

# Introduction to LRBase.Dme.eg.db and LRBase.XXX.eg.db type packages

*Koki Tsuyuzaki<sup>1</sup>, Manabu Ishii<sup>1</sup>, and Itoshi Nikaido<sup>1</sup>.*

January 2, 2019

<sup>1</sup>Laboratory for Bioinformatics Research, RIKEN Center for Biosystems Dynamics Research, Japan  
k.t.the-answer@hotmail.co.jp

## 1 Introduction

---

This document provides the way to use LRBase.Dme.eg.db and LRBase.XXX.eg.db-type packages. LRBase.XXX.eg.db-type packages provide the pair list of ligand - receptor genes. The packages are generated by the LRBase.Dme.eg.db package. LRBase.Dme.eg.db has two role; class-definition and the construction of LRBase.XXX.eg.db-type packages. LRBase.Dme.eg.db defines the class LRBaseDb and unify the objects's behavior such as column function described later. The makeLRBasePackage function of LRBase.Dme.eg.db generates the user's original LRBase.XXX.eg.db-type packages.

## 2 makeLRBasePackage

---

Here we use makeLRBasePackage function to create a LRBase.XXX.eg.db-type package. Only user have to specify are 1. a LR-list containing the columns GENEID\_L (ligand NCBI Gene IDs) and GENEID\_R (receptor NCBI Gene IDs) and 2. a meta information table describing the LR-list. Here we use the demo data of LR-list of FANTOM5 project.

```
> library('LRBase.Dme.eg.db')

[1] "LRBase.Dme.eg.db" "LRBaseDbi"      "stats"          "graphics"
[5] "grDevices"        "utils"          "datasets"       "methods"
[9] "base"

> if(interactive()){
+   example('makeLRBasePackage')
+ }

NULL
```

## Introduction to LRBase.Dme.eg.db and LRBase.XXX.eg.db-type packages

After `makeLRBasePackage`, `FANTOM5.Hsa.eg.db` is generated. Here, we will install the package.

```
> if(interactive()){
+   filepath <- list.files(destination, full.names=TRUE)
+   install.packages(filepath, repos=NULL, type='source')
+   library('FANTOM5.Hsa.eg.db')
+ }
```

## 3 columns, keytypes, keys, and select

---

All `LRBase.XXX.eg.db`-type package has same name object and it is instantiated by `LRBaseDb`-class. Many data access function for this object are implemented. For example, `columns` returns the rows which we can retrieve in `LRBase.XXX.eg.db`-type packages. `keytypes` returns the rows which can be used as the optional parameter in `keys` and `select` functions against `LRBase.XXX.eg.db`-type packages. `keys` function returns the value of keytype. `select` function returns the rows in particular columns, which are having user-specified keys. This function returns the result as a dataframe.

```
> if(interactive()){
+   columns(FANTOM5.Hsa.eg.db)
+   keytypes(FANTOM5.Hsa.eg.db)
+   key_FN5 <- keys(FANTOM5.Hsa.eg.db, keytype='GENEID_R')
+   head(select(FANTOM5.Hsa.eg.db, keys=key_FN5[1:2],
+             columns=c('GENEID_L', 'GENEID_R'), keytype='GENEID_R'))
+ }
```

## 4 Other functions

---

Other additional functions like `species`, `nomenclature`, and `listDatabases` are available. In each `LRBase.XXX.eg.db`-type package, `species` function returns the common name and `nomenclature` returns the scientific name. `listDatabases` function returns the source of data. `dbInfo` returns the information of the package. `dbfile` returns the directory where sqlite file is stored. `dbschema` returns the schema of database. `dbconn` returns the connection to the sqlite database.

```
> if(interactive()){
+   species(FANTOM5.Hsa.eg.db)
+   nomenclature(FANTOM5.Hsa.eg.db)
+   listDatabases(FANTOM5.Hsa.eg.db)
+   dbInfo(FANTOM5.Hsa.eg.db)
+ }
```

```
+ dbfile(FANTOM5.Hsa.eg.db)
+ dbschema(FANTOM5.Hsa.eg.db)
+ dbconn(FANTOM5.Hsa.eg.db)
+ }
```

## 5 Redirecting to the scTensor package

---

Description for any LRBase-related packages is written in the vignette of *scTensor* package. Please follow the link below

<http://www.bioconductor.org/packages/release/bioc/html/scTensor.html>

or just type

```
> if(interactive()){
+   if (!requireNamespace('BiocManager', quietly = TRUE)){
+     install.packages('BiocManager')
+   }
+   BiocManager::install('scTensor')
+   library('scTensor')
+   vignette('scTensor')
+ }
```

in R console window.