Package 'MagellanNTK'

November 20, 2026

Type Package

Title Configurable Shiny data-analysis workflows

Version 0.99.34

Date 2026-09-07

Description MagellanNTK (Magellan Navigation ToolKit) is a collection of Shiny modules which provide the basis for a generic workflow manager for data processing tools with several steps. It works in combination with any package which implements data processing pipelines compliant with MagellanNTK such as Prostar2.

License Artistic-2.0

URL https://github.com/edyp-lab/MagellanNTK, https://edyp-lab.github.io/MagellanNTK/

BugReports https://github.com/edyp-lab/MagellanNTK/issues

Depends R (>= 4.6.0)

biocViews Software, GUI, ShinyApps, WorkflowStep

Imports S4Vectors, methods, bs4Dash, DT, shiny, shinyEffects, htmlwidgets, shinyjs, sass, stats, utils, BiocGenerics, htmltools, crayon, stringr, waiter, MultiAssayExperiment, SummarizedExperiment, markdown, graphics

Suggests shinyjqui, BiocStyle, testthat, shinytest2, shinyTree, knitr, colourpicker, gtools, XML, R.utils, RColorBrewer, openxlsx, shinyvalidate, shinyAce, rclipboard, shinyWidgets, KernSmooth, MASS, boot, class, cluster, foreign, mgcv, nlme, nnet, rpart, spatial, survival

Collate MagellanNTK.R process_layout.R default_funcs.R mod_sweetAlert.R mod_open_workflow.R directoryInput.R mod_errorModal.R default_variables.R mod_download_dataset.R config_class.R doc-data.R sidebar.R utils.R code_used_by_external_modules.R timeline_process.R timeline_pipeline.R core_process.R core_singleProcess.R core_pipeline.R general_core_funcs.R mod_modalDialog.R mod_format_DT.R mod_infos_dataset.R mod_history_dataset.R mod_insert_md.R mod_homepage.R mod_view_dataset.R globals.R mod_open_dataset.R mod_popover_for_help.R mod_release_notes.R mod_mainapp.R utils2.R

Encoding UTF-8

VignetteBuilder knitr

StagedInstall yes

LazyLoad yes

Config/roxygen2/version 8.0.0

git_url https://git.bioconductor.org/packages/MagellanNTK

git_branch devel

git_last_commit 36b4372

git_last_commit_date 2026-09-07

Repository Bioconductor 3.24

Date/Publication 2026-11-19

Author Samuel Wieczorek [aut] (ORCID: <https://orcid.org/0000-0002-5016-1203>),
Thomas Burger [aut],
Manon Gaudin [cre, aut]

Maintainer Manon Gaudin <manon.gaudin@cea.fr>

Contents

Add2History . . . 3
BuildData2Send . . . 4
call_func . . . 5
choose_dir . . . 5
Config-class . . . 6
dataset-processing . . . 8
default_vars . . . 10
directoryInput . . . 10
Discover_Skipped_Steps . . . 11
download_dataset . . . 12
format_DT . . . 13
generic_mod_open_dataset . . . 14
GetExtension . . . 15
GetFirstMandatoryNotValidated . . . 16
GetHistory . . . 17
GetListDatasets . . . 17
GetMaxValidated_AllSteps . . . 18
GetMaxValidated_BeforePos . . . 18
GetPackageVersion . . . 19
globalvariables . . . 19
history_dataset . . . 20
infos_dataset . . . 21
initComplete . . . 22
InitializeHistory . . . 22
insertCodeForExternalModules . . . 23
Insert_Call_to_Config . . . 25
isSubstr . . . 25
lldata . . . 26
lldata1 . . . 26
lldata12 . . . 27

lldata123 . . . 27
magellanNTK . . . 28
module_exists . . . 29
mod_err_modal . . . 29
mod_homepage . . . 30
mod_insert_md . . . 31
mod_main_page . . . 32
mod_modalDialog . . . 33
mod_open_workflow . . . 34
mod_popover_for_help . . . 35
mod_release_notes . . . 35
mod_sweetAlert . . . 36
NavPage . . . 37
nav_pipeline . . . 37
nav_process . . . 39
nav_single_process . . . 40
pkgsRequire . . . 42
process_layout . . . 42
readConfigFile . . . 43
readDirectoryInput . . . 44
runDirinputExample . . . 44
SetCurrentPosition . . . 45
SetHistory . . . 45
show,Config-method . . . 46
sidebars . . . 47
source_shinyApp_files . . . 47
source_wf_files . . . 48
timelines . . . 48
Timestamp . . . 50
ToggleState_NavBtns . . . 50
ToggleState_Screens . . . 51
toggleWidget . . . 52
updateDirectoryInput . . . 52
UpdateStepsStatus . . . 53
Update_State_Screens . . . 54
view_dataset . . . 54

Index **56**

`Add2History` *Get the last validated step before current position.*

Description

This function returns the indice of the last validated step before the current step.

Usage

```
Add2History(history, step, substep, param.name, value)
```

Arguments

history	A ‘data.frame()‘
step	A ‘character()‘
substep	A ‘character()‘
param.name	A ‘character()‘
value	The value corresponding to the param.name

Value

A ‘data.frame()‘

Examples

```
history <- InitializeHistory()
Add2History(history, "Example step", "First sub-step", "my param", "THE value")
```

BuildData2Send *Builds a vector of data*

Description

The names of the slots in dataIn are a subset of the names of the steps (names(stepsNames)). Each item is the result of a process and when a process has been validated, it creates a new slot with its own name

Usage

```
BuildData2Send(session, dataIn, stepsNames)
```

Arguments

session	internal parameter
dataIn	An instance of an object of type ‘list()‘.
stepsNames	A vector in which items is the name of a step in the pipeline.

Value

A vector of the same length of the vector ‘stepsNames‘ in which each item is an object (the type of object used by the pipeline). This object must have the same behavior of a ‘list()‘.

Examples

```
data(lldata)
stepsNames <- c("data1", "data2", "data3")
BuildData2Send(NULL, lldata, stepsNames)

stepsNames <- c("data1", "data2", "data3", "data4", "data5")
BuildData2Send(NULL, lldata, stepsNames)
```

call_func *Wrapper to the function 'do.call'*

Description

Wrapper to the function 'do.call'

Usage

```
call_func(fname, args)
```

Arguments

fname	The name of the function to execute
args	The 'list()' of its arguments

Value

The result of the function called

See Also

[do.call()]

Examples

```
call_func("stats::rnorm", list(n = 10, mean = 3))
```

choose_dir *Choose a Folder Interactively*

Description

This code is heavily based on the following repository: https://github.com/wleepang/shiny-directory-input Display an OS-native folder selection dialog under Mac OS X, Linux GTK+ or Windows.

Given x and y, return y only if both x and y are set

Usage

```
x %AND% y

file_sep(sysinfo = Sys.info())

choose_dir(default = NA, caption = NA, useNew = TRUE, sysinfo = Sys.info())

choose_dir.darwin(default = NA, caption = NA)

choose_dir.linux(default = NA, caption = NA)

choose_dir.windows(default = NA, caption = NA, useNew = TRUE)
```

Arguments

x	left operand
y	right operand
sysinfo	System inforamtions
default	which folder to show initially
caption	the caption on the selection dialog
useNew	boolean, selects the type of dialog shown in windows

Details

Uses an Apple Script, Zenity or Windows Batch script to display an OS-native folder selection dialog.

For Apple Script, with `default = NA`, the initial folder selection is determined by default behavior of the "choose folder" script. Otherwise, paths are expanded with `path.expand`.

For Linux, with `default = NA`, the initial folder selection is determined by defaul behavior of the zenity script.

The new windows batch script allows both initial folder and caption to be set. In the old batch script for Windows the initial folder is always ignored.

Value

A length one character vector, character NA if 'Cancel' was selected.

A length one character vector, character NA if 'Cancel' was selected.

A length one character vector, character NA if 'Cancel' was selected.

A length one character vector, character NA if 'Cancel' was selected.

See Also

`choose_dir`

Examples

```
NULL
```

Config-class *Config class definition*

Description

This class is used to store the configuration of any process used with MagellanNTK It contains the following slots:

Wrapper function to the constructor of the class

Usage

```
Config(
  fullname = "",
  mode = "",
  steps = NULL,
  mandatory = NULL,
  steps.source.file = NULL
)

## S4 method for signature 'Config'
initialize(.Object, fullname, mode, steps, mandatory, steps.source.file)

Config(
  fullname = "",
  mode = "",
  steps = NULL,
  mandatory = NULL,
  steps.source.file = NULL
)
```

Arguments

`fullname`	The complete name of the pipeline or the process (in this case, it is the concatenation of the name of the pipeline and the name of the process itself, separated by '_')
`mode`	A 'character()' which indicates whether the current module is used as a 'process' nor a 'pipeline.' Default value is NULL
`steps`	A vector containing the names of steps in a process or a pipeline.
`mandatory`	A vector of boolean where each element indicates whether the corresponding steps is mandatory or not. It has the same length of the vector steps.
`steps.source.file`	A 'vector' of which each item is the source code of the corresponding step.
`.Object`	The object

Value

An instance of the class 'Config'

Slots

`fullname` The name of the process which is the concatenation of the name of the pipeline and the name of the process itself, separated by '_'.

`name` The name of the process nor pipeline.

`parent` The name of the pipeline/process which owns this instance

`mode` A 'character()' which indicates if the configuration is about a whole process nor a process of a pipeline.

`steps` A 'vector' of 'character()' which contains the primary steps of the pipeline. Two steps will be automatically inserted in this vector: 'Description 'in the first position and 'Save' at the end.

`mandatory` A ‘vector‘ of ‘boolean()‘ in which each item is the necessary code for the GUI of a step. The size of this vector is the number of steps

`ll.UI` A ‘vector‘ of Shiny source code. Each item is the necessary code for the GUI of a step. The size of this vector is the number of steps

`steps.source.file` A ‘vector‘ of which each item is the source code of the corresponding step.

`steps` A ‘vector‘ of ‘character()‘ which contains the primary steps of the pipeline. Two steps will be automatically inserted in this vector: ’Description ’in the first position and ’Save’ at the end.

`mandatory` A ‘vector‘ of ‘boolean()‘ in which each item is the necessary code for the GUI of a step. The size of this vector is the number of steps

Examples

```
# Example of a single process with one step
proc1step <- Config(
  fullname = "PipelineDemo_DataGeneration",
  mode = "process",
  steps = c("DataGeneration"),
  mandatory = c(TRUE)
)


# Example of a single process with two steps
proc2steps <- Config(
  fullname = "PipelineDemo_PreProcessing",
  mode = "process",
  steps = c("Filtering", "Normalization"),
  mandatory = c(TRUE, FALSE)
)

# Example of pipeline with three process
pipe3proc <- Config(
  mode = "pipeline",
  fullname = "PipelineDemo",
  steps = c("DataGeneration", "Preprocessing", "Clustering"),
  mandatory = c(TRUE, FALSE, FALSE)
)


# Example of a particular description module (A process with no step)
description.process <- Config(
  fullname = "PipelineDemo_Description",
  mode = "process",
  steps = "",
  mandatory = ""
)
```

`dataset-processing` *Datasets processing*

Description

The following functions are currently available:

- ‘keepDatasets(object, range)‘ keep datasets in object which are in range

- ‘addDatasets(object, dataset, name)‘ add the ’dataset’ to the object (of type list)

This manual page describes manipulation methods using [list] objects. In index or name ‘i‘ can be specified to define the array (by name of index) on which to operate.

The following functions are currently available:

- ‘keepDatasets(object, range)‘ keep datasets in object which are in range

- ‘addDatasets(object, dataset, name)‘ add the ’dataset’ to the object (of type list)

- ‘Save(object, file)‘ stores the object to a .RData file

Usage

```
addDatasets(object, dataset, name)

keepDatasets(object = NULL, range = seq_along(object))
```

Arguments

object	An object of class ‘list‘.
dataset	‘character()‘ providing the base with respect to which logarithms are computed. Default is log2.
name	A ‘character()‘ naming the new array name.
range	A interval of integers

Details

The object must be of type list. Thetwo functions are implemented here for of the package which uses MagellanNTK

The object must be of type list. Thetwo functions are implemented here for of the package which uses MagellanNTK

Value

An instance of class ‘MultiAssayExperiment‘.

An processed object of the same class as ‘object‘.

Examples

```
data(lldata)
obj.se <- lldata[[1]]
new.obj <- addDatasets(lldata, obj.se, "mynewobj")

data(lldata123)
keepDatasets(lldata123, c(1, 3))

NULL
```

default_vars *Default variables used in MagellanNTK*

Description

left_panel = left_sidebar + width_sidebar top_sidebar = top_panel width_sidebar = width_process_btns

Usage

```
default_funcs()

default_base_URL()

default_workflow()

default_layout

default_theme(mode)
```

Arguments

mode	A ‘character()‘ to specifies the running mode of MagellanNTK: ’user’ (default) or ’dev’.

Value

NA

Examples

```
NULL
```

directoryInput *Directory Selection Control*

Description

Directory Selection Control

Usage

```
directoryInput(inputId, label, value = NULL)
```

Arguments

inputId	The input slot that will be used to access the value
label	Display label for the control, or NULL for no label
value	Initial value. Paths are expanded via path.expand.

Details

This widget relies on choose_dir to present an interactive dialog to users for selecting a directory on the local filesystem. Therefore, this widget is intended for shiny apps that are run locally - i.e. on the same system that files/directories are to be accessed - and not from hosted applications (e.g. from shinyapps.io).

Value

A directory input control that can be added to a UI definition.

See Also

updateDirectoryInput, readDirectoryInput, choose_dir

Examples

```
NULL
```

Discover_Skipped_Steps *Discover Skipped Steps*

Description

Discover Skipped Steps

Usage

```
Discover_Skipped_Steps(steps.status)
```

Arguments

steps.status	A vector of 'character()' which reflects the status of the steps in the pipeline. Thus, the length of this vector is equal to the number of steps

Value

A vector of integers of the same length as steps.status and where skipped steps are identified with '-1'

Examples

```
steps <- c(1, 1, 0, 1)
Discover_Skipped_Steps(steps)
```

download_dataset *dl*

Description

A shiny Module.

Usage

```
download_dataset_ui(id)

download_dataset_server(
  id,
  dataIn = reactive({
     NULL
 }),
  filename = "myDataset",
  remoteReset = reactive({
     0
 }),
  is.enabled = reactive({
     TRUE
 })
)

download_dataset(dataIn = NULL, filename = "myDataset")
```

Arguments

id	A 'character()' as the id of the Shiny module
dataIn	An instance of the class 'MultiAssayExperiment'
filename	internal
remoteReset	A 'logical(1)' which acts as a remote command to reset the module to its default values. Default is FALSE.
is.enabled	A 'boolean'. This variable is used as a remote command to specify if the corresponding module is enabled (TRUE) or disabled (FALSE).

Value

NA

Examples

```
if (interactive()) {
  data(lldata, package = "MagellanNTK")
  download_dataset(lldata)
}
```

format_DT *format_DT_ui and format_DT_server*

Description

A shiny Module.

Usage

```
format_DT_ui(id)

format_DT_server(
  id,
  dataIn = reactive({
     NULL
 }),
  hidden = reactive({
     NULL
 }),
  withDLBtns = FALSE,
  showRownames = FALSE,
  dom = "Bt",
  max.rows = 20,
  hc_style = reactive({
     NULL
 }),
  remoteReset = reactive({
     0
 }),
  is.enabled = reactive({
     TRUE
 })
)

format_DT(
  dataIn,
  hidden = NULL,
  withDLBtns = FALSE,
  showRownames = FALSE,
  dom = "Bt",
  hc_style = NULL,
  remoteReset = NULL,
  is.enabled = TRUE
)
```

Arguments

id	A ‘character()‘ as the id of the Shiny module
dataIn	A ‘data.frame()‘
hidden	Default is NULL,

withDLBtns	Default is FALSE,
showRownames	Default is FALSE,
dom	Default is 'Bt',
max.rows	Default is 20,
hc_style	Default is NULL,
remoteReset	An 'integer' which acts as a remote command to reset the module. Its value is incremented on a external event and it is used to trigger an event in this module
is.enabled	Default is TRUE

Value

A shiny App

Examples

```
if (interactive()) {
  data(lldata)
  obj <- SummarizedExperiment::assay(lldata[[1]])
  obj <- as.data.frame(obj)
  format_DT(obj)

  # Compute style from within third party tab
  obj <- as.data.frame(matrix(seq_len(30), byrow = TRUE, nrow = 6))
  colnames(obj) <- paste0("col", seq_len(5))

  mask <- as.data.frame(matrix(rep(LETTERS[seq_len(5)], 6),
    byrow = TRUE, nrow = 6
  ))

  style <- list(
    cols = colnames(obj),
    vals = colnames(mask),
    unique = unique(mask),
    pal = RColorBrewer::brewer.pal(5, "Dark2")[seq_len(5)]
  )

  format_DT(obj,
    hidden = mask,
    hc_style = style
  )
}
```

---

generic_mod_open_dataset

Change the default functions in 'MagellanNTK'

Description

This module allows to change

Usage

```
open_dataset_ui(id)

open_dataset_server(
  id,
  class = NULL,
  extension = NULL,
  remoteReset = reactive({
     NULL
 }),
  is.enabled = reactive({
     TRUE
 })
)

open_dataset(extension = NULL)
```

Arguments

id	A 'character()' as the id of the Shiny module
class	The type of file to allow for other packages dataset
extension	The extension file allowed
remoteReset	An 'integer' which acts as a remote command to reset the module. Its value is incremented on a external event and it is used to trigger an event in this module
is.enabled	A 'boolean'. This variable is used as a remote command to specify if the corresponding module is enabled/disabled in the calling module of upper level. For example, if this module is disabled, then this variable is set to TRUE. Then, all the widgets will be disabled. If not, the enabling/disabling of widgets is deciding by this module.

Value

A Shiny app

Examples

```
if (interactive()) {
  open_dataset(extension = "rdata")
}
```

GetExtension *Get file extension*

Description

Get file extension

Usage

```
GetExtension(name)
```

Arguments

name	A complete filename

Value

The extension of the given filename

Examples

```
GetExtension("foo.xlsx")
```

GetFirstMandatoryNotValidated

Get the first mandatory step not validated

Description

Get the first mandatory step not validated

Usage

```
GetFirstMandatoryNotValidated(range, rv)
```

Arguments

range	A ‘vector‘ of integers. The min of this vector must be greater of equal to 0 and the max must be less or equal to the size of the object
rv	A ‘list‘ with at least two slots : * mandatory: A vector of ‘booelan‘ which indicates whther the steps are mandatory or not * steps.status: A vector of ‘interger()‘ which indicates the status of the steps

Value

An integer which is the indice of the identified step in the vector rv$steps.status

Examples

```
NULL
```

GetHistory *Get the history of an assay*

Description

Get the history of an assay

Usage

```
GetHistory(dataIn, name)
```

Arguments

dataIn	An instance of 'MultiAssayExperiment' class
name	The name of a slot in the object

Value

A 'data.frame()'

Examples

```
data(lldata123)
GetHistory(lldata123, "Clustering")
```

GetListDatasets *Get datasets from packages which owns datasets of a given class*

Description

Get datasets from packages which owns datasets of a given class

Usage

```
GetListDatasets(class = NULL, demo_package = NULL)
```

Arguments

class	A 'character()' which is the class of the datasets one looking for
demo_package	A 'character()' which specifies a particular package to search

Value

A 'vector' in which items are the name of datasets which inherits the class given in parameter

Examples

```
foo1 <- GetListDatasets()
```

GetMaxValidated_AllSteps

Get the last validated step

Description

This function returns the indice of the last validated step among all the steps

Usage

```
GetMaxValidated_AllSteps(steps.status)
```

Arguments

steps.status	A vector of strings where each item is the status of a step. The length of this vector is the same of the number of steps.

Value

A 'integer(1)' which is the indice of the value

Examples

```
GetMaxValidated_AllSteps(c(1, 1, 0, 1, 0, 0))
```

GetMaxValidated_BeforePos

Get the last validated step before current position.

Description

This function returns the indice of the last validated step before the current step.

Usage

```
GetMaxValidated_BeforePos(pos = NULL, rv)
```

Arguments

pos	A 'integer()' which is the indice of the active position.
rv	A 'list()' which stores reactiveValues()

Value

A 'integer()'

Examples

```
pos <- 3
rv <- list(steps.status = c(1, 1, 0, 1, 0, 0), current.pos = 3)
GetMaxValidated_BeforePos(pos, rv)
```

GetPackageVersion *Get the version of a package installed in the local R distribution*

Description

Get the version of a package installed in the local R distribution

Usage

```
GetPackageVersion(pkg)
```

Arguments

pkg	The name of a package

Value

A ‘character()‘

Examples

```
GetPackageVersion("MagellanNTK")
```

globalvariables *Global variables*

Description

Global constants and configuration variables used throughout the package.

Usage

```
tl_v_next_icon

stepStatus

default_pos

listBrewerPalettes
```

Value

A collection of exported constants, lists and UI configuration objects.

Examples

```
NULL
```

history_dataset *history_dataset_ui and history_dataset_server*

Description

A shiny Module which show th content of the slot 'history' in the metadata() of the last SE of the dataset

Usage

```
history_dataset_ui(id)

history_dataset_server(
  id,
  dataIn = reactive({
     NULL
 }),
  remoteReset = reactive({
     0
 }),
  is.enabled = reactive({
     TRUE
 })
)

history_dataset(obj)
```

Arguments

id	A 'character()' as the id of the Shiny module
dataIn	An instance of the class 'MultiAssayExperiment'.

Value

A shiny app

Examples

```
if (interactive()) {
  data(lldata)
  history_dataset(lldata)
}
```

infos_dataset *infos_dataset_ui and infos_dataset_server*

Description

A shiny Module.

Usage

```
infos_dataset_ui(id)

infos_dataset_server(
  id,
  dataIn = reactive({
     NULL
 }),
  remoteReset = reactive({
     0
 }),
  is.enabled = reactive({
     TRUE
 })
)

infos_dataset(obj)
```

Arguments

id	A 'character()' as the id of the Shiny module
dataIn	An instance of the class 'MultiAssayExperiment'.

Value

A shiny app

Examples

```
if (interactive()) {
  data(lldata123)
  infos_dataset(lldata123)
}
```

initComplete *Initialisation function for DT*

Description

Initialisation function for DT

Usage

```
initComplete()
```

Value

literal JavaScript code

Examples

```
initComplete()
```

InitializeHistory *Get the last validated step before current position.*

Description

This function returns the indice of the last validated step before the current step.

Usage

```
InitializeHistory()
```

Value

A ‘data.frame()‘ with four columns: ’Process’, ’Step’, ’Parameter’ and ’Value’

Examples

```
InitializeHistory()
```

`insertCodeForExternalModules` *R code to update the 'config' variable of a process module*

Description

This function generates the necessary code to modify the variable 'config' (slots steps and mandatory). It adds a 'Description' step and a TRUE value at the beginning of the 'steps' and 'mandatory' list, erases all white spaces for the names of the steps.

This function generates the necessary code to modify the variable 'config' (slots steps and mandatory). It adds a 'Description' step and a TRUE value at the beginning of the 'steps' and 'mandatory' list, erases all white spaces for the names of the steps.

This function create the source code needed inside a module to declare the reactive variable called 'widgets.default.values'. # Declaration of the variables that will contain the values of the widgets To avoid confusion, the first string is the name of the step while the second is the name of the widget

This function create the source code needed inside a module to declare the reactive variable called 'rv.custom.default.values'. # Declaration of the variables that will contain the values of the user variables. To avoid confusion, the first string is the name of the step while the second is the name of the widget

Returned value of the process

This function generates dynamically the observeEvent function for each widget

Usage

```
Get_Code_Update_Config_Variable()

Get_Code_Declare_widgets(widgets.names = NULL)

Get_Code_Declare_rv_custom(rv.custom.names = NULL)

Get_Code_for_ObserveEvent_widgets(widgets.names = NULL)

Get_Code_for_rv_reactiveValues()

Get_Code_for_dataOut()

Get_Code_for_General_observeEvents()

Get_Code_for_remoteReset(
  widgets = TRUE,
  custom = TRUE,
  dataIn = "dataIn()",
  addon = ""
)

Get_Code_for_resetting_widgets()

Get_Code_for_resetting_custom()
```

```
Module_Return_Func()

AdditionnalCodeForExternalModules(w.names = NULL, rv.custom.names = NULL)

Get_Workflow_Core_Code(
  mode = NULL,
  name = NULL,
  w.names = NULL,
  rv.custom.names = NULL
)

Get_AdditionalModule_Core_Code(w.names = NULL, rv.custom.names = NULL)
```

Arguments

widgets.names	A 'vector' containing the names of the widgets in all steps of the module.
rv.custom.names	A 'vector' containing the names of the custom variables in all steps of the module.
widgets	A 'list' containing the names of the widgets in all steps of the module with their default values
custom	A 'list' of the custom variables used in the process. Each custom variable is accompanied with its default value
dataIn	An instance of the 'MultiAssayExperiment' or 'SummarizedExperiment' classes
addon	Additional code for the 'observeEvent()' function which catches the 'remoteReset()' action.
w.names	Same as widgets.names
mode	A 'character()' which indicates whether the current module is used as a 'process' nor a 'pipeline.' Default value is NULL
name	The Shiny id of the process or pipeline

Value

A 'string' containing some R code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'string' containing some R code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

A 'character()' containing R source code

Author(s)

Samuel Wieczorek

Examples

```
MagellanNTK::Get_AdditionalModule_Core_Code("PipelineDemo", "")
```

Insert_Call_to_Config *Insert source code for config()*

Description

Insert source code for config()

Usage

```
Insert_Call_to_Config(name)
```

Arguments

name	The name of a pipeline nor a process

Value

A 'character()' containing R source code

Examples

```
Insert_Call_to_Config("PipelineDemo")
```

---

isSubstr *Substring test*

Description

Substring test

Usage

```
isSubstr(pattern, target)
```

Arguments

pattern	A 'character()'
target	The 'character()' to look for into the pattern

Value

A boolean

Examples

```
isSubstr("cde", "abcdefghi")
```

`lldata` *lldata dataset*

Description

An instance of the class 'MultiAssayExperiment' which contains one assay (instance of class 'SummarizedExperiment'): 'Convert': Contains a data.frame with 6 columns and 100 rows fullfilled with '0'. It also contains a slot 'history' in the metadata(). This dataset is the most bacis one to run the example pipeline and learn how to work with MagellanNTK

Usage

```
data(lldata)
```

Format

An object of class 'MultiAssayExperiment'

`lldata1` *lldata1 dataset*

Description

An instance of the class 'MultiAssayExperiment' which contains two assays (instance of class 'SummarizedExperiment'): 'Convert': Contains a data.frame with 6 columns and 100 rows fullfilled with '0'. 'DataGeneration' which stores the result of the 'DataGeneration' step of the demo pipeline Each assay contains a slot 'history' in the metadata().

Usage

```
data(lldata1)
```

Format

An object of class 'MultiAssayExperiment'

lldata12 *lldata12 dataset*

Description

An instance of the class 'MultiAssayExperiment' which contains two assays (instance of class 'SummarizedExperiment'): 'Convert': Contains a data.frame with 6 columns and 100 rows full-filled with '0'. 'DataGeneration' which stores the result of the 'DataGeneration' step of the demo pipeline 'Preprocessing' which stores the result of the 'DataGeneration' and 'Preprocessing' steps of the demo pipeline Each assay contains a slot 'history' in the metadata().

Usage

```
data(lldata12)
```

Format

An object of class 'MultiAssayExperiment'

lldata123 *lldata123 dataset*

Description

An instance of the class 'MultiAssayExperiment' which contains two assays (instance of class 'SummarizedExperiment'): 'Convert': Contains a data.frame with 6 columns and 100 rows full-filled with '0'. 'DataGeneration' which stores the result of the 'DataGeneration' step of the demo pipeline 'Preprocessing' which stores the result of the 'DataGeneration' and 'Preprocessing' steps of the demo pipeline 'Clustering' which stores the result of all the three steps of the demo pipeline Each assay contains a slot 'history' in the metadata().

Usage

```
data(lldata123)
```

Format

An object of class 'MultiAssayExperiment'

`magellanNTK` *Main Shiny application*

Description

Main Shiny application

Usage

```
MagellanNTK_ui(id, sidebarSize = "medium")

MagellanNTK_server(
  id,
  workflow.path = reactive({
     NULL
 }),
  workflow.name = reactive({
     NULL
 }),
  verbose = FALSE,
  usermod = "user"
)

MagellanNTK(
  workflow.path = NULL,
  workflow.name = NULL,
  verbose = FALSE,
  usermod = "user",
  sidebarSize = "medium",
  ...
)
```

Arguments

`id`	A 'character()' as the id of the Shiny module
`sidebarSize`	The width of the sidebar. Available values are 'small', 'medium', 'large'
`workflow.path`	A 'character()' which is the path to the directory which contains the files and directories of the pipeline.
`workflow.name`	A 'character()' which is the name of the pipeline to launch and run whithin the framework of MagellanNTK. It designs the name of a directory which contains the files and directories of the pipeline.
`verbose`	A 'boolean' to indicate whether to turn off (FALSE) or ON (TRUE) the verbose mode for logs.
`usermod`	A 'character()' to specifies the running mode of MagellanNTK: 'user' (default) or 'dev'.
`...`	Additional parameters

Value

A shiny app

Examples

```
if (interactive()) {
  wf.path <- system.file('workflow/PipelineDemo', package = 'MagellanNTK')
  MagellanNTK(wf.path, 'PipelineDemo')
}
```

`module_exists` *Checks if a Shiny module is loaded*

Description

This function checks if the ui() and server() parts of a Shiny module are available in the global environment.

Usage

```
module_exists(base_name)
```

Arguments

`base_name`	The name of the module (without '_ui' nor '_server' suffixes)

Value

A boolean

Examples

```
path <- system.file("workflow/PipelineDemo", package = "MagellanNTK")
source_wf_files(path)
module_exists("PipelineDemo_Preprocessing")
```

`mod_err_modal` *A modal window to display messages*

Description

This module is not directly used by MagellanNTK core functions. It is rather a useful tool for third part pipelines and processes.

Usage

```
mod_errorModal_ui(id)

mod_errorModal_server(
  id,
  title = NULL,
  text = NULL,
  footer = modalButton("Close")
)

mod_errorModal(title = NULL, text = NULL)
```

Arguments

id	A ‘character()‘ as the id of the Shiny module
title	A ‘character()‘ as the title of the modal
text	A ‘character()‘ as the content of the modal
footer	The content of the footer . May be UI content

Value

A shiny App

Examples

```
if (interactive()) {
  mod_errorModal("myTitle", "myContent")
}
```

---

mod_homepage *mod_homepage_ui and mod_homepage_server*

Description

A shiny Module.

Usage

```
mod_homepage_ui(id)

mod_homepage_server(
  id,
 mdfile = file.path(system.file("www/md", package = "MagellanNTK"), "Presentation.Rmd"),
  dataset = reactive({
     NULL
 })
)

mod_homepage()
```

Arguments

id	A ‘character()‘ as the id of the Shiny module
mdfile	The path to the Rmd file which describes the pipeline
dataset	An instance of the class ‘MultiAssayExperiment‘

Value

A shiny App

Examples

```
if (interactive()) {
  mod_homepage()
}
```

---

mod_insert_md *insert_md_ui and insert_md_server*

Description

A shiny Module.

Usage

```
insert_md_ui(id)

insert_md_server(id, url)

insert_md(url)
```

Arguments

id	A ‘character()‘ as the id of the Shiny module
url	The path to the Rmd file to display. It can be a path to a file on the computer or a link to a file over internet.

Value

A shiny App

Examples

```
if (interactive()) {
  base <- system.file("www/md", package = "MagellanNTK")
  url <- file.path(base, "Presentation.Rmd")
  insert_md(url)
}
```

mod_main_page *mod_main_page_ui and mod_loading_page_server*

Description

A shiny Module.

Usage

```
mainapp_ui(id, session, size = "300px")

mainapp_server(
  id,
  workflow.name = reactive({
     NULL
 }),
  workflow.path = reactive({
     NULL
 }),
  verbose = FALSE,
  usermod = "user"
)

mainapp(usermod = "user")
```

Arguments

id	A 'character()' as the id of the Shiny module
session	shiny internal
size	The width of the sidebar.in pixels
workflow.name	A 'character()' which is the name of the pipeline to launch and run whithin the framework of MagellanNTK. It designs the name of a directory which contains the files and directories of the pipeline.
workflow.path	A 'character()' which is the path to the directory which contains the files and directories of the pipeline.
verbose	A 'boolean' to indicate whether to turn off (FALSE) or ON (TRUE) the verbose mode for logs.
usermod	A 'character()' to specifies the running mode of MagellanNTK: 'user' (default) or 'dev'.

Value

A shiny App

Examples

```
if (interactive()) {
  mainapp()
}
```

`mod_modalDialog` *Predefined modal with embeded shiny module*

Description

Displays of formatted modal-dialog with 'Cancel' and 'Ok' buttons. This modal (a shiny module itself) can embed a shiny module and it is able to return the return value of the embeded shiny module.

Usage

```
mod_modalDialog_ui(id)

mod_modalDialog_server(
  id,
  title = NULL,
  typeWidget = "button",
  width = NULL,
  uiContent = NULL,
  external_mod = NULL,
  external_mod_args = list()
)

mod_modalDialog(
  title,
  typeWidget = "button",
  uiContent = NULL,
  external_mod = NULL,
  external_mod_args = list()
)
```

Arguments

`id`	A 'character(1)' which is the id of the instance of the module
`title`	A 'character(1)'
`typeWidget`	The type of command widget to show to open the modal. Available values are 'button' (default) and 'link'.
`width`	A 'character(1)' indicating the size of the modal window. Can be "s" for small (the default), "m" for medium, or "l" for large.
`uiContent`	The content of the modal dialog. By default, its value is NULL which means that the user wants to embed a shiny module. In this case, the module rather take into account the two next parameters.
`external_mod`	The name of a shiny module which is already loaded in the R session. This name corresponds to the name of the functions '_ui' and '_server' without these suffixes. For example, if the user wants to add a module defined by the functions 'myModule_ui()' and 'myModule_server()', then the parameter 'external_mod' should be set to 'myModule'.
`external_mod_args`	A 'list()' which contains the parameters sent to the function 'myModule_server()'.

Value

A Shiny modal-dialog

Examples

```
if (interactive()) {
  mod_modalDialog(
    title = "test modalDialog",
    uiContent = p("test")
  )
}
```

`mod_open_workflow` *mod_open_workflow_ui and mod_open_workflow_server*

Description

A shiny Module.

Usage

```
open_workflow_ui(id)

open_workflow_server(id)

open_workflow()
```

Arguments

`id`	A ‘character()‘ as the id of the Shiny module

Value

A shiny App

Examples

```
if (interactive()) {
  open_workflow()
}
```

mod_popover_for_help *Opens a small tooltip info over a widget.*

Description

Actually, this module does not work because we do not allow the use of the package 'shinyBS' package (conflicts with BS versions). In the future, one will fix this module with native functions in the package 'bs4Dash' (https://bs4dash.rinterface.com/reference/tooltip).

Usage

```
mod_popover_for_help_ui(id)

mod_popover_for_help_server(id, title, content)

mod_popover_for_help(title, content)
```

Arguments

id	A 'character()' as the id of the Shiny module
title	The title of the tooltip window
content	The main text of the tooltip window

Value

A shiny App

Examples

```
if (interactive()) {
  mod_popover_for_help("myTitle", "myContent")
}
```

mod_release_notes *mod_release_notes_ui and mod_release_notes_server*

Description

A shiny Module.

Usage

```
mod_release_notes_ui(id)

mod_release_notes_server(id, URL_releaseNotes)

mod_release_notes(URL_releaseNotes)
```

Arguments

id	A 'character()' as the id of the Shiny module
URL_releaseNotes	The path to the Rmd file to display. It can be a path to a file on the computer or a link to a file over internet.

Value

A shiny App

Examples

```
if (interactive()) {
  url <- "http://www.prostar-proteomics.org/md/versionNotes.md"
  mod_release_notes(url)

  local.url <- system.file("/workflow/PipelineDemo/md/links.Rmd",
    package = "MagellanNTK"
  )
  mod_release_notes(local.url)
}
```

---

mod_sweetAlert *A shiny app to display a modal*

Description

This module is not directly used by MagellanNTK core functions. It is rather a useful tool for third part pipelines and processes.

Usage

```
mod_SweetAlert_ui(id)

mod_SweetAlert_server(
  id,
  title = NULL,
  text = NULL,
  showClipBtn = TRUE,
  type = "error"
)

mod_SweetAlert(title, text, type = "warning")
```

Arguments

id	A 'character()' as the id of the Shiny module
title	The title to be displayed in the window
text	The titmessage to be displayed in the window
showClipBtn	A 'boolean' to indicates if the copy to clipboard button is shown or not.
type	A 'character()' for the type of message: 'error', 'warning', etc.

Value

A shiny App

Examples

```
if (interactive()) {
  mod_SweetAlert("my title", "my message")
}
```

`NavPage` *Move the position of the cursor*

Description

Move the position of the cursor

Usage

```
NavPage(direction, current.pos, len)
```

Arguments

`direction`	A 'integer(1)' which is the direction in the timeline: forward (1), backwards (-1).
`current.pos`	A 'integer(1)' which is the current position.
`len`	A 'integer(1)' which is the number of steps in the process.

Value

A 'integer(1)' which is the new current position.

Examples

```
NavPage(-1, 5, 5)
```

`nav_pipeline` *Shiny module 'nav_pipeline()'*

Description

The module navigation can be launched via a Shiny app. This is the core module of MagellanNTK

Usage

```
nav_pipeline_ui(id)

nav_pipeline_server(
  id = NULL,
  dataIn = reactive({
     NULL
 }),
  is.enabled = reactive({
     TRUE
 }),
  remoteReset = reactive({
     0
 }),
  verbose = FALSE,
  usermod = "user"
)

nav_pipeline()
```

Arguments

id	A 'character(1)' which defines the id of the module. It is the same as for the server() function.
dataIn	An instance of the class 'MultiAssayExperiment'
is.enabled	A 'boolean'. This variable is a remote command to specify if the corresponding module is enabled/disabled in the calling module of upper level. For example, if this module is part of a pipeline and the pipeline calculates that it is disabled (i.e. skipped), then this variable is set to TRUE. Then, all the widgets will be disabled. If not, the enabling/disabling of widgets is deciding by this module.
remoteReset	An 'integer' which acts as a remote command to reset the module. Its value is incremented on a external event and it is used to trigger an event in this module
verbose	A 'boolean' to indicate whether to turn off (FALSE) or ON (TRUE) the verbose mode for logs.
usermod	A 'character()' to specifies the running mode of MagellanNTK: 'user' (default) or 'dev'.

Value

A list of four items: * dataOut A dataset of the same class of the parameter dataIn * steps.enabled A vector of 'boolean' of the same length than config@steps * status A vector of 'integer(1)' of the same length than the config@steps vector

Author(s)

Samuel Wieczorek

Examples

```
if (interactive()) {
  library(shiny)
  nav_pipeline()
```

```
}
```

`nav_process` *The server() function of the module 'nav_process'*

Description

The module navigation can be launched via a Shiny app. This is the core module of MagellanNTK

Usage

```
nav_process_ui(id)

nav_process_server(
  id = NULL,
  dataIn = reactive({
     NULL
 }),
  status = reactive({
     NULL
 }),
  is.enabled = reactive({
     TRUE
 }),
  remoteReset = reactive({
     0
 }),
  remoteResetUI = reactive({
     0
 }),
  is.skipped = reactive({
     FALSE
 }),
  verbose = FALSE,
  usermod = "user"
)

nav_process()
```

Arguments

`id`	A 'character(1)' which defines the id of the module. It is the same as for the ui() function.
`dataIn`	An instance of the 'SummarizedExperiment' class
`status`	A boolean which indicates whether the current status of the process
`is.enabled`	A 'boolean'. This variable is a remote command to specify if the corresponding module is enabled/disabled in the calling module of upper level. For example, if this module is part of a pipeline and the pipeline calculates that it is disabled (i.e. skipped), then this variable is set to TRUE. Then, all the widgets will be disabled. If not, the enabling/disabling of widgets is deciding by this module.

remoteReset	An ‘integer‘ which acts as a remote command to reset the module. Its value is incremented on a external event and it is used to trigger an event in this module
remoteResetUI	An ‘integer‘ which acts as a remote command to reset the UI part of the module Its value is incremented on a external event and it is used to trigger an event in the module
is.skipped	A ‘boolean‘ which indicates whether the pipeline or process is skipped (TRUE) or not (FALSE)
verbose	A ‘boolean‘ to indicate whether to turn off (FALSE) or ON (TRUE) the verbose mode for logs.
usermod	A ‘character()‘ to specifies the running mode of MagellanNTK: ’user’ (default) or ’dev’.

Value

A shiny App

Author(s)

Samuel Wieczorek

Examples

```
if (interactive()) {
  nav_process()
}
```

`nav_single_process` *Shiny module ‘nav_single_process()‘*

Description

The module navigation can be launched via a Shiny app. This is the core module of MagellanNTK

Usage

```
nav_single_process_ui(id)

nav_single_process_server(
  id = NULL,
  dataIn = reactive({
     NULL
 }),
  status = reactive({
     NULL
 }),
  remoteResetUI = reactive({
     0
 }),
  is.enabled = TRUE,
  is.skipped = FALSE,
```

```
  verbose = FALSE,
  usermod = "user",
  sendDataIfReset = TRUE
)

nav_single_process()
```

Arguments

id	A 'character(1)' which defines the id of the module. It is the same as for the server() function.
dataIn	An instance of the 'SummarizedExperiment' class
status	A boolean which indicates whether the current status of the process
remoteResetUI	An 'integer' which acts as a remote command to reset the UI part of the module Its value is incremented on a external event and it is used to trigger an event in the module
is.enabled	A 'boolean'. This variable is used as a remote command to specify if the corresponding module is enabled/disabled in the calling module of upper level. For example, if this module is disabled, then this variable is set to TRUE. Then, all the widgets will be disabled. If not, the enabling/disabling of widgets is deciding by this module.
is.skipped	A 'boolean' which indicates whether the pipeline or process is skipped (TRUE) or not (FALSE)
verbose	A 'boolean' to indicate whether to turn off (FALSE) or ON (TRUE) the verbose mode for logs.
usermod	A 'character()' to specifies the running mode of MagellanNTK: 'user' (default) or 'dev'.
sendDataIfReset	A 'boolean' to indicate if the reseted value must be send to the caller in case of reseting the process. This is usefule for example for the Convert process

Value

A shiny App

Author(s)

Samuel Wieczorek

Examples

```
if (interactive()) {
  nav_single_process()
}
```

pkgsRequire *Loads packages*

Description

Checks if a package is available to load it

Checks if a package is available to load it

Usage

```
pkgsRequire(ll.deps)

pkgsRequire(ll.deps)
```

Arguments

ll.deps	A vector of 'character()' which contains packages names

Value

NA

Author(s)

Samuel Wieczorek

Examples

```
NULL


pkgsRequire(c("stats"))
```

process_layout *Layout for workflows*

Description

Creates a layout to place the sidebar and the main screen for pipelines and processes.

Usage

```
process_layout(session, ns, sidebar, content)

process_layout_process(session, ns, sidebar, content)

process_layout_pipeline(session, ns, sidebar, content)
```

Arguments

session	The R session parameter
ns	Namespaced IDs for inputs/outputs
sidebar	Shiny components
content	Shiny components

Value

A ‘div()‘ containing shiny components

Examples

```
NULL
```

---

readConfigFile *Read pipelines configuration files*

Description

Read the configuration file of a pipeline and extract formatted (as a list) information

Usage

```
readConfigFile(path, usermod = "user")
```

Arguments

path	A ‘character()‘ which is the path to the directory which contains the files and directories of the pipeline.
usermod	A ‘character()‘ to specifies the running mode of MagellanNTK: ’user’ (default) or ’dev’.

Value

A list containing the following items: value <- list(* funcs = tmp, * verbose = get_data(lines, "verbose") == "enabled", * UI_view_debugger: a ‘boolean‘ to show (TRUE) of hide (FALSE) the debugger interface * UI_view_open_pipeline: a ‘boolean‘ to show (TRUE) of hide (FALSE) the UI to open a pipeline * UI_view_convert_dataset a ‘boolean‘ to show (TRUE) of hide (FALSE) the UI for convert/import a dataset * UI_view_change_Look_Feel a ‘boolean‘ to show (TRUE) of hide (FALSE) the UI to change the L&F * UI_view_change_core_funcs a ‘boolean‘ to show (TRUE) of hide (FALSE) the UI to change the generic functions * extension A ‘character()‘ which specifies the extension file allowed to be processed in the pipeline, * class A ‘character()‘ which specifies the class of the dataset to be processed in the pipeline, * package A ‘character()‘ which specifies the package which owns the pipeline * demo_package A ‘character()‘ which specifies a particular package to search * URL_manual The path to the Rmd file containing the user manual of the pipeline. It can be a path to a file on the computer or a link to a file over internet. * URL_ReleaseNotes The path to the Rmd file containing release notes about the pipeline. It can be a path to a file on the computer or a link to a file over internet.

Examples

```
path <- system.file("workflow/PipelineDemo", package = "MagellanNTK")
readConfigFile(path)
```

`readDirectoryInput` *Read the value of a directoryInput*

Description

Read the value of a directoryInput

Usage

```
readDirectoryInput(session, inputId)
```

Arguments

`session`	The `session` object passed to function given to `shinyServer`.
`inputId`	The id of the input object

Details

Reads the value of the text field associated with a `directoryInput` object that stores the user selected directory path.

Value

NA

Examples

```
NULL
```

`runDirinputExample` *Runs a demo app with the code*

Description

Runs a demo app with the code

Usage

```
runDirinputExample()
```

Details

Runs a demo app with the code. See the code in the example_application folder for how this is accomplished.

Value

NA

Examples

```
if (interactive()) {
  shiny::runApp(runDirinputExample())
}
```

`SetCurrentPosition` *Get the position of the last validated item*

Description

Get the position of the last validated item

Usage

```
SetCurrentPosition(stepsstatus)
```

Arguments

`stepsstatus`	A vector of integers which reflects the status of the steps in the pipeline. Thus, the length of this vector is equal to the number of steps

Value

An integer

Examples

```
status <- c(1, 1, 1, 0, 0)
SetCurrentPosition(status)

status <- c(1, 1, 0, 1, 0)
SetCurrentPosition(status)
```

`SetHistory` *Standardize names*

Description

Standardize names

Usage

```
SetHistory(obj.se, history)
```

Arguments

obj.se	An instance of the class 'SummarizedExperiment'
history	A 'data.frame()'

Value

An instance of the class 'SummarizedExperiment'

Author(s)

Samuel Wieczorek

Examples

```
data(lldata)
history <- GetHistory(lldata, "Convert")
history <- Add2History(history, "Step example", "Substep example", "ex_param", "Ex")
lldata[[1]] <- SetHistory(lldata[[1]], history)
```

---

show,Config-method *Prints the content of a 'Config' class*

Description

Prints the content of a 'Config' class

Usage

```
## S4 method for signature 'Config'
show(object)
```

Arguments

object	An instance of the class 'Config'

Value

NA

sidebars *Sidebar functions*

Description

These functions content the UI code for the main sidebar in the interface of MagellanNTK

Usage

```
Insert_User_Sidebar()
```

Value

Shiny UI component

Examples

```
Insert_User_Sidebar()
```

source_shinyApp_files *Source ui.R, server.R ang global.R files*

Description

Loads the source code of basic Shiny app files

Usage

```
source_shinyApp_files()
```

Value

NA

Examples

```
if (interactive()) {
  source_shinyApp_files()
}
```

source_wf_files *Source workflow files*

Description

This function goes into the 'R' directory of the file structure of a pipeline and load into memory the source code of the R scripts (which contains the interfaces of the process)

Usage

```
source_wf_files(dirpath, verbose = FALSE)
```

Arguments

dirpath	A 'character()' which is the path to the directory which contains the files and directories of the pipeline.
verbose	A 'boolean' to indicate whether to turn off (FALSE) or ON (TRUE) the verbose mode for logs.

Value

NA

Examples

```
path <- system.file("workflow/PipelineDemo", package = "MagellanNTK")
source_wf_files(path)
```

timelines *Shiny module for the process timeline*

Description

Define the appearance of the pipeline timeline.

Define the appearance of the pipeline timeline.

Usage

```
timeline_process_ui(id)

timeline_process_server(
  id,
  config,
  status = reactive({
     NULL
 }),
  position = reactive({
     1
```

```
 }),
  enabled = reactive({
     NULL
 })
)

timeline_process(config, status, position, enabled)

timeline_pipeline_ui(id)

timeline_pipeline_server(id, config, status, position, enabled)

timeline_pipeline(config, status, position, enabled)
```

Arguments

id	A ‘character()‘ as the id of the Shiny module
config	An instance of the class ‘Config‘
status	A boolean which indicates whether the current status of the pipeline
position	An integer which reflects the current position of the cursor within the steps.
enabled	A vector of booleans with the same length as the number of steps (See the slot steps in the config object). Each element indicates if the corresponding step is enabled (TRUE) or (DISABLED)

Value

A shiny app

A shiny App

Examples

```
if (interactive()) {
  config <- Config(
    mode = "process",
    fullname = "PipelineDemo_Preprocessing",
    steps = c("Filtering", "Normalization"),
    mandatory = c(FALSE, TRUE)
  )
  status <- reactive({
    c(1, 1, 0, 0)
  })
  pos <- reactive({
    2
  })
  enabled <- reactive({
    c(0, 0, 1, 1)
  })
  shiny::runApp(timeline_process(config, status, pos, enabled))
}

if (interactive()) {
  config <- Config(
    mode = "pipeline",
```

```
    fullname = "PipelineDemo",
    steps = c("DataGeneration", "Preprocessing", "Clustering"),
    mandatory = c(TRUE, FALSE, FALSE)
  )
  status <- reactive({
    c(1, 1, -1, 1, 0)
  })
  pos <- reactive({
    4
  })
  enabled <- reactive({
    c(0, 0, 0, 0, 1)
  })
  timeline_pipeline(config, status, pos, enabled)
}
```

`Timestamp` *Timestamp in UNIX format.*

Description

Returns the date and time in timestamp UNIX format.

Usage

```
Timestamp()
```

Value

A 'integer()'.

Examples

```
NULL
```

`ToggleState_NavBtns` *Updates the state of navigation buttons*

Description

Enables/disables the buttons 'Next step' and 'Previous step' wrt the current position of the cursor in the timeline.

Usage

```
ToggleState_NavBtns(current.pos, nSteps)
```

Arguments

current.pos	An ‘integer‘ which gives the current position of the cursor in the corresponding timeline
nSteps	The number of steps in the timeline

Value

NA

Examples

```
NULL
```

ToggleState_Screens *Updates the status of steps in a given range*

Description

Updates the status of steps in a given range

Usage

```
ToggleState_Screens(cond, range, is.enabled, rv)
```

Arguments

cond	A ‘boolean‘
range	A ‘vector‘ of integers. The min of this vector must be greater of equal to 0 and the max must be less or equal to the size of the vector rv$steps.enabled
is.enabled	A ‘boolean‘
rv	A ‘list‘ containing at least a slot named ’steps.enabled’ which is a vector of integers

Value

An updated version of the vector rv$steps.enabled

Examples

```
NULL
```

toggleWidget *Hide/show a widget w.r.t a condition.*

Description

Wrapper for the ‘toggleWidget()‘ function of the package ‘shinyjs‘

Usage

```
toggleWidget(widget, condition)
```

Arguments

widget	The id of a ‘Shiny‘ widget
condition	A ‘logical()‘ to hide/show the widget.

Value

NA

Author(s)

Samuel Wieczorek

Examples

```
NULL
```

updateDirectoryInput *Change the value of a directoryInput on the client*

Description

Change the value of a directoryInput on the client

Usage

```
updateDirectoryInput(session, inputId, value = NULL, ...)
```

Arguments

session	The session object passed to function given to shinyServer.
inputId	The id of the input object.
value	A directory path to set
...	Additional arguments passed to choose_dir. Only used if value is NULL.

Details

Sends a message to the client, telling it to change the value of the input object. For `directoryInput` objects, this changes the value displayed in the text-field and triggers a client-side change event. A directory selection dialog is not displayed.

Value

NA

Examples

```
NULL
```

UpdateStepsStatus *Update the status of steps in a pipeline nor process.*

Description

Update the status of steps in a pipeline nor process.

Usage

```
UpdateStepsStatus(dataIn, config)
```

Arguments

`dataIn`	An instance of the 'SummarizedExperiment' class
`config`	An instance of the 'Config' class

Value

A vector of boolean in which each item indicates the status (VALIDATED, UNDONE, SKIPPED) of the corresponding step.

Examples

```
NULL
```

Update_State_Screens *Update the status enabled/disabled of the steps of a pipeline/process*

Description

Update the status enabled/disabled of the steps of a pipeline/process

Usage

```
Update_State_Screens(is.skipped, is.enabled, rv)
```

Arguments

is.skipped	A 'boolean' indicating whether the current step of a process or process of a pipeline has the status SKIPPED
is.enabled	A 'boolean' indicating whether the current step of a process or process of a pipeline is enabled (TRUE) or not (FALSE)
rv	A 'list()' containing at least an item named 'steps.status' which is a vector of of names for the steps of a pipeline nor a process.

Value

A 'vector' of boolean which gives the status enabled (TRUE) or disabled (FALSE) of the steps from a pipeline nor a process.

Examples

```
NULL
```

view_dataset *mod_open_demo_dataset_ui and mod_open_demo_dataset_server*

Description

A shiny Module.

Usage

```
view_dataset_ui(id)

view_dataset_server(id, dataIn = NULL, ...)

view_dataset(dataIn)
```

Arguments

id	A 'character()' as the id of the Shiny module
dataIn	An instance of the class 'MultiAssayExperiment'
...	Additional parameters

Value

A shiny App

Examples

```
if (interactive()) {
  data(lldata123)
  view_dataset(lldata123)
}
```

Index

∗ **datasets**
  lldata, 26
  lldata1, 26
  lldata12, 27
  lldata123, 27
∗ **data**
  lldata, 26
  lldata1, 26
  lldata12, 27
  lldata123, 27
∗ **internal**
  history_dataset, 20
  infos_dataset, 21
%AND% *(*choose_dir*)*, 5

Add2History, 3
addDatasets *(*dataset-processing*)*, 8
addDatasets,list-method *(*dataset-processing*)*, 8
AdditionnalCodeForExternalModules *(*insertCodeForExternalModules*)*, 23

BuildData2Send, 4

call_func, 5
choose_dir, 5, *6*, *11*, *52*
Config *(*Config-class*)*, 6
Config-class, 6

dataset-processing, 8
default_base_URL *(*default_vars*)*, 10
default_funcs *(*default_vars*)*, 10
default_layout *(*default_vars*)*, 10
default_pos *(*globalvariables*)*, 19
default_theme *(*default_vars*)*, 10
default_vars, 10
default_workflow *(*default_vars*)*, 10
directoryInput, 10
Discover_Skipped_Steps, 11
download_dataset, 12
download_dataset_server *(*download_dataset*)*, 12
download_dataset_ui *(*download_dataset*)*, 12

file_sep *(*choose_dir*)*, 5
format_DT, 13
format_DT_server *(*format_DT*)*, 13
format_DT_ui *(*format_DT*)*, 13

generic_mod_open_dataset, 14
Get_AdditionalModule_Core_Code *(*insertCodeForExternalModules*)*, 23
Get_Code_Declare_rv_custom *(*insertCodeForExternalModules*)*, 23
Get_Code_Declare_widgets *(*insertCodeForExternalModules*)*, 23
Get_Code_for_dataOut *(*insertCodeForExternalModules*)*, 23
Get_Code_for_General_observeEvents *(*insertCodeForExternalModules*)*, 23
Get_Code_for_ObserveEvent_widgets *(*insertCodeForExternalModules*)*, 23
Get_Code_for_remoteReset *(*insertCodeForExternalModules*)*, 23
Get_Code_for_resetting_custom *(*insertCodeForExternalModules*)*, 23
Get_Code_for_resetting_widgets *(*insertCodeForExternalModules*)*, 23
Get_Code_for_rv_reactiveValues *(*insertCodeForExternalModules*)*, 23
Get_Code_Update_Config_Variable *(*insertCodeForExternalModules*)*, 23
Get_Workflow_Core_Code *(*insertCodeForExternalModules*)*, 23
GetExtension, 15
GetFirstMandatoryNotValidated, 16

GetHistory, 17
GetListDatasets, 17
GetMaxValidated_AllSteps, 18
GetMaxValidated_BeforePos, 18
GetPackageVersion, 19
globalvariables, 19

history_dataset, 20
history_dataset_server
(history_dataset), 20
history_dataset_ui (history_dataset), 20

infos_dataset, 21
infos_dataset_server (infos_dataset), 21
infos_dataset_ui (infos_dataset), 21
initComplete, 22
initialize,Config-method
(Config-class), 6
InitializeHistory, 22
Insert_Call_to_Config, 25
insert_md (mod_insert_md), 31
insert_md_server (mod_insert_md), 31
insert_md_ui (mod_insert_md), 31
Insert_User_Sidebar (sidebars), 47
insertCodeForExternalModules, 23
isSubstr, 25

keepDatasets (dataset-processing), 8
keepDatasets,list-method
(dataset-processing), 8

listBrewerPalettes (globalvariables), 19
lldata, 26
lldata1, 26
lldata12, 27
lldata123, 27

MagellanNTK (magellanNTK), 28
magellanNTK, 28
MagellanNTK_server (magellanNTK), 28
MagellanNTK_ui (magellanNTK), 28
mainapp (mod_main_page), 32
mainapp_server (mod_main_page), 32
mainapp_ui (mod_main_page), 32
mod_err_modal, 29
mod_errorModal (mod_err_modal), 29
mod_errorModal_server (mod_err_modal),
29
mod_errorModal_ui (mod_err_modal), 29
mod_homepage, 30
mod_homepage_server (mod_homepage), 30
mod_homepage_ui (mod_homepage), 30
mod_insert_md, 31
mod_main_page, 32
mod_modalDialog, 33
mod_modalDialog_server
(mod_modalDialog), 33
mod_modalDialog_ui (mod_modalDialog), 33
mod_open_workflow, 34
mod_popover_for_help, 35
mod_popover_for_help_server
(mod_popover_for_help), 35
mod_popover_for_help_ui
(mod_popover_for_help), 35
mod_release_notes, 35
mod_release_notes_server
(mod_release_notes), 35
mod_release_notes_ui
(mod_release_notes), 35
mod_SweetAlert (mod_sweetAlert), 36
mod_sweetAlert, 36
mod_SweetAlert_server (mod_sweetAlert),
36
mod_SweetAlert_ui (mod_sweetAlert), 36
module_exists, 29
Module_Return_Func
(insertCodeForExternalModules),
23

nav_pipeline, 37
nav_pipeline_server (nav_pipeline), 37
nav_pipeline_ui (nav_pipeline), 37
nav_process, 39
nav_process_server (nav_process), 39
nav_process_ui (nav_process), 39
nav_single_process, 40
nav_single_process_server
(nav_single_process), 40
nav_single_process_ui
(nav_single_process), 40
NavPage, 37

open_dataset
(generic_mod_open_dataset), 14
open_dataset_server
(generic_mod_open_dataset), 14
open_dataset_ui
(generic_mod_open_dataset), 14
open_workflow (mod_open_workflow), 34
open_workflow_server
(mod_open_workflow), 34
open_workflow_ui (mod_open_workflow), 34

path.expand, *6*, *10*
pkgsRequire, 42
process_layout, 42

process_layout_pipeline
(process_layout), 42
process_layout_process
(process_layout), 42

readConfigFile, 43
readDirectoryInput, *11*, 44
runDirinputExample, 44

SetCurrentPosition, 45
SetHistory, 45
show,Config-method, 46
sidebars, 47
source_shinyApp_files, 47
source_wf_files, 48
stepStatus (globalvariables), 19

timeline_pipeline (timelines), 48
timeline_pipeline_server (timelines), 48
timeline_pipeline_ui (timelines), 48
timeline_process (timelines), 48
timeline_process_server (timelines), 48
timeline_process_ui (timelines), 48
timelines, 48
Timestamp, 50
tl_v_next_icon (globalvariables), 19
ToggleState_NavBtns, 50
ToggleState_Screens, 51
toggleWidget, 52

Update_State_Screens, 54
updateDirectoryInput, *11*, 52
UpdateStepsStatus, 53

view_dataset, 54
view_dataset_server (view_dataset), 54
view_dataset_ui (view_dataset), 54