Package: pepr (via r-universe)

September 2, 2024

Type Package

Title Reading Portable Encapsulated Projects

Version 0.5.0

Date 2023-11-16

Maintainer Nathan Sheffield <nathan@code.databio.org>

Description A PEP, or Portable Encapsulated Project, is a dataset that subscribes to the PEP structure for organizing metadata. It is written using a simple YAML + CSV format, it is your one-stop solution to metadata management across data analysis environments. This package reads this standardized project configuration structure into R. Described in Sheffield et al. (2021) <doi:10.1093/gigascience/giab077>.

Imports yaml, stringr, pryr, data.table, methods, RCurl

Suggests knitr, testthat, rmarkdown, curl

VignetteBuilder knitr

License BSD_2_clause + file LICENSE

BugReports https://github.com/pepkit/pepr

RoxygenNote 7.2.3

Encoding UTF-8

Repository https://pepkit.r-universe.dev

RemoteUrl https://github.com/pepkit/pepr

RemoteRef HEAD

RemoteSha 150b52dccee40c5c7cc82806b6ef6ae3411c4857

Contents

activateAmendments 2
checkSection 3
config 4
Config-class 4

fetchSamples 5
getSample 6
getSubsample 7
listAmendments 8
makeSectionsAbsolute 9
pepr 9
Project 10
Project-class 10
sampleTable 11
select-config 12

Index **13**

`activateAmendments` *Activate amendments in objects of* `"Project"`

Description

This method switches between the amendments within the `"Project"` object

Usage

```
activateAmendments(.Object, amendments)

## S4 method for signature 'Project,character'
activateAmendments(.Object, amendments)
```

Arguments

`.Object`	an object of class `"Project"`
`amendments`	character with the amendment name

Details

To check what are the amendments names call `listAmendments(p)`, where `p` is the object of `"Project"` class

Value

an object of class `"Project"` with activated amendments

Methods (by class)

- `activateAmendments(.Object = Project, amendments = character)`: activate amendments in a `"Project"` object

Examples

```
projectConfig = system.file("extdata",
"example_peps-master",
"example_amendments1",
"project_config.yaml",
package = "pepr")
p = Project(file = projectConfig)
availAmendments = listAmendments(p)
activateAmendments(p, availAmendments[1])
```

---

checkSection *Check for existence of a section in the Project config*

Description

This function checks for the section/nested sections in the config YAML file. Returns TRUE if it exist(s) or FALSE otherwise.

Usage

```
checkSection(object, sectionNames)

## S4 method for signature 'Config'
checkSection(object, sectionNames)
```

Arguments

object	object of "Config"
sectionNames	the name of the section or names of the nested sections to look for

Details

Element indices can be used instead of the actual names, see Examples.

Value

a logical indicating whether the section exists

Methods (by class)

- checkSection(Config): checks for existence of a section in "Config" objects

Examples

```
projectConfig = system.file("extdata", "example_peps-master",
"example_amendments1", "project_config.yaml", package="pepr")
p=Project(projectConfig)
checkSection(config(p),sectionNames = c("amendments","newLib"))
checkSection(config(p),sectionNames = c("amendments",1))
```

config *Extract "Project"*

Description

This method can be used to view the config slot of the "Project" class

Usage

```
config(object)

## S4 method for signature 'Project'
config(object)
```

Arguments

object	an object of "Project"

Value

project config

Methods (by class)

- config(Project): Extract "Project" of the object of "Project"

Examples

```
projectConfig = system.file("extdata", "example_peps-master",
"example_amendments1", "project_config.yaml", package="pepr")
p=Project(projectConfig)
config(p)
```

Config-class *Config objects and specialized list obejcts and expand string attributes*

Description

Config objects are used with the "Project" object

Usage

```
Config(file, amendments = NULL)
```

Arguments

file	a character with project configuration yaml file
amendments	a character with the amendments names to be activated

Value

an object of "Config" class

Examples

```
projectConfig = system.file("extdata", "example_peps-master",
"example_amendments1", "project_config.yaml", package="pepr")
c=Config(projectConfig)
```

---

fetchSamples *Collect samples fulfilling the specified requirements*

Description

This funciton collects the samples from a data.table-class object that fulfill the requirements of an attribute attr specified with the fun argument

Usage

```
fetchSamples(samples, attr = NULL, func = NULL, action = "include")
```

Arguments

samples	an object of data.table-class class
attr	a string specifying a column in the samples
func	an anonymous function, see Details for more information
action	a string (either include or exclude) that specifies whether the function should select the row or exclude it.

Details

The anonymous function provided in the func argument has to return an integer that indicate the rows that the action should be performed on. Core expressions which are most useful to implement the anonymous function are:

- which with inequality signs: ==,>,<
- grep

Value

an object of data.table-class class filtered according to specified requirements

Examples

```
projectConfig = system.file("extdata", "example_peps-master",
"example_amendments1", "project_config.yaml", package="pepr")
p = Project(projectConfig)
s = sampleTable(p)
fetchSamples(s,attr = "sample_name", func=function(x){ which(x=="pig_0h") },action="include")
fetchSamples(s,attr = "sample_name", func=function(x){ which(x=="pig_0h") },action="exclude")
fetchSamples(s,attr = "sample_name", func=function(x){ grep("pig_",x) },action="include")
```

---

getSample *Extract samples*

Description

This method extracts the samples

Usage

```
getSample(.Object, sampleName)

## S4 method for signature 'Project,character'
getSample(.Object, sampleName)
```

Arguments

.Object	An object of Project class
sampleName	character the name of the sample

Value

data.table one row data table with the sample associated metadata

Methods (by class)

- `getSample(.Object = Project, sampleName = character)`: extracts the sample from the "Project" object

Examples

```
projectConfig = system.file(
"extdata",
"example_peps-master",
"example_basic",
"project_config.yaml",
package = "pepr"
)
p = Project(projectConfig)
sampleName = "frog_1"
getSample(p, sampleName)
```

getSubsample *Extract subsamples*

Description

This method extracts the subsamples

Usage

```
getSubsample(.Object, sampleName, subsampleName)

## S4 method for signature 'Project,character,character'
getSubsample(.Object, sampleName, subsampleName)
```

Arguments

.Object	An object of Project class
sampleName	character the name of the sample
subsampleName	character the name of the subsample

Value

data.table one row data table with the subsample associated metadata

Methods (by class)

- getSubsample(.Object = Project, sampleName = character, subsampleName = character): extracts the subsamples from the "Project" object

Examples

```
projectConfig = system.file(
"extdata",
"example_peps-master",
"example_subtable1",
"project_config.yaml",
package = "pepr"
)
p = Project(projectConfig)
sampleName = "frog_1"
subsampleName = "sub_a"
getSubsample(p, sampleName, subsampleName)
```

listAmendments *List amendments*

Description

Lists available amendments within a "Project" object.

Usage

```
listAmendments(.Object)

## S4 method for signature 'Project'
listAmendments(.Object)
```

Arguments

.Object	an object of "Project"

Details

The amendments can be activated by passing their names to the activateAmendments method

Value

names of the available amendments

Methods (by class)

- listAmendments(Project): list amendments in a "Project" object

Examples

```
projectConfig = system.file("extdata",
"example_peps-master",
"example_amendments1",
"project_config.yaml",
package = "pepr")
p = Project(file = projectConfig)
availAmendemtns = listAmendments(p)
```

makeSectionsAbsolute *Make selected sections absolute using config path*

Description

Make selected sections absolute using config path

Usage

```
makeSectionsAbsolute(object, sections, cfgPath)

## S4 method for signature 'Config,character,character'
makeSectionsAbsolute(object, sections, cfgPath)
```

Arguments

object	"Config"
sections	character set of sections to make absolute
cfgPath	character absolute path to the config YAML file

Value

Config with selected sections made absolute

Methods (by class)

- makeSectionsAbsolute(object = Config, sections = character, cfgPath = character): Make selected sections absolute using config path from "Project"

pepr *pepr*

Description

Package documentation

Author(s)

Michal Stolarczyk, Nathan Sheffield

References

GitHub: https://github.com/pepkit/pepr, Documentation: https://code.databio.org/pepr/

Project — *The constructor of a class representing a Portable Encapsulated Project*

Description

This is a helper that creates the project with empty samples and config slots

Usage

```
Project(
  file = NULL,
  amendments = NULL,
  sampleTableIndex = NULL,
  subSampleTableIndex = NULL
)
```

Arguments

file	a string specifying a path to a project configuration YAML file
amendments	a string with the amendments names to be activated
sampleTableIndex	a string indicating the sample attribute that is used to index the sample table
subSampleTableIndex	a string indicating the sample attribute that is used to index the sample table

Value

an object of "Project"

Examples

```
projectConfig = system.file("extdata", "example_peps-master",
"example_amendments1", "project_config.yaml", package="pepr")
p=Project(projectConfig)
```

Project-class — *Portable Encapsulated Project object*

Description

Provides an in-memory representation and functions to access project configuration and sample annotation values for a PEP.

Details

Can be created with the constructor: "Project"

Slots

`file` character vector path to config file on disk.

`samples` a data table object holding the sample metadata

`config` a list object holding contents of the config file

`sampleNameAttr` a string indicating the sample attribute that is used to index the sample table

`subSampleNameAttr` a string indicating the sample attribute that is used to index the sample table

---

sampleTable *View samples in the objects of "Project"*

Description

This method can be used to view the samples slot of the "Project" class

Usage

```
sampleTable(object)

## S4 method for signature 'Project'
sampleTable(object)
```

Arguments

`object`	an object of "Project"

Value

a data.table with the with metadata about samples

Methods (by class)

- `sampleTable(Project)`: extract sample table from a "Project"

Examples

```
projectConfig = system.file("extdata", "example_peps-master",
"example_amendments1", "project_config.yaml", package="pepr")
p=Project(projectConfig)
sampleTable(p)
```

select-config *Access "Config" object elements*

Description

You can subset Config by identifier or by position using the `[`, `[[` or `$` operator. The string will be expanded if it's a path.

Usage

```
## S4 method for signature 'Config'
x[i]

## S4 method for signature 'Config'
x[[i]]

## S4 method for signature 'Config'
x$name
```

Arguments

x	a "Config" object.
i	position of the identifier or the name of the identifier itself.
name	name of the element to access.

Value

An element held in "Config" object

Examples

```
projectConfig = system.file("extdata", "example_peps-master",
"example_amendments1", "project_config.yaml", package="pepr")
c=Config(projectConfig)
c[[2]]
c[2]
c[["sample_table"]]
c$sample_table
```

Index

[,Config-method *(select-config)*, 12
[[,Config-method *(select-config)*, 12
$,Config-method *(select-config)*, 12

activateAmendments, 2, *8*
activateAmendments,Project,character-method
(activateAmendments), 2

checkSection, 3
checkSection,Config-method
(checkSection), 3
Config, *3*, *5*, *9*, *12*
Config *(Config-class)*, 4
config, 4
config,Project-method *(config)*, 4
Config-class, 4

fetchSamples, 5

getSample, 6
getSample,Project,character-method
(getSample), 6
getSubsample, 7
getSubsample,Project,character,character-method
(getSubsample), 7
grep, *5*

listAmendments, 8
listAmendments,Project-method
(listAmendments), 8

makeSectionsAbsolute, 9
makeSectionsAbsolute,Config,character,character-method
(makeSectionsAbsolute), 9

pepr, 9
Project, *2*, *4*, *6–10*, 10, *11*
Project-class, 10

sampleTable, 11
sampleTable,Project-method
(sampleTable), 11
select-config, 12

which, *5*