

Package ‘iTReX’

May 30, 2022

Type Package

Title interactive Therapy Response eXploration

Version 1.2.0

Description An R/Shiny web app for interactive exploration of mono- and combination therapy dose response profiling data and drug target interaction network mapping.

License GPL-3

URL <https://github.com/iTReX-Shiny/iTReX>

BugReports <https://github.com/iTReX-Shiny/iTReX/issues>

Depends R (>= 3.5.0), shiny

biocViews ShinyApps, Network

Imports ComplexHeatmap, imageHTS, OmnipathR, base64enc, bayestestR, circlize, data.table, desc, dplyr, DT, future, ggplot2, ggpubr, git2r, glue, htmlwidgets, igraph, knitr, nplr, openxlsx, patchwork, plotly, purrr, readr, readxl, rlang, rmarkdown, shinyjs, shinythemes, shinyWidgets, stats, stringr, tools, tibble, tidyr, utils, zeallot, zip

Suggests BiocCheck, Cairo, curl, devtools, fs, httr, kableExtra, lintr, processx, remotes, rmdformats, shinytest, styler, testthat (>= 3.0.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

Config/testthat/edition 3

VignetteBuilder knitr

R topics documented:

build_manually	2
check_testnames	2
clean_and_convert_layout	3
compute_dss	3
debug_save	4
prepare	4
prepare_demos	5
prep_and_check	5

record_test	5
run	6
show_system_deps	6
test	6
test_and_update	7
test_github	7
update_tests	7
version	8

Index	9
--------------	----------

build_manuals	<i>Build manuals (including vignettes) in the working copy.</i>
---------------	---

Description

Build manuals (including vignettes) in the working copy.

Usage

```
build_manuals(pkg = getwd())
```

Arguments

pkg	The directory holding the working copy.
-----	---

check_testnames	<i>Check test names against pattern.</i>
-----------------	--

Description

Check test names against pattern.

Usage

```
check_testnames(pkg = getwd())
```

Arguments

pkg	The directory holding the working copy.
-----	---

`clean_and_convert_layout`

Convert layout from old to new format. Can be safely applied to new layouts.

Description

Convert layout from old to new format. Can be safely applied to new layouts.

Usage

```
clean_and_convert_layout(df, is_demo = FALSE)
```

Arguments

<code>df</code>	A data frame.
<code>is_demo</code>	Whether to apply further normalization to minimize entries.

`compute_dss`

Compute a Drug Sensitivity Score (DSS)

Description

Compute a Drug Sensitivity Score (DSS)

Usage

```
compute_dss(  
  pars_table,  
  resp_min = 10,  
  score_type = 3,  
  conc_scale_unused = 1e-09,  
  apply_log_to_conc = TRUE,  
  curve = NA,  
  force_symm = FALSE,  
  resp_max = 100,  
  bottom_corr = TRUE,  
  allow_big_mid = FALSE,  
  int_decreasing = FALSE,  
  allow_nan = TRUE  
)
```

Arguments

<code>pars_table</code>	can be a data.frame of (in that order!)
<code>resp_min</code>	TODO
<code>score_type</code>	TODO
<code>conc_scale_unused</code>	TODO

apply_log_to_conc	TODO
curve	TODO
force_symm	TODO
resp_max	TODO
bottom_corr	TODO
allow_big_mid	TODO
int_decreasing	TODO
allow_nan	TODO

debug_save	<i>Save arguments in rds files named after arguments.</i>
------------	---

Description

Save arguments in rds files named after arguments.

Usage

```
debug_save(...)
```

Arguments

... Variables to be saved (not quoted).

prepare	<i>Prepare working copy for commit.</i>
---------	---

Description

Prepare working copy for commit.

Usage

```
prepare(pkg = getwd(), manual = TRUE)
```

Arguments

pkg	The directory holding the working copy.
manual	Whether to run <code>build_manually()</code> .

prepare_demos	<i>Prepare demo data (xlsx and zip) in the working copy.</i>
---------------	--

Description

Prepare demo data (xlsx and zip) in the working copy.

Usage

```
prepare_demos(pkg = getwd())
```

Arguments

pkg	The directory holding the working copy.
-----	---

prep_and_check	<i>Prepare and check iTReX in the working copy.</i>
----------------	---

Description

Prepare and check iTReX in the working copy.

Usage

```
prep_and_check(pkg = getwd(), strict = FALSE)
```

Arguments

pkg	The directory holding the working copy.
strict	Whether to check for CRAN and BioConductor compatibility.

record_test	<i>Record a new test and store it in the working copy.</i>
-------------	--

Description

Record a new test and store it in the working copy.

Usage

```
record_test(pkg = getwd())
```

Arguments

pkg	The directory holding the working copy.
-----	---

run	<i>Run iTReX Shiny app</i>
-----	----------------------------

Description

Run iTReX Shiny app

Usage

```
run(test.mode = FALSE)
```

Arguments

test.mode	Whether to run the application in test mode.
-----------	--

show_system_deps	<i>Generate system dependencies for README.md</i>
------------------	---

Description

Generate system dependencies for README.md

Usage

```
show_system_deps(pkg = getwd())
```

Arguments

pkg	The directory holding the working copy.
-----	---

test	<i>Run iTReX tests</i>
------	------------------------

Description

Run iTReX tests

Usage

```
test(testnames = NULL, quiet = FALSE, url = NULL)
```

Arguments

testnames	Passed to shinytest::testApp()
quiet	Passed to shinytest::testApp()
url	URL of test-mode Shiny instance to connect to. Requires a customized shinytest instance as of shinytest v1.5.1.

Examples

```
test(testnames = c("tabs_home"))
```

test_and_update	<i>Run (and update) tests in the working copy.</i>
-----------------	--

Description

Run (and update) tests in the working copy.

Usage

```
test_and_update(..., pkg = getwd())
```

Arguments

...	Parameters (such as testnames) passed to shinytest::testApp() .
pkg	The directory holding the working copy.

test_github	<i>Install from GitHub and test</i>
-------------	-------------------------------------

Description

Install from GitHub and test

Usage

```
test_github()
```

update_tests	<i>Update test-expected in the working copy.</i>
--------------	--

Description

Update test-expected in the working copy.

Usage

```
update_tests(..., pkg = getwd())
```

Arguments

...	Parameters passed to shinytest::snapshotUpdate() .
pkg	The directory holding the working copy.

version	<i>Report iTReX version</i>
---------	-----------------------------

Description

Report iTReX version

Usage

```
version(long = FALSE)
```

Arguments

long	Whether to print the full version string with git commit hash.
------	--

Examples

```
version()  
version(long = TRUE)
```


Index

build_manuals, [2](#)
build_manuals(), [4](#)

check_testnames, [2](#)
clean_and_convert_layout, [3](#)
compute_dss, [3](#)

debug_save, [4](#)

prep_and_check, [5](#)
prepare, [4](#)
prepare_demos, [5](#)

record_test, [5](#)
run, [6](#)

shinytest::snapshotUpdate(), [7](#)
shinytest::testApp(), [6](#), [7](#)
show_system_deps, [6](#)

test, [6](#)
test_and_update, [7](#)
test_github, [7](#)

update_tests, [7](#)

version, [8](#)