lumiMouseIDMapping

February 8, 2012

`lumiMouseIDMapping_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

```
lumiMouseIDMapping_dbconn()
lumiMouseIDMapping_dbfile()
lumiMouseIDMapping_dbInfo()
```

Details

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

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

`lumiMouseIDMapping_dbInfo` prints other information about the package annotation DB.

Value

`lumiMouseIDMapping_dbconn`: a DBIConnection object representing an open connection to the package annotation DB.

`lumiMouseIDMapping_dbfile`: a character string with the path to the package annotation DB.

`lumiMouseIDMapping_dbInfo`: none (invisible `NULL`).

See Also

`dbConnect`

Examples

```
## Show the database information (meta data)
lumiMouseIDMapping_dbInfo()

## List the tables included in the database
conn <- lumiMouseIDMapping_dbconn()
dbListTables(conn)
```

`lumiMouseIDMapping_nuID` *Mapping nuIDs of Illumina Mouse chips to the most recent Mus musculus RefSeq release*

Description

We mapped nuIDs of Illumina Mouse chips by BLASTing each probe sequence (converted from nuID) against the the most recent Mus musculus RefSeq release. The mapping also includes the mapping quality information.

Usage

```
lumiMouseIDMapping_nuID()
```

Details

The nuID mapping information is kept in the nuID_MappingInfo table in the ID Mapping library. The nuID mapping table includes following fields (columns):

1. nuID: nuID for the probe sequence

2. Refseq: The refseq IDs with perfect matching with probe sequence. If there are more than one refseq IDs, they are separated by ",".

3. EntrezID: The Entrez gene IDs correspond to the refseq IDs. If there are more than one Entrez gene IDs, they are separated by ",".

4. QualityScore: The mapping quality from probe sequence to RefSeq, see reference 2 for more details.

5. Refseq_old: the refseq ID provided by Illumina company when they designed the chip (included in the chip manifest file).

For the version after 1.4.0, the mapping information was got from the Computational Biology Group at University of Cambridge, see reference link for more details.

Value

`lumiMouseIDMapping_nuID` returns a nuID mapping summary of Illumina Mouse chips.

References

1. Du, P., Kibbe, W.A. and Lin, S.M., "nuID: A universal naming schema of oligonucleotides for Illumina, Affymetrix, and other microarrays", Biology Direct 2007, 2:16 (31May2007). 2. http://www.compbio.group.cam.ac.uk/Resources/Annotation/index.html

Examples

```
## List the fields in the nuID_MappingInfo table
conn <- lumiMouseIDMapping_dbconn()
dbListFields(conn, 'nuID_MappingInfo')

## Summary of nuID mapping
lumiMouseIDMapping_nuID()
```

`lumiMouseIDMapping` *Illumina ID Mapping information of all Mouse expression chips*

Description

Welcome to the lumiMouseIDMapping ID mapping Package. The purpose of this package is to provide ID mappings between different types of Illumina identifiers of Mouse Expression chips and nuIDs, and also mappings from nuIDs to the the most recent Mus musculus RefSeq release and Entez_Gene_ID. The library includes the data tables corresponding to all released Illumian Mouse Expression chips before the package releasing date. Each table includes columns "Search_key" ("Search_Key"), "Target" ("ILMN_Gene"), "Accession", "Symbol", "ProbeId" ("Probe_Id") and "nuID". It also includes a nuID_MappingInfo table, which keeps the mapping information of nuID to Accession ID, Entrez_Gene_ID and Symbol. For the version after 1.8.0, all mapping information was based on Illumina manifest files. Information from new versions of manifest files will replace the old ones with the same probe id. The package is supposed to be used together with the Bioconductor lumi package.

You can learn what objects this package supports with the following command:

`ls("package:lumiMouseIDMapping")`

Each of these objects has their own manual page detailing where relevant data was obtained along with some examples of how to use it.

Examples

```
ls("package:lumiMouseIDMapping")
```

Index