**Short tutorial to use ELISAtools**

Table of Contents

[Installation of ELISAtools R package 1](#_Toc522844669)

[Data input 1](#_Toc522844670)

[Model Fitting and unknown concentration estimation 3](#_Toc522844671)

[Report 3](#_Toc522844672)

[Save/load the batch data in R data set 4](#_Toc522844673)

[Code Example 4](#_Toc522844674)

# Installation of ELISAtools R package

ELISAtools is an R package. The source code are on the github website at

<https://github.com/BULQI/ELISAtools>

To install the code, users need to do the followings:

1. Start the R console and make sure you have the permission to do installation (Under windows system, please start the R software with the administrator’s permission)
2. Do the following steps
   1. install devtools by ‘install.packages("devtools")’
   2. Install the R package by ‘devtools::install\_github("BULQI/ELISAtools")’
3. To load the R package, run

library(ELISAtools)

Another way to install is to download the R package and then unzip to the some folder on the local machine. Finally run the following commands to install,

#install from local using install.packages utility

setwd( "E:\\feng\\LAB\\hg\\ELISA\\" ) #REPLACE THE FOLDER WITH ONE ON YOUR LOCAL MACHINE.

install.packages(c("ELISAtools"),repos=NULL, type="source") #remember to have the administrator’s permission on a Windows system

# Data input

To read the data into software, we rely on 3 other sets of files, design file, annotation file, standard concentration file and OD data file.

1. Standard concentration file: this is a table containing the concentrations for the standards. It contains only two columns, ID and Std.

ID Std

s1 3000

s2 1500

It doesn’t matter how the users name the columns. The software will only read the first two columns. The rest will be ignored. It does matter how the users name the standards. The software expects the standard samples to be names as “sxx”, a lower case letter “s” followed by digits. The digits could be single or many. The IDs in this sample will be cross referenced with the annotation file. So please keep the IDs/names of standards in consistence.

1. Annotation file: this file is the map for ELISA plate. It is a 96-well plate format (8x12 grids). It basically tells the software which OD data is for which samples when reading the plate. It contains the sample names in the plate wells. It accepts samples names with no fixed rules, but does expect standards with a strict format (see above).

1 2 3 4 5 6 7 8 9 10 11

A s1 s1 s1

B s2 s2 s2 TCP-00038-01hr TCP-00038-01hr TCP-00038-01hr TCP-00039-02hr

1. OD data file: this is the actually OD input. It is a exported text file from the original “.sdf” file. It contains the OD readings plus other analysis content done by the plate reader. Only the OD data will be read. The OD data are arranged in a format of 96 well plate. The ODs will be assigned the names by referencing the above annotation file.

#BLOCKS= 4

Plate: Plate1 1.3 TimeFormat Endpoint Absorbance Raw FALSE 1 2 450 620 1 12 96 1 8

Temperature(¡C) A1 A2 A3 A4 A5 A6 A7 A8

24.6 2.2416 2.6971 2.4565 0.0647 0.0611 0.0591 0.0635 0.0592 0.0698 0.0629

24.6 0.0372 0.0363 0.039 0.0384 0.036 0.0352 0.0383 0.0371 0.0386 0.0369

~End

1. Design file: this is the file to be fed directly into the package. The software will read the data correctly by the information contained in this file. It is in a table format. Each row of the table is about one run of ELISA reading, and each reading could be made up of one or multiple ELISA plate. Multiple ELISA readings (runs) become one batch, depending on the standard lot number for example. Each run (row) might have the same or different annotation or standard files. Of course, each run (row) should have different OD files.



Two notes about the input file: 1) Standard, annotation and OD data files are arranged into different sections, each of which is for one elisa plate. The section/plate is denoted by lines starting with “Plate:….” and “~End”. The content within these two lines are read and the rest will be discarded. Each run could have many plates. Please keep the files in consistent about number of plates; 2) it is not important how the files named as long as they are specified correctly.

Please see the example data input files at package installed folder. The install folder could be obtained with the following R commands

system.file("extdata", package="ELISAtools")

To read the input files, please generate the files specified above and then call the command of “loadData”. As an example, the package comes with one set of example data input. To read them, run the following.

dir\_file<-system.file("extdata", package="ELISAtools")

setwd(dir\_file)

batches<-loadData(file.path(dir\_file,"design.txt"))

The above code will read in data for two batches of ELISA experiments, and each contains one or two runs, which include one or two elisa plates.

# Model Fitting and unknown concentration estimation

After the data input, the model fitting can be called to run by runFit() function. Currently, only 5-parameter logistic fitting is implemented. In future, we will add 4-parameter logistic function for fitting. Unknown sample concentration is estimated by calling predictAll() function. This function estimate both the unknown centration based on the standards on the same plate and will also correct for the batch effects. The following code will run the analysis based on the sample input in the package folder.

#make a guess for the parameters, the other two parameters a and d

#will be estimated based on data.

pars<-c(7.2,0.05, 0.015)

names(pars)<-c("xmid", "scal", "g")

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

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

#now call to do predications based on the model.

batches<-predictAll(batches);

# Report

To generate a report, the function “reportHtml” will report the results in a html format and it also include some brief QC for the fitting.

Run the following, (PLEASE DO remember to specify a directory, where you have the write permission, to write the report. The “ELISAtools” package “extdata” directory is most likely not writable for regular users.)

#reporting.

reportHtml(batches)

An html report will be generated like



A text format file is also saved to the disk. It contains the elisa data analysis results like below,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Example Output for Prothrombin ELISA Solution |  |  |  |  |  |  |
| Date and Time: | (Date of Analysis here) |  |  |  |  |  |
| Shift Parameter for Run/Plate: | (Shift parameter here) |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Sample ID or Control | Raw O.D. 1 | Raw O.D. 2 | Raw O.D. 3 | O.D. Avg. | Concentration (Uncorrected) | Concentration (Corrected) |
| (Sample ID 1 here) | (Triplicate 1) | (Triplicate 2) | (Triplicate 3) | (=Average of B6,C6,D6) | (=Uncorrected concentration) | (=Corrected Concentration) |
| (Sample ID 2 here) |  |  |  |  |  |  |
| (Sample ID 3 here) |  |  |  |  |  |  |
| (Sample ID 4 here) |  |  |  |  |  |  |

The file has the same name as the html report, but a “txt” suffix.

# Save/load the batch data in R data set

The ELISA data can be saved in R data set and the previous saved data can be load/read to combine with the current data. The example R code are listed as below,

#now saving the combine data.

saveDB(batches, "elisa\_tool1.rds");

batches.old<-loadDB("elisa\_tool1.rds");

We can also combine the two ELISA batch data into one to analyze together and save for the future use.

#now suppose want to join/combine the two batches, old and new

batches.com<-combineData(batches.old, batches);

reportHtml(batches.com)

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

#now call to do predications based on the model.

batches.com<-predictAll(batches.com);

reportHtml(batches.com);

One thing to mention is that we need to rerun the fitting after combining the two batches of data.

# Code Example

The R code named “tutorial.R” is available at the package installation folder.