Package ‘beastier’

September 19, 2021

Type Package
Title Call ‘BEAST2’
Version 2.4.8
Maintainer Richèl J.C. Bilderbeek <richel@richelbilderbeek.nl>
Description ‘BEAST2’ (<https://www.beast2.org>) is a widely used Bayesian phylogenetic tool, that uses DNA/RNA/protein data and many model priors to create a posterior of jointly estimated phylogenies and parameters. ‘BEAST2’ is a command-line tool. This package provides a way to call ‘BEAST2’ from an ‘R’ function call.
License GPL-3
Encoding UTF-8
RoxygenNote 7.1.1
Imports ape, assertive, beautier (>= 2.6.2), devtools, phangorn, rappdirs, readr, rJava, stringr, tibble, xml2
Suggests hunspell, knitr, markdown, rmarkdown, spelling, testit, testthat (>= 2.1.0), tracerer
URL https://docs.ropensci.org/beastier/ (website)
https://github.com/ropensci/beastier/
BugReports https://github.com/ropensci/beastier
Language en-US
VignetteBuilder knitr
SystemRequirements BEAST2 (https://www.beast2.org/)
NeedsCompilation no
Author Richèl J.C. Bilderbeek [aut, cre]
(<https://orcid.org/0000-0003-1107-7049>), Joëlle Barido-Sottani [rev] (Joëlle reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/209), David Winter [rev] (David reviewed the package for rOpenSci, see
R topics documented:

https://github.com/ropensci/onboarding/issues/209),
Jason Griffiths [ctb] (<https://orcid.org/0000-0002-1667-8233>),
Thijs Janzen [ctb]

Repository CRAN

Date/Publication 2021-09-19 10:20:05 UTC

R topics documented:

add_quotes_if_has_spaces .................................................. 4
are_beast2_input_lines ...................................................... 5
are_beast2_input_lines_deep .............................................. 6
are_beast2_input_lines_fast .............................................. 7
are_identical_alignments ................................................. 8
beast2_options_to_table ................................................... 8
beastier ................................................................. 9
beastier_report ............................................................ 9
check_beast2 ............................................................ 10
check_beast2_options ...................................................... 11
check_beast2_optionses ................................................... 12
check_beast2_options_data_types ....................................... 12
check_beast2_options_do_not_overwrite_existing_files ............ 13
check_beast2_options_filenames_differ ................................ 14
check_beast2_options_names ............................................. 14
check_beast2_path ........................................................ 15
check_can_create_dir_for_state_output_file ...................... 16
check_can_create_file ................................................... 16
check_can_create_screenlog_file ...................................... 17
check_can_create_state_output_file .................................. 18
check_can_create_tracelog_file ........................................ 18
check_can_create_treelog_file ......................................... 19
check_empty_beastier_folder ............................................. 19
check_input_filename .................................................... 20
check_input_filename_validity ......................................... 21
check_n_threads .......................................................... 22
check_os ................................................................. 22
check_rng_seed ............................................................ 23
continue_beast2 .......................................................... 24
create_beast2_continue_cmd_from_options ............................. 24
create_beast2_input_file_folder ....................................... 25
create_beast2_options .................................................... 26
create_beast2_run_cmd .................................................... 27
create_beast2_run_cmd_from_options ................................ 29
create_beast2_screenlog_folder ....................................... 30
create_beast2_state_output_file_folder .............................. 30
create_beast2_tracelog_folder ......................................... 31
create_beast2_treelog_folder ........................................... 31
create_beast2_validate_cmd ............................................. 32
create_beast2_validate_cmd_bin ........................................ 33
create_beast2_validate_cmd_jar ........................................ 34
create_beast2_version_cmd ............................................ 35
create_beast2_version_cmd_bin ........................................ 35
create_beast2_version_cmd_jar ........................................ 36
create_beastier_tempfolder ........................................... 37
create_beautier_tempfolder ........................................... 37
create_mcbette_beast2_options ....................................... 38
create_random_alignment .............................................. 39
create_random_fasta ................................................... 40
create_random_phylogeny .............................................. 41
create_temp_input_filename ......................................... 42
create_temp_state_filename ......................................... 42
default_params_doc ................................................... 43
do_minimal_run .......................................................... 46
extract_screenlog_filename_from_beast2_input_file ............ 47
extract_trace_log_filename_from_beast2_input_file ............ 47
extract_treelog_filename_from_beast2_input_file .................. 48
get_alignment_ids_from_xml_filename .............................. 49
get_beast2_example_filename ........................................ 50
get_beast2_example_filenames ..................................... 51
get_beast2_main_class_name ........................................ 51
get_beast2_options_filenames ....................................... 52
get_beast2_version ................................................... 53
get_beastier_folder .................................................. 53
get_beastier_path .................................................... 54
get_beastier_paths .................................................... 55
get_beastier_tempfilename ......................................... 56
get_default_beast2_bin_path ....................................... 57
get_default_beast2_download_url .................................... 58
get_default_beast2_download_url_linux ............................. 58
get_default_beast2_download_url_win .............................. 59
get_default_beast2_folder ......................................... 60
get_default_beast2_jar_path ....................................... 60
get_default_beast2_path ............................................ 61
get_default_beast2_version ....................................... 62
give_beast2_warning .................................................. 65
has_unique_ids ......................................................... 66
install_beast2 ........................................................ 67
is_alignment ............................................................ 68
is_beast2_input_file ................................................... 68
is_beast2_installed .................................................... 70
is_bin_path ............................................................. 70
is_jar_path ............................................................. 71
add_quotes_if_has_spaces

Add quotes around the string if it contains spaces.

Description

Add quotes around the string if it contains spaces. Does nothing if the string contains no spaces. This is used for filenames

Usage

add_quotes_if_has_spaces(filename)

Arguments

filename a filename

Value

a filename. If the filename did not contain spaces, it is returned as-is. If the filename did contain spaces, the filename is surrounded by quotes

Author(s)

Richèl J.C. Bilderbeek

Examples

add_quotes_if_has_spaces("x")
add_quotes_if_has_quotes("a b")
are_beast2_input_lines

*Would these lines of text, when written to a file, result in a valid BEAST2 input file?*

**Description**

Would these lines of text, when written to a file, result in a valid BEAST2 input file?

**Usage**

```r
are_beast2_input_lines(
  lines,
  verbose = FALSE,
  method = ifelse(is_on_ci(), "deep", "fast"),
  beast2_path = get_default_beast2_path()
)
```

**Arguments**

- **lines**: lines of text
- **verbose**: if TRUE, additional information is displayed, that is potentially useful in debugging
- **method**: the method to check. Can be 'deep' or 'fast'. The 'deep' method uses BEAST2 to validate the complete file. The 'fast' method uses some superficial tests (for example: if all IDs are unique)
- **beast2_path**: name of either a BEAST2 binary file (usually simply `beast`) or a BEAST2 jar file (usually has a `.jar` extension). Use `get_default_beast2_bin_path` to get the default BEAST binary file’s path Use `get_default_beast2_jar_path` to get the default BEAST jar file’s path

**Value**

TRUE if the text is valid, FALSE if not

**Author(s)**

Richèl J.C. Bilderbeek

**See Also**

Use `is_beast2_input_file` to check a file

**Examples**

```r
if (is_beast2_installed() && is_on_ci()) {
  are_beast2_input_lines(get_beastier_path("anthur_2_4.xml"))
}
```
are_beast2_input_lines_deep

Would these lines of text, when written to a file, result in a valid BEAST2 input file?

Description

Would these lines of text, when written to a file, result in a valid BEAST2 input file?

Usage

are_beast2_input_lines_deep(
  lines,
  verbose = FALSE,
  beast2_path = get_default_beast2_path()
)

Arguments

lines lines of text
verbose if TRUE, additional information is displayed, that is potentially useful in debugging
beast2_path name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_bin_path to get the default BEAST binary file’s path Use get_default_beast2_jar_path to get the default BEAST jar file’s path

Value

TRUE if the text is valid, FALSE if not

Author(s)

Richèl J.C. Bilderbeek

See Also

Use is_beast2_input_file to check a file

Examples

if (is_beast2_installed() && is_on_ci()) {
  beast2_filename <- get_beastier_path("anthus_2_4.xml")
  text <- readLines(beast2_filename)
  are_beast2_input_lines_deep(text)
}
are_beast2_input_lines_fast

Would these lines of text, when written to a file, result in a valid BEAST2 input file?

Description

Would these lines of text, when written to a file, result in a valid BEAST2 input file?

Usage

are_beast2_input_lines_fast(lines)

Arguments

lines  lines of text

Value

TRUE if the text is valid, FALSE if not

Author(s)

Richèl J.C. Bilderbeek

See Also

Use is_beast2_input_file to check a file

Examples

beast2_filename <- get_beastier_path("anthus_2_4.xml")
text <- readLines(beast2_filename)

# TRUE
are_beast2_input_lines_fast(text)
**are_identical_alignments**

*Determines if the two alignments are equal*

**Description**

Determines if the two alignments are equal

**Usage**

```r
ggplot2::ggplot() + gggplot2::geom_bar()```

**Arguments**

- `p`: the first alignment
- `q`: the second alignment

**Value**

TRUE or FALSE

**Author(s)**

Richèl J.C. Bilderbeek

---

**beast2_options_to_table**

*Convert a beast2_options to a table*

**Description**

Convert a beast2_options to a table

**Usage**

```r
beast2_options_to_table(beast2_options)
```

**Arguments**

- `beast2_options`: a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

**Value**

a tibble with two columns, called ‘parameter’ and ‘value’. Each ‘parameter’ is the name of the element of the ‘beast2_options’ structure, where the ‘value’ on the same row holds the value of that parameter
beastier

Author(s)
Richèl J.C. Bilderbeek

Examples
beast2_options_to_table(create_beast2_options())

Description
beastier allows to call BEAST2, a popular Bayesian phylogenetics tool, using an R interface. 'beastier' closely follows the interface of BEAST2, including its default settings.

See Also
These are packages associated with beastier:
- The package beautier can create BEAST2 input files from R
- The package tracerer can parse BEAST2 output files from R
- The package babette combines the functionality of beautier, beastier and tracerer into a single workflow

Examples
beast2_options <- create_beast2_options(
  input_filename = get_beastier_path("2_4.xml")
)
if (is_beast2_installed() && is_on_ci()) {
  run_beast2_from_options(beast2_options)
  file.remove(beast2_options$output_state_filename)
}

beastier_report

Description
Creates a beastier report

Description
Creates a beastier report, to be used when reporting bugs. Uses message
Usage

beastier_report(
    beast2_folder = get_default_beast2_folder(),
    os = rappdirs::app_dir()$os
)

Arguments

beast2_folder  the folder where the BEAST2 is installed. Note that this is not the folder
where the BEAST2 executable is installed: the BEAST2 executable is in a sub-
folder. Use `get_default_beast2_folder` to get the default BEAST2 folder. Use
`get_default_beast2_bin_path` to get the full path to the default BEAST2 exe-
cutable.
os            name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)

Value

No return value, the information will be shown using `message`

Author(s)

Richèl J.C. Bilderbeek

Examples

beastier_report()

check_beast2  Check if BEAST2 is installed properly.

Description

Calls `stop` if BEAST2 is improperly installed

Usage

check_beast2(beast2_path = beastier::get_default_beast2_path())

Arguments

beast2_path        name of either a BEAST2 binary file (usually simply `beast`) or a BEAST2 jar
file (usually has a `.jar` extension). Use `get_default_beast2_bin_path` to get the
default BEAST binary file’s path Use `get_default_beast2_jar_path` to get the
default BEAST jar file’s path

Value

nothing Will `stop` if BEAST2 is improperly installed
check_beast2_options

Author(s)
Richèl J.C. Bilderbeek

Examples
if (is_beast2_installed()) {
    check_beast2()
}

check_beast2_options Check if the beast2_options is a valid BEAST2 options object.

Description
Calls stop if the BEAST2 option object is invalid

Usage
check_beast2_options(beast2_options)

Arguments
beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value
nothing Will stop if the BEAST2 option object is invalid

Author(s)
Richèl J.C. Bilderbeek

See Also
Use create_beast2_options to create a valid BEAST2 options object

Examples
check_beast2_options(create_beast2_options())
check_beast2_optionses

*Check if the beast2_options is a valid BEAST2 options object.*

**Description**

Calls stop if the BEAST2 option object is invalid

**Usage**

```r
check_beast2_optionses(beast2_optionses)
```

**Arguments**

- `beast2_optionses`
  - list of one or more `beast2_options` structures, as can be created by `create_beast2_options`.
  - Use of reduplicated plural to achieve difference with `beast2_options`

**Value**

Nothing. Will stop if the BEAST2 option object is invalid

**Author(s)**

Richèl J.C. Bilderbeek

**See Also**

Use `create_beast2_options` to create a valid BEAST2 options object

**Examples**

```r
check_beast2_optionses(list(create_beast2_options()))
```

---

check_beast2_options_data_types

*Check if the beast2_options, which is a list, has all elements of the right data types*

**Description**

Calls stop if not.

**Usage**

```r
check_beast2_options_data_types(beast2_options)
```
Arguments

beast2_options  a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use `check_beast2_options` to check the entire `beast2_options` object

Description

Check if the `beast2_options` will not overwrite existing files, when the 'overwrite' options is set to FALSE.

Usage

`check_beast2_options_do_not_overwrite_existing_files(beast2_options)`

Arguments

beast2_options  a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

Details

Will stop if a file is threatened to be overwritten

Value

Nothing. Will stop if a file is threatened to be overwritten

Author(s)

Richèl J.C. Bilderbeek
check_beast2_options_names

Examples

```r
check_beast2_options_do_not_overwrite_existing_files(
  beast2_options = create_beast2_options()
)
```

check_beast2_options_filenames_differ

*Check if the filenames in beast2_options differ*

Description

Calls stop if not.

Usage

```r
check_beast2_options_filenames_differ(beast2_options)
```

Arguments

- `beast2_options` a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use `check_beast2_options` to check the entire `beast2_options` object
check_beast2_path

Arguments

beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use check_beast2_options to check the entire beast2_options object

Description

Checks the BEAST2 .jar path. Will stop if there is a problem with the BEAST2 .jar path.

Usage

check_beast2_path(beast2_path)

Arguments

beast2_path name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_bin_path to get the default BEAST binary file’s path Use get_default_beast2_jar_path to get the default BEAST jar file’s path

Value

nothing. Will call stop if the BEAST2 .jar path has a problem

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed()) {
  beast2_path <- get_default_beast2_jar_path()
  check_beast2_path(beast2_path)
}
check_can_create_dir_for_state_output_file

*Internal function*

**Description**
Check if the folder for the state output file can be created. Will *stop* otherwise.

**Usage**
```r
check_can_create_dir_for_state_output_file(beast2_options)
```

**Arguments**
- `beast2_options` a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

**Value**
Nothing. Will *stop* if the folder for the state output file cannot be created.

**Author(s)**
Richèl J.C. Bilderbeek

**Examples**
```r
check_can_create_dir_for_state_output_file(
  beast2_options = create_beast2_options()
)
```

check_can_create_file

*Internal function*

**Description**
Check that a file can be created at a certain path.

**Usage**
```r
check_can_create_file(filename, overwrite = TRUE)
```

**Arguments**
- `filename` file that may or may not be created
- `overwrite` if TRUE, if `filename` already exists, it will be deleted by this function
check_can_create_screenlog_file

Details

Will stop if not. Will stop if the file already exists. Does so by creating an empty file at the path, and then deleting it.

Value

Nothing. Will stop if a file cannot be created at a certain path.

Author(s)

Richèl J.C. Bilderbeek

Examples

check_can_create_file("my_local_file.txt")

check_can_create_screenlog_file

Internal function

Description

Check if the MCMC’s screenlog file can be created. Will stop if not

Usage

check_can_create_screenlog_file(beast2_options)

Arguments

beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value

Nothing. Will stop if the MCMC’s screenlog file cannot be created.

Author(s)

Richèl J.C. Bilderbeek
check_can_create_tracelog_file

Internal function to check if the MCMC’s tracelog file can be created.

Description
Check if the MCMC’s tracelog file can be created. Will stop if not. If the tracelog file already exists, it is assumed that a new file can be created, by overwriting the existing one.

Usage
check_can_create_tracelog_file(beast2_options)

Arguments
beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value
Nothing. Will stop if the MCMC’s tracelog file is absent and cannot be created.
**check_can_create_treelog_file**

*Internal function*

**Description**

Check if the MCMC’s treelog file can be created. Will *stop* if not.

**Usage**

check_can_create_treelog_file(beast2_options)

**Arguments**

beast2_options  a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

**Value**

Nothing. Will *stop* if the MCMC’s treelog file is absent and cannot be created.

**Author(s)**

Richêl J.C. Bilderbeek

---

**check_empty_beastier_folder**

*Check there are no files in the default beastier folder*

**Description**

Check there are no files in the default *beastier* folder. The goal is to make sure no temporary files are left undeleted. Will *stop* if there are files in the *beastier* folder.

**Usage**

check_empty_beastier_folder(beastier_folder = get_beastier_folder())

**Arguments**

beastier_folder  the path to the *beastier* temporary files folder
check_input_filename

Value

Nothing. Will stop if there are files in the beastier folder

Author(s)

Richèl J.C. Bilderbeek

Description

Checks the input filename. Will stop if there is a problem with the input filename.

Usage

check_input_filename(input_filename)

Arguments

input_filename the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

Value

Nothing. Will stop if the input file is invalid

Author(s)

Richèl J.C. Bilderbeek

Examples

check_input_filename(
    get_beastier_path("beast2_example_output.log")
)
check_input_filename_validity

Checks the input filename. Will stop if there is a problem with the input filename.

Description

Checks the input filename. Will stop if there is a problem with the input filename.

Usage

check_input_filename_validity(
    beast2_options,
    input_filename = "deprecated",
    beast2_path = "deprecated",
    verbose = "deprecated"
)

Arguments

beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

input_filename the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

beast2_path name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_bin_path to get the default BEAST binary file’s path Use get_default_beast2_jar_path to get the default BEAST jar file’s path

verbose if TRUE, additional information is displayed, that is potentially useful in debugging

Value

nothing. Will call stop if the input file is invalid

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed()) {
    check_input_filename_validity(
        create_beast2_options(
            input_filename = get_beastier_path("2_4.xml")
        )
    )
**check_n_threads**

*Check if the input is a valid number of threads.*

**Description**

Will stop if not.

**Usage**

`check_n_threads(n_threads)`

**Arguments**

- `n_threads`: the number of computational threads to use. Use NA to use the BEAST2 default of 1.

**Value**

Nothing. Will stop if the number of threads invalid.

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

# Can have 1 or more threads
check_n_threads(1)
check_n_threads(2)

# Can have NA threads
check_n_threads(NA)

---

**check_os**

*Checks if the operating system is supported*

**Description**

Checks if the operating system is supported.

**Usage**

`check_os(os)`
check_rng_seed

**Arguments**

- **os**
  name of the operating system, must be unix (Linux, Mac) or win (Windows)

**Value**

Nothing. Will **stop** if the OS is unsupported

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
check_os("mac")
check_os("unix")
check_os("win")
```

---

check_rng_seed  
*x*

**Description**

Will **stop** if not.

**Usage**

`check_rng_seed(rng_seed)`

**Arguments**

- **rng_seed**
  the random number generator seed of the BEAST2 run. Must be a non-zero positive integer value or **NA**. If rng_seed is **NA**, BEAST2 will pick a random seed

**Value**

Nothing. Will **stop** if the RNG seed is invalid

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
# Numbers from 1 and higher are valid RNG seeds
check_rng_seed(1)
check_rng_seed(2)
# Also NA is a valid RNG seed
check_rng_seed(NA)
```
continue_beast2  

*Continue a BEAST2 run*

**Description**

Continue a BEAST2 run

**Usage**

```r
continue_beast2(beast2_options = create_beast2_options())
```

**Arguments**

- `beast2_options`  
a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

**Value**

The text sent to STDOUT and STDERR. It will create the file with name `output_state_filenames`

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
if (is_beast2_installed() && is_on_ci()) {
  beast2_options <- create_beast2_options(
    input_filename = get_beastier_path("2_4.xml")
  )
  run_beast2_from_options(beast2_options)
  continue_beast2(beast2_options)
  file.remove(beast2_options$output_state_filename)
}
```

---

create_beast2_continue_cmd_from_options  

*Creates the terminal command to run BEAST2 from a beast2_options*

**Description**

If the BEAST2 input .xml filename or the BEAST2 state .state.xml filename contain spaces, these filenames are quoted, so that the command-line interface to BEAST2 correctly parses its arguments
create_beast2_input_file_folder

Usage

create_beast2_input_file_folder(beast2_options)

Arguments

beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value

a character vector with the command and arguments to call BEAST2

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed()) {
  create_beast2_input_file_folder(
    beast2_options = create_beast2_options()
  )
}

create_beast2_input_file_folder

Create the folder where the BEAST2 input file will be created

Description

Create the folder where the BEAST2 input file will be created

Usage

create_beast2_input_file_folder(beast2_options)

Arguments

beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value

nothing

Author(s)

Richèl J.C. Bilderbeek
**create_beast2_options**

Function to create a set of BEAST2 options.

**Description**

These BEAST2 options are the R equivalent of the command-line options.

**Usage**

```r
create_beast2_options(
  input_filename = create_temp_input_filename(),
  output_state_filename = create_temp_state_filename(),
  rng_seed = NA,
  n_threads = NA,
  use_beagle = FALSE,
  overwrite = TRUE,
  beast2_path = get_default_beast2_path(),
  verbose = FALSE,
  output_log_filename = "deprecated",
  output_trees_filenames = "deprecated",
  beast2_working_dir = "deprecated"
)
```

**Arguments**

- **input_filename**  
  the name of a BEAST2 input XML file. This file usually has an .xml extension. Use `create_temp_input_filename` to create a temporary filename with that extension.

- **output_state_filename**  
  name of the .xml.state file to create. Use `create_temp_state_filename` to create a temporary filename with that extension.

- **rng_seed**  
  the random number generator seed of the BEAST2 run. Must be a non-zero positive integer value or NA. If `rng_seed` is NA, BEAST2 will pick a random seed.

- **n_threads**  
  the number of computational threads to use. Use NA to use the BEAST2 default of 1.

- **use_beagle**  
  use BEAGLE if present

- **overwrite**  
  if TRUE: overwrite the .log and .trees files if one of these exists. If FALSE, BEAST2 will not be started if
  - the .log file exists
  - the .trees files exist

```r
beast2_options <- create_beast2_options()
create_beast2_input_file_folder(beast2_options)
```
create_beast2_run_cmd

- the .log file created by BEAST2 exists
- the .trees files created by BEAST2 exist

beast2_path
name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_bin_path to get the default BEAST binary file’s path Use get_default_beast2_jar_path to get the default BEAST jar file’s path

verbose
if TRUE, additional information is displayed, that is potentially useful in debugging

output_log_filename
name of the .log file to create

output_trees_filenames
one or more names for .trees file to create. There will be one .trees file created per alignment in the input file. The number of alignments must equal the number of .trees filenames, else an error is thrown. Alignments are sorted alphabetically by their IDs

beast2_working_dir
a folder where BEAST2 can work in isolation. For each BEAST2 run, a new subfolder is created in that folder. Within this folder, BEAST2 is allowed to create all of its output files, without the risk of overwriting existing ones, allowing BEAST2 to run in multiple parallel processes.

Value

a BEAST2 options structure, which is a list of all function arguments, of which all elements are checked (by check_beast2_options)

Author(s)

Richèl J.C. Bilderbeek

Examples

beast2_options <- create_beast2_options()
check_beast2_options(beast2_options)

create_beast2_run_cmd

Creates the terminal command to run BEAST2

Description

Creates the terminal command to run BEAST2
Usage

create_beast2_run_cmd(
  input_filename,
  output_state_filename,
  rng_seed = NA,
  n_threads = NA,
  use_beagle = FALSE,
  overwrite = FALSE,
  beast2_path = get_default_beast2_path(),
  verbose = FALSE
)

Arguments

input_filename  the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.
output_state_filename  name of the BEAST2 output file that stores the state (usually has a .xml.state extension)
rng_seed  the random number generator seed of the BEAST2 run. Must be a non-zero positive integer value or NA. If rng_seed is NA, BEAST2 will pick a random seed
n_threads  the number of computational threads to use. Use NA to use the BEAST2 default of 1.
use_beagle  use BEAGLE if present
overwrite  if TRUE: overwrite the .log and .trees files if one of these exists. If FALSE, BEAST2 will not be started if
  • the .log file exists
  • the .trees files exist
  • the .log file created by BEAST2 exists
  • the .trees files created by BEAST2 exist
beast2_path  name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_bin_path to get the default BEAST binary file’s path Use get_default_beast2_jar_path to get the default BEAST jar file’s path
verbose  if TRUE, additional information is displayed, that is potentially useful in debugging

Value

a character vector with the command and arguments to call BEAST2

Author(s)

Richèl J.C. Bilderbeek
create_beast2_run_cmd_from_options

Creates the terminal command to run BEAST2 from a beast2_options

Usage

create_beast2_run_cmd_from_options(beast2_options)

Arguments

beast2_options  a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value

a character vector with the command and arguments to call BEAST2

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed()) {
    create_beast2_run_cmd_from_options(
        beast2_options = create_beast2_options()
    )
}

if (is_beast2_installed()) {
    create_beast2_run_cmd(
        input_filename = "input.xml",
        output_state_filename = "output.xml.state",
        beast2_path = get_default_beast2_jar_path()
    )
}

if (is_beast2_installed()) {
    create_beast2_run_cmd_from_options(
        beast2_options = create_beast2_options()
    )
}
create_beast2_screenlog_folder

Internal function

Description
Create the folder for the BEAST2 screenlog file

Usage
create_beast2_screenlog_folder(beast2_options)

Arguments
beast2_options    a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Author(s)
Richèl J.C. Bilderbeek

create_beast2_state_output_file_folder

Create the folder where the BEAST2 state output file will be created

Description
Create the folder where the BEAST2 state output file will be created

Usage
create_beast2_state_output_file_folder(beast2_options)

Arguments
beast2_options    a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value
nothing

Author(s)
Richèl J.C. Bilderbeek
create_beast2_tracelog_folder

Examples

```r
beast2_options <- create_beast2_options()
create_beast2_state_output_file_folder(beast2_options)
```

Description

Create the folder for the BEAST2 tracelog file

Usage

```r
create_beast2_tracelog_folder(beast2_options)
```

Arguments

- `beast2_options`: a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

Author(s)

Richèl J.C. Bilderbeek

create_beast2_treelog_folder

Description

Create the folder for the BEAST2 treelog file

Usage

```r
create_beast2_treelog_folder(beast2_options)
```

Arguments

- `beast2_options`: a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

Author(s)

Richèl J.C. Bilderbeek
create_beast2_validate_cmd

Creates the terminal command to validate a BEAST2 input file

Description

Creates the terminal command to validate a BEAST2 input file

Usage

create_beast2_validate_cmd(
    input_filename,
    beast2_path = get_default_beast2_path()
)

Arguments

input_filename  the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

beast2_path     name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_bin_path to get the default BEAST binary file's path Use get_default_beast2_jar_path to get the default BEAST jar file's path

Value

a character vector, of which the first element is the command (java, in this case), and the others are arguments (-jar, in this case, followed by more arguments.

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed() && is_on_ci()) {
    create_beast2_validate_cmd(
        input_filename = "input.xml"
    )
}
create_beast2_validate_cmd_bin

Description

Creates the terminal command to validate a BEAST2 input file using a call to the launcher.jar file

Usage

create_beast2_validate_cmd_bin(
    input_filename,
    beast2_bin_path = get_default_beast2_bin_path()
)

Arguments

input_filename  the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

beast2_bin_path name of the BEAST2 binary file (usually simply beast). Use get_default_beast2_bin_path to get the default BEAST binary file’s path

Value

a character vector, of which the first element is the command (java, in this case), and the others are arguments (-jar, in this case, followed by more arguments.

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed() && is_on_ci()) {
    create_beast2_validate_cmd_bin(
        input_filename = "input.xml"
    )
}
create_beast2_validate_cmd_jar

Creates the terminal command to validate a BEAST2 input file using a call to the launcher.jar file

Description

Creates the terminal command to validate a BEAST2 input file using a call to the launcher.jar file

Usage

create_beast2_validate_cmd_jar(
    input_filename,
    beast2_jar_path = get_default_beast2_jar_path()
)

Arguments

input_filename the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

beast2_jar_path name of the BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_jar_path to get the default BEAST jar file's path

Value

a character vector, of which the first element is the command (java, in this case), and the others are arguments (-jar, in this case, followed by more arguments.

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (is_beast2_installed() && is_on_ci()) {
    create_beast2_validate_cmd_jar(
        input_filename = "input.xml"
    )
}
```
create_beast2_version_cmd

*Creates the terminal command to version a BEAST2 input file*

**Description**

Creates the terminal command to version a BEAST2 input file

**Usage**

`create_beast2_version_cmd(beast2_path = beastier::get_default_beast2_path())`

**Arguments**

- `beast2_path`: name of either a BEAST2 binary file (usually simply `beast`) or a BEAST2 jar file (usually has a `.jar` extension). Use `get_default_beast2_bin_path()` to get the default BEAST binary file’s path. Use `get_default_beast2_jar_path()` to get the default BEAST jar file’s path.

**Value**

A character vector, of which the first element is the command (`java`, in this case), and the others are arguments (`-jar`, in this case, followed by more arguments).

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
if (is_beast2_installed()) {
  create_beast2_version_cmd()
}
```

create_beast2_version_cmd_bin

*Creates the terminal command to version a BEAST2 input file using a call to the launcher.jar file*

**Description**

Creates the terminal command to version a BEAST2 input file using a call to the `launcher.jar` file

**Usage**

`create_beast2_version_cmd_bin(beast2_bin_path = get_default_beast2_bin_path())`
create_beast2_version_cmd_jar

Arguments

beast2_bin_path

name of the BEAST2 binary file (usually simply beast). Use get_default_beast2_bin_path to get the default BEAST binary file’s path

Value

a character vector, of which the first element is the command (java, in this case), and the others are arguments (-jar, in this case, followed by more arguments.

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed() && is_on_ci()) {
  create_beast2_version_cmd_bin()
}

create_beast2_version_cmd_jar

Creates the terminal command to version a BEAST2 input file using a call to the launcher.jar file

Description

Creates the terminal command to version a BEAST2 input file using a call to the launcher.jar file

Usage

create_beast2_version_cmd_jar(beast2_jar_path = get_default_beast2_jar_path())

Arguments

beast2_jar_path

name of the BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_jar_path to get the default BEAST jar file’s path

Value

a character vector, of which the first element is the command (java, in this case), and the others are arguments (-jar, in this case, followed by more arguments.

Author(s)

Richèl J.C. Bilderbeek
create_beastier_tempfolder

Description
Create the temporary folder as used by beastier

Examples
if (is_beast2_installed()) {
    create_beast2_version_cmd_jar()
}

create_beautier_tempfolder

Description
Create the temporary folder as used by beautier

Examples
create_beautier_tempfolder()
create_mcbette_beast2_options

Create a 'beast2_options' structure for the 'mcbette' R package

Description

Create a 'beast2_options' structure to be used for the 'mcbette' R package, which is a package that allows one to do model comparison. The generated filenames indicating 'mcbette' usage, as well as the correct BEAST2 binary/executable type

Usage

```r
create_mcbette_beast2_options(
  input_filename = beastier::create_temp_input_filename(),
  output_state_filename = beastier::create_temp_state_filename(),
  rng_seed = NA,
  n_threads = NA,
  use_beagle = FALSE,
  overwrite = TRUE,
  beast2_bin_path = beastier::get_default_beast2_bin_path(),
  verbose = FALSE
)
```

Arguments

- `input_filename` the name of a BEAST2 input XML file. This file usually has an .xml extension. Use `create_temp_input_filename` to create a temporary filename with that extension.
- `output_state_filename` name of the .xml.state file to create. Use `create_temp_state_filename` to create a temporary filename with that extension.
- `rng_seed` the random number generator seed of the BEAST2 run. Must be a non-zero positive integer value or NA. If `rng_seed` is NA, BEAST2 will pick a random seed
- `n_threads` the number of computational threads to use. Use NA to use the BEAST2 default of 1.
- `use_beagle` use BEAGLE if present
- `overwrite` if TRUE: overwrite the .log and .trees files if one of these exists. If FALSE, BEAST2 will not be started if
create_random_alignment

- the .log file exists
- the .trees files exist
- the .log file created by BEAST2 exists
- the .trees files created by BEAST2 exist

beast2_bin_path
name of the BEAST2 binary file (usually simply beast). Use get_default_beast2_bin_path
to get the default BEAST binary file’s path

verbose
if TRUE, additional information is displayed, that is potentially useful in debug-

ging

Value

a `beast2_options` structure suitable to be used by the `mcbette` R package, which is a list of all
function arguments, of which all elements are checked (by check_beast2_options)

Author(s)

Richèl J.C. Bilderbeek

See Also
to create a regular (that is, not intended for model comparison) BEAST2 options structure, use
create_beast2_options

Examples

create_mcbette_beast2_options()

create_random_alignment

Create a random alignment

Description

Create a random alignment

Usage

create_random_alignment(n_taxa, sequence_length, rate = 1, taxa_name_ext = ")"

Arguments

n_taxa
sequence_length
rate
taxa_name_ext

The number of taxa
The number of base pairs the alignment will have
mutation rate
the extension of the taxa names
Value

an alignment of class DNAbin

Author(s)

Richèl J.C. Bilderbeek

Examples

alignment <- create_random_alignment(
  n_taxa = 5,
  sequence_length = 10
)
image(alignment)

create_random_fasta  Create a random FASTA file

Description

Create a random FASTA file

Usage

create_random_fasta(
  n_taxa, sequence_length, fasta_filename,
  taxa_name_ext = ""
)

Arguments

n_taxa    The number of taxa
sequence_length    a DNA sequence length, in base pairs
fasta_filename    a FASTA filename.
taxa_name_ext    the extension of the taxa names

Value

Nothing, creates a FASTA file

Author(s)

Richèl J.C. Bilderbeek
create_random_phylogeny

Create a random phylogeny

Description

Create a random phylogeny

Usage

create_random_phylogeny(n_taxa, taxa_name_ext = "")

Arguments

n_taxa The number of taxa
taxa_name_ext the extension of the taxa names

Value

a phylogeny of class 'phylo' (which is part of the 'ape' package)

Author(s)

Richèl J.C. Bilderbeek

Examples

create_random_phylogeny(n_taxa = 6)
create_temp_input_filename

Create a temporary filename for the BEAST2 XML filename

Description
Create a temporary filename for the BEAST2 XML filename

Usage
create_temp_input_filename()

Value
a temporary filename, that starts with ‘beast2_’ and has extension ‘.xml’

Author(s)
Richèl J.C. Bilderbeek

Examples
create_temp_input_filename()

create_temp_state_filename

Create a temporary file for the BEAST2 XML output file that stores its state.

Description
Create a temporary file for the BEAST2 XML output file that stores its state.

Usage
create_temp_state_filename()

Value
a temporary filename, that starts with ‘beast2_’ and has extension ‘.xml.state’

Author(s)
Richèl J.C. Bilderbeek

Examples
create_temp_state_filename()
This function does nothing. It is intended to inherit its parameters’ documentation.

**Description**

This function does nothing. It is intended to inherit its parameters’ documentation.

**Usage**

default_params_doc(
    beast2_bin_path,
    beast2_folder,
    beast2_jar_path,
    beast2_options,
    beast2_optionsets,
    beast2_path,
    beast2_version,
    beast2_working_dir,
    beastier_folder,
    clock_model,
    clock_models,
    crown_age,
    crown_ages,
    fasta_filename,
    fasta_filenames,
    fixed_crown_age,
    fixed_crown_ages,
    initial_phylogenies,
    input_filename,
    mcmc,
    misc_options,
    n_taxa,
    n_threads,
    os,
    output_filename,
    output_log_filename,
    output_state_filename,
    output_trees_filenames,
    overwrite,
    rename_fun,
    rng_seed,
    sequence_length,
    site_model,
    site_models,
    tree_prior,
    tree_priors,
Arguments

**beast2_bin_path**
- name of the BEAST2 binary file (usually simply beast). Use `get_default_beast2_bin_path` to get the default BEAST binary file’s path

**beast2_folder**
- the folder where the BEAST2 is installed. Note that this is not the folder where the BEAST2 executable is installed: the BEAST2 executable is in a sub-folder. Use `get_default_beast2_folder` to get the default BEAST2 folder. Use `get_default_beast2_bin_path` to get the full path to the default BEAST2 executable.

**beast2_jar_path**
- name of the BEAST2 jar file (usually has a .jar extension). Use `get_default_beast2_jar_path` to get the default BEAST jar file’s path

**beast2_options**
- a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`

**beast2_optionses**
- list of one or more `beast2_options` structures, as can be created by `create_beast2_options`. Use of reduplicated plural to achieve difference with `beast2_options`

**beast2_path**
- name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use `get_default_beast2_bin_path` to get the default BEAST binary file’s path Use `get_default_beast2_jar_path` to get the default BEAST jar file’s path

**beast2_version**
- the version of BEAST2. By default, this is the version as returned by `get_default_beast2_version`

**beast2_working_dir**
- a folder where BEAST2 can work in isolation. For each BEAST2 run, a new subfolder is created in that folder. Within this folder, BEAST2 is allowed to create all of its output files, without the risk of overwriting existing ones, allowing BEAST2 to run in multiple parallel processes.

**beastier_folder**
- the path to the `beastier` temporary files folder

**clock_model**
- a `beastier` clock model

**clock_models**
- a list of one or more `beastier` clock models

**crown_age**
- the crown age of the phylogeny

**crown_ages**
- the crown ages of the phylogenies. Set to NA if the crown age needs to be estimated

**fasta_filename**
- a FASTA filename.

**fasta_filenames**
- One or more FASTA filenames.

**fixed_crown_age**
- determines if the phylogeny’s crown age is fixed. If FALSE, crown age is estimated by BEAST2. If TRUE, the crown age is fixed to the crown age of the initial phylogeny.
fixed_crown_ages
one or more booleans to determine if the phylogenies’ crown ages are fixed. If FALSE, crown age is estimated by BEAST2. If TRUE, the crown age is fixed to the crown age of the initial phylogeny.

initial_phylogenies
one or more MCMC chain’s initial phylogenies. Each one set to NA will result in BEAST2 using a random phylogeny. Else the phylogeny is assumed to be of class ape::phylo.

input_filename the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

mcmc one beautier MCMC

misc_options one beautier misc_options object

n_taxa The number of taxa

n_threads the number of computational threads to use. Use NA to use the BEAST2 default of 1.

os name of the operating system, must be unix (Linux, Mac) or win (Windows)

output_filename Name of the XML parameter file created by this function. BEAST2 uses this file as input.

output_log_filename name of the .log file to create

output_state_filename name of the .xml.state file to create. Use create_temp_state_filename to create a temporary filename with that extension.

output_trees_filenames one or more names for .trees file to create. There will be one .trees file created per alignment in the input file. The number of alignments must equal the number of .trees filenames, else an error is thrown. Alignments are sorted alphabetically by their IDs

overwrite if TRUE: overwrite the .log and .trees files if one of these exists. If FALSE, BEAST2 will not be started if

- the .log file exists
- the .trees files exist
- the .log file created by BEAST2 exists
- the .trees files created by BEAST2 exist

rename_fun a function to rename a filename, as can be checked by check_rename_fun. This function should have one argument, which will be a filename or NA. The function should return one filename (when passed one filename) or one NA (when passed one NA). Example rename functions are:

- get_remove_dir_fun get a function that removes the directory paths from the filenames, in effect turning these into local files
- get_replace_dir_fun get a function that replaces the directory paths from the filenames
do_minimal_run

- **get_remove_hex_fun** get a function that removes the hex string from filenames. For example, tracelog_82c1a522040.log becomes tracelog.log

rng_seed  
the random number generator seed of the BEAST2 run. Must be a non-zero positive integer value or `NA`. If `rng_seed` is `NA`, BEAST2 will pick a random seed

sequence_length  
a DNA sequence length, in base pairs

site_model  
a beautier site model

site_models  
one or more beautier site models

tree_prior  
a beautier tree prior

tree_priors  
one or more beautier tree priors

use_beagle  
use BEAGLE if present

verbose  
if TRUE, additional information is displayed, that is potentially useful in debugging

Value

Nothing. This is an internal function that does nothing

Note

This is an internal function, so it should be marked with `@noRd`. This is not done, as this will disallow all functions to find the documentation parameters

Author(s)

Richèl J.C. Bilderbeek

---

do_minimal_run  
*Do a minimal BEAST2 run*

Description

To achieve this, `run_beast2_from_options` is called.

Usage

```r
do_minimal_run()
```

Value

The text sent to STDOUT and STDERR. It will create the files with name `output_state_filename`

Author(s)

Richèl J.C. Bilderbeek
Examples

```java
if (is_beast2_installed() && is_on_ci()) {
    do_minimal_run()
}
```

---

**extract_screenlog_filename_from_beast2_input_file**

*Internal function to extract the screenlog filename for a BEAST2 input file*

**Description**

Extract the screenlog filename from a BEAST2 input file

**Usage**

```python
extract_screenlog_filename_from_beast2_input_file(input_filename)
```

**Arguments**

- `input_filename` the name of a BEAST2 input XML file. This file usually has an `.xml` extension. Use `create_temp_input_filename` to create a temporary filename with that extension.

**Value**

the screenlog filename for a BEAST2 input file

**Author(s)**

Richèl J.C. Bilderbeek

---

**extract_tracelog_filename_from_beast2_input_file**

*Internal function to extract the tracelog filename for a BEAST2 input file*

**Description**

Extract the tracelog filename for a BEAST2 input file

**Usage**

```python
extract_tracelog_filename_from_beast2_input_file(input_filename)
```
extract_treelog_filename_from_beast2_input_file

Arguments

input_filename  the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

Value

the name of the trace log file

Author(s)

Richèl J.C. Bilderbeek

Examples

beast2_input_filename <- get_beastier_tempfilename()
tracelog_filename <- get_beastier_tempfilename()
beautier::create_beast2_input_file_from_model(
  input_filename = beautier::get_beautier_path("test_output_0.fas"),
  output_filename = beast2_input_filename,
  inference_model = beautier::create_inference_model(
    mcmc = beautier::create_mcmc(
      tracelog = beautier::create_tracelog(
        filename = tracelog_filename
      )
    )
  )
)

extract_tracelog_filename_from_beast2_input_file(
  input_filename = beast2_input_filename
)

file.remove(beast2_input_filename)

extract_treelog_filename_from_beast2_input_file

Internal function to extract the treelog filename for a BEAST2 input file

Description

Extract the treelog filename from a BEAST2 input file

Usage

extract_treelog_filename_from_beast2_input_file(input_filename)
get_alignment_ids_from_xml_filename

Arguments

input_filename the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

Value

the treelog filename for a BEAST2 input file

Author(s)

Richèl J.C. Bilderbeek

Examples

beast2_input_filename <- get_beastier_tempfilename()
beautier::create_beast2_input_file_from_model(
    input_filename = beautier::get_beautier_path("test_output_0.fas"),
    output_filename = beast2_input_filename
)
extract_treelog_filename_from_beast2_input_file(
    input_filename = beast2_input_filename
)
file.remove(beast2_input_filename)

get_alignment_ids_from_xml_filename

Get the alignment ID from a file with one alignment

Description

Get the alignment ID from a file with one alignment

Usage

get_alignment_ids_from_xml_filename(xml_filename)

Arguments

xml_filename name of a BEAST2 XML input filename

Value

one or more alignment IDs

Author(s)

Richèl J.C. Bilderbeek
get_beast2_example_filename

Get the full path of a BEAST2 example file

Description

Will stop if the filename is not a BEAST2 example file

Usage

get_beast2_example_filename(
    filename,
    beast2_folder = get_default_beast2_folder()
)

Arguments

filename name of the BEAST2 example file. This should exclude the full path; this function exists to add that full path
beast2_folder the folder where the BEAST2 is installed. Note that this is not the folder where the BEAST2 executable is installed: the BEAST2 executable is in a sub-folder. Use get_default_beast2_folder to get the default BEAST2 folder. Use get_default_beast2_bin_path to get the full path to the default BEAST2 executable.

Value

the full path of a BEAST2 example file, will stop if the filename is not a BEAST2 example file

Examples

if (is_beast2_installed()) {
    get_beast2_example_filename("testJukesCantor.xml")
}
**get_beast2_example_filenames**

*Get a list with the full paths of all BEAST2 example filenames*

**Description**

Get a list with the full paths of all BEAST2 example filenames

**Usage**

```r
get_beast2_example_filenames(beast2_folder = get_default_beast2_folder())
```

**Arguments**

`beast2_folder`  
The folder where the BEAST2 is installed. Note that this is not the folder where the BEAST2 executable is installed: the BEAST2 executable is in a sub-folder. Use `get_default_beast2_folder` to get the default BEAST2 folder. Use `get_default_beast2_bin_path` to get the full path to the default BEAST2 executable.

**Value**

A list with the full paths of all BEAST2 example filenames

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
if (is_beast2_installed()) {
  get_beast2_example_filenames()
}
```

---

**get_beast2_main_class_name**

*Get the BEAST2 main class name.*

**Description**

One way to fix the error *no main manifest attribute* is to specify the main class name.

**Usage**

```r
get_beast2_main_class_name()
```
get_beast2_options_filenames

Value
the BEAST2 main class name

Author(s)
Richèl J.C. Bilderbeek

Examples
get_beast2_main_class_name()

get_beast2_options_filenames
Extract the filenames from a `beast2_options`

Description
Extract the filenames from a `beast2_options`

Usage
get_beast2_options_filenames(beast2_options)

Arguments
beast2_options a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value
the filenames from a `beast2_options`

Author(s)
Richèl J.C. Bilderbeek

Examples
beast2_options <- create_beast2_options()
get_beast2_options_filenames(beast2_options)
**get_beast2_version**

*Get the BEAST2 version*

---

**Description**

Get the BEAST2 version.

**Usage**

```r
get_beast2_version(beast2_path = get_default_beast2_path())
```

**Arguments**

- `beast2_path` name of either a BEAST2 binary file (usually simply `beast`) or a BEAST2 jar file (usually has a `.jar` extension). Use `get_default_beast2_bin_path` to get the default BEAST binary file’s path. Use `get_default_beast2_jar_path` to get the default BEAST jar file’s path.

**Value**

The BEAST2 version.

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
if (is_beast2_installed() && is_on_ci()) {
  get_beast2_version()
}
```

---

**get_beastier_folder**

*Get the path to the beastier temporary files folder*

---

**Description**

Get the path to the beastier temporary files folder.

**Usage**

```r
get_beastier_folder()
```

**Value**

The path to the beastier temporary files folder.
get_beastier_path

Author(s)
Richèl J.C. Bilderbeek

Examples
get_beastier_folder()

get_beastier_path Get the full path of a file in the inst/extdata folder

Description
Get the full path of a file in the inst/extdata folder

Usage
get_beastier_path(filename)

Arguments
filename the file’s name, without the path

Value
the full path to the filename. Will stop if the file is absent in the inst/extdata folder

Author(s)
Richèl J.C. Bilderbeek

See Also
for more files, use get_beastier_paths

Examples
get_beastier_path("beast2_example_output.log")
get_beastier_path("beast2_example_output.trees")
get_beastier_path("beast2_example_output.xml")
get_beastier_path("beast2_example_output.xml.state")
**get_beastier_paths**

Get the full paths of files in the `inst/extdata` folder

**Description**

Get the full paths of files in the `inst/extdata` folder

**Usage**

```
get_beastier_paths(filenames)
```

**Arguments**

- **filenames**
  
  the files' names, without the path

**Value**

the filenames' full paths. Will stop if a file is absent in the `inst/extdata` folder

**Author(s)**

Richèl J.C. Bilderbeek

**See Also**

for one file, use `get_beastier_path`

**Examples**

```r
get_beastier_paths(
  c(
    "beast2_example_output.log",
    "beast2_example_output.trees",
    "beast2_example_output.xml",
    "beast2_example_output.xml.state"
  )
)
```
get_beastier_tempfilename

Get a temporary filename

Description

Get a temporary filename, similar to tempfile, except that it always writes to a temporary folder named beastier.

Usage

get_beastier_tempfilename(pattern = "file", fileext = "")

Arguments

pattern       a non-empty character vector giving the initial part of the name.
fileext       a non-empty character vector giving the file extension

Value

name for a temporary file

Note

this function is added to make sure no temporary cache files are left undeleted

Author(s)

Richèl J.C. Bilderbeek

Examples

get_beastier_tempfilename()
get_beastier_tempfilename(pattern = "my_pattern_")
get_beastier_tempfilename(fileext = ".ext")
**get_default_beast2_bin_path**

*Get the default BEAST2 binary file (beast, that is) path*

---

**Description**

Get the default BEAST2 binary file (beast, that is) path

**Usage**

```r
get_default_beast2_bin_path(
  beast2_folder = get_default_beast2_folder(),
  os = rappdirs::app_dir()$os
)
```

**Arguments**

- `beast2_folder`  
  the folder where the BEAST2 is installed. Note that this is not the folder where the BEAST2 executable is installed: the BEAST2 executable is in a sub-folder. Use `get_default_beast2_folder` to get the default BEAST2 folder. Use `get_default_beast2_bin_path` to get the full path to the default BEAST2 executable.

- `os`  
  name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)

**Value**

the default BEAST2 binary file’s path

**Author(s)**

Richèl J.C. Bilderbeek

**See Also**

Use `get_default_beast2_folder` to get the default folder in which BEAST2 is installed. Use `install_beast2` with default arguments to install BEAST2 to this location.

**Examples**

```r
if (is_beast2_installed()) {
  get_default_beast2_bin_path()
}
```
get_default_beast2_download_url

Get the default BEAST2 download URL, which depends on the operating system

Description
Get the default BEAST2 download URL, which depends on the operating system.

Usage
get_default_beast2_download_url(
  beast2_version = beastier::get_default_beast2_version(),
  os = rappdirs::app_dir()$os
)

Arguments
- **beast2_version**: the version of BEAST2. By default, this is the version as returned by `get_default_beast2_version`.
- **os**: name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows).

Value
the URL where BEAST2 can be downloaded from.

Author(s)
Richèl J.C. Bilderbeek

Examples
get_default_beast2_download_url()

---

get_default_beast2_download_url_linux

Get the BEAST2 download URL for Linux

Description
Get the BEAST2 download URL for Linux.

Usage
get_default_beast2_download_url_linux(
  beast2_version = beastier::get_default_beast2_version()
)
Arguments

beast2_version  the version of BEAST2. By default, this is the version as returned by get_default_beast2_version

Value

the URL where BEAST2 can be downloaded from

Author(s)

Richèl J.C. Bilderbeek

get_default_beast2_download_url_win

Get the BEAST2 download URL for Windows

Description

Get the BEAST2 download URL for Windows

Usage

get_default_beast2_download_url_win(
    beast2_version = beastier::get_default_beast2_version()
)

Arguments

beast2_version  the version of BEAST2. By default, this is the version as returned by get_default_beast2_version

Value

the URL where BEAST2 can be downloaded from

Author(s)

Richèl J.C. Bilderbeek
get_default_beast2_folder

Get the path to the folder where this package installs BEAST2 by default

Description
Get the path to the folder where this package installs BEAST2 by default

Usage
get_default_beast2_folder()

Value
the path to the folder where this package installs BEAST2 by default

Author(s)
Richèl J.C. Bilderbeek

See Also
Use get_default_beast2_jar_path to get the path to the BEAST2 jar file, when installed by this package. Use install_beast2 with default arguments to install BEAST2 to this folder.

Examples
message(get_default_beast2_folder())

get_default_beast2_jar_path

Get the default BEAST2 jar file’s path

Description
Get the default BEAST2 jar file’s path

Usage
get_default_beast2_jar_path(
  beast2_folder = beastier::get_default_beast2_folder(),
  os = rappdirs::app_dir()$os
)

Arguments

beast2_folder  the folder where the BEAST2 is installed. Note that this is not the folder where the BEAST2 executable is installed: the BEAST2 executable is in a subfolder. Use `get_default_beast2_folder` to get the default BEAST2 folder. Use `get_default_beast2_bin_path` to get the full path to the default BEAST2 executable.

os  name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)

Value

the default BEAST2 jar file’s path

Author(s)

Richèl J.C. Bilderbeek

See Also

Use `get_default_beast2_folder` to get the default folder in which BEAST2 is installed. Use `install_beast2` with default arguments to install BEAST2 to this location.

Examples

get_default_beast2_jar_path()

get_default_beast2_path

Get the default BEAST2 path

Description

Get the default BEAST2 path

Usage

get_default_beast2_path(
  beast2_folder = beastier::get_default_beast2_folder(),
  os = rappdirs::app_dir()$os
)

Arguments

beast2_folder  the folder where the BEAST2 is installed. Note that this is not the folder where the BEAST2 executable is installed: the BEAST2 executable is in a subfolder. Use `get_default_beast2_folder` to get the default BEAST2 folder. Use `get_default_beast2_bin_path` to get the full path to the default BEAST2 executable.

os  name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)
get_default_beast2_version

Value
the default BEAST2 path

Author(s)
Richèl J.C. Bilderbeek

See Also
Use get_default_beast2_bin_path to get the default path to the BEAST2 binary file. Use get_default_beast2_jar_path to get the default path to the BEAST2 jar file. Use get_default_beast2_folder to get the default folder in which BEAST2 is installed. Use install_beast2 with default arguments to install BEAST2 to this location.

Examples
if (is_beast2_installed()) {
    get_default_beast2_path()
}

get_default_beast2_version

Get the default BEAST2 version that is used by beastier

Description
Get the default BEAST2 version that is used by beastier

Usage
get_default_beast2_version()

Value
the BEAST2 version

Author(s)
Richèl J.C. Bilderbeek

Examples
get_default_beast2_version()
**get_default_java_path**  
*Obtains the default path to the Java executable*

**Description**

Obtains the default path to the Java executable

**Usage**

```r
get_default_java_path(os = rappdirs::app_dir()$os)
```

**Arguments**

- `os`  
  name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)

**Value**

the default path to the Java executable

**Author(s)**

Richèl J.C. Bilderbeek

---

**get_duplicate_param_ids**  
*Find duplicate RealParameter IDs*

**Description**

Find duplicate RealParameter IDs

**Usage**

```r
get_duplicate_param_ids(text)
```

**Arguments**

- `text`  
  the XML as text

**Value**

a vector of duplicate IDs, will be empty if all IDs are unique

**Author(s)**

Richèl J.C. Bilderbeek
See Also

to see if all IDs are unique, use has_unique_ids

Examples

```r
line_1 <- "<parameter id="RealParameter.1" ...</parameter>
line_2 <- "<parameter id="RealParameter.2" ...</parameter>

testit::assert(
  length(get_duplicate_param_ids(c(line_1, line_2))) == 0)

get_duplicate_param_ids(c(line_1, line_1)) == c("RealParameter.1")

testit::assert(
  get_duplicate_param_ids(
    c(line_2, line_2)) == c("RealParameter.2")
)
```

---

get_java_version  Get the Java version  

Description

Get the Java version

Usage

```r
get_java_version()
```

Value

the Java version

Author(s)

Richèl J.C. Bilderbeek

Examples

```r
if (is_beast2_installed() && is_on_ci()) {
  get_java_version()
}
```
get_trees_filenames

Get the .trees filenames that BEAST2 will produce

Description
Get the .trees filenames that BEAST2 will produce

Usage
get_trees_filenames(input_filename)

Arguments
input_filename  the name of a BEAST2 input XML file. This file usually has an .xml extension. Use create_temp_input_filename to create a temporary filename with that extension.

Value
character vector with the names of the .trees files that BEAST2 will produce

Author(s)
Richèl J.C. Bilderbeek

Examples
get_trees_filenames(get_beastier_path("2_4.xml"))
get_trees_filenames(get_beastier_path("anthus_2_4.xml"))

gives_beast2_warning
Determines if BEAST2 issues a warning when using the BEAST2 XML input file

Description
Determines if BEAST2 issues a warning when using the BEAST2 XML input file

Usage
gives_beast2_warning(
  filename,
  verbose = FALSE,
  beast2_path = beastier::get_default_beast2_path()
)
has_unique_ids

Determine if the XML text has unique parameter IDs

Description
Determine if the XML text has unique parameter IDs

Usage
has_unique_ids(text)
install_beast2

Arguments

text the XML as text

Value

TRUE if all parameter IDs are unique, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also
to obtain the duplicate parameter IDs, use get_duplicate_param_ids

Examples

```
line_1 <- "<parameter id="RealParameter.1" ...</parameter>"
line_2 <- "<parameter id="RealParameter.2" ...</parameter>"
# Unique IDs
has_unique_ids(c(line_1, line_2))
# No unique ID
has_unique_ids(c(line_1, line_1))
```

install_beast2  Deprecated function to install BEAST2

Description

This function is deprecated as it violated CRAN policy.

Usage

```
install_beast2(
  folder_name = rappdirs::user_data_dir(),
  beast2_version = beastier::get_default_beast2_version(),
  verbose = FALSE,
  os = rappdirs::app_dir()$os
)
```

Arguments

- `folder_name`: name of the folder where the BEAST2 files will be put. The name of the BEAST2 binary file will be at `[folder_name]/beast/bin/beast` The name of the BEAST2 jar file will be at `[folder_name]/beast/lib/launcher.jar`
- `beast2_version`: the version of BEAST2. By default, this is the version as returned by `get_default_beast2_version`
- `verbose`: if TRUE, additional information is displayed, that is potentially useful in debugging
- `os`: name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)
**is_alignment**

Determines if the input is an alignment of type **DNAbin**

**Description**

Determines if the input is an alignment of type **DNAbin**

**Usage**

```
is_alignment(input)
```

**Arguments**

- **input**
  
The input to be tested

**Value**

TRUE or FALSE

**Author(s)**

Richèl J.C. Bilderbeek

---

**is_beast2_input_file**

Is a file a valid BEAST2 input file?

**Description**

Is a file a valid BEAST2 input file?

**Usage**

```
is_beast2_input_file(
    filename,
    show_warnings = FALSE,
    verbose = FALSE,
    beast2_path = get_default_beast2_path()
)
```
is_beast2_input_file

Arguments

- **filename**: name of the BEAST2 XML input file
- **show_warnings**: if TRUE, warnings will shown
- **verbose**: if TRUE, additional information is displayed, that is potentially useful in debugging
- **beast2_path**: name of either a BEAST2 binary file (usually simply `beast`) or a BEAST2 jar file (usually has a `.jar` extension). Use `get_default_beast2_bin_path` to get the default BEAST binary file's path Use `get_default_beast2_jar_path` to get the default BEAST jar file's path

Value

TRUE if the file is valid, FALSE if not

Note

this function only works on standard BEAST2 input files: if a BEAST2 input file is modified to use a certain BEAST2 package, this function will label it as an invalid file

Author(s)

Richèl J.C. Bilderbeek

See Also

Use `are_beast2_input_lines` to check the lines

Examples

```r
if (is_beast2_installed() && is_on_ci()) {

  filename <- get_beastier_path("anthus_2_4.xml")
  # TRUE, this is a BEAST2 input file
  is_beast2_input_file(filename)

  filename <- get_beastier_path("beast2_example_output.log")
  # FALSE, this is not a BEAST2 input file,
  # it is a BEAST2 output log file instead
  is_beast2_input_file(filename)
}
```
is_beast2_installed  Checks if BEAST2 is installed

Description
Checks if BEAST2 is installed

Usage

```r
is_beast2_installed(
    folder_name = get_default_beast2_folder(),
    os = rappdirs::app_dir()$os
)
```

Arguments

- `folder_name` name of the folder where the BEAST2 files are put. The name of the BEAST2 binary file will be at `folder_name/beast/bin/beast`
- `os` name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)

Value
TRUE if BEAST2 is installed

Author(s)
Richèl J.C. Bilderbeek

Examples

```r
is_beast2_installed()
```

is_bin_path  Is the path a path to the BEAST2 binary file? Does not check if the file at that path is present

Description
Is the path a path to the BEAST2 binary file? Does not check if the file at that path is present

Usage

```r
is_bin_path(path)
```
is_jar_path

Arguments

path a string to a path

Value

TRUE if the path is a path to a BEAST2 binary file

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed()) {
    # TRUE
    is_bin_path("beast")
    is_bin_path("BEAST.exe")
    is_bin_path(get_default_beast2_bin_path())
    # FALSE
    is_bin_path("launcher.jar")
    is_bin_path(get_default_beast2_jar_path())
}

Description

Is the path a path to the BEAST2 jar file? Does not check if the file at that path is present

Usage

is_jar_path(path)

Arguments

path a string to a path

Value

TRUE if the path is a path to a BEAST2 jar file

Author(s)

Richèl J.C. Bilderbeek
Examples

# Returns TRUE
is_jar_path("beast.jar")
is_jar_path("launcher.jar")
is_jar_path(get_default_beast2_jar_path())

# Returns FALSE
is_jar_path("beast")
is_jar_path(get_default_beast2_bin_path())

---

is_on_appveyor  
_Determines if the environment is AppVeyor_

Description

Determines if the environment is AppVeyor

Usage

is_on_appveyor()

Value

TRUE if run on AppVeyor, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_on_appveyor()) {
    message("Running on AppVeyor")
}

---

is_on_ci  
_Determines if the environment is a continuous integration service_

Description

Determines if the environment is a continuous integration service

Usage

is_on_ci()
Value

TRUE if run on AppVeyor or Travis CI, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```r
if (is_on_ci()) {
    message("Running on a continuous integration service")
}
```

---

**is_on_travis**  
Determines if the environment is Travis CI

Description

Determines if the environment is Travis CI

Usage

```r
is_on_travis()
```

Value

TRUE if run on Travis CI, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```r
if (is_on_ci()) {
    message("Running on Travis CI")
}
```
is_win_bin_path  

**Description**

Is the path a path to the BEAST2 binary file? Does not check if the file at that path is present

**Usage**

is_win_bin_path(path)

**Arguments**

path  a string to a path

**Value**

TRUE if the path is a path to a BEAST2 binary file

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
# TRUE
is_win_bin_path("BEAST.exe")
# FALSE
is_win_bin_path("beast")
is_win_bin_path("launcher.jar")
```

print_beast2_options  

**Description**

Pretty-print a ‘beast2_options’

**Usage**

print_beast2_options(beast2_options)

**Arguments**

beast2_options  a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by `create_beast2_options`
Value

Nothing. Will display the ‘beast2_options’ using cat.

Author(s)

Richèl J.C. Bilderbeek

Examples

```r
print_beast2_options(create_beast2_options())
```

---

**remove_file_if_present**

*Remove a file if it is present, will do nothing if it is not.*

Description

Remove a file if it is present, will do nothing if it is not.

Usage

```r
remove_file_if_present(filename)
```

Arguments

- `filename` name of a file

Value

Nothing. Will remove the file if it is presented, will do nothing if it is not.

Author(s)

Richèl J.C. Bilderbeek

Examples

```r
filename <- tempfile()
file.create(filename)
remove_file_if_present(filename)
remove_file_if_present(filename)
```
rename_beast2_options_filenames

Rename the filenames in the BEAST2 options

Description

Rename the filenames in the BEAST2 options

Usage

rename_beast2_options_filenames(beast2_options, rename_fun)

Arguments

beast2_options  a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options
rename_fun a function to rename a filename, as can be checked by check_rename_fun. This function should have one argument, which will be a filename or NA. The function should return one filename (when passed one filename) or one NA (when passed one NA). Example rename functions are:
  • get_remove_dir_fun get a function that removes the directory paths from the filenames, in effect turning these into local files
  • get_replace_dir_fun get a function that replaces the directory paths from the filenames
  • get_remove_hex_fun get a function that removes the hex string from filenames. For example, tracelog_82c1a522040.log becomes tracelog.log

Value

a ‘beast2_options’ with the filenames it contains renamed

Author(s)

Richèl J.C. Bilderbeek

Examples

# beast2_options with local filenames
beast2_options <- create_beast2_options(
  input_filename = "my.fas",
  output_state_filename = "my_state.xml.state"
)

# Rename filenames to be in /my/new/folder
rename_beast2_options_filenames(
  beast2_options = beast2_options,
  rename_fun = beautier::get_replace_dir_fun("/my/new/folder")
)
Description

Run BEAST2

Usage

```r
run_beast2(
  input_filename,
  output_log_filename = "output_log_filename_is_deprecated",
  output_trees_filenames = "output_trees_filenames_is_deprecated",
  output_state_filename = create_temp_state_filename(),
  rng_seed = NA,
  n_threads = NA,
  use_beagle = FALSE,
  overwrite = TRUE,
  beast2_working_dir = "beast2_working_dir_is_deprecated",
  beast2_path = get_default_beast2_path(),
  verbose = FALSE
)
```

Arguments

- `input_filename` : the name of a BEAST2 input XML file. This file usually has an .xml extension. Use `create_temp_input_filename` to create a temporary filename with that extension.
- `output_log_filename` : name of the .log file to create
- `output_trees_filenames` : one or more names for .trees file to create. There will be one .trees file created per alignment in the input file. The number of alignments must equal the number of .trees filenames, else an error is thrown. Alignments are sorted alphabetically by their IDs
- `output_state_filename` : name of the .xml.state file to create. Use `create_temp_state_filename` to create a temporary filename with that extension.
- `rng_seed` : the random number generator seed of the BEAST2 run. Must be a non-zero positive integer value or NA. If `rng_seed` is NA, BEAST2 will pick a random seed
- `n_threads` : the number of computational threads to use. Use NA to use the BEAST2 default of 1.
- `use_beagle` : use BEAGLE if present
- `overwrite` : if TRUE: overwrite the .log and .trees files if one of these exists. If FALSE, BEAST2 will not be started if
run_beast2_from_options

Run BEAST2

Description

Run BEAST2

Value

The text sent to STDOUT and STDERR. It will create the file with name output_state_filenames

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed() && is_on_ci()) {

    output_state_filename <- create_temp_state_filename()

    run_beast2(
        input_filename = get_beastier_path("2_4.xml"),
        output_state_filename = output_state_filename
    )
    file.remove(output_state_filename)
}

beast2_working_dir

a folder where BEAST2 can work in isolation. For each BEAST2 run, a new subfolder is created in that folder. Within this folder, BEAST2 is allowed to create all of its output files, without the risk of overwriting existing ones, allowing BEAST2 to run in multiple parallel processes.

beast2_path

name of either a BEAST2 binary file (usually simply beast) or a BEAST2 jar file (usually has a .jar extension). Use get_default_beast2_bin_path to get the default BEAST binary file’s path Use get_default_beast2_jar_path to get the default BEAST jar file’s path

verbose

if TRUE, additional information is displayed, that is potentially useful in debugging

• the .log file exists
• the .trees files exist
• the .log file created by BEAST2 exists
• the .trees files created by BEAST2 exist
Usage

run_beast2_from_options(beast2_options = create_beast2_options())

Arguments

beast2_options  a set of BEAST2 options, that are the R equivalent of the BEAST2 command-line options, as can be created by create_beast2_options

Value

The text sent to STDOUT and STDERR. It will create the file with name output_state_filenames

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_beast2_installed() && is_on_ci()) {
    beast2_options <- create_beast2_options(
        input_filename = get_beastier_path("2_4.xml")
    )
    run_beast2_from_options(beast2_options)
    file.remove(beast2_options$output_state_filename)
}

save_lines

Save text (a container of strings) to a file

Description

Save text (a container of strings) to a file

Usage

save_lines(filename, lines)

Arguments

filename  filename of the file to have the text written to
lines  lines of text to be written to file

Value

Nothing. Will save the lines to file

Author(s)

Richèl J.C. Bilderbeek
Examples

```r
text <- c("hello", "world")
filename <- get_beastier_tempfilename()
save_lines(filename = filename, lines = text)
file.remove(filename)
```

---

**save_nexus_as_fasta**  
*Save a NEXUS file as a FASTA file*

**Description**

Save a NEXUS file as a FASTA file

**Usage**

```r
save_nexus_as_fasta(nexus_filename, fasta_filename)
```

**Arguments**

- `nexus_filename`: name of an existing NEXUS file
- `fasta_filename`: name of the FASTA file to be created

**Value**

nothing. The NEXUS file will be saved as a FASTA file

---

**uninstall_beast2**  
*Deprecated function to uninstall BEAST2*

**Description**

Deprecated function to uninstall BEAST2

**Usage**

```r
uninstall_beast2(
  folder_name = rappdirs::user_data_dir(),
  os = rappdirs::app_dir()$os,
  verbose = FALSE
)
```
Arguments

folder_name  name of the folder where the BEAST2 files are installed. The name of the BEAST2 binary file will be at `[folder_name]/beast/bin/beast` The name of the BEAST2 jar file will be at `[folder_name]/beast/lib/launcher.jar`

os  name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)

verbose  if TRUE, additional information is displayed, that is potentially useful in debugging

Value

Nothing. A deprecation message using `stop` will be triggered

Author(s)

Richèl J.C. Bilderbeek

---

`upgrade_beast2`  
*Deprecated function to upgrade BEAST2.*

Description

Deprecated function to upgrade BEAST2.

Usage

```r
upgrade_beast2(
  folder_name = rappdirs::user_data_dir(),
  os = rappdirs::app_dir()$os
)
```

Arguments

folder_name  name of the folder where the BEAST2 files will be put. The name of the BEAST2 binary file will be at `[folder_name]/beast/bin/beast` The name of the BEAST2 jar file will be at `[folder_name]/beast/lib/launcher.jar`

os  name of the operating system, must be `unix` (Linux, Mac) or `win` (Windows)

Value

Nothing. A deprecation message using `stop` will be triggered

Author(s)

Richèl J.C. Bilderbeek
Index

add_quotes_if_has_spaces, 4
are_beast2_input_lines, 5, 66, 69
are_beast2_input_lines_deep, 6
are_beast2_input_lines_fast, 7
are_identical_alignments, 8
beast2_options_to_table, 8
beastier, 9, 9, 19, 20, 37, 44, 53, 56
beastier_report, 9
beautier, 37
cat, 75
check_beast2, 10
check_beast2_options, 11, 13–15, 27, 39
check_beast2_options_data_types, 12
check_beast2_options_do_not_overwriteExistingFiles, 13
check_beast2_options_filenames_differ, 14
check_beast2_options_names, 14
check_beast2_optionses, 12
check_beast2_path, 15
check_can_create_dir_for_state_output_file, 16
check_can_create_file, 16
check_can_create_screenlog_file, 17
check_can_create_state_output_file, 18
check_can_create_tracelog_file, 18
check_empty_beastier_folder, 19
check_input_filename, 20
check_input_filename_validity, 21
check_n_threads, 22
check_os, 22
check_rename_fun, 45, 76
check_rng_seed, 23
continue_beast2, 24
create_beast2_continue_cmd_from_options, 24
create_beast2_input_file_folder, 25
create_beast2_input_lines, 5, 66, 69
create_beast2_input_lines_deep, 6
create_beast2_input_lines_fast, 7
create_beast2_options, 8, 11–19, 21, 24, 25, 26, 29–31, 39, 44, 52, 74, 76, 79
create_beast2_run_cmd, 27
create_beast2_run_cmd_from_options, 29
create_beast2_screenlog_folder, 30
create_beast2_state_output_file_folder, 30
create_beast2_tracelog_folder, 31
create_beast2_treelog_folder, 31
create_beast2_validate_cmd, 32
create_beast2_validate_cmd_bin, 33
create_beast2_validate_cmd_jar, 34
create_beast2_version_cmd, 35
create_beast2_version_cmd_bin, 35
create_beast2_version_cmd_jar, 36
create_beastier_tempfolder, 37
create_beautier_tempfolder, 37
create_mcbette_beast2_options, 38
create_random_alignment, 39
create_random_fasta, 40
create_random_phylogeny, 41
create_temp_input_filename, 20, 21, 26, 28, 32–34, 38, 42, 45, 47–49, 65, 77
create_temp_state_filename, 26, 38, 42, 45, 77
default_params_doc, 43
DNAbin, 40, 68
do_minimal_run, 46
extract_screenlog_filename_from_beast2_input_file, 47
extract_tracelog_filename_from_beast2_input_file, 47
extract_treelog_filename_from_beast2_input_file, 48
get_alignment_ids_from_xml_filename, 49
get_beast2_example_filename, 50
INDEX

get_beast2_example_filenames, 51
get_beast2_main_class_name, 51
get_beast2_options_filenames, 52
get_beast2_version, 53
get_beastier_folder, 53
get_beastier_path, 54, 55
get_beastier_paths, 54, 55
get_beastier_tempfilename, 56
get_default_beast2_bin_path, 5, 6, 10, 15, 21, 27, 28, 32, 33, 35, 36, 39, 44, 50, 51, 53, 57, 61, 62, 66, 69, 78
get_default_beast2_download_url, 58
get_default_beast2_download_url_linux, 58
get_default_beast2_download_url_win, 59
get_default_beast2_folder, 10, 44, 50, 51, 57, 60, 61, 62
get_default_beast2_jar_path, 5, 6, 10, 15, 21, 27, 28, 32, 34–36, 44, 53, 60, 62, 66, 69, 78
get_default_beast2_path, 61
get_default_beast2_version, 44, 58, 59, 62, 67
get_default_java_path, 63
get_duplicate_param_ids, 63, 67
get_java_version, 64
get_remove_dir_fun, 45, 76
get_remove_hex_fun, 46, 76
get_replace_dir_fun, 45, 76
get_trees_filenames, 65
gives_beast2_warning, 65

has_unique_ids, 64, 66

install_beast2, 57, 60–62, 67
is_alignment, 68
is_beast2_input_file, 5–7, 66, 68
is_beast2_installed, 70
is_bin_path, 70
is_jar_path, 71
is_on_appveyor, 72
is_on_ci, 72
is_on_travis, 73
is_win_bin_path, 74

list, 27, 39

message, 9, 10

print_beast2_options, 74
remove_file_if_present, 75
rename_beast2_options_filenames, 76
return, 45, 76
run_beast2, 77
run_beast2_from_options, 46, 78
save_lines, 79
save_nexus_as_fasta, 80
stop, 10, 13, 15–23, 50, 68, 81
tempfile, 56
tibble, 8

uninstall_beast2, 80
upgrade_beast2, 81

NA, 22, 23, 26, 28, 38, 45, 46, 76, 77

run_beast2, 77
run_beast2_from_options, 46, 78

save_lines, 79
save_nexus_as_fasta, 80
stop, 10, 13, 15–23, 50, 68, 81