

# org.Mm.cross.db

October 16, 2009

## R topics documented:

org.Mm.cross_dbconn	1
org.Mm.cross.db	2
org.Mm.crossMAPCOUNTS	3
org.Mm.crossMATCH	3
org.Mm.crossORGANISM	4
org.Mm.crossSEQ	4

<b>Index</b>	<b>5</b>
--------------	----------

---

org.Mm.cross\_dbconn  
*Collect information about the package annotation DB*

---

## Description

Some convenience functions for getting a connection object to (or collecting information about) the package annotation DB.

## Usage

```
org.Mm.cross_dbconn()  
org.Mm.cross_dbfile()  
org.Mm.cross_dbschema(file="", show.indices=FALSE)  
org.Mm.cross_dbInfo()
```

## Arguments

`file` A connection, or a character string naming the file to print to (see the file argument of the cat function for the details).

`show.indices` The CREATE INDEX statements are not shown by default. Use show.indices=TRUE to get them.

## Details

`org.Mm.cross_dbconn` returns a connection object to the package annotation DB. **IMPORTANT:** Don't call `dbDisconnect` on the connection object returned by `org.Mm.cross_dbconn` or you will break all the `AnnDbObj` objects defined in this package!

`org.Mm.cross_dbfile` returns the path (character string) to the package annotation DB (this is an SQLite file).

`org.Mm.cross_dbschema` prints the schema definition of the package annotation DB.

`org.Mm.cross_dbInfo` prints other information about the package annotation DB.

## Examples

```
## Show the first three rows.
dbGetQuery(org.Mm.cross_dbconn(), "select * from match limit 3")

## The connection object returned by org.Mm.cross_dbconn() was created with:
dbConnect(SQLite(), dbname=org.Mm.cross_dbfile(), cache_size=64000, synchronous=0)

org.Mm.cross_dbschema()

org.Mm.cross_dbInfo()
```

---

org.Mm.cross.db      *annotation data package*

---

## Description

Welcome to the `org.Mm.cross.db` annotation Package. The annotation package was built using a downloadable R package - `PAnnBuilder` (download and build your own). The purpose is to provide protein identifier mapping among several primary protein databases: [ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot\\_sprot.fasta.gz](ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.fasta.gz); <ftp://ftp.ebi.ac.uk/pub/databases/IPI/current/ipi.MOUSE.fasta.gz>; [ftp://ftp.ncbi.nih.gov/refseq/M\\_musculus/mRNA\\_Prot/mouse.protein.faa.gz](ftp://ftp.ncbi.nih.gov/refseq/M_musculus/mRNA_Prot/mouse.protein.faa.gz) UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009 ; Mus musculus 3.56, 3 Mar 2009 ; March 10, 2008

Each of these objects has their own manual page detailing where relevant data was obtained along with examples of how to use it. Many of these objects also have a reverse map available. When this is true, expect to usually find relevant information on the same manual page as the forward map.

## Examples

```
# You can learn what objects this package supports with the following command:
ls("package:org.Mm.cross.db")
```

---

`org.Mm.crossMAPCOUNTS`*Number of mapped keys for the maps in package org.Mm.cross.db*

---

## Description

`org.Mm.crossMAPCOUNTS` provides the "map count" (i.e. the count of mapped keys) for each map in package `org.Mm.cross.db`.

## Details

This "map count" information is precalculated and stored in the package annotation DB. This allows some quality control and is used by the `checkMAPCOUNTS` function defined in `AnnotationDbi` to compare and validate different methods (like `count.mappedkeys(x)` or `sum(!is.na(as.list(x)))`) for getting the "map count" of a given map.

## See Also

[mappedkeys](#), [count.mappedkeys](#), [checkMAPCOUNTS](#)

## Examples

```
org.Mm.crossMAPCOUNTS
mapnames <- names(org.Mm.crossMAPCOUNTS)
org.Mm.crossMAPCOUNTS[mapnames[1]]
x <- get(mapnames[1])
sum(!is.na(as.list(x)))
count.mappedkeys(x) # much faster!

## Check the "map count" of all the maps in package org.Mm.cross.db
checkMAPCOUNTS("org.Mm.cross.db")
```

---

`org.Mm.crossMATCH` *Map protein identifier to the matched proteins*

---

## Description

`org.Mm.crossMATCH` provides mappings between a protein identifier and the similar proteins based on BLAST program.

## Details

Mappings were based on data provided by: [ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot\\_sprot.fasta.gz](ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.fasta.gz); <ftp://ftp.ebi.ac.uk/pub/databases/IPI/current/ipi.MOUSE.fasta.gz>; [ftp://ftp.ncbi.nih.gov/refseq/M\\_musculus/mRNA\\_Prot/mouse.protein.faa.gz](ftp://ftp.ncbi.nih.gov/refseq/M_musculus/mRNA_Prot/mouse.protein.faa.gz)) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009 ; Mus musculus 3.56, 3 Mar 2009 ; March 10, 2008

**Examples**

```
x <- org.Mm.crossMATCH
# Get the protein identifiers that has matched proteins.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
# randomly display 10 proteins
sample(xx, 10)
```

---

```
org.Mm.crossORGANISM
```

*The Organism for org.Mm.cross.db*

---

**Description**

org.Mm.crossORGANISM is an R object that contains a single item: a character string that names the organism for which org.Mm.cross.db was built.

**Details**

Although the package name is suggestive of the organism for which it was built, org.Mm.crossORGANISM provides a simple way to programmatically extract the organism name.

**Examples**

```
org.Mm.crossORGANISM
```

---

```
org.Mm.crossSEQ
```

*Map protein identifier to the protein sequence*

---

**Description**

org.Mm.crossSEQ provides mappings between a protein identifier and the protein Sequence.

**Details**

Mappings were based on data provided by: [ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot\\_sprot.fasta.gz](ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/uniprot_sprot.fasta.gz); <ftp://ftp.ebi.ac.uk/pub/databases/IPI/current/ipi.MOUSE.fasta.gz>; [ftp://ftp.ncbi.nih.gov/refseq/M\\_musculus/mRNA\\_Prot/mouse.protein.faa.gz](ftp://ftp.ncbi.nih.gov/refseq/M_musculus/mRNA_Prot/mouse.protein.faa.gz)) on UniProtKB/Swiss-Prot Release 56.9 of 03-Mar-2009 ; Mus musculus 3.56, 3 Mar 2009 ; March 10, 2008

**Examples**

```
x <- org.Mm.crossSEQ
# Get the protein identifiers that are mapped to protein sequence.
mapped_proteins <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_proteins])
# randomly display 10 proteins
sample(xx, 10)
```

# Index

## \*Topic **datasets**

org.Mm.crossMAPCOUNTS, [2](#)

AnnDbObj, [1](#)

checkMAPCOUNTS, [2](#), [3](#)

count.mappedkeys, [3](#)

dbDisconnect, [1](#)

mappedkeys, [3](#)

org.Mm.cross.db, [2](#)

org.Mm.cross\_dbconn, [1](#)

org.Mm.cross\_dbfile  
(*org.Mm.cross\_dbconn*), [1](#)

org.Mm.cross\_dbInfo  
(*org.Mm.cross\_dbconn*), [1](#)

org.Mm.cross\_dbschema  
(*org.Mm.cross\_dbconn*), [1](#)

org.Mm.crossMAPCOUNTS, [2](#)

org.Mm.crossMATCH, [3](#)

org.Mm.crossORGANISM, [3](#)

org.Mm.crossSEQ, [4](#)