toxvaldbBMDh

February 23, 2024

Type Package

Title Calculates BMDh Values From Records in ToxValDB

Version 0.1.0

Author Richard Judson

Maintainer Richard Judson <judson.richard@epa.gov>

Description This package extracts data fro mteh ToxValDB database, filters it
for redundant records (this redundancy should be reduced in later versions),
calculates study-level BMD valuesand finally calculates chemical-level BMD values.
To run th entire process, uses the functions export.for.bmdh(), filter.for.bmdh(),
bmd.per.study(), bmd.per.chemical() and bmdh.percentile.plot(). The function driver() runs all of the
methods sequentially. The last function provides the best percentile to use.
All of the input and output data lives in the data/ folder

License BSD3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

R topics documented:

`bmdh.aurisano.check.plot` *Plot the different between the Aurisano and current BMDh values*

Description

Plot the different between the Aurisano and current BMDh values

Usage

```
bmdh.aurisano.check.plot(
  to.file = F,
  toxval.db = "res_toxval_v95",
  sys.date = "2024-01-04"
)
```

Arguments

`dir`	The directory where the lists are stored

`bmdh.per.chemical` *Calculate BMDh values one per chemical*

Description

'bmdh.per.study' Calculates one BMDh value per chemical. This is done by taking various percentiles of the distribution of the BMDh values and building a table with one column per percentile per chemical. The values are calibrated against regulatory values. The list of high-quality, regulator sources is given as one of the calling arguments.

Usage

```
bmdh.per.chemical(
  toxval.db = "res_toxval_v95",
  sys.date = "2024-02-23",
 regulatory.sources = c("IRIS", "PPRTV (CPHEA)", "ATSDR MRLs 2022", "ATSDR PFAS 2021",
    "EPA OPP", "HEAST")
)
```

Arguments

`toxval.db`	Database version
`sys.date`	The date of the database export
`regulatory.sources`	This is the list of sources that will be used to select the optimal quantile to use for selecting the final chemical-level BMDh.

Value

Write a file with the results: toxval_PODs_for_BMDh chemical level toxval.db sys.date.xlsx

`bmdh.per.study` *Calculate the BMDh values per study*

Description

'bmdh.per.study' Calculates one BMDh value per study using the Aurisano algorithm. Because EPA has not fully developed the mapping from critical effects in ToxValDB to standardized effects, teh values from Aurisano are used where records match. Aurisano used ToxValDB 9.1, whereas 9.5 is used here. There is also code here to do the other required mappings, and thos may need to be updated.For records in both the old and new databases, an on-the-fly plot is produced to show the corresponded between study-level BMDh values.

Usage

```
bmdh.per.study(toxval.db = "res_toxval_v95", sys.date = "2024-02-23")
```

Arguments

`toxval.db`	Database version
`sys.date`	The date of the database export

Value

Write a file with the results: toxval_PODs_for_BMDh toxval.db sys.date.xlsx

`bmdh.percentile.plot` *Plot the BMDs vs the regulatory values for different percentiles and determine the best fit*

Description

'bmdh.percentile.plot' Helps determine the optimal percentile. The output file shows the fit statistics for different percentiles, and one should select the one with the lowest RMSE and highest R2.

Usage

```
bmdh.percentile.plot(
  to.file = F,
  toxval.db = "res_toxval_v95",
  sys.date = "2024-02-23",
  minstudies = 10,
  cutoff.logsd = 2
)
```

Arguments

to.file	If TRUE, send the plot to a file
toxval.db	Database version
sys.date	The date of the database export
minstudies	- only chemicals with this minimum number of studies will be used in the calculation
cutoff.logsd	Only chemicals with their log SD of BMDh values will be used in the calculation

Value

Write a file with the results: toxval_PODs_for_BMDh chemical level toxval.db sys.date.xlsx

---

contains *Find out if one string contains another*

Description

Find out if one string contains another

Usage

```
contains(x, query, verbose = F)
```

Arguments

x	The string to be searched in
query	the second string
verbose	if TRUE, the two strings are printed

Value

if x contains query, return TRUE, FALSE otherwise

---

driver *Run all of the calculations to go from database export to calcualtion of final BMDh values*

Description

'driver' Run all of the calculations to go from database export to calcualtion of final BMDh values

Usage

```
driver(toxval.db = "res_toxval_v95", sys.date = "2024-02-23", user, password)
```

Arguments

toxval.db	Database version
sys.date	The date of the database export
user	The username for the MySQL database. The database instance is hard-coded in the function setDBConn().
password	The user's MySQL database password.

---

export.for.bmdh *Export records required for calculating BMDh values.*

Description

'export.for.bmdh' Exports all of the data required for the BMDh calculations. The main query may need to be modified to extract more columns if needed for the final application. Certain sources have been excluded because they have a high percentage of read-across values. Species are filtered to only include Human, Dog, Mouse, Rat and Rabbit. If more species are to be included, then allometric scaling factors for those need to added to the function bmd.per.study().

Usage

```
export.for.bmdh(toxval.db = "res_toxval_v95", user, password)
```

Arguments

toxval.db	Database version
user	The username for the MySQL database. The database instance is hard-coded in the function setDBConn().
password	The user's MySQL database password.

Value

Write a file with the results: ToxValDB for BMDh toxval.db Sys.Date().xlsx

---

filter.for.bmdh *Filter the exported records for redundancy*

Description

'filter.for.bmdh' Filters redundant rows in the raw database export. There are two kinds of redundancy. The first filters extra reference rows from the record_source table. The main data is in the toxval tables, and references are linked through the toxval_id to the record_source table. During the curation process, these references get cleaned and hence repeated, so a single (and the final or best) is selected. Then, there are redundancies in the toxval table itself, likely caused by upstream processing issues. These will be solved there, but the filtering in this function takes care of this issue for the moment.

Usage

```
filter.for.bmdh(toxval.db = "res_toxval_v95", sys.date = "2024-02-23")
```

Arguments

toxval.db	Database version
sys.date	The date of the export

Value

Write a file with the filtered results:ToxValDB for BMDh filtered toxval.db sys.date.xlsx

---

printCurrentFunction *Print the name of the current function*

Description

Print the name of the current function

Usage

```
printCurrentFunction(comment.string = NA)
```

Arguments

comment.string	An optinal string to be printed

---

runInsert *Insert a record into a database. if auto.increment=TRUE, return the auto incremented primary key of the record. otherwise, return -1*

Description

Insert a record into a database. if auto.increment=TRUE, return the auto incremented primary key of the record. otherwise, return -1

Usage

```
runInsert(query, db, do.halt = F, verbose = F, auto.increment.id = F)
```

Arguments

query	a properly formatted SQL query as a string
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information
auto.increment	if TRUE, add the auto increment primary key even if not part of the query

Value

Returns the database table auto incremented primary key ID

runInsertTable *Inserts multiple rows into a database table*

Description

Inserts multiple rows into a database table

Usage

```
runInsertTable(mat, table, db, do.halt = T, verbose = F, get.id = T)
```

Arguments

mat	data frame containing the data, with the column names corresponding
table	name of the database table to which data will be inserted
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information

runQuery *Runs a database query and returns a result set*

Description

Runs a database query and returns a result set

Usage

```
runQuery(query, db, do.halt = T, verbose = F)
```

Arguments

query	a properly formatted SQL query as a string
db	the name of the database
do.halt	if TRUE, halt on errors or warnings
verbose	if TRUE, print diagnostic information

setDBConn *set SQL connection to the database*

Description

set SQL connection to the database

Usage

```
setDBConn(
  server = "ccte-mysql-res.epa.gov",
  user = "rjudson",
  password = NA,
  port = -1
)
```

Arguments

server	SQL server on which relevant database lives
user	SQL username to access database
password	SQL password corresponding to username

toxval.redundancies *Detect potential redundancies in ToxValDB.*

Description

Detect potential redundancies in ToxValDB.

Usage

```
toxval.redundancies(toxval.db = "res_toxval_v95", user = "rjudson", password)
```

Arguments

toxval.db	Database version
user	The username for the MySQL database. The database instance is hard-coded in the function setDBConn().
password	The user's MySQL database password.

Value

Write a file with the results

Index