How to use the MeSH.db Package

Koki Tsuyuzaki[1], Itoshi Nikaido[2,3] and Gota Morota[4].

January 7, 2013

[1]Department of Medical and Life Science, Tokyo University of Science.
[2]Functional Genomics Unit, RIKEN Center for Developmental Biology.
[3]Laboratory for Systems Biology, RIKEN Center for Developmental Biology.
[4]Department of Dairy Science, University of Wisconsin-Madison.

`k.t.the-answer@hotmail.co.jp`

Contents

1 Introduction

This document provides the way to use *MeSH.db* package. MeSH (Medical Subject Headings) is the NLM controlled vocabulary used to manually index articles for MEDLINE/Pubmed (Nelson and et al., 2004). The amount of vocabulary in MeSH is about twice as large as that of GO (Gene Ontology)(Ashburner and et al., 2000) and its category is also wider. Therefore MeSH is expected to be much detailed and exhaustive annotation tool. Some softwares or databases using MeSH are now proposed (Nakazato and et al., 2007, 2009; Saurin and et al., 2010; Sartor and et al., 2012).

     MeSH.db is a free R package for handling MeSH in R. Its data are retrieved from MeSH ftp site (`http://www.nlm.nih.gov/mesh/filelist.html`). MeSH in 2013 has 19 hierarchy and MeSH.db provides 16 of them, which are actually assigned to some MeSH Terms. Each category is expressed to single capital alphabet defined in MeSH as Abbreviations.

Abbreviation	Category
A	Anatomy
B	Organisms
C	Diseases
D	Chemicals and Drugs
E	Analytical, Diagnostic and Therapeutic Techniques and Equipment
F	Psychiatry and Psychology
G	Phenomena and Processes
H	Disciplines and Occupations
I	Anthropology, Education, Sociology and Social Phenomena
J	Technology and Food and Beverages
K	Humanities
L	Information Science
M	Persons
N	Health Care
V	Publication Type
Z	Geographical Locations

MeSH Term also have hierarchy structure like GO. *MeSH.db* provides its Ancestor-Offspring Relationships (AOR) and Parent-Child Relationships (PCR) as corresponding table. Data of PCR and AOR are also used for calculating the conditional probability in enrichment analysis (*meshr* package).

2   Getting started

To load the *MeSH.db* package, just type library(MeSH.db). 5 methods and 36 data are provided by *MeSH.db*.

3   Methods

Following 5 methods are provided by *MeSH.db*.

MeSH	Function for retrieval of the summary of all object in MeSH.db
MeSH_dbconn	Function for retrieval of the connection of .sqlite database
MeSH_dbfile	Function for retrieval of the directory of .sqlite file
MeSH_dbschema	Function for retrieval of the schema of .sqlite database
MeSH_dbInfo	Function for retrieval of the information of .sqlite database

4 Data

Following 36 data are provided by *MeSH.db*.

MeSHMAPCOUNTS	The number of row of all data
MeSHTERM	MeSH Term
MeSHSYNONYM	The synonym of MeSH Term
MeSHQUALIFIER	Substantial Information of MeSH Term
MeSHAAOR	Ancestor-Offspring Relationships in A category
MeSHBAOR	Ancestor-Offspring Relationships in B category
MeSHCAOR	Ancestor-Offspring Relationships in C category
MeSHDAOR	Ancestor-Offspring Relationships in D category
MeSHEAOR	Ancestor-Offspring Relationships in E category
MeSHFAOR	Ancestor-Offspring Relationships in F category
MeSHGAOR	Ancestor-Offspring Relationships in G category
MeSHHAOR	Ancestor-Offspring Relationships in H category
MeSHIAOR	Ancestor-Offspring Relationships in I category
MeSHJAOR	Ancestor-Offspring Relationships in J category
MeSHKAOR	Ancestor-Offspring Relationships in K category
MeSHLAOR	Ancestor-Offspring Relationships in L category
MeSHMAOR	Ancestor-Offspring Relationships in M category
MeSHNAOR	Ancestor-Offspring Relationships in N category
MeSHVAOR	Ancestor-Offspring Relationships in V category
MeSHZAOR	Ancestor-Offspring Relationships in Z category
MeSHAPCR	Parent-Child Relationships in A category
MeSHBPCR	Parent-Child Relationships in B category
MeSHCPCR	Parent-Child Relationships in C category
MeSHDPCR	Parent-Child Relationships in D category
MeSHEPCR	Parent-Child Relationships in E category
MeSHFPCR	Parent-Child Relationships in F category
MeSHGPCR	Parent-Child Relationships in G category
MeSHHPCR	Parent-Child Relationships in H category
MeSHIPCR	Parent-Child Relationships in I category
MeSHJPCR	Parent-Child Relationships in J category
MeSHKPCR	Parent-Child Relationships in K category
MeSHLPCR	Parent-Child Relationships in L category
MeSHMPCR	Parent-Child Relationships in M category

MeSHNPCR	Parent-Child Relationships in N category
MeSHVPCR	Parent-Child Relationships in V category
MeSHZPCR	Parent-Child Relationships in Z category

In *MeSH.db*, all data are extracted by 4 functions defined by *AnnotationForge*; **keytypes**, **cols**, **keys** and **select**. keys function has 1 optional parameter keytype and select function also has 3 optional parameter keys, cols and keytype. cols is the columns which you can retrieved by select and keytype is the columns which you can specify as the option in keys and select functions.

Object Name	cols	keytype
MeSHMAPCOUNTS	MAPNAME, COUNT	MAPNAME
MeSHTERM	MESHID, MESHTERM, CATEGORY	MESHID, MESHTERM, CATEGORY
MeSHSYNONYM	MESHID, MESHSYNONYM	MESHID
MeSHQUALIFIER	QUALIFIERID, SUBHEADING, MESHID	QUALIFIERID, MESHID
MeSHAAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHBAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHCAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHDAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHEAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHFAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHGAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHHAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHIAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHJAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHKAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHLAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID

MeSHMAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHNAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHVAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHZAOR	ANCESTERMESHID, OFFSPRINGMESHID	ANCESTERMESHID, OFFSPRINGMESHID
MeSHAPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHBPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHCPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHDPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHEPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHFPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHGPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHHPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHIPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHJPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHKPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHLPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHMPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHNPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHVPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID
MeSHZPCR	PARENTMESHID, CHILDMESHID	PARENTMESHID, CHILDMESHID

5 Examples

5.1 Exercises in cols, keytypes, keys and select method

MeSH.db uses cols, keytypes, keys and select functions defined by *AnnotationForge*. Here we shows how to use these functions in *MeSH.db*.

First, install and load the *MeSH.db*.

```
> library(MeSH.db)
```

ls shows all object in *MeSH.db*.

```
> ls("package:MeSH.db")
```

```
 [1] "MeSH"          "MeSH_dbconn"   "MeSH_dbfile"   "MeSH_dbInfo"
 [5] "MeSH_dbschema" "MeSHAAOR"      "MeSHAPCR"      "MeSHBAOR"
 [9] "MeSHBPCR"      "MeSHCAOR"      "MeSHCPCR"      "MeSHDAOR"
[13] "MeSHDPCR"      "MeSHEAOR"      "MeSHEPCR"      "MeSHFAOR"
[17] "MeSHFPCR"      "MeSHGAOR"      "MeSHGPCR"      "MeSHHAOR"
[21] "MeSHHPCR"      "MeSHIAOR"      "MeSHIPCR"      "MeSHJAOR"
[25] "MeSHJPCR"      "MeSHKAOR"      "MeSHKPCR"      "MeSHLAOR"
[29] "MeSHLPCR"      "MeSHMAOR"      "MeSHMAPCOUNTS" "MeSHMPCR"
[33] "MeSHNAOR"      "MeSHNPCR"      "MeSHQUALIFIER" "MeSHSYNONYM"
[37] "MeSHTERM"      "MeSHVAOR"      "MeSHVPCR"      "MeSHZAOR"
[41] "MeSHZPCR"
```

Here we use cols, keytypes, keys and select against MeSHMAPCOUNTS.

cols returns the rows which you can retrieve in MeSHMAPCOUNTS.

```
> cols(MeSHMAPCOUNTS)
```

```
[1] "MAPNAME" "COUNT"
```

keytypes returns the rows which you can use the option as keys and select functions against MeSHMAPCOUNTS.

```
> keytypes(MeSHMAPCOUNTS)
```

```
[1] "MAPNAME"
```

Here we get to know that MAPNAME is available.

keys function specifies the value of keytype.

```
> k <- keys(MeSHMAPCOUNTS, keytype="MAPNAME")
> head(k)

        MAPNAME
1       MeSHTERM
2    MeSHSYNONYM
3 MeSHQUALIFIER
4       MeSHAAOR
5       MeSHBAOR
6       MeSHCAOR
```

select method specifies the rows in particular cols having user-defined keys and retrieved data as single dataframe like SQL's SELECT statement. Now we retrieve the rows in which MAPNAME is equivalent to "MeSHTERM".

```
> select(MeSHMAPCOUNTS, keys=k[1,], cols=c("MAPNAME","COUNT"),
+         keytype="MAPNAME")

    MAPNAME COUNT
1 MeSHTERM 54843
```

By the way, here we don't have to specify keytype option against MeSHMAPCOUNTS, because MeSHMAPCOUNTS only has single col which is possible to be keytype and keytype is consequently specified.

```
> select(MeSHMAPCOUNTS, keys=k[1,], cols=c("MAPNAME","COUNT"))

    MAPNAME COUNT
1 MeSHTERM 54843
```

The same can be said of MeSHSYNONYM.

5.2 Annotation of *Cystic Fibrosis*

Next we will annotate one of genetic diseases, *Cystic Fibrosis* (*CF*) by MeSH. Let's try to search *CF* in MeSHTERM.

```
> cols(MeSHTERM)

[1] "MESHID"       "MESHTERM"      "MESHCATEGORY"
```

MESHID, MESHTERM and MESHCATEGORY can be retrieved and ...

```
> keytypes(MeSHTERM)

[1] "MESHID"       "MESHTERM"      "MESHCATEGORY"
```

all of them are available as keytype option.

select function retrieves the rows in which MESHTERM is "*Cystic Fibrosis*" in MeSHTERM table.

```
> CF <- select(MeSHTERM, keys="Cystic Fibrosis",
+            cols=c("MESHID", "MESHTERM", "MESHCATEGORY"), keytype="MESHTERM")
> CF

    MESHID         MESHTERM MESHCATEGORY
1 D003550 Cystic Fibrosis              C
```

select function shows that MESHID of *CF* is D003550 and *CF* is in C (Disease) category.

Using MeSHSYNONYM, we can also check whether *CF* has synonyms.

```
> select(MeSHSYNONYM, keys=CF[1,1],
+        cols=c("MESHID","MESHSYNONYM"), keytype="MESHTERM")

MESHID
1 D003550
3 D003550
4 D003550
5 D003550
6 D003550
7 D003550
8 D003550
9 D003550
10 D003550
MESHSYNONYM
1 Mucoviscidosis|T047|NON|EQV|OMIM (2013)|ORD (2010)|UNK
(19XX)|740329|abcdeeef
3 Fibrocystic Disease of Pancreas|T047|NON|NRW|UNK
```

8

```
(19XX)|740329|FIBROCYSTIC DIS OF PANCREAS|abcdefv
4 Pancreatic Cystic Fibrosis|T047|NON|NRW|UNK (19XX)|740329|abcdef
5 Pulmonary Cystic Fibrosis|T047|NON|NRW|NLM (2005)|031030|PULM CYSTIC
FIBROSIS|abcdefv
6 Cystic Fibrosis, Pancreatic
7 Cystic Fibrosis, Pulmonary
8 Fibrosis, Cystic
9 Pancreas Fibrocystic Disease
10 Pancreas Fibrocystic Diseases
```

We can get to know that CF has some synonyms like Mucoviscidosis, Fibrocystic Disease of Pancreas, Pancreastic Cystic Fibrosis and so on.

MeSH also defines QUALIFIER, which is more rough category (SUBHEADING). We can also use select function against MeSHQUALIFIER.

```
> select(MeSHQUALIFIER, keys=CF[1,1],
+        cols=c("QUALIFIERID","SUBHEADING","MESHID"), keytype="MESHID")

   QUALIFIERID          SUBHEADING  MESHID
1      Q000097               blood D003550
2      Q000134  cerebrospinal fluid D003550
3      Q000139    chemically induced D003550
4      Q000145       classification D003550
5      Q000150        complications D003550
6      Q000175            diagnosis D003550
7      Q000178         diet therapy D003550
8      Q000188         drug therapy D003550
9      Q000191            economics D003550
10     Q000196           embryology D003550
11     Q000201           enzymology D003550
12     Q000208            ethnology D003550
13     Q000209             etiology D003550
14     Q000235             genetics D003550
15     Q000266              history D003550
16     Q000276           immunology D003550
17     Q000378           metabolism D003550
18     Q000382         microbiology D003550
19     Q000401            mortality D003550
20     Q000451              nursing D003550
21     Q000453         epidemiology D003550
22     Q000469          parasitology D003550
23     Q000473            pathology D003550
24     Q000503       physiopathology D003550
25     Q000517 prevention & control D003550
```

```
26      Q000523            psychology D003550
27      Q000530           radiography D003550
28      Q000531 radionuclide imaging D003550
29      Q000532          radiotherapy D003550
30      Q000534        rehabilitation D003550
31      Q000601               surgery D003550
32      Q000628               therapy D003550
33      Q000652                 urine D003550
34      Q000662            veterinary D003550
35      Q000736        ultrasonography D003550
36      Q000821              virology D003550
```

As mentioned before, MeSH has hierarchical structure. AOR tell us upper (or lower) hierarchical MeSH Term. We already know CF is categorized in C, so MeSH**C**AOR is suitable here.

```
> ao <- select(MeSHCAOR, keys=CF[1,1],
+        cols=c("ANCESTORMESHID","OFFSPRINGMESHID"), keytype="OFFSPRINGMESHID")
> ao


  ANCESTORMESHID OFFSPRINGMESHID
1       D007232         D003550
2       D008171         D003550
3       D010182         D003550
4       D030342         D003550
```

There are D007232, D008171, D010182 and D030342 above CF.

We will translate these MeSH ID to MeSH Term.

```
> select(MeSHTERM, keys=ao[,1], cols=c("MESHTERM"), keytype="MESHID")


                    MESHTERM
1 Infant, Newborn, Diseases
2             Lung Diseases
3        Pancreatic Diseases
4   Genetic Diseases, Inborn
```

CF is recognized as various kind of diseases.

PCR tell us the direct upper (or lower) MeSH Term. select function is applied to MeSH**C**PCR in the same way.

```
> pc <- select(MeSHCPCR, keys=CF[1,1],
+        cols=c("PARENTMESHID","CHILDMESHID"), keytype="CHILDMESHID")
> pc
```

```
   PARENTMESHID CHILDMESHID
1      D010182     D003550
2      D008171     D003550
3      D030342     D003550
4      D007232     D003550
```

Same MeSH IDs are retrieved, which means MeSH Term of CF is assigned in different branches of MeSH hierarchy separetely.

Once we set keytype to opposite direction (OFFSPRINGMESHID to ANCESTORMESHID), we can also retrieved MeSH ID in lower hierarchy.

```
> select(MeSHCAOR, keys=CF[1,1],
+        cols=c("ANCESTORMESHID","OFFSPRINGMESHID"), keytype="ANCESTORMESHID")

[1] ANCESTORMESHID  OFFSPRINGMESHID
<0 rows> (or 0-length row.names)
```

There are any MeSH ID, which means CF has no lower hierarchy.

References

Ashburner, M. and et al. (2000). Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat. Genet.*, 25(1):25–29.

Nakazato, T. and et al. (2007). Biocompass: a novel functional inference tool that utilizes mesh hierarchy to analyze groups of genes. *In Silico Biol.*, 8(1):53–61.

Nakazato, T. and et al. (2009). Gendoo: functional profiling of gene and disease features using mesh vocabulary. *Nucleic Acids Res.*, 37:W166–W169.

Nelson, S. J. and et al. (2004). The mesh translation maintenance system: structure, interface design, and implementation. *Stud. Health Technol. Inform.*, 107:67–69.

Sartor, M. A. and et al. (2012). Metab2mesh: annotating compounds with medical subject headings. *Bioinformatics*, 28(10):1408–1410.

Saurin, D. J. and et al. (2010). Genemesh: a web-based microarray analysis tool for relating differentially expressed genes to mesh terms. *BMC Bioinformatics*, 11:166.