Package: BranchRename (via r-universe)

August 22, 2024

Title Utilities for renaming branches on GitHub

Version 0.99.2

Description The package provides utility functions at the user and organization level for renaming branches on GitHub. The functions are written mainly for modifying batches of either repositories or packages. It uses tools to authenticate and call git from within R.

Imports gert, gh, ReleaseLaunch, utils

License Artistic-2.0

Encoding UTF-8

Remotes Bioconductor/ReleaseLaunch

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

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

RemoteUrl https://github.com/Bioconductor/BranchRename

RemoteRef HEAD

RemoteSha 669d1bf1d9774a4500619b81cbfb615731dd12c9

Contents

add_bioc_remote *A convenience function to set the 'upstream' Bioconductor remote*

Description

The function will create an 'upstream' remote using `git@git.bioconductor.org` as the primary address. If an `upstream` remote already exists, it will be validated. The remote name can be changed to the desired name via the `remote` argument but it is customarily called the 'upstream' remote.

Usage

```
add_bioc_remote(package_path, remote = "upstream")
```

Arguments

`package_path`	`character(1)` The local path to a package directory whose upstream remote should be set
`remote`	`character(1)` The name of the remote to be created. This is usually named 'upstream' (default)

Value

Called for the side effect of creating an 'upstream' remote with the Bioconductor git address for a given package

fix_bioc_remotes *A helper function to adjust git remotes the Bioconductor way*

Description

This function will update a repository's remotes by setting `origin` to the GitHub location and `upstream` to the Bioconductor git server.

Usage

```
fix_bioc_remotes(repo_dir, org = "Bioconductor", is_bioc)
```

Arguments

`repo_dir`	`character(1)` The full path to the directory of the local repository whose default branch should be updated
`org`	`character(1)` The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").
`is_bioc`	`logical(1)` Whether the repository in question is a Bioconductor package

Value

Called for the side-effect of adjusting `git remotes` in the `repo_dir`; check them with `git remote -v`

`packages_with_default_branch` *Find packages with default branches for an organization*

Description

This function will search through an organizations repositories and identify packages whose default branches are in the `branches` argument. This allows the user to identify which repositories will need to have a `devel` branch added.

Usage

```
packages_with_default_branch(
  version = BiocManager::version(),
  branches = c("master", "main"),
  org = "Bioconductor"
)
```

Arguments

version	`character(1)` The current development version of Bioconductor given by `BiocManager::version()` (default). It is used to obtain the packages currently available in the Bioconductor CRAN-like repository with `BiocManager::repositories()`.
branches	`character()` A vector of branches that are sought as default branches
org	`character(1)` The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").

Details

The output of this function is used in `rename_branch_packages`.

Value

A named character vector of default branches whose names correspond to package repositories on GitHub

See Also

rename_branch_packages

rename_branch_packages   *Convenience function to create the devel branch for all GitHub packages*

Description

This function identifies an organization's repositories that are packages given the current version of Bioconductor (from `BiocManager::version()`) and identifies which repositories need to have a `devel` branch added. It then adds the `devel` branch using the `rename_branch_to_devel` function. It is highly recommended that the user run this on the devel version of Bioconductor to avoid missing packages that are only in devel.

Usage

```
rename_branch_packages(
  packages = character(0L),
  version = BiocManager::version(),
  old_branches = c("master", "main"),
  org = "Bioconductor",
  set_upstream = c("origin/devel", "upstream/devel"),
  clone = TRUE
)
```

Arguments

`packages`	`named character()` A character vector of default branches whose names correspond to Bioconductor package names. See `packages_with_default_branch`.
`version`	`character(1)` The current development version of Bioconductor given by `BiocManager::version()` (default). It is used to obtain the packages currently available in the Bioconductor CRAN-like repository with `BiocManager::repositories()`.
`old_branches`	`character()` A vector of default branch names to be replaced, both 'master' and 'main' are included by default. This argument only works when either `packages` or `repos` are not specified.
`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"
`clone`	`logical(1)` Whether to clone the GitHub repository into the current working directory (default: TRUE)

Details

Note that the `clone` argument allows the user to clone the repository first from GitHub via SSH. It is recommended that this be enabled and that the user running this function can clone packages via SSH and have access to modifying packages on the GitHub organization.

See Also

packages_with_default_branch

`rename_branch_repos` *Convenience function to create the devel branch for all GitHub repositories*

Description

This function identifies all repositories within an organization that have `old_branches`, i.e., either 'master' or 'main' by default. It then sets the default branch to `devel`.

Usage

```
rename_branch_repos(
  repos = character(0L),
  old_branches = c("master", "main"),
  org = "Bioconductor",
  set_upstream = c("origin/devel", "upstream/devel"),
  clone = TRUE
)
```

Arguments

`repos`	`named character()` A vector of default branches whose names correspond to repositories hosted on GitHub. If missing, `repos_with_default_branch` is called and its result is used.
`old_branches`	`character()` A vector of default branch names to be replaced, both 'master' and 'main' are included by default. This argument only works when either `packages` or `repos` are not specified.
`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"
`clone`	`logical(1)` Whether to clone the GitHub repository into the current working directory (default: TRUE)

See Also

repos_with_default_branch

rename_branch_to_devel

Create the 'devel' branch locally and on GitHub

Description

The function is meant to be run one level up from the local git repository. It will create the 'devel' branch and push to the `origin` remote which should be set to GitHub. Upstream tracking can be configured to either the `origin` or `upstream` remote.

Usage

```
rename_branch_to_devel(
  package_name,
  from_branch = "master",
  org = "Bioconductor",
  set_upstream = c("origin/devel", "upstream/devel"),
  clone = FALSE,
  is_bioc_pkg = TRUE
)
```

Arguments

`package_name` `character(1)` The name of the organization R package that is also available on GitHub.

`from_branch` `character(1)` The old default branch from which to base the new 'devel' branch from (default: 'master')

`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"

`clone` `logical(1)` Whether to clone the GitHub repository into the current working directory (default: TRUE)

`is_bioc_pkg` `logical(1)` Whether the repository is an R package that has an upstream remote on Bioconductor, i.e., <git@git.bioconductor.org:packages/Package>. If so, additional validity checks will be run on the git remotes.

Details

The `origin` remote is assumed to be GitHub, i.e., `git@github.com:user/package` but this requirement is not checked or enforced; thus, allowing flexibility in remote `origin` locations. The `upstream` remote is validated against the Bioconductor git repository address, i.e., `git@git.bioconductor.org:packages/` The local repository is validated before the `devel` branch is created.

Value

Called for the side effect of creating a 'devel' branch on the local and remote repositories on GitHub

Examples

```
if (interactive()) {

  rename_branch_to_devel(
    package_name = "SummarizedExperiment",
    org = "Bioconductor",
    set_upstream = "upstream/devel"
  )

}
```

---

repos_with_default_branch

Identify repositories that have old default branches

Description

The function obtains all the repositories within the given organization (Bioconductor) that match the `branches` argument.

Usage

```
repos_with_default_branch(branches = c("master", "main"), org = "Bioconductor")
```

Arguments

`branches`	`character()` A vector of branches that are sought as default branches
`org`	`character(1)` The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").

Details

The output of this function is used to rename branches with `rename_branch_repos`.

Value

A named character vector of default branches whose names correspond to organization repositories on GitHub

See Also

rename_branch_repos

`update_local_repos` *Update local repositories after GitHub branch rename*

Description

These functions works *only* after renaming branches on GitHub from `master` to `devel`. It updates local folders so that branches are renamed to `devel` locally. It also ensures that the `HEAD` is pointing to the new remote location, i.e., `origin/devel` (see `gert::git_remote_info()$head`). For convenience, the singular `update_local_repo` function will update a particular local repository on the user's system.

Usage

```
update_local_repos(
  basedir,
  org = "Bioconductor",
  username,
  new_branch = "devel",
  set_upstream = "origin/devel"
)

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

Arguments

`basedir` `character(1)` The base directory where all packages / repositories exist for the user

`org` `character(1)` The organization for which to extract the names of the repositories on GitHub (default "Bioconductor").

`username` `character(1)` (optional) The GitHub username used in the query to check default packages

`new_branch` target name of the branch once the move is performed; this name is validated for consistency.

`set_upstream` `character(1)` The remote location that will be tracked by the local branch, either "origin/devel" (default) or "upstream/devel"

`repo_dir` `character(1)` The full path to the directory of the local repository whose default branch should be updated

`is_bioc` `logical(1)` Whether the repository in question is a Bioconductor package

Details

Note that `update_local_repos` (plural) queries the GitHub API to discover repositories whose default branches are set to `devel`. The function then matches those repository names with local folder names under `basedir` and performs the necessary steps to rename the branch and set the `HEAD` to `origin/devel`. The singular function `update_local_repo` assumes that the remote change has been made and will rename the local branch and update the local repository to match the remote 'HEAD'.

Index