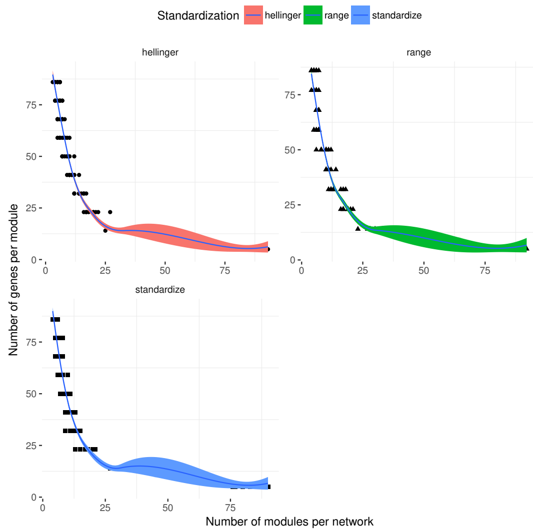
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

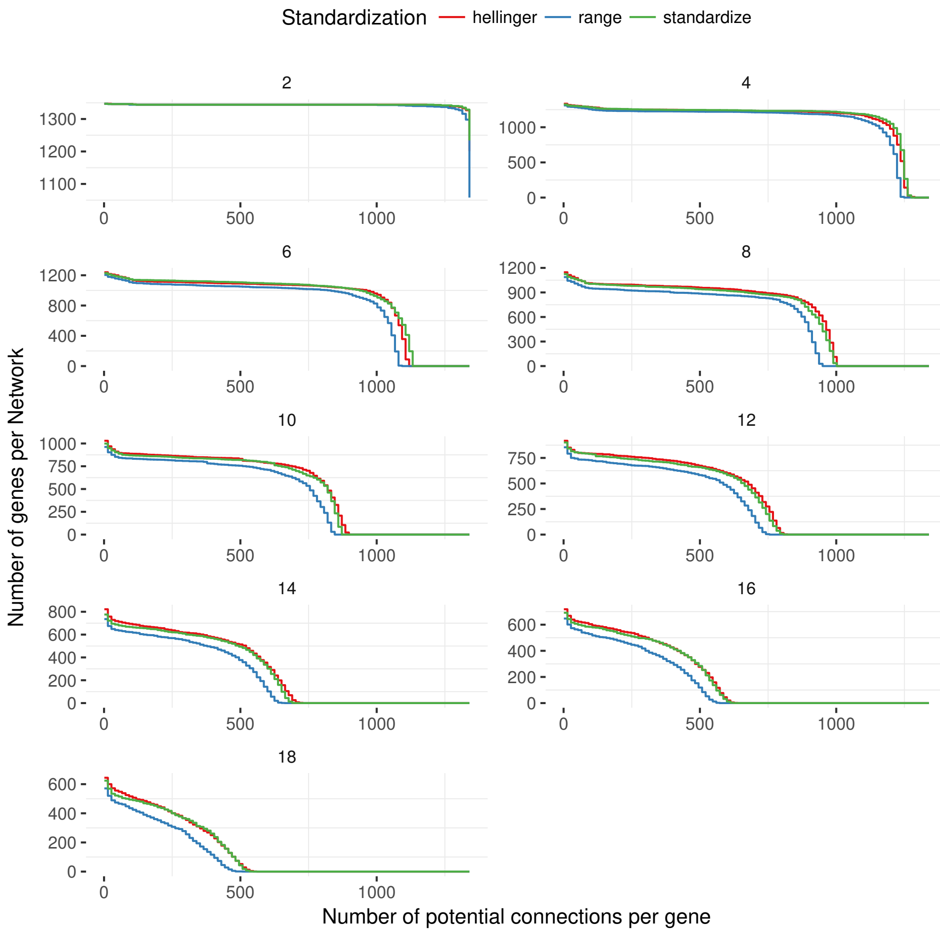
```r
read.table("./data/modules.summary.119754.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x = "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

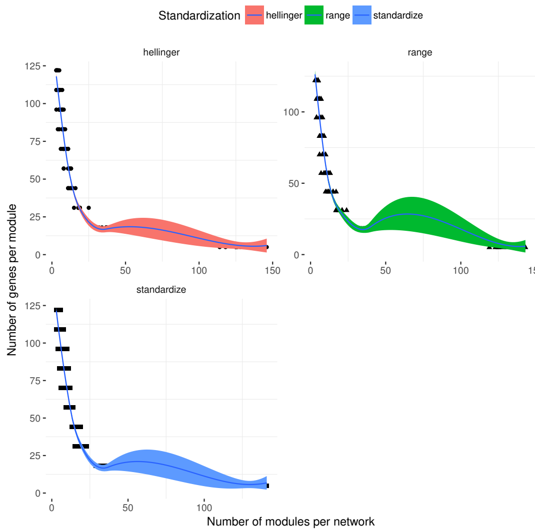
Genetic networks from differentially expressed genes selected by comparing sample cases with cell of origin classification based on ABC or GCB subtypes.

```
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

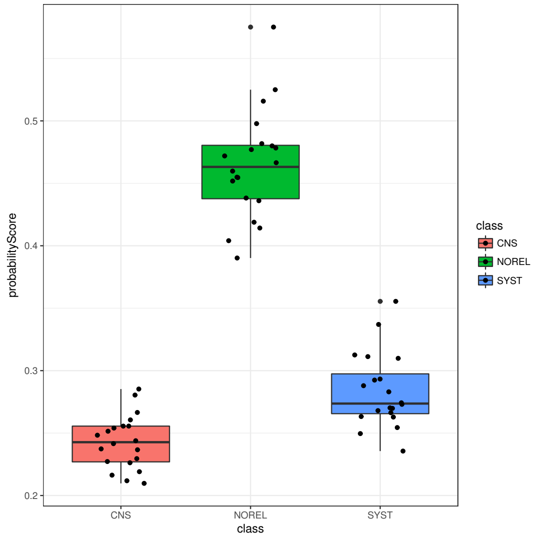
Least absolute shrinkage and selection operator (LASSO) was used for dimension reduction. Gene expression profiles were extracted from networks with significant connectivity. Subset selection using lasso, penalizes genes based on coefficient estimates, to increase accuracy of classification.

4.1.1 Uncertainty estimation for selected genes from expression networks

Plot showing, across a range of iterations, the mean probabilities of a subset of genes to correctly predict a patient case, or the certainty of a subset to estimate a correct 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.

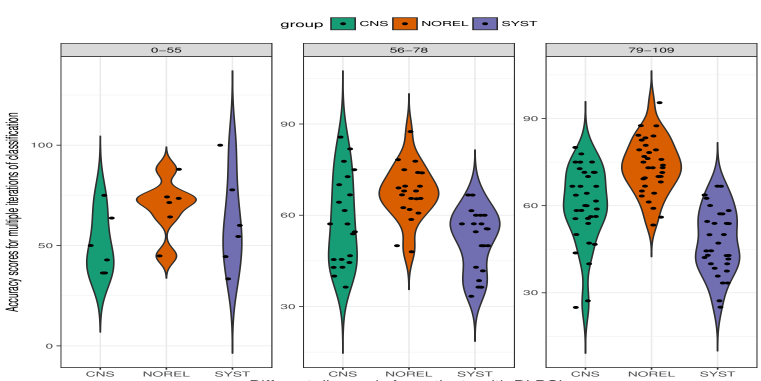
↰If a subset has 50 genes, the reported probabilities are the mean of each gene individual probability to predict all patient cases

```
read.table("./data/summary.lambda.iterations20.multinomial.probabilities.txt",
           row.names = 1, header = T) %>%
    ggplot(aes(x = class,
               y = probabilityScore,
               fill = class)) +
    theme_bw() +
    geom_boxplot() +
    geom_jitter(width = .2)
```



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/summary.lambda.iterations20.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") +
    facet_wrap( ~ grouped,
               ncol = 3,
               scales = "free") +
    theme(legend.position = "top")
```



4.2 Models selected for classification

4.3 Machine learning performance benchmarks

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 (relative

Table 1: Machine learning models

Model	R package*	Parameters	Abbreviation
Naive bayes	naivebayes	laplace, usekernel, adjust	naive_bayes
k-Nearest Neighbors	kknn	kmax, distance, kernel	kknn
Penalized multinomial regression	nnet	decay	multinom
Random forest	randomForest	mtry	rf
Regularized random forest	RRF	mtry, coefReg, coefImp	RRF
Linear discriminant analysis (LDA)	MASS	dimen	lda2
Localized LDA	klaR	k	loclda
Flexible discriminant analysis (FDA)	mda	degree, nprune	fda
Bagged FDA	mda	degree, nprune	bagFDA
Bagged FDA using gCV pruning	earth	degree	bagFDAGCV
Penalized discriminant analysis	mda	lambda	pda
Partial least squares	pls	ncomp	pls
Partial least squares	pls	ncomp	kernelpls
Bagged cart	e1071		treebag
Support vector machines (SVM) with linear kernel	kernlab	C	svmLinear
SVM with polynomial kernel	kernlab	degree, scale, C	svmPoly
SVM with radial basis function kernel	kernlab	sigma, C	svmRadialSigma
L2 regularized SVM (dual) with linear kernel	LiblineaR	cost, loss	svmLinear3
Neural network (NN)	nnet	size, decay	nnet
Monotone multi-layer perceptron NN	monmlp	hidden1, n.ensemble	monmlp
Stacked autoencoder deep NN	deepnet	layer1, layer2, layer3, hidden_dropout, visible_dropout	dnn
C5.0	C50	trials, model, winnow	C5.0
Boosted logistic regression	caTools	nIter	LogitBoost
Regularized logistic regression	LiblineaR	cost, loss, epsilon	regLogistic
Stochastic gradient boosting	gbm	n.trees, interaction.depth, shrinkage, n.minobsinnode	gbm

* The version of each package is shared at section 4.4.

to R^2).

```
accuracy <- read.table("./data/accuracy.performance.metrics.multianalysis.ml.txt", row.names = 1)
kappa <- read.table("./data/kappa.performance.metrics.multianalysis.ml.txt", row.names = 1)
```

4.4 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        finalfit_0.7.4      Hmisc_4.1-1
 [4] Formula_1.2-3       survival_2.42-3     brotools_0.2
 [7] scales_0.5.0        DescTools_0.99.23   igraph_1.1.2
[10] tidyr_0.8.0         dplyr_0.7.4         ggplot2_2.2.1
[13] latticeExtra_0.6-28 RColorBrewer_1.1-2  lattice_0.20-35
[16] gdata_2.18.0        knitr_1.20

loaded via a namespace (and not attached):
 [1] gtools_3.5.0      tidyselect_0.2.4  purrr_0.2.4
 [4] splines_3.4.4     colorspace_1.3-2  expm_0.999-2
 [7] htmltools_0.3.6   base64enc_0.1-3   rlang_0.2.0
[10] pillar_1.1.0      foreign_0.8-70    glue_1.2.0
[13] bindr_0.1         plyr_1.8.4        stringr_1.3.1
[16] munsell_0.4.3     gtable_0.2.0      htmlwidgets_1.2
[19] mvtnorm_1.0-7     evaluate_0.10.1   labeling_0.3
[22] manipulate_1.0.1  htmlTable_1.11.2  highr_0.6
[25] Rcpp_0.12.16      acepack_1.4.1     backports_1.1.1
[28] checkmate_1.8.5   gridExtra_2.3     digest_0.6.12
[31] stringi_1.2.2     grid_3.4.4        tools_3.4.4
[34] magrittr_1.5      lazyeval_0.2.1    tibble_1.4.2
[37] cluster_2.0.7-1   pkgconfig_2.0.1   MASS_7.3-47
[40] Matrix_1.2-11     data.table_1.11.2 rstudioapi_0.7
[43] assertthat_0.2.0  R6_2.2.2          boot_1.3-20
[46] rpart_4.1-13      nnet_7.3-12       compiler_3.4.4
```