Analyzing functional data using R

Author: Pablo Fonseca

February 5, 2021

Why use R to search for positional and functional candidate genes?

In the last tutorials, we learned how to search for positional and functional candidate genes using several softwares. This is a great strategy. However, during the process, several output files are created and we must manage these files in a very cautious way. We must convert different outputs to use as input files in other software, combine different results to select candidate genes, to create plots in order to summarize the results. The larger number of genes in the input files and the larger amount of information retrieved using the software showed up to now, can increase the complexity of those processes. R is a good alternative to manage and combine multiple results in a very straightforward way. There are several packages available in R that performs the same (and additional) analyses. Work with this kind of analysis within R allows the design of a pipeline to combine different steps and to create outputs in the same environment. These characteristics result in an analysis that is performed in a much more efficient way and simplify the management process of the intermediary files. Additionally, using R, it is possible to create new kind of analysis and /or to adapt previous approaches. Here, we will see some possibilities to use R to identify positional and functional candidate genes.

Packages to be installed

Before to star, the following packages must be installed:

biomaRt

WebGestaltR

meshes

MeSH.Bta.eg.db

STRINGdb

GALLO

All the abivementioned packages can be found in the CRAN or Bioconductor repositories. The code to install and to load each one of these packages is shown before the its use in the present tutorial.

##Package: biomaRt

The package biomaRt "provides an interface to a growing collection of databases implementing the BioMart software suite (http://www.biomart.org). The package enables retrieval of large amounts of data in a uniform way without the need to know the underlying database schemas or write complex SQL queries". Therefore, it is possible to retrieve all the information available using the online version of Biomart, using R.

Here, we will use the same input file used in the online version of Biomart. The file to be imported in R is called "input_candidate_markers.txt".

```
#The following command can  be used to install the package
#(you just need to remove the comment character #)
#BiocManager::install("biomaRt")

#Before start, we must set the work directory
setwd("/Volumes/Backup Plus/post_doc/courses/course_UFBA_2021")
```

```r
#Loading the input files
input.regions<-read.table("Candidate_markers_GWAS.csv", h=T, sep="\t")

#Checking the imported file
head(input.regions)
```

```
##    Associated.marker SNP.reference                                      Trait
## 1 BovineHD0500013194   rs134152961 blood hormone levels of inhibin at 4 months
## 2 BovineHD0500013195   rs109440480 blood hormone levels of inhibin at 4 months
## 3 BovineHD0500013211   rs109628263 blood hormone levels of inhibin at 4 months
## 4 BovineHD0500013212   rs136159056 blood hormone levels of inhibin at 4 months
## 5 BovineHD0500013217   rs109160518 blood hormone levels of inhibin at 4 months
## 6 BovineHD0500035412   rs109344578 blood hormone levels of inhibin at 4 months
##   CHR       BP  P.value              Breed            Reference
## 1   5 45853150 7.64e-06 Tropical Composite Fortes et al. (2013)
## 2   5 45857076 7.64e-06 Tropical Composite Fortes et al. (2013)
## 3   5 45947079 1.50e-09 Tropical Composite Fortes et al. (2013)
## 4   5 45958046 1.50e-09 Tropical Composite Fortes et al. (2013)
## 5   5 45980892 1.50e-09 Tropical Composite Fortes et al. (2013)
## 6   5 46039437 5.38e-07 Tropical Composite Fortes et al. (2013)
```

```r
#Loading the package
library(biomaRt)
```

After loading the input file and loading the package, it is possible to start the search of genes within our selected interval.

1) The first step is to choose the database and the organism to choose the genes

```r
#The following command shows the databases that are available to retrive information
#listMarts()

#We will choose ENSEMBL_MART_ENSEMBL, which correspond to Ensembl Genes 91
#It is necessary to creat a object of the type "Mart" to load the database information.
#This is performed with the following command:
mart <- useMart("ENSEMBL_MART_ENSEMBL")

#Once the object mart was created using useMart() function, we need to select
#the organism. The following comand can be used to identify the correspondent
#argument to each organism.

#listDatasets(mart)

#The function useDataset() can be used to insert the organism
#infortation in the mart object.
mart <- useDataset("btaurus_gene_ensembl", mart)
```

2) Now, it is time to define the filters and attributes to be retrieved from the database

```r
#The following command can be used to list the filters available to use
#in the retriving process

#listFilters(mart)

#Here, we will use the chromosome region filter ("chromosomal_region") to retrive
#the genes mapped in the intervals selected
```

```
filter<-"chromosomal_region"

#The following command will be used to creat the chromosomal coordinates using a 500Kb
#upstream and downstream for each slected marker

value<-paste(input.regions$CHR, ":", (input.regions$BP)-500000, ":",
             (input.regions$BP)+500000, sep="")

head(value)
```

```
## [1] "5:45353150:46353150" "5:45357076:46357076" "5:45447079:46447079"
## [4] "5:45458046:46458046" "5:45480892:46480892" "5:45539437:46539437"
```

```
#The following command allows to list all the attirbutes possible to be retrived
#from Biomart

#listAttributes(mart)

#The attributes to be retrived will be inserted in a object
attributes <- c("ensembl_gene_id","hgnc_symbol","external_gene_name",
                "chromosome_name","start_position","end_position",
                "entrezgene_id")
```

3) Now, after defining the filters and the attributes, it is time to retrieve the information

```
#At this point, we need to inform the objects created in the previous steps as arguments
#of the function getBM()
all.genes <- getBM(attributes=attributes, filters=filter, values=value, mart=mart)

#Checking the output
str(all.genes)
```

```
## 'data.frame':    280 obs. of  7 variables:
##  $ ensembl_gene_id   : chr  "ENSBTAG00000042334" "ENSBTAG00000049087" "ENSBTAG00000050378" "ENSBTAG0
##  $ hgnc_symbol       : chr  "" "" "" "" ...
##  $ external_gene_name: chr  "U6" "" "" "" ...
##  $ chromosome_name   : int  13 13 13 13 13 13 13 13 13 13 ...
##  $ start_position    : int  69128984 69239616 69247298 69373089 69378600 69721975 69723092 69771141 6
##  $ end_position      : int  69129090 69241015 69250310 69374544 69379571 69779394 69723198 69771247 6
##  $ entrezgene_id     : int  NA NA NA NA 532376 534799 NA NA NA 281987 ...
```

```
head(all.genes)
```

```
##      ensembl_gene_id hgnc_symbol external_gene_name chromosome_name
## 1 ENSBTAG00000042334                             U6              13
## 2 ENSBTAG00000049087                                             13
## 3 ENSBTAG00000050378                                             13
## 4 ENSBTAG00000048871                                             13
## 5 ENSBTAG00000003396                           MAFB              13
## 6 ENSBTAG00000007960                           TOP1              13
##   start_position end_position entrezgene_id
## 1       69128984     69129090            NA
## 2       69239616     69241015            NA
## 3       69247298     69250310            NA
## 4       69373089     69374544            NA
## 5       69378600     69379571        532376
```

```
## 6       69721975     69779394        534799
```

```
#Saving the ouput

write.table(all.genes, file="output_biomart_genes_within_interval.txt",
            row.names=F, sep="\t", quote=F)
```

Using the command listed above, it is possible to obtain the same results obtained in the online version of biomart. Notice that it is possible to run these commands for all the organisms available in the Ensembl database, for all the filters and attributes. Check the help material for each function to obtain more information. For example "?getBM()".

The manual of biomaRt can be found here:

https://bioconductor.org/packages/release/bioc/manuals/biomaRt/man/biomaRt.pdf

##Retrieving the positional candidate genes and QTLs within selected interval using Genomic functional Annotation in Livestock for positional candidate LOci (GALLO)

Recently, we are working in the development of a R package for functional annotation of positional candidate loci. The initial name of this package is Genomic functional Annotation in Livestock for positional candidate LOci (GALLO). Among other functions, this package allows the identification of positional candidate genes, QTL data mining, QTL enrichment, plot results, etc.

Before to load the package it is necessary to install the package. However, the package is not available for all the scientific community because it is still under development.

The source code to install GALLO is deposited in an GitHub repository and can be installed directly using the following code:

```
#install.packages("GALLO")
```

After this step we can load the package.

```
library(GALLO)
```

Retrieving the positional candidate genes

We can use a function of GALLO to retrive positional candidate genes. The function is called "find_genes_qtls_around_markers". For this function, you should inform some arguments:

find_genes_qtls_around_markers(db_file, marker_file, method = c("gene","qtl"), marker = c("snp", "haplotype"), interval = 0, nThreads = NULL)

db_file: The dataframe created using the _import_gff_gtf function.

marker_file: The file with the SNP or haplotype positions. Detail: For SNP files, you must have a column called "CHR" and a column called "BP" with the chromosome and base pair position, respectively. For the haplotype, you must have three columns: "CHR", "BP1" and "BP2". All the columns names are in uppercase.

method: "gene" or "qtl". If "gene" method is selected, a .gtf files must be provided for the db_file argument. On the other hand, if the method "qtl" is selected, a .gff file from Animal QTLdb must be provided for the db_file argument.

marker: "snp" or "haplotype". If "snp" option is selected, a dataframe with at least two mandatory columns (CHR and BP) must be provided for the marker_file argument. On the other hand, if "haplotype" option is selected, a dataframe with at least three mandatory columns (CHR, BP1 and BP2) must be provided for the marker_file argument. Any additional column can be included in the dataframe provided for the marker_file argument, for example, a column informing the study, model, breed, etc. from which the results were obtained

interval: The interval in base pair which can be included upstream and downstream from the markers or haplotype coordinates

nThreads: Number of threads to be used in the analysis

All positions must be informed using base pairs (not Kb or Mb).

Important: This function is a beta version. Therefore, errors or bugs might be found. Use with care.

```
#Importing gff file
genes.inp <- import_gff_gtf(db_file="Bos_taurus.UMD3.1.94.gtf",file_type="gtf")
```

```
#Running the function to search genes in a interval of 500Kb upstream and downstream
#from each marker
genes.interval<-find_genes_qtls_around_markers(db_file=genes.inp,
                                               marker_file=input.regions,
                                               method="gene",
                                               marker="snp",
                                               interval=500000)
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
head(genes.interval)
```

```
##    Associated.marker SNP.reference                              Trait CHR
## 1 BovineHD0500013025   rs132874802 Scrotal circumference at 420 days   5
## 2 BovineHD0500013026   rs109156482 Scrotal circumference at 420 days   5
## 3 BovineHD0500013033   rs109284796 Scrotal circumference at 420 days   5
## 4 BovineHD0500013025   rs132874802 Scrotal circumference at 420 days   5
## 5 BovineHD0500013026   rs109156482 Scrotal circumference at 420 days   5
## 6 BovineHD0500013033   rs109284796 Scrotal circumference at 420 days   5
##         BP  P.value   Breed               Reference chr start_pos  end_pos
## 1 45258841 8.17e-06 Canchim Buzanskas et al. (2017)   5  44765461 44793485
## 2 45260223 8.17e-06 Canchim Buzanskas et al. (2017)   5  44765461 44793485
## 3 45289852 5.66e-06 Canchim Buzanskas et al. (2017)   5  44765461 44793485
## 4 45258841 8.17e-06 Canchim Buzanskas et al. (2017)   5  44886446 44886802
## 5 45260223 8.17e-06 Canchim Buzanskas et al. (2017)   5  44886446 44886802
## 6 45289852 5.66e-06 Canchim Buzanskas et al. (2017)   5  44886446 44886802
##   width strand            gene_id gene_name         gene_biotype
## 1 28025      - ENSBTAG00000007323     CPSF6       protein_coding
## 2 28025      - ENSBTAG00000007323     CPSF6       protein_coding
## 3 28025      - ENSBTAG00000007323     CPSF6       protein_coding
## 4   357      - ENSBTAG00000002741      <NA> processed_pseudogene
## 5   357      - ENSBTAG00000002741      <NA> processed_pseudogene
## 6   357      - ENSBTAG00000002741      <NA> processed_pseudogene
```

Note that the output was created merging the columns from your input files with the gene coordinates obtained in the gtf file. Therefore, each interval is repeated in time. Where n is equal the number of genes within this interval.

Comparing genes among groups/studies

Now we can compare the number of genes shared between the studies. For that we will use 2 functions: overlapping_among_groups and plot_overlapping.

```
#Now we will run the first function
out.overlapping<-overlapping_among_groups(genes.interval,x="Reference",y="gene_id")
```

```
#Check the results
out.overlapping
```

```
## $N
##                          Buzanskas et al. (2017) Fortes et al. (2013)
## Buzanskas et al. (2017)                     493                  142
## Fortes et al. (2013)                         69                 5163
##
## $percentage
##                          Buzanskas et al. (2017) Fortes et al. (2013)
## Buzanskas et al. (2017)                    1.00                 0.29
## Fortes et al. (2013)                       0.01                 1.00
##
## $combined
##                          Buzanskas et al. (2017) Fortes et al. (2013)
## Buzanskas et al. (2017) "493 (1)"               "142 (0.29)"
## Fortes et al. (2013)    "69 (0.01)"             "5163 (1)"
```

Now we can plot the results using the function plot_overlapping using the following arguments:

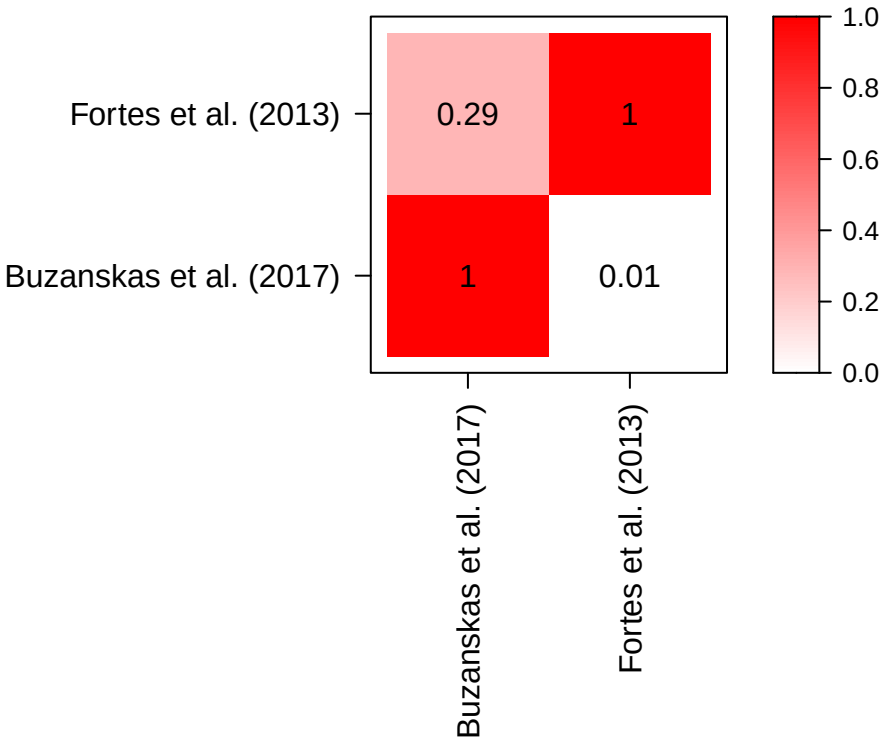
overlapping_matrix: The output from overlapping_among_groups

nmatrix: An interger from 1 to 3 indicating wich matrix will be used to plot the overlapping, where: 1) A matrix with the number of overllaping data; 2) A matrix with the percentage of overlapping; 3) A matrix with the combination of the two previous one

ntext: An interger from 1 to 3 indicating wich matrix will be used as the text matrix for the heatmap, where: 1) A matrix with the number of overllaping data; 2) A matrix with the percentage of overlapping; 3) A matrix with the combination of the two previous one

group: A vector with the size of groups. This vector will be plotted as row and column names in the heatmap

```
plot_overlapping(overlapping_matrix=out.overlapping, nmatrix=2,ntext=2,
group = unique(genes.interval$Reference))
```



Retrieving QTLs

The find_genes_qtls_around_markers can also be used to compare our regions of interest with QTLs/asscoiations already reported in QTLdb.

```
#Importing the input file
input.regions<-read.table("Candidate_markers_GWAS.csv", h=T, sep="\t")
```

```
#Importing gff file
qtl.inp <- import_gff_gtf(db_file="QTL_UMD_3.1.gff",file_type="gff")
```

```
#Running the function to search QTLs in a interval of 500Kb upstream and downstream
#from each marker
qtls.interval<-find_genes_qtls_around_markers(db_file=qtl.inp,
                                              marker_file=input.regions,
                                              method="qtl",
                                              marker="snp",
                                              interval=500000)
```

```
head(qtls.interval)
```

```
##    Associated.marker SNP.reference                                    Trait
## 1 BovineHD0500016138   rs136816142 blood hormone levels of inhibin at 4 months
## 2 BovineHD0500016144   rs133559518 blood hormone levels of inhibin at 4 months
## 3 BovineHD0500016153   rs134428680 blood hormone levels of inhibin at 4 months
## 4 BovineHD0500016154   rs137391646 blood hormone levels of inhibin at 4 months
## 5 BovineHD0500016160   rs135647468 blood hormone levels of inhibin at 4 months
## 6 BovineHD0500016162   rs134828282 blood hormone levels of inhibin at 4 months
##   CHR       BP  P.value              Breed            Reference chr
## 1   5 56968913 2.80e-06 Tropical Composite Fortes et al. (2013)   5
## 2   5 56996576 2.80e-06 Tropical Composite Fortes et al. (2013)   5
## 3   5 57056862 1.57e-06 Tropical Composite Fortes et al. (2013)   5
```

```
## 4   5 57059198 1.57e-06 Tropical Composite Fortes et al. (2013)   5
## 5   5 57095707 5.48e-06 Tropical Composite Fortes et al. (2013)   5
## 6   5 57103035 1.57e-06 Tropical Composite Fortes et al. (2013)   5
##       database QTL_type start_pos  end_pos Association_type QTL_ID
## 1 Animal QTLdb     Milk  57360743 57360843      Association  34690
## 2 Animal QTLdb     Milk  57360743 57360843      Association  34690
## 3 Animal QTLdb     Milk  57360743 57360843      Association  34690
## 4 Animal QTLdb     Milk  57360743 57360843      Association  34690
## 5 Animal QTLdb     Milk  57360743 57360843      Association  34690
## 6 Animal QTLdb     Milk  57360743 57360843      Association  34690
##         trait_ID       breed           Name Abbrev     Model       Test_Base
## 1 Milk C14 index rs134688325 Milk C14 index   MC14 Mendelian Experiment-wise
## 2 Milk C14 index rs134688325 Milk C14 index   MC14 Mendelian Experiment-wise
## 3 Milk C14 index rs134688325 Milk C14 index   MC14 Mendelian Experiment-wise
## 4 Milk C14 index rs134688325 Milk C14 index   MC14 Mendelian Experiment-wise
## 5 Milk C14 index rs134688325 Milk C14 index   MC14 Mendelian Experiment-wise
## 6 Milk C14 index rs134688325 Milk C14 index   MC14 Mendelian Experiment-wise
##   pubmed_id     p_value bayes_value Flank_Markers
## 1  25511820 0.062391488  Suggestive        jersey
## 2  25511820 0.062391488  Suggestive        jersey
## 3  25511820 0.062391488  Suggestive        jersey
## 4  25511820 0.062391488  Suggestive        jersey
## 5  25511820 0.062391488  Suggestive        jersey
## 6  25511820 0.062391488  Suggestive        jersey
```

```
unique(qtls.interval$QTL_type)
```

```
## [1] "Milk"             "Production"       "Reproduction"     "Health"
## [5] "Exterior"         "Meat_and_Carcass"
```

Note that, as well as the gene searching output, the output was created merging the columns from your input files with the gene coordinates obtained in the gtf file. Therefore, each interval is repeated n times. Where n is equal the number of genes within this interval.

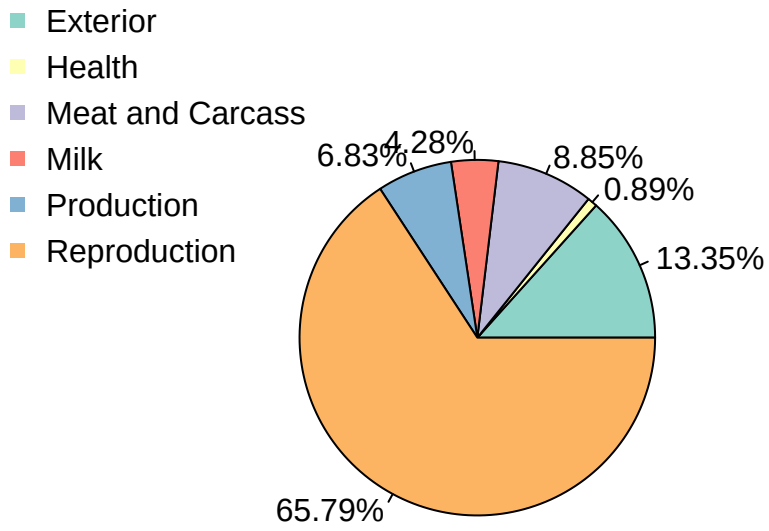
The same analyses can be performed for haplotype data. You only need to change the argument informed to "marker=" and use the option "haplotype" intead of "snp" to perform the analysis with haplotypes.

We can use the function **plot_qtl_info** to obtain a pie plot with all the percentages of each QTL type. In order to run this function, it is necessary to inform the following arguments:

qtl_file: The output from the find_genes_qtls_around_markers function

qtl_plot: "qtl_type" or "qtl_name". Now, we will choose "qtl_type"

```
#Getting the QTL plot
par(mar=c(5,10,5,5))
plot_qtl_info(qtl_file=qtls.interval, qtl_plot = "qtl_type", cex=1)
```



Additionally, it is possible to obtain a barplot with all the QTL names within the selected QTL types. In order to obtain this plot, we must run the same function "plot_qtl_info". However, now we will use some additional arguments:

qtl_file: The output from the find_genes_qtls_around_markers function

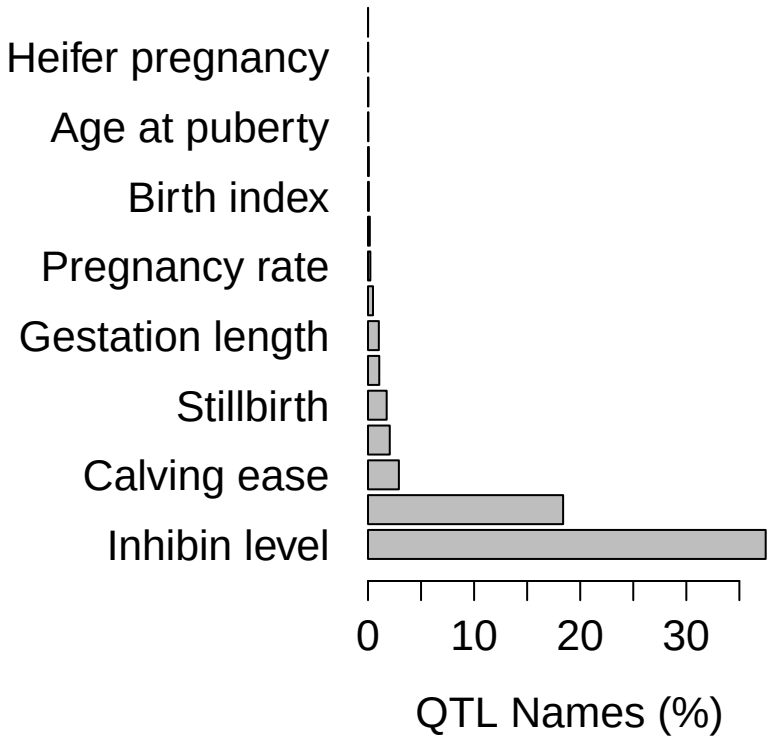
qtl_plot: "qtl_type" or "qtl_name". Now, we will choose "qtl_name"

n: "all" or a number of QTLs to be plotted

qtl_class: "Milk", "Production", "Reproduction", "Health", "Exterior", "Meat_and_Carcass"

Let's plot the reproduction associated QTLs:

```
#First, we will set some graphical parameters using th option par().
#Using the option mar(), it is possible to set the margins around
#the plot canvas informing a vector with the c(bottom,left,top,right)
#margins.
par(mar=c(5,20,2,2))
#Plotting the information.
plot_qtl_info(qtl_file=qtls.interval, qtl_plot = "qtl_name",
n = "all", qtl_class = "Reproduction",
cex.lab=1.3, cex.names=1.3, cex.axis=1.3)
```



The simple bias of investigation for some traits (such as milk production related traits in the QTL database for cattle) may result in a larger proportion of records in the database. Consequently, the simple investigation of the proportion of each QTL type might not be totally useful. In order to reduce the impact of this bias, a QTL enrichment analysis can be performed. The QTL enrichment analysis performed by GALLO package is in a hypergeometric test using the number of annoatted QTLs within the candidate regions and the total number of the same QTL in the QTL database.

qtl_enrich

qtl_enrich(qtl_db, qtl_file, qtl_type = c("QTL_type", "trait"),enrich_type = c("genome", "chromosome"), chr.subset = NULL, nThreads = NULL, padj = c("holm", "hochberg","hommel", "bonferroni", "BH", "BY", "fdr", "none"))

qtl_db: The .gff file that can be downloaded from Animal QTlLdb

qtl_file: The output from find_genes_qtls_around_markers function

qtl_type: A character indicating which type of enrichment will be performed. "QTL_type" indicates that the enrichment processes will be performed for the QTL classes, while "Name" indicates that the enrichment analysis will be performed for each trait individually.

enrich_type: A character indicating if the enrichment analysis will be performed for all the chromosomes ("genome") or for a subset of chromosomes ("chromosome"). If the "genome" option is selected, the results reported are the merge of all chromosomes.

chr.subset: If enrich_type is equal "chromosome", it is possible to define a subset of chromosomes to be analyzed. The default is equal NULL. Therefore, all the chromosomes will be analyzed.

nThreads: The number of threads used.

padj: The algorithm for multiple testing correction to be adopted ("holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none").

As an example, we are going to perform a enrichment analysis for all the QTL information annotated around the candidate markers using a chromosome-based enrichment analysis. The adjusted p-values will be

calculated based on False-Discovery Rate (FDR).

This step might take some minutes to run depending of the user‘s system.

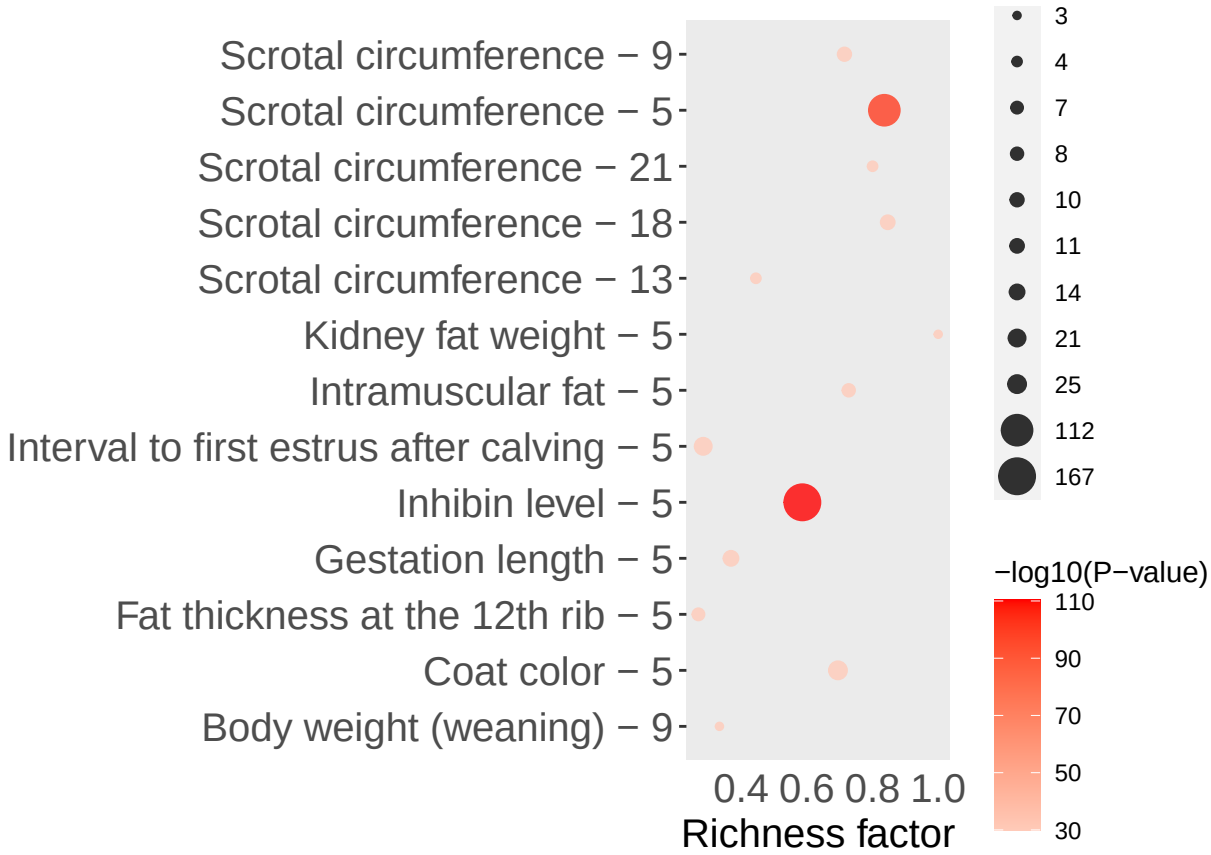
```
#QTL enrichment analysis
out.enrich<-qtl_enrich(qtl_db=qtl.inp,
                       qtl_file=qtls.interval,
                       qtl_type = "Name",
                       enrich_type = "chromosome",
                       chr.subset = NULL,
                       padj ="fdr",nThreads = 1)
```

```
## End of QTL enrichment analysis
```

```
#Filtering enriched QTLs
out.enrich.enrich<-out.enrich[which(out.enrich$adj.pval<0.05),]
dim(out.enrich.enrich)
out.enrich.enrich
```

```
#Truncating adj p-val to maximum at 1e-30
out.enrich.enrich[which(out.enrich.enrich$adj.pval>1e-30),"adj.pval"]<-1e-30

#Creating a new ID
out.enrich.enrich$ID<-paste(out.enrich.enrich$QTL, "-", out.enrich.enrich$CHR, sep=" ")
```

```
#Plotting results
QTLenrich_plot(out.enrich.enrich, x="ID", pval="adj.pval")
```



relationship_plot

relationship_plot(qtl_file, x, y, grid.col = "gray60", degree = 90,canvas.xlim = c(-2, 2), canvas.ylim = c(-2, 2), cex)

qtl_file: The output from find_genes_qtls_around_markers function

x: The first grouping factor, to be plotted in the left hand side of the chord plot

y: The second grouping factor, to be plotted in the left hand side of the chord plot

grid.col: A character with the grid color for the chord plot or a vector with different colors to be used in the grid colors. Note that when a color vector is provided, the lenght of this vector must be equal the number of sectors in the chord plot

degree: A numeric value corresponding to the starting degree from which the circle begins to draw. Note this degree is always reverse-clockwise

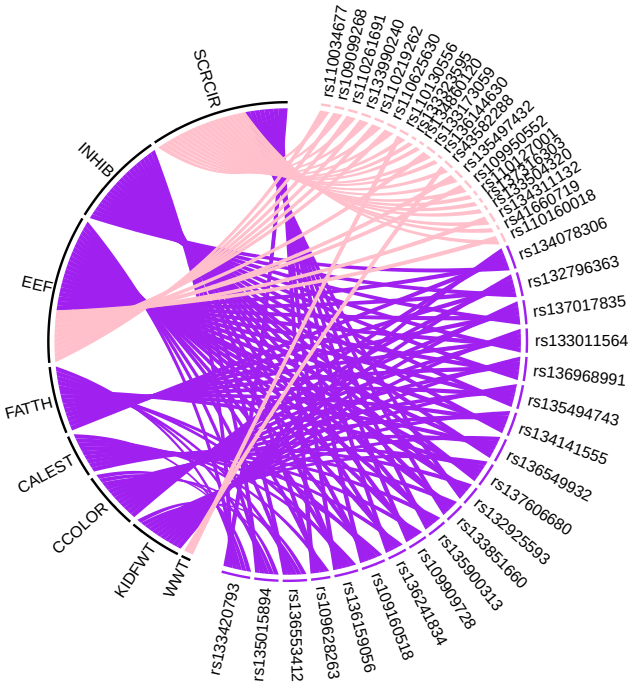
canvas.xlim: The coordinate for the canvas in the x-axis. By default is c(-1,1)

canvas.ylim: The coordinate for the canvas in the y-axis. By default is c(-1,1)

cex: The size of the labels to be printed in the plot

```
#Creating a new ID to filter the top 5 enriched QTLs
qtls.interval$ID<-paste(qtls.interval$Name,"-",qtls.interval$CHR,sep=" ")

out.enrich.enrich<-out.enrich.enrich[
  which(out.enrich.enrich$adj.pval<0.05),]

#Filtering QTL annotation output for only those
#enriched QTLs
out.qtls.filtered<-qtls.interval[
which(qtls.interval$ID%in%out.enrich.enrich$ID),]

#Selection the 20 SNPs with smallest p-values from
#Fortes et al. (2013) and Buzanskas et al. (2017)
out.qtls.filtered<-out.qtls.filtered[order(out.qtls.filtered$P.value),]

snp.list.fortes<-unique(out.qtls.filtered[which(out.qtls.filtered$Reference=="Fortes et al. (2013)"),"SN
snp.list.fortes<-snp.list.fortes[1:20]

snp.list.buzankas<-unique(out.qtls.filtered[which(out.qtls.filtered$Reference=="Buzanskas et al. (2017)
snp.list.buzankas<-snp.list.buzankas[1:20]

snp.list<-c(snp.list.fortes,snp.list.buzankas)

out.qtls.filtered<-out.qtls.filtered[which(out.qtls.filtered$SNP.reference%in%snp.list),]

#Creating color scheme based on the References
out.qtls.filtered[which(
  out.qtls.filtered$Reference=="Fortes et al. (2013)"),
  "color_ref"]<-"purple"

out.qtls.filtered[which(
  out.qtls.filtered$Reference== "Buzanskas et al. (2017)"),
  "color_ref"]<-"pink"

#Creating a color vector filled with black for all the traits abbreviation
#and with the respective colors for each reference
```

```
color.grid<-c(rep("black",
                  length(unique(out.qtls.filtered$Abbrev))),
              out.qtls.filtered[!duplicated(out.qtls.filtered$SNP.reference),"color_ref"])

#Naming the vector
names(color.grid)<-c(unique(
  out.qtls.filtered$Abbrev),unique(
    out.qtls.filtered$SNP.reference))

#Plotting the relationship plot using the grid color created above
relationship_plot(qtl_file=out.qtls.filtered, x="Abbrev",
                  y="SNP.reference", cex=0.5,gap=1,
                  degree = 90, canvas.xlim = c(-2, 2),
                  canvas.ylim = c(-1.5, 1.5),
                  grid.col = color.grid)
```



Using WebGestaltR to run Gene Ontology enrichment analysis

The Gene Ontology enrichment analyses can also be performed using R.

The WebGestaltR package is a very useful package to run several types of enrichment analysis, such as GO terms and metabolic pathway analysis.

For the following tutorial we will use a group of genes that are differentially co-expressed in the endometrium of subfertily and fertily cows. This genes were identified using the Weighted correlation network analysis (WGCNA).

> The online version of Webgestalt can be found here: http://webgestalt.org/
>
> More information about the R package WebgestaltR can be found here: https://cran.r-project.org/web/packages/WebGestaltR/WebGestaltR.pdf

First, we need to install and load the package

```
#BiocManager::install("WebGestaltR")
```

```
#Loading the package
library(WebGestaltR)
```

```
## ****************************************
## *                                      *
## *          Welcome to WebGestaltR !    *
## *                                      *
## ****************************************
```

The following commands can be used to obtain a complete list of options to be used in each one of the arguments informed to the webGestaltR function:

List of servers: listArchiveUrl()

List of organisms: listOrganism()

List of enrichment options: listGeneSet(*organism*)

List of available IDs listIdType(*organism*)

```
#The WebGestaltR is responsible to run the enrichment analysis

#First, upload your list of candidate genes
genes.interval<-read.table("Genes_RNAseq_modules.csv", h=T, sep="\t")
head(genes.interval)
```

```
##                 gene module entrezgene external_gene_name
## 1 ENSBTAG00000011808  coral     281187               MSTN
## 2 ENSBTAG00000011873  coral     527762              KCNE3
## 3 ENSBTAG00000013210  coral     286806            ADAMTS4
## 4 ENSBTAG00000013578  coral     513513             CHI3L2
## 5 ENSBTAG00000015086  coral     282589            HSD11B1
## 6 ENSBTAG00000015204  coral     615975               SMPX
```

```
#Biological Process

out.WebGestaltR.BP<-WebGestaltR(enrichMethod="ORA",
                                organism="btaurus",
                                enrichDatabase="geneontology_Biological_Process_noRedundant",
                                interestGene=as.character(genes.interval$gene),
                                interestGeneType="ensembl_gene_id",collapseMethod="mean",
                                referenceGeneType="ensembl_gene_id",
                                referenceSet="genome",minNum=1,maxNum=500,
                                fdrMethod="BH",sigMethod="top",fdrThr=1,
                                topThr=100,reportNum=40,perNum=1000,
                                isOutput=F,hostName="http://www.webgestalt.org/")
```

```
## Loading the functional categories...
## Loading the ID list...
## Loading the reference list...
## Performing the enrichment analysis...
```

```
out.WebGestaltR.BP[1:10,1:9]
```

```
##       geneSet
```

```
## 1  GO:0007600
## 2  GO:0044057
## 3  GO:0032504
## 4  GO:0051606
## 5  GO:0010469
## 6  GO:0044703
## 7  GO:0034762
## 8  GO:0003012
## 9  GO:0007200
## 10 GO:0009725
##                                                                    description
## 1                                                          sensory perception
## 2                                                 regulation of system process
## 3                                       multicellular organism reproduction
## 4                                                    detection of stimulus
## 5                                  regulation of signaling receptor activity
## 6                                        multi-organism reproductive process
## 7                                     regulation of transmembrane transport
## 8                                                    muscle system process
## 9  phospholipase C-activating G protein-coupled receptor signaling pathway
## 10                                                     response to hormone
##                                                    link size overlap    expect
## 1  http://amigo.geneontology.org/amigo/term/GO:0007600  491      22 8.2278326
## 2  http://amigo.geneontology.org/amigo/term/GO:0044057  209      12 3.5022750
## 3  http://amigo.geneontology.org/amigo/term/GO:0032504  323      15 5.4126068
## 4  http://amigo.geneontology.org/amigo/term/GO:0051606  367      16 6.1499279
## 5  http://amigo.geneontology.org/amigo/term/GO:0010469  332      15 5.5634225
## 6  http://amigo.geneontology.org/amigo/term/GO:0044703  384      16 6.4348019
## 7  http://amigo.geneontology.org/amigo/term/GO:0034762  220      10 3.6866053
## 8  http://amigo.geneontology.org/amigo/term/GO:0003012  164       8 2.7481966
## 9  http://amigo.geneontology.org/amigo/term/GO:0007200   45       4 0.7540783
## 10 http://amigo.geneontology.org/amigo/term/GO:0009725  356      13 5.9655976
##    enrichmentRatio       pValue         FDR
## 1         2.673851 2.117346e-05 0.009909177
## 2         3.426344 1.956299e-04 0.040942612
## 3         2.771308 3.267407e-04 0.040942612
## 4         2.601657 4.171416e-04 0.040942612
## 5         2.696182 4.374211e-04 0.040942612
## 6         2.486479 6.839555e-04 0.053348526
## 7         2.712523 3.836884e-03 0.256523123
## 8         2.911000 6.247605e-03 0.278469447
## 9         5.304489 6.624356e-03 0.278469447
## 10        2.179161 6.702695e-03 0.278469447
```

```r
#Molecular Function

out.WebGestaltR.MF<-WebGestaltR(enrichMethod="ORA",
                                organism="btaurus",
                                enrichDatabase="geneontology_Molecular_Function_noRedundant",
                                interestGene=as.character(genes.interval$gene),
                                interestGeneType="ensembl_gene_id",
                                collapseMethod="mean",
                                referenceGeneType="ensembl_gene_id",
                                referenceSet="genome",minNum=1,maxNum=500,
```

```
                                fdrMethod="BH",sigMethod="top",fdrThr=1,
                                topThr=100,reportNum=40,perNum=1000,
                                isOutput=F,hostName="http://www.webgestalt.org/")
```

```
## Loading the functional categories...
## Loading the ID list...
## Loading the reference list...
## Performing the enrichment analysis...
```

```
out.WebGestaltR.MF[1:10,1:9]
```

```
##       geneSet                                                description
## 1  GO:0030545                                  receptor regulator activity
## 2  GO:0022803                   passive transmembrane transporter activity
## 3  GO:0008324                    cation transmembrane transporter activity
## 4  GO:0005539                                  glycosaminoglycan binding
## 5  GO:0005126                                  cytokine receptor binding
## 6  GO:0016229                               steroid dehydrogenase activity
## 7  GO:1901681                                sulfur compound binding
## 8  GO:0015318 inorganic molecular entity transmembrane transporter activity
## 9  GO:0070405                                   ammonium ion binding
## 10 GO:0042165                                neurotransmitter binding
##                                                   link size overlap    expect
## 1  http://amigo.geneontology.org/amigo/term/GO:0030545  331      14 5.6136935
## 2  http://amigo.geneontology.org/amigo/term/GO:0022803  250      11 4.2399497
## 3  http://amigo.geneontology.org/amigo/term/GO:0008324  377      14 6.3938442
## 4  http://amigo.geneontology.org/amigo/term/GO:0005539   96       6 1.6281407
## 5  http://amigo.geneontology.org/amigo/term/GO:0005126  196       9 3.3241206
## 6  http://amigo.geneontology.org/amigo/term/GO:0016229   23       3 0.3900754
## 7  http://amigo.geneontology.org/amigo/term/GO:1901681  101       6 1.7129397
## 8  http://amigo.geneontology.org/amigo/term/GO:0015318  484      16 8.2085427
## 9  http://amigo.geneontology.org/amigo/term/GO:0070405   52       4 0.8819095
## 10 http://amigo.geneontology.org/amigo/term/GO:0042165   32       3 0.5427136
##    enrichmentRatio      pValue       FDR
## 1         2.493902 0.001404164 0.1592662
## 2         2.594370 0.003392939 0.1592662
## 3         2.189606 0.004625316 0.1592662
## 4         3.685185 0.005669774 0.1592662
## 5         2.707483 0.005995903 0.1592662
## 6         7.690821 0.006591325 0.1592662
## 7         3.502750 0.007237765 0.1592662
## 8         1.949189 0.007629516 0.1592662
## 9         4.535613 0.011423355 0.2119667
## 10        5.527778 0.016527682 0.2760123
```

```
#Cellular Component

out.WebGestaltR.CC<-WebGestaltR(enrichMethod="ORA",
                                organism="btaurus",
                                enrichDatabase="geneontology_Cellular_Component_noRedundant",
                                interestGene=as.character(genes.interval$gene),
                                interestGeneType="ensembl_gene_id",
                                collapseMethod="mean",
                                referenceGeneType="ensembl_gene_id",
                                referenceSet="genome",minNum=1,maxNum=500,
```

```
                                fdrMethod="BH",sigMethod="top",fdrThr=1,
                                topThr=100,reportNum=40,perNum=1000,
                                isOutput=F,hostName="http://www.webgestalt.org/")
```

```
## Loading the functional categories...
## Loading the ID list...
## Loading the reference list...
## Performing the enrichment analysis...
```

```
out.WebGestaltR.CC[1:10,1:9]
```

```
##       geneSet                    description
## 1  GO:0005581                collagen trimer
## 2  GO:0044815           DNA packaging complex
## 3  GO:0032993             protein-DNA complex
## 4  GO:0009986                    cell surface
## 5  GO:1990351             transporter complex
## 6  GO:0031012            extracellular matrix
## 7  GO:0098590          plasma membrane region
## 8  GO:0000785                      chromatin
## 9  GO:0005791     rough endoplasmic reticulum
## 10 GO:0098797 plasma membrane protein complex
##                                                  link size overlap    expect
## 1  http://amigo.geneontology.org/amigo/term/GO:0005581   39       5 0.4762211
## 2  http://amigo.geneontology.org/amigo/term/GO:0044815   71       5 0.8669666
## 3  http://amigo.geneontology.org/amigo/term/GO:0032993  118       5 1.4408740
## 4  http://amigo.geneontology.org/amigo/term/GO:0009986  340       9 4.1516710
## 5  http://amigo.geneontology.org/amigo/term/GO:1990351  160       5 1.9537275
## 6  http://amigo.geneontology.org/amigo/term/GO:0031012  167       5 2.0392031
## 7  http://amigo.geneontology.org/amigo/term/GO:0098590  474      10 5.7879177
## 8  http://amigo.geneontology.org/amigo/term/GO:0000785  304       7 3.7120823
## 9  http://amigo.geneontology.org/amigo/term/GO:0005791   44       2 0.5372751
## 10 http://amigo.geneontology.org/amigo/term/GO:0098797  287       6 3.5044987
##    enrichmentRatio       pValue        FDR
## 1        10.499325 9.897709e-05 0.01138236
## 2         5.767235 1.652248e-03 0.09500425
## 3         3.470116 1.426451e-02 0.54680604
## 4         2.167802 2.187098e-02 0.62879059
## 5         2.559211 4.522191e-02 1.00000000
## 6         2.451938 5.264299e-02 1.00000000
## 7         1.727737 6.114843e-02 1.00000000
## 8         1.885734 7.653850e-02 1.00000000
## 9         3.722488 1.002778e-01 1.00000000
## 10        1.712085 1.369899e-01 1.00000000
```

##Using WebGestaltR to run KEGG enrichment analysis

The WebGestaltR package is a very useful package to run several types of enrichment analysis, such as GO terms and metabolic pathway analysis.

```
#The WebGestaltR is responsible to run the enrichment analysis
genes.interval<-read.table("Genes_RNAseq_modules.csv", h=T, sep="\t")

out.WebGestaltR<-WebGestaltR(enrichMethod="ORA",
                             organism="btaurus",
                             enrichDatabase="pathway_KEGG",
```

```
                            interestGene=as.character(genes.interval$gene),
                            interestGeneType="ensembl_gene_id",
                            collapseMethod="mean",
                            referenceGeneType="ensembl_gene_id",
                            referenceSet="genome",minNum=1,maxNum=500,
                            fdrMethod="BH",sigMethod="top",fdrThr=1,
                            topThr=100,reportNum=40,perNum=1000,
                            isOutput=F,hostName="http://www.webgestalt.org/")
```

```
## Loading the functional categories...
## Loading the ID list...
## Loading the reference list...
## Performing the enrichment analysis...
```

```
out.WebGestaltR[1:10,1:9]
```

```
##     geneSet                                    description
## 1  bta00140                 Steroid hormone biosynthesis
## 2  bta00590                  Arachidonic acid metabolism
## 3  bta04913                      Ovarian steroidogenesis
## 4  bta04080      Neuroactive ligand-receptor interaction
## 5  bta00830                           Retinol metabolism
## 6  bta00980 Metabolism of xenobiotics by cytochrome P450
## 7  bta05322                 Systemic lupus erythematosus
## 8  bta00053           Ascorbate and aldarate metabolism
## 9  bta04974            Protein digestion and absorption
## 10 bta04060       Cytokine-cytokine receptor interaction
##
## 1                          http://www.kegg.jp/kegg-bin/show_pathway?bta00140+100296421+280934+28174
## 2                                   http://www.kegg.jp/kegg-bin/show_pathway?bta00590+286820+51189
## 3                                              http://www.kegg.jp/kegg-bin/show_pathway?bta04913
## 4  http://www.kegg.jp/kegg-bin/show_pathway?bta04080+281126+281642+281798+281900+282133+338070+51751
## 5                                            http://www.kegg.jp/kegg-bin/show_pathway?bta00830+1
## 6                                            http://www.kegg.jp/kegg-bin/show_pathway?bta00980+1
## 7                    http://www.kegg.jp/kegg-bin/show_pathway?bta05322+107131385+107131750+11244727
## 8                                                                  http://www.kegg.jp/kegg-bin/show_pathw
## 9                                          http://www.kegg.jp/kegg-bin/show_pathway?bta04974+28241
## 10  http://www.kegg.jp/kegg-bin/show_pathway?bta04060+100138192+100329206+281095+281187+511674+51251
##    size overlap    expect enrichmentRatio       pValue         FDR
## 1    67       8 0.9673926        8.269652 4.785686e-06 0.001545777
## 2    83       7 1.1984117        5.841064 1.839557e-04 0.029708850
## 3    53       5 0.7652509        6.533805 9.524218e-04 0.102544085
## 4   302      12 4.3604861        2.751987 1.339791e-03 0.108188163
## 5    64       5 0.9240765        5.410807 2.231127e-03 0.144130790
## 6    67       5 0.9673926        5.168532 2.730867e-03 0.147011648
## 7   181       8 2.6134039        3.061142 4.579875e-03 0.206650274
## 8    25       3 0.3609674        8.311000 5.354963e-03 0.206650274
## 9   113       6 1.6315726        3.677434 5.758057e-03 0.206650274
## 10  323      11 4.6636987        2.358643 6.847689e-03 0.221180363
```

##MeSH (Medical Subject Headings) terms enrichment analysis

MeSH (Medical Subject Headings) is the NLM controlled vocabulary thesaurus used for indexing articles for PubMed.

First we need to install and load the following packages:

```
#BiocManager::install("org.Bt.eg.db")
#BiocManager::install("MeSH.db")
#BiocManager::install("meshes")
#BiocManager::install("MeSH.Bta.eg.db")
```

```
library("biomaRt")
library("meshes")
```

```
## Warning: package 'meshes' was built under R version 4.0.3
```

```
##
```

```
## meshes v1.16.0
##
## If you use meshes in published research, please cite the most appropriate paper(s):
##
## Guangchuang Yu. Using meshes for MeSH term enrichment and semantic analyses. Bioinformatics 2018, 34
```

```
#Loading
library("biomaRt")
```

```
#Importing dataset
genes.modules<-read.table("Genes_RNAseq_modules.csv",h=T, sep="\t", stringsAsFactors = F)

#Getting Entrez IDs
mart <- useMart("ENSEMBL_MART_ENSEMBL")
mart <- useDataset("btaurus_gene_ensembl", mart)
filter<-"ensembl_gene_id"
value<-unique(genes.modules$gene)
attributes <- c("ensembl_gene_id","external_gene_name","entrezgene_id")
all.genes <- getBM(attributes=attributes, filters=filter, values=value, mart=mart)


#Anatomy
mesh.A<-enrichMeSH(all.genes[which(!is.na(all.genes$entrezgene_id)),"entrezgene_id"],
                   MeSHDb = "MeSH.Bta.eg.db", database='gene2pubmed',
                   category = 'A',pvalueCutoff = 1,
                   qvalueCutoff = 1,minGSSize = 1)
```

```
## Loading required package: MeSH.Bta.eg.db
```

```
## Loading required package: MeSHDbi
```

```
## Warning: package 'MeSHDbi' was built under R version 4.0.3
```

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

```
## Warning: package 'BiocGenerics' was built under R version 4.0.3
```

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

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

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

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

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

##
## Attaching package: 'MeSHDbi'

## The following object is masked from 'package:utils':
##
##     packageName
```

```
meshA.result<-mesh.A@result
meshA.result$group<-"Anatomy"

#Diseases
mesh.C<-enrichMeSH(all.genes[which(!is.na(all.genes$entrezgene_id)),"entrezgene_id"],
                   MeSHDb = "MeSH.Bta.eg.db", database='gene2pubmed',
                   category = 'C',
                   pvalueCutoff = 1,qvalueCutoff = 1,minGSSize = 1)
meshC.result<-mesh.C@result
meshC.result$group<-"Diseases"

#Biological Sciences
mesh.G<-enrichMeSH(all.genes[which(!is.na(all.genes$entrezgene_id)),"entrezgene_id"],
                   MeSHDb = "MeSH.Bta.eg.db", database='gene2pubmed',
                   category = 'G',pvalueCutoff = 1,
                   qvalueCutoff = 1,minGSSize = 1)
meshG.result<-mesh.G@result
meshG.result$group<-"Biological Sciences"


#Combining all the results in a single data frame
mesh.final<-rbind(meshA.result,meshC.result,meshG.result)

#Checking the results
head(mesh.final)
```

```
##              ID       Description GeneRatio   BgRatio       pvalue    p.adjust
## D004848 D004848        Epithelium     6/207  69/24570 2.563068e-05 0.003271232
## D000311 D000311    Adrenal Glands     6/207  72/24570 3.271232e-05 0.003271232
## D001854 D001854 Bone Marrow Cells     3/207  11/24570 9.252027e-05 0.006168018
## D015571 D015571  Follicular Fluid     5/207  61/24570 1.640086e-04 0.008200429
## D013799 D013799       Theca Cells     5/207  65/24570 2.215945e-04 0.008863778
## D002462 D002462     Cell Membrane    10/207 304/24570 2.719265e-04 0.009064215
##              qvalue
## D004848 0.002117692
## D000311 0.002117692
```

```
## D001854 0.003992980
## D015571 0.005308699
## D013799 0.005738130
## D002462 0.005867887
##                                                                                 geneID
## D004848                                281095/286820/280934/281569/281129/493988
## D000311                                281355/281824/338048/521831/525480/338092
## D001854                                                          530116/280846/493725
## D015571                                      280934/281740/282589/281900/338092
## D013799                                      281187/281740/282589/281900/512385
## D002462 281798/280846/317695/281355/282133/281569/286806/281483/282338/281642
##         Count   group
## D004848     6 Anatomy
## D000311     6 Anatomy
## D001854     3 Anatomy
## D015571     5 Anatomy
## D013799     5 Anatomy
## D002462    10 Anatomy
```

##Gene network

The gene network analysis is an interesting approach to better understand the relationship between the positional candidate genes. The identification of gene networks can help to select the functional candidate genes, to identify key regulatory genes, and to better understand the biological processes related with the development of complex traits.

The package STRINGdb provides a R interface to the STRING protein-protein interactions database (http://www.string-db.org).

To install the packages, the following commands can be used:

```
#Installing STRINGdb
#BiocManager::install("STRINGdb")
```

```
#Loading the package
library(STRINGdb)
```

```
## Warning: package 'STRINGdb' was built under R version 4.0.3
```

```
#Creating a Data frame with the Gene symbols
genes.modules<-read.table("Genes_RNAseq_modules.csv",h=T, sep="\t", stringsAsFactors = F)

head(genes.modules)
```

```
##                 gene module entrezgene external_gene_name
## 1 ENSBTAG00000011808  coral     281187               MSTN
## 2 ENSBTAG00000011873  coral     527762              KCNE3
## 3 ENSBTAG00000013210  coral     286806            ADAMTS4
## 4 ENSBTAG00000013578  coral     513513             CHI3L2
## 5 ENSBTAG00000015086  coral     282589            HSD11B1
## 6 ENSBTAG00000015204  coral     615975               SMPX
```

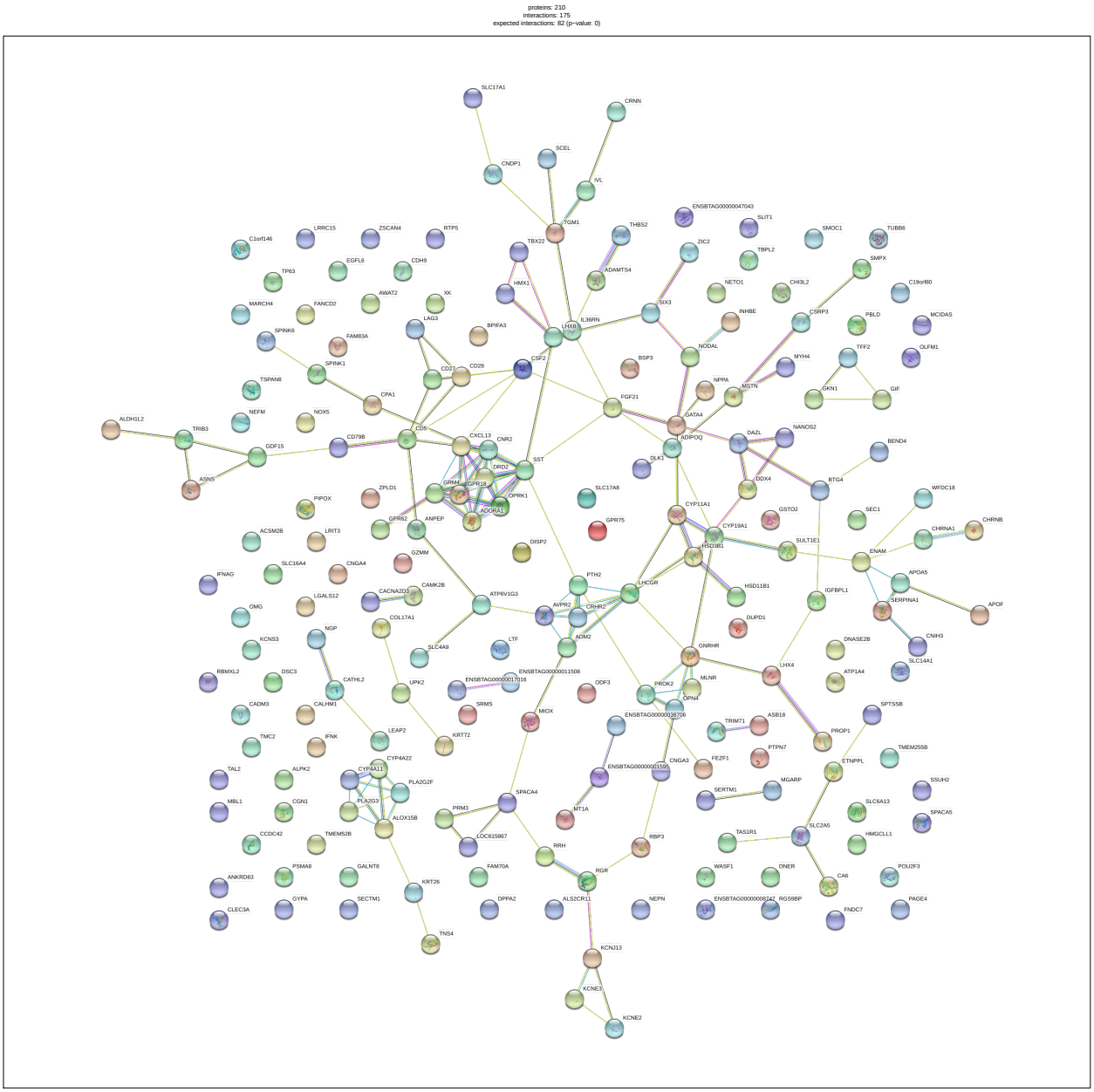
```
candidate_ID<-data.frame(gene=unique(genes.modules$external_gene_name))

#Loading the STRING interface for Bos taurus (ID=9913)
string_db <- STRINGdb$new(species=9913,version="11.0")
```

```
## WARNING: Score threshold is not specified. We will be using medium stringency cut-off of 400.
```

```
#Mapping the interactions of the genes presen in our input list
gene_mapped <- string_db$map(candidate_ID, "gene", removeUnmappedRows = TRUE )
```

```
## Warning:  we couldn't map to STRING 3% of your identifiers
```

```
hits <- gene_mapped$STRING_id

#Plotting the gene network
string_db$plot_network(hits, payload_id=NULL, required_score=NULL, add_link=T, add_summary=T)
```



It is possible to perform GO and KEGG pathways enrichment analyses using STRINGdb package, as well as in the online version of STRING db.

```
#GO enrichment analysis
enrichmentGO <- string_db$get_enrichment( hits, category = "Process",
```

```
                                          methodMT = "fdr", iea = TRUE )
```

```
## Warning in string_db$get_enrichment(hits, category = "Process", methodMT =
## "fdr", : methodMT parameter is depecated. Only FDR correction is available.
```

```
## Warning in string_db$get_enrichment(hits, category = "Process", methodMT =
## "fdr", : iea parameter is deprecated.
```

```
## [1] "Process"
```

```
head(enrichmentGO)
```

```
##    category       term number_of_genes number_of_genes_in_background
## 56  Process GO.0003008              15                           312
## 57  Process GO.0032501              31                          1333
## 58  Process GO.0008217               6                            52
## 59  Process GO.1903556               3                             9
## 60  Process GO.1901652               6                            65
## 61  Process GO.0071375               5                            42
##    ncbiTaxonId
## 56        9913
## 57        9913
## 58        9913
## 59        9913
## 60        9913
## 61        9913
##
## 56
## 57 9913.ENSBTAP00000001209,9913.ENSBTAP00000001704,9913.ENSBTAP00000002054,9913.ENSBTAP00000002271,99
## 58
## 59
## 60
## 61
##
## 56                                                                                                  OPRK1,RBI
## 57 OPRK1,LTF,CSF2,ODF3,BSP3,RBP3,GPR18,NPPA,DDX4,DRD2,ADORA1,PRM3,PBLD,SMPX,SPINK1,RGR,LHCGR,WASF1,AI
## 58
## 59
## 60
## 61
##     p_value    fdr
## 56 1.81e-06 0.0028
## 57 3.24e-05 0.0237
## 58 3.03e-05 0.0237
## 59 2.30e-04 0.0336
## 60 9.59e-05 0.0336
## 61 1.20e-04 0.0336
##                                                                  description
## 56                                                            system process
## 57                                          multicellular organismal process
## 58                                              regulation of blood pressure
## 59 negative regulation of tumor necrosis factor superfamily cytokine production
## 60                                                        response to peptide
## 61                                cellular response to peptide hormone stimulus
```

```
#KEGG enrichment analysis
enrichmentKEGG <- string_db$get_enrichment( hits, category = "KEGG",
                                            methodMT = "fdr", iea = TRUE )
```

```
## Warning in string_db$get_enrichment(hits, category = "KEGG", methodMT = "fdr", :
## methodMT parameter is depecated. Only FDR correction is available.
```

```
## Warning in string_db$get_enrichment(hits, category = "KEGG", methodMT = "fdr", :
## iea parameter is deprecated.
```

```
## [1] "KEGG"
```

```
head(enrichmentKEGG)
```

```
##    category       term number_of_genes number_of_genes_in_background
## 56  Process GO.0003008              15                           312
## 57  Process GO.0032501              31                          1333
## 58  Process GO.0008217               6                            52
## 59  Process GO.1903556               3                             9
## 60  Process GO.1901652               6                            65
## 61  Process GO.0071375               5                            42
##    ncbiTaxonId
## 56        9913
## 57        9913
## 58        9913
## 59        9913
## 60        9913
## 61        9913
##
## 56
## 57 9913.ENSBTAP00000001209,9913.ENSBTAP00000001704,9913.ENSBTAP00000002054,9913.ENSBTAP00000002271,99
## 58
## 59
## 60
## 61
##
## 56                                                                                                  OPRK1,RBI
## 57 OPRK1,LTF,CSF2,ODF3,BSP3,RBP3,GPR18,NPPA,DDX4,DRD2,ADORA1,PRM3,PBLD,SMPX,SPINK1,RGR,LHCGR,WASF1,AI
## 58
## 59
## 60
## 61
##     p_value    fdr
## 56 1.81e-06 0.0028
## 57 3.24e-05 0.0237
## 58 3.03e-05 0.0237
## 59 2.30e-04 0.0336
## 60 9.59e-05 0.0336
## 61 1.20e-04 0.0336
##                                                                  description
## 56                                                            system process
## 57                                        multicellular organismal process
## 58                                            regulation of blood pressure
## 59 negative regulation of tumor necrosis factor superfamily cytokine production
## 60                                                        response to peptide
## 61                                cellular response to peptide hormone stimulus
```

There are several other options to perform clustering and additional enrichment analysis in the STRINGdb package. Some examples can be found here: https://rdrr.io/bioc/STRINGdb/f/inst/doc/STRINGdb.pdf

This tutorial showed some interesting analyses that can be performed using R. These analyses can be integrated in a pipeline, where the output from different functions and/or packages can be directly used by other functions/packages. This procedure increase the efficiency and the management process for functional studies when hundreds (or even thousands) of genes are analyzed.