Descriptive analysis

Sleiman Bassim, PhD

March 21, 2018

Contents

Loaded functions.

↰ Project started Dec 10 2017, updated March 21, 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')
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.**

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

```
 [1] "SAMPLE_ID"                     "PATIENT_ID"
 [3] "Timepoint"                     "OTHER_ID"
 [5] "res_id"                        "INCLUDE_MATCHING"
 [7] "INCLUDED_SUBMISSION_TCAG"      "GROUP"
 [9] "SITE"                          "Normalization"
[11] "Score"                         "ABClikelihood"
[13] "Prediction"                    "BCL2_BA"
[15] "BCL6_BA"                       "MYC_BA"
[17] "DH"                            "COMMENT"
[19] "CODE_OS"                       "CODE_DSS"
[21] "CODE_PFS"                      "CODE_TTP"
[23] "CODE_CNS"                      "Overall.survival..y."
[25] "Disease.specific.survival..y." "Progression.free.survival..y."
[27] "Time.to.progression..y."       "Time.to.CNS.relapse..y."
[29] "SEX"                           "AGE"
[31] "STAGE"                         "STAGEGRP"
[33] "E4SITE"                        "PS"
[35] "LDH"                           "LDHNORML"
[37] "LDHRATIO"                      "MASS"
[39] "IPI"                           "IPI_GROUP"
[41] "CNS.RiskScore"                 "CNS.RiskGrp"
[43] "Rehyb"
```

In the first steps of the analysis, the samples will be classified (supervised) into the following categories.

```
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)

summary(metadata)
```

```
     SAMPLE_ID   Timepoint                              GROUP
 CNR1001T1:  1   T1:236    NO_RELAPSE                     :96
 CNR1002T1:  1   T2:  0    SYTEMIC_RELAPSE_NO_CNS         :64
 CNR1003T1:  1             CNS_RELAPSE_RCHOP              :39
 CNR1006T1:  1             TESTICULAR_NO_CNS_RELAPSE      :12
 CNR1007T1:  1             CNS_DIAGNOSIS                  :11
 CNR1008T1:  1             CNS_RELAPSE_CHOPorEQUIVALENT: 8
 (Other)  :230             (Other)                        : 6
      SITE          Score      Prediction ABClikelihood    Groups
 LN     :127   Min.   :-881   ABC : 92   Min.   :0.00   CNS  : 58
 SO     : 20   1st Qu.: 676   GCB :103   1st Qu.:0.00   CTRL :  6
 TE     : 18   Median :2106   U   : 39   Median :0.02   NOREL:108
 TO     : 16   Mean   :1820   NA's:  2   Mean   :0.47   SYST : 64
 GI     : 11   3rd Qu.:2941              3rd Qu.:1.00
 SP     :  7   Max.   :4323              Max.   :1.00
 (Other): 37   NA's   :2                 NA's   :4
 ABClassify ABCScore  Nodes    Lymphnodes
 ABC:103    ABC: 92   EN: 86   0: 86
 GCB:117    GCB:103   LN:150   1:150
 U  : 16    U  : 41
```

1.1 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 %>%
```

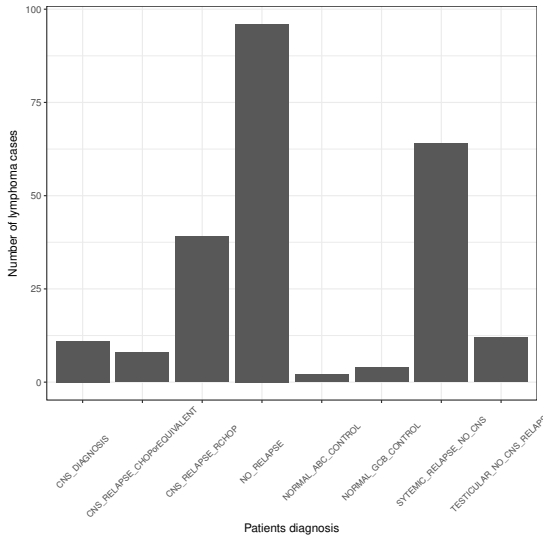
```
    select(Prediction, ABClassify, ABCScore) %>%
    summary
```

```
 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.

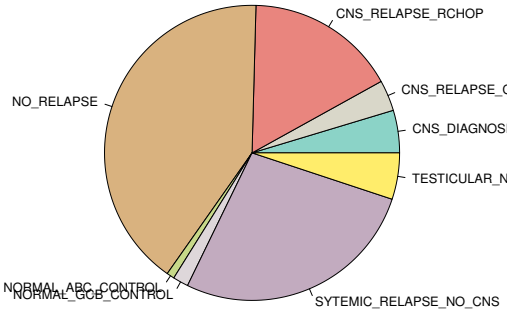
```
metadata %>%
    select(GROUP) %>%
    ggplot(aes(x = GROUP)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of lymphoma cases",
         x = "Patients diagnosis") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
                                     angle = 45,
  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

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



Or as a pie chart.

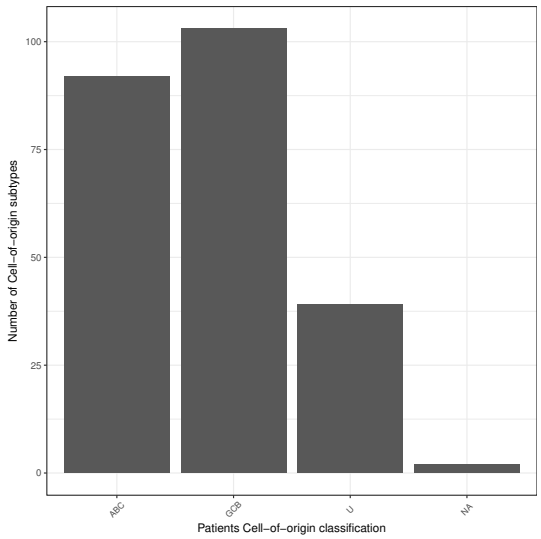
```
palette.pies <- brewer.pal(12, name = "Set3")
palette.pies.adj <- colorRampPalette(palette.pies)(length(unique(metadata$GROUP)))
pie(table(metadata$GROUP), col=palette.pies.adj)
```



Distribution of samples with different cells of origin subtypes.

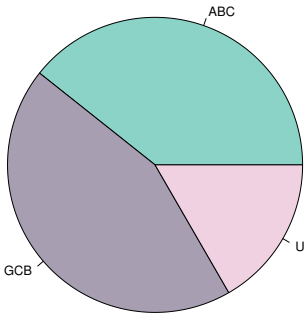
```
metadata %>%
    select(Prediction) %>%
    ggplot(aes(x = Prediction)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of Cell-of-origin subtypes",
         x = "Patients Cell-of-origin classification") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
            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

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



Or as pie chart.

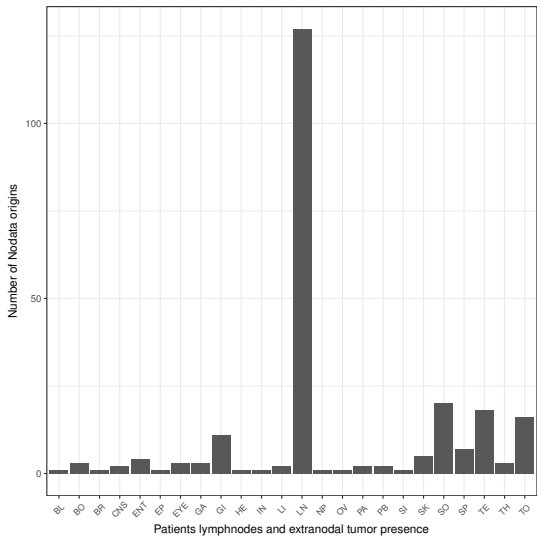
```
palette.pies <- brewer.pal(12, name = "Set3")
palette.pies.adj <- colorRampPalette(palette.pies)(length(unique(metadata$Prediction)))
pie(table(metadata$Prediction), col=palette.pies.adj)
```



Distribution of samples with different lymphnodes and extranodal cancer metastasis.

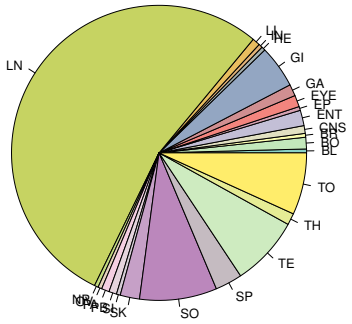
```
par(mfrow=c(2,2))
metadata %>%
    select(SITE) %>%
    ggplot(aes(x = SITE)) +
    geom_histogram(stat = "count") +
    labs(y = "Number of Nodata origins",
         x = "Patients lymphnodes and extranodal tumor presence") +
    theme_bw() +
    theme(axis.text.x = element_text(vjust = .5,
    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

```
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).

```
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.

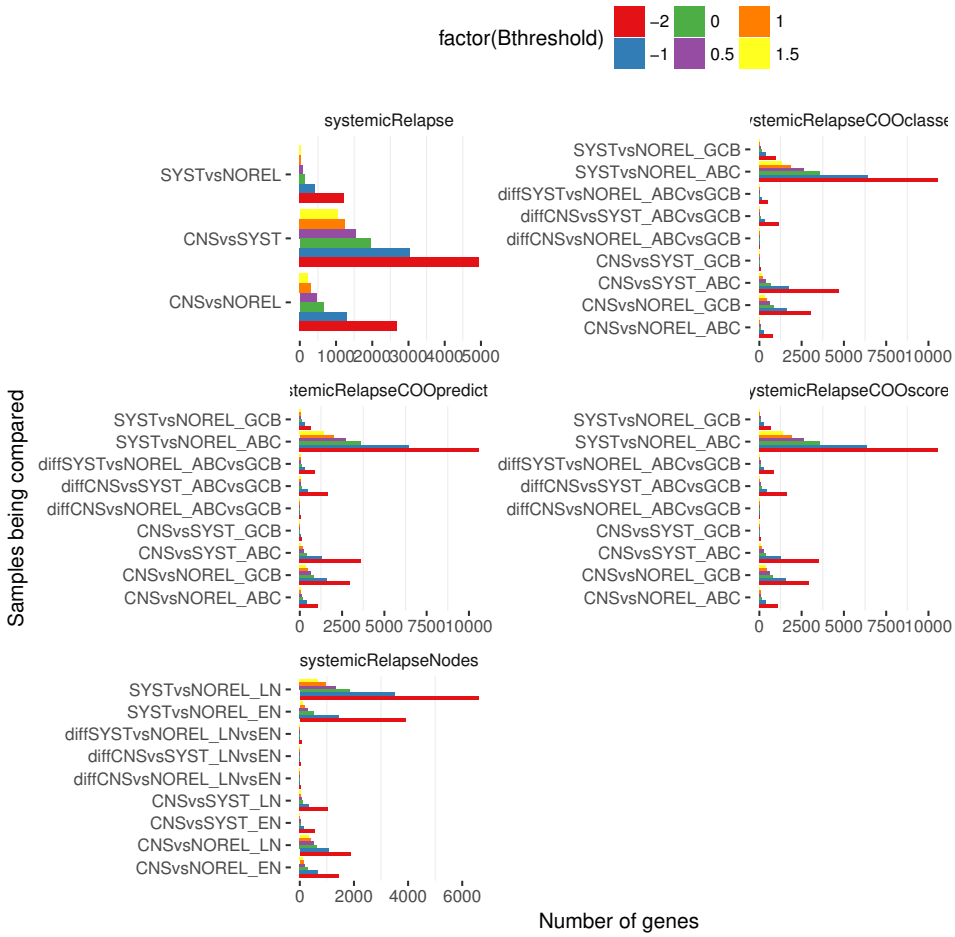
```
aggregate( Transcripts ~ Design, data=expression, FUN=range)
```

```
                    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.

```
expression %>%
```

```
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. Each array contains probes of 75,523 different RNAs. Either ncRNA, mRNA, and non annotated genes. More than 46% 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

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

was transformed (standardization protocol) before calculating means and variances. This helps a better signal recovery from a large dataset with potential expression bias.

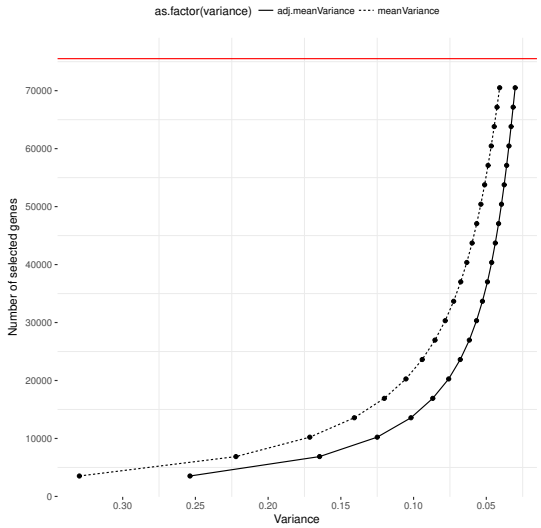
↰ Each array correspond to a DLBCL patient's case

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.

```
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"))
```

↰ The smaller the variance, the better

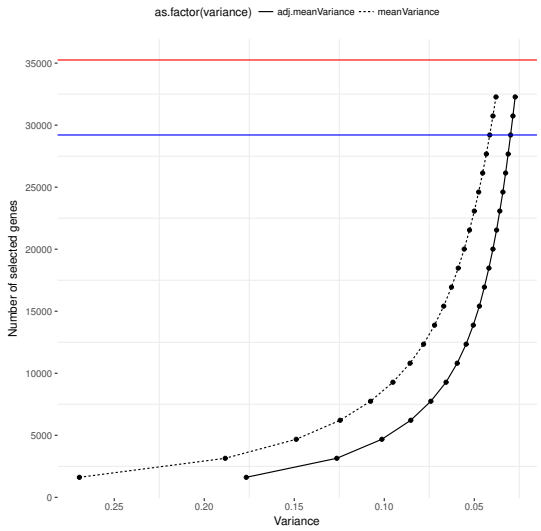


Same plot description as above however we removed ncRNA which account for 46% of the probes. The total number of transcripts is now 35,253 (red horizontal line). The blue horizontal line represents the threshold that was selected for subsequent analysis.

↰ 29,207 genes were selected for clustering and nets

```
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"))
```



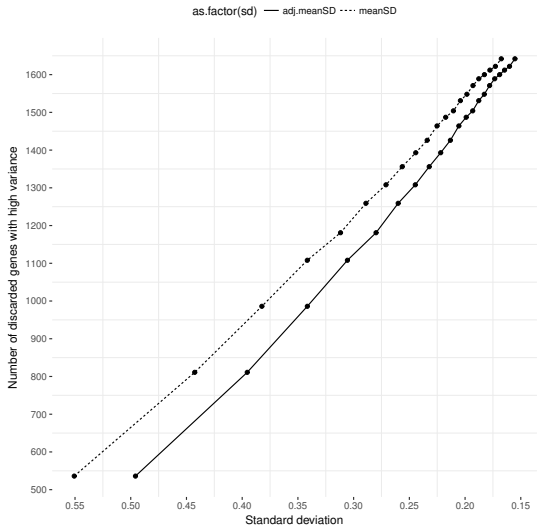
2.1.2 Standard deviation optimization for each array

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 shows how the standard deviation, **spread** of the data, is getting smaller, more we discard genes with high/low variance. All array probes with all RNAs.

↰ Best if small spread between 2 SDs

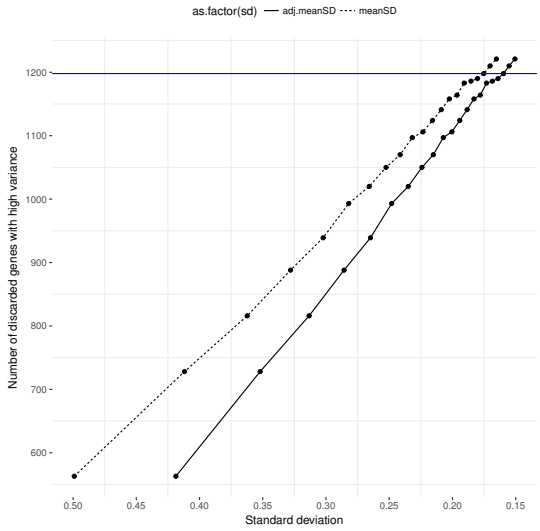
```
read.table("./data/summary.139102.adjusted.means.subsetting.txt", header = T) %>%
```

```
    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).

[4]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

[5]Effect of correlation methods is seen on module content

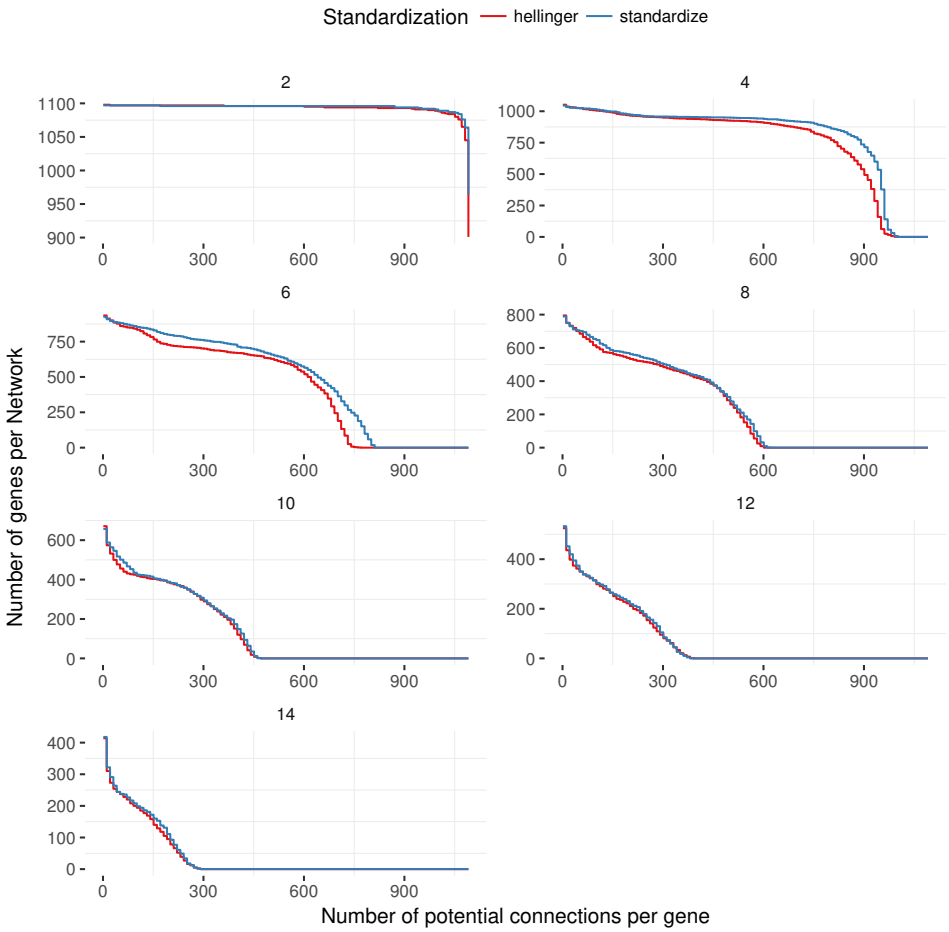
```
ns <- read.table("./data/networks.summary.104795.txt", header = T)
```

```
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

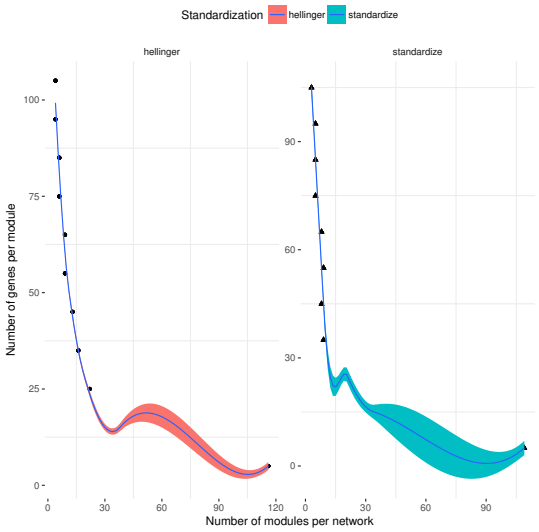
```
ns %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,
               ncol = 2,
               scales = "free")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

[4]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)

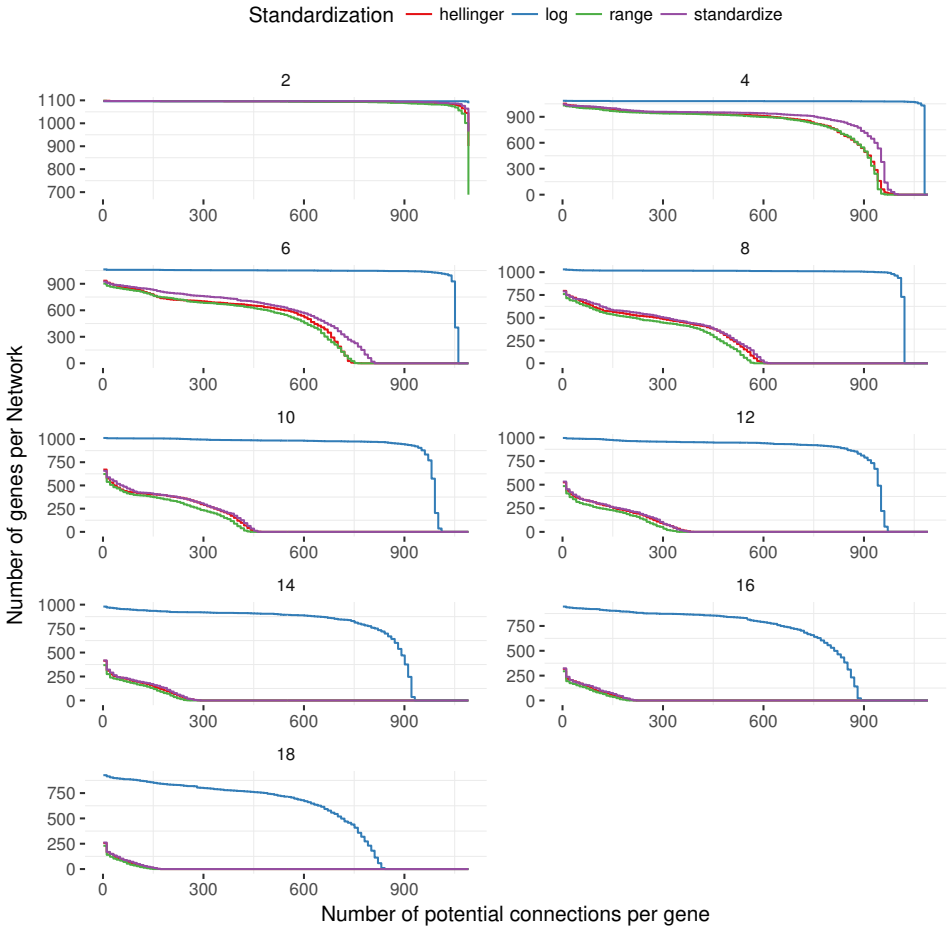
Thresholds based on the Empirical Bayes approach to rank genes and determine if a gene is significantly expressed. Limma implementation.

- **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

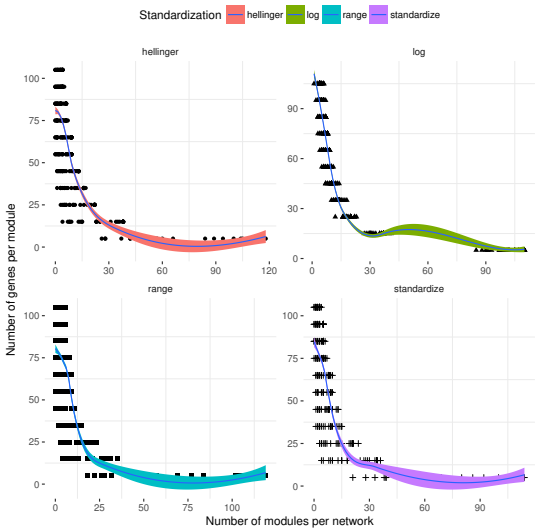
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
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.

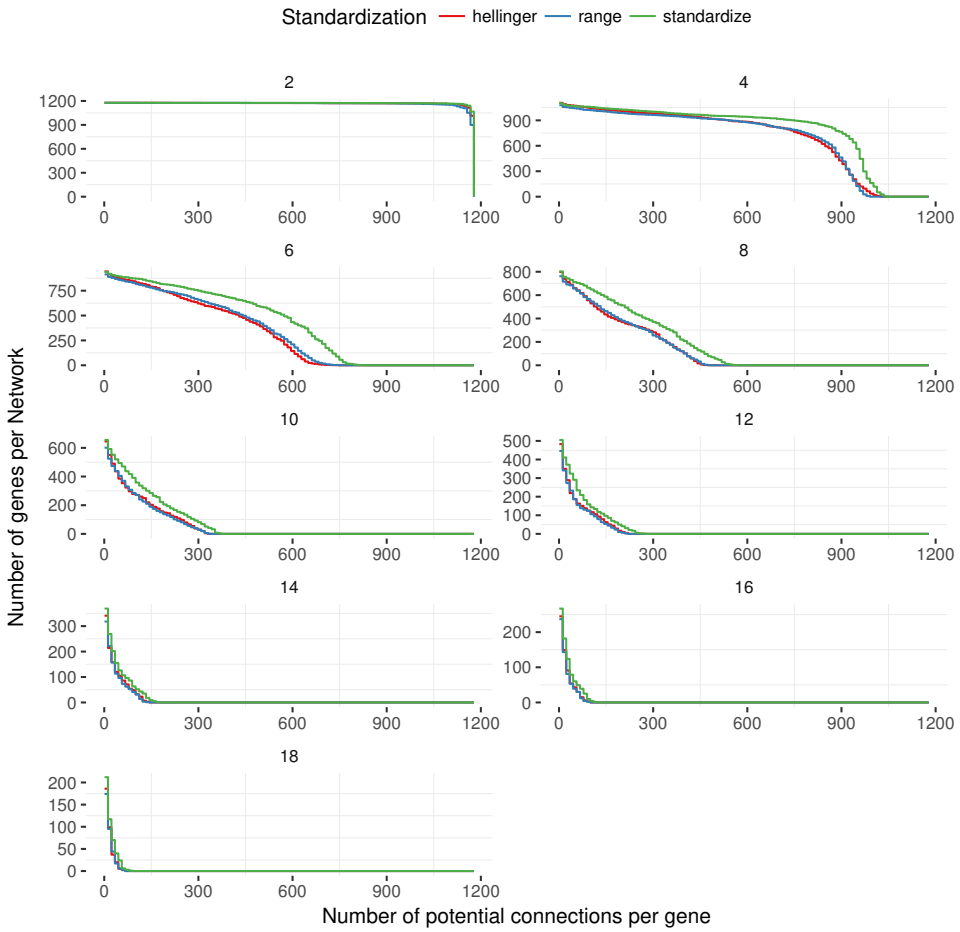
```
read.table("./data/modules.summary.104859.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

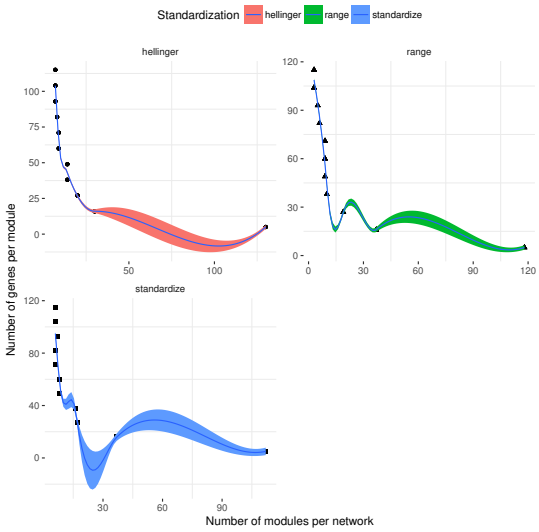
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

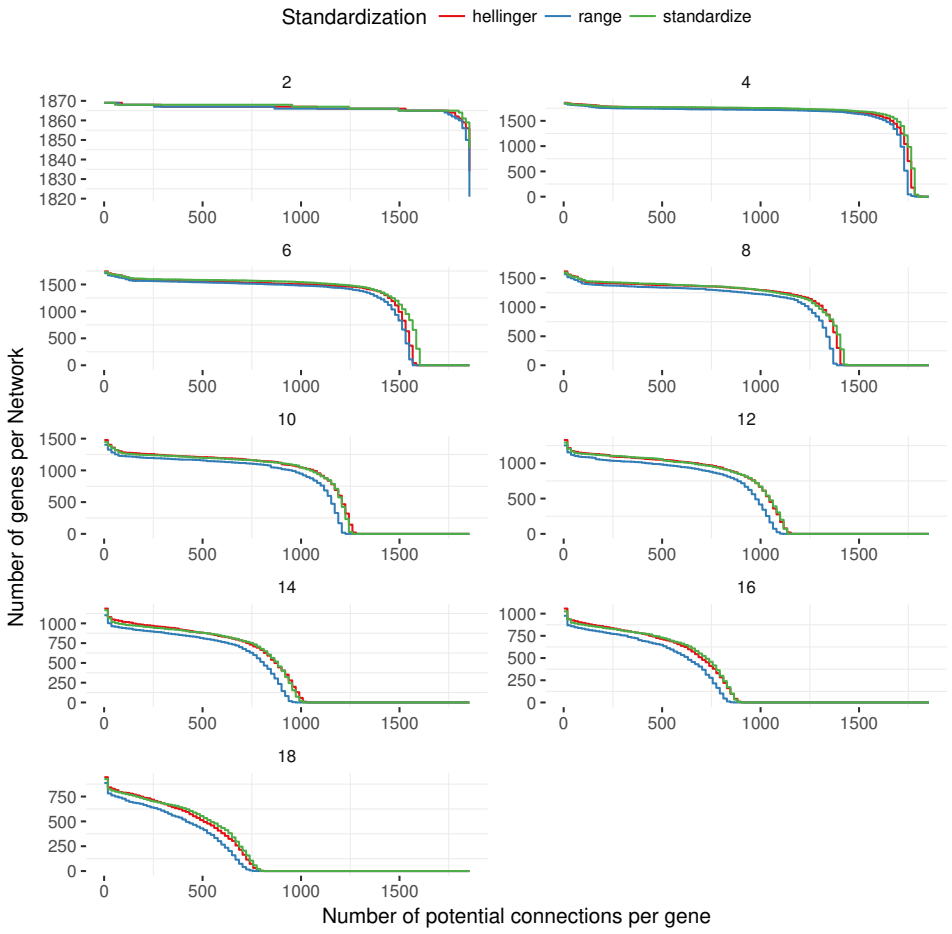
```
read.table("./data/modules.summary.114018.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

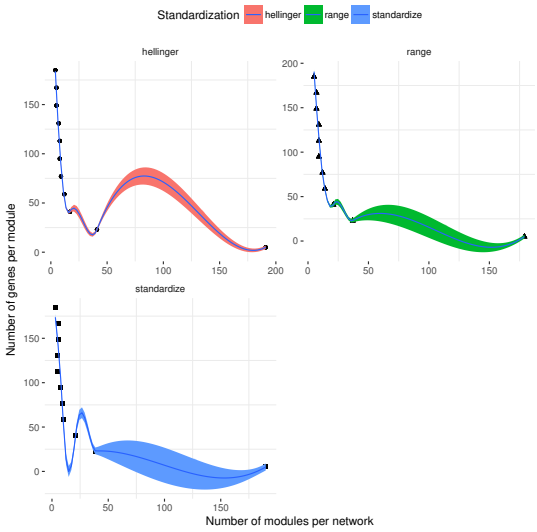
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

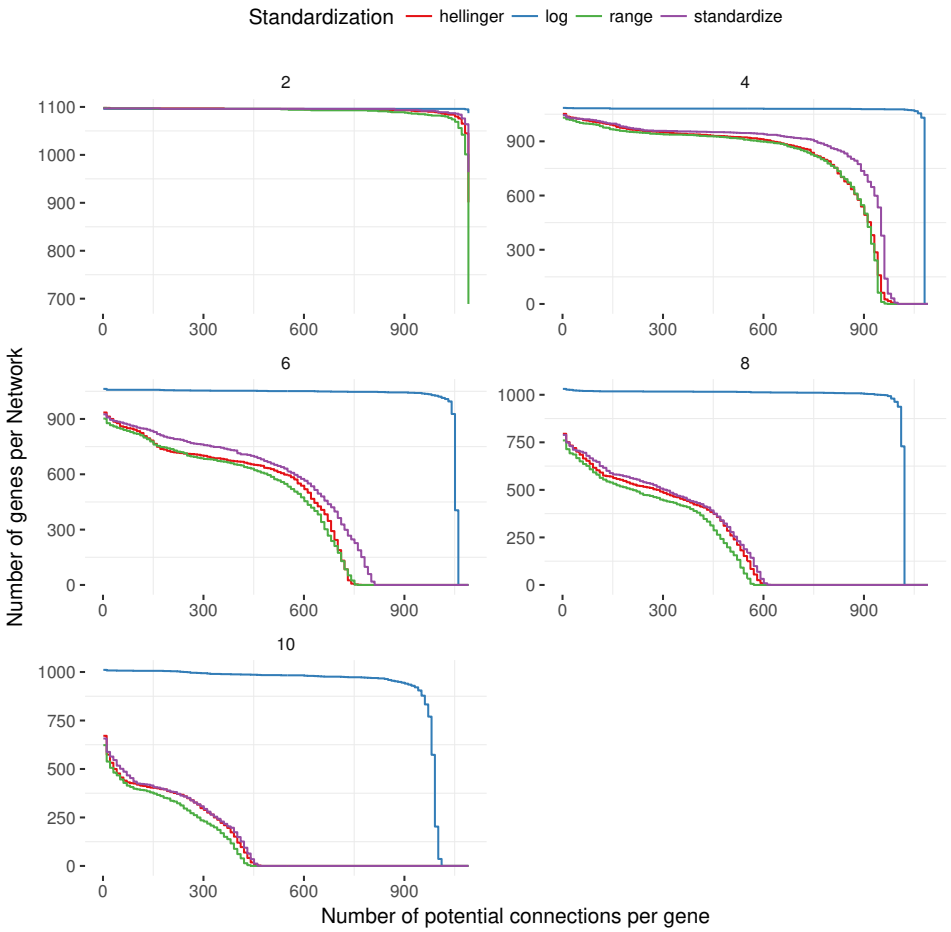
[†]With pearson, we can only raise the data to power 10. All are discarded after 10.

- **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.

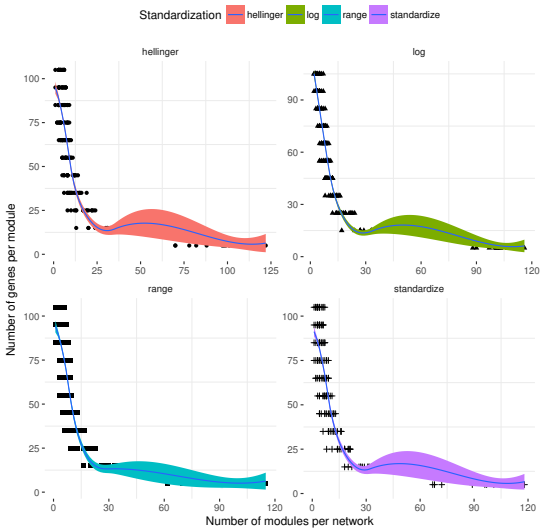
```
read.table("./data/networks.summary.104862.txt", header = TRUE) %>%
    ggplot(aes(
        x = MaxEdgesPerGene,
        y = NbNodes,
        fill = Standardization)) +
    theme_bw() +
    geom_step(aes(color = Standardization),
              stat = "identity") +
    facet_wrap( ~ CorrelationPower,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

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

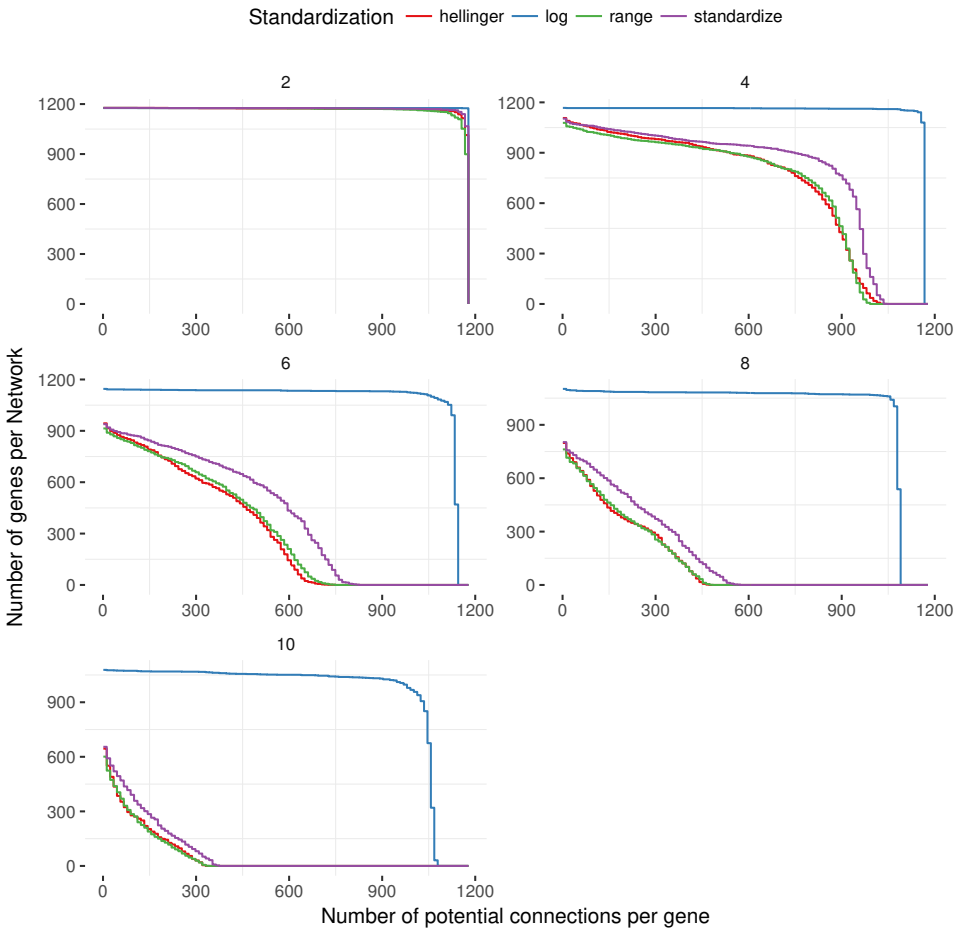
```
read.table("./data/modules.summary.104862.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

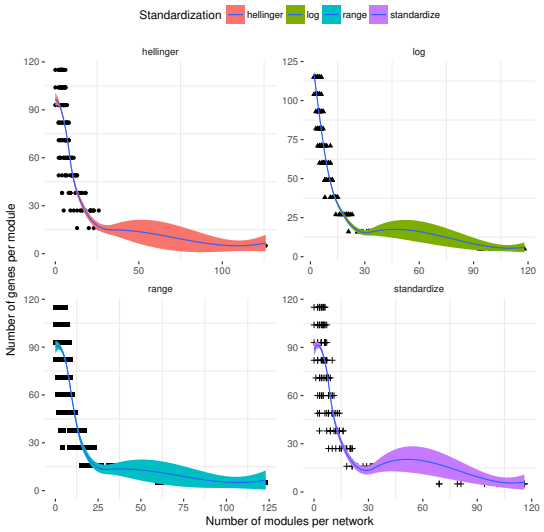
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

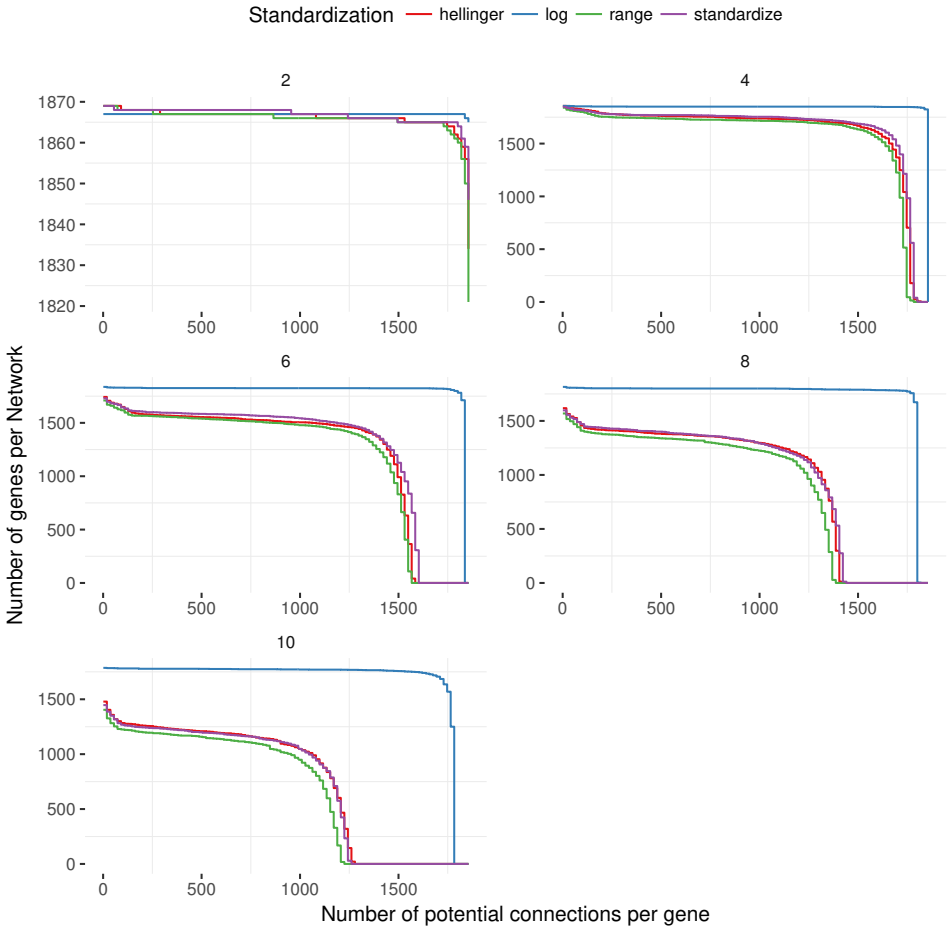
```
read.table("./data/modules.summary.104863.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

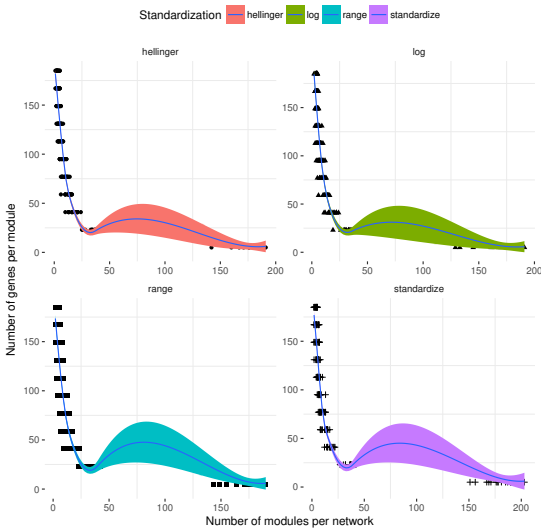
Genetic networks from differentially expressed genes selected by comparing sample cases with cell of origin classification based on ABC or GCB subtypes.

```
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.

```
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.

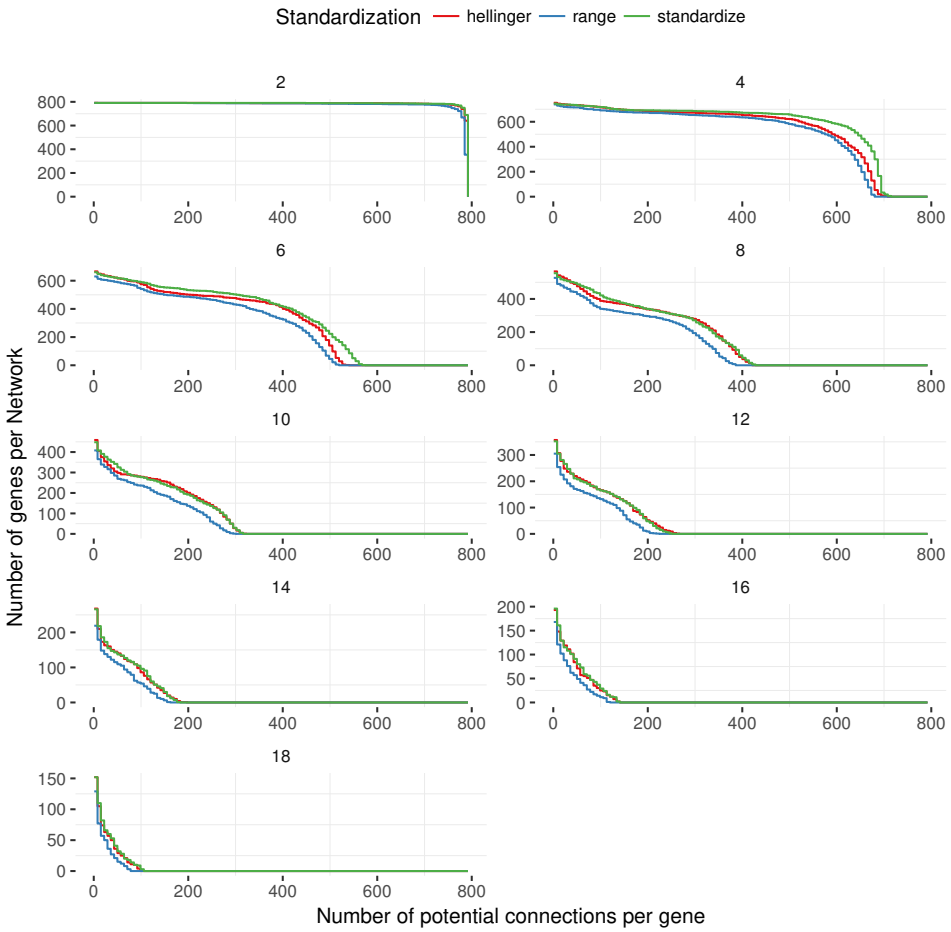
[†]Same analysis with more stringent parameters

- **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

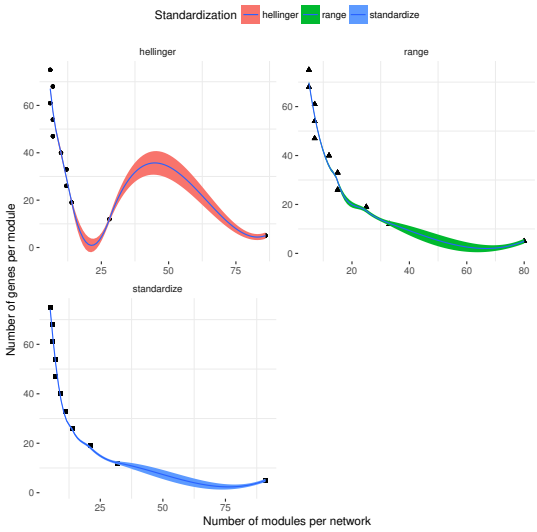
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
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.

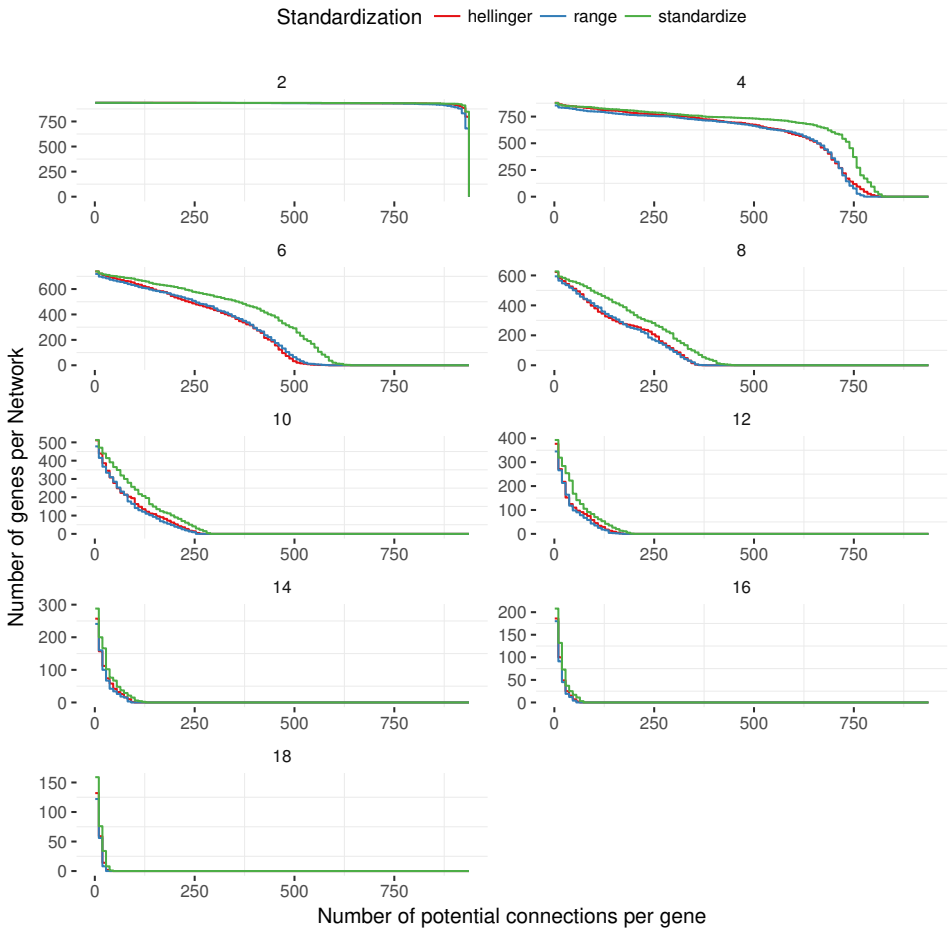
```
read.table("./data/modules.summary.119759.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

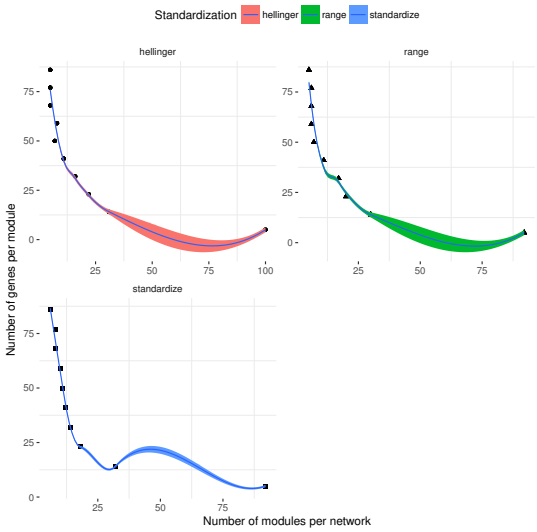
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

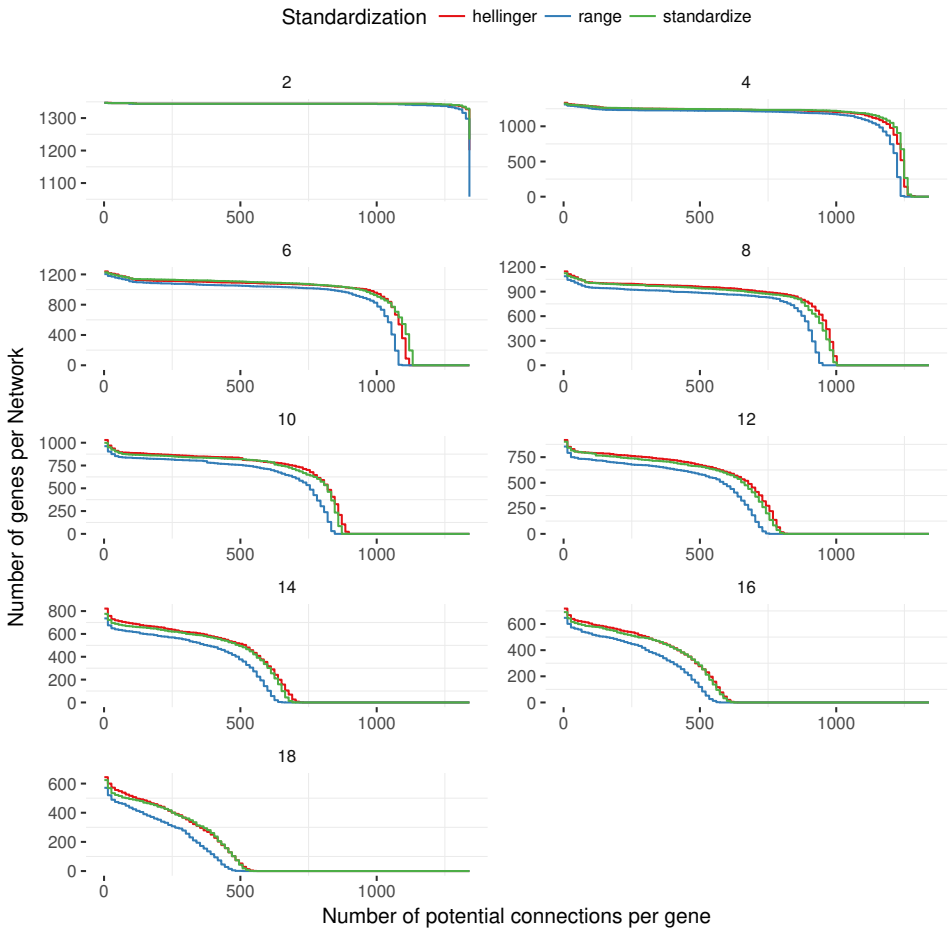
```
read.table("./data/modules.summary.119760.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

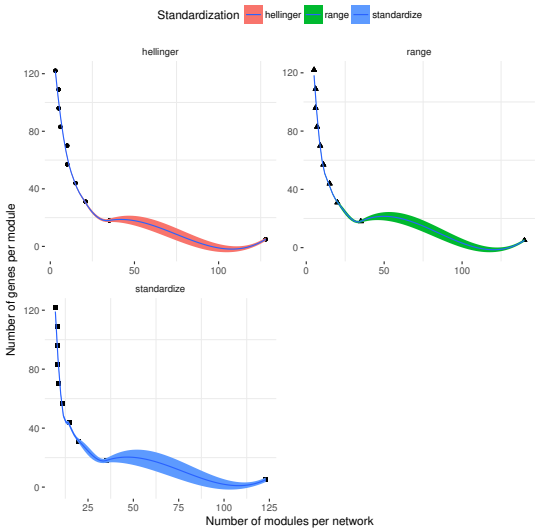
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

```
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.

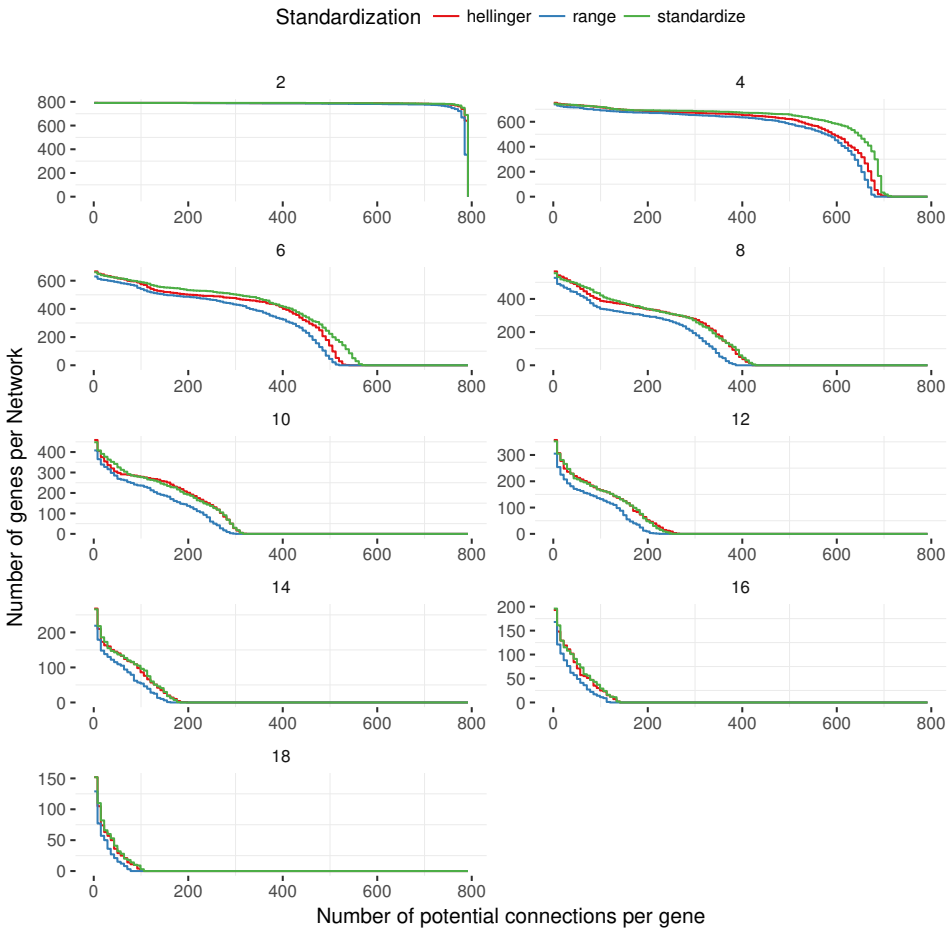
[†]Same analysis with more stringent parameters

- **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

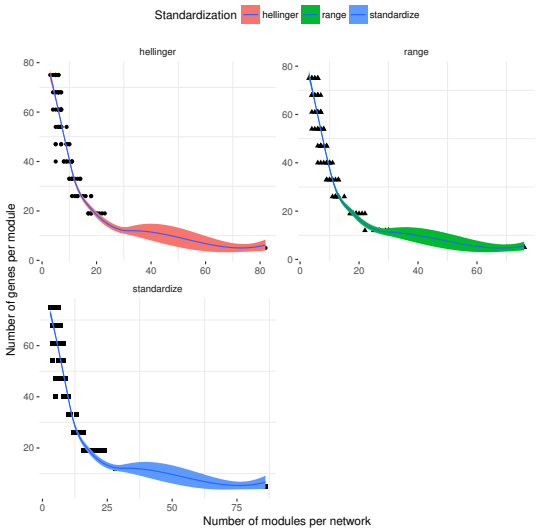
Genetic networks from differentially expressed genes selected by comparing sample cases with nodal and extranodal lymphoma.

```
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.

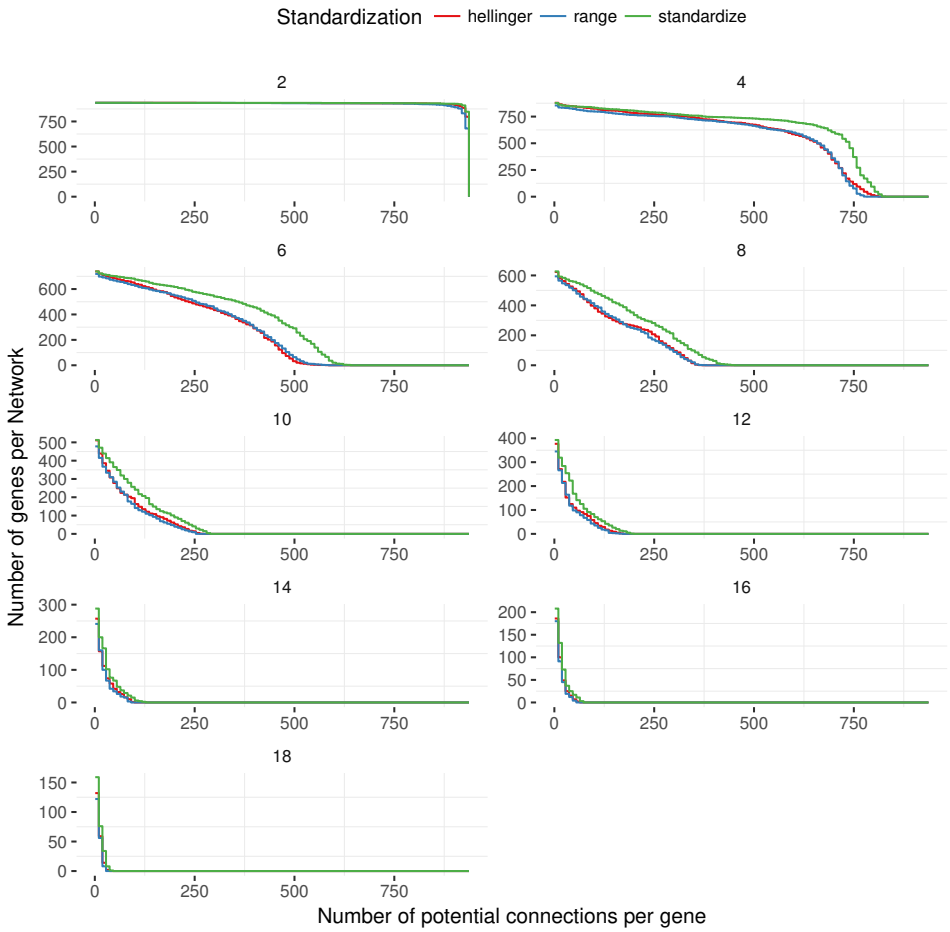
```
read.table("./data/modules.summary.119755.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

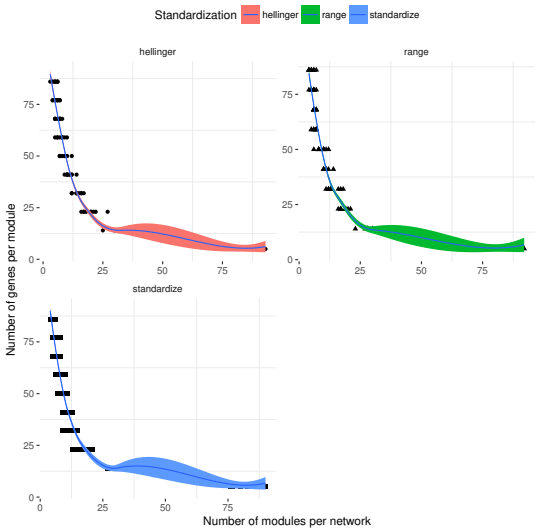
Genetic networks from differentially expressed genes selected by comparing sample cases with systemic or no CNS relapse lymphoma.

```
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.

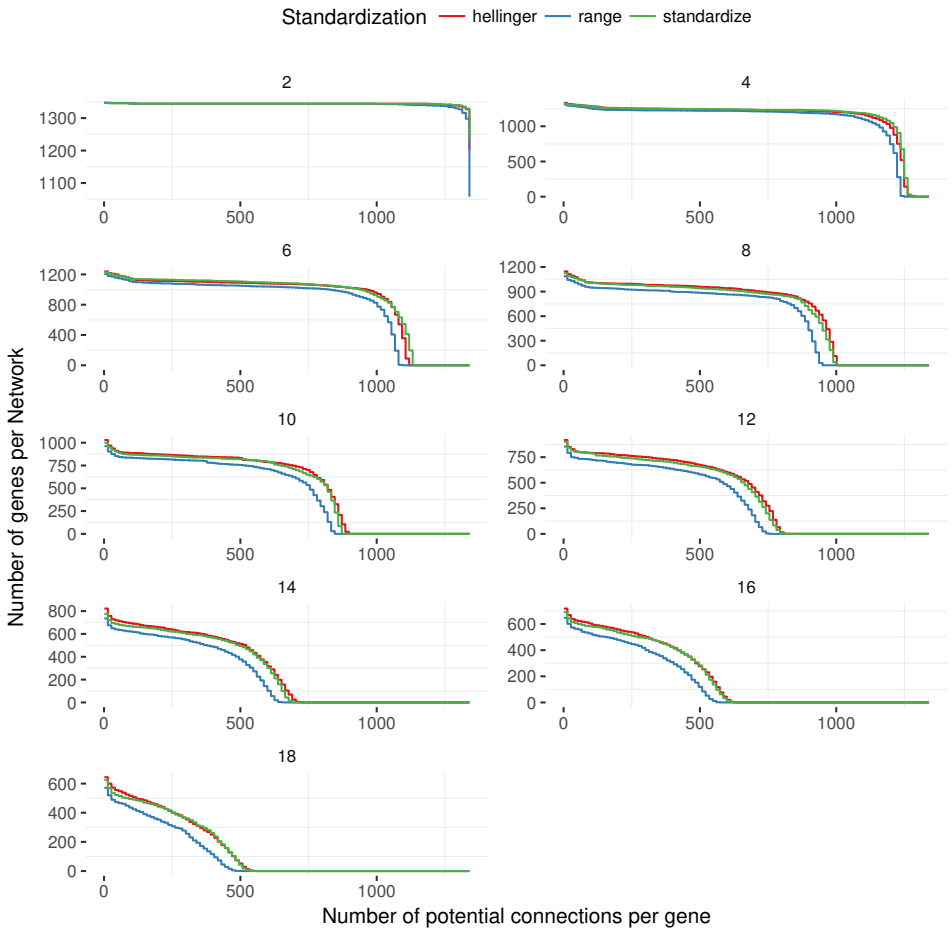
```
read.table("./data/modules.summary.119754.txt", header = TRUE) %>%
    ggplot(aes(
        x = NbModules,
        y = MaxGenesPerModule,
        fill = Standardization)) +
    theme_bw() +
    geom_point(aes(shape = Standardization)) +
    scale_color_brewer(type = "qual", palette = 6) +
    labs(x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

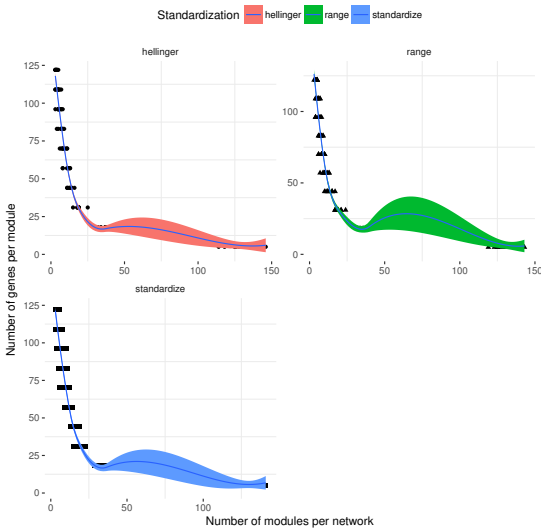
Genetic networks from differentially expressed genes selected by comparing sample cases with cell of origin classification based on ABC or GCB subtypes.

```
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 System Information

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

```
###save(list=ls(pattern=".*|.*"),file="PD.Rdata")
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        scales_0.5.0        DescTools_0.99.23
 [4] igraph_1.1.2        tidyr_0.7.2         dplyr_0.7.4
 [7] ggplot2_2.2.1       latticeExtra_0.6-28 RColorBrewer_1.1-2
[10] lattice_0.20-35     gdata_2.18.0        knitr_1.17

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.13     pillar_1.1.0     compiler_3.4.4
 [4] plyr_1.8.4       highr_0.6        bindr_0.1
 [7] tools_3.4.4      digest_0.6.12    boot_1.3-20
[10] evaluate_0.10.1  tibble_1.4.2     manipulate_1.0.1
[13] gtable_0.2.0     pkgconfig_2.0.1  rlang_0.1.2
[16] Matrix_1.2-11    expm_0.999-2     mvtnorm_1.0-6
[19] stringr_1.2.0    gtools_3.5.0     tidyselect_0.2.2
[22] grid_3.4.4       glue_1.2.0       R6_2.2.2
[25] foreign_0.8-69   purrr_0.2.4      magrittr_1.5
[28] MASS_7.3-47      assertthat_0.2.0 colorspace_1.3-2
[31] labeling_0.3     stringi_1.1.5    lazyeval_0.2.1
[34] munsell_0.4.3
```