Level 1: Module 5 – Functional Analysis in R

Table of Contents

PACKAGES ***2***

clusterProfiler **2**

GENERATE DEG VECTOR ***3***

FUNCTIONAL ENRICHMENT ANALYSIS ***4***

Gene Ontology **4**

KEGG (Kyoto Encyclopedia of Genes and Genomes) Analysis **5**

GENE-CONCEPT NETWORK ***6***

GLOBAL/ UNIVERSAL GENE SET ENRICHMENT ANALYSIS (GSEA) ***7***

TRANSCRIPTION FACTOR ANALYSIS ***10***

PREPARATION FOR EXTERNAL TOOLS ***12***

ALL DELIVERABLES ***12***

Before starting with the analysis:
Please refer to previous modules for more information.

- Identify top DEGs (Module 4)

PACKAGES

Install and load the following packages; and explore their documentation for more information on their functions and use cases.

- use `BiocManager::install("package_name")`
  - org.Hs.eg.db
  - clusterProfiler
  - enrichplot
  - msigdb
- use `install.packages("package_name")`
  - magrittr
  - tidyr
  - ggnewscale

The goal of this module is to identify the important functions related to our identified DEGs. By identifying these processes, we will be able to infer pathways and targets involved in the expression of the disease (colorectal cancer) and start developing possible gene and drug therapies.

clusterProfiler

The clusterProfiler package supports both hypergeometric tests and gene set enrichment analyses of many ontologies and pathways. It's a widely used package in bioinformatic analyses. Check out the link above for more information on this package and feel free to experiment with other functions it offers. For now, we'll be using the functions below to look at enriched processes.

GENERATE DEG VECTOR

This DEG vector will be used as the input for many of the analyses explained below. The vector should be a sorted, named, numeric vector of logFC (log fold change) values. This means the vector should contain numbers, each element of the vector should be named with the corresponding gene ID (ENTREZID), and the vector should be sorted. Follow the directions below to ensure that your DEG vector is formatted properly.

1. Extract the logFC column from your top table (created in Module 4) to a new vector variable.
2. Name the elements of your vector with the corresponding gene symbol (found as either the ID column or row names of your top table).
   Tip: You can access the names of your vector elements using `names(vector)`.
3. Define a threshold for fold change. If a gene has a logFC greater than the threshold, it will be included in further analysis. If not, it will be excluded. By setting a positive threshold, we are limiting our analysis to up-regulated genes. If you want, you can set a negative threshold and conduct analysis on down-regulated genes.
   Tip: I set my threshold to 1.5. However, you can set the threshold to whatever you want – it varies from dataset to dataset. A lower threshold will include more genes in your analysis. For this project, we want to look at no more than 500 genes and preferably more than 100.
4. Sort your vector in descending order.
5. Convert the gene symbols to ENTREZIDs using the `select()` function from Annotation DBI (similar to step 3 in Module 4) and **save as a new variable**. ENTREZIDs are another widely used way to identify genes and the database is curated by the NIH. You can learn more about ENTREZIDs here.
   Tip: You want two separate variables: the sorted, named, numeric vector and the data frame with ENTREZIDs.

The sorted, named, numeric vector should appear in your environment panel. The item should look similar to the image below:

The data frame with ENTREZIDs should appear similar to the image below:

	SYMBOL	ENTREZID
1	FOXQ1	94234
2	CEMIP	57214
3	TCN1	6947
4	TACSTD2	4070
5	MMP7	4316
6	CRNDE	643911

FUNCTIONAL ENRICHMENT ANALYSIS

In this analysis, we'll be looking at which pathways and functions are enriched in up-regulated genes. "Enriched" indicates that the pathway or process may be more active or dysregulated in diseased tissue compared with normal tissue.

Gene Ontology

Gene ontology defines concepts and classes used to describe gene function and relationships between these concepts. First, we'll be looking at the cellular components, molecular functions, and biological processes that our up-regulated DEG are involved in.

Identify Enriched Gene Ontology Terms

Function: `enrichGO()`

Input: ENTREZID

Additional Arguments:

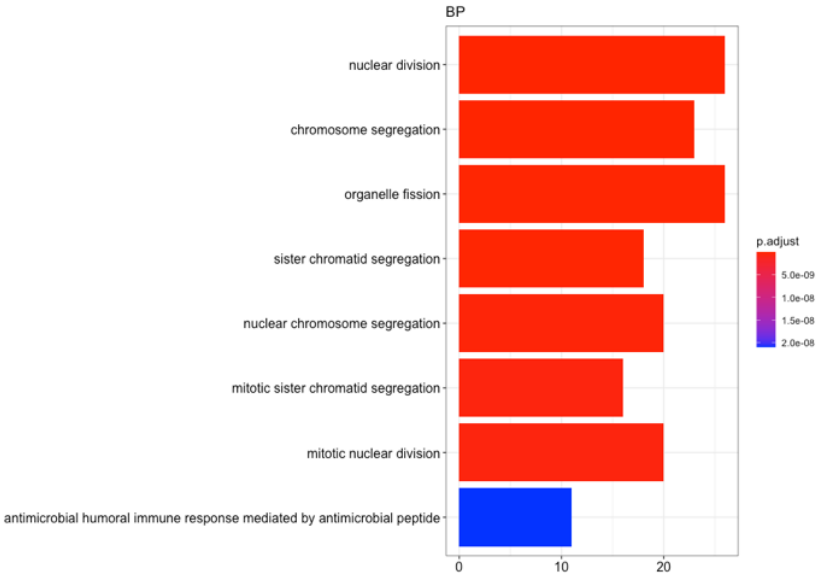
`OrgDb=org.Hs.eg.db`	specifies database for gene IDs
`Ont=["CC", "MF", "BP"]`	select 1; specifies the category to analyze
`readable=T`	map gene ID to gene name

Visualization of Results

Function: `barplot()`

Input: enrichResult object (output generated by `enrichGO()`)

Desired Output: Your visualization should look similar to the plot below.



Deliverable 1: 3 bar plots representing enriched gene ontology terms for cellular components, biological processes, and molecular functions of up-regulated DEG.

KEGG (Kyoto Encyclopedia of Genes and Genomes) Analysis

KEGG is a database resource for high-level functions and utilities of the biological system, such as the cell level, the organism, and the ecosystem. You can view more information on specific KEGG pathways here.

Identify Enriched KEGG Pathways

Function: `enrichKEGG()`

Input: ENTREZID

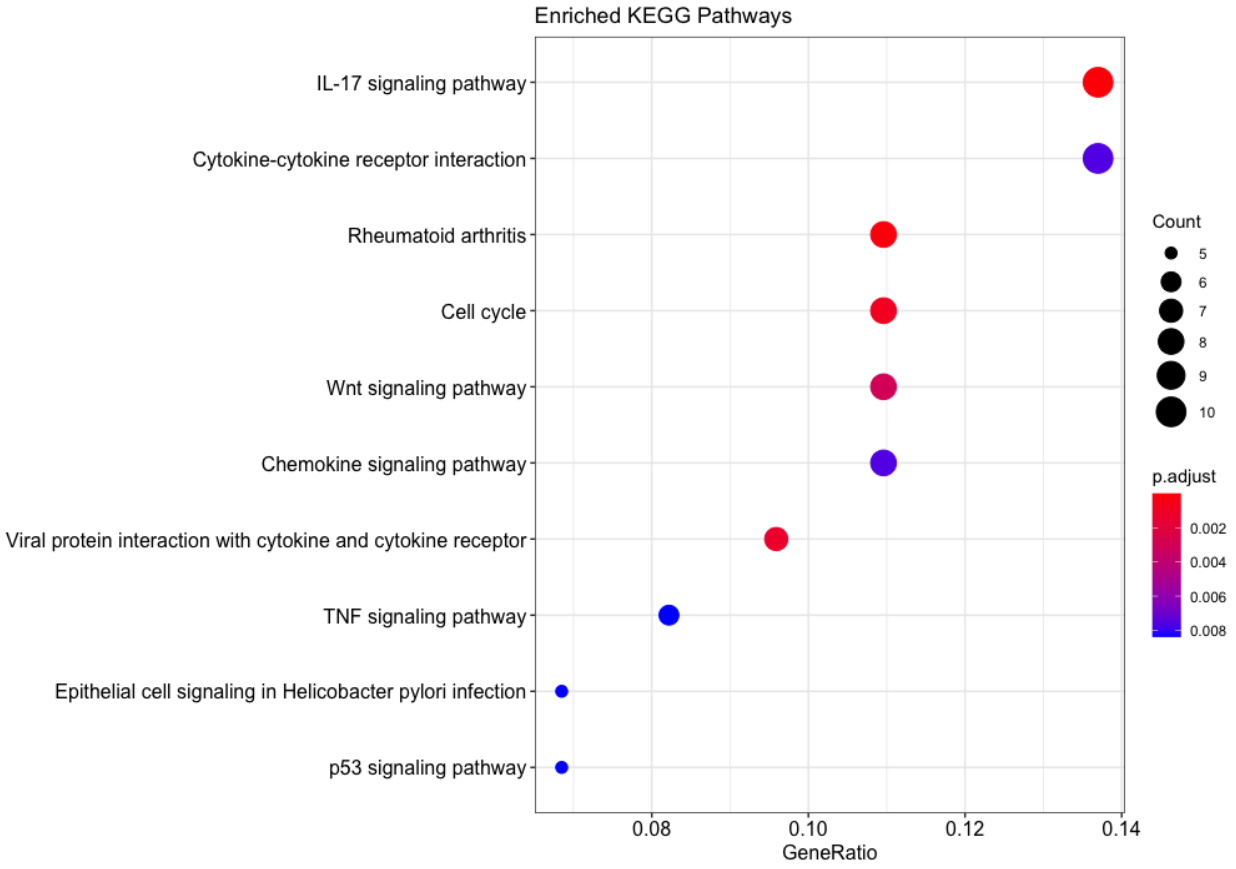
Visualization of Results

Function: `dotplot()`

Input: enrichResult object (output generated by `enrichKEGG()`)

Tip: For more information on how to interpret a dot plot, see this link.

Desired Output: Your visualization should look similar to the plot below.



Deliverable 2: 1 dot plot of enriched KEGG pathways.

GENE-CONCEPT NETWORK

Bar plots and dot plots from GO and KEGG analysis can only display the most significantly enriched terms. However, we also may want to know which genes are involved in those terms. To decide which genes may belong to multiple annotation categories, we can view a gene-concept network. This network depicts the linkages of genes and biological concepts with the genes as the colored dots and the biological concepts as the tan dots. Let's view the gene-concept network for our top 5 most significant KEGG pathways.

Convert ENTREZIDs to Gene Symbols

Function: `setReadable()`
Input: enrichResult object (output generated by `enrichKEGG()`)
Additional Arguments:

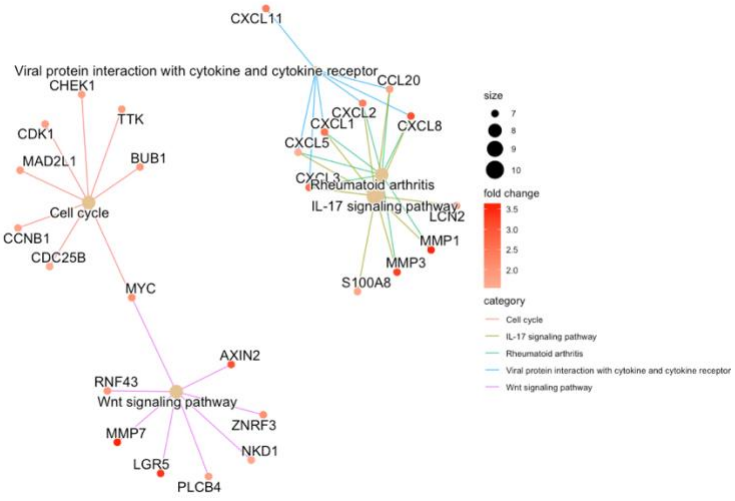
`OrgDb=org.Hs.eg.db`	specifies database for gene IDs
`ketType="ENTREZID"`	sets the key's type to ENTREZID

Visualization of Network

Function: `cnetplot()`
Input: enrichResult object (output generated by `setReadable()`)
Additional Arguments:

`foldChange=DEG.genes`	sorted, named, numeric vector
`categorySize="pvalue"`	sets the key's type to ENTREZID

Desired Output: Your visualization should look similar to the plot below.



Tip: You may not see the lines when view your plot in R. Try exporting your image to view the connections.

Deliverable 3: 1 gene-concept network of enriched KEGG pathways.

GLOBAL/ UNIVERSAL GENE SET ENRICHMENT ANALYSIS (GSEA)

Now, that we've done a ton of analysis our up-regulated genes and the processes they are involved in, we want to extract meaning out our sorted, named, numeric vector (this vector should include all DEGs, unfiltered). From this list indicating gene expression importance, we want to interpret the functional annotations to discover how the gene expressions are associated with certain biological functions or have certain molecular functions.

The most common approach is to focus on the top-most and bottom-most genes. However, this approach has a few limitations. One such limitation of this method is that it doesn't account for genes that vary between, which is a trend that also affects pathway analysis. To overcome these limitations, we perform GSEA.

GSEA is used for the evaluation of microarray data at the level of gene sets. These are *a priori* defined gene groups based on knowledge like published data and research. The goal is to determine whether the members in a particular gene set are randomly distributed or tend to occur towards the top or bottom of the ranked list. If they show the latter distribution, the gene set is correlated with the phenotypic class distinction.

GSEA-MSigDB has gene sets divided into eight major collections. For this analysis we'll be using the **H: hallmark gene sets** collection.

There are two ways to access this data set:

1. Download the appropriate file from the webpage linked above and read the data into R. The download will be a GMT file, so you will need to use `read.gmt()` in order to parse the file properly.

2. Use the msigdb package in R. This package consists of MsigDB gene sets in tidy data format.

   Obtain Gene Set
   Function: `msigdbr()`
   Input:

`species="Homo sapiens"`	specifies species for gene sets
`category="H"`	specifies we want the "H" gene set

   Select Desired Information
   Function: `h <- msig %>% select (gs_name, entrez_gene)`

Generating Global DEG Vector

The first step of this analysis is to generate our gene vector. **This vector is different from the one generated in the beginning of this module.** This vector consists of sorted logFC (in descending order) of our all of our DEGs (not just up-regulated) named with ENTREZIDs. Follow the directions below to ensure that your DEG vector is formatted properly.

1. Start by converting the gene symbols in your top table (from Module 4) to ENTREZIDs using the `select()` function from Annotation DBI (similar to step 3 in Module 4).
2. Remove duplicate symbols.
3. Merge your `select()` output with your top table by row names (gene symbols).
   Tip: You may have to set `rownames(data)` to be the gene symbols (data$ID for the top table and data$SYMBOL for the `select()` output) in order for the merge to work.
4. Extract the logFC column from the merged data to a new vector variable.
5. Name the elements of your vector with the corresponding ENTREZID (found as ENTREZID column of the merged data).
6. Sort your vector in descending order.

Perform GSEA Analysis

Function: `GSEA()`
Input: DEG vector (from the steps directly above)
Additional Arguments:

`TERM2GENE=h`	where `h` is the data frame mentioned on the pervious page or the output of `read.gmt()`

Tip: If you receive an error similar to "no term enriched under specific pvalue cutoff" play around with other parameters of the function (ex. minGSSize, pvalueCutoff, etc.)

Visualize GSEA Plot

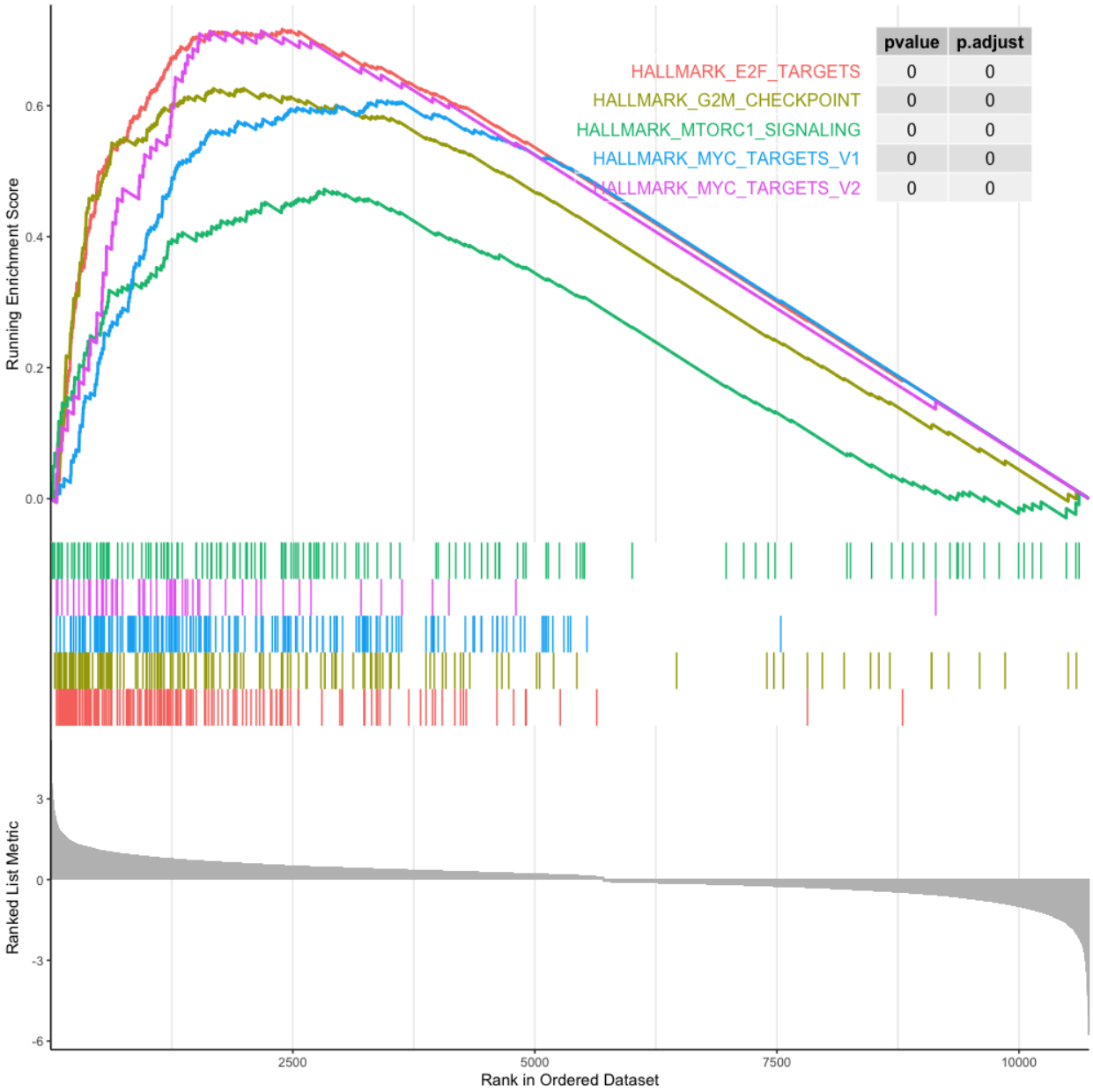
Function: `gseaplot2()`
Input: gseaResult object (output from `GSEA()`)
Additional Arguments:

`geneSetID=#`	# can be any index form such as a single number or a range (ex. 1; 2:6, etc.)

Tip: geneSetID specifies which gene set you want to look at. You can specify a single gene set (using a single number), a range of gene sets (using standard R indexing – e.x,. `2:5`), or selected gene sets (using a vector – e.x., `c(2,5,8,10)`).

Tip: You can view the names of the possible gene sets using the command below. This will help you identify the number of gene sets which might be of interest for further analysis. The number corresponds to the index of the gene set in the array outputted by: `gsea@result$ID`.

Desired Output: Your visualization should look similar to the plot below.



Tip: You can interpret your plot using the helpful explanations on the [GSEA-MSigDB website](#).

Deliverable 4: 1 GSEA plot.

TRANSCRIPTION FACTOR ANALYSIS

Now that we've analyzed the enriched and up-regulated genes in depth, we should also look at regulatory factors such as transcriptional factors, microRNAs, and lncRNAs. Looking at these items will enable us to look into not only the genes of interest but also how they are regulated. We'll do this by looking at a gene-concept network of transcription factors.

First, we want the **C3: Regulatory target gene sets** from the GSEA-MsigDB. Follow similar steps to those above under the GSEA section to obtain the relevant data set.

Analyze the C3 Gene Set Collection

Function: `enrichr()`
Input: ENTREZID (**$ENTREZ of the data set created on page 3**)
Additional Arguments:

`TERM2GENE=c3`	where `c3` is the data frame mentioned above or the output of `read.gmt()`

Convert ENTREZIDs to Gene Symbols

Function: `setReadable()`
Input: enrichResult object (output generated by `enrichr()`)
Additional Arguments:

`OrgDb=org.Hs.eg.db`	specifies database for gene IDs
`ketType="ENTREZID"`	sets the key's type to ENTREZID

Visualization of Network

Function: `cnetplot()`
Input: setReadable object (output generated by `setReadable()`)
Additional Arguments:

`foldChange=DEG.genes`	sorted, named, numeric vector
`categorySize="pvalue"`	sets the key's type to ENTREZID

Tip: If you receive an error similar to "Error in graph.data.frame… the data frame should contain at least 2 columns," check your output from enricher. If you see something similar to the image below where the S4 object has dimensions [0x9], you need to expand your data set of DEG. To do this follow the instructions below the image.

e	S4 [0 x 9] (DOSE::enrichResult	S4 object of class enrichResult
result	list [1864 x 9] (S3: data.frame	A data.frame with 1864 rows and 9 columns
pvalueCutoff	double [1]	0.05
pAdjustMethod	character [1]	'BH'
qvalueCutoff	double [1]	0.2
organism	character [1]	'UNKNOWN'
ontology	character [1]	'UNKNOWN'
gene	character [178]	'94234' '57214' '6947' '4070' '4316' '643911' ...
keytype	character [1]	'UNKNOWN'
universe	character [26190]	'10257' '23172' '81' '90' '8754' '11096' ...
gene2Symbol	character [0]	
geneSets	list [2167]	List of length 2167
readable	logical [1]	FALSE
termsim	double [0 x 0]	
method	character [0]	

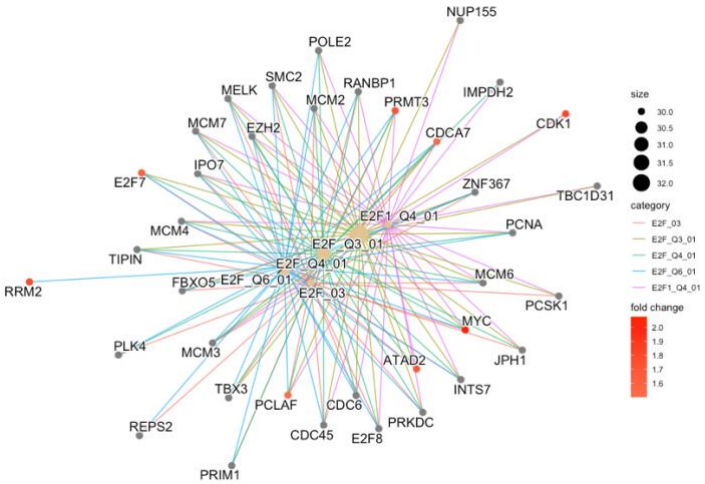
Replace the ENTREZID argument in the `enrichr()` function with the following

```
names(gsea_DEG_vector[gsea_DEG_vector > #])
```

Where `gsea_DEG_vector` is the DEG vector generated in the GSEA section and `#` is the threshold value discussed in step 3 of the Generate DEG Vector section on page 3. Specify a `#` lower than you did originally and rerun the `enrichr()` function until you get non-zero dimensions.

My function worked with a threshold of 1.

Desired Output: Your visualization should look similar to the plot below.



Deliverable 5: 1 gene-concept network of transcription factors.

PREPARATION FOR EXTERNAL TOOLS

For Module 6, we'll be conducting functional analysis using a subset of the plethora of external, web-based tools. For these tools, we need a text file of gene symbols or ENTREZIDs sorted by logFC. You can create this text file simply by writing out the names of your up-regulated DEG vector (from the Generate DEG Vector section).

Function: `write()`
Input: names of DEG vector
Additional Arguments:

`file="path/file.txt"`	desired output file name and path

ALL DELIVERABLES

1. **3 bar plots representing enriched gene ontology terms for cellular components, biological processes, and molecular functions of up-regulated DEG.**
2. **1 dot plot of enriched KEGG pathways.**
3. **1 gene-concept network of enriched KEGG pathways.**
4. **1 GSEA plot.**
5. **1 gene-concept network of transcription factors.**