Package ‘beastier’

October 30, 2020

Type Package
Title Call 'BEAST2'
Version 2.2.1
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
LazyData true
RoxygenNote 7.1.1
Imports ape, assertive, beautier (>= 2.3.5), phangorn, pryr, rappdirs, remotes, rJava, stringr, xml2
Suggests hunspell, knitr, 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 https://github.com/ropensci/onboarding/issues/209), Thijs Janzen [ctb]

Repository  CRAN

Date/Publication  2020-10-30 14:50:03 UTC

R topics documented:

are_beast2_input_lines ................................. 4
are_beast2_input_lines_deep .............................. 5
are_beast2_input_lines_fast ............................. 6
are_identical_alignments ............................... 7
beast2_internal_filenames_to_table ...................... 7
beast2_options_to_table ................................ 8
beastier .................................................. 8
beastier_report .......................................... 9
check_beast2 .............................................. 9
check_beast2_internal_filenames ......................... 10
check_beast2_internal_filenames_data_types .............. 11
check_beast2_internal_filenames_filenames_differ ........ 11
check_beast2_internal_filenames_names ................. 12
check_beast2_options .................................... 13
check_beast2_optionses ................................. 13
check_beast2_options_data_types ......................... 14
check_beast2_options_do_not_overwrite_existing_files ...... 15
check_beast2_options_filenames_differ ................. 15
check_beast2_options_names ............................. 16
check_beast2_path ....................................... 17
check_can_create_file .................................. 17
check_input_filename ..................................... 18
check_n_threads ......................................... 19
check_os .................................................. 19
check_rng_seed .......................................... 20
create_beast2_internal_filenames ....................... 21
create_beast2_options .................................... 21
create_beast2_run_cmd ................................... 23
create_beast2_validate_cmd .............................. 24
create_beast2_validate_cmd_bin ......................... 25
create_beast2_validate_cmd_jar ......................... 26
create_beast2_version_cmd .............................. 27
create_beast2_version_cmd_bin ......................... 28
create_beast2_version_cmd_jar ......................... 29
create_mcbette_beast2_options ......................... 29
create_temp_input_filename ............................ 31
create_temp_state_filename ............................ 31
default_params_doc ..................................... 31
do_minimal_run .......................................... 35
R topics documented:

get_alignment_ids_from_xml_filename .............................................. 36
get_beast2_example_filename .......................................................... 36
get_beast2_example_filenames .......................................................... 37
get_beast2_main_class_name ............................................................. 38
get_beast2_options_filenames ........................................................... 38
get_beast2_version ........................................................................ 39
get_beastier_path ........................................................................ 39
get_beastier_paths ........................................................................ 40
get_default_beast2_bin_path ............................................................... 41
get_default_beast2_download_url ......................................................... 42
get_default_beast2_download_url_linux ................................................. 43
get_default_beast2_download_url_win .................................................... 43
get_default_beast2_folder ................................................................ 44
get_default_beast2_jar_path ............................................................... 44
get_default_beast2_path .................................................................. 45
get_default_beast2_version ................................................................. 46
get_default_java_path ..................................................................... 47
get_duplicate_param_ids .................................................................. 47
get_java_version ........................................................................... 48
get_trees_filenames ........................................................................ 49
gives_beast2_warning ..................................................................... 49
has_unique_ids ............................................................................... 50
install_beast2 .................................................................................. 51
is_alignment .................................................................................... 52
is_beast2_input_file ....................................................................... 53
is_beast2_installed ......................................................................... 54
is_bin_path ...................................................................................... 55
is_jar_path ....................................................................................... 55
is_on_appveyor ............................................................................... 56
is_on_ci .......................................................................................... 57
is_on_travis .................................................................................... 57
print_beast2_internal_filenames ......................................................... 58
print_beast2_options ...................................................................... 58
remove_file_if_present ................................................................... 59
rename_beast2_options_filenames ....................................................... 59
run_beast2 ....................................................................................... 60
run_beast2_from_options ................................................................ 61
save_lines ....................................................................................... 62
save_nexus_as_fasta ....................................................................... 63
uninstall_beast2 .............................................................................. 63
update_beastier .............................................................................. 64
upgrade_beast2 ............................................................................... 64

Index 65
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

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 debug-
           ging
  method  the method to check. Can be 'deep' or 'fast'. The 'deep' method uses BEAST2
t          to validate the complete file. The 'fast' method uses some superficial tests (for
t          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()) {
  get_beastier_path("anthus_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)
testit::assert(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
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_internal_filenames_to_table

Convert a beast2_internal_filenames to a table

Description
Convert a beast2_internal_filenames to a table

Usage
beast2_internal_filenames_to_table(beast2_internal_filenames)

Arguments
beast2_internal_filenames
a list of internally used BEAST2 filenames, as created by create_beast2_internal_filenames
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**

```r
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)
}
```
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 = 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_internal_filenames

_Check if the beast2_internal_filenames is a valid BEAST2 internal filenames object._

Description

Calls stop if the BEAST2 internal filenames object is invalid

Usage

```r
check_beast2_internal_filenames(beast2_internal_filenames)
```

Arguments

- `beast2_internal_filenames`:
  
a list of internally used BEAST2 filenames, as created by `create_beast2_internal_filenames`

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use `create_beast2_internal_filenames` to create a valid BEAST2 internal filenames object

Examples

```r
if (beastier::is_beast2_installed()) {
  check_beast2_internal_filenames(
    create_beast2_internal_filenames(
      create_beast2_options(
        input_filename = get_beastier_path("2_4.xml")
      )
    )
  )
}
```
check_beast2_internal_filenames_data_types
   Check if the beast2_internal_filenames, which is a list, has all
elements of the right data types

Description
   Calls stop if not.

Usage
   check_beast2_internal_filenames_data_types(beast2_internal_filenames)

Arguments
   beast2_internal_filenames
       a list of internally used BEAST2 filenames, as created by create_beast2_internal_filenames

Value
   nothing

Author(s)
   Richèl J.C. Bilderbeek

See Also
   Use check_beast2_internal_filenames to check the entire beast2_internal_filenames object

check_beast2_internal_filenames_filenames_differ
   Check if the filenames in beast2_internal_filenames differ

Description
   Calls stop if not.

Usage
   check_beast2_internal_filenames_filenames_differ(beast2_internal_filenames)

Arguments
   beast2_internal_filenames
       a list of internally used BEAST2 filenames, as created by create_beast2_internal_filenames
check_beast2_internal_filenames_names

Check if the beast2_internal_filenames, which is a list, has all the elements needed.

Description
Calls stop if not.

Usage
check_beast2_internal_filenames_names(beast2_internal_filenames)

Arguments
beast2_internal_filenames
a list of internally used BEAST2 filenames, as created by create_beast2_internal_filenames

Value
nothing

Author(s)
Richèl J.C. Bilderbeek

See Also
Use check_beast2_internal_filenames to check the entire beast2_internal_filenames object
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

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_optionses is a valid BEAST2 options object.

Description
Calls stop if the BEAST2 option object is invalid

Usage
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

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
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,
  beast2_internal_filenames
)

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

beast2_internal_filenames  a list of internally used BEAST2 filenames, as created by create_beast2_internal_filenames

Author(s)

Richêl 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
check_beast2_options_names

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)

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_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
check_can_create_file(filename, overwrite = TRUE)
check_input_filename

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_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_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**

```r
# 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)

**Value**

nothing. Will stop if the OS is unsupported
check_rng_seed

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```r
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**

```r
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**

```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)
```
create_beast2_internal_filenames

Create a list with the internally used BEAST2 filenames

Description
Create a list with the internally used BEAST2 filenames

Usage
create_beast2_internal_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
a list with the internally used BEAST2 filenames

Examples
beast2_options <- create_beast2_options(
  input_filename = get_beastier_path("2_4.xml")
)
if (is_beast2_installed()) {
  create_beast2_internal_filenames(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,
)
```r
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
  - 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

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()
    )

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
create_beast2_validate_cmd

Create the terminal command to validate a BEAST2 input file

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()
)
```
create_beast2_validate_cmd_bin

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()) {
  cmds <- create_beast2_validate_cmd(
    input_filename = "input.xml"
  )
  testit::assert(cmds[2] == "-cp")
}

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

create_beast2_validate_cmd_bin(
  input_filename,
  beast2_bin_path = get_default_beast2_bin_path()
)
create_beast2_validate_cmd_jar

**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()) {
  cmds <- create_beast2_validate_cmd_bin(
    input_filename = "input.xml"
  )
  testit::assert(length(cmds) == 3)
  testit::assert(cmds[2] == "-validate")
}
```

---

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**

```r
create_beast2_validate_cmd_jar(
  input_filename,  
  beast2_jar_path = get_default_beast2_jar_path()
)
```
create_beast2_version_cmd

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()) {
  cmds <- create_beast2_validate_cmd_jar(
    input_filename = "input.xml"
  )
  testit::assert(length(cmds) == 6)
  testit::assert(cmds[2] == "-cp")
}
```

---

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

```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
create_beast2_version_cmd_bin

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()) {
  cmds <- create_beast2_version_cmd()
  testit::assert(cmds[2] == "-cp")
}

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())

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()) {
  cmds <- create_beast2_version_cmd_bin()
  testit::assert(length(cmds) == 2)
  testit::assert(cmds[2] == "-version")
}
create_beast2_version_cmd_jar

*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**

```
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**

```
if (is_beast2_installed()) {
  cmds <- create_beast2_version_cmd_jar()
  testit::assert(length(cmds) == 5)
  testit::assert(cmds[2] == \"-cp\")
}
```

---

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
  - 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 debugging.

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()

default_params_doc

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

Description

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

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()
Usage

default_params_doc(
    beast2_bin_path,
    beast2_folder,
    beast2_internal_filenames,
    beast2_jar_path,
    beast2_options,
    beast2_optionses,
    beast2_path,
    beast2_version,
    beast2_working_dir,
    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,
    use_beagle,
    verbose
)

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_internal_filenames**

a list of internally used BEAST2 filenames, as created by `{create_beast2_internal_filenames}`

**beast2_jar_path**

name of the BEAST2 jar file (usually has a `.jar` extension). Use `{get_default_beast2_jar_path}` to get the default BEAST2 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 `beautier` clock model

**clock_models**

a list of one or more `beautier` 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.

cmpc: one beautier MCMC

misc_options: one beautier misc_options object

ntaxa: The number of taxa

nthreads: 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 beutier site model
do_minimal_run

site_models  one or more beautier site models

Description

To achieve this, run_beast2_from_options is called.

Usage

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

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

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

# 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

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

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.

Examples

```r
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
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)

Richèl 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
get_beastier_paths

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
get_default_beast2_bin_path

Examples

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

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() && 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

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
get_default_beast2_jar_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_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
print(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 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**

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
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
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_2)) == 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**

```
get_java_version()
```

**Author(s)**

Richèl J.C. Bilderbeek

**Examples**

```
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**

```r
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**

```r
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**

```r
gives_beast2_warning(
    filename,
    verbose = FALSE,
    beast2_path = get_default_beast2_path()
)
```
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

```r
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**  
*Determine if the XML text has unique parameter IDs*

Description

Determine if the XML text has unique parameter IDs

Usage

```r
has_unique_ids(text)
```
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

```r
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))
```
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)

Value

Nothing. Will install BEAST2

Author(s)

Richèl J.C. Bilderbeek

Examples

## 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

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

```r
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
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()) {
  print("BEAST2 is installed")
}
```
is_bin_path

<table>
<thead>
<tr>
<th>Description</th>
<th>Is the path a path to the BEAST2 binary file? Does not check if the file at that path is present</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td>is_bin_path(path)</td>
</tr>
<tr>
<td>Arguments</td>
<td>path a string to a path</td>
</tr>
<tr>
<td>Value</td>
<td>TRUE if the path is a path to a BEAST2 binary file</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Richèl J.C. Bilderbeek</td>
</tr>
</tbody>
</table>
| 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

<table>
<thead>
<tr>
<th>Description</th>
<th>Is the path a path to the BEAST2 jar file? Does not check if the file at that path is present</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td>is_jar_path(path)</td>
</tr>
</tbody>
</table>
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

```r
if (is_on_appveyor()) {
  print("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()) {
    print("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()) {
    print("Running on Travis CI")
}
print_beast2_internal_filenames

Print a beast2_internal_filenames as a table

Description

Print a beast2_internal_filenames as a table

Usage

print_beast2_internal_filenames(beast2_internal_filenames)

Arguments

beast2_internal_filenames

a list of internally used BEAST2 filenames, as created by create_beast2_internal_filenames

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

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

Description

Run BEAST2

Usage

run_beast2_from_options(beast2_options = create_beast2_options())
Argument

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)
}
```

---

**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 <- tempfile(fileext = ".txt")
save_lines(filename = filename, lines = text)
```
save_nexus_as_fasta  Save a NEXUS file as a FASTA file

Description

Save a NEXUS file as a FASTA file

Usage

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  Uninstall BEAST2

Description

Uninstall BEAST2

Usage

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

Author(s)

Richèl J.C. Bilderbeek
Examples

## Not run:
uninstall_beast2()

## End(Not run)

update_beastier

Update all beastier dependencies, by installing their latest versions

Description

Update all beastier dependencies, by installing their latest versions

Usage

update_beastier()

Author(s)

Richèl J.C. Bilderbeek

upgrade_beast2

Upgrade BEAST2.

Description

Will stop if BEAST2 is not installed

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 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)

Author(s)

Richèl J.C. Bilderbeek
Index

are_beast2_input_lines, 4, 50, 53
are_beast2_input_lines_deep, 5
are_beast2_input_lines_fast, 6
are_identical_alignments, 7
beast2_internal_filenames_to_table, 7
beast2_options_to_table, 8
beastier, 8, 9
beastier_report, 9
check_beast2, 9
check_beast2_internal_filenames, 10, 11, 12
check_beast2_internal_filenames_data_types, 11
check_beast2_internal_filenames_filenames_differ, 11
check_beast2_internal_filenames_names, 12
check_beast2_options, 13, 14, 16
check_beast2_options_data_types, 14
check_beast2_options_do_not_overwrite_existing, 15
check_beast2_options_filenames_differ, 15
check_beast2_options_names, 16
check_beast2_optionses, 13
check_beast2_path, 17
check_can_create_file, 17
check_input_filename, 18
check_n_threads, 19
check_os, 19
check_rename_fun, 34, 59
check_rng_seed, 20
create_beast2_internal_filenames, 7, 10–12, 15, 21, 33, 58
create_beast2_options, 8, 13–16, 21, 21, 30, 33, 38, 58, 59, 62
create_beast2_run_cmd, 23
create_beast2_validate_cmd, 24
create_beast2_validate_cmd_bin, 25
create_beast2_validate_cmd_jar, 26
create_beast2_version_cmd, 27
create_beast2_version_cmd_bin, 28
create_beast2_version_cmd_jar, 29
create_mcbette_beast2_options, 29
create_temp_input_filename, 18, 22, 23, 25–27, 30, 31, 34, 49, 60
create_temp_state_filename, 22, 30, 31, 34, 60
default_params_doc, 31
DNABin, 52
do_minimal_run, 35
get_alignment_ids_from_xml_filename, 36
get_beast2_example_filename, 36
get_beast2_example_filenames, 37
get_beast2_main_class_name, 38
get_beast2_options_filenames, 38
get_beast2_version, 39
get_beastier_path, 39, 40
get_beastier_paths, 40, 40
get_default_beast2_bin_path, 4, 5, 9, 17, 22, 24–28, 30, 32, 33, 37, 39, 41, 41, 45, 46, 50, 53, 61
get_default_beast2_download_url, 42
get_default_beast2_download_url_linux, 43
get_default_beast2_download_url_win, 43
get_default_beast2_folder, 33, 37, 41, 44, 45, 46
get_default_beast2_jar_path, 4, 5, 9, 17, 22, 24, 25, 27, 29, 33, 39, 44, 44, 46, 50, 53, 61
get_default_beast2_path, 45
get_default_beast2_version, 33, 42, 43, 46, 51, 52

65
get_default_java_path, 47
get_duplicate_param_ids, 47, 51
get_java_version, 48
get_remove_dir_fun, 34, 59
get_remove_hex_fun, 34, 59
get_replace_dir_fun, 34, 59
get_trees_filenames, 49

gives_beast2_warning, 49

has_unique_ids, 48, 50

install_beast2, 41, 44–46, 51
is_alignment, 52
is_beast2_input_file, 4–6, 50, 53
is_beast2_installed, 54
is_bin_path, 55
is_jar_path, 55
is_on_appveyor, 56
is_on_ci, 57
is_on_travis, 57

NA, 19, 20, 22, 23, 30, 34, 59, 60

print_beast2_internal_filenames, 58
print_beast2_options, 58

remove_file_if_present, 59
rename_beast2_options_filenames, 59
return, 34, 59
run_beast2, 60
run_beast2_from_options, 35, 61

save_lines, 62
save_nexus_as_fasta, 63
stop, 15, 17–20, 36, 64

uninstall_beast2, 63
update_beastier, 64
upgrade_beast2, 64