`psygenet2r`: a R/Bioconductor package for the analysis of psychiatric disease genes

Alba Gutierrez-Sacristan Carles Hernandez-Ferrer Juan R. Gonzalez
Laura I. Furlong

February 27, 2017

Contents

Introduction

The `psygenet2r` package contains functions to query PsyGeNET [?], a resource on psychiatric diseases and their genes. The `psygenet2r` package includes analysis functions to study psychiatric diseases, their genes and disease comorbidities. A special focus is made on visualization of the results, providing a variety of representation formats such as networks, heatmaps and barplots (Table ??).

Background

During the last years there has been a growing interest in the genetics of psychiatric disorders, leading to a concomitant increase in the number of publications that report these studies [?]. However, there is still limited understanding on the celular and molecular mechanisms leading to psychiatric diseases, which has limited the application of this wealth of data in the clinical practice. This situation also applies to psychiatric comorbidities. Some of the factors that explain the current situation is the heterogeneity of the information about psychiatric disorders and its fragmentation into knowledge silos, and the lack of resources that collect these wealth of data, integrate them, and supply the information in an intuitive, open access manner to the community. PsyGeNET has been developed to fill this gap. `psygenet2r` has been developed to facilitate statistical analysis of PsyGeNET data, allowing its integration with other packages available in R to develop data analysis workflows.

PsyGeNET is a resource for the exploratory analysis of psychiatric diseases and their associated genes. The second release of PsyGeNET (version 2.0) contains updated information on depression, bipolar disorder, alcohol use disorders and cocaine use disorders, and has been expanded to cover other psychiatric diseases of interest: bipolar disorder, schizophrenia, substance-induced depressive disorder and psychoses and cannabis use disorder (Table ??). PsyGeNET allows the exploration of the molecular basis of psychiatric disorders by providing a comprehensive set of genes associated to each disease. Moreover, it allows the analysis of the molecular mechanisms underlying psychiatric disease comorbidities.

Table 1: Psychiatric diseases included in PsyGeNET

Long Name	Short Name	Acronym
`Alcohol use disorders`	Alcohol UD	AUD
`Bipolar disorders and related disorders`	Bipolar disorder	BD
`Depressive disorders`	Depression	DEP
`Schizophrenia spectrum and other psychotic disorders`	Schizophrenia	SCHZ
`Cocaine use disorders`	Cocaine UD	CUD
`Substance induced depressive disorder` SI-Depression	SI-DEP	
`Cannabis use disorders`	Cannabis UD	CanUD
`Substance induced psychosis`	SI-Psychosis	SI-PSY

PsyGeNET database is the result of the data extracted from the literature by text mining using BeFree [?], followed by manual curation by domain experts. A team of 22 experts participates as curators of the database. The current version of PsyGeNET (version 2.0) contains 3,771 associations between 1,549 genes and 117 psychiatric disease concepts.

With `psygenet2r` package the user is able to submit queries to PsyGeNET from R, perform a variety of analysis on the data, and visualize the results through different types of graphical representations.

The tasks that can be performed with `psygenet2r` package are the following:

1. Retrieve Gene-Disease Associations (GDAs) from PsyGeNET using as query a gene or a disease (single or a set of genes/diseases) of interest

2. Visualize the results according to the GDAs' attributes: PsyGeNET Evidence Index, number of publications, sentences that report the GDA, source dadatabase
3. Visualize the results according to the disease (disease class) or gene (Panther class) attributes
4. Analyze the association between two diseases based on shared genes (using the Jaccard index)
5. Characterizing the disease genes by molecular function using Panther classes or expression site using TopAnat / Bgee database.

In the following sections the specific functions that can be used to address each of these tasks are presented.

Installation

The package `psygenet2r` is provided through Bioconductor. To install `psygenet2r` the user must type the two following commands in an R session:

```
source( "http://bioconductor.org/biocLite.R" )
biocLite( "psygenet2r" )
```

```
library( psygenet2r )
```

DataGeNET.Psy object

`DataGeNET.Psy` object is obtained when `psygenetGene` and `psygenetDisease` functions are applied. This object is used as input for the rest of `psyGeNET2r` functions, like the `plot` function.

`DataGeNET.Psy` object contains all the information about the different diseases/genes associated with the gene/disease of interested retrieved from PsyGeNET. This object contains a summary of the search, such as the search input (gene or disease), the selected database, the gene or disease identifier, the number of associations found (N. Results) and the number of unique results obtained (U. Results).

```
## Error:  'replacement' must be a character vector
```

```
t1

## Error in eval(expr, envir, enclos):  object 't1' not found

class( t1 )

## Error in eval(expr, envir, enclos):  object 't1' not found
```

This object comes with a series of functions to allow users to interact with the information retrieved from PsyGeNET. These functions are `ngene`, `ndisease`, `extract` and `plot`. The first function `ngene` returns the number of retrieved genes for a given query. `ndisease` is the homologous function but for the diseases. The function `extract` returns a formatted `data.frame` with the complete set of information downloaded from PsyGeNET. Finally, the `plot` function allows the visualization of the results in a variety of ways such as gene-disease association networks or heatmaps.

PsyGeNET and psygenet2r

The PsyGeNET web interface can be explored by searching a specific gene or a specific disease, and **psygenet2r** package has the same options. Therefore, the starting point for **psygenet2r** are **psygenetGene** and **psygenetDisease** functions.

PsyGeNET data is classified according to the database used as a source of information ("source database"). Therefore, any query run on PsyGeNET requires to specify the source database using the argument called **database**. Table **??** shows the source databases in PsyGeNET and their description. By default, the database **ALL** is used in **psygenet2r**. For illustrating purposes along the vignette, database **ALL** will be used in most of code snippets.

Table 2: Source databases included in PsyGeNET

Name	Description
psycur15	Genes associated to DEP, BD, AUD and CUD between 1980 and 2013 (PsyGeNET release v1.0)
psycur16	Genes associated to DEP, BD, AUD, CUD, SCHZ, S-DEP, CanUD and D-PSY between 1980 and 2015
ALL	All previous Databases

Retrieve gene-disease associations (GDAs) from psygenet2r

Using genes as a query

psygenet2r package allows exploring PsyGeNET information using a specifc gene or a list of genes. It retrieves the information that is available in PsyGeNET (associated diseases, source database, PsyGeNET Evidence Index, number of publications, attributes of genes, etc) and allows to visualize the results in different ways.

Using as a query a single gene

In order to look for a single gene into PsyGeNET, we can use the **psygenetGene** function. This function retrieves PsyGeNET's information using both, the NCBI gene identifier and the official Gene Symbol from HUGO. It contains also other arguments like the database to query, the PsyGeNET evidence index (score argument).

As an example, the gene *NPY*, whose entrez id is *4852* is queried using **psygenetGene** function, and using alternatively the official HUGO Gene Symbol. In this example database **"ALL"**.

```
t1 <- psygenetGene( gene = 4852,
                    database = "ALL")

## Error:  'replacement' must be a character vector

t1

## Error in eval(expr, envir, enclos):  object 't1' not found
```

```
t2 <- psygenetGene( gene = "NPY", database = "ALL" )
t2

## Object of class 'DataGeNET.Psy'
##   . Type:          gene
##   . Database:      ALL
##   . Term:          NPY
##   . N. Results:    13
##   . U. Diseases:   13
##   . U. Genes:      1
```

```
plot( t1, type = "individual disease" )
    ## Error in plot(t1, type = "individual disease"):  object 't1' not found
```

Both cases result in an `DataGeNET.Psy` object:

```
class( t1 )

## Error in eval(expr, envir, enclos):  object 't1' not found

class( t2 )

## [1] "DataGeNET.Psy"
## attr(,"package")
## [1] "psygenet2r"
```

In the particular example used, by inspecting the `DataGeNET.Psy` object, we can see that the gene *NPY* is associated to 13 different diseases in PsyGeNET (with no restriction on the PsyGeNET evidence index).

Ploting the results of a Single Gene Query

`psygenet2r` offers several options to visualize the results from PysGeNET in networks by changing the `type` argument when applying the `plot` function. A network showing the diseases (`type = "individual disease"`) or the psychiatric disorders (`type = "disease class"`) related to the gene of interest is obtained.

By default, `psygenet2r` shows a network when ploting a `DataGeNET.Psy` object obtained by a gene-query. The result is a network where green nodes are diseases and the orange node is the gene of interest.

On the other hand, results can be visualized according to the 8 psychiatric disorders classes available in PsyGeNET (depression, bipolar disorder, alcohol use disorders, cocaine use disorder, bipolar disorder, schizophrenia, cannabis use disorder, substance-induced depressive disorder and psychoses) setting the `type` argument to `"disease class"`. As a result, a network with a maximum of 9 nodes is obtained (8 nodes that represent the psychiatric disorders and 1 node that represents the gene). The node's size of each psychiatric disorder is proportional to the number of disease concepts that belongs to each disease class, from the total number of diseases associated to the gene.

```
plot( t1, type = "disease class" )
      ## Error in plot(t1, type = "disease class"):  object 't1' not found
```

In our example, NPY is associated to four of the eight psychiatric disorders present in PsyGeNET, with an important contribution of depression.

Using as a query a list of genes

In the same way, `psygenet2r` allows to query PsyGeNET given a list of genes of interest. The same function, `psygenetGene`, accepts a vector of NCBI gene identifiers or HUGO official gene symbols.

To illustrate this functionality, a list of 20 genes was extracted from the article entitled *"The Genetics of Major Depression"* [**?**], where these genes are associated to depression. The vector of genes can be defined as follows:

```
genesOfInterest <- c( "COMT", "CLOCK", "DRD3", "GNB3", "HTR1A",
                      "MAOA", "HTR2A","HTR2C", "HTR6", "SLC6A4",
                      "ACE",  "BDNF", "DRD4", "HTR1B", "HTR2B",
                      "HTR2C", "MTHFR", "SLC6A3", "TPH1", "SLC6A2",
                      "GABRA3"
)
```

Then, the function `psygenetGene` is applied. In this case an extra argument called `verbose` was set to `TRUE`. This shows some information during the query-process, for example the message informing that there are repeated genes in the list (the gene *HTR2C* was placed twice in the list to raise this message) and the message informing that one or more of the given genes is not in PsyGeNET (in this case the gene HTR2B).

```
m1 <- psygenetGene(
    gene     = genesOfInterest,
    database = "ALL",
    verbose  = TRUE
)

## Warning in psygenetGene(gene = genesOfInterest, database = "ALL", verbose = TRUE):
Removing duplicates from input genes list.
## Staring querying PsyGeNET for COMT, CLOCK, DRD3, GNB3, HTR1A, MAOA, HTR2A, HTR2C,
HTR6, SLC6A4, ACE, BDNF, DRD4, HTR1B, HTR2B, MTHFR, SLC6A3, TPH1, SLC6A2, GABRA3 in
ALL database.
## Warning in psygenetGene(gene = genesOfInterest, database = "ALL", verbose = TRUE):
One or more of the given genes is not in PsyGeNET ( 'ALL' ):
##    - HTR2B
```

A `DataGeNET.Psy` object is obtained. In this particular example, 19 genes are present in PsyGeNET and are associated to 42 diseases, involving 212 GDAs.

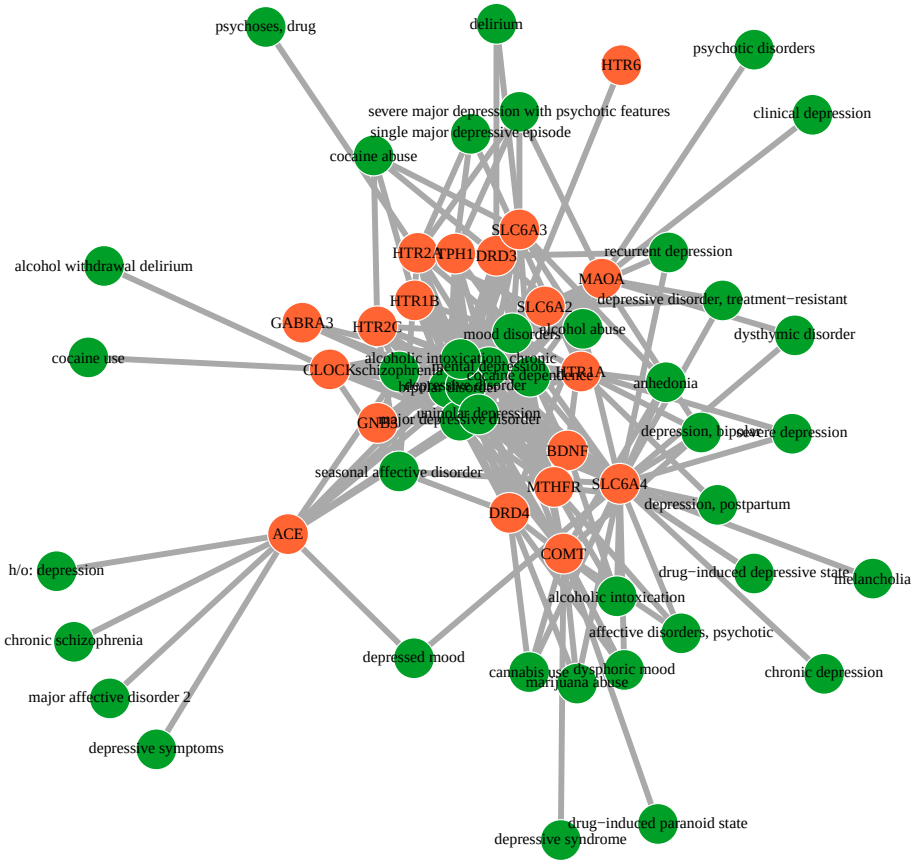
```
m1

## Object of class 'DataGeNET.Psy'
##   . Type:          gene
##   . Database:      ALL
##   . Term:          COMT ... GABRA3
##   . N. Results:    212
##   . U. Diseases:   42
##   . U. Genes:      19
```

Ploting the results of the query using a list of genes

`psygenet2r` provides several options to visualize the results of these queries, such as networks, heatmaps and barplots.

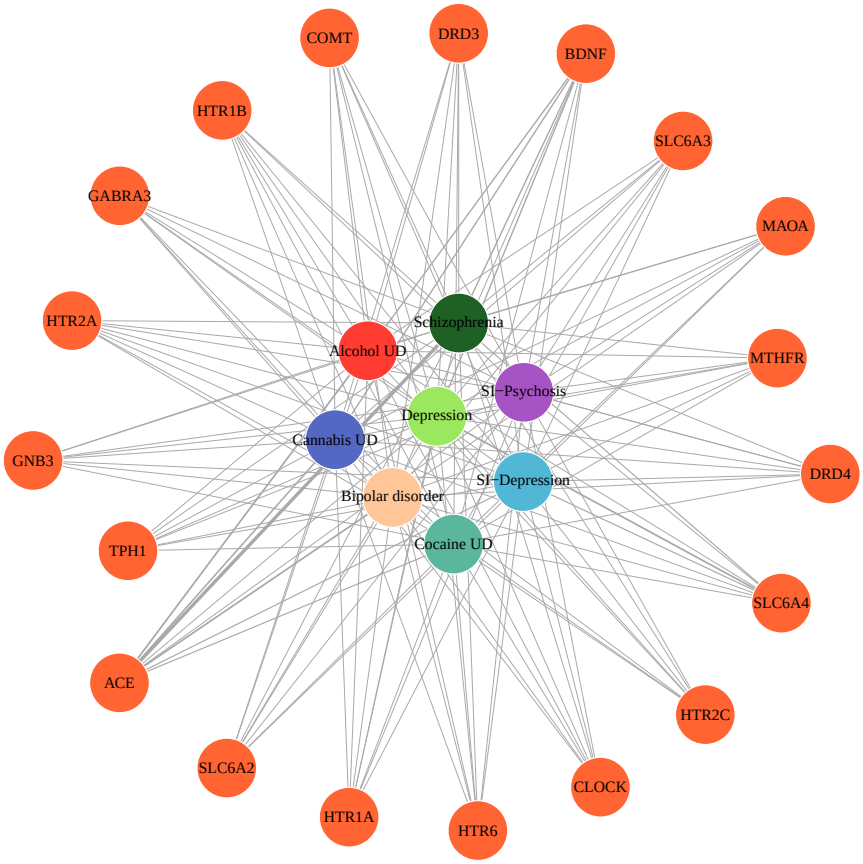
As for the single gene example, the default option in `psygenet2r` results is a network chart, where the green nodes represent diseases and the oranges nodes represent genes.

```
plot( m1 )
```



It is also possible to visualize the results by grouping the diseases according to the psychiatric disorders present in PsyGeNET (depression, bipolar disorder, alcohol use disorders, cocaine use disorder, bipolar disorder, schizophrenia, cannabis use disorder, substance-induced depressive disorder and psychoses) setting the `type` argument to `"disease class"`. As a result, a psychiatric disorder genes network is obtained. The edge's size is proportional to the number of disease concepts that belongs to each disease class, from the total number of diseases associated to the gene.

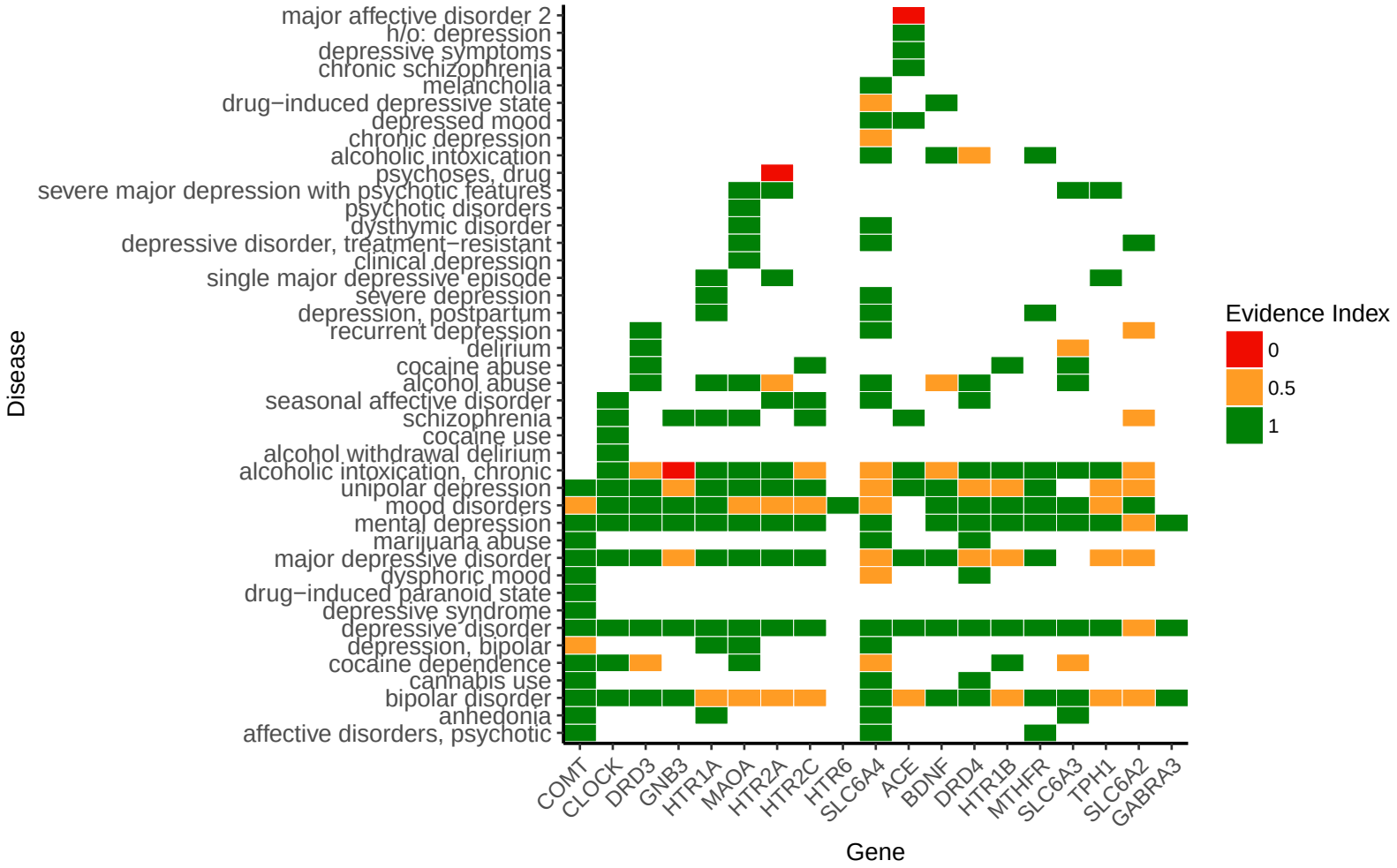
```r
plot( m1, type = "disease class" )
```



`psygenet2r` package allows to visualize the GDAs attributes in a heatmap. The argument `type` must be `"heatmapGenes"` and the PsyGeNET evidence index can also be determined by the user setting the `cut-off` argument to the evidence index of interest. In this example, the cut-off is set to 0, in order to obtain all the results. If we set the cut-off to 0.5, only those associations with at least half of the publications supporting the association will be shown. In this kind of representation we can identify genes that are associated to several diseases (e.g. SLC6A4), others that are associated only to one disease (e.g. HTR6) and we can visualize the evidence index for each association.

Note that heatmap cells can be coloured in green, yellow or red. Green cells represent those GDAs where all the evidences reviewed by the experts support the existence of an association between the gene and the disease (EI = 1); it will be yellow when there is contradictory evidence for the GDA (some publications support the association while others publications do not support it, $1 >$ EI > 0), and it will be red when all the evidences reviewed by the experts report that there is no association between the gene and the disease (EI = 0).

```
plot( m1, type="heatmapGenes" )
```

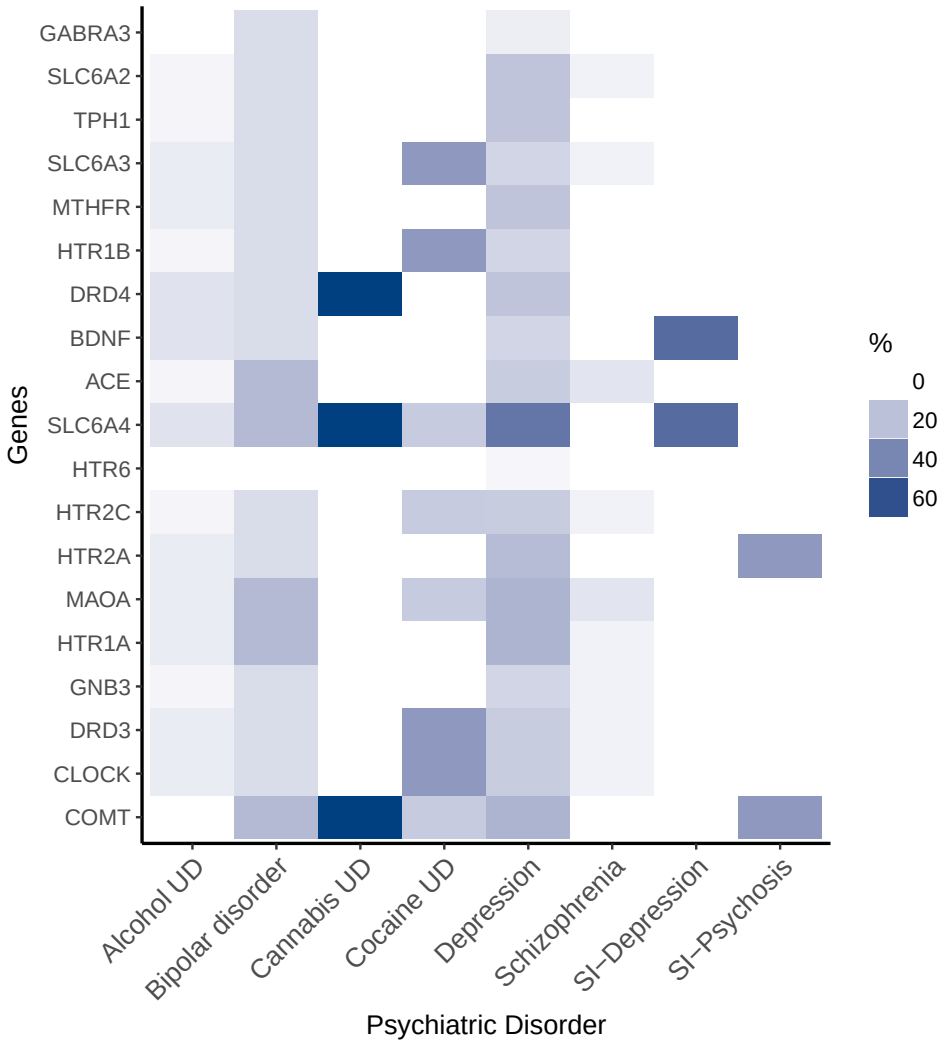


In this example we can see that there are 3 GDAs in red (GNB3-chronic alcoholic intoxication; HTR2A-drug psychoses and ACE-major affective disorder 2), indicating that all publications report that there is no association.

An alternative graphical representation is a psychiatric disorder heatmap related to the percentage of diseases in a class to which a gene is associated. It allows to analyze if the genes that are being studied present a specific association with a subtype of disorder or if they are associated with several of them in the same psychiatric disorder class. The percentage of diseases to which a gene appear associated with each psychiatric disorder is estimated. This percentage is relative to the total number of subtypes of disorders present in PsyGeNET (33 for alcohol UD, 9 for bipolar disorder, 37 for depression, 24 for schizophrenia, 6 for cocaine UD, 3 for cannabis UD, 3 for DI-Psychosis and 2 for SI-Depression). The resulting values are represented in a heatmap according to a blue color scale.

In order to obtain this graphic, `type` must be set to `"heatmap"`. The resulting heatmap shows which are the genes that are higher or lower associated to each one of the psychiatric disorders.

```
plot( m1, type = "heatmap" )
```



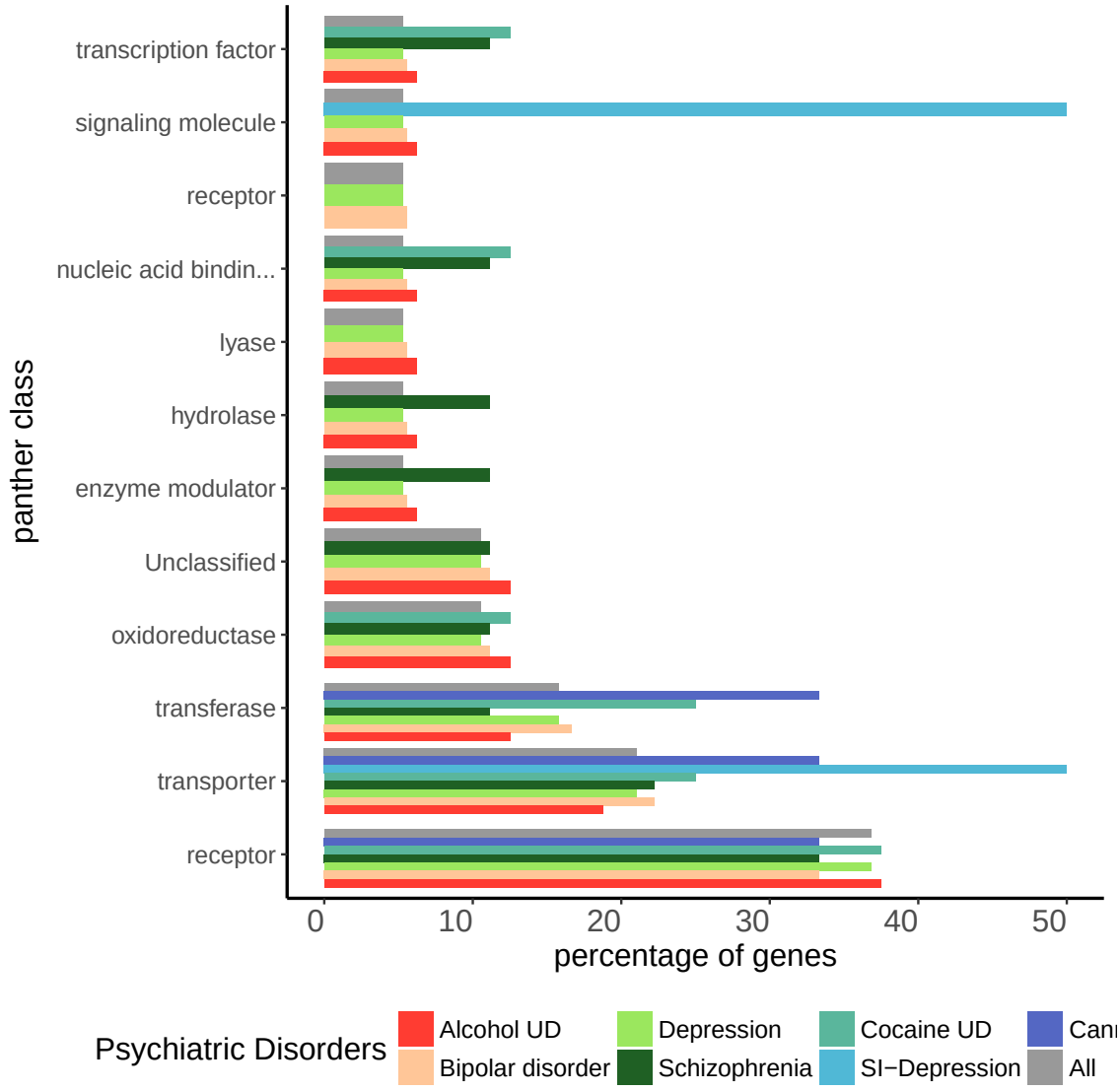
As it could be expected for the genes that are used in this query, all of them are associated to depression, to a greater or lesser extent. Some of them are also associated to the other six

psychiatric disorders, following a similar pattern alcohol UD and bipolar disorder.

`psygenet2r` package also allows to analyze a gene list according to the function of the proteins encoded by these genes. The PANTHER Protein Class Ontology classifies proteins according to their function.

The `pantherGraphic` function shows the Panther class of the proteins for each psychiatric disorder class. It provides a graphic with the results of these analysis, being the input a list of genes and the database (`ALL`, `psycur15`, `psycur16`). The input genes can be from a vector that contains the genes of interest, or from the genes obtained in the `DataGeNET.psy` object in a disease or disease-list query. A score argument can be added to filter results. It can also be done given a DataGeNET.psy object obtained by querying with a single gene.

```
genesOfInterest <- unique( genesOfInterest )
pantherGraphic( genesOfInterest, "ALL" )

## Warning in psygenetGene(geneList, database, verbose = verbose):  One or more of
the given genes is not in PsyGeNET ( 'ALL' ):
##    - HTR2B
```

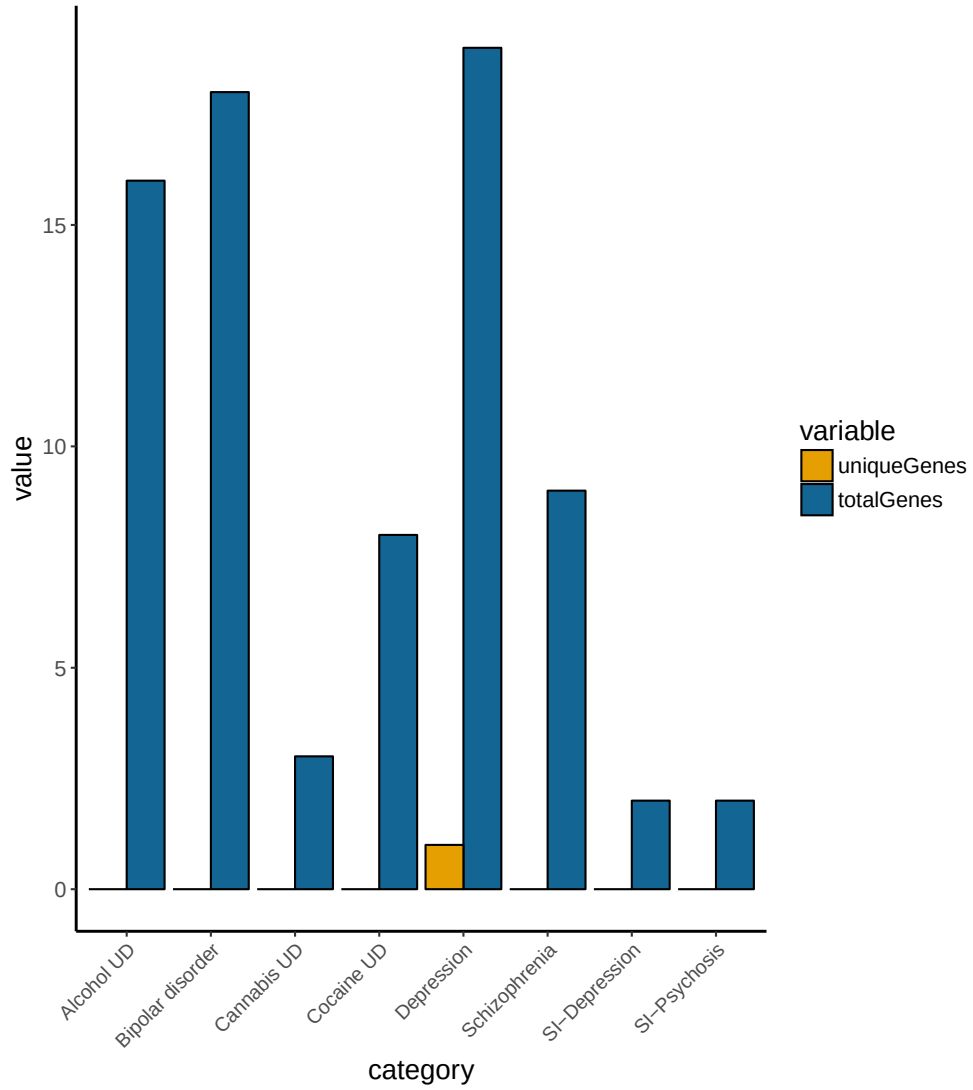


`psygenet2r` offers additional options to visualize the results of a query using a list of genes or from

the genes obtained in the `DataGeNET.psy` object in a gene or gene-list query. Barplots and pie charts showing the gene attributes can be obtained by applying the `geneAttrPlot` function.

`psygenet2r` package allows to visualize how many of our genes of interest are associated to each psychiatric disorder present in PsyGeNET and how many of them are exclusively associated to a particular psychiatric disorder. This can be done applying the `geneAttrPlot` function and setting the `type` argument to `"cateogry"`.

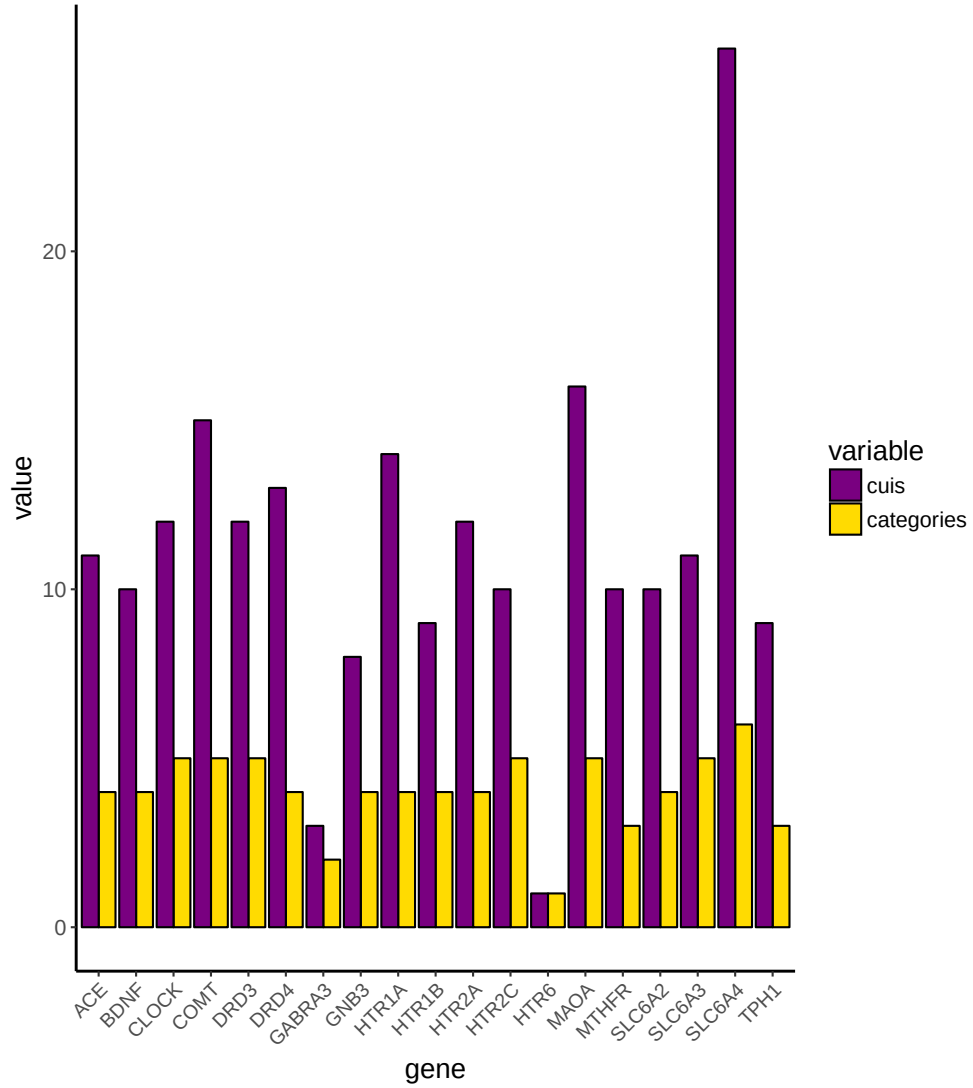
```
geneAttrPlot( m1, type = "category" )
```



As a result, a barplot is obtained. The X axis contains the psychiatric disorders, sorted alphabetically and the number of genes are represented in the Y axis. For each psychiatric disorder two bars are plotted, the blue bar represents the total number of genes for the psychiatric disorder. For those cases in which some of these genes are specific for the disorder according to PsyGeNET, a second bar coloured in orange will be displayed.

Alternatively, `psygenet2r` package allows to visualize for each gene, how many disease concepts and how many psychiatric categories are associated to it. This can be done applying the `geneAttrPlot` function and setting the `type` argument to `"gene"`.

```
geneAttrPlot( m1, type = "gene" )
```

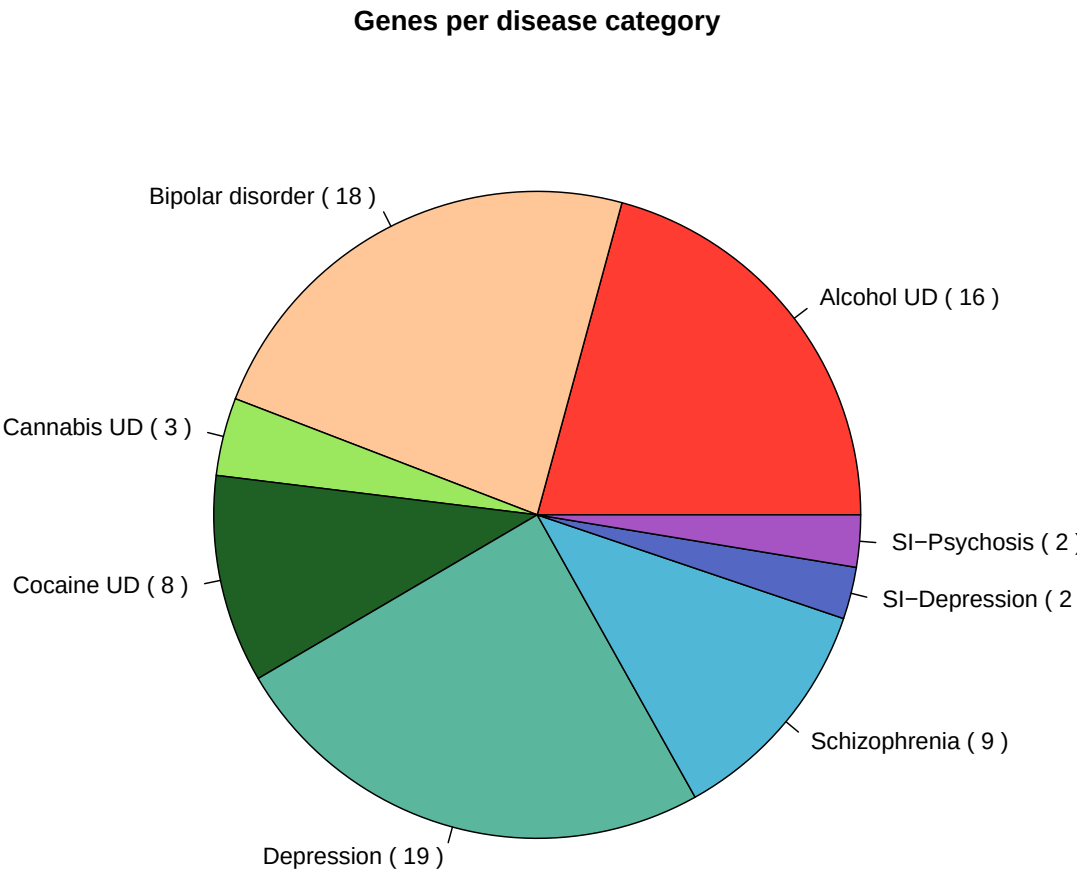


As a result, a barplot is obtained. The X axis contains the genes, sorted alphabetically and the number of cuis and psychiatric categories are represented in the Y axis. For each psychiatric disorder two bars are plotted, the purple bar represents the number of cuis and the yellow one belongs to the number of categories.

In our example, the gene SLC6A4 is the one whith more associated cuis (26) and categories (6).

If we are only interested in how many of our input genes are associated to each disease category, `psygenet2r` package allows to visualize it in a pie chart by applying the `geneAttrPlot` function and setting up the `type` argument to `pieChart`.

```
geneAttrPlot( m1, type = "pie" )
```

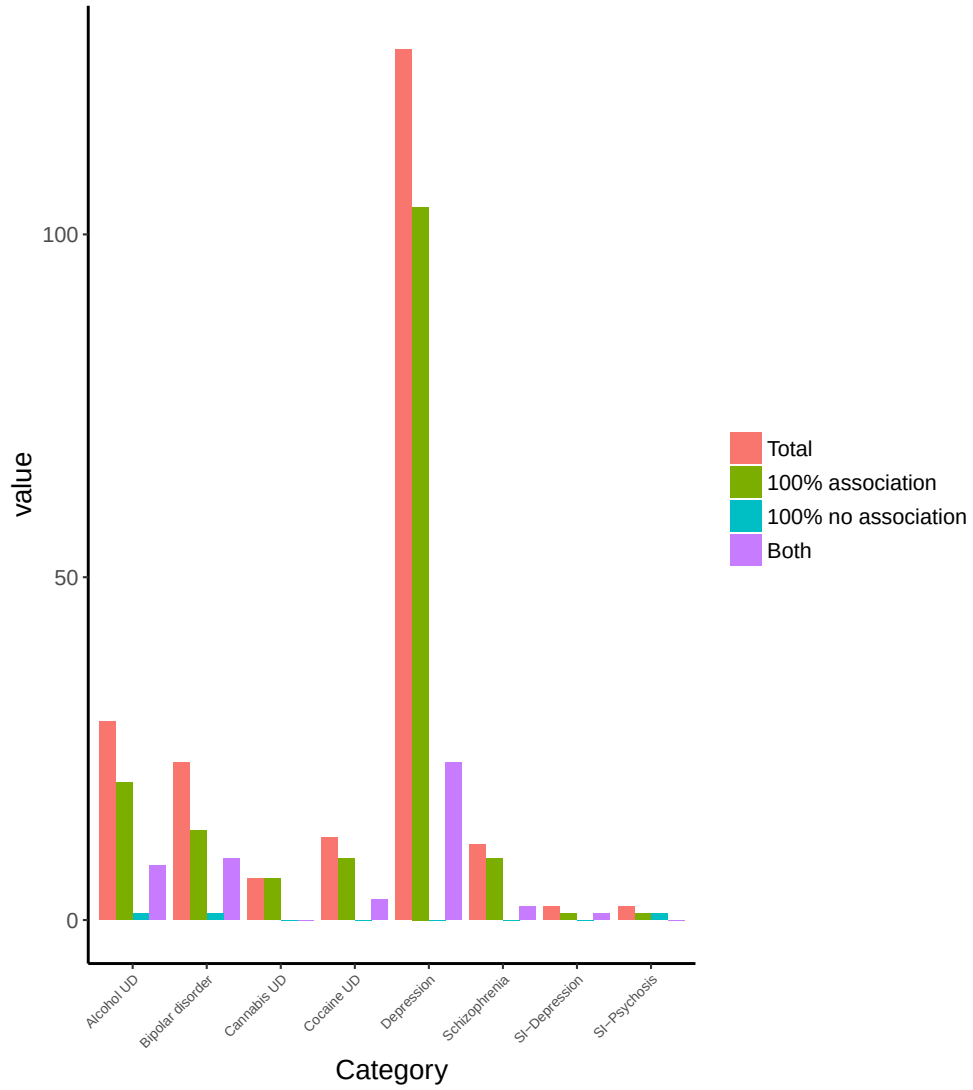


In our example, depression has 19 associated genes, and the majority of them are also associated to other psychiatric disorders, like bipolar disorder (18 genes) and alcohol UD (16 genes).

In PsyGeNET it is important to keep track of both "positive" and the "negative" findings, and let the user make their own judgements based on the available evidence. Thus, for each GDA and each supporting publication,the association type information is provided. According to the evidence, there are two types: "Association" and "No Association" (e.g. the "negative findings").

`psygenet2r` package allows to visualize this information in a barplot. It can be done by applying the `geneAttrPlot` function and setting the `type` argument to `"index"`. As a result a barplot showing, for each psychiatric disorder, how many gene-disease association are in total (red bar), how many of them are 100% association (green bar), 100% no association (blue bar) and how many GDAs are supported by both association types (purple bar).

```
geneAttrPlot( m1, type = "index" )
```



In our example, the barplot shows that depression is the psychiatric disorder with more gene-disease associations (127 GDAs), and there is no negative evidence for any of the associations.

Enrichment analysis for a list of genes

The R package `psygenet2r` allows to perform an enrichment analysis on a list of genes with PsyGeNET diseases. It is done by using the function `enrichedPD`. In order to illustrate this function, the previous list of 20 genes associated to depression [**?**] will be used.

```
tbl <- enrichedPD( genesOfInterest, database = "ALL")
tbl
```

```
##                                                MPD      p.value
## 1                              Alcohol use disorders 1.873960e-08
## 2             Bipolar disorders and related disorders 2.559998e-08
## 3                               Depressive disorders 6.286367e-09
## 4 Schizophrenia spectrum and other psychotic disorders 8.854607e-01
## 5                              Cocaine use disorders 3.697393e-07
## 6               Substance induced depressive disorder 1.361419e-02
```

```
## 7                                Cannabis use disorders 3.406556e-03
## 8                            Substance induced psychosis 1.737779e-02
```

The result is a table with a p-value of the enrichment of the given list of genes for each psychiatric disorder in PsyGeNET. As we can see, if we put a p-value cut-off of 0.01, these genes are enriched in 5 of the 8 psychiatric disorders, being alcohol UD (p-val 9e-10) and depression (p-val 6e-9), the ones with the lowest p-value.

Enrichment analysis based on anatomical terms (TopAnat) for a list of genes

`psygenet2r` package allows to perform gene set enrichment test based on expression of genes in anatomical structures, importing data from the Bgee database [**?**] and importing functions from BgeeDB R package [**?**].

It is done by using the function `topAnatEnrichment`. This function perform the enrichment analysis using a list of genes (NCBI gene identifier or official Gene Symbol from HUGO). It contains also other arguments like the `dataType` (rna_seq or affymetrix), `statistics`, that by default is `fisher` and the `cutOff` argument.

```
tpAnat <- topAnatEnrichment( genesOfInterest, cutOff = 1 )
```

```
head( tpAnat )
```

```
##                        organId                  organName annotated
## UBERON:0000013 UBERON:0000013 sympathetic nervous system       338
## UBERON:0000407 UBERON:0000407          sympathetic trunk       338
## UBERON:0002410 UBERON:0002410   autonomic nervous system       338
## UBERON:0001769 UBERON:0001769                      iris       182
## UBERON:0011892 UBERON:0011892              anterior uvea       187
## UBERON:0001697 UBERON:0001697             orbital region       117
##                significant expected foldEnrichment    pValue FDR
## UBERON:0000013           5     3.75       1.333333 0.3141495   1
## UBERON:0000407           5     3.75       1.333333 0.3141495   1
## UBERON:0002410           5     3.75       1.333333 0.3141495   1
## UBERON:0001769           3     2.02       1.485149 0.3278057   1
## UBERON:0011892           3     2.07       1.449275 0.3438253   1
## UBERON:0001697           2     1.30       1.538462 0.3763757   1
```

The result is a data frame that contains the anatomical structures. Results are sorted by p-value, and FDR values are calculated.

Sentences that report a GDA

`psygenet2r` package also allows to extract the sentences that report a gene-disease asssociation from the supporting publications. It is done by using two different functions, `psygenetGeneSentences` and `extractSentences`. `psygenetGeneSentences` needs as input a gene list and a database to query in. The output of this function is a `DataGeNET.Psy` object. This object is passed to the `extractSentences` function, that also needs the disorder of interest.

```
genesOfInterest
```

```
##  [1] "COMT"   "CLOCK"  "DRD3"   "GNB3"   "HTR1A"  "MAOA"   "HTR2A"
##  [8] "HTR2C"  "HTR6"   "SLC6A4" "ACE"    "BDNF"   "DRD4"   "HTR1B"
## [15] "HTR2B"  "MTHFR"  "SLC6A3" "TPH1"   "SLC6A2" "GABRA3"

sss <- psygenetGeneSentences( geneList = genesOfInterest,
                              database = "ALL")

## Warning in psygenetGeneSentences(geneList = genesOfInterest, database = "ALL"):
One or more of the given genes is not in PsyGeNET ( 'ALL' ). Genes:  HTR2B

sss

## Object of class 'DataGeNET.Psy'
##   . Type:          gene
##   . Database:      ALL
##   . Term:          COMT ... GABRA3
##   . N. Results:    743
##   . U. Diseases:   42
##   . U. Genes:      19

geneSentences <- extractSentences( object = sss,
                                   disorder = "alcohol abuse")
dim( geneSentences )

## [1] 14  8
```

The result is a data frame that contains the gene symbol, gene identifier, disease name, original db, the pmid, the annotation type and the sentence.

Using diseases as a query

`psygenet2r` package allows to explore PsyGeNET information searching a specifc disease or a list of diseases. As in the case of genes, it retrieves the information that is available in PsyGeNET and allows to visualize the results in several ways.

Using as a query a single disease

In order to look for a single disease into PsyGeNET, `psygenet2r` has the `psygenetDisease` function. This function allows you to obtain PsyGeNET's information using both disease id or disease name, and the database as input (by default is `ALL`).
If the user does not know the disease identifier, the `getUMLS` function can be used to obtain disease names and UMLS CUIs from a string query. Providing as input the term and source of interest, `getUMLs` function retrieves all the PsyGeNET concepts that contain it. As an example it is shown the query results for `depressive` term in `ALL` databases.

```
getUMLs( "depressive", database = "ALL" )

##                                    DiseaseName
## 12                    major depressive disorder
## 17                          depressive disorder
## 36                          depressive symptoms
## 71               single major depressive episode
## 230         recurrent major depressive episodes
## 304                drug-induced depressive state
## 392    depressive disorder, treatment-resistant
```

```
## 935             depressive episode, unspecified
## 1688      mixed anxiety and depressive disorder
## 2034 recurrent depressive disorder, unspecified
## 2089                         depressive syndrome
##                        PsychiatricDisorder          umls
## 12                     Depressive disorders umls:C1269683
## 17                     Depressive disorders umls:C0011581
## 36                     Depressive disorders umls:C0086132
## 71                     Depressive disorders umls:C0024517
## 230                    Depressive disorders umls:C0154409
## 304  Substance induced depressive disorder umls:C0338715
## 392                    Depressive disorders umls:C2063866
## 935                    Depressive disorders umls:C0349217
## 1688                   Depressive disorders umls:C0338908
## 2034                   Depressive disorders umls:C0349218
## 2089                   Depressive disorders umls:C0086133
```

As an example, the disease *major affective disorder 2*, whose disease id is *umls:C1839839* is queried using `psygenetDisease` function, and using both, disease name and disease id. The argument `score` is filled with a vector which first position can be `'<'` or `'>'` to indicate if the threshold is read as lower or upper. The second argument is the threshold in itself which will always be included. This argument is also present in `psygenetGene`.

The score is the PsyGeNET evidence index (EI), which ranges from 0 to 1 (EI=1, when all the evidences reviewed by the experts support the existence of an association between the gene and the disease; $1 > \text{EI} > 0$, when there is contradictory evidence for the GDA and EI=0 when all the evidences reviewed by the experts report that there is no association between the gene and the disease).

For this example database `"ALL"` and score `> 0.5` is selected:

```
d1 <- psygenetDisease( disease  = "umls:C1839839",
                       database = "ALL",
                       score    = c('>', 0.5 ) )
d1
```

```
## Object of class 'DataGeNET.Psy'
##   . Type:         disease
##   . Database:     ALL
##   . Term:         umls:C1839839
##   . N. Results:   17
##   . U. Genes:     17
##   . U. Diseases:  1
```

```
d2 <- psygenetDisease( disease = "major affective disorder 2",
                       database = "ALL",
                       score    = c('>', 0.5 ) )
d2
```

```
## Object of class 'DataGeNET.Psy'
##   . Type:         disease
##   . Database:     ALL
##   . Term:         major affective disorder 2
##   . N. Results:   17
```

```
##  . U. Genes:     17
##  . U. Diseases:  1
```

Both cases result in an `DataGeNET.Psy` object, that contains the same information as in the gene query search:

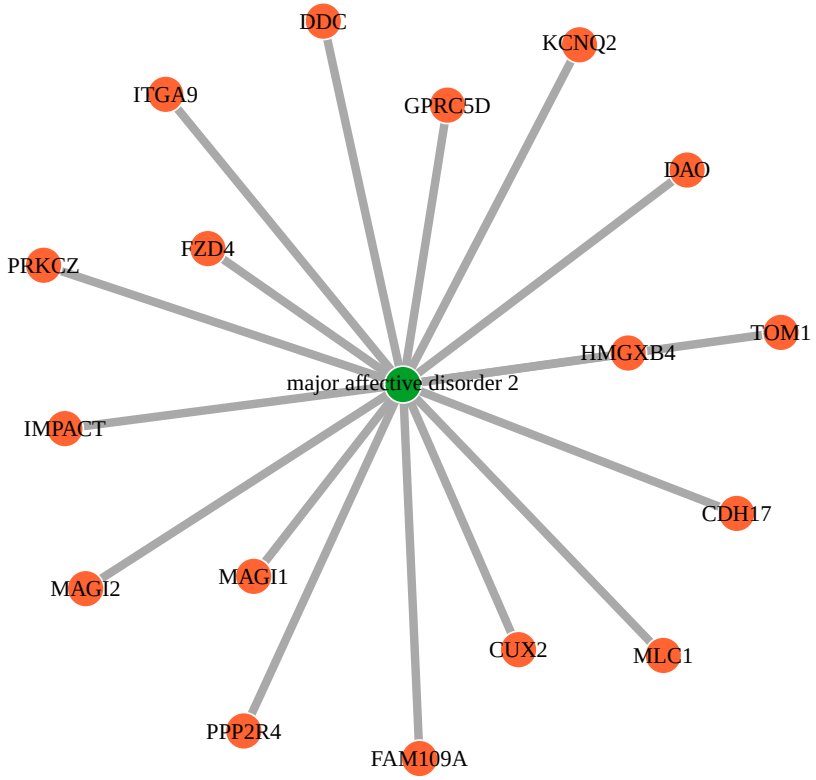
```
class( d1 )

## [1] "DataGeNET.Psy"
## attr(,"package")
## [1] "psygenet2r"

class( d2 )

## [1] "DataGeNET.Psy"
## attr(,"package")
## [1] "psygenet2r"
```

Plotting results of a Single Disease Query

`psygenet2r` package offers several options to visualize the results from PysGeNET given a disease: a network showing the genes related to the disease of interest and a barplot showing how many publications report each one of the gene-disease associations.

By default, `psygenet2r` shows the GDAs network when ploting a `DataGeNET.Psy` object with a disease-query. The result is a network where, orange nodes are genes and the central and green node is the disease of interest.

```
plot ( d1 )
```



Using a list of diseases as a query

In the same way, `psygenet2r` allows to query PsyGeNET given a set of diseases of interest. The same function, `psygenetDisease`, accepts a vector of disease-names or disease-ids (umls code).

To illustrate this functionality, two disorders has been selected: chronic schizophrenia and alcohol use disorder. The vector of diseases can be defined for example, as follows:

```
diseasesOfInterest <- c( "chronic schizophrenia","alcohol use disorder" )
```

```
tt <- psygenetDisease( disease  = diseasesOfInterest,
                       database = "ALL" )
tt

## Object of class 'DataGeNET.Psy'
##   . Type:          disease
##   . Database:      ALL
##   . Term:          chronic schizophrenia ... alcohol use disorder
```

```
##   . N. Results:   25
##   . U. Genes:     25
##   . U. Diseases:  2
```

```
dm <- psygenetDisease( disease  = c( "umls:C0221765", "umls:C0001956" ),
                       database = "ALL" )
dm
```

```
## Object of class 'DataGeNET.Psy'
##   . Type:         disease
##   . Database:     ALL
##   . Term:         umls:C0221765 ... umls:C0001956
##   . N. Results:   25
##   . U. Genes:     25
##   . U. Diseases:  2
```

```
tm <- psygenetDisease( disease  = c( "chronic schizophrenia","umls:C0001956" ),
                       database = "ALL" )
tm
```

```
## Object of class 'DataGeNET.Psy'
##   . Type:         disease
##   . Database:     ALL
##   . Term:         chronic schizophrenia ... umls:C0001956
##   . N. Results:   25
##   . U. Genes:     25
##   . U. Diseases:  2
```

Three cases result in an `DataGeNET.Psy` object:

```
class( tt )
```

```
## [1] "DataGeNET.Psy"
## attr(,"package")
## [1] "psygenet2r"
```

```
class( dm )
```

```
## [1] "DataGeNET.Psy"
## attr(,"package")
## [1] "psygenet2r"
```

```
class( tm )
```

```
## [1] "DataGeNET.Psy"
## attr(,"package")
## [1] "psygenet2r"
```

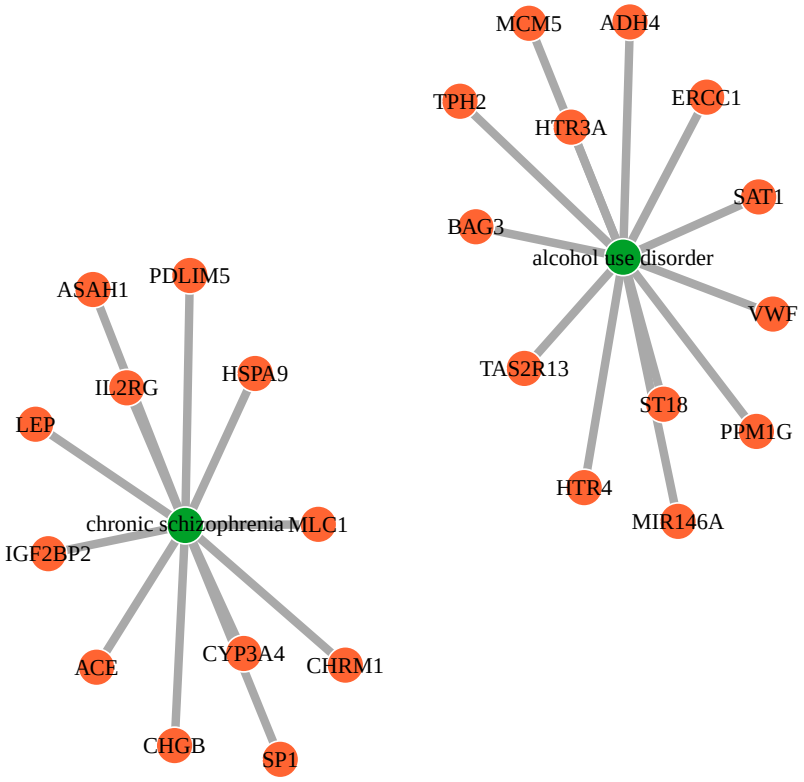
This type of object contains all the information about the different genes associated with the diseases of interest retrieved from PsyGeNET. By inspecting the `DataGeNET.Psy` object we can see that, according to PsyGeNET and querying in ALL databases, the 2 disorders of interest are associated to 25 different genes in 25 different associations.

Ploting results: Multiple Diseases

`psygenet2r` provides a network graphic and a heatmap to visualize the results of search with multiple input items.

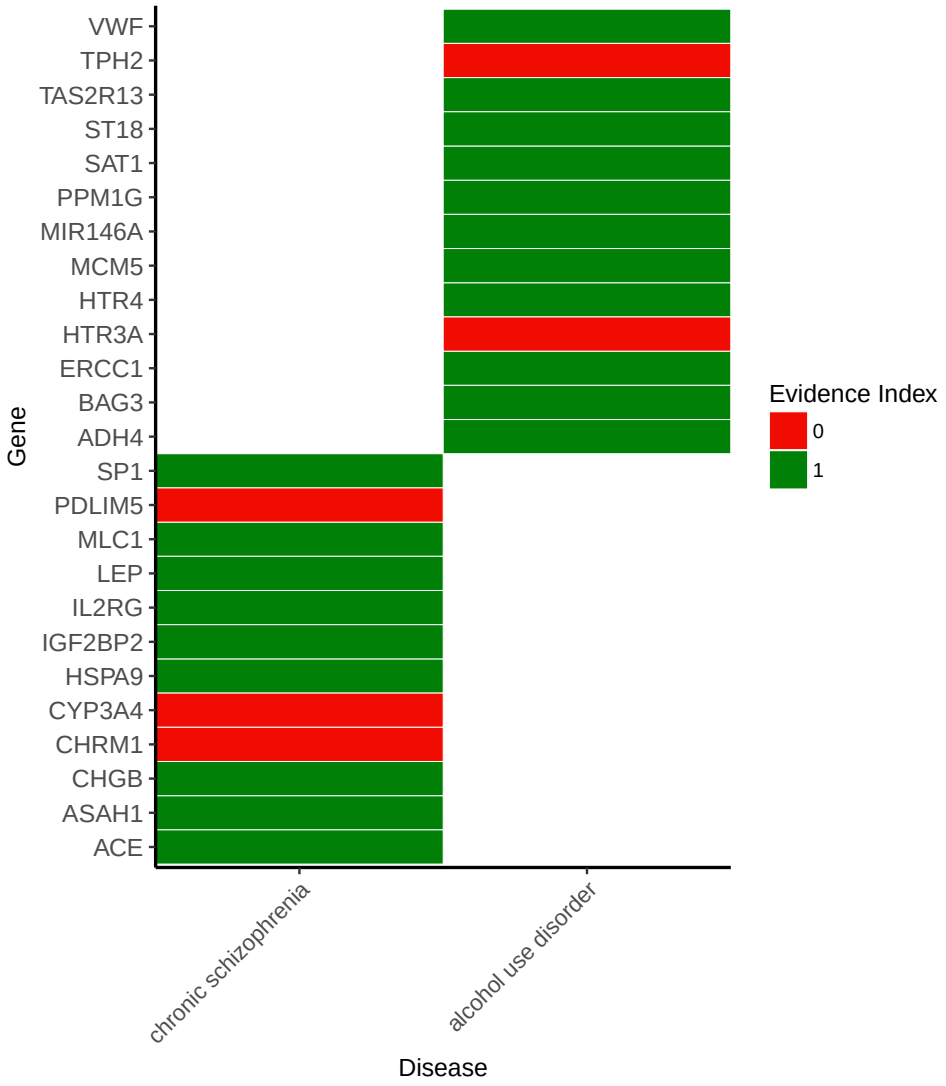
As for single disease, GDAs network is the default option in `psygenet2r`. In the resulting network chart, the green nodes represent diseases and the oranges nodes represent genes.

```
plot( tm )
```



Another possible option is visualize it in a heatmap. The argument `type` can be set to `"heatmap"`.

```
plot( tm, type = "heatmap" )
```

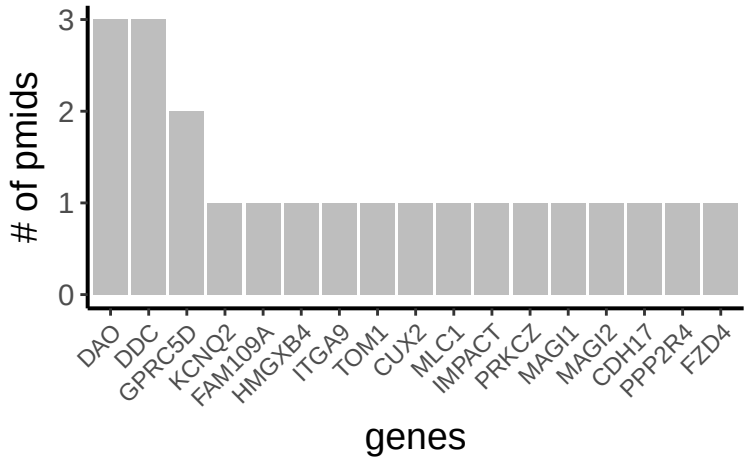


The result is a heatmap where the genes are located at X axis, and disordes appear at Y axis. The red rank color is related to the PsyGeNET score, being the darkest one the association with the highest score. The score is the PsyGeNET evidence index (EI), which ranges from 0 to 1 (EI=1, when all the evidences reviewed by the experts support the existence of an association between the gene and the disease; $1 > \text{EI} > 0$, when there is contradictory evidence for the GDA and EI=0 when all the evidences reviewed by the experts report that there is no association between the gene and the disease).

Barplot according to number of publications that support the GDA

`psygenet2r` package allows to see how many publications support each gene-disease association. This can be visualized in a barplot by determining the gene or disease id in the `name` argument and setting `type` argument to `"barplot"`.

```
plot( d1, name = "major affective disorder 2", type = "barplot" )
```



As a result, a barplot is obtained. The X axis contains the genes related to the disease of interest, sorted by the number of pubmed ids in which we can find the gene-disease association. Alternatively, the results can be visualized for the diseases.

```
plot( t1, name = "NPY", type = "barplot" )

## Error in plot(t1, name = "NPY", type = "barplot"):  object 't1' not found
```

Analyze the association between two diseases based on shared genes

We can study the association between two diseases from the point of view of shared genetic contribution. More precisely, we can estimate the degree of association of two diseases by means of the number of genes that are shared between the two diseases, over the total number of disease genes. Similarity measures such as the Jaccard Index can be used to estimate disease similarity.

The strategy to calculate the Jaccard Index and its p-value as follows:

1. Calculate the Jaccard Index between the pair of diseases. Let's call it *rJI*.
2. Randomly select a set of genes from DisGeNET for each one of the input diseases (or set of genes) and compute their Jaccard Index (iJI).
3. Calculate the p-value by dividing the count of the *iJI* higher than the real *rJI* by the number of attempts we performed the step B plus one (`nboot + 1`).

`JaccardIndexPsy` object

`JaccardIndexPsy` object is obtained when `jaccardEstimation` function is applied. The results of this object can be visualized by applying the `plot` function.

`JaccardIndexPsy` object contains all the information about the Jaccard estimation characteristics and the results retrieved from PsyGeNET. This object contains a summary of the search, such as the number of iterations used to compute the pvalue associted to the calculated Jaccard Index (`#Boot`), the type of search (`Type`) and the number results obtained (`#Results`).

```
ji1

## Object of class 'JaccardIndexPsy'
##   . #Boot:               100
##   . Type:                gene-list - disease
##   . #Results:            117

class( ji1 )

## [1] "JaccardIndexPsy"
## attr(,"package")
## [1] "psygenet2r"
```

This object comes with a series of functions to allow users to interact with the information retrieved from PsyGeNET. These functions are `extract` and `plot`. The function `extract` returns a formatted `data.frame` with the complete set of information downloaded from PsyGeNET. The `plot` function allows the visualization of the results in a variety of ways such as barplots or heatmaps.

Using the Jaccard Index

The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistic used for comparing the similarity of two sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{\mid A \cap B \mid}{\mid A \cup B \mid}$$

`psygenet2r` comes with functions to compute the Jaccard Index as an estimation of the similarity of two diseases based on shared genes, given information retrieved from PsyGeNET. The user can compute the Jaccard Index using the function `jaccardEstimation`. This function accepts multiple inputs:

1. Using a list of genes of interest the function will compute the Jaccard Index between the set of genes and all the diseases in PsyGeNET.

```
genes_interest <- c("SLC6A4", "DRD2", "HTR1B", "PLP1", "TH", "DRD3")
ji1 <- jaccardEstimation(genes_interest, database = "ALL")

## Warning in singleInput.genes(diseases$diseases$`gene-list`$genes, psy, universe,
:  Jaccard Index for all diseases in PsyGeNET will be calculated.
```

2. Using a list of genes of interest and a list of diseases of interest, the function computes the Jaccard Index between the set of genes and each disease:

```
disease_interest <-
  c("delirium", "bipolar i disorder", "severe depression", "cocaine dependence")
ji2 <- jaccardEstimation(genes_interest, disease_interest, database = "ALL")
```

3. With a list of diseases of interest, the function will calculate the Jaccard Index between themselves:

```
ji3 <- jaccardEstimation(disease_interest, database = "ALL")
```

To determine if the association between two diseases as estimated by the Jaccard Index was statistically significant, we apply a bootstrap procedure to estimate the likelihood of obtaining a Jaccard Index greater than the one obtained for the association between the diseases by chance. In other words, we sample at random gene sets of size n and p (n, p is the number of genes associated to disease 1 and 2, respectively) from a population of human disease genes obtained from DisGeNET [**?**]. These random gene sets (n and p) are then used to compute the Jaccard Index for diseases 1 and 2. This procedure is repeated by default 100 times. The number of iterations used to compute the pvalue associted to the calculated Jaccard Index can be changed by the user with the `nboot` argument. Then, we calculated the number of times we obtained a Jaccard Index for the random gene sets larger than the observed value of the Jaccard Index.

The raw results are stored in `JaccardIndexPsy` and can be obtained using the function `extract`. For example:

```
head(extract(ji1))
```

```
##   Disease1 Disease2 NGenes1 NGenes2 JaccardIndex pval
## 1    genes C0001973       6     279   0.01754386    0
## 2    genes C0005586       6     502   0.00984252    0
## 3    genes C0011581       6     276   0.01773050    0
## 4    genes C1269683       6     254   0.01923077    0
## 5    genes C0525045       6     185   0.02617801    0
## 6    genes C0011570       6     260   0.01879699    0
```

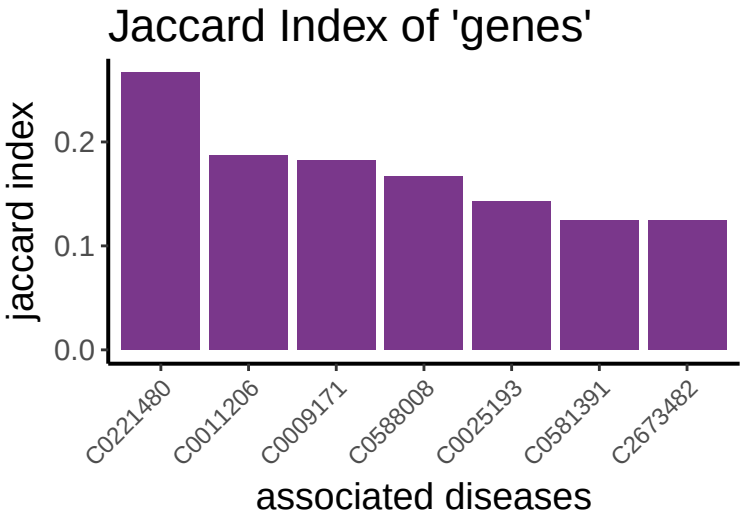
```
tail(extract(ji1))
```

```
##    Disease1 Disease2 NGenes1 NGenes2 JaccardIndex       pval
## 78    genes C0556385       6       3  0.000000000 0.01980198
## 15    genes C0001969       6      33  0.025641026 0.02970297
## 35    genes C0086132       6      30  0.000000000 0.02970297
## 37    genes C0349204       6      84  0.000000000 0.04950495
## 38    genes C0033975       6     109  0.000000000 0.10891089
## 36    genes C0036341       6     861  0.001153403 0.40594059
```

Plotting results: Jaccard Index

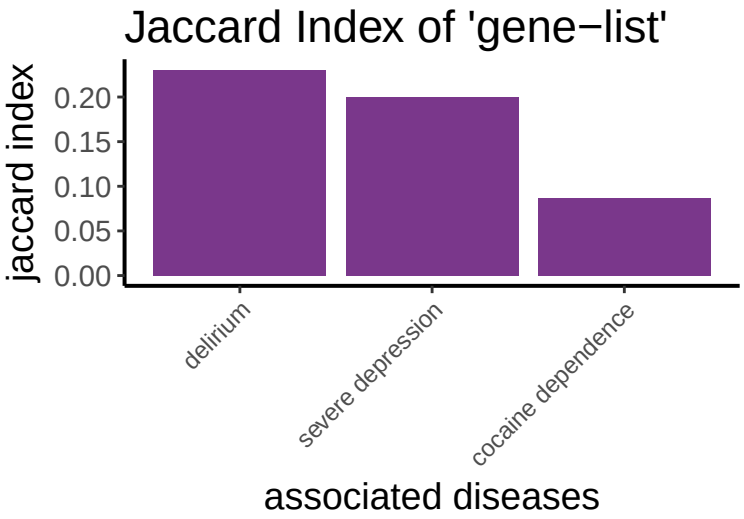
The plot of the result of a `jaccardEstimation` using a singe set of genes corresponds to a bar-plot of the Jaccard Index with each disease:

```
plot(ji1, cutOff = 0.1)
```



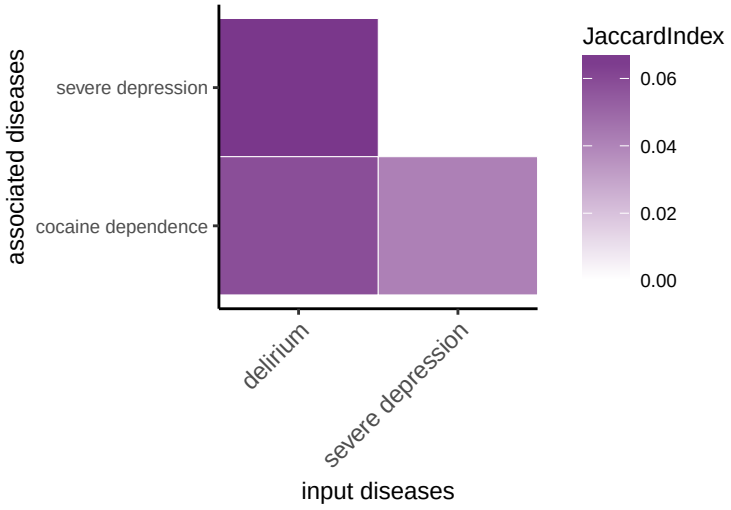
The previous bar-plot shows the Jaccard Index greater than 0.1 obtained from testing each diseases in PsyGeNET. When given a set of genes and a set of diseases, the resulting plot is equivalent:

```
plot(ji2)
```



The plot resulting from more than one disease is a heat-map with the given disease as X axis and all the diseases that share genes with them placed as Y axis. The intensity of the color represents the value of the Jaccard Index between, being the darker one the major Jaccard Index.

```
plot(ji3)
```



Warnings

`heatmap` type argument do not allow queries for single gene:

```
> plot( t1, type = "heatmap" )
==> Error: For this type of chart, a multiple gene query created with
 'psygenetGene' is required.
```

Summary of visualization options

Table 3: Visualization options

Input Object	**psygenet2r function**	**Argument Type**	**Output Generated**
DataGeNET.Psy	geneAttrPlot	category	genes asociated with each DC
		gene	disease concepts and DC associated with each gene
		pie	pie chart number of genes per DC
		index	barplot showing the type of association
	plot	individual disease	GDA network (default type)
		disease class	GDCAs network
		heatmapGenes	EI heatmap GDCAs heatmap (gene)
		heatmap	EI heatmap for GDA (disease)
		Bar-plot	number of publications supporting GDA
JaccardIndexPsy	plot	/	JI Bar-plot / heatmap

References

[1] Alba Gutierrez-Sacristan; Solene Grosdidier; Olga Valverde; Marta Torrens; Alex Bravo; Janet Pinero; Ferran Sanz; Laura I. Furlong. **PsyGeNET: a knowledge platform on psychiatric disorders and their genes** Bioinformatics 2015 doi: 10.1093/bioinformatics/btv301

[2] Sullivan, Patrick F; Daly, Mark J; O'Donovan, Michael. **Genetic architectures of psychiatric disorders: the emerging picture and its implications** Nature reviews. Genetics (2012) vol. 13 (8) p. 537-51

[3] Piñero, Janet and Bravo, Àlex and Queralt-Rosinach, Núria and Gutiérrez-Sacristán, Alba and Deu-Pons, Jordi and Centeno, Emilio and García-García, Javier and Sanz, Ferran and Furlong, Laura I **DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants** Nucleic Acids Research (2016)

[4] Bravo, À.; Piñero, J.; Queralt, N.; Rautschka, M.; Furlong, L.I. **Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research** BMC Bioinformatics 2015, 16:55 doi:10.1186/s12859-015-0472-9

[5] Flint, Jonathan; Kendler, Kenneth S. **The genetics of major depression** Neuron (2014) Vol. 81 (3) p. 484-503

[6] Jens Treutlein, Sven Cichon et al. **Genome-wide association study of alcohol dependence**. Archives of general psychiatry (2009) vol.66(7) p.773 doi: 10.1001/archgenpsychiatry.2009.83.

[7] Komljenovic A and Roux J **BgeeDB: an R package for annotation and gene expression data retrieval from Bgee database** (2016)

[8] Bastian F **Bgee: Integrating and Comparing Heterogeneous Transcriptome Data Among Species** Data Integration Life Sci. Lecture Notes in Computer Science (2008), pp. 124-31.