Package ‘rworkflows’

January 12, 2023

Type Package

Title Test, Document, Containerise, and Deploy R Packages

Version 0.99.5

Description Continuous integration for R packages.
Automates testing, documentation website building,
and containerised deployment.

URL `https://github.com/neurogenomics/rworkflows`,
`https://CRAN.R-project.org/package=rworkflows`

BugReports `https://github.com/neurogenomics/rworkflows/issues`

Encoding UTF-8

Depends R (>= 4.1)

biocViews Genetics, FunctionalGenomics, SystemsBiology

Imports stats, here, yaml, utils, desc, badger

Suggests markdown, rmarkdown, magick, remotes, knitr, BiocStyle, covr, testthat (>= 3.0.0), hexSticker, htmltools, data.table

RoxygenNote 7.2.3

VignetteBuilder knitr

License GPL-3

Config/testthat/edition 3

NeedsCompilation no

Author Brian Schilder [aut, cre] (<`https://orcid.org/0000-0001-5949-2191`>),
Alan Murphy [aut, ctb] (<`https://orcid.org/0000-0002-2487-8753`>),
Nathan Skene [aut] (<`https://orcid.org/0000-0002-6807-3180`>)

Maintainer Brian Schilder <`brian_schilder@alumni.brown.edu`>

Repository CRAN

Date/Publication 2023-01-12 12:40:02 UTC

R topics documented:

`bioc_r_versions` *Bioconductor / R versions*

Description

Get the respective version of R for a given version of Bioconductor.

Usage

```
bioc_r_versions(bioc_version = NULL)
```

Arguments

`bioc_version`	Version of Bioc to return info for. Can be: • "devel"Get the current development version of Bioc. • "release"Get the current release version of Bioc. • <numeric>A specific Bioc version number (e.g. `3.16`). • NULLReturn info for all Bioc versions.

Value

Named list of Bioc/R versions

Examples

```
ver <- bioc_r_versions(bioc_version="devel")
```

`construct_runners` *Construct runners*

Description

Construct runner configurations across multiple Operating Systems (OS) for GitHub Actions workflow.

Usage

```
construct_runners(
  os = c("ubuntu-latest", "macOS-latest", "windows-latest"),
  bioc = stats::setNames(list("devel", "release", "release"), os),
  r = stats::setNames(list("auto", "auto", "auto"), os),
 cont = stats::setNames(list(paste0("bioconductor/bioconductor_docker:", bioc[[1]]),
    NULL, NULL), os),
  rspm = stats::setNames(list(paste0("https://packagemanager.rstudio.com/",
    "cran/__linux__/focal/release"), NULL, NULL), os),
  versions_explicit = FALSE
)
```

Arguments

`os`	Which OS to launch GitHub Actions on.
`bioc`	Which Bioconductor version to use on each OS. See bioc_r_versions documentation for all options.
`r`	Which R version to use on each OS.
`cont`	Which Docker container to use on each OS (`NULL` means no container will be used for that OS). See here for a list of all official Bioconductor Docker container versions.
`rspm`	Which R repository manager to use on each OS (`NULL` means the default will be used for that OS).
`versions_explicit`	Specify R/Bioc versions explicitly (e.g. `r: 4.2.0`, `bioc: 3.16`) as opposed to flexibly (e.g. `r: "latest"`, `bioc: "release"`).

Value

Named list of configurations for each runner OS.

Examples

```
runners <- construct_runners()
```

`use_badges` *Use badges*

Description

Create one or more badges showing the status of your R package. Uses the package **badger**.

Usage

```
use_badges(
  add_hex = TRUE,
  add_actions = "rworkflows",
  add_doi = NULL,
  add_github_version = TRUE,
  add_commit = TRUE,
  add_code_size = TRUE,
  add_codecov = TRUE,
  add_license = TRUE,
  add_authors = TRUE,
  add_bioc_release = FALSE,
  add_bioc_download_month = FALSE,
  add_bioc_download_total = FALSE,
  add_bioc_download_rank = FALSE,
  add_cran_release = FALSE,
  add_cran_checks = FALSE,
  add_cran_download_month = FALSE,
  add_cran_download_total = FALSE,
  branch = "master",
  as_list = FALSE,
  sep = "\n",
  hex_height = 600,
 colors = list(github = "black", bioc = "green", cran = "blue", default = "blue"),
  verbose = TRUE
)
```

Arguments

`add_hex`	Add a hex sticker. If `add_hex=TRUE`, will assume the sticker is located at the following relative path: "inst/hex/hex.png". If `add_hex` is a character string, this will instead be used as the relative hex path (e.g. `"/images/mysticker.png"`).
`add_actions`	The name of one or more GitHub Actions to show the status for with badge_github_actions (e.g. c("rworkflows","rworkflows_static")).
`add_doi`	Add the DOI of a given package or publication associated with the package. Must be provided as a character string.
`add_github_version`	Add package version with badge_github_version.

`add_commit`	Add the last GitHub repo commit date with badge_last_commit.
`add_code_size`	Add code size with badge_code_size.
`add_codecov`	Add CodeCov status with badge_codecov.
`add_license`	Add license info with badge_license.
`add_authors`	Add author names inferred from the `DESCRIPTION` file.
`add_bioc_release`	Add Bioc release version with badge_bioc_release.
`add_bioc_download_month`	Add the number of Bioc downloads last month badge_bioc_download.
`add_bioc_download_total`	Add the number of Bioc downloads total badge_bioc_download.
`add_bioc_download_rank`	Add the download rank of the package on Bioc badge_bioc_download_rank.
`add_cran_release`	Add Bioc release version with badge_cran_release.
`add_cran_checks`	Add whether package is passing all checks on CRAN with badge_cran_checks.
`add_cran_download_month`	Add the number of CRAN downloads last month badge_cran_download.
`add_cran_download_total`	Add the number of CRAN downloads total badge_cran_download.
`branch`	Name of the GitHub repository branch to use.
`as_list`	Return the header as a named list (`TRUE`), or a collapsed text string (default: `FALSE`).
`sep`	Character to separate each item in the list with using paste.
`hex_height`	Height of the hex sticker in pixels (when `add_hex=TRUE`).
`colors`	Colors to assign to each group of badges (when possible).
`verbose`	Print messages.

Value

A named list of selected badges in markdown format.

Examples

```
## Causes issues bc examples can't find the  the DESCRIPTION file.
 ## Not run:
 rworkflows::use_badges()

## End(Not run)
```

`use_dockerfile` *Use Dockerfile*

Description

Creates a Docker file to be used with the GitHub Actions (GHA) workflows distributed by **rworkflows**.

Usage

```
use_dockerfile(
  save_dir = here::here(),
  path = file.path(save_dir, "Dockerfile"),
  force_new = FALSE,
  show = FALSE,
  verbose = TRUE
)
```

Arguments

`save_dir`	Directory to save the Docker file to.
`path`	Path to the Docker file.
`force_new`	If a Docker file already exists, overwrite it (default: `FALSE`).
`show`	Print the contents of the Docker file in the R console.
`verbose`	Print messages.

Value

Path to Docker file.

Examples

```
path <- use_dockerfile(save_dir=tempdir())
```

`use_issue_template` *Use Issue Template*

Description

Creates one or more Issue Templates to be used in a GitHub repository.

Usage

```
use_issue_template(
  templates = c("bug_report.md", "feature_request.md"),
  save_dir = here::here(".github", "ISSUE_TEMPLATE"),
  path = file.path(save_dir, templates),
  force_new = FALSE,
  show = FALSE,
  verbose = TRUE
)
```

Arguments

`templates`	The names of templates to be used.
`save_dir`	Directory to save the Docker file to.
`path`	Path to the Docker file.
`force_new`	If a Docker file already exists, overwrite it (default: `FALSE`).
`show`	Print the contents of the Docker file in the R console.
`verbose`	Print messages.

Value

Path to Issue Templates.

Examples

```
path <- use_issue_template(save_dir=tempdir())
```

`use_readme` *Use README*

Description

Creates an rmarkdown README file that autofills using metadata from the R package *DESCRIPTION* file.

Usage

```
use_readme(
  save_dir = here::here(),
  path = file.path(save_dir, "README.Rmd"),
  force_new = FALSE,
  show = FALSE,
  verbose = TRUE
)
```

Arguments

`save_dir`	Directory to save the vignette file to.
`path`	Path to the vignette file.
`force_new`	If the file already exists, overwrite it (default: `FALSE`).
`show`	Print the contents of the vignette file in the R console.
`verbose`	Print messages.

Value

Path to README file.

Examples

```
## use default save_dir in practice
path <- use_readme(save_dir = tempdir())
```

`use_vignette_docker` *Use vignette: Docker*

Description

Creates a vignette rmarkdown file demonstrates how to create a Docker/Singularity image from a container stored in Dockerhub.

Usage

```
use_vignette_docker(
  docker_org,
  title = "Docker/Singularity Containers",
  vignette_index_entry = "docker",
  save_dir = here::here(),
  path = file.path(save_dir, "vignettes", "docker.Rmd"),
  force_new = FALSE,
  show = FALSE,
  verbose = TRUE
)
```

Arguments

`docker_org`	DockerHub organization name. Can simply be your Dockerhub username instead.
`title`	Title of vignette.
`vignette_index_entry`	Index entry of the vignette, which is used when creating the navigation bar in the **pkgdown** site.

`save_dir`	Directory to save the vignette file to.
`path`	Path to the vignette file.
`force_new`	If the file already exists, overwrite it (default: `FALSE`).
`show`	Print the contents of the vignette file in the R console.
`verbose`	Print messages.

Value

Path to vignette file.

Examples

```
path <- use_vignette_docker(docker_org = "neurogenomicslab",
                            ## use default save_dir in practice
                            save_dir = tempdir())
```

---

use_vignette_getstarted

Use vignette: Get started

Description

Creates a "Get started" rmarkdown vignette file.

Usage

```
use_vignette_getstarted(
  package,
  title = "Get started",
  vignette_index_entry = package,
  save_dir = here::here(),
  path = file.path(save_dir, "vignettes", paste0(package, ".Rmd")),
  force_new = FALSE,
  show = FALSE,
  verbose = TRUE
)
```

Arguments

`package`	R package name.
`title`	Title of vignette.
`vignette_index_entry`	Index entry of the vignette, which is used when creating the navigation bar in the **pkgdown** site.
`save_dir`	Directory to save the vignette file to.

`path`	Path to the vignette file.
`force_new`	If the file already exists, overwrite it (default: `FALSE`).
`show`	Print the contents of the vignette file in the R console.
`verbose`	Print messages.

Value

Path to vignette file.

Examples

```
path <- use_vignette_getstarted(package = "mypackage",
                                ## use default save_dir in practice
                                save_dir = tempdir())
```

`use_workflow` *Use GitHub Actions workflow*

Description

Create workflow that calls an rworkflows GitHub Actions (GHA)

Usage

```
use_workflow(
  name = "rworkflows",
  tag = "@master",
  on = c("push", "pull_request"),
  branches = c("master", "main", "RELEASE_**"),
  runners = construct_runners(),
  run_bioccheck = FALSE,
  run_rcmdcheck = TRUE,
  as_cran = TRUE,
  run_vignettes = TRUE,
  has_testthat = TRUE,
  run_covr = TRUE,
  run_pkgdown = TRUE,
  has_runit = FALSE,
  has_latex = FALSE,
  run_docker = FALSE,
  github_token = "${{ secrets.PAT_GITHUB }}",
  docker_user = NULL,
  docker_org = docker_user,
  docker_token = "${{ secrets.DOCKER_TOKEN }}",
  cache_version = "cache-v1",
  enable_act = FALSE,
  save_dir = here::here(".github", "workflows"),
```

```
  return_path = TRUE,
  force_new = FALSE,
  preview = FALSE,
  verbose = TRUE
)
```

Arguments

`name`	Workflow name. - "rworkflows"A short workflow script that calls the GitHub action from the GitHub Marketplace. The action is continually updated so users do not need to worry about maintaining it. - "rworkflows_static"A longer workflow scripts that explicitly copies all steps from the **rworkflows** action into a static file. Users may need to update this file themselves over time, though this does allow for a fully customisable workflow.
`tag`	Which version of the `rworkflows` action to use. Can be a branch name on the GitHub repository (e.g. "\@master"), or a Release Tag (e.g. "\@v1").
`on`	GitHub trigger conditions.
`branches`	GitHub trigger branches.
`runners`	Runner configurations for multiple Operating Systems (OS), including R versions, Bioc versions, and container sources. Can use the construct_runners functions to assist in constructing customized runners configurations.
`run_bioccheck`	Run Bioconductor checks using `BiocCheck::BiocCheck()`. Must pass in order to continue workflow.
`run_rcmdcheck`	Run R CMD checks using `rcmdcheck::rcmdcheck()`. Must pass in order to continue workflow.
`as_cran`	When running R CMD checks, use the ’–as-cran’ flag to apply CRAN standards
`run_vignettes`	Build and check R package vignettes.
`has_testthat`	Run unit tests and report results.
`run_covr`	Run code coverage tests and publish results to codecov.
`run_pkgdown`	Knit the *README.Rmd* (if available), build documentation website, and deploy to *gh-pages* branch.
`has_runit`	Run R Unit tests.
`has_latex`	Install a suite of LaTeX dependencies used for rendering Sweave (.rnw) and other documentation files.
`run_docker`	Whether to build and push a Docker container to DockerHub.
`github_token`	Token for the repo. Can be passed in using secrets.PAT_GITHUB .
`docker_user`	DockerHub username.
`docker_org`	DockerHub organization name. Is the same as `docker_user` by default.
`docker_token`	DockerHub token.
`cache_version`	Name of the cache sudirectory to be used when reinstalling software in GHA.

`enable_act`	Whether to add extra lines to the yaml to enable local workflow checking with act.
`save_dir`	Directory to save workflow to.
`return_path`	Return the path to the saved *yaml* workflow file (default: `TRUE`), or return the *yaml* object directly.
`force_new`	If the GHA workflow yaml already exists, overwrite with new one (default: `FALSE`).
`preview`	Print the yaml file to the R console.
`verbose`	Print messages.

Value

Path or yaml object.

Source

Issue reading in "on:"/"y","n" elements.

Issue writing "on:" as "’as’:"

Examples

```
path <- use_workflow(save_dir = file.path(tempdir(),”.github”,”workflows”))
```

Index