

PANTHER.db: An annotation package to access the PANTHER Classification System

Julius Müller

October 3, 2014

1 Introduction to *PANTHER.db*

The *PANTHER.db* package provides a `select` interface to the compiled PANTHER ontology residing within a SQLite database.

```
library(PANTHER.db)

## Loading required package: AnnotationDbi
## Loading required package: stats4
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:parallel':
##
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##   clusterExport, clusterMap, parApply, parCapply, parLapply,
##   parLapplyLB, parRapply, parSapply, parSapplyLB
##
## The following object is masked from 'package:stats':
##
##   xtabs
##
## The following objects are masked from 'package:base':
##
##   Filter, Find, Map, Position, Reduce, anyDuplicated, append,
##   as.data.frame, as.vector, cbind, colnames, do.call,
##   duplicated, eval, evalq, get, intersect, is.unsorted, lapply,
##   mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##   pmin.int, rank, rbind, rep.int, rownames, sapply, setdiff,
##   sort, table, tapply, union, unique, unlist, unsplit
```

```
##
## Loading required package: Biobase
## Welcome to Bioconductor
##
## Vignettes contain introductory material; view with
## 'browseVignettes()'. To cite Bioconductor, see
## 'citation("Biobase)", and for packages 'citation("pkgname)".
##
## Loading required package: GenomeInfoDb
## Loading required package: S4Vectors
## Loading required package: IRanges
##
## Attaching package: 'AnnotationDbi'
##
## The following object is masked from 'package:GenomeInfoDb':
##
## species
##
## Loading required package: RSQLite
## Loading required package: DBI
## PANTHER.db version 1.0.1
```

If you already know about the select interface, you can immediately learn about the various methods for this object by just looking at the help page.

```
help("PANTHER.db")
```

When you load the *PANTHER.db* package, it creates a *PANTHER.db* object. If you look at the object you will see some helpful information about it.

```
PANTHER.db
## PANTHER.db object:
## | SPECIES: ANOPHELES|ARABIDOPSIS|BOVINE|CANINE|CHICKEN|CHIMP|COELICOLOR|ECOLI|FLY|HUMAN|MALARIA|
## | PANTHERVERSION: 9.0
## | PANTHERSOURCEURL: ftp.pantherdb.org
## | PANTHERSOURCEDATE: 2014-Jun27
## | package: AnnotationDbi
## | Db type: PANTHER
## | DBSCHEMA: PANTHER_DB
## | DBSCHEMAVERSION: 2.1
## | UNIPROT to ENTREZ mapping: 2014-Jun27
```

By default, you can see that the *PANTHER.db* object is set to retrieve records from various species. The choice of the species is the intersection of species supported by *PANTHER* and the

core annotation packages in bioconductor. Methods are provided to restrict all queries to a specific kind of species. In order to change it, you first need to look up the appropriate species identifier for the species that you are interested in. The PANTHER gene ontology is based on the Uniprot reference proteome set. In order to display the choices, we have provided the helper function `availablePantherSpecies` which will list all the supported species along with their Uniprot species name and taxonomy ids:

```
availablePantherSpecies(PANTHER.db)

##      PANTHER Species
## 1          ANOPHELES
## 2          ARABIDOPSIS
## 3            BOVINE
## 4            CANINE
## 5            CHICKEN
## 6            CHIMP
## 7          COELICOLOR
## 8            ECOLI
## 9            FLY
## 10           HUMAN
## 11           MALARIA
## 12           MOUSE
## 13           PIG
## 14           RAT
## 15           RHESUS
## 16           WORM
## 17           XENOPUS
## 18           YEAST
## 19          ZEBRAFISH
##
##                               UNIPROT Species
## 1                Anopheles gambiae
## 2                Arabidopsis thaliana
## 3                   Bos taurus
## 4                Canis familiaris
## 5                   Gallus gallus
## 6                Pan troglodytes
## 7 Streptomyces coelicolor (strain ATCC BAA-471 / A3(2) / M145)
## 8                Escherichia coli (strain K12)
## 9                Drosophila melanogaster
## 10                 Homo sapiens
## 11 Plasmodium falciparum (isolate 3D7)
## 12                 Mus musculus
## 13                 Sus scrofa
## 14                Rattus norvegicus
## 15                 Macaca mulatta
```

```

## 16                               Caenorhabditis elegans
## 17                               Xenopus tropicalis
## 18      Saccharomyces cerevisiae (strain ATCC 204508 / S288c)
## 19                               Danio rerio
##      UNIPROT Taxon ID
## 1              7165
## 2              3702
## 3              9913
## 4              9615
## 5              9031
## 6              9598
## 7             100226
## 8              83333
## 9              7227
## 10             9606
## 11             36329
## 12             10090
## 13             9823
## 14             10116
## 15             9544
## 16             6239
## 17             8364
## 18             559292
## 19             7955

```

Once you have learned the PANTHER species name for the species of interest, you can then change the species for the PANTHER.db object:

```

species(PANTHER.db) <- "HUMAN"
PANTHER.db

## PANTHER.db object:
## | SPECIES: HUMAN
## | PANTHERVERSION: 9.0
## | PANTHERSOURCEURL: ftp.pantherdb.org
## | PANTHERSOURCEDATE: 2014-Jun27
## | package: AnnotationDbi
## | Db type: PANTHER
## | DBSCHEMA: PANTHER_DB
## | DBSCHEMAVERSION: 2.1
## | UNIPROT to ENTREZ mapping: 2014-Jun27

resetSpecies(PANTHER.db)

## ANOPHELES | ARABIDOPSIS | BOVINE | CANINE | CHICKEN | CHIMP | COELICOLOR | ECOLI | FLY | HUMAN | MALARIA | MOUSE | P

```

```
PANTHER.db
```

```
## PANTHER.db object:
## | SPECIES: ANOPHELES|ARABIDOPSIS|BOVINE|CANINE|CHICKEN|CHIMP|COELICOLOR|ECOLI|FLY|HUMAN|MALARIA|
## | PANTHERVERSION: 9.0
## | PANTHERSOURCEURL: ftp.pantherdb.org
## | PANTHERSOURCEDATE: 2014-Jun27
## | package: AnnotationDbi
## | Db type: PANTHER
## | DBSCHEMA: PANTHER_DB
## | DBSCHEMAVERSION: 2.1
## | UNIPROT to ENTREZ mapping: 2014-Jun27
```

As you can see the species is now restricted to Homo sapiens. To display all data which can be returned from a select query, the columns method can be used:

```
columns(PANTHER.db)
```

```
## [1] "FAMILY_ID"      "GOSLIM_ID"      "GOSLIM_TERM"
## [4] "UNIPROT"        "SPECIES"        "FAMILY_TERM"
## [7] "SUBFAMILY_TERM" "CLASS_ID"       "CLASS_TERM"
## [10] "PATHWAY_ID"     "PATHWAY_TERM"   "COMPONENT_ID"
## [13] "COMPONENT_TERM" "EVIDENCE"       "EVIDENCE_TYPE"
## [16] "CONFIDENCE_CODE" "ENTREZ"
```

Some of these fields can also be used as keytypes:

```
keytypes(PANTHER.db)
```

```
## [1] "FAMILY_ID"      "GOSLIM_ID"      "CLASS_ID"       "PATHWAY_ID"
## [5] "COMPONENT_ID"  "UNIPROT"        "SPECIES"        "ENTREZ"
```

It is also possible to display all possible keys of a table for any keytype. If keytype is unspecified, the FAMILY_ID will be returned.

```
go_ids<-head(keys(PANTHER.db,keytype="GOSLIM_ID"))
```

```
go_ids
```

```
## [1] "GO:0000003" "GO:0000165" "GO:0000166" "GO:0000228" "GO:0000375"
## [6] "GO:0000398"
```

Finally, you can loop up whatever combinations of columns, keytypes and keys that you need when using `select`.

```

cols <- c("FAMILY_ID", "CLASS_ID")
res <- select(PANTHER.db, keys=go_ids, columns=cols, keytype="GOSLIM_ID")
head(res)

##      GOSLIM_ID      FAMILY_ID CLASS_ID
## 1 GO:0000003      PTHR10150  PC00009
## 2 GO:0000003      PTHR10150  PC00170
## 3 GO:0000003      PTHR10150  PC00093
## 4 GO:0000003      PTHR10150  PC00171
## 5 GO:0000003 PTHR10150:SFO  PC00009
## 6 GO:0000003 PTHR10150:SFO  PC00170

```

To access the PANTHER Protein Class ontology tree structure, the method `traverseClassTree` can be used:

```

term<-"PC00209"
select(PANTHER.db, term, "CLASS_TERM", "CLASS_ID")

##      CLASS_ID      CLASS_TERM
## 1 PC00209 sodium channel

ancestors<-traverseClassTree(PANTHER.db, term, scope="ANCESTOR")
select(PANTHER.db, ancestors, "CLASS_TERM", "CLASS_ID")

##      CLASS_ID      CLASS_TERM
## 1      PC00133 ion channel
## 949 PC00227 transporter

parents<-traverseClassTree(PANTHER.db, term, scope="PARENT")
select(PANTHER.db, parents, "CLASS_TERM", "CLASS_ID")

##      CLASS_ID      CLASS_TERM
## 1 PC00133 ion channel

children<-traverseClassTree(PANTHER.db, term, scope="CHILD")
select(PANTHER.db, children, "CLASS_TERM", "CLASS_ID")

##      CLASS_ID      CLASS_TERM
## 1 PC00243 voltage-gated sodium channel

offspring<-traverseClassTree(PANTHER.db, term, scope="OFFSPRING")
select(PANTHER.db, offspring, "CLASS_TERM", "CLASS_ID")

##      CLASS_ID      CLASS_TERM
## 1 PC00243 voltage-gated sodium channel

```