

Package: shinyMolBio (via r-universe)

September 10, 2024

Type Package

Title Molecular Biology Visualization Tools for 'Shiny' Apps

Version 0.4

Date 2020-05-07

Description Interactive visualization of 'RDML' files via 'shiny' apps. Package provides (1) PCR plate interface with ability to select individual tubes; (2) amplification/melting plots with fast hiding and highlighting individual curves; (3) 2D allelic discrimination plot.

License MIT + file LICENSE

URL <https://github.com/PCRuniversum/shinyMolBio>,
<https://pcruniversum.github.io/shinyMolBio/>

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0)

Imports dplyr, purrr, checkmate, RDML, shiny, stringr, whisker, plotly, RColorBrewer

Collate 'global.R' 'pcrPlate-input.R' 'renderCurves.R'
'renderADplot.R' 'runExample.R'

Suggests knitr, chipPCR

RoxygenNote 7.1.0

Repository <https://pcruniversum.r-universe.dev>

RemoteUrl <https://github.com/pcruniversum/shinymolbio>

RemoteRef HEAD

RemoteSha 6de5b243f77a81fabae1101b20d18f1f978e9e8b

Contents

pcrPlateInput	2
print.adPlot	4
print.pcrCurves	4
print.pcrPlate	5
renderADplot	6
renderAmpCurves	7
renderMeltCurves	8
runExample	9
updateADplot	10
updateCurves	10
updatePcrPlateInput	11

Index	13
--------------	-----------

pcrPlateInput	<i>Create a PCR plate input control</i>
---------------	---

Description

Create an input control for representing PCR plate and dynamically selecting wells inside it.

Usage

```
pcrPlateInput(inputId,
  label = NULL,
  plateDescription,
  pcrFormat = pcrFormatType$new(8, 12, labelFormatType$new("ABC"),
    labelFormatType$new("123")),
  selection = NULL,
  highlighting = NULL,
  wellLabelTemplate = "{{sample}}",
  onHoverWellTextTemplate = "{{position}}\u000A{{sample}}\u000A{{targets}}",
  wellClassTemplate = NULL,
  wellStyleTemplate = NULL,
  wellGroupTemplate = "{{sample}}-{{targets}}",
  cssFile = system.file("/css/pcrPlateInputStyle.css",
    package = "shinyMolBio"),
  cssText = NULL,
  legend = NULL,
  interactive = TRUE)
```

Arguments

inputId	The input slot that will be used to access the selected wells positions.
label	Display label for the control, or NULL for no label.

plateDescription	Plate description - basically output from RDML AsTable() function.
pcrFormat	PCR plate parametrs. Should be pcrFormatType.
selection	Set preselected wells (e.g. c("A01", "A02") or c(1, 2))
highlighting	Set highlighted wells (e.g. c("A01", "A02") or c(1, 2))
wellLabelTemplate	Template of the well label.
onHoverWellTextTemplate	Template of the text on hover.
wellClassTemplate	Template of the well class (css class).
wellStyleTemplate	Template of the well style (css).
wellGroupTemplate	Template of the well group for selecting.
cssFile	Path to the css styles file.
cssText	CSS styles as text.
legend	Plate legend (any HTML content).
interactive	Should be this pcrPlate interactive or not.

Value

A PCR plate control that can be added to a UI definition.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

See Also

[updatePcrPlateInput](#)

Examples

```
## Only run examples in interactive R sessions
library(RDML)
if (interactive()) {
  ui <- fluidPage(
    pcrPlateInput("plate1",
                  "Plate 1",
                  RDML$new(system.file("/extdata/stepone_std.rdm1", package = "RDML"))$AsTable(),
                  pcrFormatType$new(8,12,labelFormatType$new("ABC"),
                                     labelFormatType$new("123"))),
    verbatimTextOutput("selected")
  )
  server <- function(input, output) {
    output$selected <- renderText({ input$plate1 })
  }
}
```

```
  shinyApp(ui, server)
}
```

`print.adPlot` *Printing adPlot*

Description

Print a adPlot

Usage

```
## S3 method for class 'adPlot'
print(x)
```

Arguments

x object of class adPlot

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

See Also

[renderAmpCurves](#), [renderMeltCurves](#)

`print.pcrCurves` *Printing pcrCurves*

Description

Print a pcrCurves

Usage

```
## S3 method for class 'pcrCurves'
print(x)
```

Arguments

x object of class pcrCurves

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

See Also

[renderAmpCurves](#), [renderMeltCurves](#)

Examples

```
library(RDML)
rdml <- RDML$new(system.file("/extdata/test.rdml", package = "shinyMolBio"))
curves <- renderMeltCurves("curves1", meltCurves = rdml$GetFData(dp.type = "mdp",
  long.table = TRUE))
curves
```

<code>print.pcrPlate</code>	<i>Printing PCR plate input control</i>
-----------------------------	---

Description

Print a pcrPlate

Usage

```
## S3 method for class 'pcrPlate'
print(x)
```

Arguments

x object of class pcrPlate

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

See Also

[pcrPlateInput](#)

renderADplot

Renders an allelic discrimination viewer

Description

Renders a reactive allelic discrimination plot that is suitable for assigning to an UI output slot.

Usage

```
renderADplot(inputId, label = NULL, adData,
             targetColumn = "target.dyeId", xAxisTarget = "FAM",
             yAxisTarget = "HEX", valueColumn = "endPointRFU",
             colorBy = "genotype", polar = FALSE, showLegend = FALSE,
             plotlyCode = NULL, cssFile = NULL, cssText = NULL,
             interactive = TRUE)
```

Arguments

inputId	The input slot that will be used to modify plot.
label	Display label for the control, or NULL for no label.
adData	Allelic discrimination data with <code>RDML\$AsTable()</code> format.
targetColumn	Column name that contains axis splitting (dye or target).
xAxisTarget	X axis target (dye) name.
yAxisTarget	X axis target (dye) name.
valueColumn	Column name that contains discrimination value (RFU or cq).
colorBy	Column name that contains color levels data.
polar	Enables polar coordinates.
showLegend	Show plot legend.
plotlyCode	Your quoted custom plotly code.
cssFile	Path to the css styles file.
cssText	CSS styles as text.
interactive	Should be this adPLot interactive or not.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

See Also

[updateADplot](#)

Other render elements: [renderAmpCurves\(\)](#), [renderMeltCurves\(\)](#)

Examples

```

library(RDML)
rdml <- RDML$new(system.file("/extdata/test.rdml", package = "shinyMolBio"))
tbl <- rdml$AsTable(endPointRFU = mean(tail(data$adp$fpoints$fluor, 5)))
tbl <- group_by(tbl, position)
tbl <- mutate(tbl,
              genotype = paste(
                if (endPointRFU[1] > 400 && endPointRFU[2] > 400) "AG"
                else if (endPointRFU[1] > 400) "AA"
                else if (endPointRFU[2] > 400) "GG"
                else "NA"
              ))
renderADplot("f", "aa", tbl, polar = TRUE, showLegend = TRUE)

```

renderAmpCurves

*Renders an amplification curves viewer***Description**

Renders a reactive PCR amplification plot that is suitable for assigning to an UI output slot.

Usage

```

renderAmpCurves(inputId, label = NULL, ampCurves, colorBy = NULL,
                 linetypeBy = NULL, logScale = FALSE, showCq = FALSE, showLegend = FALSE,
                 thBy = NULL, plotlyCode = NULL, cssFile = NULL, cssText = NULL,
                 interactive = TRUE)

```

Arguments

inputId	The input slot that will be used to modify plot.
label	Display label for the control, or NULL for no label.
ampCurves	Amplification curves data with <code>RDML\$GetFData(long.table = TRUE)</code> format.
colorBy	Column name that contains color levels data.
linetypeBy	Column name that contains linetype levels data.
logScale	Converts plot to $\log(\text{RFU})$.
showCq	Shows Cq with dots (cq column have to be provided!).
showLegend	Show plot legend.
thBy	Column name that separates threshold values (quantFluor column have to be provided!).
plotlyCode	Your quoted custom plotly code.
cssFile	Path to the css styles file.
cssText	CSS styles as text.
interactive	Should be this pcrPlate interactive or not.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

See Also

[updateCurves](#)

Other render elements: [renderADplot\(\)](#), [renderMeltCurves\(\)](#)

Examples

```
library(RDML)
rdml <- RDML$new(system.file("/extdata/test.rdml", package = "shinyMolBio"))
curves <- renderAmpCurves("curves1", ampCurves = rdml$GetFData(long.table = TRUE))
curves
```

renderMeltCurves	<i>Renders a melting curves viewer</i>
------------------	--

Description

Renders a reactive melting plot that is suitable for assigning to an UI output slot.

Usage

```
renderMeltCurves(inputId, label = NULL, meltCurves,
  fluorColumn = "fluor", colorBy = NULL, linetypeBy = NULL, showTm = FALSE,
  showLegend = FALSE, plotlyCode = NULL, cssFile = NULL, cssText = NULL,
  interactive = TRUE)
```

Arguments

inputId	The input slot that will be used to modify plot.
label	Display label for the control, or NULL for no label.
meltCurves	Melting curves data with <code>RDML\$GetFData(dp.type = "mdp", long.table = TRUE)</code> format.
fluorColumn	Column name that contains fluorescence values (can be <code>diffFluor</code> for derivative curves).
colorBy	Column name that contains color levels data.
linetypeBy	Column name that contains linetype levels data.
showTm	Shows Tm with dots (tm column have to be provided!)
showLegend	Show plot legend.
plotlyCode	Your quoted custom plotly code.
cssFile	Path to the css styles file.
cssText	CSS styles as text.
interactive	Should be this <code>pcrPlate</code> interactive or not.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

See Also

[updateCurves](#)

Other render elements: [renderADplot\(\)](#), [renderAmpCurves\(\)](#)

Examples

```
library(RDML)
rdml <- RDML$new(system.file("/extdata/test.rdml", package = "shinyMolBio"))
curves <- renderMeltCurves("curves1", meltCurves = rdml$GetFData(dp.type = "mdp",
  long.table = TRUE))
curves
```

runExample

Run shinyMolBio Example Applications

Description

Launch shinyMolBio example applications

Usage

```
runExample(example)
```

Arguments

example	The name of the example to run, or NA (the default) to list the available examples.
---------	---

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

Examples

```
## Only run this example in interactive R sessions
if (interactive()) {
  # List all available examples
  runExample()

  # Run one of the examples
  runExample("pcrPlateInput")

  # Print the directory containing the code for all examples
  system.file("shiny-examples", package="shinyMolBio")
}
```

updateADplot	<i>Change the value of a render ADplot control on the client</i>
--------------	--

Description

Change the value of a render ADplot control on the client

Usage

```
updateADplot(  
  session,  
  inputId,  
  label = NULL,  
  hidePoints = NULL,  
  highlightPoints = NULL  
)
```

Arguments

session	The session object passed to function given to shinyServer.
inputId	The id of the input object.
label	The label to set for the input object.
hidePoints	The position of the points to be hidden.
highlightPoints	The position of the points to be highlighted.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

updateCurves	<i>Change the value of a render PCR curves control on the client</i>
--------------	--

Description

Change the value of a render PCR curves control on the client

Usage

```
updateCurves(  
  session,  
  inputId,  
  label = NULL,  
  hideCurves = NULL,  
  highlightCurves = NULL  
)
```

Arguments

session	The session object passed to function given to shinyServer.
inputId	The id of the input object.
label	The label to set for the input object.
hideCurves	The fdata.names of the curves to be hidden.
highlightCurves	The fdata.names of the curves to be highlighted.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

updatePcrPlateInput *Change the value of a PCR plate input control on the client*

Description

Change the value of a PCR plate input control on the client

Usage

```
updatePcrPlateInput(  
  session,  
  inputId,  
  label = NULL,  
  selection = NULL,  
  highlighting = NULL  
)
```

Arguments

session	The session object passed to function given to shinyServer.
inputId	The id of the input object.
label	The label to set for the input object.
selection	The positions of the wells to be selected.
highlighting	The positions of the wells to be highlighted.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>

Examples

```
## Only run examples in interactive R sessions
library(RDML)
if (interactive()) {
  ui <- fluidPage(
    pcrPlateInput("plate1",
                  "Plate 1",
                  RDML$new(system.file("/extdata/stepone_std.rdml", package = "RDML"))$AsTable(),
                  pcrFormatType$new(8,12,labelFormatType$new("ABC"),
                                    labelFormatType$new("123"))),
    verbatimTextOutput("selected"),
    actionButton("selectWellBtn",
                 "Select Well A01-A03")
  )
  server <- function(input, output, session) {
    output$selected <- renderText({ input$plate1 })

    observeEvent(
      input$selectWellBtn,
      {
        updatePcrPlateInput(session,
                            "plate1",
                            selection = c("A01", "A02", "A03"))
      }
    )
  }
  shinyApp(ui, server)
}
```

Index

* **Input**

- pcrPlateInput, 2
- print.adPlot, 4
- print.pcrCurves, 4
- print.pcrPlate, 5
- renderADplot, 6
- renderAmpCurves, 7
- renderMeltCurves, 8
- updateADplot, 10
- updateCurves, 10
- updatePcrPlateInput, 11

* **PCR**

- pcrPlateInput, 2
- print.adPlot, 4
- print.pcrCurves, 4
- print.pcrPlate, 5
- renderADplot, 6
- renderAmpCurves, 7
- renderMeltCurves, 8
- updateADplot, 10
- updateCurves, 10
- updatePcrPlateInput, 11

* **RDML**

- pcrPlateInput, 2
- print.adPlot, 4
- print.pcrCurves, 4
- print.pcrPlate, 5
- renderADplot, 6
- renderAmpCurves, 7
- renderMeltCurves, 8
- updateADplot, 10
- updateCurves, 10
- updatePcrPlateInput, 11

* **Shiny**

- pcrPlateInput, 2
- print.adPlot, 4
- print.pcrCurves, 4
- print.pcrPlate, 5
- renderADplot, 6

- renderAmpCurves, 7
- renderMeltCurves, 8
- updateADplot, 10
- updateCurves, 10
- updatePcrPlateInput, 11

* **input elements**

- pcrPlateInput, 2

* **render elements**

- renderADplot, 6
- renderAmpCurves, 7
- renderMeltCurves, 8

- pcrPlateInput, 2, 5

- print.adPlot, 4
- print.pcrCurves, 4
- print.pcrPlate, 5

- renderADplot, 6, 8, 9
- renderAmpCurves, 4–6, 7, 9
- renderMeltCurves, 4–6, 8, 8
- runExample, 9

- updateADplot, 6, 10
- updateCurves, 8, 9, 10
- updatePcrPlateInput, 3, 11