

# Package: AntibodyTitters (via r-universe)

August 23, 2024

**Version** 0.1.24

**Date** 2022-05-24

**Title** Antibody Titer Analysis of Vaccinated Patients

**Author** Yuuki Mizumoto [aut], Hiroaki Kato [aut, cre]  
(<https://orcid.org/0000-0001-9591-1875>), Takeshi Urano [ctb]

**Maintainer** Hiroaki Kato <hkato@med.shimane-u.ac.jp>

**Depends** openxlsx (>= 4.2.4), R (>= 3.3.0), DescTools (>= 0.99.43)

**Description** Visualization of antibody titer scores is valuable for examination of vaccination effects. 'AntibodyTitters' visualizes antibody titers of all or selected patients. This package also produces empty excel files in a specified format, in which users can fill in experimental data for visualization. Excel files with toy data can also be produced, so that users can see how it is visualized before obtaining real data. The data should contain titer scores at pre-vaccination, after-1st shot, after-2nd shot, and at least one additional sampling points. Patients with missing values can be included. The first two sampling points (pre-vaccination and after-1st shot) will be plotted discretely, whereas those following will be plotted on a continuous time scale that starts from the day of second shot. Half-life of titer can also be calculated for each pair of sampling points.

**License** GPL-3

**NeedsCompilation** no

**Type** Package

**Date/Publication** 2022-05-24 04:10:02 UTC

**Repository** <https://hkatomed.r-universe.dev>

**RemoteUrl** <https://github.com/cran/AntibodyTitters>

**RemoteRef** HEAD

**RemoteSha** c180717f9d78e1cf5b5332ba0c11abd69c47d3ba

## Contents

AntibodyTiters-package . . . . .	2
emptyABT . . . . .	3
groupAttribABT . . . . .	4
halfLifeABT . . . . .	5
idABT . . . . .	5
plotAllABT . . . . .	6
plotEachABT . . . . .	9
readABT . . . . .	10
toyABT . . . . .	11
<b>Index</b>	<b>12</b>

---

AntibodyTiters-package

*Antibody Titer Analysis of Vaccinated Patients.*

---

## Description

Visualization of antibody titer scores is valuable for examination of vaccination effects. **AntibodyTiters** visualizes antibody titers of all or selected patients. This package also produces empty excel files in a specified format, in which users can fill in experimental data for visualization. Excel files with toy data can also be produced, so that users can see how it is visualized before obtaining real data. The data should contain titer scores at pre-vaccination, after-1st shot, after-2nd shot, and at least one additional sampling points. Patients with missing values can be included. The first two sampling points (pre-vaccination and after-1st shot) will be plotted discretely, whereas those following will be plotted on a continuous time scale that starts from the day of second shot. Half-life of titer can also be calculated for each pair of sampling points.

## Details

Package: AntibodyTiters  
 Type: Package  
 Version: 0.1.24  
 Date: 2022-05-24  
 License: GPL-3

[emptyABT](#): Production of an empty xlsx file in which experimental results should be recorded.

[groupAttribABT](#): Addition of a new attribute that categorizes patients in specified groups to an ABT class object.

[idABT](#): Extracting patient IDs from an ABT class object.

[plotAllABT](#): Plotting antibody titers of All patients.

[plotEachABT](#): Plotting antibody titers of each patient.

**readABT**: Reading an xlsx file and examining the validity of the dataset.

**halfLifeABT**: Calculating half-life from an ABT class object.

**toyABT**: Production of an xlsx file with toy data.

### Author(s)

Yuuki Mizumoto, Hiroaki Kato, Takeshi Urano

Maintainer: Hiroaki Kato <hkato@med.shimane-u.ac.jp>

---

emptyABT	<i>Production of an empty xlsx file in which experimental results should be recorded.</i>
----------	---

---

### Description

**AntibodyTiters** requires the input xlsx files be in a specific format. This function produces an empty xlsx file in the format.

### Usage

```
emptyABT(fileName = "empty.xlsx", pmax = 7, returnDF = FALSE,
  Attrib = c("Sex", "Age", "VeryLow"),
  attribFactors = list(c("F", "M"), c(18, 80), c(TRUE, FALSE)))
```

### Arguments

fileName	a file name to be produced.
pmax	an integer ( $\geq 4$ & $< 18$ ) for specifying the maximum sampling point.
returnDF	a logical value indicating whether a data.frame is returned instead of being stored as an xlsx file.
Attrib	a character vector for the names of Attrib.
attribFactors	a list determining the values of Attrib. If the names of Attrib contained "Age", two numbers provided here are used as lower and upper limits.

### Value

When the argument returnDF is FALSE (default), an xlsx file will be stored in the current directory. When returnDF is TRUE, a data.frame will be returned.

### Examples

```
# emptyABT(fileName = "empty.xlsx", pmax = 7, returnDF = FALSE,
# Attrib = c("Sex", "Age", "VeryLow"),
# attribFactors = list(c("F", "M"), c(18, 80), c(TRUE, FALSE)))
# The above command will create an empty excel file named empty.xlsx
# in the current directory.
```

---

groupAttribABT	<i>Grouping of patients by scores.</i>
----------------	--

---

## Description

Addition of a new attribute that categorizes patients in specified groups to an ABT class object.

## Usage

```
groupAttribABT(objName = "inData", sourceAttrib = "Age",
  newAttribName = "AgeGroup", groupNames = c("young", "middle", "elderly"),
  groupLimmits = list(c(0, 39), c(40, 64), c(65, 200)))
```

## Arguments

objName	a character specifying the name of an ABT class object produced by readABT.
sourceAttrib	a character specifying the name of source attribute. The contents of this attribute must be integer of numeric.
newAttribName	a character specifying the name of the new attribute.
groupNames	a character vector specifying the names of categories.
groupLimmits	a list of two value integers specifying the lower and upper limmits of scores. The order of values in this list must correspond to the order of values in groupNames

## Value

an ABT class object

## Examples

```
inData <- readABT(fileName = system.file("extdata", "toy.xlsx", package="AntibodyTiters"))
inDataA <- groupAttribABT(objName = "inData", sourceAttrib = "Age",
  newAttribName = "AgeGroup", groupNames = c("young", "middle", "elderly"),
  groupLimmits = list(c(0, 39), c(40, 64), c(65, 200)))
plotAllABT("inDataA", lineAttrib = "AgeGroup", addPoints = TRUE, savePDF = FALSE,
  orderOfCategories = c("young", "middle", "elderly"), type = "weeks")
plotAllABT("inDataA", lineAttrib = "AgeGroup", addPoints = TRUE, savePDF = FALSE,
  orderOfCategories = c("young", "middle", "elderly"), type = "M1", geometricMean = TRUE)
```

---

halfLifeABT	<i>Calculating half-life from an ABT class object.</i>
-------------	--

---

**Description**

This function calculates  $T_{1/2}$  for each set of sampling points.

**Usage**

```
halfLifeABT(x, output = "list", OutFileName = "Thalf.xlsx")
```

**Arguments**

x	an ABT class object.
output	a character specifying the type of output; "list", "data.frame" or "xlsx".
OutFileName	a character specifying the file name to be stored in the current directory when output = "xlsx".

**Value**

a list, a data.frame or an xlsx file depending on the output argument. The list object (produced with output="list") contains data.frame objects for each patients, whereas the data.frame object (produced with output="data.frame") is a combined form derived from the list (simply produced with rbind). The data.frames contain columns named ID (patient ID), START (starting point), END (end point), SPAN (t), MID (mid point), C0 (initial titer), Ct (titer at t) and Thalf ( $T_{1/2}$ ).

**Examples**

```
inData <- readABT(fileName = system.file("extdata","toy.xlsx",package="AntibodyTiters"))
inDataVL <- subset(inData, VeryLow == TRUE)
halfLifeABT(inDataVL, output = "data.frame")
```

---

idABT	<i>Extracting patient IDs from an ABT class object.</i>
-------	---

---

**Description**

With this function, patient IDs in the ABT class object can be extracted.

**Usage**

```
idABT(objName = "inData")
```

**Arguments**

objName	a character specifying the name of an ABT class object produced by readABT.
---------	---

**Value**

a character vector

**Examples**

```
inData <- readABT(fileName = system.file("extdata", "toy.xlsx", package="AntibodyTiters"))
idABT(objName = "inData")[1:6]
```

---

plotAllABT

*Plotting antibody titers of All patients.*

---

**Description**

This function plots antibody titers of all the patients in the ABT class object produced by [readABT](#).

**Usage**

```
plotAllABT(objName = "inData", prefix = "",
  dayStart = as.integer(NA), dayEnd = as.integer(NA),
  type = "weeks", rainbow = FALSE, ylab = "Titer (AU/ml)", savePDF = FALSE,
  alphaFactor = 10, lwd = 2, lineAttrib = "", addPoints = FALSE,
  orderOfCategories = "", lowessSmooth = FALSE, geometricMean = FALSE,
  lineForPre = FALSE, lineFor1st = FALSE, logY = TRUE, PDFwidth = 8, PDFheight = 5,
  main = NULL, omitPreVac = FALSE, lineColDark = FALSE)
```

**Arguments**

objName	a character specifying the name of an ABT class object produced by readABT.
prefix	a character to be attached to the file name. This is active when savePDF==TRUE.
dayStart	an integer specifying the start day, from which scores are plotted.
dayEnd	an integer specifying the end day, to which scores are plotted.
type	a character determining the time scale; "weeks", "days", "M1", "M2", "M3", "M4", "M6". When "weeks" or "days" is chosen, plotting will be done on a continuous time scale. In contrast, when "M1" or other "M-" is chosen, time scale will be discrete. Bin sizes are one month (M1), two (M2), three (M3), four (M4), and six (M6) months.
rainbow	a logical indicating whether the line color is rainbow.
ylab	a character for the y-axis label.
savePDF	a logical indicating whether a PDF file is stored in the current directory.
alphaFactor	an integer for controlling the transparency of lines.
lwd	an integer for controlling the width of lines.
lineAttrib	a character specifying the name of attribute for which lowess or geometric mean lines are drawn for each group.
addPoints	a logical indicating whether points are plotted. This is active when lineAttrib!="".

orderOfCategories	a character vector specifying the order of categories listed as legends. This is active when <code>lineAttrib!= ""</code> .
lowessSmooth	a logical indicating whether lowess lines are plotted. This is active when <code>lineAttrib!= ""</code> .
geometricMean	a logical indicating whether geometric mean lines are plotted. This is active when <code>lineAttrib!= ""</code> .
lineForPre	a logical indicating whether lines that connects the pre-vaccination sampling point to others are plotted. This is active when <code>lineAttrib!= ""</code> & <code>geometricMean==TRUE</code> .
lineFor1st	a logical indicating whether lines that connects the after-1st shot sampling point to others are plotted. This is active when <code>lineAttrib!= ""</code> & <code>geometricMean==TRUE</code> .
logY	a logical indicating whether Y-axis is in log scale.
PDFwidth	an integer specifying the width of PDF. This is active when <code>savePDF==TRUE</code> .
PDFheight	an integer specifying the height of PDF. This is active when <code>savePDF==TRUE</code> .
main	a character for the main title. When not specified, <code>objName</code> will be used as the title.
omitPreVac	a logical indicating whether pre-vaccination will be omitted.
lineColDark	a logical indicating whether line colors will be darkened.

### Value

No object will be returned. Instead, this function will draw a plot in a graphic device, or save a PDF file in the current directory.

### Examples

```
inData <- readABT(fileName = system.file("extdata", "toy.xlsx", package="AntibodyTiters"))

# logY
plotAllABT(objName = "inData", type = "weeks", rainbow = TRUE, logY = TRUE)
plotAllABT(objName = "inData", type = "weeks", rainbow = TRUE, logY = FALSE)

# lineAttrib, lowessSmooth, addPoints
plotAllABT(objName = "inData", type = "weeks", rainbow = FALSE,
  lineAttrib = "Sex")
plotAllABT(objName = "inData", type = "weeks", rainbow = FALSE,
  lineAttrib = "Sex", lowessSmooth = TRUE)
plotAllABT(objName = "inData", type = "weeks", rainbow = FALSE,
  lineAttrib = "Sex", lowessSmooth = TRUE, addPoints = TRUE)
plotAllABT(objName = "inData", type = "weeks", rainbow = FALSE,
  lineAttrib = "VeryLow", lowessSmooth = TRUE, addPoints = TRUE)

# Discrete mode, geometricMean
plotAllABT(objName = "inData",
  type = "M1", lineAttrib = "Sex", geometricMean = TRUE)
plotAllABT(objName = "inData",
  type = "M3", lineAttrib = "Sex", geometricMean = TRUE)

# alphaFactor, lwd
```

```

plotAllABT(objName = "inData", type = "weeks", rainbow = TRUE)
plotAllABT(objName = "inData", type = "weeks", rainbow = TRUE, alphaFactor = 20)
plotAllABT(objName = "inData", type = "weeks", rainbow = TRUE, lwd = 4)

# dayStart, dayEnd
plotAllABT(objName = "inData", dayStart = as.integer(NA), dayEnd = as.integer(NA),
  type = "weeks", rainbow = TRUE)
plotAllABT(objName = "inData", dayStart = 140, dayEnd = 350,
  type = "weeks", rainbow = TRUE)
plotAllABT(objName = "inData", dayStart = 140, dayEnd = 350,
  type = "days", rainbow = TRUE)

# orderOfCategories, lowessSmooth
inDataA <- groupAttribABT(objName = "inData")
plotAllABT(objName = "inDataA",
  type = "weeks", lineAttrib = "AgeGroup", addPoints = TRUE, lowessSmooth = TRUE)
plotAllABT(objName = "inDataA",
  type = "weeks", lineAttrib = "AgeGroup", addPoints = TRUE, lowessSmooth = TRUE,
  orderOfCategories = c("young", "middle", "elderly"))

# lineForPre, lineFor1st
plotAllABT(objName = "inDataA",
  type = "M1", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), geometricMean = TRUE,
  lineForPre = FALSE, lineFor1st = FALSE)
plotAllABT(objName = "inDataA",
  type = "M1", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), geometricMean = TRUE,
  lineForPre = TRUE, lineFor1st = TRUE)

# omitPreVac (type = "days")
plotAllABT(objName = "inDataA",
  type = "days", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), omitPreVac = FALSE)
plotAllABT(objName = "inDataA",
  type = "days", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), omitPreVac = TRUE)

# omitPreVac (type = "M1")
plotAllABT(objName = "inDataA",
  type = "M1", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), geometricMean = TRUE,
  lineForPre = TRUE, lineFor1st = TRUE, omitPreVac = FALSE)
plotAllABT(objName = "inDataA",
  type = "M1", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), geometricMean = TRUE,
  lineForPre = FALSE, lineFor1st = TRUE, omitPreVac = TRUE)

# lineColDark (type = "days")
plotAllABT(objName = "inDataA",
  type = "days", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), lineColDark = FALSE)
plotAllABT(objName = "inDataA",

```



```

type = "days", lineAttrib = "AgeGroup", addPoints = TRUE,
orderOfCategories = c("young", "middle", "elderly"), lineColDark = TRUE)

# lineColDark (type = "M1")
plotAllABT(objName = "inDataA",
  type = "M1", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), geometricMean = TRUE,
  lineForPre = TRUE, lineFor1st = TRUE, lineColDark = FALSE)
plotAllABT(objName = "inDataA",
  type = "M1", lineAttrib = "AgeGroup", addPoints = TRUE,
  orderOfCategories = c("young", "middle", "elderly"), geometricMean = TRUE,
  lineForPre = TRUE, lineFor1st = TRUE, lineColDark = TRUE)

```

plotEachABT

*Plotting antibody titers of each patient.***Description**

This function plots antibody titers of one specified patient in the ABT class object produced by [readABT](#).

**Usage**

```

plotEachABT(patientID = "patient A", objName = "inData",
  prefix = "", dayStart = as.integer(NA), dayEnd = as.integer(NA),
  type = "weeks", ylab = "Titer (AU/ml)", savePDF = FALSE, addPoints = FALSE, lwd = 2,
  logY = TRUE, PDFwidth = 8, PDFheight = 5, main = NULL)

```

**Arguments**

patientID	a character specifying the patient to be plotted.
objName	a character specifying the name of an ABT class object produced by readABT.
prefix	a character to be attached to the file name. This is active when savePDF==TRUE.
dayStart	an integer specifying the start day, from which scores are plotted.
dayEnd	an integer specifying the end day, to which scores are plotted.
type	a character determining the time scale; "weeks" or "days". There is no discrete mode (e.g. M1) for this function.
ylab	a character for the y-axis label.
savePDF	a logical indicating whether a PDF file is stored in the current directory.
addPoints	a logical indicating whether points are added.
lwd	an integer for controlling the width of lines.
logY	a logical indicating whether Y-axis is in log scale.
PDFwidth	an integer specifying the width of PDF. This is active when savePDF==TRUE.
PDFheight	an integer specifying the height of PDF. This is active when savePDF==TRUE.
main	a character for the main title. When not specified, patientID and objName will be used as the title.

**Value**

No object will be returned. Instead, this function will draw a plot in a graphic device, or save a PDF file in the current directory.

**Examples**

```
inData <- readABT(fileName = system.file("extdata", "toy.xlsx", package="AntibodyTiters"))
plotEachABT(patientID = "patient 2", objName = "inData",
  dayStart = as.integer(NA), dayEnd = as.integer(NA), type = "weeks", savePDF = FALSE)
plotEachABT(patientID = "patient 2", objName = "inData",
  dayStart = as.integer(NA), dayEnd = as.integer(NA), type = "weeks", savePDF = FALSE,
  lwd = 2)
plotEachABT(patientID = "patient 2", objName = "inData",
  dayStart = as.integer(NA), dayEnd = as.integer(NA), type = "weeks", savePDF = FALSE,
  addPoints = TRUE)
plotEachABT(patientID = "patient 2", objName = "inData",
  dayStart = as.integer(NA), dayEnd = as.integer(NA), type = "days", savePDF = FALSE)
plotEachABT(patientID = "patient 2", objName = "inData",
  dayStart = 140, dayEnd = 350, type = "weeks", savePDF = FALSE)
plotEachABT(patientID = "patient 2", objName = "inData",
  dayStart = 140, dayEnd = 350, type = "days", savePDF = FALSE)
```

---

readABT

---

*Reading an xlsx file and examining the validity of the dataset.*


---

**Description**

With this function, a specifically formatted xlsx file can be loaded. After validation of the file, 'pmax' and other key values were examined for the following analysis. The output is an ABT object. An object of this class is a list with four contents named DATA, pmax, longestFromSecond and shortestFromSecond. DATA is a data.frame. The other three are integer objects. These values are used for the plotting functions [plotAllABT](#) and [plotEachABT](#).

**Usage**

```
readABT(fileName = "xxx.xlsx", attribNumeric = "Age")
```

**Arguments**

fileName            a file name to be read.  
attribNumeric      a character vector specifying attribute columns to have numeric values.

**Value**

an ABT-class object: see the Description above for the detail of this class.

**Examples**

```
# inData <- readABT(fileName = "xxx.xlsx")
```

toyABT

*Production of an xlsx file with toy data.***Description**

**AntibodyTiters** requires the input xlsx files be in a specific format. This function produces a toy xlsx file in the format.

**Usage**

```
toyABT(fileName = "toy.xlsx", pmax = 7, patients = 20,
        outsiderPercent = 2, NaPercent = 10,
        Attrib = c("Sex", "Age", "VeryLow"),
        attribFactors = list(c("F", "M"), c(18, 80), c(TRUE, FALSE)))
```

**Arguments**

fileName	a file name to be produced.
pmax	an integer ( $\geq 4$ & $< 18$ ) for specifying the maximum sampling point.
patients	an integer specifying the number of patients.
outsiderPercent	an integer specifying the percentage of patients with outsider values.
NaPercent	an integer specifying the percentage of patients with NA values.
Attrib	a character vector for the names of Attrib.
attribFactors	a list determining the values of Attrib. If the names of Attrib contained "Age", two numbers provided here are used as lower and upper limits.

**Value**

An xlsx file will be stored in the current directory.

**Examples**

```
# toyABT(fileName = "toy.xlsx", pmax = 9, patients = 120,
# outsiderPercent = 2, NaPercent = 10,
# Attrib = c("Sex", "Age", "VeryLow"),
# attribFactors = list(c("F", "M"), c(18, 80), c(TRUE, FALSE)))
# The above command will create an excel file with toy data in the current directory.
```

# Index

## \* **package**

AntibodyTiters-package, [2](#)

AntibodyTiters

(AntibodyTiters-package), [2](#)

AntibodyTiters-package, [2](#)

emptyABT, [2](#), [3](#)

groupAttribABT, [2](#), [4](#)

halfLifeABT, [3](#), [5](#)

idABT, [2](#), [5](#)

plotAllABT, [2](#), [6](#), [10](#)

plotEachABT, [2](#), [9](#), [10](#)

readABT, [3](#), [6](#), [9](#), [10](#)

toyABT, [3](#), [11](#)