Package 'ELISAtools'

April 6, 2025

Title ELISA Data Analysis with Batch Correction

Version 0.1.8

Description To run data analysis for enzyme-link immunosorbent assays (ELISAs).
Either the five- or four-parameter logistic model will be fitted for data of single ELISA.
Moreover, the batch effect correction/normalization will be car-
ried out, when there are more than one batches of ELISAs.
Feng (2018) <doi:10.1101/483800>.

Depends R (>= 4.0.0), R2HTML (>= 2.3.2), stringi (>= 1.1.7),
minpack.lm (>= 1.2-1)

Imports methods

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Collate 'BatchCorrection.R' 'ELISAplate.R' 'ELISAtools_IO.R'
'ELISAtools.R' 'Regression.R'

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Feng Feng [aut, cre]

Maintainer Feng Feng <ffeng@BU.edu>

Repository CRAN

Date/Publication 2025-04-06 16:50:01 UTC

Contents

ELISAtools-package   *ELISA data analysis with batch correction*

Description

An R package to run ELISA data analysis with the ability to do batch correction/normalization

Details

This package is developed to run analysis of ELISA data. First, the calibration data are used to fit either the five- or four-parameter logistic model. Then the fitted model is used to predict the concentrations of unknown samples. If the batches of calibration data exist, the correction/normalization could be done. The corrected calibration curve are then used for predication.

Please refer to the vignettes to see details.

Author(s)

Maintainer: Feng Feng <ffeng@BU.edu>

References

Feng, et al 2018 doi:10.1101/483800

---

annotate.plate	*S3 method to annotate ELISA plate*

Description

to write annotations for an ELISA plate as an input to guide the functions to read OD values

Usage

```
annotate.plate(
  sample.id,
  sample.prefix,
  sample.suffix,
  num.sample,
  num.std = 8,
  byRow.sample = TRUE,
  byRow.replicates = TRUE,
  replicates.sample = 3,
  replicates.std = 3,
  rows,
  columns,
  std.first = TRUE
)
```

Arguments

sample.id	character vector to specify the names/ids of the samples on the plate. Note, standard/calibration sample ids/names is fixed to be "s1","s2", etc, which are specified by the software and users don't need to privide.
sample.prefix	characters will be added to the beginning of sample names
sample.suffix	characters will be added to the end of sample names
num.sample	numeric number of samples to write
num.std	numeric number of standards
byRow.sample	boolean indicate whether to write sample names horizontally by row (TURE) or vertically by column (FALSE)
byRow.replicates	
	boolean indicate whether to write sample replicates horizontally by row (TURE) or vertically by column (FALSE)
replicates.sample	
	numeric number of replicates for each sample
replicates.std	numeric number of replicates for each standards
rows	numeric vector to specify which rows to be included in the annotation
columns	numeric vector to specify which columns to be included in the annotation
std.first	boolean to indicate whether to write standards first or the samples first.

Details

Based on the input to quickly write the annotations for ELISA plate. The output is in a 96-well format and will be used to giude the reading of OD plates. This way only a nxm dataframe can be used. To write non-regular annotation, you have to do it mannually.

Value

a dataframe holding the annotations for the plate.

---

avoidZero   *Get rid of zeros in a numeric vector*

Description

Get rid of zeros in a numeric vector before taking the logarithm of them. We basically replace the "zeros" with a negligible small value in order to avoid NaN upon the log-transformation.

Usage

```
avoidZero(x, fac = 10)
```

Arguments

x   numeric values as input

fac   numeric a factor of scale in order to get a "small" value to replace zeros

Value

a vector of value with zeros replaced.

---

combineData   *Combine elisa_batch data*

Description

Combine the two lists of elisa_batch data.

Usage

```
combineData(eb1, eb2)
```

Arguments

eb1   list of elisa_batch data

eb2   list of elisa_batch data

Details

When combining, we not only concatenate the two data sets, but also combine batches, meaning the two batches with same batch ID will be merged into one. We will not merge the runs. Therefore, same batch from different list will always have different runs. It is the user's responsibility to make sure the runs are different.

Value

a list of elisa_batch data combining the two input lists (sorted);

See Also

[elisa_batch-class loadData saveDB](#)

Examples

```
#R code to run 5-parameter logistic regression on ELISA data
#load the library
library(ELISAtools)

#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

batches<-loadData(file.path(dir_file,"design.txt"))

#make a guess for the parameters, the other two parameters a and d
#will be estimated based on data.
model<-"5pl"
pars<-c(7.2,0.5, 0.015) #5pl inits
names(pars)<-c("xmid", "scal", "g")

#do fitting. model will be written into data set.
batches<-runFit(pars=pars,  batches=batches, refBatch.ID=1, model=model  )

#call to do predications based on the model.
batches<-predictAll(batches);

batches.old<-batches;

#now suppose want to join/combine the two batches, old and new
batches.com<-combineData(batches.old, batches);
```

---

elisa_batch	*Constructor function to build an elisa_batch object*

Description

S3 method as a constructor to build the S4 class object of elisa_batch [elisa_batch-class](#)

Usage

```
elisa_batch(
  batchID = NA_character_,
  desc = NA_character_,
  runs = list(),
  model.fit = list(),
  model.name = NA_character_,
  pars = c(-1),
  num.runs = 1,
  range.ODs = c(-1, -1),
  normFactor = NaN
)
```

Arguments

batchID	character string to specify a batches
desc	character string for the data/experiment information
runs	list of elisa_plates in this run. There could be one or many plates in a run.
model.fit	list intend to contain information for the fitting of nls.lm. But not using it now.
model.name	character string of either the 5pl (5-parameter) or 4pl (4-parameter) logistic function
pars	numeric the actually parameters for the fitting. for example for the 5pl model they are c(a, d,xmid, scal, g).
num.runs	numeric the number of plates in this run.
range.ODs	numeric the min and max ODs
normFactor	numeric the batch effect normalization factor ("S").

Details

S3 method as a constructor to build the S4 class object of elisa_batch [elisa_batch-class](#). Normally this is called to build a empty object with default values and then load the elisa_run data into it

Value

an elisa_batch object

See Also

[nls.lm](#) [elisa_run-class](#) [elisa_plate-class](#) [elisa_batch-class](#)

Examples

```
elisa_batch();
```

---

elisa_batch-class   *S4 class definition of an elisa_batch object*

Description

[elisa_batch-class](#) define the S4 class of an elisa_batch object

Arguments

batchID	characters to specify the batch
runs	list of elisa_run objects
num.runs	numeric the number of elisa_runs in this batch
pars	numeric the actually parameters for the fitting. for example for the 5pl they are c(a, d,xmid, scal, g).
model.fit	list intend to contain information for the fitting of nls.lm. But not using it now.
model.name	characters of either the 5pl (5-parameter) or 4pl (4-parameter) logistic function
range.ODs	numeric the min and max ODs
normFactor	numeric the batch normalization factor ("S").

Details

defining the S4 class of the elisa_batch. This holds the data for elisa batch. It contains one or many elisa_run objects.

Slots

batchID character
desc character
runs list
num.runs numeric
pars numeric
model.fit list
model.name character
range.ODs numeic
normFactor numeric

See Also

[nls.lm](#) [elisa_plate-class](#) [elisa_run-class](#) [elisa_batch-class](#)

Examples

elisa_batch();

---

elisa_plate   *Constructor function to build an elisa_plate object*

Description

S3 method as a constructor to build the S4 class object of the elisa_plate elisa_plate

Usage

```
elisa_plate(
  batchID = NA_character_,
  expID = NA_character_,
  desc = NA_character_,
  data.std = data.frame(),
  mdata.std = data.frame(),
  data.unknown = data.frame(),
  mdata.unknown = data.frame(),
  normFactor = NaN,
  range.ODs = c(-1, -1)
)
```

Arguments

batchID	characters to specify the batch
expID	characters to specify experiment or plate ID
desc	characters for the data/experiment information
data.std	data.frame for standard curve data
mdata.std	data.frame containing the mean ODs and concentration of the calibration data
data.unknown	data.frame for data of samples with unknown concentration fitted with either four- or five-parameter logistic function.
mdata.unknown	data.frame containing the mean ODs and concentration by sample IDs.
normFactor	numeric the correction factor for batch effects.
range.ODs	numeric the min and max ODs in the plate.

Details

S3 method as a constructor to build the S4 class object of elisa_plate elisa_plate. Normally this is called to build an empty object with default values and then load data into it by calling loadData loadData or load.ODs load.ODs

Value

an elisa_plate object

See Also

nls.lm loadData elisa_plate load.ODs

Examples

```
elisa_plate();
```

---

elisa_plate-class   *S4 class definition of an elisa_plate object*

Description

elisa_plate define the S4 class of an elisa_plate object

Arguments

batchID	characters to specify the batch
expID	characters to specify experiment or plate ID
desc	characters for the data/experiment information
data.std	data.frame for standard curve data
data.unknown	data.frame containing data for samples with unknown concentration
normFactor	numeric the correction factor for batch effects ("S").
mdata.unknown	data.frame containing the mean ODs and concentration by sample IDs.
mdata.std	data.frame containing the mean ODs and concentrations for standard data

Details

defining the S4 class of the elisa_plate object. This is the data structure to hold the elisa_plate Data. It contains different slots for holding both standard and unknown data. It also defines the regression model and the correction parameter for the batch effects.
Note: we assume each plate has its own standard curve.

Slots

batchID character

expID character

data.std data.frame

data.unknown data.frame

normFactor numeric

desc character

range.ODs numeric

mdata.unknown data.frame

mdata.std data.frame

See Also

[nls.lm](#)

Examples

```
elisa_plate();
```

---

elisa_run   *Constructor function to build an elisa_run object*

Description

S3 method as a constructor to build the S4 class object of elisa_run [elisa_run-class](#)

Usage

```
elisa_run(
  batchID = NA_character_,
  desc = NA_character_,
  plates = list(),
  num.plates = 1,
  date = NA_character_,
  range.ODs = c(-1, -1)
)
```

Arguments

batchID	characters to specify the batch
desc	characters for data/experiment information
plates	list of elisa_plates in this run. could be one or many
num.plates	numeric the number of plates in this run.
date	charaters the date to run ELISA measurements
range.ODs	numeric the range of ODs for the measurements

Details

S3 method as a constructor to build the S4 class object of elisa_run [elisa_run-class](#). Normally this is called to build a empty object with default values and then load the elisa_run data into. #

Value

an elisa_run object

See Also

[nls.lm](#) [elisa_run-class](#) [elisa_plate-class](#)

Examples

```
elisa_run();
```

---

elisa_run-class   *S4 class definition of an elisa_run object*

Description

[elisa_run-class](#) defines the S4 class of elisa_run

Arguments

batchID	characters to specify the batch
desc	characters for the data/experiment information
plates	list of elisa_plates
num.plates	numeric the number of plates in this run
date	characters for the date of running the ELISA measurements

Details

defining the S4 class of the elisa_run object. This is list to hold the data for each elisa run. It contains one or many elisa plate objects.

Slots

batchID character

plates list

desc character

num.plates numeric

date character

range.ODs numeric

See Also

[nls.lm](#)

Examples

```
elisa_run();
```

---

f5pl *The five-parameter logistic function*

Description

read in the paramters and the independent variable value(s), and then return the 5pl function value(s). For the 4pl model, set g to be 1.

Usage

```
f5pl(pars, x)
```

Arguments

pars   numeric the parameters of the 5pl (or 4pl). It has the following content: [a, d, xmid, scal, g].

x   numeric the log-transformed x value(s).

Details

The function has the following form
$f(x)=a+(d-a)/((1+\exp((xmid-x)/scal))^g)$

Value

the 5pl function value(s).

---

inv.f5pl *The inverse of the 5-parameter logistic function*

Description

The inverse function of the 5pl. Set the value of g to be 1, if the 4pl is of interest.

Usage

```
inv.f5pl(pars, y)
```

Arguments

pars   the parameters of the function. It has the following content: [a, d, xmid, scale, g].

y   the value to be reverse calculated.

Value

the value of the reverse function

---

load.ODs   *Function to load OD data*

Description

Generic function to load OD data into an elisa_plate object

Usage

```
load.ODs(x, plate.header, plate.data, plate.blank, annotation, ....)

## S4 method for signature 'elisa_plate'
load.ODs(x, plate.header, plate.data, plate.blank, annotation)
```

Arguments

x	the elisa_plate object to load data into
plate.header	characters
plate.data	data.frame OD readings
plate.blank	data.frame OD blank readings
annotation	data.frame annotation to guide reading.
....	other parameters that will help reading data.

Details

It loads OD data into an elisa_plate object. The data usually read int from design file, annotation file, OD file and standard concentration data.

Methods (by class)

- `load.ODs(elisa_plate)`: to load ODs to an elisa_plate object

---

loadData   *Read data according to the design file*

Description

Read the design file and then load the data according to the information in the design file.

Usage

```
loadData(design.file)
```

Arguments

design.file   characters to specify the path and the file name of the design file.

Details

The design file contains all the information necessary to read data. It has the following format

ExpID	FileName	Batch	Num_Plate	Date	AnnotationFile	Std_Conc	Dir_Annotation	Dir_StdConc
Exp1	file1.txt	Batch1	1	9/18/2009	annote.txt	stdConc.txt		
Exp2	file2.txt	Batch2	2	9/18/2009	annote.txt	stdConc.txt		

The return data is a list of batches ([elisa_batch-class](#)), which are made of one or many elisa runs([elisa_run-class](#)) . The run could contain one or many elisa plates ([elisa_plate-class](#)) with data or annotation of each plate.

```
list   |
       |  –batch1   |
       |            |  –run1   |
       |            |          |  –plate1
       |            |          |  –plate2
       |  –batch2   |
       |            |
```

Value

a list of batches holding different runs of elisa, which could contain one or many elisa_plates with data and annotations for each plate.

See Also

[elisa_batch-class](#) [elisa_plate-class](#) [elisa_run-class](#)

Examples

```
file.dir<-system.file("extdata", package="ELISAtools")
loadData(file.path(file.dir,"design.txt"));
```

loadDB   *Read the saved elisa_batch data*

Description

Load the serialized elisa_batch data from disk.

Usage

```
loadDB(db)
```

Arguments

db   characters to specify the path and file name the elisa data file.

Details

Here we deserialize elisa_batch data by wrapping the readRds() function call. The serialized elisa_batch data are assumed to have been correctly analyzed. We will print a summary for what has been read.

Value

a list of batches holding different runs of elisa, which could contain one or many elisa_plates with data and annotations for each plate.

See Also

[elisa_batch-class](#) [loadData](#) [saveDB](#)

Examples

```
#R code to run 5-parameter logistic regression on ELISA data
#load the library
library(ELISAtools)

#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

batches<-loadData(file.path(dir_file,"design.txt"))

#make a guess for the parameters, the other two parameters a and d
#will be estimated based on data.
model<-"5pl"
pars<-c(7.2,0.5, 0.015) #5pl inits
names(pars)<-c("xmid", "scal", "g")


#do fitting. model will be written into data set.
```

```
batches<-runFit(pars=pars,  batches=batches, refBatch.ID=1, model=model  )

#now call to do predications based on the model.
batches<-predictAll(batches);

#now saving the data.
saveDB(batches, file.path(tempdir(),"elisa_tool1.rds"));

loadDB(file.path(tempdir(),"elisa_tool1.rds"));
```

---

plotAlignData   *Plot all batch data together*

Description

Plot the batch data together for visualization.

Usage

```
plotAlignData(batches, graph.file = NULL)
```

Arguments

batches	list of batch data objects either raw or analyzed data.
graph.file	characters as the output graph file name. If specified, a SVG (*.svg) graph will be saved to the disk. Otherwise, the graph will be send to the stdout.

Details

If the data has been analysed, a fitted line will be drawn too. If there are more than one batches, each batch will be plotted with different color and different synmbols. Different batches will also be shifted/adjusted based on their "S" factor, and one single fitted line (based on the "reference" batch) will be plotted.

Value

characters which specify the graph file name, if graph.file is specified. NULL otherwise.

Examples

```
#load the library
library(ELISAtools)

#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

#load the data
batches<-loadData(file.path(dir_file,"design.txt"))
```

```
#plot the raw batch data together
plotAlignData(batches);
```

---

plotBatchData   *Plot ELISA data for one batch*

Description

Plot the individual batch data for visualization.

Usage

```
plotBatchData(batch, graph.file = NULL)
```

Arguments

batch	batch data objects with either raw or analyzed data.
graph.file	characters as the output graph file name. If specified, a SVG (*.svg) graph will be saved to the disk. Otherwise, the graph will be send to the stdout.

Details

If the data has been analysed, a fitted line will be drawn too.

Value

characters which is the graph file name, if graph.file is specified. NULL otherwise.

Examples

```
#load the library
library(ELISAtools)

#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

#load the data
batches<-loadData(file.path(dir_file,"design.txt"))

#plot the raw batch 1 data
plotBatchData(batches[[1]]);
```

---

predictAll	*Predict the concentration of samples based on fitting*

Description

Based on the 5pl or 4pl regression, predict the concentration of of unknown samples. Assume the regression has been accomplished.

Usage

```
predictAll(batches)
```

Arguments

batches   list of elisa_batch objects containing both the raw data and the fitted regression model.

Details

The input data structure contains both the data (ODs) and the fitted regression model. The estimation of unknonw concentration based on the ODs and the standard curve of each plate. The batch effects are corrected/normalized and the corrected concentrations also are also written into the batch data structure, if there are more than one batches in the data.

Value

The same list of elisa_batch with estimated sample concentrations based on ODs and the fitted regression model. The estimated concentrations normalized/corrected between different batches are also calculated and recorded.

References

Feng 2018 doi:10.1101/483800

See Also

elisa_batch elisa_run elisa_plate

---

prepareInitsLM   *Prepare initial values for fitting shifts*

Description

Generate the initial values for fitting shifts with a model of the 5-parameter logistic function.

Usage

```
prepareInitsLM(batches, ref.batch = 1)
```

Arguments

batches   list of elisa_batch data

ref.batch   numeric the index of the reference batch. It is 1 by default.

Details

This is a more complicated way to prepare the initials for shifting.

Value

a data list contain the following elements,

- inits, the initial values for the standard curves of all the plates

- ref.ibatch, the index of the reference batch

  This one is specified by the input ref.batch.
- ref.irun,the index of the reference run

- ref.index,the index of the reference line in the order of the inits vector

---

prepareRegInput   *Prepare the input for regressoin*

Description

Prepare the input data to feed in the fitting.

Usage

```
prepareRegInput(batches)
```

Arguments

batches   list of the ELISA data arranged in batches. Each element of list contains a
                   batch (list) data, and each batch contains one or many the elisa_run objects
                   [elisa_plate](#)

Value

list of data that will feed in to do regression.

See Also

[elisa_plate](#) [elisa_batch](#)[elisa_run](#)

---

rangeOD   *Get the OD ranges (min/max)*

Description

Going through the list of batches to get the OD range (min and max)

Usage

```
rangeOD(batches)
```

Arguments

batches   list of batches data

---

read.annotation   *Read the annotation of single ELISA plate*

Description

Parse the annotations for one single ELISA plate from a section of a file and output the annotations
for standard and unknown separately.

Usage

```
read.annotation(annotation, std.conc)
```

Arguments

annotation   characters to specify the path and name of the annotation file
std.conc   data.frame containing standard concentration data. Only first two columns are
                   used with first one to be the standard IDs and second the concentrations.

Details

The annotation file may contain annotations for more than one plate. Each plate is marked by "Plate: plate 1..." and "~End". This function is fed in with the content for each section and we do actually parsing in here. Store the annotations into data frame. It also parse the standard concentration and include this information in the data frame. For each section,we expect the following format

	1	2	3	4	...
C	s1	s1	sample1	sample1	...
D	s2	s2	sample2	sample	...
...	...	...	...	...	...

In addition, the row name and column names indicate the the plate row and column indices. As input, the stardard and unknown are returned separately in two tables.

Value

a list of data.frames holding the annotations for the plate.

Examples

```
#get example annotation file path from the system folder
fileName<-system.file("extdata", "annote_single.txt", package="ELISAtools")
#prepare the standard concentration file.
std.conc<-data.frame(id=c("s1","s2","s3","s4","s5","s6"), conc=c(1:6))
#read the data as a data frame.
ann<-read.table(fileName, header=TRUE,  sep="\t", stringsAsFactors=FALSE)

#call to do the reading.
```

---

read.annotations   *Read the annotations of plates*

Description

Parse annotations for multiple ELISA plates from files, one annotation file and one standard concentration file, and output the annotations for standard and unknown separately.

Usage

```
read.annotations(
  annotation,
  std.conc,
  dir.annotation,
  dir.stdConc,
  num.plate = 1
)
```

Arguments

`annotation`	characters to specify the path and name of the annotation file
`std.conc`	characters to specify the standard concentration file.
`dir.annotation`	characters specifying the file to the annotatoin file.
`dir.stdConc`	characters specifying the path to the annotatoin file.
`num.plate`	numeric indicating the number of plates in the annotation files.

Details

The annotation file may contain annotations for more than one plate. Each plate is marked by "Plate: plate 1..." and "~End". This function parses each section in both annotation file and standard concentration file. Then passes the section on to do the parsing. For each section,we expect the following format

	1	2	3	4	
C	s1	s1	sample1	sample1	...
D	s2	s2	sample2	sample	...
...	...	...	...	...	...

Value

a list of annotations for elisa plates.

Examples

```
#get example annotation file path from the system folder
ann<-system.file("extdata", "annote.txt", package="ELISAtools")
std.conc<-system.file("extdata", "stdConc.txt", package="ELISAtools")

#read them in and there are 2 plates.
read.annotations(annotation=ann,  std.conc=std.conc, num.plate=2)
```

---

read.plate	*Read the single ELISA OD plate*

Description

Read the individual ELISA plate to parse the ODs.

Usage

```
read.plate(ODs, annotation, batchID, expID)
```

Arguments

ODs	characters containing data of ODs for one plate
annotation	list of data containing annotations of the plate
batchID	characters specifying the batchID read from the design file
expID	characters specifying the expID or plateID read from the design file

Details

The input is a text file imported from the sdf file. We only read the first section with both the OD and blank file. The OD data are read in according to the annotation file.

Value

an object of elisa_plate holding data and annotations for a single plate.

---

read.plates   *Read the ELISA OD files*

Description

Read the ELISA OD file to parse the ODs.

Usage

```
read.plates(
  fileName,
  annotations,
  num.plate = 1,
  batchID,
  expID,
  date = NA_character_
)
```

Arguments

fileName	characters containing file name of OD data
annotations	list of data containing annotations of the plates
num.plate	numeric number of OD plates in the OD file.
batchID	characters specify the batchID read from the design file
expID	characters specify the expID or plateID read from the design file
date	characters the date running the ELISA exps.

Details

The input is a text file imported from the sdf file. The file may contain multiple plates of OD. We will parse each file section and then read them according to the annotation to load the data. We assume for each file the data are for the same batch and experiment. If otherwise, please split the file into different ones.

Value

an object of elisa_run holding data and annotations for one or multiple plates.

Examples

```
#get example annotation file path from the system folder
ann<-system.file("extdata", "annote.txt", package="ELISAtools")
std.conc<-system.file("extdata", "stdConc.txt", package="ELISAtools")

#read them in and there are 2 plates.
annotations<-read.annotations(annotation=ann,  std.conc=std.conc, num.plate=2)

#now start reading the OD plate file
fileName <-system.file("extdata", "Assay_3_and_4.txt", package="ELISAtools")
plates<-read.plates(fileName, annotations=annotations, num.plate=2, batchID="b1", expID="e1")
```

---

reportHtml   *Report ELISA data in HTML format.*

Description

Writting the ELISA analysis results by batch in HTML format.

Usage

```
reportHtml(batches, file.name = "report", file.dir = ".", desc = "")
```

Arguments

batches	list of elisa batch data objects. The data can be raw or after analyzed and batch-corrected.
file.name	character string denoting the report file. The file will be written in HTML format.
file.dir	character string denoting the directory to save the report.
desc	character string describing the project and experiment. Will be written into the report.

Value

the function returns NULL. But it will save the html report to the disk. Therefore, it is IMPORTANT to specify a directory you have write permission to run this function.

See Also

[elisa_batch](#) [elisa_run](#) [elisa_plate](#)

Examples

```
#R code to run 5-parameter logistic regression on ELISA data
#load the library
library(ELISAtools)

##
#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

batches<-loadData(file.path(dir_file,"design.txt"))


#----IMPORTANT-----
#please make sure you have the write permission to save the html report
reportHtml(batches,file.dir=tempdir());
```

---

runFit   *Fit 5- or 4-parameter logistic function*

Description

Fit 5- or 4-parameter logistic function to estimate the parameters by pooling the standard curves from all batches

Usage

```
runFit(pars, a, d, batches, refBatch.ID = 1, model = c("5pl", "4pl"))
```

Arguments

pars	numeric vector initial values to estimate the paramters
a	numeric the initial value for a (the lower limit of the function)
d	numeric the initial value for d (the upper limit of the function)
batches	list of the batch data used to fit the model
refBatch.ID	numeric or string indicating the reference batch, by default is set to be the first one.
model	characters to indicate either 5-parameter logistic function (5pl, default one) or 4-parameter logistic (4pl) to be used in the fitting.

Details

In this fitting, we first "guess" the initial values and then estimate the parameters based on 5- or 4-parameter function by shifting every single standard curves towards the reference line. We are reasoning that the intra-batch and inter-batch factors affect the curve similarly by shifting the curve left or right without changing its shapes. So we combine them together to to fit one single reference curve. To model the inter-batch effects, we take the average of the shifts of curves withine each batch, and use it to correct/normalize between different batches.

To summerize, each individual curve has its own shifts, which contains the information about intra- and inter-batch effects. each batch has one batch level shifts (S Factor), which is an average of shifts of curves within its batch and contains information about inter-batch effects. When we try to normalize between batches, we will apply the batch level shift to all the curves within the same batch.

Value

the batch data with the fitted model

References

Feng, et al 2018 doi:10.1101/483800

Examples

```
#R code to run 5-parameter logistic regression on ELISA data
#load the library
library(ELISAtools)

#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

batches<-loadData(file.path(dir_file,"design.txt"))

#make a guess for the parameters, the other two parameters a and d
#will be estimated based on data.
model<-"5pl"
pars<-c(7.2,0.5, 0.015) #5pl inits
names(pars)<-c("xmid", "scal", "g")


#do fitting. model will be written into data set.
batches<-runFit(pars=pars,  batches=batches, refBatch.ID=1, model=model  )
```

---

saveDataText   *Save elisa_batch analysis results*

Description

Save the data analysis results to disk in text format.

Usage

```
saveDataText(batches, file.name)
```

Arguments

batches	list of elisa batch data to be serialized.
file.name	character specifying name of the output file.

Details

The results are written to disk in the text format (tab-delimited) and is easy to be used for other analysis.

Examples

```
#'#R code to run 5-parameter logistic regression on ELISA data
#load the library
library(ELISAtools)

#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

batches<-loadData(file.path(dir_file,"design.txt"))

#make a guess for the parameters, the other two parameters a and d
#will be estimated based on data.
model<-"5pl"
pars<-c(7.2,0.5, 0.015) #5pl inits
names(pars)<-c("xmid", "scal", "g")


#do fitting. model will be written into data set.
batches<-runFit(pars=pars,  batches=batches, refBatch.ID=1, model=model  )

#now call to do predications based on the model.
batches<-predictAll(batches);

#now saving the data in text.
saveDataText(batches, file.path(tempdir(),"elisa_data.txt"));
```

---

saveDB   *Save the elisa_batch data*

Description

Serialize elisa_batch data to disk.

Usage

```
saveDB(batches, db)
```

Arguments

batches   list of elisa batch data to be serialized.

db   character the file name specifying name of the db.

Details

We serialize elisa_batch data by wrapping the saveRds() function call. The serialized elisa_batch data are assumed to have been correctly analyzed. We will print a summary for what has been saved.

See Also

[elisa_batch-class](#) [loadData](#) [saveDB](#)

Examples

```
#R code to run 5-parameter logistic regression on ELISA data
#load the library
library(ELISAtools)

#get file folder
dir_file<-system.file("extdata", package="ELISAtools")

batches<-loadData(file.path(dir_file,"design.txt"))

#make a guess for the parameters, the other two parameters a and d
#will be estimated based on data.
model<-"5pl"
pars<-c(7.2,0.5, 0.015) #5pl inits
names(pars)<-c("xmid", "scal", "g")


#do fitting. model will be written into data set.
batches<-runFit(pars=pars,  batches=batches, refBatch.ID=1, model=model  )

#now call to do predications based on the model.
batches<-predictAll(batches);

#now saving the data.
saveDB(batches, file.path(tempdir(),"elisa_tool1.rds"));
```

Index