

Package: ReleaseLaunch (via r-universe)

September 29, 2026

Title R utilities related to Bioconductor release tasks

Version 0.99.30

Encoding UTF-8

Description The package facilitates a Bioconductor release. It currently updates existing Bioconductor repositories on GitHub after a release. The script creates the RELEASE_XX_YY branch and pushes it to the Bioconductor organization.

License Artistic-2.0

Imports BiocBaseUtils, BiocManager, gert, gh, methods, utils, yaml

Suggests BiocParallel, BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Date 2026-04-29

Config/pak/sysreqs git libgit2-dev libssl-dev

Repository <https://link-ny.r-universe.dev>

Date/Publication 2026-04-29 21:43:08 UTC

RemoteUrl <https://github.com/Bioconductor/ReleaseLaunch>

RemoteRef HEAD

RemoteSha d1e3957e08b6a8fd9a25bea6844f75a00a8a2cad

Contents

bioc_remotes	2
branch-release-gh	3
extractNEWS	4
get-bioc-packages	5
get-github-repos	6
getPackagesNEWS	7
packages_without_release_branch	8
update_local_repos	9

Index**11**

bioc_remotes

*Convenience functions to set, rename, or validate a repo's remotes***Description**

The function `add_bioc_remote` adds an 'upstream' remote that points to `git@git.bioconductor.org`. If an upstream remote already exists, it will be validated against the SSH URL. By design, the Bioconductor remote location is called the 'upstream' remote. The 'origin' is set to the GitHub location.

Usage

```
add_bioc_remote(repo = ".")
```

```
set_bioc_remotes(repo = ".", org = "Bioconductor")
```

Arguments

repo	character(1) The local path to the git repository whose upstream remote should be set
org	character(1) The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").

Details

The `set_bioc_remotes` function will update a repository's remotes by setting origin to the GitHub location and upstream to the Bioconductor git server. If the origin remote is not set to the GitHub location, the function will overwrite the URL. If the upstream remote is not set, the function will add it. If the upstream remote is set, the function will overwrite the URL with the Bioconductor SSH URL. Note that the remotes will follow this template:

```
origin git@github.com:{org}/{package}
upstream git@git.bioconductor.org:packages/{package}
```

Value

- `add_bioc_remote`: adds an 'upstream' remote with the Bioconductor git address for a given package.
- `set_bioc_remotes`: updates the remotes for a given package, setting the 'origin' remote to the GitHub location and the 'upstream' remote to the Bioconductor git server.

branch-release-gh	<i>Add the release branch to GitHub package repositories</i>
-------------------	--

Description

This function assumes that you have admin push access to the GitHub organization indicated by `org`.

Usage

```
add_gh_release_branch(
  package_name,
  release = bioc_release_yaml(),
  gh_branch = .BIOC_DEFAULT_BRANCH,
  bioc_branch = .BIOC_DEFAULT_BRANCH,
  org = "Bioconductor"
)

add_gh_release_branches(
  packages = character(0L),
  release = bioc_release_yaml(),
  bioc_branch = .BIOC_DEFAULT_BRANCH,
  org = "Bioconductor"
)

bioc_version_yaml(config = .BIOC_CONFIG_FILE)

bioc_release_yaml(config = .BIOC_CONFIG_FILE)
```

Arguments

<code>package_name</code>	<code>character(1)</code> The name of the organization R package that is also available on GitHub.
<code>release</code>	<code>character(1)</code> The Bioconductor version branch tag, e.g., "RELEASE_3_17"
<code>gh_branch</code>	<code>character(1)</code> The name of the default branch on GitHub. It may be 'devel' or 'main' depending on the repository
<code>bioc_branch</code>	<code>character(1)</code> The name of the default branch on the Bioconductor git server (default 'devel')
<code>org</code>	<code>character(1)</code> The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").
<code>packages</code>	<code>named character()</code> A character vector of default branches whose names correspond to Bioconductor package names. See <code>packages_without_release_branch</code> .
<code>config</code>	<code>character(1)</code> The path to the Bioconductor configuration file that contains the release version (defaults to website URL from <code>.BIOC_CONFIG_FILE</code>)

See Also

packages_without_release_branch

Examples

```
if (interactive()) {
  bioc_version_yaml()
  bioc_release_yaml()

  gitcreds::gitcreds_set()
  ## gitcreds::gitcreds_get()

  add_gh_release_branch(
    package_name = "BiocParallel",
    release = bioc_release_yaml()
  )

  add_gh_release_branches(
    release = bioc_release_yaml(),
    org = "Bioconductor"
  )
}
```

extractNEWS

Extract NEWS files from source package tarballs

Description

Extracts NEWS files from source tarballs of packages.

Usage

```
extractNEWS(
  reposRoot,
  srcContrib,
  destDir = file.path(reposRoot, "news"),
  unpackDir = tempdir()
)
```

Arguments

reposRoot	Top level path for CRAN-style repos
srcContrib	Location of source packages
destDir	character(1) The top-level directory location for extracted NEWS file. destDir will be followed by the package subfolder and NEWS file, i.e., /<packageName>/NEWS
unpackDir	character(1) The directory to extract files to from the tarball (default is the tempdir() location)

get-bioc-packages	<i>Get the Bioconductor packages in the software manifest</i>
-------------------	---

Description

To ensure that a GitHub repository is a software package, its name is checked against a list of Bioconductor packages. This list is called the manifest. The `get_bioc_software_manifest` function obtains the manifest using `git`. Alternatively, `get_org_packages` uses `available.packages()` with Bioconductor repositories (`repos=...`).

Usage

```
get_bioc_software_manifest()

get_org_packages(version, org, type)
```

Arguments

version	character(1) The numeric version (as character) of the Bioconductor release, e.g., "3.16".
org	character(1) The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").
type	character() The repository names to look through as returned by <code>BiocManager::repositories()</code> . Currently, only software and experiment data ('BioCsoft' and 'BioCexp', respectively) are supported.

Value

- `get_bioc_software_manifest`: A character vector of package names
- `get_org_packages`: A named character vector of package names and their default branches

Examples

```
get_bioc_software_manifest()

get_org_packages(
  version = bioc_version_yaml(),
  org = "Bioconductor",
  type = c("BioCsoft", "BioCexp")
)
```

get-github-repos

Get all repositories for a given GitHub organization or username

Description

get_user_github_repos and get_org_github_repos use the GitHub API, to obtain the repositories hosted on GitHub via the 'username' or the organization ('org') names, respectively.

Usage

```
get_org_github_repos(
  per_page = 100,
  pages = Inf,
  org = "Bioconductor",
  archived = FALSE
)
```

```
get_user_github_repos(per_page = 100, pages = Inf, username, archived = FALSE)
```

Arguments

per_page	Number of items to return per page. If omitted, will be substituted by <code>max(.limit, 100)</code> if <code>.limit</code> is set, otherwise determined by the API (never greater than 100).
pages	<code>numeric(1)</code> The number of pages to 'flip' through (default <code>Inf</code>)
org	<code>character(1)</code> The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").
archived	<code>logical(1)</code> Whether to include archived repositories in the query results (default <code>FALSE</code>)
username	<code>character(1)</code> The GitHub username used to query repositories

Value

A vector of default branches whose names correspond to the organization or user GitHub repositories

Examples

```
if (interactive()) {
  get_user_github_repos(username = "github-username")
}
```

getPackagesNEWS	<i>Compare the old and current releases to generate the release announcements NEWS compilation</i>
-----------------	--

Description

The function uses previous and current versions of Bioconductor to generate a single package's NEWS file.

Usage

```
getPackagesNEWS(
  prevRepos = "3.15",
  currRepos = "3.16",
  repo = c("bioc", "data/experiment", "workflows"),
  srcdir = NULL
)

printNEWS(
  dbs,
  destfile,
  overwrite = FALSE,
  width = 68,
  output = c("md", "text"),
  relativeLink = FALSE,
  ...
)
```

Arguments

prevRepos	character(1) The version string indicating the old release version of Bioconductor
currRepos	character(1) The version string indicating the newest and current release version of Bioconductor
repo	character(1) The repository nickname indicating which repository to compare news
srcdir	(Optional) character(1) The source directory in which all the Bioconductor packages, whose NEWS files are to be interrogated, reside
dbs	list() The news_db list as returned by getPackageNEWS.
destfile	character(1) The file path location to where NEWS will be saved.
overwrite	logical(1) Indicate whether destfile should be over-written, if it exists.
width	numeric(1) The maximum number of characters NEWS items can span before wrapping, excludes indent.
output	character(1) Either 'md' or 'text' indicating the output in markdown or text format, respectively.

relativeLink	logical(1) Whether to include links relative to the Bioconductor website or provide the full URL, i.e., https://bioconductor.org (default: FALSE).
...	additional arguments, unused.

Value

A list of NEWS

packages_without_release_branch

Generate the list of packages to be updated

Description

These functions obtain all the repositories from the designated organization and filters them to only valid R packages and repositories that do not have a RELEASE_X_Y branch or that do, depending on the function called.

Usage

```
packages_without_release_branch(
  version = bioc_version_yaml(),
  release = bioc_release_yaml(),
  org = "Bioconductor",
  type = c("BioCsoft", "BioCexp")
)
```

```
packages_with_release_branch(
  version = bioc_version_yaml(),
  release = bioc_release_yaml(),
  org = "Bioconductor",
  type = c("BioCsoft", "BioCexp")
)
```

Arguments

version	character(1) The numeric version (as character) of the Bioconductor release, e.g., "3.16".
release	character(1) The Bioconductor version branch tag, e.g., "RELEASE_3_17"
org	character(1) The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").
type	character() The repository names to look through as returned by <code>BiocManager::repositories()</code> . Currently, only software and experiment data ('BioCsoft' and 'BioCexp', respectively) are supported.

Value

A named scalar string of the default branch whose name corresponds to a Bioconductor GitHub repository

Examples

```
if (interactive()) {
  packages_without_release_branch(
    version = bioc_version_yaml()
  )
}
```

update_local_repos	<i>Update local repositories after Bioconductor Release</i>
--------------------	---

Description

These functions should be used after a Bioconductor Release. They ensure that the local repositories are in sync with Bioconductor. For convenience, the singular update_local_repo function will update a single local repository on the user's system. The function checks repositories against the available.packages() in the Bioconductor owned BiocManager::repositories().

Usage

```
update_local_repos(
  repos_dir,
  release = bioc_release_yaml(),
  username,
  org = username,
  set_upstream = "origin/devel",
  restart_from = character(0L),
  BPPARAM = NULL
)

update_local_repo(
  repo_dir,
  release,
  set_upstream = "origin/devel",
  org = "Bioconductor"
)
```

Arguments

repos_dir	character(1) The base directory where all packages / repositories exist for the user
release	character(1) The Bioconductor release version, as "RELEASE_X_Y", to use for updating the local repositories. By default, the value of bioc_release_yaml().

username	character(1) (optional) The GitHub username used in the query to check default packages
org	character(1) The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").
set_upstream	character(1) The remote location that will be tracked by the local branch, either "origin/devel" (default) or "upstream/devel"
restart_from	character(1) The name of a package found in repos_dir that should be used as the starting point for updating the local repos in case that the process was interrupted due to merge conflicts etc. Default character(0L).
BPPARAM	BiocParallelParam An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation. Default NULL.
repo_dir	character(1) The full path to a single package / repository whose default branch should be updated
packages	character() An optional vector of package names that correspond to folders available locally.

Details

The `update_local_repos` function will update all local repositories in the `repos_dir` directory that are found in either the users or organization's GitHub repository. The function syncs the devel and current release branches from both the origin (GH) and upstream (Bioconductor) remotes. If a upstream remote is not set, the function will set it to the Bioconductor git server. Finally, the function sets origin/devel as the remote branch to track with the `set_upstream` argument.

Examples

```
if (interactive()) {
  ## add PAT credentials
  gitcreds::gitcreds_set()
  ## update multiple packages at a time
  update_local_repos(
    repos_dir = "~/bioc/",
    org = "Bioconductor",
    release = "RELEASE_3_20"
  )

  ## update a single package
  update_local_repo(
    "~/bioc/AnnotationHub",
    release = bioc_release_yaml(),
    org = "Bioconductor"
  )
  setwd("~/bioc/AnnotationHub")
  update_local_repo(
    ".",
    release = bioc_release_yaml(),
    org = "Bioconductor"
  )
}
```

Index

* **utilities**

- extractNEWS, [4](#)
- add_bioc_remote (bioc_remotes), [2](#)
- add_gh_release_branch
 - (branch-release-gh), [3](#)
- add_gh_release_branches
 - (branch-release-gh), [3](#)
- bioc_release_yaml (branch-release-gh), [3](#)
- bioc_remotes, [2](#)
- bioc_version_yaml (branch-release-gh), [3](#)
- branch-release-gh, [3](#)
- extractNEWS, [4](#)
- get-bioc-packages, [5](#)
- get-github-repos, [6](#)
- get_bioc_software_manifest
 - (get-bioc-packages), [5](#)
- get_org_github_repos
 - (get-github-repos), [6](#)
- get_org_packages (get-bioc-packages), [5](#)
- get_user_github_repos
 - (get-github-repos), [6](#)
- getPackagesNEWS, [7](#)
- packages_with_release_branch
 - (packages_without_release_branch),
[8](#)
- packages_without_release_branch, [8](#)
- printNEWS (getPackagesNEWS), [7](#)
- set_bioc_remotes (bioc_remotes), [2](#)
- update_local_repo (update_local_repos),
[9](#)
- update_local_repos, [9](#)