

# Package ‘RGalaxy’

March 26, 2013

**Title** Make an R function available in the Galaxy web platform

**Description** Given an R function and its manual page, make the documented function available in Galaxy.

**Version** 1.0.0

**Author** The Bioconductor Dev Team

**Maintainer** Bioconductor Package Maintainer <maintainer@bioconductor.org>

**biocViews** Infrastructure

**Depends** XML, methods, tools, optparse, digest

**Imports** BiocGenerics, Biobase, roxygen2

**Suggests** RUnit, hgu95av2.db

**License** Artistic-2.0

## R topics documented:

anotherTestFunction . . . . .	2
foo . . . . .	3
functionToGalaxify . . . . .	3
galaxy . . . . .	4
GalaxyConfig-class . . . . .	6
GalaxyInputFile-class . . . . .	6
GalaxyOutput-class . . . . .	7
GalaxyParam-class . . . . .	8
getFriendlyName . . . . .	9
testFunctionWithSelect . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

anotherTestFunction     *Add two matrices*

---

### Description

An example function that can be made into a Galaxy tool. Reads matrices from two tab-delimited files, adds them, and writes the result to a comma-separated file and a PDF plot.

### Usage

```
anotherTestFunction(inputfile1=GalaxyInputFile(),
                    inputfile2=GalaxyInputFile(), plotTitle=c("TitleA"="A", "TitleB"="B"),
                    plotSubTitle="My subtitle",
                    outputfile1=GalaxyOutput("mydata", "csv"),
                    outputfile2=GalaxyOutput("myplot", "pdf"))
```

### Arguments

inputfile1	The filename of the first tab-separated matrix.
inputfile2	The filename of the second tab-separated matrix.
plotTitle	The title of the plot to create.
plotSubTitle	The subtitle of the plot to create.
outputfile1	The filename of the comma-separated output file to generate.
outputfile2	The filename of the PDF plot file to create.

### Details

This trivial method illustrates some best practices to use when writing functions to be adapted as Galaxy tools. For example, any error conditions should be handled with `stop` with a useful/informative error message. The Galaxy user will see these messages if an error occurs.

Functions which take datasets as input should accept as arguments the filenames pointing to those datasets. The Galaxy user interface will allow the user to choose the dataset graphically.

Return values of functions are ignored. Function output should be written to one or more files, and the names of these files should be passed into the function as arguments.

Functions should be documented with a manual page. RGalaxy will use this manual page to fill in relevant sections of the Galaxy XML file.

### Value

`invisible(NULL)`

### See Also

[galaxy](#), [GalaxyParam](#), [GalaxyConfig](#), [GalaxyOutput](#)

**Examples**

```
anotherTestFunction(system.file("extdata", "a.tsv", package="RGalaxy"),
  system.file("extdata", "b.tsv", package="RGalaxy"),
  "My Plot Title", "My Plot Subtitle",
  "output.csv", "output.pdf")
```

---

foo	<i>a foo function</i>
-----	-----------------------

---

**Description**

a foo function

**Usage**

```
foo(input = GalaxyInputFile(), x = numeric(), y = TRUE,
  z = c("Seattle", "Tacoma", "Olympia"),
  output = GalaxyOutput("pdf"))
```

**Arguments**

input	An input dataset
x	the x param
y	the y param
z	the z param
output	the output

**Details**

nothing

---

functionToGalaxify	<i>Add two matrices</i>
--------------------	-------------------------

---

**Description**

An example function that can be made into a Galaxy tool. Reads matrices from two tab-delimited files, adds them, and writes the result to a comma-separated file and a PDF plot.

**Usage**

```
functionToGalaxify(inputfile1, inputfile2, plotTitle=character(),
  plotSubTitle="My subtitle",
  outputfile1=GalaxyOutput("mydata", "csv"),
  outputfile2=GalaxyOutput("myplot", "pdf"))
```

**Arguments**

inputfile1	The filename of the first tab-separated matrix.
inputfile2	The filename of the second tab-separated matrix.
plotTitle	The title of the plot to create.
plotSubTitle	The subtitle of the plot to create.
outputfile1	The filename of the comma-separated output file to generate.
outputfile2	The filename of the PDF plot file to create.

**Details**

This trivial method illustrates some best practices to use when writing functions to be adapted as Galaxy tools. For example, any error conditions should be handled with `stop` with a useful/informative error message. The Galaxy user will see these messages if an error occurs.

Functions which take datasets as input should accept as arguments the filenames pointing to those datasets. The Galaxy user interface will allow the user to chose the dataset graphically.

Return values of functions are ignored. Function output should be written to one or more files, and the names of these files should be passed into the function as arguments.

Functions should be documented with a manual page. RGalaxy will use this manual page to fill in relevant sections of the Galaxy XML file.

**Value**

invisible(NULL)

**See Also**

[galaxy](#), [GalaxyParam](#), [GalaxyConfig](#), [GalaxyOutput](#)

**Examples**

```
functionToGalaxify(system.file("extdata", "a.tsv", package="RGalaxy"),
  system.file("extdata", "b.tsv", package="RGalaxy"),
  "My Plot Title", "My Plot Subtitle",
  "output.csv", "output.pdf")
```

---

galaxy

*Make a function available in Galaxy*

---

**Description**

Makes an R function available in the Galaxy web platform. Automates all the work necessary to expose an R function in Galaxy. See the vignette for more information

**Usage**

```
galaxy(func, manpage, ..., name=getFriendlyName(deparse(substitute(func))),
  package=NULL, is.exported=NULL, version, galaxyConfig, packageSourceDir)
```

**Arguments**

func	<b>Required.</b> The function to make available in Galaxy. This function should be entirely-self contained, and should accept as arguments the full paths to its input and output file(s).
manpage	<b>Required.</b> The full path to the Rd-formatted manual page for the function, or if package is specified, an alias that will pull up that manpage.
...	<b>Required.</b> A list of at least one <a href="#">GalaxyParam</a> object and at least one <a href="#">GalaxyOutput</a> objects.
name	<b>Required.</b> Text describe the action this function performs. Becomes a clickable link in Galaxy. By default, if your function is called "fooBar", name is set to "Foo Bar".
package	The name of the package where func lives, or NULL (the default) if it is not in a package.
is.exported	If package is supplied, a logical indicating whether func is exported from package.
version	The version of this function. If package is TRUE, defaults to the version of package.
galaxyConfig	<b>Required.</b> A link{GalaxyConfig} object describing your Galaxy configuration.
packageSourceDir	If present, points to the directory of a source package upon which to run roxygenize() from the roxygen2 package, creating manual pages from source code comments.

**Value**

invisible(NULL)

**See Also**[GalaxyParam](#), [GalaxyConfig](#), [GalaxyOutput](#)**Examples**

```
## set up galaxyHome (you don't need to do this if
## you really have Galaxy installed)
galaxyHome <- tempdir()
dir.create(galaxyHome, recursive=TRUE, showWarnings=FALSE)
file.copy(system.file("galaxy", "tool_conf.xml", package="RGalaxy"),
  file.path(galaxyHome, "tool_conf.xml"), overwrite=FALSE)

galaxy(functionToGalaxify,
  manpage="functionToGalaxify",
  package="RGalaxy",
  version=packageDescription("RGalaxy")$Version,
  galaxyConfig=GalaxyConfig(galaxyHome, "mytool", "Test Section", "testSectionId"))
```

---

GalaxyConfig-class      *Class "GalaxyConfig"*

---

### Description

Represents information about the configuration of Galaxy.

### Usage

GalaxyConfig(galaxyHome, toolDir, sectionName, sectionId)

### Arguments

galaxyHome	<b>Required.</b> The full path to your Galaxy distribution.
toolDir	<b>Required.</b> The directory where the files associated with your function will reside.
sectionName	<b>Required</b> A friendly name for the section of Galaxy tools that will contain your function. Multiple tools can be in a single section; section names are headings in the left-hand side of the Galaxy window. If multiple tools are to reside in the same section, they must have identical values for sectionName and sectionId.
sectionId	An internal identifier for the section of Galaxy tools that will contain your function. If multiple tools are to reside in the same section, they must have identical values for sectionName and sectionId.

### Methods

No methods defined with class "GalaxyConfig" in the signature.

### See Also

link{galaxy}, link{GalaxyParam}, [GalaxyOutput](#)

### Examples

```
GalaxyConfig(galaxyHome=getwd(), toolDir="mytool", sectionName="Test Section",
             sectionId="testSectionId")
```

---

GalaxyInputFile-class      *Class "GalaxyInputFile"*

---

### Description

Represents a dataset hosted on Galaxy, to be passed to an R function that you expose in Galaxy.

### Usage

GalaxyInputFile(required=TRUE)

**Arguments**

required            Whether to require that the user provide this dataset.

**Methods**

No methods defined with class "GalaxyInputFile" in the signature.

**See Also**

link{galaxy}, [GalaxyConfig](#), [GalaxyParam](#)

**Examples**

```
inputfile1=GalaxyInputFile()
```

---

GalaxyOutput-class      *Class* "GalaxyOutput"

---

**Description**

Represents an output file generated by an R function in Galaxy.

**Usage**

```
GalaxyOutput(basename, format)
```

**Arguments**

basename            **Required.** The name of the output file, minus the extension.

format                **Required.** The file type of the output file. For suggested extensions, see GALAXY\_HOME/datatypes

**Methods**

No methods defined with class "GalaxyOutput" in the signature.

**See Also**

link{galaxy}, [GalaxyConfig](#), [GalaxyParam](#)

**Examples**

```
params <- list(
  outputfile1=GalaxyOutput("plot", "pdf"),
  outputfile2=GalaxyOutput("data", "csv"))
```

---

GalaxyParam-class      *Class* "GalaxyParam"

---

### Description

Represents information about a parameter to a function that is exposed in Galaxy. Should only be used if the function signature itself does not fully describe the way the parameter should be exposed in Galaxy.

### Usage

```
GalaxyParam(label, min, max, force_select, display,
            checked, size, required=FALSE,
            requiredMsg="This field is required.")
```

### Arguments

label	<b>Required</b> A friendly label for the parameter, if the name of the parameter is not sufficient.
min	The minimum allowed value for the parameter, if the parameter type is numeric or integer.
max	The maximum allowed value for the parameter if the parameter type is numeric or integer.
force_select	Logical. Whether to force user to select an option in the list. Only supplied if the parameter is a select list, that is, a vector of more than one element.
display	If the parameter is a select list, this can be either checkboxes or radio, and determines how to render this parameter in HTML. Default is a drop-down select list.
checked	Logical. If the parameter type is logical, determines whether check box should be checked.
size	If the parameter type is character, this determines the width (in characters) of the resulting text box.
required	Logical. Whether the item is required. If TRUE, the user will be forced by Galaxy to enter a value for this item.
requiredMsg	If required is TRUE and the user tries to run the Galaxy tool without specifying a value for this parameter, this message will be displayed.

### Methods

No methods defined with class "GalaxyParam" in the signature.

### See Also

link{galaxy}, [GalaxyConfig](#), link{GalaxyOutput}

### Examples

```
title <- GalaxyParam(required=TRUE, label="Override the default label")
```



---

getFriendlyName	<i>Change a camelCase name to a friendlier version</i>
-----------------	--

---

**Description**

Takes a string like "fooBarBaz" and returns "Foo Bar Baz".

**Usage**

```
getFriendlyName(camelName)
```

**Arguments**

camelName      The "camelCased" name to make friendly.

**Details**

Used by galaxy() to create default labels based on function and parameter names.

**Value**

The friendly version of the camel-cased name.

**See Also**

[galaxy](#), [GalaxyParam](#), [GalaxyConfig](#), [GalaxyOutput](#)

**Examples**

```
getFriendlyName("fooBarBaz")
```

---

testFunctionWithSelect	<i>A variation on functionToGalaxify that takes a multiple-choice option.</i>
------------------------	---

---

**Description**

A variation on functionToGalaxify that takes a multiple-choice option.

**Usage**

```
testFunctionWithSelect(inputfile1 = GalaxyInputFile(),  
  inputfile2 = GalaxyInputFile(),  
  plotTitle = c(TitleA = "A", TitleB = "B"),  
  plotSubTitle = "My subtitle",  
  outputfile1 = GalaxyOutput("mydata", "csv"),  
  outputfile2 = GalaxyOutput("myplot", "pdf"))
```

**Arguments**

inputfile1	the first matrix
inputfile2	the second matrix
plotTitle	the plot title
plotSubTitle	the plot subtitle
outputfile1	the csv output file
outputfile2	the pdf output file

**Details**

There are no details.

# Index

## \*Topic **classes**

- GalaxyConfig-class, [6](#)
- GalaxyInputFile-class, [6](#)
- GalaxyOutput-class, [7](#)
- GalaxyParam-class, [8](#)

[anotherTestFunction](#), [2](#)

[foo](#), [3](#)

[functionToGalaxify](#), [3](#)

[galaxy](#), [2](#), [4](#), [4](#), [9](#)

[GalaxyConfig](#), [2](#), [4](#), [5](#), [7–9](#)

[GalaxyConfig \(GalaxyConfig-class\)](#), [6](#)

[GalaxyConfig-class](#), [6](#)

[GalaxyInputFile \(GalaxyInputFile-class\)](#), [6](#)

[GalaxyInputFile-class](#), [6](#)

[GalaxyOutput](#), [2](#), [4–6](#), [9](#)

[GalaxyOutput \(GalaxyOutput-class\)](#), [7](#)

[GalaxyOutput-class](#), [7](#)

[GalaxyParam](#), [2](#), [4](#), [5](#), [7](#), [9](#)

[GalaxyParam \(GalaxyParam-class\)](#), [8](#)

[GalaxyParam-class](#), [8](#)

[getFriendlyName](#), [9](#)

[testFunctionWithSelect](#), [9](#)