Package ‘BioInstaller’

November 20, 2018

Title Integrator of Bioinformatics Resources

Version 0.3.7

Maintainer Jianfeng Li <lee_jianfeng@sjtu.edu.cn>

Description Can be used to integrate massive bioinformatics resources, such as tool/script and database. It provides the R functions and Shiny web application. Hundreds of bioinformatics tool/script and database have been included.

Depends R (>= 3.3.0)

URL https://github.com/JhuangLab/BioInstaller

BugReports https://github.com/JhuangLab/BioInstaller/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports stringr (>= 1.2.0), futile.logger (>= 1.4.1), configr (>= 0.3.3), jsonlite, git2r (>= 0.0.3), R.utils (>= 2.5.0), RCurl (>= 1.95-4.8), rvest (>= 0.3.2), devtools (>= 1.13.2), stringi (>= 1.1.5), shiny, liteq

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, testthat, prettydoc, DT

VignetteBuilder knitr

NeedsCompilation no

Author Jianfeng Li [aut, cre] (<https://orcid.org/0000-0003-2349-208X>)

Repository CRAN

Date/Publication 2018-11-20 15:50:09 UTC

R topics documented:

`BioInstaller` *This package is a new platform to construct interactive and reproducible biological data analysis applications based on R language, which includes the R functions and R Shiny application, REST APIs.*

Description

This package is a new platform to construct interactive and reproducible biological data analysis applications based on R language, which includes the R functions and R Shiny application, REST APIs.

Author(s)

Li Jianfeng `lee_jianfeng@sjtu.edu.cn`

See Also

Useful links:

`https://github.com/JhuangLab/BioInstaller`

Report bugs at `https://github.com/JhuangLab/BioInstaller/issues`

change.info *Update biologly softwares infomation of system*

Description

Update biologly softwares infomation of system

Usage

```
change.info(name = "", installed = TRUE, source.dir = "",
  bin.dir = "", executable.files = "",
  db = Sys.getenv("BIO_SOFTWARES_DB_ACTIVE", tempfile()), ...,
  verbose = TRUE)
```

Arguments

`name`	Software name
`installed`	Wheather be installed successful in system
`source.dir`	Directorie of softwares source code
`bin.dir`	Directorie of softwares bin
`executable.files`	Executable files in bin.dir
`db`	File saving softwares infomation
`...`	Other key and value paired need be saved in BioInstaller
`verbose`	Ligical indicating wheather show the log message

Value

Bool Value

Examples

```
db <- sprintf('%s/.BioInstaller', tempdir())
set.biosoftwares.db(db)
change.info(name = 'demo', installed = 'yes', source.dir = '',
bin.dir = '', excutable.files = c('demo'), others.customer = 'demo')
unlink(db)
```

conda *Wrapper function of conda*

Description

Wrapper function of conda

Usage

```
conda(suffix_params = "", prefix_params = "",
  conda = Sys.which("conda"), ...)
```

Arguments

`suffix_params`	Command line parameters of conda
`prefix_params`	Command line parameters of conda
`conda`	Default is Sys.which('conda')
`...`	Parameters pass to 'system'

Examples

```
## Not run: 
  conda()

## End(Not run)
```

conda.env.create *Wrapper function of 'conda env create', create an environment based on an environment file*

Description

Wrapper function of 'conda env create', create an environment based on an environment file

Usage

```
conda.env.create(env_name = "", env_file = "", env_path = "",
  params = "", ...)
```

Arguments

`env_name`	Name of environment
`env_file`	Environment definition file (default: environment.yml)
`env_path`	Full path to environment prefix
`params`	Extra command line parameters of conda
`...`	Parameters pass to `conda`

Examples

```
## Not run:
  conda.env.create(params = 'vader/deathstar')
  conda.env.create(env_name = 'name')
  conda.env.create(env_file = '/path/to/environment.yml')
  conda.env.create(env_name = 'deathstar',
                   env_file = '/path/to/requirements.txt')
  conda.env.create(env_file = '/path/to/requirements.txt',
  env_path = '/home/user/software/deathstar')

## End(Not run)
```

`conda.env.list` *Wrapper function of 'conda env list', list the Conda environments*

Description

Wrapper function of 'conda env list', list the Conda environments

Usage

```
conda.env.list(...)
```

Arguments

`...`	Parameters pass to `conda`

Examples

```
## Not run:
  conda.env.list()

## End(Not run)
```

`conda.list` *Wrapper function of 'conda list', list linked packages in a conda environment.*

Description

Wrapper function of 'conda list', list linked packages in a conda environment.

Usage

```
conda.list(env_name = "base", ...)
```

Arguments

env_name Name of environment, default is current

... Parameters pass to conda

Examples

```
## Not run:
  conda.list()
  conda.list(env_name = 'your_env')

## End(Not run)
```

copy_configs *Function to copy the default configuration file of BioInstaller*

Description

Function to copy the default configuration file of BioInstaller

Usage

```
copy_configs(config_dir = "~/.BioInstaller/",
  template_dir = Sys.getenv("BIOINSTALLER_SHINY_CONFIG",
  system.file("extdata", "config/shiny/", package = "BioInstaller")),
  pattern = "shiny.config.yaml", auto_create = FALSE)
```

Arguments

config_dir The destdir to store plugins [~/.BioInstaller]

template_dir The template dir system.file('extdata', 'config/shiny/', package = 'BioInstaller')

pattern Used in list.files ['shiny.config.yaml']

auto_create Auto create dir, default is FALSE

Examples

```
copy_configs(tempdir())
```

copy_plugins *Function to copy the default plugins of BioInstaller*

Description

Function to copy the default plugins of BioInstaller

Usage

```
copy_plugins(plugin_dir = "~/.BioInstaller/plugins",
  template_dir = system.file("extdata", "config/shiny/", package =
  "BioInstaller"), pattern = "shiny.*.parameters.toml",
  auto_create = FALSE)
```

Arguments

`plugin_dir`	The destdir to store plugins [~/.BioInstaller/plugins]
`template_dir`	The template dir system.file('extdata', 'config/shiny/', package = 'BioInstaller')
`pattern`	Used in `list.files` ['shiny.*.parameters.toml']
`auto_create`	Auto create dir, default is FALSE

Examples

```
copy_plugins(tempdir())
```

crawl.all.versions *A function can be used to craw all source code from nongithub.cfg stored information*

Description

A function can be used to craw all source code from nongithub.cfg stored information

Usage

```
crawl.all.versions(name, download.dir = "./",
  nongithub.cfg = c(system.file("extdata",
  "config/nongithub/nongithub.toml", package = "BioInstaller"),
  system.file("extdata", "config/db/db_main.toml", package =
  "BioInstaller"), system.file("extdata", "config/db/db_annovar.toml",
  package = "BioInstaller"), system.file("extdata",
  "config/db/db_blast.toml", package = "BioInstaller")),
  parse.extra.params = list(extra.list = list(), rcmd.parse = TRUE,
  bash.parse = TRUE, glue.parse = TRUE), license = "")
```

Arguments

`name`	Software name
`download.dir`	Download destdir
`nongithub.cfg`	Configuration file of installed by non github url, default is system.file('extdata', 'config/nongithub/nongithub.toml', package='BioInstaller')
`parse.extra.params`	Other parameters pass to `parse.extra`
`license`	The BioInstaller download license code.

Examples

```
crawl.all.versions('demo')
```

`del.info` *Delete biologly softwares infomation of system*

Description

Delete biologly softwares infomation of system

Usage

```
del.info(name = "", db = Sys.getenv("BIO_SOFTWARES_DB_ACTIVE",
  tempfile()), verbose = TRUE)
```

Arguments

`name`	Software name
`db`	File saving softwares infomation
`verbose`	Ligical indicating wheather show the log message

Value

Bool Value

Examples

```
db <- sprintf('%s/.BioInstaller', tempdir())
set.biosoftwares.db(db)
change.info(name = 'bwa', installed = 'yes', source.dir = '',
bin.dir = '', excutable.files = c('demo'), others.customer = 'demo')
del.info('bwa')
unlink(db)
```

`docker.pull` *Use docker to pull image*

Description

Use docker to pull image

Usage

```
docker.pull(repo, name, version = NULL, docker.bin = NULL,
  all.tags = FALSE, disable.content.trust = TRUE, verbose = TRUE)
```

Arguments

`repo,`	Repository name of docker hub, e.g life2cloud
`name`	Software name, e.g bwa
`version`	Image version
`docker.bin`	Docker executable file, default is ’docker’ in $PATH
`all.tags`	Download all tagged images in the repository
`disable.content.trust`	Skip image verification (default true)
`verbose`	Ligical indicating wheather show the log message

Value

Bool Value

Examples

```
docker.bin <- unname(Sys.which('docker'))
if (docker.bin != '') {
  docker.pull(repo = 'learn', name = 'tutorial')
}
```

`docker.search` *Search softwares docker infomation in BioInstaller docker database*

Description

Search softwares docker infomation in BioInstaller docker database

Usage

```
docker.search(name, docker.db = system.file("extdata",
  "config/docker/docker.toml", package = "BioInstaller"))
```

Arguments

`name`	Software name, e.g bwa
`docker.db`	A list including docker repo infomation, default to use built-in config/docker/docker.toml

Value

A list

Examples

```
docker.search('bwa')
```

`get.info` *Show biologly softwares infomation of system*

Description

Show biologly softwares infomation of system

Usage

```
get.info(name = "", db = Sys.getenv("BIO_SOFTWARES_DB_ACTIVE",
  tempfile()), verbose = TRUE)
```

Arguments

`name`	Software name
`db`	File saving softwares infomation
`verbose`	Ligical indicating wheather show the log message

Value

Bool Value

Examples

```
db <- sprintf('%s/.BioInstaller', tempdir())
set.biosoftwares.db(db)
change.info(name = 'bwa', installed = 'yes', source.dir = '',
bin.dir = '', excutable.files = c('demo'), others.customer = 'demo')
get.info('bwa')
unlink(db)
```

`get.meta` *Get meta information of BioInstaller collected sources, such as database, GitHub source, non-GitHub source, web source*

Description

Get meta information of BioInstaller collected sources, such as database, GitHub source, non-GitHub source, web source

Usage

```
get.meta(value = NULL, config = NULL, get.meta.files.params = NULL,
  read.config.params = NULL)
```

Arguments

`value` Avaliable option for ‘db‘, ‘github‘, ‘nongithub‘: ‘cfg_meta‘, ‘item‘; for web: ‘item‘

`config` Avaliable option: ‘db‘, ‘db_meta_file‘, ‘github‘, ‘github_meta_file‘, ‘nongithub‘, ‘nongithub_meta_file‘, ‘web‘, ‘web_meta_file‘

`get.meta.files.params` Params pass to `get.meta.files`

`read.config.params` Params pass to `read.config`

Value

List contain the meta files path of BioInstaller collected sources

Examples

```
meta <- get.meta()
db_cfg_meta <- get.meta(config = 'db', value = 'cfg_meta')
db_meta_file <- get.meta(config = 'db_meta_file')
db_cfg_meta_parsed <- get.meta(value = 'cfg_meta', config = 'db',
read.config.params = list(rcmd.parse = TRUE))
```

get.meta.files *Get all BioInstaller meta files path, such as database, GitHub source, non-GitHub source, web source*

Description

Get all BioInstaller meta files path, such as database, GitHub source, non-GitHub source, web source

Usage

```
get.meta.files(db.meta = system.file("extdata", "config/db/db_meta.toml",
  package = "BioInstaller"), github.meta = system.file("extdata",
  "config/github/github_meta.toml", package = "BioInstaller"),
  nongithub.meta = system.file("extdata",
  "config/nongithub/nongithub_meta.toml", package = "BioInstaller"),
  web.meta = system.file("extdata", "config/web/web_meta.toml", package =
  "BioInstaller"))
```

Arguments

`db.meta` Database source meta file path, default is system.file('extdata', 'config/db/db_meta.toml', package = 'BioInstaller')

`github.meta` Github source meta file path, default is system.file('extdata', 'config/github/github_meta.toml', package = 'BioInstaller')

`nongithub.meta` non-Github source meta file path, default is system.file('extdata', 'config/nongithub/nongithub_meta.toml package = 'BioInstaller')

`web.meta` Web source meta file path, default is system.file('extdata', 'config/web/web_meta.toml', package = 'BioInstaller')

Value

List contain the meta files path of BioInstaller collected sources

Examples

```
get.meta.files()
```

install.bioinfo *Download and install biology software or database*

Description

Download and install biology software or database

Usage

```
install.bioinfo(name = c(), download.dir = c(), destdir = c(),
  name.saved = NULL, github.cfg = system.file("extdata",
  "config/github/github.toml", package = "BioInstaller"),
  nongithub.cfg = c(system.file("extdata",
  "config/nongithub/nongithub.toml", package = "BioInstaller"),
  system.file("extdata", "config/db/db_main.toml", package =
  "BioInstaller"), system.file("extdata", "config/db/db_annovar.toml",
  package = "BioInstaller"), system.file("extdata",
  "config/db/db_blast.toml", package = "BioInstaller")), version = c(),
  local.source = NULL, show.all.versions = FALSE,
  show.all.names = FALSE, db = Sys.getenv("BIO_SOFTWARES_DB_ACTIVE",
  system.file("extdata", "demo/softwares_db_demo.yaml", package =
  "BioInstaller")), download.only = FALSE, decompress = TRUE,
  dependence.need = TRUE, showWarnings = FALSE, extra.list = list(),
  rcmd.parse = TRUE, bash.parse = TRUE, glue.parse = TRUE,
  glue.flag = "!!glue", save.to.db = TRUE, license = "",
  overwrite = FALSE, verbose = TRUE, ...)
```

Arguments

`name`	Software name
`download.dir`	A string, point the source code download destdir
`destdir`	A string, point the install path
`name.saved`	Software name when you want to install different version, you can use this to point the installed softwares name like 'GATK-3.7'
`github.cfg`	Configuration file of installed by github url, default is system.file('extdata', 'config/github/github.toml', package='BioInstaller')
`nongithub.cfg`	Configuration file of installed by non github url, default is c(system.file('extdata', 'config/nongithub/nongithub.toml', package = 'BioInstaller'), system.file('extdata', 'config/db/db_main.toml', package = 'BioInstaller'), system.file('extdata', 'config/db/db_annovar.toml', package = 'BioInstaller'), system.file('extdata', 'config/db/db_blast.toml', package = 'BioInstaller'))
`version`	Software version
`local.source`	Install from local source, github softwares need a cloned dir, and nongithub softwares can be installed from a compressed file (if it is a dir, you need set decompress to FALSE)

`show.all.versions`
Logical wheather show all avaliable versions can be install

`show.all.names` Logical wheather show all avaliable names can be install

`db` File of saving softwares infomation, default is Sys.getenv(’BIO_SOFTWARES_DB_ACTIVE’, system.file(’extdata’, ’demo/softwares_db_demo.yaml’, package = ’BioInstaller’))

`download.only` Logicol indicating wheather only download source or file (non-github)

`decompress` Logicol indicating wheather need to decompress the downloaded file, default is TRUE

`dependence.need`
Logical should the dependence should be installed

`showWarnings` Logical should the warnings on failure be shown?

`extra.list` A list that can replace the configuration file ’debug’ by list(debug = TRUE), and debug will be setted to TRUE

`rcmd.parse` Logical wheather parse ’@>@str_replace(’abc’, ’b’, ’c’)@<@’ in config to ’acc’

`bash.parse` Logical wheather parse ’#>#echo $HOME#<#’ in config to your HOME PATH

`glue.parse` Logical wheather parse ’!!glue1:5’ in config to [’1’,’2’,’3’,’4’,’5’]; [’nochange’, ’!!glue(1:5)’, ’nochange’] => [’nochange’, ’1’, ’2’, ’3’, ’4’, ’5’, ’nochange’]

`glue.flag` A character flage indicating wheater run glue() function to parse (Default is !!glue)

`save.to.db` Ligical indicating wheather save the install infomation in db

`license` The BioInstaller download license code.

`overwrite` Force delete the destdir or download dir without a interactive message (careful)

`verbose` Ligical indicating wheather show the log message

`...` Other key and value paired need be saved in BioInstaller passed to `change.info`

Value

Bool Value or a list

Examples

```
db <- sprintf('%s/.BioInstaller', tempdir())
set.biosoftwares.db(db)
tryCatch(install.bioinfo('bwa', show.all.versions = TRUE),
error = function(e) {
  message('Connecting Github failed. Please try it again later.')
})
unlink(db)
```

install.github *Install or download softwares from Github*

Description

Install or download softwares from Github

Usage

```
install.github(name = "", download.dir = NULL, destdir = NULL,
  version = NULL, local.source = NULL, show.all.versions = FALSE,
  name.saved = NULL, github.cfg = system.file("extdata",
  "config/github/github.toml", package = "BioInstaller"),
  db = Sys.getenv("BIO_SOFTWARES_DB_ACTIVE", system.file("extdata",
  "demo/softwares_db_demo.yaml", package = "BioInstaller")),
  download.only = FALSE, showWarnings = FALSE,
  dependence.need = TRUE, extra.list = list(), rcmd.parse = TRUE,
  bash.parse = TRUE, glue.parse = TRUE, glue.flag = "!!glue",
  save.to.db = TRUE, overwrite = FALSE, verbose = TRUE, ...)
```

Arguments

`name` Software name

`download.dir` A string, point the source code download destdir

`destdir` A string, point the install path

`version` Software version

`local.source` Install from local source, github softwares need a cloned dir

`show.all.versions`
Logical wheather show all avaliable version can be install

`name.saved` Software name when you want to install different version, you can use this to point the installed softwares name like 'GATK-3.7'

`github.cfg` Configuration file of installed by github url, default is system.file('extdata', 'config/github/github.toml', package='BioInstaller')

`db` File of saving softwares infomation, default is Sys.getenv('BIO_SOFTWARES_DB_ACTIVE', system.file('extdata', 'demo/softwares_db_demo.yaml', package = 'BioInstaller'))

`download.only` Logicol indicating wheather only download source or file (non-github)

`showWarnings` Logical should the warnings on failure be shown?

`dependence.need`
Logical should the dependence should be installed

`extra.list` A list that can replace the configuration file 'debug' by list(debug = TRUE), and debug will be setted to TRUE

`rcmd.parse` Logical wheather parse '@>@str_replace('abc', 'b', 'c')@<@' in config to 'acc' `bash.parse` Logical wheather parse ’#>#echo $HOME#<#’ in config to your HOME PATH

`glue.parse` Logical wheather parse ’!!glue1:5’ in config to [’1’,’2’,’3’,’4’,’5’]; [’nochange’, ’!!glue(1:5)’, ’nochange’] => [’nochange’, ’1’, ’2’, ’3’, ’4’, ’5’, ’nochange’]

`glue.flag` A character flage indicating wheater run glue() function to parse (Default is !!glue)

`save.to.db` Ligical indicating wheather save the install infomation in db

`overwrite` Force delete the destdir or download dir without a interactive message (careful)

`verbose` Ligical indicating wheather show the log message

`...` Other key and value paired need be saved in BioInstaller passed to `change.info`

Value

Bool Value

Examples

```
db <- sprintf('%s/.BioInstaller', tempdir())
set.biosoftwares.db(db)
tryCatch(install.github('bwa', show.all.versions = TRUE),
error = function(e) {
  message('Connecting Github failed. Please try it again later.')
})
unlink(db)
```

`install.nongithub` *Install or download softwares from non-Github Web site*

Description

Install or download softwares from non-Github Web site

Usage

```
install.nongithub(name = "", download.dir = NULL, destdir = NULL,
  version = NULL, local.source = NULL, show.all.versions = FALSE,
  name.saved = NULL, nongithub.cfg = c(system.file("extdata",
  "config/nongithub/nongithub.toml", package = "BioInstaller"),
  system.file("extdata", "config/db/db_main.toml", package =
  "BioInstaller"), system.file("extdata", "config/db/db_annovar.toml",
  package = "BioInstaller"), system.file("extdata",
  "config/db/db_blast.toml", package = "BioInstaller")),
  db = Sys.getenv("BIO_SOFTWARES_DB_ACTIVE", system.file("extdata",
  "demo/softwares_db_demo.yaml", package = "BioInstaller")),
  download.only = FALSE, decompress = TRUE, dependence.need = TRUE,
  showWarnings = FALSE, extra.list = list(), rcmd.parse = TRUE,
  bash.parse = TRUE, glue.parse = TRUE, glue.flag = "!!glue",
  save.to.db = TRUE, overwrite = FALSE, verbose = TRUE, ...)
```

Arguments

`name` Software name

`download.dir` A string, point the source code download destdir

`destdir` A string, point the install path

`version` Software version

`local.source` Install from local source (a compressed file, if it is a dir, you need set decompress to FALSE)

`show.all.versions` Logical wheather show all avaliable version can be install

`name.saved` Software name when you want to install different version, you can use this to point the installed softwares name like ’GATK-3.7’

`nongithub.cfg` Configuration file of installed by non github url, default is c(system.file(’extdata’, ’config/nongithub/nongithub.toml’, package = ’BioInstaller’), system.file(’extdata’, ’config/db/db_main.toml’, package = ’BioInstaller’), system.file(’extdata’, ’config/db/db_annovar.toml’, package = ’BioInstaller’), system.file(’extdata’, ’config/db/db_blast.toml’, package = ’BioInstaller’))

`db` File of saving softwares infomation, default is Sys.getenv(’BIO_SOFTWARES_DB_ACTIVE’, system.file(’extdata’, ’demo/softwares_db_demo.yaml’, package = ’BioInstaller’))

`download.only` Logicol indicating wheather only download source or file (non-github)

`decompress` Logicol indicating wheather need to decompress the downloaded file, default is TRUE

`dependence.need` Logical should the dependence should be installed

`showWarnings` Logical should the warnings on failure be shown?

`extra.list` A list that can replace the configuration file ’debug’ by list(debug = TRUE), and debug will be setted to TRUE

`rcmd.parse` Logical wheather parse ’@>@str_replace(’abc’, ’b’, ’c’)@<@’ in config to ’acc’

`bash.parse` Logical wheather parse ’#>#echo $HOME#<#’ in config to your HOME PATH

`glue.parse` Logical wheather parse ’!!glue1:5’ in config to [’1’,’2’,’3’,’4’,’5’]; [’nochange’, ’!!glue(1:5)’, ’nochange’] => [’nochange’, ’1’, ’2’, ’3’, ’4’, ’5’, ’nochange’]

`glue.flag` A character flage indicating wheater run glue() function to parse (Default is !!glue)

`save.to.db` Ligical indicating wheather save the install infomation in db

`overwrite` Force delete the destdir or download dir without a interactive message (careful)

`verbose` Ligical indicating wheather show the log message

`...` Other key and value paired need be saved in BioInstaller passed to `change.info`

Value

Bool Value

Examples

```
db <- sprintf('%s/.BioInstaller', tempdir())
set.biosoftwares.db(db)
tryCatch(install.nongithub('gmap', show.all.versions = TRUE),
error = function(e) {
  message('Connecting Gmap website failed. Please try it again later.')
})
unlink(db)
```

is.biosoftwares.db.active *Test active configuration file*

Description

Check whether a Bio Softwares DB is active

Usage

```
is.biosoftwares.db.active(biosoftwares.db)
```

Arguments

`biosoftwares.db`
Configuration filename of bio-softwares db

Value

Logical indicating whether the specified configuration file is active

Examples

```
is.biosoftwares.db.active('config.cfg')
```

new.bioinfo *Create new BioInstaller items to github forum*

Description

Create new BioInstaller items to github forum

Usage

```
new.bioinfo(config.file = "github.toml", title = "",
  description = "", publication = "")
```

Arguments

`config.file`	github.toml, nongithub.toml, db_annovar.toml, db_main.toml, or new
`title`	Name of new item
`description`	Description of new item
`publication`	Publication of new item

Examples

```
new.bioinfo('db_main.toml', 'test_item', 'Just is a test item', 'NA')
```

`set.biosoftwares.db` *Set BIO_SOFWARES_DB_ACTIVE as the BioInstaller db*

Description

Set BIO_SOFWARES_DB_ACTIVE as the BioInstaller db

Usage

```
set.biosoftwares.db(biosoftwares.db)
```

Arguments

`biosoftwares.db`	Configuration filename of bio-softwares db

Value

Logical indicate wheather set db successful

Examples

```
set.biosoftwares.db(sprintf('%s/.BioInstaller', tempdir()))
```

set_shiny_workers *Function to set shiny workers for background service*

Description

Function to set shiny workers for background service

Usage

```
set_shiny_workers(n,
  shiny_config_file = Sys.getenv("BIOINSTALLER_SHINY_CONFIG",
  system.file("extdata", "config/shiny/shiny.config.yaml", package =
  "BioInstaller")), auto_create = FALSE)
```

Arguments

n	Number of needed workers
shiny_config_file	BioInstaller shiny configuration file
auto_create	Auto create log dir, default is FALSE

Examples

```
## Not run:
set_shiny_workers(4)

## End(Not run)
```

show.installed *Show all installed bio-softwares in system*

Description

Show all installed bio-softwares in system

Usage

```
show.installed(db = Sys.getenv("BIO_SOFTWARES_DB_ACTIVE", tempfile()),
  only.installed = TRUE, verbose = TRUE)
```

Arguments

db	File saving softwares infomation
only.installed	Logical wheather only show installed softwares in db
verbose	Ligical indicating wheather show the log message

Value

Bool Value

Examples

```
db <- sprintf('%s/.BioInstaller', tempdir())
set.biosoftwares.db(db)
change.info(name = 'bwa', installed = 'yes', source.dir = '',
bin.dir = '', excutable.files = c('demo'), others.customer = 'demo')
show.installed()
unlink(db)
```

spack *Wrapper function of spack*

Description

Wrapper function of spack

Usage

```
spack(suffix_params = "", prefix_params = "",
  spack = Sys.which("spack"), ...)
```

Arguments

`suffix_params`	Command line parameters of spack (prefix_params spack suffix_params)
`prefix_params`	Command line parameters of spack (prefix_params spack suffix_params)
`spack`	Default is Sys.which('spack')
`...`	Parameters pass to 'system'

Examples

```
## Not run:
  spack()

## End(Not run)
```

spack.list *Wrapper function of 'spack list', list and search available packages*

Description

Wrapper function of 'spack list', list and search available packages

Usage

```
spack.list(...)
```

Arguments

`...`	Parameters pass to `spack`

Examples

```
## Not run:
  spack.list()

## End(Not run)
```

web *Function to run BioInstaller shiny APP service*

Description

Function to run BioInstaller shiny APP service

Usage

```
web(appDir = system.file("extdata", "shiny", package = "BioInstaller"),
  auto_create = FALSE, ...)
```

Arguments

`appDir`	The application to run. Default is system.file('extdata', 'tools/shiny/R', package = 'BioInstaller')
`auto_create`	Auto create dir, default is FALSE
`...`	Other parameters pass to `runApp`

Examples

```
## Not run:
  web(auto_create = TRUE)

## End(Not run)
```

Index