Introduction to ThETA

Mario Failli, Jussi Pananen and Vittorio Fortino

2019-11-28

Introduction to ThETA

Mario Failli, Jussi Pananen and Vittorio Fortino

2019-11-28

Compile transcriptome-driven efficacy estimates of target(gene)-disease associations

The R package ThETA implements two novel algorithms to identify and rank target-disease associations based on efficacy estimates compiled from gene expression profiles of gene perturbations and human diseases (Modulation Score), and tissue-specific gene expression networks (Tissue-specific Efficacy Scores). These methods are described in Failli et al. 2019 (https://www.nature.com/articles/s41598-019-46293-7).

Current functions provided by ThETA

- Compile tissue-specific expression networks by using GTEx and StringDB (Human PPI).
- Compile diseas-relevant tissues by implementing the algorithm proposed by Kitsak et al. 2016 (https://www.nature.com/articles/srep35241).
- Extract disease-relevant genes from DisGeNET and mark these genes on the disease-relevant tissue-specific gene expression.
- Compile the tissue-specific efficacy scores on disease-relevant tissues.
- Compile the modulation score, which estimates the likelihood of a gene perturbation (e.g., knockout and knockdown) to result in specific reversion of disease gene-expression profiles (lists of down- and up-regualted genes are downbaloded from Enrichr: https://amp.pharm.mssm.edu/Enrichr/).
- Integrate multiple efficacy scores with the max function and the harmonic sum (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5210543/).

- Build igraph objects including tissue-specific networks and paths connecting selected drug-targets (or genes) and disease-relevant genes (it also include info on the gene modulation scores).

1. How to compile the tissue-specific efficacy estimates of target(gene)-disease associations

In order to compile tissue-specific efficacy estimates of drug-target disease associations we need to:

- Collect tissue-specific gene expression profiles from GTEX.
- Define a protein-protein interaction network.
- Identify disease-associated genes (from DisGeNET) and disease relevant tissues.
- Compile node centrality scores

These steps are computationally expensive! Therefore, ThETA provides pre-compiled .rda files that can be used to rapidly generate tissue-specific efficacy scores.

.rda file	Description
gtexv7_zscore.rda	z-scores compiled from log transformed TPM expression profiles of GTEx.
ppi_strdb_700.rda	human protein-protein interaction network extracted from StringDB (combined scores >= 700)
dis_vrnts.rda	disease-associated genes (from DisGeNET; score >= 0.6)
disease_tissue_zscores.rda	significances (z-scores) of disease-tissue associations
dis_vrnts.rda	tissue-specific node centrality scores (integration of degree, clust. coeff. and betweenness

First, we upload the ThETA package and the 5 .rda files:

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

Then, given a disease (i.e. Diabetes Mellitus Type II - T2DM), we can compile the tissue-specific efficacy scores.

1. Variant genes related to T2DM are selected from *dis_vrnts* by using the EFO-id.

```r
T2DM_genes = dis_vrnts[[which(names(dis_vrnts) == "EFO:0001360")]]
```

2. Then, significant tissues for T2DM are obtained from *disease_tissue_zscores*.

```
T2DM_rel_tissue_scores = disease_tissue_zscores$z[which(rownames(disease_tissue_zscores$z) == "EFO:0001360"),]
```

3. A tissue-specific efficacy (TSE) score is then estimated for all genes that are expressed in the tissues that are relevant for T2D. It should be noted that the following script is computer-intensive. Indeed, we specified only two genes in input. However, it is highly recommended to use the whole set of T2D-relevant genes.

```
T2DM_Tscores <- tissue.specific.scores(T2DM_genes$entrez[1:2],
                                        ppi_network = ppi_strdb_700,
                                        directed_network = FALSE,
                                        tissue_expr_data = gtexv7_zscore,
                                        dis_relevant_tissues = T2DM_rel_tissue_scores,
                                        W = centrality_score$borda.disc,
                                        cutoff = 4, verbose = TRUE)
```

The output is a *data.frame* object containing the TSE score for all genes-tissue pairs.

```
#> Warning in kableExtra::kable_styling(., bootstrap_options = "striped",
#> full_width = F): Please specify format in kable. kableExtra can customize
#> either HTML or LaTeX outputs. See https://haozhu233.github.io/kableExtra/
#> for details.
```

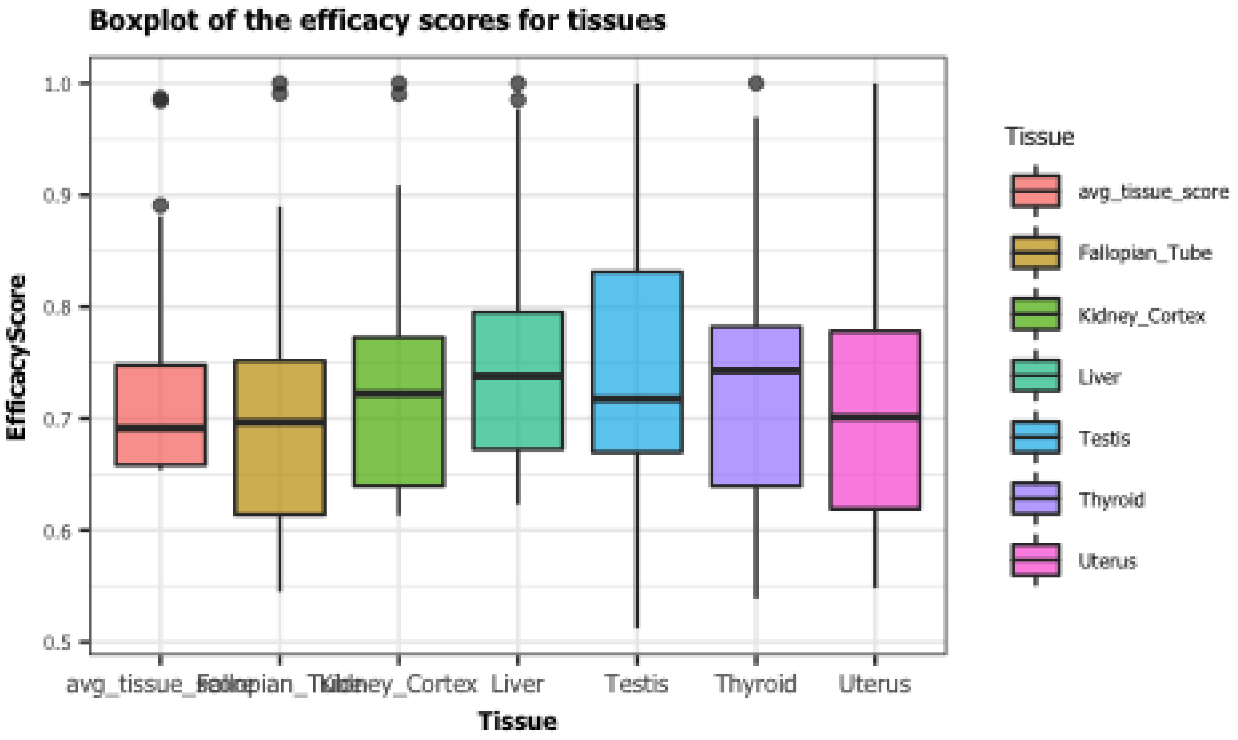
	Fallopian_Tube	Kidney_Cortex	Liver	Testis	Thyroid	Uterus	avg_tissue_score
1080	0.6030391	0.6152350	0.6311006	0.6045757	0.5482383	0.5823116	0.5974167
6356	0.4022889	0.5029331	0.6117747	0.5240006	0.4777560	0.4218802	0.4901056
5734	0.5131348	0.5667641	0.5722064	0.5692878	0.7050607	0.4737322	0.5666977
3308	0.6161507	0.5682854	0.5524524	0.6434554	0.5724015	0.6165954	0.5948901
136	0.5348355	0.5012205	0.5654343	0.5561065	0.6517366	0.4500510	0.5432307

This *data.frame* can be subsequently ordered based on the average of the TSE scores in order to prioritize putative gene targets (e.g. top 50 genes).

```
T2DM_top50 <- T2DM_Tscores[order(T2DM_Tscores$avg_tissue_score,
                                  decreasing = TRUE)[1:50],]
```

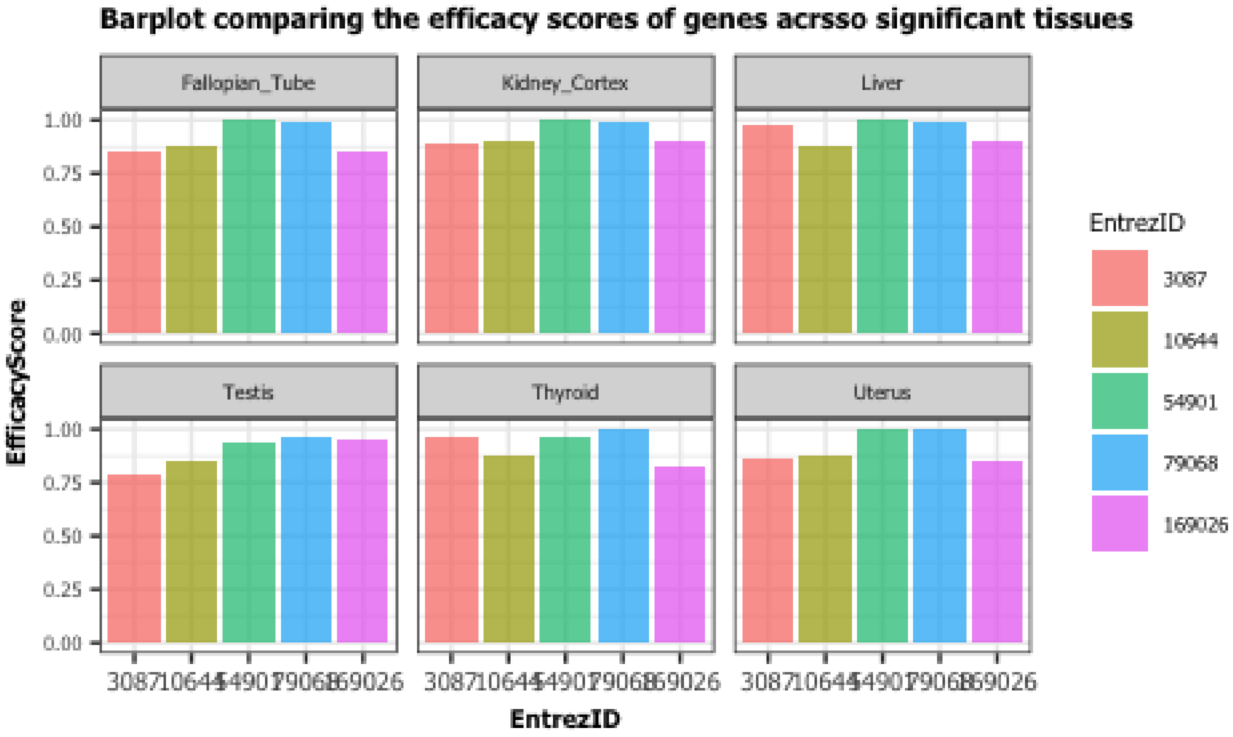
The following plot shows the distribution of the TSE scores for the top 50 genes within each disease relevant tissue,.

```r
library(ggplot2)
library(reshape)
#>
#> Attaching package: 'reshape'
#> The following object is masked from 'package:dplyr':
#>
#>     rename
data_t2d50 <- reshape::melt(as.matrix(T2DM_top50), id = 0)
colnames(data_t2d50) <- c("EntrezID", "Tissue", "EfficacyScore")
ggplot(data_t2d50, aes(x = Tissue, y = EfficacyScore, fill = Tissue)) +
        geom_boxplot(alpha = 0.7) +
        ggtitle("Boxplot of the efficacy scores for tissues") +
        theme_bw() +
        theme(plot.title = element_text(size = 8, family = "Tahoma", face = "bold"),
                text = element_text(size = 7, family = "Tahoma"),
                axis.title = element_text(face="bold"),
                axis.text.x=element_text(size = 7))
```



Additionally, the following barplot reports the tissue-specific scores for the top 5 genes within each tissue.

```
library(ggplot2)
library(reshape)
data_t2d5 <- reshape::melt(as.matrix(T2DM_top50[1:5,-ncol(T2DM_top50)]), id = 0)
colnames(data_t2d5) <- c("EntrezID", "Tissue", "EfficacyScore")
data_t2d5$EntrezID = factor(data_t2d5$EntrezID)
ggplot(data_t2d5, aes(x = EntrezID, y = EfficacyScore, fill = EntrezID)) +
        geom_bar(stat='identity', alpha = 0.7) +
        ggtitle("Barplot comparing the efficacy scores of genes acrsso significant tissues") +
        facet_wrap(~Tissue) +
        theme_bw() +
        theme(plot.title = element_text(size = 8, family = "Tahoma", face = "bold"),
                    text = element_text(size = 7, family = "Tahoma"),
                    axis.title = element_text(face="bold"),
                    axis.text.x=element_text(size = 7))
```



2. How to compile the modulation efficacy estimates of target(gene)-disease associations

In order to compile modulation efficacy estimates of drug-target disease associations the users need to collect lists of up- and down-regulated gene sets identified in disease and gene perturbations. Currentely, ThETA includes lists of up- or down-regulated gene sets retrieved from EnrichR (https://amp.pharm.mssm.edu/Enrichr/). However, the users could compile the modulation score based on a different set of up- and down-regulated gene sets.

```
data(geo_gene_sets)
```

The ThETA package provides a function to calculate the modulation score for all the genes (since it is not a computationaly intensive task).

```
modulation_scores <- modulation.score(geneSets = geo_gene_sets)
```

target.id	disease.id	modscore	target.modulationType	disease.name
PMP22	DOID:2841	0.4614037	OE	asthma
HIPK2	DOID:2841	0.5404949	KD	asthma
KDM6A	DOID:2841	0.6425815	KD	asthma
SMC3	DOID:2841	0.5807682	KD	asthma
NR3C1	DOID:2841	0.6336941	KO	asthma

The following section show how to deal with gene-disease repetitions

EnrichR provides either Disease Ontology (DO) ids or Concept Unique Identifiers (CUIs) to label disease perturbations. Use of different types of disease id may cause gene-disease pair repetitions in the final output of *modulation.score* (different ids might be associated with the same disease). To overcome this issue, a .csv file containing manually curated mapping between either DO ids or CUIs and EFO ids is available in the data folder.

The following code shows how to:

- Cross-link the output of *modulation.score* with the .csv file in data.
- Remove duplicated gene-disease pairs from the output.

First, DO ids or CUIs are replaced with EFO ids.

```
enrichr_to_efo <- read.csv(system.file("conversion_enrichr_efo.csv",
                                        package = "ThETA"), row.names = 1,
                            stringsAsFactors = F)
modulation_scores$disease.id <- enrichr_to_efo[modulation_scores$disease.id,'disease.id']
```

Then, gene symbols need to be converted to Entrez Gene IDs in order to facilitate the integration between TSE and modulation scores.

```
library(org.Hs.eg.db)
modulation_scores$target.entrez <- AnnotationDbi::mapIds(org.Hs.eg.db, modulation_scores$target.id,'ENTREZID','SYMBOL')
modulation_scores <- modulation_scores[modulation_scores$disease.id != '' &
                                       !is.na(modulation_scores$target.entrez),]
```

Finally, for each gene-disease pair, only the perturbation giving the maximum score are selected (see Failli et al. 2019).

```
library(data.table)
modul_score <- data.table::as.data.table(modulation_scores)
modul_score <- as.data.frame(modul_score[, .SD[which.max(modscore)],
                                         by=list(disease.id, target.entrez)])
```

Let's now select the modulation scores for T2D.

```
T2DM_Mscores = data.frame(modul_score[modul_score$disease.id=='EFO:0001360',
                                      c("target.entrez", "modscore")], row.names = 1)
```

	modscore
5376	0.8885534
28996	0.6405335
7403	0.6394576
9126	0.6087785
2908	0.5864897
4209	0.7700771

3. How to integrate TSE and modulation scores

The tissue-specifc and modulation scores can be combined together in order to provide a multi-evidence based ranking of disease-gene-targets.

```
common_t2d_genes <- intersect(rownames(T2DM_Mscores), rownames(T2DM_Tscores))
T2DM_Iscores <- data.frame("Mscore" = T2DM_Mscores[common_t2d_genes,],
                           "TSEscore" = T2DM_Tscores[common_t2d_genes,],
                           row.names = common_t2d_genes)
```

	Mscore	TSEscore.Fallopian_Tube	TSEscore.Kidney_Cortex	TSEscore.avg_tissue_score
5376	0.8885534	0.4080139	0.3245370	0.3405319

	Mscore	TSEscore.Fallopian_Tube	TSEscore.Kidney_Cortex	TSEscore.avg_tissue_score
28996	0.6405335	0.6021129	0.6343235	0.5993317
7403	0.6394576	0.4560764	0.3476318	0.4438615
9126	0.6087785	0.6297193	0.5543101	0.5976275
2908	0.5864897	0.6044743	0.5592048	0.5856605

Multi-evidence rankings of putative drug targets can be further extended by including efficacy scores obtained from other computational platform for drug target discovery such as Open Target platform.

The OT Platform REST API allows access to data available on the OT Platform. The following examples shows how to retrieve disease-gene association scores from the OT platform.

A typical access to the OT Platform REST API requires three inputs: name server, endpoint parameters and optional parameters.

```
server <- 'https://platform-api.opentargets.io/v3/platform'
endpoint_prmtrs <- '/public/association/filter'
optional_prmtrs <- '?size=10000&disease=EFO_0001360&fields=disease.id&fields=target.gene_info.symbol&fields=association_score.ove
uri <- paste(server,endpoint_prmtrs,optional_prmtrs,sep='')
```

Then, a `GET` request is made to pull raw data into our environment. Pulled data, in the JavaScript Object Notification (JSON) format, are subsequently converted into a usable format.

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

get_association_json <- httr::content(httr::GET(uri),'text')
get_association_usable <- jsonlite::fromJSON(get_association_json, flatten = TRUE)

OT_score <- get_association_usable$data[,c(2:3,1,4)]
OT_score$disease.id <- gsub('_',':',OT_score$disease.id)
colnames(OT_score)[c(1,4)] <- c('target.id', 'disease.name')

# remove duplicated gene symbols
OT_score = OT_score[-which(duplicated(OT_score$target.id)),]
```

Gene symbols are then converted to Entrez Gene IDs in order to allign the OT scores with those provided by ThETA.

```
library(org.Hs.eg.db)
OT_score$target.entrez <- AnnotationDbi::mapIds(org.Hs.eg.db,OT_score$target.id,'ENTREZID','SYMBOL')
OT_score <- OT_score[!is.na(OT_score$target.entrez),]
```

target.id	disease.id	association_score.overall	disease.name	target.entrez
PPARG	EFO:0001360	1	type II diabetes mellitus	5468
KCNJ11	EFO:0001360	1	type II diabetes mellitus	3767
INSR	EFO:0001360	1	type II diabetes mellitus	3643
ABCC8	EFO:0001360	1	type II diabetes mellitus	6833
TCF7L2	EFO:0001360	1	type II diabetes mellitus	6934
HNF1B	EFO:0001360	1	type II diabetes mellitus	6928

The scores obtained from the OT platform are first concatenated to the TSE and modulation scores.

```
all_scores <- base::merge(OT_score, T2DM_Iscores, by.x = "target.entrez", by.y = "row.names", all = TRUE)
```

Then, the function *integrate.scores* is used to provide meerged scores: harmonic sum or maximum score.

```
T2DM_allsc <- integrate.scores(all_scores, c("association_score.overall",
                                             "Mscore",
                                             "TSEscore.avg_tissue_score"))
#> [1] 3704   13
T2DM_allsc <- T2DM_allsc[order(T2DM_allsc$HS, decreasing = TRUE),]
rownames(T2DM_allsc) <- T2DM_allsc[,1]

# let's semplify the final table of the disease-gene association scores
tab_score <- T2DM_allsc[,c("target.id","association_score.overall", "Mscore",
                           "TSEscore.avg_tissue_score", "HS","MAX")]
colnames(tab_score)[1:4] <- c("GeneTarget","OTScore","ModulationScore","TissueEfficacyScore")
```

	GeneTarget	OTScore	ModulationScore	TissueEfficacyScore	HS	MAX
5468	PPARG	1	0.9851352	0.6542443	1.318978	1
3643	INSR	1	1.0000000	0.6189753	1.318775	1
208	AKT2	1	1.0000000	0.6138788	1.318209	1

	GeneTarget	OTScore	ModulationScore	TissueEfficacyScore	HS	MAX
79068	FTO	1	0.6372124	0.9868802	1.317521	1
5465	PPARA	1	0.9795581	0.6051955	1.312134	1
7421	VDR	1	0.9642260	0.5808759	1.305598	1

4. How to visualize tissue-specific networks and biological annotations of selected drug(gene) targets

A shiny-based application was built for the visualization of tissue-specific gene networks highlighting connections between disease-genes and drug(gene)-targets.

```
library(shiny)
library(visNetwork)
library(org.Hs.eg.db)

visualize.graph(tissue_scores = T2DM_Tscores,
                 disease_genes =T2DM_genes$entrez[1:5],
                 ppi_network = ppi_strdb_700,
                 tissue_expr_data = gtexv7_zscore,
                 top_targets = rownames(T2DM_top50)[1:5],
                 db='BP')
```

The following example shows how to use the function *build_tissue_specific_networks* which returns

- tissue-specific networks (igraph objects);
- shortest-paths linking a set of gene targets (e.g. top 5 from the tissue-specific efficacy score) to known disease-genes;
- a list of genes closely related to the set of the specified gene targets.

```
tsrwr = build.tissue.specific.networks(tissue_scores = T2DM_Tscores, disease_genes = T2DM_genes$entrez,
                                       ppi_network = ppi_strdb_700, tissue_expr_data = gtexv7_zscore,
                                       top_targets = rownames(T2DM_top50)[1:5], verbose = FALSE)
```

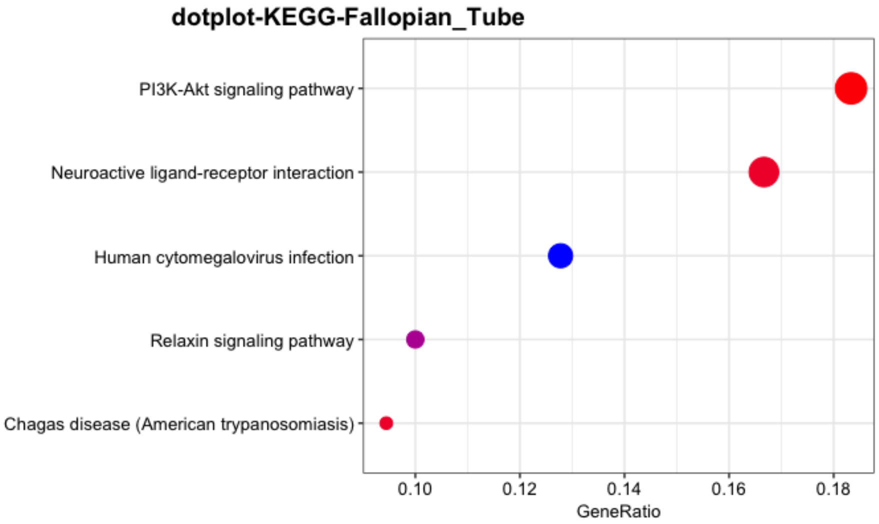
Then, ThETA provides functions to compile

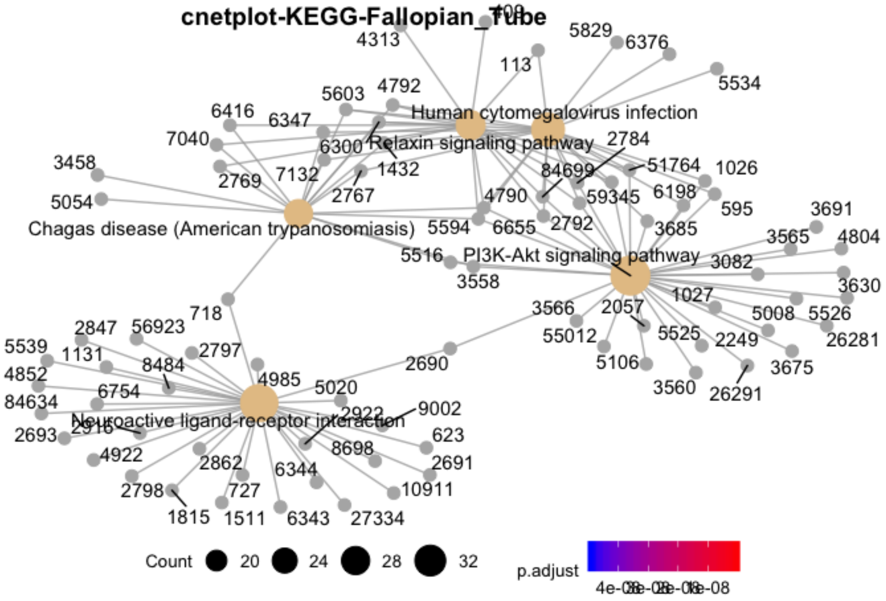
- biological annotations which are significantly associated with a set of genes (by using over-representation analysis);
- plots for interpreting the ORA analysis;
- pubmed trend plots based on a set of gene targets.

```
library(org.Hs.eg.db)
T2D_ora_data_shp = generate.ora.data(tsrwr$shp[[1]], databases = "KEGG")
#> [1] "ORA completed for KEGG-79068."
#> [1] "ORA completed for KEGG-54901."
#> [1] "ORA completed for KEGG-3087."
#> [1] "ORA completed for KEGG-169026."
#> [1] "ORA completed for KEGG-10644."
T2D_ora_data_rwr = generate.ora.data(tsrwr$rwr, databases = "KEGG")
#> [1] "ORA completed for KEGG-Fallopian_Tube."
#> [1] "ORA completed for KEGG-Kidney_Cortex."
#> [1] "ORA completed for KEGG-Liver."
#> [1] "ORA completed for KEGG-Testis."
#> [1] "ORA completed for KEGG-Thyroid."
#> [1] "ORA completed for KEGG-Uterus."
```

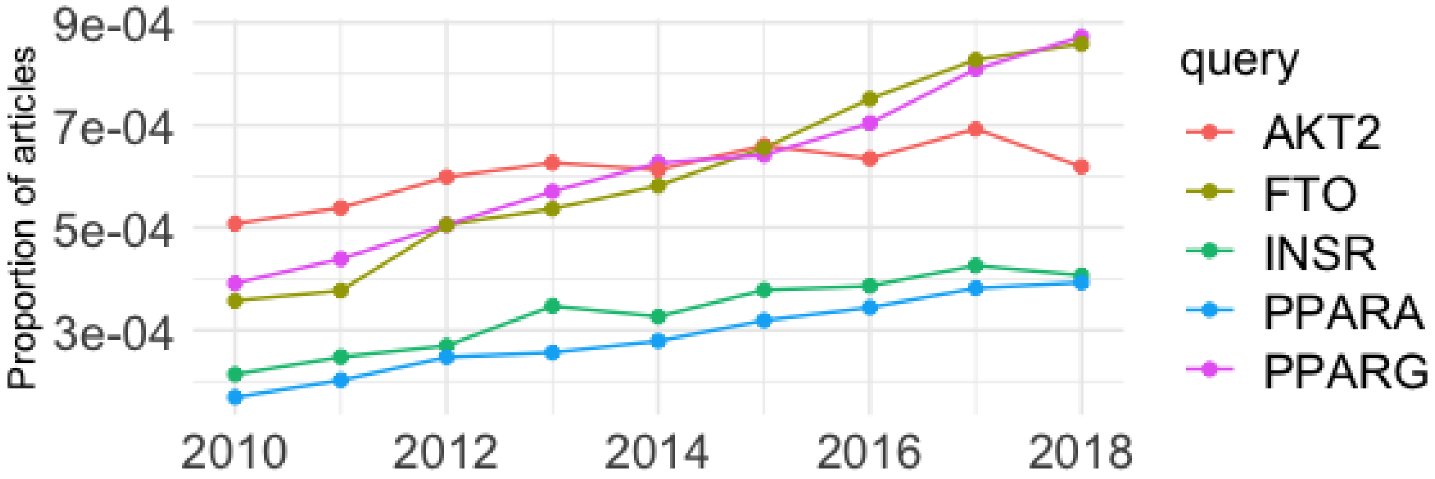
```
T2D_ora_plot_rwr = generate.ora.plots(T2D_ora_data_rwr, set_plots = c("dotplot","cnetplot"),
                                      showCategory = 5, font_size = 10)
#>  [1] "dotplot-KEGG-Fallopian_Tube"  "cnetplot-KEGG-Fallopian_Tube"
#>  [3] "dotplot-KEGG-Kidney_Cortex"   "cnetplot-KEGG-Kidney_Cortex"
#>  [5] "dotplot-KEGG-Liver"           "cnetplot-KEGG-Liver"
#>  [7] "dotplot-KEGG-Testis"          "cnetplot-KEGG-Testis"
#>  [9] "dotplot-KEGG-Thyroid"         "cnetplot-KEGG-Thyroid"
#> [11] "dotplot-KEGG-Uterus"          "cnetplot-KEGG-Uterus"
```

```
figure <- ggpubr::ggarrange(plotlist = T2D_ora_plot_rwr[1:2], nrow = 2, ncol = 1,
                            common.legend = TRUE, legend = "bottom", labels=names(T2D_ora_plot_rwr)[1:2])
figure
```


dotplot-KEGG-Fallopian_Tube
PI3K-Akt signaling pathway
Neuroactive ligand-receptor interaction
Human cytomegalovirus infection
Relaxin signaling pathway
Chagas disease (American trypanosomiasis)
0.10
0.12
0.14
0.16
0.18
GeneRatio


cnetplot-KEGG-Fallopian_Tube
Human cytomegalovirus infection
Relaxin signaling pathway
Chagas disease (American trypanosomiasis)
PI3K-Akt signaling pathway
Neuroactive ligand-receptor interaction
409
4313
5829
6376
113
5534
5603
4792
6416
6347
7040
6300
1432
2784
84699
51764
1026
3458
2769
7132
2767
4790
59345
6198
595
3691
5054
5594
6655
2792
3685
3565
4804
5516
3558
3082
3630
1027
3566
2057
5008
5526
55012
5525
2249
26281
5106
3675
3560
26291
718
2690
2847
56923
5539
1131
2797
8484
4852
4985
5020
6754
84634
2922
9002
2693
2916
623
8698
4922
2862
6344
2691
2798
727
10911
1815
1511
6343
27334
Count
20
24
28
32
p.adjust
4e-08
3e-08
2e-08
1e-08

```
library(org.Hs.eg.db)
pmc_genes = as.character(AnnotationDbi::mapIds(org.Hs.eg.db, rownames(T2DM_allsc)[c(1:5)], 'SYMBOL', 'ENTREZID'))
print(pmc_genes)
#> [1] "PPARG" "INSR"  "AKT2"  "FTO"   "PPARA"
T2D_pmd_plot_top = novelty.plots(pmc_genes, font_size = 14, pubmed = c(2010,2018))
T2D_pmd_plot_top
```


9e-04
7e-04
5e-04
3e-04
Proportion of articles
query
AKT2
FTO
INSR
PPARA
PPARG