

# Package ‘toxboot’

August 29, 2016

**Type** Package

**Title** Bootstrap Methods for 'ToxCast' High Throughput Screening Data

**Version** 0.1.1

**Date** 2016-08-23

**URL** <https://github.com/ericwatt/toxboot>

**BugReports** <https://github.com/ericwatt/toxboot/issues>

**Description** Provides methods to use bootstrapping to quantify uncertainty in fitting 'ToxCast' concentration response data. Data is stored in memory, written to file, or stored in 'MySQL' or 'MongoDB' databases.

**License** MIT + file LICENSE

**LazyData** TRUE

**Depends** R (>= 3.2)

**Imports** data.table, tcpl (>= 1.0),

**Suggests** DBI, RMySQL, rmongodb, ggplot2, knitr, parallel, rmarkdown, testthat, magrittr, pander, covr

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Eric D. Watt [aut, cre]

**Maintainer** Eric D. Watt <eric@ericdwatt.com>

**Repository** CRAN

**Date/Publication** 2016-08-23 20:00:18

## R topics documented:

dat1000	2
dat10000	3
erl3data	3
erl5data	4

gnls_curve . . . . .	5
hill_curve . . . . .	5
listTonumeric . . . . .	6
nullToNA . . . . .	7
toxboot . . . . .	7
Toxboot configuration functions . . . . .	9
toxbootConnectMongo . . . . .	10
toxbootGetMongoFields . . . . .	10
toxbootGetMySQLFields . . . . .	11
toxbootGetOpts . . . . .	12
toxbootHitParamCI . . . . .	12
toxbootmc . . . . .	13
toxbootMongoCount . . . . .	14
toxbootMysqlCreateTable . . . . .	14
toxbootProjectionBuild . . . . .	15
toxbootQueryBuild . . . . .	15
toxbootQueryToxCast . . . . .	16
toxbootReplicates . . . . .	16
toxbootSetOpts . . . . .	17
toxbootWriteMongo . . . . .	17

## Index 19

---

dat1000	<i>Bootstrap results for m4id = 9057756 using 1000 replicates</i>
---------	---

---

### Description

A dataset containing the bootstrapped results of `m4id = 9057756` found in the `erl3data` dataset. Used as an example for plotting in the vignette, which shows the code used to generate the data.

### Usage

```
dat1000
```

### Format

A `data.table` with 1000 rows and 59 columns

---

`dat10000`*Bootstrap results for m4id = 9057756 using 10000 replicates*

---

**Description**

A dataset containing the bootstrapped results of `m4id = 9057756` found in the `erl3data` dataset. Used as an example for plotting in the vignette, which shows the code used to generate the data.

**Usage**`dat10000`**Format**

A `data.table` with 10000 rows and 59 columns

---

`erl3data`*Concentration response data from 200 ER assay curves*

---

**Description**

A dataset containing the output from `toxbootQueryToxCast`. A representative assay from each vendor was chosen to pass to `toxbootQueryToxCast`, and a random sampling of 200 `m4ids` were then selected from this output.

**Usage**`erl3data`**Format**

A `data.table` with 5580 rows and 16 variables:

- `m3id` : unique identifier for each concentration response value
- `spid` : sample id
- `aeid` : assay endpoint id
- `logc` :  $\log_{10}$  of the  $\mu\text{M}$  concentration
- `resp` : response value
- `m4id` : unique identifier for the concentration response curve
- `bmad` : baseline median absolute deviation. A measure of baseline noise that is used in the fitting routine by `tcplFit` and is used as by `toxboot` as a measurement of noise when `boot_method = smooth`.

**Details**

The assays chosen were:

- NVS\_NR\_bER
- OT\_ER\_ERaERa\_1440
- ATG\_ERa\_TRANS\_up
- TOX21\_ERa\_LUC\_BG1\_Agonist
- ACEA\_T47D\_80hr\_Positive

**Source**

<https://www.epa.gov/chemical-research/toxicity-forecaster-toxcasttm-data/>

---

er15data

*Fit parameters, model selection, and hit calls from 200 ER assay curves*

---

**Description**

A dataset containing the ToxCast level 5 data corresponding to the m4ids in [er13data](#). This data is provided so that the cutoff values for these assays can be used in [toxbootHitParamCI](#) to determine winning model and hit call selection for bootstrap results. This dataset is also useful to compare min/med/max bootstrap results to the point estimate in the ToxCast pipeline value.

**Usage**

er15data

**Format**

A data.table with 200 rows and 67 variables

**Source**

<https://www.epa.gov/chemical-research/toxicity-forecaster-toxcasttm-data/>

---

gnls_curve	<i>Gain Loss Curve.</i>
------------	-------------------------

---

**Description**

gnls\_curve is used calculate response values given gnls parameters.

**Usage**

```
gnls_curve(top, ga, gw, la, lw, lconc)
```

**Arguments**

top	Numeric, gnls top parameter.
ga	Numeric, gnls gain AC50 parameter.
gw	Numeric, gnls gain coefficient parameter.
la	Numeric, gnls loss AC50 parameter.
lw	Numeric, gnls loss coefficient parameter.
lconc	Numeric, concentration (in log uM) to calculate the response.

**Details**

Given input values for the gnls parameters and a concentration, this function returns the response value at the given concentration. Used to plot the gnls equation resulting from a given fit.

**Value**

y Numeric, the calculated response value at lconc.

**Examples**

```
gnls_curve(top = 100, ga = 0.5, gw = 1, la = 1.5, lw = 8, lconc = 0)
```

---

hill_curve	<i>Hill Curve.</i>
------------	--------------------

---

**Description**

hill\_curve is used calculate response values given hill parameters.

**Usage**

```
hill_curve(hill_tp, hill_ga, hill_gw, lconc)
```

**Arguments**

hill_tp	Numeric, hill top parameter.
hill_ga	Numeric, hill AC50 parameter.
hill_gw	Numeric, hill coefficient parameter.
lconc	Numeric, concentration (in log uM) to calculate the response.

**Details**

Given input values for the hill parameters and a concentration, this function returns the response value at the given concentration. Used to plot the hill equation resulting from a given fit.

**Value**

y Numeric, the calculated response value at lconc.

**Examples**

```
hill_curve(hill_tp = 100, hill_ga = 0.5, hill_gw = 1, lconc = 0)
```

---

listTonumeric	<i>Convert x to numeric if x is a list.</i>
---------------	---

---

**Description**

listTonumeric is used to convert x to a numeric type if x is a list.

**Usage**

```
listTonumeric(x)
```

**Arguments**

x	A list of length Y without NULL values.
---	---

**Details**

This function is used as part of reading data from mongoDB using rmongodb. Typically values will be stored as numeric vectors. However, if an NA is in the original vector, rmongodb will convert to NULL prior to writing to the database. This gets read from the database as a list of length Y rather than a numeric of length Y with positions that were NA now with value NULL. The function nullToNA converts the NULL values to NA, and this function converts the list back to a numeric type.

**Value**

x a numeric vector of length Y.

**Examples**

```
## Not run: class(listToNumeric(list(1,2,3,4,NA,6,7)))
```

---

nullToNA	<i>Replace NULL values with NA.</i>
----------	-------------------------------------

---

**Description**

nullToNA is used to convert any NULL values to NA in a list.

**Usage**

```
nullToNA(x)
```

**Arguments**

x                    A list.

**Details**

This function is used when reading vectors from mongoDB. If the original vector had a value of NA, rmongodb converts to NULL before writing to the database. When rmongodb performs a find, these NULL values are left as NULL, and when unlisted will shorten the vector. Before unlisting the vector, this function is run to convert the NULL values to NA so that the vector remains the correct length and that NAs are in the correct positions.

**Value**

x with all NULL at the top level of the list replaced by NA.

**Examples**

```
## Not run: nullToNA(list(1,5,'a', NULL, NA, 4))
```

---

toxboot	<i>Main function to sample, fit, and write results to mongoDB</i>
---------	---

---

**Description**

toxboot is the main function that performs the bootstrap sampling, fitting, and writing to the database

**Usage**

```
toxboot(dat, m4id, boot_method, replicates = 100, concvals = F,
        destination = "memory")
```

## Arguments

dat	A data.table. Required columns are: logc: numeric, contains concentrations resp: numeric, normalized response values paired with concentrations m3id: numeric, value unique to each row corresponding to an individual concentration and response m4id: numeric, value unique to an aeid/spid pair. Multiple m3ids per m4id aeid: numeric, assay id spid: character, sample ID bmad: numeric, baseline mad. Unique to an aeid.
m4id	numeric length 1, m4id to bootstrap. Choice of m4id will determine which rows are selected, and therefore the values of logc, resp, m3id, aeid, spid, and bmad.
boot_method	parameter passed to toxbootReplicates to determine sampling method
replicates	number of bootstrap samples. Default 100
concvals	logical, default is FALSE. If TRUE, dose response samples written to the database as well.
destination	string length 1, options are "mongo", "mysql", "file", "memory"

## Details

toxboot is the workhorse function of this package. This function will typically be wrapped in a mclapply to perform in parallel using [toxbootmc](#). The dose response data is passed to toxbootReplicates. The returned matrix is passed to tcpl::tcplFit.

There are multiple options for saving the results, based on the value of destination.

- mongo : If destination is set to "mongo" a connection to the mongo database will be created and the results will be written using the rmongodb package. The connection will be established using the parameters retrieved using [toxbootConfList](#) by the function [toxbootConnectMongo](#). See the documentation on these functions as well as [toxbootConf](#) for how to properly setup the MongoDB environment. For large scale screening of uncertainty parameters it is recommended that MongoDB be used for performance and scaling.
- mysql :
- file : A directory toxboot/ will be created. A csv file for each m4id will be created with name set to the value of m4id. The format of the file will be tabular with one row for each bootstrap replicate. Subsequent runs will be appended onto the file. This way further bootstrap results can be created without loss of previous computational work. Note that the file size can get quite large if many curves are run. With 1000 replicates the file size will typically range from 300 to 600 KB per m4id.
- memory : Results will be returned as a single data.table. This is a reasonable option for checking a few curves or even an entire assay with 1000 replicates if a suitable amount of memory is available. Care must be taken as the resulting data.table can become multiple GB in memory

The fitted results are assembled into a bson object using rmongodb and written to the mongoDB.

## See Also

[toxbootReplicates](#)



---

**Toxboot configuration functions***Functions for configuring the toxboot package*

---

**Description**

This set of functions are used to configure the settings used to interact with the mongoDB database.

**Usage**

```
toxbootConf(mongo_host = NULL, DBNS = NULL, user = NULL, pass = NULL,  
            db = NULL)
```

```
toxbootConfSave()
```

```
toxbootConfReset()
```

```
toxbootConfLoad()
```

```
toxbootConfList(show.pass = FALSE)
```

**Arguments**

<code>mongo_host</code>	Character of length 1, database IP address
<code>DBNS</code>	Character of length 1, the collection on the database to read from and write to
<code>user</code>	Character of length 1, username to authenticate
<code>pass</code>	Character of length 1, password that corresponds to username
<code>db</code>	Character of length 1, database where the username and pass are authenticated. Can be an IP address or a url.
<code>show.pass</code>	Logical, should the password be returned

**Details**

Parameter settings are stored in two places. Long term storage on disk is in a configuration file, `toxboot.config`, located in the `toxboot` package directory within the library. This allows default settings to be maintained between sessions without a need for including IP addresses and login credentials within user scripts. During runtime, parameter settings are set in `options` to be used by various `toxboot` functions. The `toxboot` configuration functions are used to read and/or write parameters from/to the `toxboot.config` configuration files at the settings in `options`.

`toxbootConf` changes `options` to set the `toxboot` options used to interact with mongoDB when reading or writing results. This includes the location of the database, which collection to use, and a username and password to authenticate against the database. `toxbootConf` will only change non-null values, and can be used to change a single value if needed.

`toxbootConfSave` modifies the `toxboot.config` file. Current global parameters are ready from `options` using `toxbootConfList` and the values are stored in `toxboot.config` within the `toxboot` directory within the library.

toxbootConfList returns the toxboot global parameters from options.

toxbootConfLoad reads the toxboot parameters from toxboot.config and updates options with these settings.

toxbootConfReset will overwrite toxboot.config setting all parameters to NA. This is used to remove any user information from the file.

---

toxbootConnectMongo    *Setup connection to toxboot MongoDB*

---

### Description

toxbootConnectMongo imports toxboot options and creates the connection to the mongoDB.

### Usage

```
toxbootConnectMongo()
```

### Value

mongo, a mongo connection object.

---

toxbootGetMongoFields    *Query mongoDB and get requested fields*

---

### Description

toxbootGetMongoFields will query the mongo database for matching bootstrap type(s), assay acid(s), and sample id(s). It will then return a data.table with the requested fields.

### Usage

```
toxbootGetMongoFields(fields, ...)
```

### Arguments

fields	String, a vector of fields to return from the database. Passed to toxbootProjectionBuild
...	parameters to query on, passed to toxbootQueryBuild. Format is query_field = query_values

### Details

Longer description here

### Value

data table of requested fields

**Examples**

```
## Not run:
toxbootGetMongoFields(boot_type = "smooth", aeid = 1508L,
fields = c("hill_ga", "gnls_ga"))

## End(Not run)
```

---

toxbootGetMySQLFields *Function to query toxboot results from MySQL*

---

**Description**

toxbootGetMySQLFields queries the toxboot MySQL table and returns a data.table with the requested results

**Usage**

```
toxbootGetMySQLFields(fields = "*", table = "toxboot", ...)
```

**Arguments**

fields	a vector specifying which columns to return. Default to '*' which will return all columns
table	the name of the table to query. By default 'toxboot'.
...	parameters to query on. Format is query_field = query_values.

**Details**

Use the fields parameters to specify which columns to return. The parameter 'table' defaults to 'toxboot' which is the default table for writing and reading results. All other parameters will be passed as values to select on.

**Examples**

```
## Not run:
toxbootmc(dat = erl3data,
          m4ids = unique(erl3data[, m4id])[1:50],
          boot_method = "smooth",
          destination = "mysql",
          replicates = 10,
          cores = 8)
toxbootGetMySQLFields()
toxbootGetMySQLFields(m4id = 1288500)
toxbootGetMySQLFields(m4id = 1288500, fields = c("boot_method", "hill_ga"))
toxbootGetMySQLFields(m4id = 1288500, replicates = 10, fields = c("boot_method", "hill_ga"))

## End(Not run)
```

---

toxbootGetOpts	<i>DEPRECATED</i>
----------------	-------------------

---

**Description**

toxbootGetOpts is deprecated. See toxbootConfList.

**Usage**

```
toxbootGetOpts(show.pass = FALSE)
```

**Arguments**

show.pass	Logical, should the password be returned
-----------	--

---

toxbootHitParamCI	<i>Performs Hit Call and Model Winner Selection on Bootstrap Results</i>
-------------------	--

---

**Description**

toxbootHitParamCI takes input dose response data and returns a matrix of sampled dose response values.

**Usage**

```
toxbootHitParamCI(dat_db, dat_pipe)
```

**Arguments**

dat_db	data.table, a data.table resulting from a toxboot query minimally with columns m4id, max_med, hill_ga, hill_gw, hill_tp, hill_aic, gnls_ga, gnls_gw, gnls_tp, gnls_la, gnls_lw, gnls_aic, cnst_aic
dat_pipe	data.table. Minimally this data.table must have a column for m4id and coff (cutoff value, used in determining the hitcall). Often use the resulting data.table from tcplLoadData(lvl = 5L).

**Details**

Based upon mc5 processing in tcpl.

This function will first choose a winning model for each bootstrap replicate, based on that row's values for cnst\_aic, hill\_aic, gnls\_aic.

Second, if the winning model is hill or gnls, the max\_med is greater than the cutoff, and the winning model modl\_tp parameter is greater than the cutoff, the hitcall for that row is 1 (boot\_hitc == 1).

Finally, if boot\_hitc == 1, values for modl\_ga, modl\_gw, and modl\_tp are set to the corresponding values in the winning model. Else, these parameters are set to NA.

**Value**

dat\_boot, a data.table equivalent to dat\_db with added columns coff (from dat\_pipe), maic (winning model aic), modl (winning model), boot\_hitc, modl\_ga, modl\_gw, and modl\_tp

---

toxbootmc	<i>Filter and multicore wrapper for <a href="#">toxboot</a></i>
-----------	---

---

**Description**

toxbootmc calculates bootstrapped curves for multiple m4ids using mclapply to perform the calculation on many cores.

**Usage**

```
toxbootmc(dat, m4ids = NULL, filter = TRUE, boot_method = "smooth",
  cores = 1L, destination = "memory", ...)
```

**Arguments**

dat	data.table returned from <a href="#">toxbootQueryToxCast</a>
m4ids	list of m4ids to perform bootstrap calculation. By default NULL, which will run on all m4ids in the in dat.
filter	logical, default TRUE. If TRUE dat will be filtered to remove m4ids that are already in the mongoDB. Ignored if destination is not "mongo".
boot_method	parameter passed to <a href="#">toxboot</a> . Used also if filter is TRUE.
cores	integer length 1, number of cores to supply as mc.cores to <a href="#">mclapply</a> . By default set to 1L which will work on all systems including Windows. Increase to the number of cores on your system (see <a href="#">detectCores</a> ) to perform calculations in parallel.
destination	string length 1, options are "mongo", "mysql", "file", "memory"
...	parameters passed to <a href="#">toxboot</a>

**Details**

The function [toxboot](#) takes performs bootstrap analysis on a single m4id included in the data.table supplied. This function acts as a wrapping and cleaning function around [toxboot](#) to both simplify setup with common defaults and to provide multicore support through the use of [mclapply](#).

**Value**

dat, a data.table corresponding to the level 3 data for the acids with columns added for m4id and bmad.

---

toxbootMongoCount      *Get number of documents that match query.*

---

**Description**

toxbootMongoCount returns the number of objects that match the query.

**Usage**

```
toxbootMongoCount(mongo, query = NULL)
```

**Arguments**

mongo,                    a mongo connection object returned from toxbootConnectMongo()  
query,                    a mongo.bson object returned from toxbootQueryBuild. By default query will match all documents in the collection.

**Value**

num\_results, an integer equal to the number of documents that match the query.

---

toxbootMysqlCreateTable  
*Function to create properly formatted toxboot table in MySQL*

---

**Description**

toxbootMysqlCreateTable creates table with correct column types which will be written to by toxboot and toxbootmc.

**Usage**

```
toxbootMysqlCreateTable()
```

**Details**

Caution. This function will delete the current toxboot table if it exists. Still need to add some safeguards.

---

`toxbootProjectionBuild`*Build projection for toxboot MongoDB*

---

**Description**

`toxbootProjectionBuild` builds a list that determined which fields are returned.

**Usage**

```
toxbootProjectionBuild(fields, id_val = 0L)
```

**Arguments**

<code>fields</code>	string vector, fields to return.
<code>id_val</code>	integer 0L or 1L. By default 0L, which will suppress Object_ID from the projection. This unique ID is often not needed. Can be set to 1L to include in projection.

**Value**

projection, a named list where names are the fields to return in the projection and the values are 0L (don't return) or 1L (return).

---

`toxbootQueryBuild`*Build query for toxboot MongoDB*

---

**Description**

`toxbootQueryBuild` imports toxboot query from list of input parameters.

**Usage**

```
toxbootQueryBuild(...)
```

**Arguments**

... parameters to query on. Format is `query_field = query_values`

**Value**

query, a mongo.bson object used for a query

---

toxbootQueryToxCast     *Query ToxCast and format data to pass to toxboot*

---

### Description

toxbootQueryToxCast passes .

### Usage

```
toxbootQueryToxCast(aeids)
```

### Arguments

aeids                    integer vector contains which aeids to query. Values will be passed to tcplLoad-Data.

### Details

Because bmad is calculated for all chemicals in the assay, the entire aeid must be queried. After calculation of bmad chemicals can be further subset prior to running toxboot.

The assumption is that [tcplConf](#) has been setup to pull data from the correct database.

### Value

dat, a data.table corresponding to the level 3 data for the aeids with columns added for m4id and bmad.

---

toxbootReplicates     *Generates bootstrap samples of dose response data*

---

### Description

toxbootReplicates takes input dose response data and returns a matrix of sampled dose response values.

### Usage

```
toxbootReplicates(datchemval, boot_method = "smooth", replicates = 100)
```

### Arguments

datchemval            data.table, columns logc, resp, bmad  
boot\_method            string, specifies method. Default is 'smooth'.  
replicates             number of bootstrap samples. Default 100



**Details**

Accepted methods are "case", "smooth", "residuals\_hill", "wild\_hill", "residuals\_gnls", "wild\_gnls", "watt\_normal", "watt\_student\_df5"

**See Also**

[toxboot](#)

---

toxbootSet0pts	<i>DEPRECATED</i>
----------------	-------------------

---

**Description**

toxbootSet0tps is deprecated. See toxbootConf.

**Usage**

```
toxbootSet0pts(mongo_host = NULL, DBNS = NULL, user = NULL, pass = NULL,
  db = NULL)
```

**Arguments**

mongo_host	Character of length 1, database IP address
DBNS	Character of length 1, the collection on the database to read from and write to
user	Character of length 1, username to authenticate
pass	Character of length 1, password that corresponds to username
db	Character of length 1, database where the username and pass are authenticated. Can be an IP address or a url.

---

toxbootWriteMongo	<i>Write bootstrap results to mongoDB</i>
-------------------	---

---

**Description**

toxbootWriteMongo writes results to a mongo database

**Usage**

```
toxbootWriteMongo(dat, this_m4id, boot_method, replicates, logc_vect, starttime,
  concvals, datchemresult, dat-sample)
```

**Arguments**

dat	A data.table. Required columns are: logc: numeric, contains concentrations resp: numeric, normalized response values paired with concentrations m3id: numeric, value unique to each row corresponding to an individual concentration and response m4id: numeric, value unique to an aeid/spid pair. Multiple m3ids per m4id aeid: numeric, assay id spid: character, sample ID bmad: numeric, baseline mad. Unique to an aeid.
this_m4id	numeric length 1, m4id to bootstrap. Choice of m4id will determine which rows are selected, and therefore the values of logc, resp, m3id, aeid, spid, and bmad.
boot_method	parameter passed to toxbootReplicates to determine sampling method
replicates	number of bootstrap samples. Default 100
logc_vect	vect
starttime	time
concvals	logical, default is FALSE. If TRUE, dose response samples written to the database as well.
datchemresult	data.table
dat-sample	data.table used if concval = T

**Details**

The fitted results are assembled into a bson object using rmongodb and written to the mongoDB.

**See Also**

[toxboot](#)

# Index

## \*Topic **datasets**

- dat1000, 2
- dat10000, 3
- erl3data, 3
- erl5data, 4
  
- dat1000, 2
- dat10000, 3
- detectCores, 13
  
- erl3data, 3, 4
- erl5data, 4
  
- gnls\_curve, 5
  
- hill\_curve, 5
  
- listTonumeric, 6
  
- mclapply, 13
  
- nullToNA, 7
  
- tcplConf, 16
- tcplFit, 3
- toxboot, 3, 7, 13, 17, 18
- Toxboot configuration functions, 9
- toxbootConf, 8
- toxbootConf (Toxboot configuration functions), 9
- toxbootConfList, 8
- toxbootConfList (Toxboot configuration functions), 9
- toxbootConfLoad (Toxboot configuration functions), 9
- toxbootConfReset (Toxboot configuration functions), 9
- toxbootConfSave (Toxboot configuration functions), 9
- toxbootConnectMongo, 8, 10
- toxbootGetMongoFields, 10
- toxbootGetMySQLFields, 11
- toxbootGetOpts, 12
- toxbootHitParamCI, 4, 12
- toxbootmc, 8, 13
- toxbootMongoCount, 14
- toxbootMysqlCreateTable, 14
- toxbootProjectionBuild, 15
- toxbootQueryBuild, 15
- toxbootQueryToxCast, 3, 13, 16
- toxbootReplicates, 8, 16
- toxbootSetOpts, 17
- toxbootWriteMongo, 17