

# Package ‘shinyMolBio’

October 14, 2022

**Type** Package

**Title** Molecular Biology Visualization Tools for 'Shiny' Apps

**Version** 0.2

**Date** 2019-08-02

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

**License** MIT + file LICENSE

**URL** <https://github.com/kablag/shinyMolBio>,  
<https://kablag.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' 'runExample.R'

**Suggests** knitr, chipPCR

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Konstantin A. Blagodatskikh [cre, aut]

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

**Repository** CRAN

**Date/Publication** 2019-08-02 10:20:03 UTC

## R topics documented:

pcrPlateInput . . . . .	2
renderAmpCurves . . . . .	4

renderMeltCurves . . . . .	5
runExample . . . . .	6
updateCurves . . . . .	6
updatePcrPlateInput . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

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 params. 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.

<code>wellClassTemplate</code>	Template of the well class (css class).
<code>wellStyleTemplate</code>	Template of the well style (css).
<code>wellGroupTemplate</code>	Template of the well group for selecting.
<code>cssFile</code>	Path to the css styles file.
<code>cssText</code>	CSS styles as text.
<code>legend</code>	Plate legend (any HTML content).
<code>interactive</code>	Should be this <code>pcrPlate</code> 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$new(system.file("/extdata/stepone_std.rdml", package = "RDML"))$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)
}
```

---

renderAmpCurves	<i>Renders an amplification curves viewer</i>
-----------------	---

---

### 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(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: [renderMeltCurves](#)

### Examples

```
library(RDML)
rdml <- RDML$new(system.file("/extdata/test.rdml", package = "shinyMolBio"))
curves <- renderAmpCurves("curves1", ampCurves = rdml$GetFData(long.table = TRUE))
curves[[2]][[3]][[2]]
```

---

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, 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.
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 pcrPlate interactive or not.

### Author(s)

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

### See Also

[updateCurves](#)

Other render elements: [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[[2]][[3]][[2]]
```

---

`runExample`*Run shinyMolBio Example Applications*

---

**Description**

Launch shinyMolBio example applications

**Usage**

```
runExample(example)
```

**Arguments**

<code>example</code>	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")
}
```

---

`updateCurves`*Change the value of a render PCR curves control on the client*

---

**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
- renderAmpCurves, 4
- renderMeltCurves, 5
- updateCurves, 6
- updatePcrPlateInput, 7

## \* **PCR**

- pcrPlateInput, 2
- renderAmpCurves, 4
- renderMeltCurves, 5
- updateCurves, 6
- updatePcrPlateInput, 7

## \* **RDML**

- pcrPlateInput, 2
- renderAmpCurves, 4
- renderMeltCurves, 5
- updateCurves, 6
- updatePcrPlateInput, 7

## \* **Shiny**

- pcrPlateInput, 2
- renderAmpCurves, 4
- renderMeltCurves, 5
- updateCurves, 6
- updatePcrPlateInput, 7

## \* **input elements**

- pcrPlateInput, 2

## \* **render elements**

- renderAmpCurves, 4
- renderMeltCurves, 5

pcrPlateInput, 2

renderAmpCurves, 4, 5  
renderMeltCurves, 4, 5  
runExample, 6

updateCurves, 4, 5, 6  
updatePcrPlateInput, 3, 7