

AptaqPCR: Melting curve analysis for diagnostics

Navneet Phogat and Matthias Kohl
Institute of Precision Medicine
Hochschule Furtwangen University, Germany

July 19, 2017

Contents

1	Introduction	1
2	S4 Class aptaqpcr	1
3	Reading the fluorescent data	2
4	Combine the replicates	2
5	Compute the first derivative of the combined replicates	3
6	Plotting the results of the combined replicates, first derivative and negative of the first derivative of the combined replicates	3
7	Two Gaussian fitting to compute the Gaussian parameters	4
8	Compute the 95 percent confidence intervals of the Gaussian means	6
9	Plot the Gaussian Mean (Temperature) vs concentration	6
10	Graphical User Interface	7
11	Report generation of the plots	7

1 Introduction

The package "AptaqPCR" has been designed to analyse the raw fluorescent data of melting curve, generated by the RTqPCR (Real time quantitative polymerase chain reaction) for

aptamer sequences, which involves the target molecules. The package analyses the data on the basis of the two Gaussian fit to get the Gaussian means of the temperature. The Gaussian means are used here to analyse the assay on the basis of the confidence intervals. On the basis of the functions, the package can be divided into following 10 parts: (1) Read in the raw fluorescent data, (2) Combine the technical replicates, (3) Compute the first derivative of the combined replicates with respect to the temperature, (4) Plot the results of the combined replicates, (5) first derivative, (6) negative of the first derivative of the combined replicates, (7) Apply the Gaussian fitting and plot the fitting results, (8) Compute the confidence intervals of Gaussian means, (9) plot the Gaussian means with respect to the concentration, (10) Graphical User Interface, (10) Report generation of the plots through Rmarkdown and Knitr.

User may load the "AptaqPCR" package.

```
> library(AptaqPCR)
```

2 S4 Class aptaqpcr

The S4 Class `aptakandt` is defined to contain the data to control the flow of the data in package "AptaqPCR". All the slots of Class `aptaqpcr` contains the data in the form of an object of Class `data.frame`. The Class `aptaqpcr` has six slots, which includes the slot `initialData` to contain the initial data of fluorescence, slot `meanData` to contain the results of the function `mean.apt`, which are combined replicates, slot `diffData` to contain the results of the function `diff.apt`, which are the first derivative of the combined replicates with respect to the temperature, slot `fitData` to contain the results of implementation of two Gaussian fit, produced as an output of the function `apt.fit`, slot `confMeanData` to contain the results of Gaussian means (`mu1` and `mu2`) of temperature and slot `confData` to contain the results of the confidence intervals of the Gaussian means.

3 Reading the fluorescent data

The `ReadAptaqPCR` function is based on the functions `read.csv` and `read.table` of the package "utils". The function `ReadAptaqPCR` reads in .csv as well as .txt (Tab delimited) files of the raw fluorescent data, generated through the RTqPCR. The format of both types of files is provided in the folder `exData`, which is present in the folder `inst` of the package "AptaqPCR": `multiapt31.csv`, `multiapt31.txt`. The final data will be stored in the new S4 Class `aptaqpcr`. To know about the arguments of the function `ReadAptaqPCR` in detail, please see the documentation file of the function `ReadAptaqPCR`. The following is the example to read in the .csv file:

```
> #library(AptaqPCR) # load the AptaqPCR package
> ##To read the .csv file
```

```

> file1 <- system.file("exData", "multiapt31.csv", package = "AptaqPCR")
> read.datacsv <- ReadAptaqPCR(file1)
> ## To express all the data
> read.datacsv
> ## To express the initial data (all the data read in, except concentration)
> slot(read.datacsv, "initialData")

```

The following is the example to read in the .txt file:

```

> ## To read the .txt file
> file1 <- system.file("exData", "multiapt31.txt", package = "AptaqPCR")
> read.dataatxt <- ReadAptaqPCR(file1, type = ".txt")
> ## To express the initial data (except concentration)
> slot(read.dataatxt, "initialData")

```

4 Combine the replicates

The function `mean.ap` combine the technical replicates (on the basis of mean) of an object of the Class `ap`, which are produced as an output of the function `ReadAptaqPCR`. The examples are shown for the data read in the .csv file format. But, in a similar way, can also be implemented on the data, read in from the .txt file. To know about the arguments of the function in detail, please see the documentation file of the function `mean.ap`. Following is the example to combine the technical replicates:

```

> ###To combine the technical replicates, on the basis of mean
> meandata <- mean.ap(read.datacsv)
> ## To express all the data
> meandata
> ## To express the combined replicates
> slot(meandata, "meanData")

```

5 Compute the first derivative of the combined replicates

The function `diff.ap` compute the first derivative of the combined replicates. The function `diff.ap` acts on an object of the Class `ap`, which is produced as an output of the function `mean.ap` and saves the results in the form of an object of the class `ap`. To know about the arguments of the function in detail, please see the documentation file of the function `diff.ap`. Following is the example to compute the first derivative of the combined replicates:

```

> ## To compute the first derivative of the combined replicates
> diffdata <- diff.ap(meandata)

```

```

> ## To express all the data
> diffdata
> ## To express the first derivative data
> slot(diffdata,"diffData")

```

6 Plotting the results of the combined replicates, first derivative and negative of the first derivative of the combined replicates

The function `apt.plot` acts on an object of the Class `aptaqpcr` and plots the results of the function `mean.apt` to generate the plots of Fluorescence vs Temperature, results of the function `diff.apt` to generate the plots of dF/dT vs Temperature and $-dF/dT$ vs Temperature. To know about the arguments in detail, see the documentation file of the function `apt.plot` of the package `AptaqPCR`. Following is the example to implement the function `apt.plot`:

```

> ## (1) To plot the results of combined replicates
> ##(Fluorescence vs Temperature)
> apt.plot(meandata) ##default plot
> ## to define the names of the plots on the basis of the concentration
> apt.plot(meandata,
+         conc = c("0 pM","30.50 pM","15.25 pM","7.62 pM",
+                 "3.81 pM","1.90 pM","0.95 pM","0.47 pM"))
> ## (2) To plot the results of the first derivative (dF/dT vs Temperature)
> apt.plot(diffdata, plot = "Diff") ##default plot
> ## To define the names of the plots on the basis of the concentration
> apt.plot(diffdata, plot = "Diff",
+         conc = c("0 pM","30.50 pM","15.25 pM","7.62 pM",
+                 "3.81 pM","1.90 pM","0.95 pM","0.47 pM"))
> ## (3) To plot the results of the negative of first derivative
> ##(-dF/dT vs Temperature)
> apt.plot(diffdata, plot = "negDiff") ##default plot
> ## plots on the name of the concentration
> apt.plot(diffdata, plot = "negDiff",
+         conc = c("0 pM","30.50 pM", "15.25 pM","7.62 pM",
+                 "3.81 pM","1.90 pM","0.95 pM","0.47 pM"))
>
> ## One can also define the other parameters like ylabel, xlabel and color,
> ##if user want to have different instead of the default ones.

```

7 Two Gaussian fitting to compute the Gaussian parameters

Function `apt.fit` acts on the object of Class `data.frame` and implements the function `optim` from the package `"stats"` for fitting the data on the basis of two Gaussian fit. The function `apt.fit` saves the fitting results in an object of Class `data.frame`. The function `apt.fit` also produces the fitting plots, which are generated on the basis of the function `ggplot` of the package `ggplot2`. To know about the arguments of the function `apt.fit` in detail, see the documentation file of the function `apt.fit`. Following is the example to produce the results of the Gaussian fitting, which includes, the Gaussian Means (Mean1 and Mean2), amplitudes (A1 and A2) and standard deviations (sd1 and sd2):

```
> ## To express the first derivative data
> data <- slot(diffdata,"diffData")
> temp <- data[, "temp"]
> ## To fit the negative of first derivative data to generate the
> ## fitting results in the form of values (an object of Class data.frame)
> nc <- apt.fit(data[, "nc"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
> b <- apt.fit(data[, "b"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
> c <- apt.fit(data[, "c"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
> d <- apt.fit(data[, "d"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
> e <- apt.fit(data[, "e"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
> f <- apt.fit(data[, "f"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
> g <- apt.fit(data[, "g"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
> h <- apt.fit(data[, "h"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp)
```

Now, the function `apt.fit.value` combines the data of the fitting results, produced as an output of the function `apt.fit` and uses the data to populate an object of the Class `aptaqpcr`. The function `apt.fit.value` acts as a link between the functions `apt.fit` and `mean.conf` as well as between `apt.fit` and `tempconc.plot`. To know about the arguments of the function `apt.fit.value` in detail, please see the documentation file of the function `apt.fit.value`. Following is the example to implement the function `apt.fit.value`:

```
> ## implement the function apt.fit.value to combine all the results
> ## default column names
> aptfitall <- apt.fit.value(nc, b, c, d, e, f, g, h)
> ## To express the combined fitting results
> slot(aptfitall, "fitData")
> ## By defining column names (specific)
> aptfitalls <- apt.fit.value(nc, b, c, d, e, f, g, h,
+ conc = c("0 pM", "30.50 pM", "15.25 pM", "7.62 pM",
+          "3.81 pM", "1.90 pM", "0.95 pM", "0.47 pM"))
```

```
> ##To express the combined fitting results
> slot(aptfitalis,"fitData")
```

Following is the example to plot the results of the Gaussian fitting:

```
> ## To fit the negative of the first derivative data to generate the
> ## fitting results in the form of plots
> nc.plot <- apt.fit(data[, "nc"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "0 pM")
> b.plot <- apt.fit(data[, "b"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "30.50 pM")
> c.plot <- apt.fit(data[, "c"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "15.25 pM")
> d.plot <- apt.fit(data[, "d"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "7.62 pM")
> e.plot <- apt.fit(data[, "e"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "3.81 pM")
> f.plot <- apt.fit(data[, "f"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "1.90 pM")
> g.plot <- apt.fit(data[, "g"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "0.95 pM")
> h.plot <- apt.fit(data[, "h"], 44, 1, 3, 53, 1, 3, "BFGS", 2e-9, temp, "0.47 pM")
> ## To arrange all the plots in a single plot
> grid.arrange(nc.plot, b.plot, c.plot, d.plot, e.plot, f.plot, g.plot,
+               h.plot, ncol = 2, nrow = 4)
```

8 Compute the 95 percent confidence intervals of the Gaussian means

Function `mean.conf` acts on an object of the Class `aptaqpcr`, which is produced as an output of the function `apt.fit.value` and populates an object of Class `aptaqpcr`. The function `mean.conf` computes the 95 percent confidence intervals of the Gaussian means to validate the experiment statistically. To know about the arguments of the function `mean.conf` in detail, see the documentation file of the function `apt.fit.value`. Following is the example to implement the function `mean.conf`:

```
> ## To compute the confidence intervals of the Gaussian means of temperature
> ## 1) By omitting the last value
> conf.int <- mean.conf(aptfitalis)
> ## To express the mean values
> slot(conf.int, "confMeanData")
> ## To express the confidence intervals
> slot(conf.int, "confData")
> ## 2) By including the last value
> conf.int.last <- mean.conf(aptfitalis, mean.all = "Yes")
> ## To express the mean values
> slot(conf.int.last, "confMeanData")
```

9 Plot the Gaussian Mean (Temperature) vs concentration

The function `tempconc.plot` acts on an object of the Class `aptaqpcr`, produced as an output of the function `apt.fit.value` to generate the plots, on the basis of the function `ggplot` of the package "ggplot2". The function `tempconc.plot` generates the plots of the Gaussian Mean (Temperature) vs concentration, where the error bars are also shown to represent the errors between the mean values. To know about the arguments of the function `tempconc.plot` in detail, see the documentation file of the function `tempconc.plot`. Following is the example to implement the function `tempconc.plot`:

```
> ## To plot the Gaussian Mean (Temperature) vs Concentration
> ## 1) By omitting the last value of mean
> gmc.plot <- tempconc.plot(aptfitalls,
+                           conc = c(0,30.50,15.25,7.62,3.81,1.90,0.95),
+                           xlabel = "Concentration (pM)")
> gmc.plot
> ## 2) By including the last value of mean
> gmc.plot.last <- tempconc.plot(aptfitalls,
+                                conc = c(0,30.50,15.25,7.62,3.81,1.90,0.95,0.47),
+                                xlabel = "Concentration (pM)")
> gmc.plot.last
```

10 Graphical User Interface

The shiny app based graphical user interface (GUI) is designed for making the package more user friendly. One can download the dynamic report in html format after implementation, which will be automatically saved in the folder `AptaqPCR.gui` in folder `inst` of the package "aptaqPCR". The graphical user interface can be launched in an easy way through the function `AptaqPCR.gui`. Following is the example to launch the graphical user interface:

```
> ## To launch the graphical user interface
> AptaqPCR.gui()
```

11 Report generation of the plots

The .Rmd files are provided in the folder `plotreport`, which is present in the folder `inst` of the package "AptaqPCR". It has `plot.Rmd` file. Here, the example is shown with the file `multiapt31.csv`, which is present in the folder `exData` of folder `inst` of package "AptaqPCR". To generate the html report, one can use `knit to html`, while To generate the pdf report, one can use `knit to pdf`. We recommend to generate the html report. The resulting examples are shown in the files `plot` in two formats of pdf and html.