

# Package ‘neurocInstall’

October 13, 2022

**Type** Package

**Title** 'Neuroconductor' Installer

**Version** 0.12.0

**Author** John Muschelli <muschellij2@gmail.com>

**Maintainer** John Muschelli <muschellij2@gmail.com>

**Description** Installs 'Neuroconductor' packages from the release repository <<https://neuroconductor.org/releases/>> or from 'GitHub'.

**Imports** devtools (>= 1.12.0.9000), utils, methods

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat, knitr, rmarkdown, httr

**VignetteBuilder** knitr

**RoxygenNote** 7.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-02-11 19:00:02 UTC

## R topics documented:

latest_neuroc_release . . . . .	2
make_full_version . . . . .	2
neuro_install . . . . .	3
neuro_packages . . . . .	4
neuro_package_table . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

latest\_neuroc\_release *Latest Neuroconductor Release Location*

---

### Description

Latest Neuroconductor Release Location

### Usage

```
latest_neuroc_release(secure = TRUE)

binary_release_repo(release = c("stable", "current"), secure = TRUE)

make_release_version(release_path = NULL, secure = TRUE, check = TRUE)

release_versions(secure = TRUE)
```

### Arguments

secure	Should https vs. http be used
release	Stable or current (development) versions
release_path	path to the release on <a href="https://neuroconductor.org/releases/">https://neuroconductor.org/releases/</a>
check	should the 'release_path' need to be checked against all the releases?

### Value

URL of release page

### Examples

```
make_release_version("2018/02/", check = FALSE)

latest_neuroc_release()
```

---

make\_full\_version *Make Full Package Version*

---

### Description

Makes a package version to have all the same length. This is helpful when using `compareVersion`.

### Usage

```
make_full_version(x)
```

**Arguments**

x                    Character vector of package versions

**Value**

Character vector of versions, each with the same length.

**Examples**

```
x = c("1.6", "1.6.0")
compareVersion(x[1], x[2])
x2 = make_full_version(x)
compareVersion(x2[1], x2[2])
x = c("1.6", "1.6.0")
compareVersion(x2[1], x2[2])
```

---

neuro_install	<i>Neuroconductor Installer</i>
---------------	---------------------------------

---

**Description**

Install function for neuroconductor packages

**Usage**

```
neuro_install(
  repo,
  release = c("stable", "current"),
  release_repo = latest_neuroc_release(),
  upgrade_dependencies = FALSE,
  type = getOption("pkgType"),
  ...
)

neuroc_install(...)

neuroclite(...)
```

**Arguments**

repo                Package name in neuroconductor

release             Stable or current (development) versions/branches

release\_repo        Repository for release repository, passed to [install.packages](#). If release\_repo = "github", then it will install using GitHub. If you set this using [make\\_release\\_version](#) or specify the URL directly, it will override release option.

upgrade\_dependencies Should dependencies be updated? passed to `install` if using `release_repo = "github"`

type character, indicating the type of package to download and install, passed to `install.packages`.

... additional arguments passed to `install.packages` or `install_github` if `release_repo = "github"`

**Value**

Result from `install.packages` or `install_github`

**Examples**

```
tlib = tempfile()
dir.create(tlib, showWarnings = FALSE)
system.time({
  install.packages("oro.asl",
    lib = tlib,
    repos = "https://neuroconductor.org/releases/2019/12/")
})
repos = getOption("repos")
print(repos)
#if (repos["CRAN"] == "@CRAN@") {
#   repos["CRAN"] = "https://cloud.r-project.org"
#   options(repos = repos)
#}
options(repos = NULL)
print(getOption("repos"))
neuro_install("oro.asl", lib = tlib,
  release_repo = "https://neuroconductor.org/releases/2019/12")
options(repos = repos)

neuro_install("cifti", type = "source", lib = tlib)
neuro_install("cifti",
  release_repo = latest_neuroc_release("stable"),
  lib = tlib)

neuro_install("cifti", release_repo = "github")
```

---

neuro\_packages

*Neuroconductor Packages*


---

**Description**

Returns the vector of Neuroconductor packages

**Usage**

```
neuro_packages(...)
```

**Arguments**

... Arguments passed to [neuro\\_package\\_table](#)

**Value**

vector of packages available on Neuroconductor

**Examples**

```
neuro_packages()
```

---

neuro\_package\_table *Neuroconductor Package Table*

---

**Description**

Returns the table of Neuroconductor packages

**Usage**

```
neuro_package_table(  
  path = "https://neuroconductor.org/neurocPackages",  
  long = FALSE,  
  deployment = FALSE  
)
```

**Arguments**

path Path to the table of package  
long Should the data be "long" (with respect to stable/current)  
deployment indicator if this is a release, not standard flag.

**Value**

data.frame of packages with commit IDs

**Note**

Package information is obtained from <https://neuroconductor.org/neurocPackages>

**Examples**

```
neuro_package_table()
```

# Index

binary\_release\_repo  
    (latest\_neuroc\_release), 2

compareVersion, 2

install, 4

install.packages, 3, 4

install\_github, 4

latest\_neuroc\_release, 2

make\_full\_version, 2

make\_release\_version, 3

make\_release\_version  
    (latest\_neuroc\_release), 2

neuro\_install, 3

neuro\_package\_table, 5, 5

neuro\_packages, 4

neuroc\_install(neuro\_install), 3

neuroclite(neuro\_install), 3

release\_versions  
    (latest\_neuroc\_release), 2