

# Introduction to LRBaseDbi

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

**April 25, 2023**

<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 LRBaseDbi and create LRBaseDb objects with AnnotationHub packages. LRBaseDb is the pair list of ligand-receptor (L-R) genes. LRBaseDb is generated by the LRBaseDbi package. LRBaseDbi has two roles; class-definition and construction of LRBaseDb. LRBaseDbi defines a class "LRBaseDb" and unifies the object's behavior such as column function described later.

To use LRBaseDbi, type as follows.

```
> library("LRBaseDbi")
> library("AnnotationHub")
> ah <- AnnotationHub()
> dbfile <- query(ah, c("LRBaseDb", "Sus scrofa", "v001"))[[1]]
> # Constructor
> LRBase.Ssc.eg.db <- LRBaseDbi::LRBaseDb(dbfile)
```

Here, we created a LRBaseDb for Sus scrofa (Pig). The internal data is retrieved from the remote server of AnnotationHub and stored in the cache directory. LRBaseDb function of LRBaseDbi is the constructor to create the LRBase object (LRBase.Ssc.eg.db).

## 2 columns, keytypes, keys, and select

---

Many data access functions for LRBaseDb object are implemented. For example, columns returns the rows which we can retrieve in LRBaseDb. keytypes returns the rows which can be used as the optional parameter in keys and select functions against LRBaseDb. 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 data frame.

```
> if(interactive()){
+   (cols <- columns(LRBase.Ssc.eg.db))
```

```

+ keytypes(LRBase.Ssc.eg.db)
+ (ks <- keys(LRBase.Ssc.eg.db, keytype='GENEID_R'))
+ head(select(LRBase.Ssc.eg.db, keys=ks[1:2],
+           columns=c('GENEID_L', 'GENEID_R'), keytype='GENEID_R'))
+ }

```

### 3 Other functions

---

Other additional functions are available. `dbconn` returns the connection to the sqlite database. `dbfile` returns the directory where sqlite file is stored. `dbschema` returns the schema of database. `dbInfo` returns the information of the package. `species` function returns the common name. `lrNomenclature` returns the scientific name. `lrListDatabases` function returns the source of data. `lrVersion` returns the version of LRBaseDb.

```

> if(interactive()){
+   # show
+   LRBase.Ssc.eg.db
+   # dbconn
+   dbconn(LRBase.Ssc.eg.db)
+   # dbfile
+   dbfile(LRBase.Ssc.eg.db)
+   # dbschema
+   dbschema(LRBase.Ssc.eg.db)
+   # dbInfo
+   dbInfo(LRBase.Ssc.eg.db)
+   # species
+   species(LRBase.Ssc.eg.db)
+   # lrNomenclature
+   lrNomenclature(LRBase.Ssc.eg.db)
+   # lrListDatabases
+   lrListDatabases(LRBase.Ssc.eg.db)
+   # lrVersion
+   lrVersion(LRBase.Ssc.eg.db)
+ }

```

### 4 Redirecting to the scTensor package

---

How to use LRBaseDb with scTensor is written in the vignette of *scTensor* package.

## Introduction to LRBaseDbi

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.