PANTHER.db: An annotation package to access PANTHER GO ontologies

Julius Müller

September 18, 2013

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)
```

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: 8.1
| PANTHERSOURCEURL: ftp.pantherdb.org
| PANTHERSOURCEDATE: 2013-Sep17
| package: AnnotationDbi
| Db type: PANTHER
| DBSCHEMA: PANTHER_DB
| DBSCHEMAVERSION: 2.1
| UNIPROT to ENTREZ mapping: 2013-Sep17
```

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                                              UNIPROT Species
1        ANOPHELES                                             Anopheles gambiae
2      ARABIDOPSIS                                          Arabidopsis thaliana
3           BOVINE                                                    Bos taurus
4           CANINE                                             Canis familiaris
5          CHICKEN                                                 Gallus gallus
6            CHIMP                                               Pan troglodytes
7       COELICOLOR Streptomyces coelicolor (strain ATCC BAA-471 / A3(2) / M145)
8            ECOLI                               Escherichia coli (strain K12)
9              FLY                                       Drosophila melanogaster
10           HUMAN                                                  Homo sapiens
11         MALARIA                         Plasmodium falciparum (isolate 3D7)
12           MOUSE                                                  Mus musculus
13             RAT                                             Rattus norvegicus
14            WORM                                        Caenorhabditis elegans
15         XENOPUS                                            Xenopus tropicalis
16           YEAST       Saccharomyces cerevisiae (strain ATCC 204508 / S288c)
17       ZEBRAFISH                                                   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            10116
14             6239
15             8364
16           559292
17             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: 8.1
| PANTHERSOURCEURL: ftp.pantherdb.org
| PANTHERSOURCEDATE: 2013-Sep17
| package: AnnotationDbi
| Db type: PANTHER
| DBSCHEMA: PANTHER_DB
| DBSCHEMAVERSION: 2.1
| UNIPROT to ENTREZ mapping: 2013-Sep17
```

```
> resetSpecies(PANTHER.db)
```

```
ANOPHELES|ARABIDOPSIS|BOVINE|CANINE|CHICKEN|CHIMP|COELICOLOR|ECOLI|FLY|HUMAN|MALARIA|MOUSE|RAT|W
```

```
> PANTHER.db
```

```
PANTHER.db object:
| SPECIES: ANOPHELES|ARABIDOPSIS|BOVINE|CANINE|CHICKEN|CHIMP|COELICOLOR|ECOLI|FLY|HUMAN|MALARIA|
| PANTHERVERSION: 8.1
| PANTHERSOURCEURL: ftp.pantherdb.org
| PANTHERSOURCEDATE: 2013-Sep17
| package: AnnotationDbi
| Db type: PANTHER
| DBSCHEMA: PANTHER_DB
| DBSCHEMAVERSION: 2.1
| UNIPROT to ENTREZ mapping: 2013-Sep17
```

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"     "UNIPROT"
 [5] "SPECIES"         "FAMILY_TERM"     "SUBFAMILY_TERM"  "CLASS_ID"
 [9] "CLASS_TERM"      "PATHWAY_ID"      "PATHWAY_TERM"    "COMPONENT_ID"
[13] "COMPONENT_TERM"  "EVIDENCE"        "EVIDENCE_TYPE"   "CONFIDENCE_CODE"
[17] "ENTREZ"
```

Some of these fields can also be used as keytypes:

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

```
[1] "FAMILY_ID"    "GOSLIM_ID"    "CLASS_ID"     "PATHWAY_ID"   "COMPONENT_ID"
[6] "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:0000228" "GO:0000229" "GO:0000398"
[6] "GO:0000739"
```

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        PTHR10177  PC00140
2 GO:0000003        PTHR10177  PC00095
3 GO:0000003        PTHR10177  PC00138
4 GO:0000003 PTHR10177:SF100  PC00140
5 GO:0000003 PTHR10177:SF100  PC00095
6 GO:0000003 PTHR10177:SF100  PC00138
```

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
1491  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
```