Integrative Analysis

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Integrative Analysis

Below are provided Integrative Analysis of the available Proteomics and Transcriptomics data. The analysis steps have been applied over the *EGF-driven protein synthesis* case-study data from D.A. Rothenberg et al. A Proteomics Approach to Profiling the Temporal Translational Response to Stress and Growth. iScience. 2018; 9:367-381 at *time-point 60min*.

Loading of R-Packages

We start by setting a seed for reproducibility of the results and then loading the R-packages we need to use for our analysis.

```r
set.seed(1234)
library("readr")
library("vsn")
library("dplyr")
library("limma")
library("ggplot2")
library("ggrepel")
library("BioNet")
library("igraph")
library("OmnipathR")
library("ggpubr")
library("mixOmics")
library("M2SMF")
library("SNFtool")
library("NEMO")
library("fgsea")
library("GSA")
library("VennDiagram")
library("RColorBrewer")
library("ggVennDiagram")
library("pheatmap")
library("tidyverse")
library("factoextra")
library("gridExtra")
library("cluster")
library("NNLM")
library("bayesCC")
```

Loading the Data

We load the Proteomics and Gene Expression data.

```r
# Differential Gene Expression Data
load(file = "../Data/ttop_dge.RData")
head(ttop_dge[, 1:(ncol(ttop_dge)-1)])
```

```
FALSE                 external_gene_name          GeneID Length    logFC   logCPM
FALSE ENSG00000125740               FOSB ENSG00000125740   5553 4.962212 6.902998
FALSE ENSG00000138166              DUSP5 ENSG00000138166   2535 3.983957 6.291272
FALSE ENSG00000119508              NR4A3 ENSG00000119508   6314 3.302585 8.107344
FALSE ENSG00000175592              FOSL1 ENSG00000175592   1887 3.307155 5.967628
FALSE ENSG00000171223               JUNB ENSG00000171223   1830 3.283124 7.112127
FALSE ENSG00000162772               ATF3 ENSG00000162772   4040 3.119576 7.369775
FALSE                        PValue           FDR         GO
FALSE ENSG00000125740  0.000000e+00  0.000000e+00 GO:0003677
FALSE ENSG00000138166  0.000000e+00  0.000000e+00 GO:0016791
FALSE ENSG00000119508  0.000000e+00  0.000000e+00 GO:0003677
FALSE ENSG00000175592 5.605306e-296 2.058689e-292 GO:0003677
FALSE ENSG00000171223 1.089103e-281 3.200002e-278 GO:0003677
FALSE ENSG00000162772 9.076298e-280 2.222331e-276 GO:0003677
```

```r
# Differential Protein Abundance
load(file = "../Data/ttop_prot.RData")
head(ttop_prot)
```

```
FALSE               name  Gene   Accession      Sequence EGF_60_vs_PBS_60_diff
FALSE 1669  RS3A_HUMAN.2 RPS3A  RS3A_HUMAN       IASDGLK             0.7896095
FALSE 400  CYR61_HUMAN.2 CYR61 CYR61_HUMAN     NNELIAVGK             1.0285004
FALSE 1570   RL3_HUMAN.2  RPL3   RL3_HUMAN       VAFSVAR             0.5627584
FALSE 1557    RL26_HUMAN RPL26  RL26_HUMAN      DDEVQVVR             0.5180723
FALSE 514   EGR1_HUMAN.2  EGR1  EGR1_HUMAN TQQPSLTPLSTIK             1.9551439
FALSE 1597  RL7A_HUMAN.3 RPL7A  RL7A_HUMAN     KVVNPLFEK             0.5591097
FALSE      EGF_60_vs_PBS_60_p.adj EGF_60_vs_PBS_60_p.val
FALSE 1669           1.717267e-09           9.248393e-10
FALSE 400            3.626256e-03           6.651558e-06
FALSE 1570           4.244779e-03           8.275133e-06
FALSE 1557           5.074382e-03           1.116453e-05
FALSE 514            1.046605e-02           2.404454e-05
FALSE 1597           1.170257e-02           2.783308e-05
```

```r
# Processed Gene Expression Data across EGF and PBS samples at time-point 60min
load(file = "../Data/proc_gene_data.RData")
head(proc_gene_data)
```

```
FALSE           PBS_1    PBS_2    EGF_1    EGF_2
FALSE ACAA2  5.943056 5.960073 5.994337 5.899598
FALSE ACACA  7.127608 7.171814 7.056132 6.949111
FALSE ACADVL 7.016260 7.081363 7.026467 7.080112
FALSE ACIN1  6.850670 6.876240 6.871092 6.796145
FALSE ACLY   9.528757 9.540166 9.468933 9.444991
FALSE ACP1   7.230251 7.261917 7.187795 7.209038
```

```r
# Processed Protein Abundance Data across EGF and PBS samples at time-point 60min
load(file = "../Data/proc_prot_data.RData")
head(proc_prot_data)
```

```
FALSE           PBS_1    PBS_2    EGF_1    EGF_2
FALSE ACAA2  15.38947 14.60145 15.43671 15.25266
FALSE ACACA  15.32710 13.89533 15.24979 15.96504
FALSE ACADVL 14.95859 14.70503 14.90234 15.84788
FALSE ACIN1  14.64568 14.77276 14.72002 13.39753
FALSE ACLY   20.73771 20.94036 20.70857 20.63984
FALSE ACP1   15.55535 15.61272 15.40002 15.80990
```

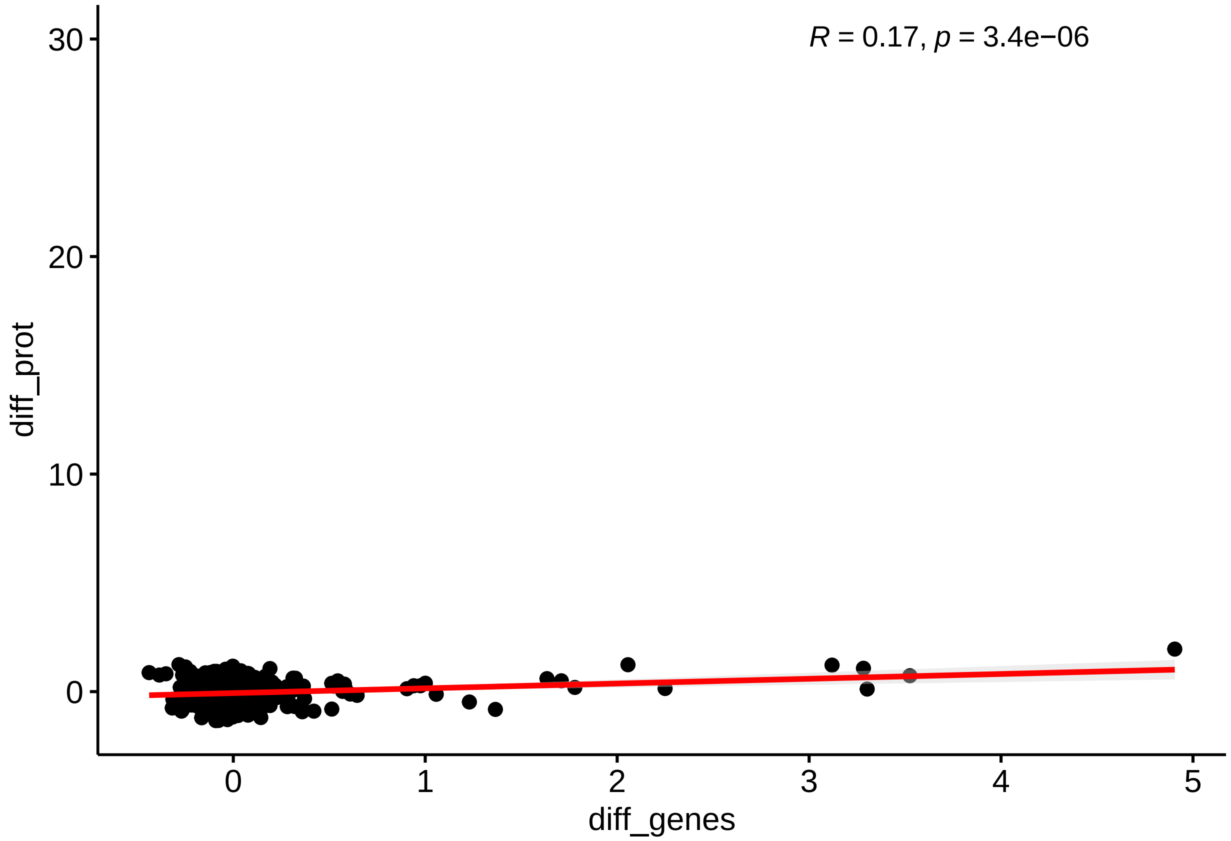
Correlation Analysis

> We can look at the **correlation** in the expression between **Differential** Gene Expression and the Abundance of the corresponding Proteins.

```
# We find common Genes and and filter each data
common_genes <- intersect(x = ttop_dge$external_gene_name, y = ttop_prot$Gene)
dge <- ttop_dge[which(ttop_dge$external_gene_name%in%common_genes), ]
prot <- ttop_prot[which(ttop_prot$Gene%in%common_genes), ]
# We create the data-frame for plotting the correlation
data <- matrix(data = , nrow = length(common_genes), ncol = 2)
rownames(data) <- common_genes[order(common_genes)]
colnames(data) <- c("diff_genes", "diff_prot")
data[, 1] <- dge$logFC[order(dge$external_gene_name)]
data[, 2] <- prot$EGF_60_vs_PBS_60_diff[order(prot$Gene)]
data <- as.data.frame(data)
head(data)
```

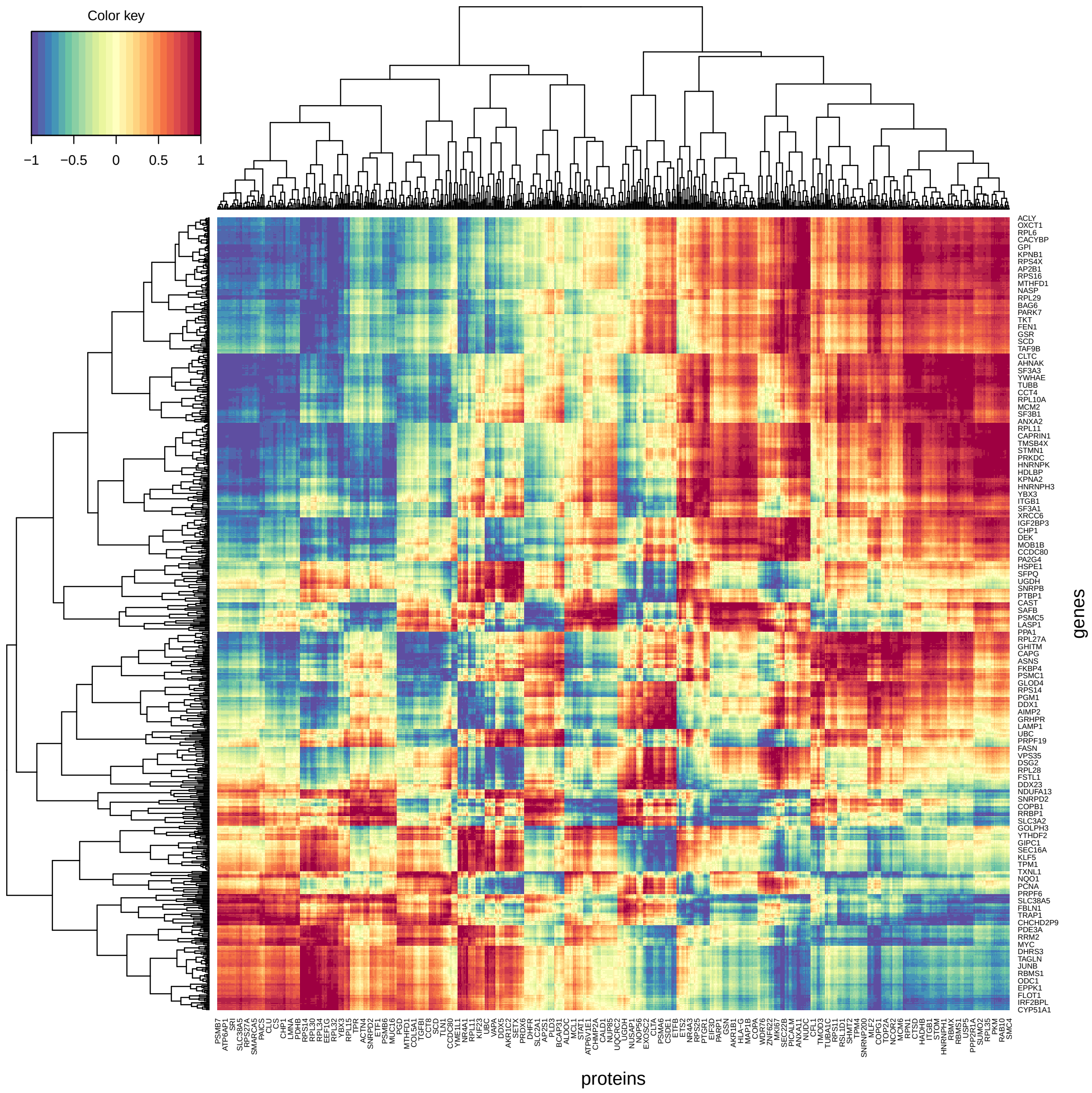
```
FALSE          diff_genes   diff_prot
FALSE ACAA2  -0.004325622 -0.26270422
FALSE ACACA  -0.146697723  0.38480990
FALSE ACADVL  0.004302090  0.87990185
FALSE ACIN1  -0.029715372  0.06214754
FALSE ACLY   -0.077486686 -1.32702673
FALSE ACP1   -0.047773211  0.28450250
```

```
# We do ascatter plot of gene expression and protein abundance and estimate the
# Pearson correlation between them
sp <- ggscatter(data, x = "diff_genes", y = "diff_prot", #mention data and axis
                add = "reg.line",  # Add regression line
                add.params = list(color = "red", fill = "lightgray"), # Customize regression line
                conf.int = TRUE # Add confidence interval
)+ stat_cor(method = "pearson", label.x = 3, label.y = 30)# Add correlation coefficient
sp
```



We can additionally look into the **correlation** in the expression between Gene Expression and the Abundance of the corresponding Proteins across samples.

```
## Correlation Plot
cim(cor(t(proc_gene_data),
        t(proc_prot_data)),
    xlab = "proteins", ylab = "genes")
```



Clustering Analysis

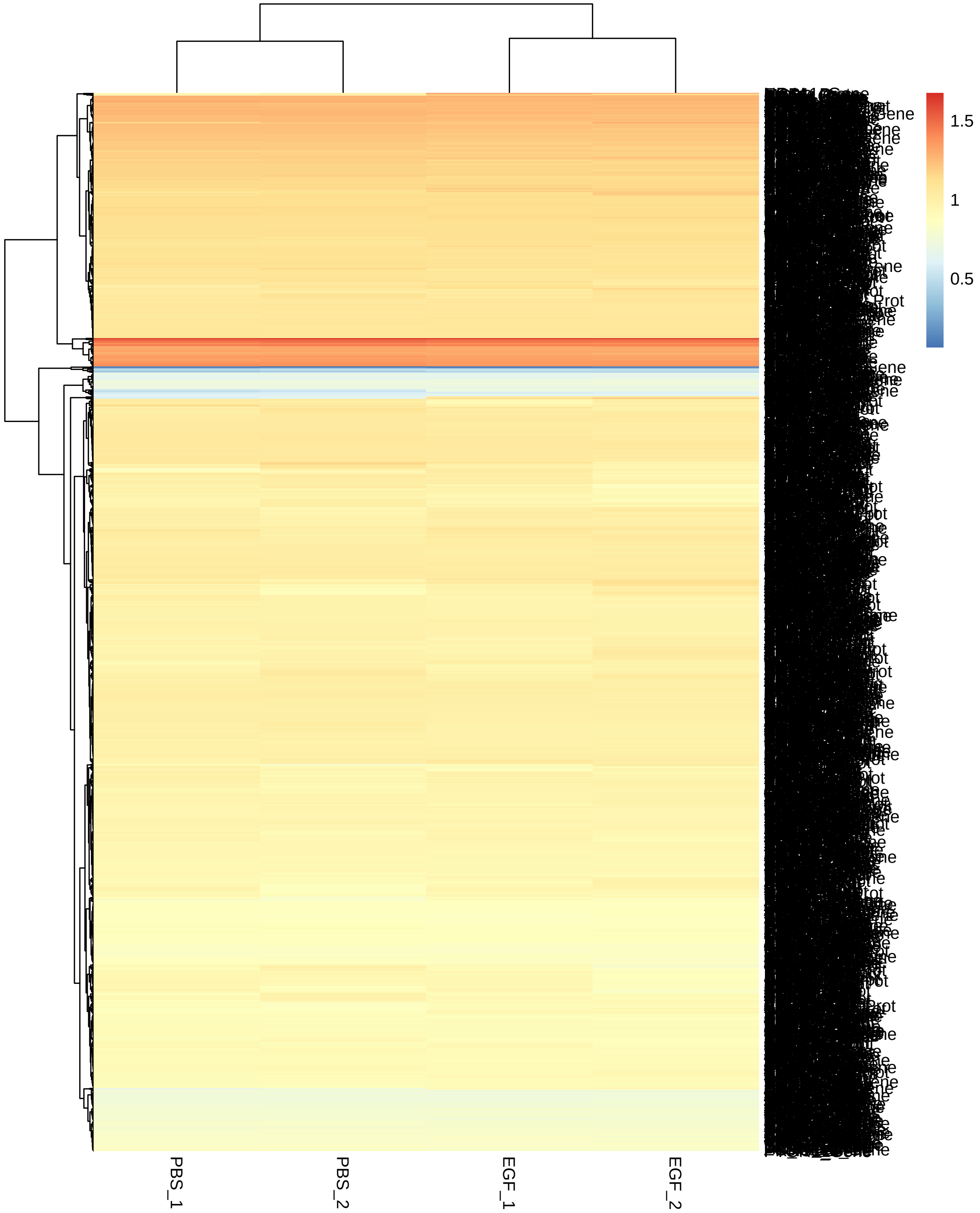
We perform **Clustering** in order to identify and group sets of samples which have similar characteristics. Here we perform two types of clustering analysis:

> **Concatenated clustering:** where combine the multi-omics data into one matrix or search for the shared structure, followed by the final clustering.

```r
# Concatenate the data and cluster the samples
conc_data <- rbind(scale(x = proc_gene_data, center = FALSE),
                   scale(x = proc_prot_data, center = FALSE))
rownames(conc_data) <- c(paste0(rownames(proc_gene_data), "_Gene"),
                         paste0(rownames(proc_prot_data), "_Prot"))
dim(conc_data)
```

```
FALSE [1] 1528    4
```

```
pheatmap(mat = conc_data, cluster_cols = TRUE, cluster_rows = TRUE)
```



Clustering of Clusters: where we obtain the clustering information fromeach omics dataset first and follow by the final clustering. For this we can rely on the NEMO R-Package.

```
# Create Omics List NEMO object and do the clustering
omic1 <- scale(x = proc_gene_data, center = FALSE)
omic2 <- scale(x = proc_prot_data, center = FALSE)

omics.list = list(omic1, omic2)

clustering = nemo.clustering(omics.list = omics.list, num.clusters = 2,
                             num.neighbors = 2) # supervised
print(clustering)
```

```
FALSE PBS_1 PBS_2 EGF_1 EGF_2
FALSE     1     1     2     2
```

```
clustering = nemo.clustering(omics.list = omics.list, num.clusters = NA,
                             num.neighbors = 2) # unsupervised
```

```
FALSE [1] 4 4
```

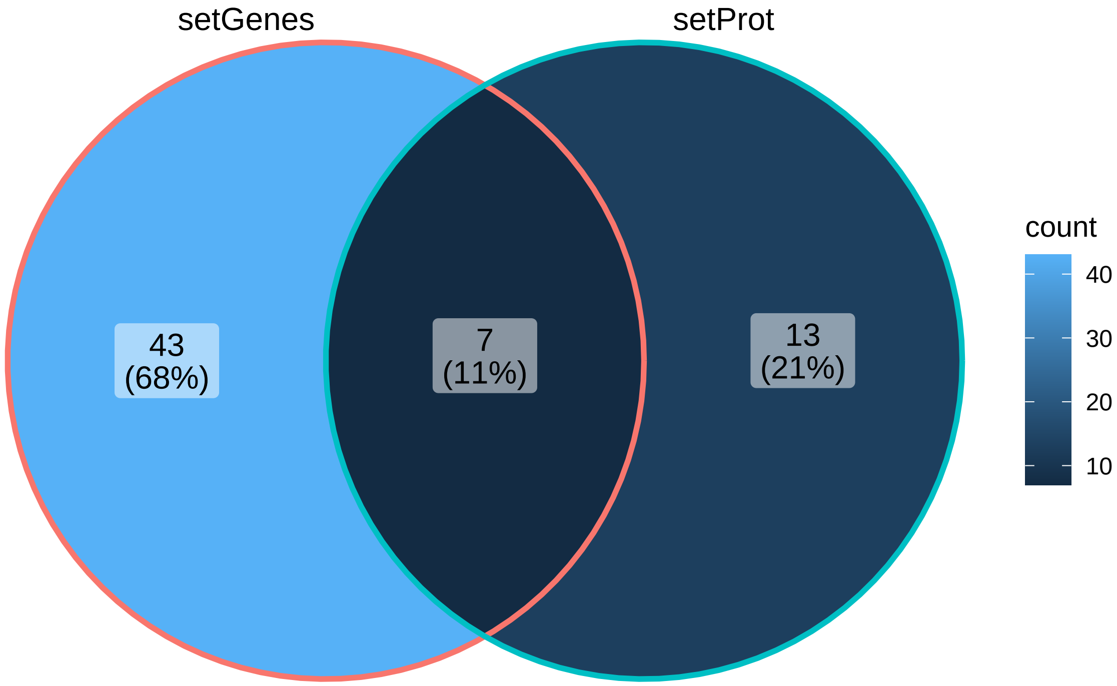
```
print(clustering)
```

```
FALSE PBS_1 PBS_2 EGF_1 EGF_2
FALSE     1     1     2     2
```

Pathway Analysis

Gene Set Enrichment Analysis (GSEA) is used to estimate **significantly regulated Pathway Sets**. We can perform GSEA on both differential gene expression as well as differential abundance data. From the individual analyses, we can then identify a consensus set of **significantly regulated pathways**.

```
# Loading the Pathway Sets
# MSigDB: http://www.gsea-msigdb.org/gsea/msigdb/collections.jsp#C2
load(file = "../Data/reactome_genelist.RData")
# Pathway Analysis from Differential Gene Expression Data
stats <- ttop_dge$logFC
names(stats) <- ttop_dge$external_gene_name
gseaGenes <- fgseaSimple(pathways = genelist, stats = stats, nperm = 10000,
                         minSize = 5, maxSize = Inf)
# Pathway Analysis from Differential Protein Abundance Data
stats <- ttop_prot$EGF_60_vs_PBS_60_diff
names(stats) <- ttop_prot$Gene
gseaProt <- fgseaSimple(pathways = genelist, stats = stats, nperm = 10000,
                        minSize = 5, maxSize = Inf)
# Identifying Pathway Sets regulated on both sets (padj<=0.05)
setGenes <- gseaGenes$pathway[which(gseaGenes$padj<=0.05)]
setProt <- gseaProt$pathway[which(gseaProt$padj<=0.05)]
x <- list(setGenes = setGenes, setProt = setProt)
ggVennDiagram(x)
```



```
print(intersect(x = setGenes, y = setProt))
```

```
FALSE [1] "REACTOME_EUKARYOTIC_TRANSLATION_ELONGATION"
FALSE [2] "REACTOME_SRP_DEPENDENT_COTRANSLATIONAL_PROTEIN_TARGETING_TO_MEMBRANE"
FALSE [3] "REACTOME_NUCLEAR_EVENTS_KINASE_AND_TRANSCRIPTION_FACTOR_ACTIVATION"
FALSE [4] "REACTOME_SELENOAMINO_ACID_METABOLISM"
FALSE [5] "REACTOME_EUKARYOTIC_TRANSLATION_INITIATION"
FALSE [6] "REACTOME_ACTIVATION_OF_THE_MRNA_UPON_BINDING_OF_THE_CAP_BINDING_COMPLEX_AND_EIFS_AND_SUBSEQU
FALSE [7] "REACTOME_NONSENSE_MEDIATED_DECAY_NMD"
```

Functional Modules

Identification of Functional Protein Interaction Modules with BioNet R-Package.

> Obtaining the *p-value* scores from the Differential Gene Expression and Differential Protein Abundance Data.

```
# Filtering DGE and DPA for common genes and retreiving p-values
data <- matrix(data = , nrow = length(common_genes), ncol = 2)
rownames(data) <- common_genes[order(common_genes)]
colnames(data) <- c("diff_genes", "diff_prot")
data[, 1] <- dge$PValue[order(dge$external_gene_name)]
data[, 2] <- prot$EGF_60_vs_PBS_60_p.val[order(prot$Gene)]
data <- as.data.frame(data)
head(data)
```

```
FALSE        diff_genes   diff_prot
FALSE ACAA2  0.94466877 0.340070467
FALSE ACACA  0.01048237 0.299471584
FALSE ACADVL 0.93791423 0.018131914
FALSE ACIN1  0.59392672 0.834295541
FALSE ACLY   0.13921717 0.001770302
FALSE ACP1   0.36572053 0.352015888
```

Obtaining protein interactions from the OmniPathR R-Package and creating an *igraph* object from the retreived interactions.

```
# Obtaining interactions from OmniPath
interactions <- import_omnipath_interactions()
interactions <- unique(as.data.frame(interactions[, 3:4]))
head(interactions)
```

```
FALSE   source_genesymbol target_genesymbol
FALSE 1             CALM2             TRPC1
FALSE 2             CALM1             TRPC1
FALSE 3             CALM3             TRPC1
FALSE 4              CAV1             TRPC1
FALSE 5              DRD2             TRPC1
FALSE 6              MDFI             TRPC1
```

```
# Transforming the obtained network into an _igraph_ object.
g <- graph_from_data_frame(d = interactions, directed = TRUE)
g <- as_graphnel(graph = g)
g
```

```
FALSE A graphNEL graph with directed edges
FALSE Number of Nodes = 8155
FALSE Number of Edges = 39429
```

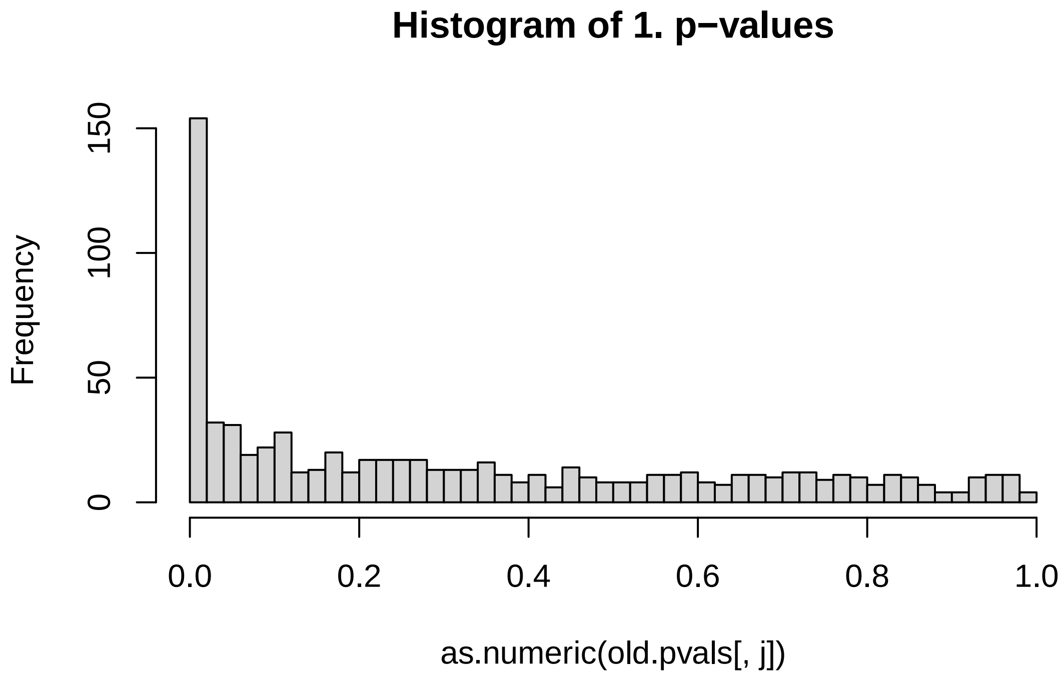
Creating a subgraph with the nodes given in the the differential gene and protein expression data and including their direct neighbors.

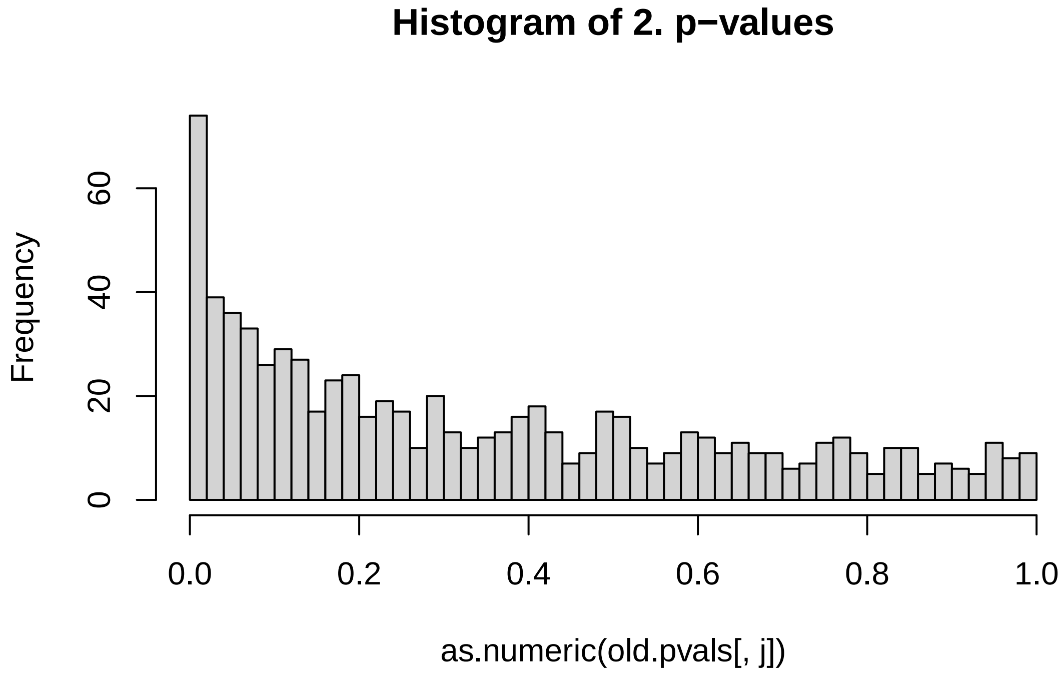
```
subnet <- subNetwork(rownames(data), g)
subnet
```

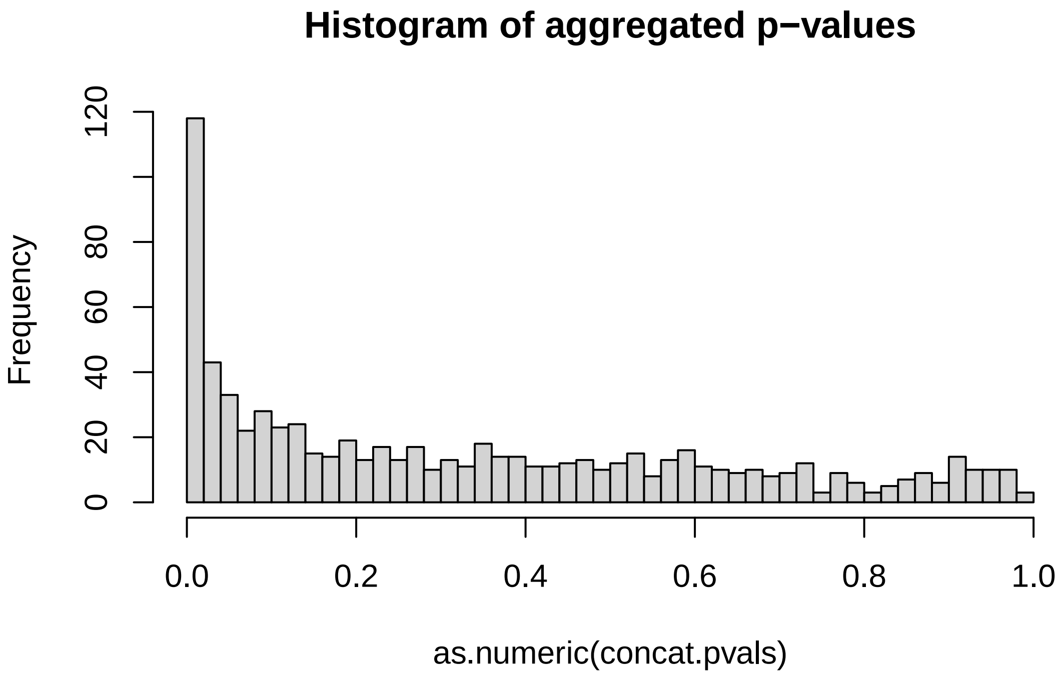
```
FALSE A graphNEL graph with directed edges
FALSE Number of Nodes = 540
FALSE Number of Edges = 298
```

Aggregating the p-values from the DGE and DPA data.

```
pvals <- cbind(data$diff_genes, data$diff_prot)
rownames(pvals) <- rownames(data)
pval <- aggrPvals(pvals, order = 2, plot = TRUE)
```


Histogram of 1. p−values
Frequency
0
50
100
150
0.0
0.2
0.4
0.6
0.8
1.0
as.numeric(old.pvals[, j])


Histogram of 2. p−values
Frequency
0
20
40
60
0.0
0.2
0.4
0.6
0.8
1.0
as.numeric(old.pvals[, j])

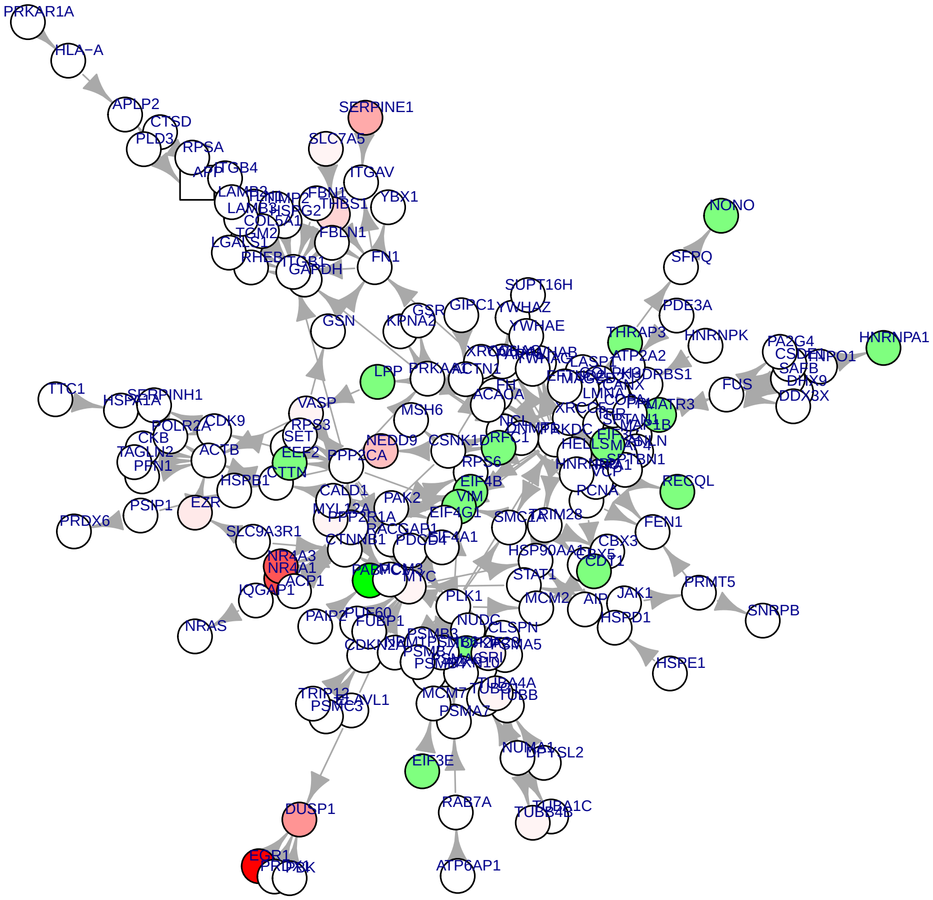


Obtaining the Functional Network Modules.

```
fb <- fitBumModel(pval, plot = FALSE)
scores <- scoreNodes(subnet, fb, fdr = 0.5)
module <- runFastHeinz(g, scores)
```

Plotting the resulting netwoks.

```
logFC <- dge$logFC
names(logFC) <- dge$external_gene_name
plotModule(module, scores = scores, diff.expr = logFC)
```


PRKAR1A
HLA−A
APLP2
CTSD
PLD3
RPSA
APP
ITGB4
SERPINE1
SLC7A5
ITGAV
LAMB2
TIMP2
FBN1
THBS1
YBX1
LAMB3
HSPG2
COL5A1
TGM2
FBLN1
LGALS1
RHEB
ITGB1
GAPDH
FN1
NONO
SFPQ
SUPT16H
GSN
GSR
GIPC1
YWHAZ
KPNA2
PDE3A
YWHAE
THRAP3
HNRNPK
PA2G4
HNRNPA1
CSDE1
TNPO1
ATP2A2
YWHAB
LASP1
YWHAG
SAFB
DHX9
LPP
PRKAA1
ACTN1
KHDRBS1
CANX
FUS
DDX3X
MAGED2
LMNA
FH
ACACA
COPA
MATR3
TTC1
SERPINH1
HSPA1A
VASP
MSH6
XRCC5
CDK9
POLR2A
RPS3
NCL
SPTAN1
MAP1B
CKB
SET
DNMT1
PRKDC
TAGLN2
ACTB
EEF2
PPP2CA
NEDD9
CSNK1D
RFC1
HELLS
MAP4
ANLN
PFN1
CTTN
RPS6
SPTBN1
HSPB1
HNRNPA1
VCP
RECQL
EIF4B
CALD1
PCNA
PSIP1
EZR
VIM
PAK2
MYL12A
EIF4G1
PPP2R1A
SMC1A
TRIM28
PRDX6
FEN1
SLC9A3R1
RACGAP1
CTNNB1
PDCD4
EIF4A1
CBX3
HSP90AA1
CBX5
NR4A3
NR4A1
CDT1
ACP1
MCM3
MYC
STAT1
IQGAP1
PRMT5
JAK1
SNRPB
MCM2
AIP
PLK1
PAIP2
PUF60
FUBP1
HSPD1
NRAS
NUDC
CLSPN
PSMB3
PSMA5
CDKN2A
PSMB7
SRI
HSPE1
TUBA4A
TRIP12
ELAVL1
MCM7
TUBB
PSMC3
PSMA7
NUMA1
DPYSL2
EIF3E
RAB7A
TUBA1C
TUBB4B
DUSP1
EGR1
ATP6AP1