Pre-processing needed to compile the available .rda files

Mario Failli, Jussi Pananen and Vittorio Fortino

2019-11-28

Pre-processing needed to compile the available .rda files

Mario Failli, Jussi Pananen and Vittorio Fortino

2019-11-28

Estimating disease relevant genes and tissues before compiling the tissue-specific efficacy scores

```r
library(ThETA)
data(gtexv7_zscore)
data(ppi_strdb_700)
data(dis_vrnts)
data(disease_tissue_zscores)
data(centrality_score)
```

The following code shows how to retrieve disease-associated genes with a specific confident score (see www.disgenet.org/help for more details) and how to compile z-scores for disease-tissue pairs.

```r
if("AnnotationDbi" %in% rownames(installed.packages()) == FALSE) {
  if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
  BiocManager::install("AnnotationDbi")}
if("MeSH.db" %in% rownames(installed.packages()) == FALSE) {
  if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
  BiocManager::install("MeSH.db")}
library(MeSH.db)
library(AnnotationDbi)
```

First, the MeshID corresponding to T2DM are retrieved.

```
T2DM_mesh <- AnnotationDbi::select(MeSH.db,keys = 'Diabetes Mellitus, Type 2',
                                    columns = c("MESHID","MESHTERM"),
                                    keytype = "MESHTERM")
```

Then, variant genes associated to T2DM are collected from DisGeNET.

```
library(RCurl)
library(XML)
T2DM_genes <- disease.vrnts(T2DM_mesh$MESHID, id_type = 'mesh', min_score = 0.6, curated = T)
```

Once the set of disease-relevant genes is downloaded, then the corresponding set of disease-relevant tissues can be determined (Failli et al. 2019). The parameter *top* can be used to extend the initial set of disease-relevant genes. It indicates how many genes closely related to the disease-genes must be added.

```
T2DM_tiss_zscore <- dis.rel.tissues(disease_genes = T2DM_genes$entrez,
                                    ppi_network = ppi_strdb_700, weighted = TRUE,
                                    tissue_expr_data = gtexv7_zscore,
                                    thr = 1, top=0, rand = 1000, verbose = T) # rand = 10000 is highly recommended
```

The previous code generates the following table of z-scores, indicating significant associations between disease and tissues in GTEx.

After compiling scoring disease-tissue associations, the user can compile the tissue-specific efficacy scores as follow.

```
if("magrittr" %in% rownames(installed.packages()) == FALSE) {install.packages("magrittr")}
library(magrittr)
T2DM_scores <- tissue.specific.scores(T2DM_genes$entrez,
                                        ppi_network = ppi_strdb_700,
                                        directed_network = F,
                                        tissue_expr_data = gtexv7_zscore,
                                        dis_relevant_tissues = T2DM_tiss_zscore$z %>%
                                                             `names<-`(rownames(T2DM_tiss_zscore)),
                                        W = centrality_score$borda.disc, cutoff = 5, verbose = T)
```

The following sections show how to build from scratch the pre-compiled .rda files needed for the tissue specific score

Collecting gene expression data per tissue from GTEx

In order to create the .rda file *gtexv7_zscore* the following R packages need to be installed.

```
if("CePa" %in% rownames(installed.packages()) == FALSE) {install.packages("CePa")}
if("grex" %in% rownames(installed.packages()) == FALSE) {install.packages("grex")}
library(grex)
library(CePa)
```

Then, the function *read.gct()* is utilized to read the GTEx file.

```
gtexv7_median_tpm <- CePa::read.gct('../data/GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_median_tpm.gct.gz')
colnames(gtexv7_median_tpm) <- sub("\\.$",'',colnames(gtexv7_median_tpm))
colnames(gtexv7_median_tpm) <- gsub("\\.+",'_',colnames(gtexv7_median_tpm))
```

The TPM values are log-transformed and then converted to z-scores by using the *tissue.exp* function.

```
gtexv7_zscore <- tissue.expr(log2(gtexv7_median_tpm+1))             # to convert exp vals to z-scores
rownames(gtexv7_zscore) <- grex::cleanid(rownames(gtexv7_zscore))   # to produce Ensembl IDs.
```

Moreover, all the Ensembl IDs are mapped to Entrez Gene ID.

```
gtexv7_gene_ann <- grex::grex(rownames(gtexv7_zscore))
gtexv7_gene_ann <- gtexv7_gene_ann[!is.na(gtexv7_gene_ann$entrez_id),]
gtexv7_zscore <- gtexv7_zscore[gtexv7_gene_ann$ensembl_id,]
rownames(gtexv7_zscore) <- gtexv7_gene_ann$entrez_id
#save(gtexv7_zscore,file='gtexv7_zscore.rda')
```

Defining a PPI network from STRINGDB

In order to create the .rda file *ppi_strdb_700* the following R packages need to be installed.

```
if("STRINGdb" %in% rownames(installed.packages()) == FALSE) {
  if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
  BiocManager::install("STRINGdb")}
if("igraph" %in% rownames(installed.packages()) == FALSE) {install.packages("igraph")}
library(STRINGdb)
library(igraph)
```

First, the Human Protein-Protein interaction network is selected. Then, genes that are annotated in GTEx and connections (or edges) with a score greater than or equal to 700 are kept.

```
string_db <- STRINGdb$new(version="10", species=9606)
gtexv7_mapped <- string_db$map(my_data_frame = gtexv7_gene_ann, my_data_frame_id_col_names='entrez_id', removeUnmappedRows=T)
string_inter <- string_db$get_interactions(gtexv7_mapped$STRING_id)
string_inter <- string_inter[string_inter$combined_score>=700,] # to select connections with minimum STRING combined score >= 700
idx_from <- match(x = string_inter$from, table = gtexv7_mapped$STRING_id)
idx_to <- match(x = string_inter$to, table = gtexv7_mapped$STRING_id)
ppi_strdb_700 <- data.frame(node1=gtexv7_mapped$entrez_id[idx_from], node2=gtexv7_mapped$entrez_id[idx_to], weight=string_inter$c
#save(ppi_strdb_700,file='ppi_strdb_700.rda')
```

The PPI compiled from STRINGdb will be a data frame containing a symbolic edge list in the first two columns and connection scores in the third column.

Compiling disease-tissue associations

In order to make the .rda files *dis_vrnts* and *disease_tissue_zscores* the following R package *ontologyIndex* need to be installed. Then, the z-score is compiled for all possible disease-tissue pairs.

```
if("ontologyIndex" %in% rownames(installed.packages()) == FALSE) {install.packages("ontologyIndex")}
library(ontologyIndex)

# select the genes associated with each disease annotated in the OBO file
efo.OBO <- get_OBO('../data/efo_feb2019.obo', extract_tags = "everything")
# remove the tag "EFO:" from each id
efo_ids <- gsub('EFO:','', grep('EFO:', efo.OBO$id, value = T))
# select gene-disease associations from DisGeNET
dis_vrnts <- sapply(efo_ids, function(x) disease.vrnts(x, id_type='efo',
                                                       min_score=0.6,
                                                       curated=F), simplify = F)
# define the toal set of genes from the PPI
all_ppi_genes <- unique(as.character(unlist(ppi_strdb_700[,1:2])))
# select the diseases with at least 5 (associated) genes
dis_vrnts <-  dis_vrnts[sapply(dis_vrnts, function(x)
                        length(intersect(x$entrez,all_ppi_genes)))>=5]
# comiple the disease relevant tissues for each selected disease
disease_tissue_zscores <- list.dis.rel.tissues(disease_gene_list = sapply(dis_vrnts,'[[',1),
                                               ppi_network = ppi_strdb_700,
                                               weighted = TRUE,
                                               tissue_expr_data = gtexv7_zscore, thr = 1,
                                               top = 0, rand = 10000, parallel = 10)
#save(dis_vrnts, file='dis_vrnts.rda')
#save(disease_tissue_zscores, file='disease_tissue_zscores.rda')
```

Compiling node centrality scores

The .rda file *centrality_score* can be compiled by using the R function *node.centrality*.

```
centrality_score <- node.centrality(ppi_network = ppi_strdb_700,
                                    tissue_expr_data = gtexv7_zscore,
                                    agg_function = 4, directed_network=F,
                                    parallel = 10, verbose = FALSE)
#save(centrality_score,file = 'centrality_score.rda')
```

Compiling tissue-specifc efficacy scores for all target-disease associations

```
idx <- apply(disease_tissue_zscores$z,1,function(x) any(x > 1.6))
disg = sapply(dis_vrnts[idx],'[[',1)   # list of disease-gene sets
disr = disease_tissue_zscores$z[idx,]  # matrix of disease-relevant tissue scores
tissue_score <- list.tissue.specific.scores(disease_gene_list = disg,
                                            ppi_network = ppi_strdb_700, directed_network = F,
                                            tissue_expr_data = gtexv7_zscore,
                                            dis_relevant_tissues = disr,
                                            W = centrality_score$borda.disc, cutoff = 10,
                                            verbose = FALSE, parallel = 2)
```

How to format averaged tissue-specifc efficacy scores

```
avg_tissue_score <- sapply(tissue_score,'[[','avg_tissue_score') %>% `rownames<-`(rownames(tissue_score[[1]]))
avg_tissue_score <- reshape2::melt(avg_tissue_score)
avg_tissue_score[,1:2] <- lapply(avg_tissue_score[,1:2],  as.character)
colnames(avg_tissue_score) <- c('target.entrez','disease.id','avg_tissue.score')
annotation <- list(disease.id=efo.OBO$id,disease.name=efo.OBO$name)
options(stringsAsFactors = FALSE)
avg_tissue_score <- merge(annotation,avg_tissue_score,by='disease.id',all.y=T)
```