Package ‘beastier’

June 5, 2021

Type Package
Title Call 'BEAST2'
Version 2.4.2
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 (>= 2.5.2), devtools, phangorn, rappdirs, readr, rJava, stringr, 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  2021-06-05 16:40:02 UTC

**R topics documented:**

- add_quotes_if_has_spaces .......................... 4
- are_beast2_input_lines ............................ 4
- are_beast2_input_lines_deep ...................... 5
- are_beast2_input_lines_fast ...................... 6
- are_identical_alignments .......................... 7
- beast2_options_to_table ........................... 8
- beastier ........................................... 8
- beastier_report ..................................... 9
- check_beast2 ....................................... 9
- check_beast2_options ............................... 10
- check_beast2_optionses ............................. 10
- check_beast2_options_data_types ................ 11
- check_beast2_options_do_not_overwrite_existing_files 12
- check_beast2_options_filenames_differ .......... 12
- check_beast2_options_names ...................... 13
- check_beast2_path ................................ 13
- check_can_create_dir_for_state_output_file ..... 14
- check_can_create_file ............................. 15
- check_can_create_screenlog_file ................ 15
- check_can_create_state_output_file ............ 16
- check_can_create_tracelog_file .................. 16
- check_can_create_treelog_file ................... 17
- check_input_filename ................................ 17
- check_input_filename_validity .................. 18
- check_n_threads .................................. 19
- check_os .......................................... 19
- check_rng_seed .................................... 20
- continue_beast2 .................................. 21
- create_beast2_continue_cmd_from_options ...... 21
- create_beast2_options ............................. 22
- create_beast2_run_cmd ............................. 24
- create_beast2_run_cmd_from_options ............ 25
- create_beast2_validate_cmd ...................... 26
- create_beast2_validate_cmd_bin .................. 27
- create_beast2_validate_cmd_jar .................. 28
- create_beast2_version_cmd ....................... 29
- create_beast2_version_cmd_bin ................... 29
- create_beast2_version_cmd_jar ................... 30
- create_mcbette_beast2_options .................. 31
R topics documented:

create_random_alignment ........................................ 32
create_random_fasta .............................................. 33
create_random_phylogeny ......................................... 34
create_temp_input_filename ..................................... 35
create_temp_state_filename ..................................... 35
default_params_doc ............................................. 35
do_minimal_run .................................................. 39
get_alignment_ids_from_xml_filename ......................... 40
get_beast2_example_filename .................................... 40
get_beast2_example_filenames .................................. 41
get_beast2_main_class_name .................................... 42
get_beast2_options_filenames ................................... 42
get_beast2_version ............................................. 43
get_beastier_path .............................................. 43
get_beastier_paths ............................................. 44
get_beastier_tempfilename ..................................... 45
get_default_beast2_bin_path .................................... 46
get_default_beast2_download_url .............................. 47
get_default_beast2_download_url_linux ....................... 47
get_default_beast2_download_url_win ......................... 48
get_default_beast2_folder ..................................... 49
get_default_beast2_version .................................... 49
get_default_beast2_jar_path ................................... 49
get_default_beast2_path ....................................... 50
get_default_beast2_version .................................... 51
get_default_java_path ......................................... 52
get_duplicate_param_ids ...................................... 52
get_java_version ............................................... 53
get_trees_filenames ............................................ 54
gives_beast2_warning ......................................... 54
has_unique_ids ................................................. 55
install_beast2 .................................................. 56
is_alignment ..................................................... 57
is_beast2_input_file .......................................... 58
is_beast2_installed ........................................... 59
is_bin_path ..................................................... 60
is_jar_path ..................................................... 60
is_on_appveyor .................................................. 61
is_on_ci ......................................................... 62
is_on_travis ..................................................... 62
is_win_bin_path ................................................ 62
print_beast2_options .......................................... 63
remove_file_if_present ....................................... 64
rename_beast2_options_filenames ............................. 64
run_beast2 ..................................................... 65
run_beast2_from_options ...................................... 66
save_lines ....................................................... 67
save_nexus_as_fasta .......................................... 68
uninstall_beast2 ................................................ 68
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_spaces("a b")
are_beast2_input_lines

Usage

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

  if (is_beast2_installed() && is_on_ci()) {
    are_beast2_input_lines(get_beastier_path("anthus_2_4.xml"))
  }

Description

  Would these lines of text, when written to a file, result in a valid BEAST2 input file?
are_beast2_input_lines_fast

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?
are_identical_alignments

Usage

are_beast2_input_lines_fast(lines)

Arguments

lines lines of text

Value

TRUE if the text is valid, FALSE if not

Author(s)

Richél 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

are_identical_alignments(p, q)

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
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

beastier beastier: A package to call BEAST2.

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

**Create a beastier report, to be used when reporting bugs**

**Description**
Create a `beastier` report, to be used when reporting bugs

**Usage**

```
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

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```
if (is_beast2_installed()) {
  check_beast2()
}
```
check_beast2_options

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

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_optioneses

Description

Calls stop if the BEAST2 option object is invalid

Usage

check_beast2_optioneses(beast2_optioneses)
check_beast2_options_data_types

Arguments
beast2_options
    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

Author(s)
Richèl J.C. Bilderbeek

See Also
Use create_beast2_options to create a valid BEAST2 options object

Examples
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
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
check_beast2_options_do_not_overwrite_existing_files

*Check if the beast2_options will not overwrite existing files, when the ‘overwrite’ options is set to FALSE*

---

**Description**

Will stop if a file is threatened to be overwritten

**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`

**Author(s)**

Richèl J.C. Bilderbeek

---

check_beast2_options_filenames_differ

*Check if the filenames in beast2_options differ*

---

**Description**

Calls stop if not.

**Usage**

```
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
check_beast2_options_names

See Also
Use check_beast2_options to check the entire beast2_options object

---

check_beast2_options_names
Check if the beast2_options, which is a list, has all the elements needed.

---

Description
Calls stop if not.

Usage
check_beast2_options_names(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
Checks the BEAST2.jar path. Will stop if there is a problem with the BEAST2.jar path.

---

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

Usage
check_beast2_path(beast2_path)
check_can_create_dir_for_state_output_file

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)
}

Description

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

Usage

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

**Description**

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.

**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

**Author(s)**

Richèl J.C. Bilderbeek

---

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

**Description**

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

**Usage**

```r
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`
check_can_create_state_output_file

Description

Check if the state output file can be created. Will stop otherwise.

Usage

check_can_create_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

check_can_create_tracelog_file

Description

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

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
check_can_create_treelog_file

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

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

check_input_filename

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(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 call 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.

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)

Arguments

os  

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

Value
nothing. Will stop if the OS is unsupported

Author(s)
Richèl J.C. Bilderbeek

Examples

check_os("mac")
check_os("unix")
check_os("win")

check_rng_seed
Check if the input is a valid RNG seed.

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

Author(s)
Richèl J.C. Bilderbeek

Examples

# 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`

**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

**Usage**

```r
create_beast2_continue_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可以 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_continue_cmd_from_options(
        beast2_options = create_beast2_options()
    )
}

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

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"
)
create_beast2_options

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
• 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

Author(s)

Richèl J.C. Bilderbeek
create_beast2_run_cmd

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
create_beast2_run_cmd_from_options

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

Examples

```r
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()
    )
}
```

create_beast2_run_cmd_from_options

\textit{Creates the terminal command to run BEAST2 from a beast2_options}

Description

Creates the terminal command to run BEAST2 from a beast2_options

Usage

\texttt{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 \texttt{create_beast2_options}

Value

da character vector with the command and arguments to call BEAST2
create_beast2_validate_cmd

**Description**

Creates the terminal command to validate a BEAST2 input file

**Usage**

```r
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
create_beast2_validate_cmd_bin

Examples

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

create_beast2_validate_cmd_bin

*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

```r
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

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

*Description*

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

*Usage*

```r
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*

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

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

**Description**

Creates the terminal command to version a BEAST2 input file

**Usage**

```r
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() && is_on Ci()) {
  create_beast2_version_cmd()
}
```

create_beast2_version_cmd_bin

*Create 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**

```r
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**

```r
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**

```r
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
Examples

```r
if (is_beast2_installed()) {
  create_beast2_version_cmd_jar()
}
```

---

`create_mcbette_beast2_options`

Create a `beast2_options` structure for `mcbette`

---

**Description**

Create a `beast2_options` structure to be used for `mcbette` (a package that allows one to do model comparison). The generated filenames indicating `mcbette` usage, as well as the correct BEAST2 binary 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
  - the `.log` file exists
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: The number of taxa
- sequence_length: The number of base pairs the alignment will have
- rate: mutation rate
- taxa_name_ext: the extension of the taxa names

Value
an alignment of class DNAbin

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_fasta**

Author(s)
Richèl J.C. Bilderbeek

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

Description
Create a random FASTA file

Usage
```r
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

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()

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()

default_params_doc

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_optionses,
    beast2_path,
    beast2_version,
    beast2_working_dir,
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.

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 beastier MCMC

misc_options  one beastier 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
• 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
**do_minimal_run**  

**Do a minimal BEAST2 run**

---

**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

---

**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**

```r
if (is_beast2_installed() && is_on_ci()) {
  do_minimal_run()
}
```
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**

```r
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

**Examples**

```r
# test_output_0
get_alignment_ids_from_xml_filename(get_beastier_path("2_4.xml"))
# c("anthus_aco","anthus_nd2")
get_alignment_ids_from_xml_filename(get_beastier_path("anthus_15_15.xml"))
```

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**

```r
get_beast2_example_filename(
    filename, 
    beast2_folder = get_default_beast2_folder() 
)
```
**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
gc.get_beast2_example_filenames(beast2_folder = gc.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 subfolder. Use `gc.get_default_beast2_folder` to get the default BEAST2 folder. Use `gc.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

**Examples**

```r
if (gc.is_beast2_installed()) {
    gc.get_beast2_example_filename("testJukesCantor.xml")
}
```
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

*Extract the filenames from a beast2_options*

**Description**

Extract the filenames from a beast2_options

**Usage**

```r
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`

**Author(s)**

Richël J.C. Bilderbeek

**Examples**

```r
beast2_options <- beastier::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.

**Author(s)**
Richèl J.C. Bilderbeek

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

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**
```r
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

```r
get_beastier_path("beast2_example_output.log")
gem_beastier_path("beast2_example_output.trees")
gem_beastier_path("beast2_example_output.xml")
gem_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

```r
gem_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`
**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**

```r
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
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
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
if (is_beast2_installed() && rappdirs::app_dir()$os == "unix") {
    testit::assert(
        grepl("beast/bin/beast",
            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

```r
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

```r
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

```r
get_default_beast2_download_url_linux(
  beast2_version = beastier::get_default_beast2_version()
)
```
**get_default_beast2_download_url_win**

**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**

```cpp
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
)
get_default_beast2_path

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 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_version

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

```java
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

```java
get_default_beast2_version()
```

Value

the BEAST2 version

Author(s)

Richèl J.C. Bilderbeek

Examples

```java
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**

```
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**

```
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
get_java_version

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)
testit::assert(
  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()
```

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

Arguments

filename name of the BEAST2 XML input file
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 produces a BEAST2 warning, FALSE if not

Author(s)

Richèl J.C. Bilderbeek

See Also

Use is_beast2_input_file to check if a file is a valid BEAST2 input file. Use are_beast2_input_lines to check if the text (for example, as loaded from a file) to be valid BEAST2 input.

Examples

if (is_beast2_installed() &&
is_on_ci() &&
rappdirs::app_dir()$os == "unix") {

  # This file is OK for BEAST2, no warning, returns FALSE
  gives_beast2_warning(filename = get_beastier_path("2_4.xml"))

  # BEAST2 will give a warning on this file, returns TRUE
  gives_beast2_warning(
    filename = get_beastier_path("beast2_warning.xml")
  )
}

has_unique_ids filename text

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

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)

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))
is_alignment

Value

Nothing. Will install BEAST2

Author(s)

Richèl J.C. Bilderbeek

Examples

```r
## Not run:
install_beast2()

## End(Not run)
```

---

**is_alignment**

_Determines if the input is an alignment of type DNAbin_

Description

Determines if the input is an alignment of type DNAbin

Usage

```r
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()  
)

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
is_beast2_installed

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` 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

TRUE if BEAST2 is installed

Author(s)

Richèl J.C. Bilderbeek

Examples

```r
if (is_beast2_installed()) {
  message("BEAST2 is 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

is_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

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())
}

is_jar_path

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

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)
is_on_appveyor

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())

---

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

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

is_on_travis()

Value

TRUE if run on Travis CI, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

if (is_on_travis()) {
    message("Running on Travis CI")
}
is_win_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
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
# TRUE
is_win_bin_path("BEAST.exe")
# FALSE
is_win_bin_path("beast")
is_win_bin_path("launcher.jar")

print_beast2_options
Pretty-print a 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
remove_file_if_present

*Remove a file if it is present*

**Description**
Remove a file if it is present

**Usage**
remove_file_if_present(filename)

**Arguments**
- filename: name of a file

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
run_beast2

run_beast2

Run BEAST2

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

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

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

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)
}

Description

Run BEAST2
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

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

---

**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
upgrade_beast2

Author(s)
Richèl J.C. Bilderbeek

Description
Deprecated function to upgrade BEAST2.

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

Author(s)
Richèl J.C. Bilderbeek
Index

add_quotes_if_has_spaces, 4
are_beast2_input_lines, 4, 55, 58
are_beast2_input_lines_deep, 5
are_beast2_input_lines_fast, 6
are_identical_alignments, 7
beast2_options_to_table, 8
beastier, 8, 9, 45
beastier_report, 9
check_beast2, 9
check_beast2_options, 10, 11, 13
check_beast2_options_data_types, 11
check_beast2_options_do_not_overwrite_existing_files, 12
check_beast2_options_filenames_differ, 12
check_beast2_options_names, 13
check_beast2_optionses, 10
check_beast2_path, 13
check_can_create_dir_for_state_output_file, 14
check_can_create_file, 15
check_can_create_screenlog_file, 15
check_can_create_state_output_file, 16
check_can_create_tracelog_file, 16
check_can_create_treeelog_file, 17
check_input_filename, 17
check_input_filename_validity, 18
check_n_threads, 19
check_os, 19
check_rename_fun, 38, 64
check_rng_seed, 20
continue_beast2, 21
create_beast2_continue_cmd_from_options, 21
create_beast2_options, 8, 10–18, 21, 22, 22, 25, 32, 36, 42, 63, 64, 67
create_beast2_run_cmd, 24
create_beast2_run_cmd_from_options, 25
create_beast2_validate_cmd, 26
create_beast2_validate_cmd_bin, 27
create_beast2_validate_cmd_jar, 28
create_beast2_version_cmd, 29
create_beast2_version_cmd_bin, 29
create_beast2_version_cmd_jar, 30
create_mcbette_beast2_options, 31
create_random_alignment
create_random_fasta, 33
create_random_phylogeny, 44
create_temp_input_filename, 17, 18, 23, 24, 26–28, 31, 35, 37, 54, 65
create_temp_state_filename, 23, 31, 35, 38, 65
default_params_doc, 35
DNAbin, 32, 57
do_minimal_run, 39
get_alignment_ids_from_xml_filename, 40
get_beast2_example_filename, 40
get_beast2_example_filenames, 41
get_beast2_main_class_name, 42
get_beast2_options_filenames, 42
get_beast2_version, 43
good_beastier_path, 43, 44
get_beastier_paths, 44, 44
get_beastier_tempfilename, 45
good_beastier_tempfilename, 45
good_default_beast2_bin_path, 5, 6, 9, 14, 18, 23, 25–27, 29, 30, 32, 36, 37, 41, 43, 46, 50, 51, 55, 58, 66
good_default_beast2_download_url, 47
good_default_beast2_download_url_linux, 47
good_default_beast2_download_url_win, 48
good_default_beast2_folder, 36, 41, 46, 49, 50, 51
get_default_beast2_jar_path, 5, 6, 9, 14, 18, 23, 25, 26, 28–30, 36, 37, 43, 49, 49, 51, 55, 58, 66
get_default_beast2_path, 50
get_default_beast2_version, 37, 47, 48, 51, 56
get_default_java_path, 52
get_duplicate_param_ids, 52, 56
get_java_version, 53
get_remove_dir_fun, 38, 64
get_remove_hex_fun, 38, 64
get_replace_dir_fun, 38, 64
get_trees_filenames, 54
gives_beast2_warning, 54

has_unique_ids, 53, 55

install_beast2, 46, 49–51, 56
is_alignment, 57
is_beast2_input_file, 5–7, 55, 58
is_beast2_installed, 59
is_bin_path, 60
is_jar_path, 60
is_on_appveyor, 61
is_on_ci, 62
is_on_travis, 62
is_win_bin_path, 63

NA, 19, 20, 23, 24, 31, 37, 38, 64, 65

print_beast2_options, 63

remove_file_if_present, 64
rename_beast2_options_filenames, 64
return, 38, 64
run_beast2, 65
run_beast2_from_options, 39, 66

save_lines, 67
save_nexus_as_fasta, 68
stop, 12, 14–20, 40

tempfile, 45

uninstall_beast2, 68
upgrade_beast2, 69