Package 'babette'

June 25, 2024

Title Control 'BEAST2'

Version 2.3.4

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 commonly accompanied by 'BEAUti 2', 'Tracer' and 'DensiTree'.
'babette' provides for an alternative workflow of using
all these tools separately. This allows doing complex Bayesian
phylogenetics easily and reproducibly from 'R'.

License GPL-3

RoxygenNote 7.2.3

VignetteBuilder knitr

URL <https://docs.ropensci.org/babette/> (website),

<https://github.com/ropensci/babette>

BugReports <https://github.com/ropensci/babette/issues>

Depends beautier (>= 2.6.12), beastier (>= 2.5), mauricer (>= 2.5.2),
R (>= 3.5.0), tracerer

Imports phangorn, rlang (>= 1.1.0), stringr, xml2

Suggests ape, ggplot2, knitr, lintr, nLTT, rappdirs, rmarkdown,
spelling, testthat (>= 2.1.0)

Language en-US

Encoding UTF-8

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/richelbilderbeek/onboarding/issues/209),
David Winter [rev] (David reviewed the package for rOpenSci, see

1

Contents

---

bbt_continue   *Continue a BEAST2 run*

Description

Do a full run: create a 'BEAST2' configuration file (like 'BEAUti 2'), run 'BEAST2', parse results (like 'Tracer')

Usage

```
bbt_continue(fasta_filename, inference_model, beast2_options)
```

Arguments

fasta_filename   a FASTA filename

inference_model

                 a Bayesian phylogenetic inference model, as returned by create_inference_model

beast2_options   'BEAST2' options, as can be created by create_beast2_options

Value

a list with the following elements:

- estimates: a data frame with 'BEAST2' parameter estimates
- [alignment_id]_trees: a multiPhylo containing the phylogenies in the 'BEAST2' poster-
  ior. [alignment_id] is the ID of the alignment. For example, when running bbt_run_from_model
  with anthus_aco.fas, this element will have name anthus_aco_trees
- operators: a data frame with the 'BEAST2' MCMC operator acceptances
- output: a numeric vector with the output sent to standard output and error streams
- ns: (optional) the results of a marginal likelihood estimation, will exist only when create_ns_mcmc
  was used for mcmc. This structure will contain the following elements:
  - marg_log_lik the marginal log likelihood estimate
  - marg_log_lik_sd the standard deviation around the estimate
  - estimates the parameter estimates created during the marginal likelihood estimation
  - trees the trees created during the marginal likelihood estimation

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [remove_burn_ins](#) to remove the burn-ins from the posterior's estimates (posterior$estimates)

Examples

```
if (beautier::is_on_ci() && is_beast2_installed()) {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  # A simple FASTA file
  fasta_filename <- beautier::get_beautier_path("test_output_0.fas")

  # Simple short inference
  inference_model <- create_test_inference_model()

  # Default BEAST2 options
  beast2_options <- create_beast2_options()

  bbt_run_from_model(
    fasta_filename = fasta_filename,
    inference_model = inference_model,
    beast2_options = beast2_options
  )

  bbt_continue(
    fasta_filename = fasta_filename,
    inference_model = inference_model,
```

```
    beast2_options = beast2_options
  )

  # Cleanup
  bbt_delete_temp_files(
    inference_model = inference_model,
    beast2_options = beast2_options
  )
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

---

bbt_delete_temp_files   *Delete all the temporary files created by [bbt_run_from_model](#)*

Description

Delete all the temporary files created by [bbt_run_from_model](#)

Usage

```
bbt_delete_temp_files(inference_model, beast2_options)
```

Arguments

inference_model

                a Bayesian phylogenetic inference model, as returned by [create_inference_model](#)

beast2_options   'BEAST2' options, as can be created by [create_beast2_options](#)

Value

Nothing.

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (beautier::is_on_ci() && is_beast2_installed()) {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  # Do a minimal run
  inference_model <- create_test_inference_model()
  beast2_options <- create_beast2_options()
  bbt_run_from_model(
    fasta_filename = get_fasta_filename(),
    inference_model = inference_model,
```

```
    beast2_options = beast2_options
  )

  # Cleanup
  bbt_delete_temp_files(
    inference_model = inference_model,
    beast2_options = beast2_options
  )

  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

bbt_run   *Run BEAST2*

Description

Do a full BEAST2 run: create a 'BEAST2' configuration file (like 'BEAUti 2'), run 'BEAST2', parse results (like 'Tracer')

Usage

```
bbt_run(
  fasta_filename,
  tipdates_filename = NA,
  site_model = beautier::create_jc69_site_model(),
  clock_model = beautier::create_strict_clock_model(),
  tree_prior = beautier::create_yule_tree_prior(),
  mrca_prior = NA,
  mcmc = beautier::create_mcmc(),
  beast2_input_filename = beastier::create_temp_input_filename(),
  rng_seed = 1,
  beast2_output_state_filename = beastier::create_temp_state_filename(),
  beast2_path = beastier::get_default_beast2_path(),
  overwrite = TRUE,
  verbose = FALSE
)
```

Arguments

fasta_filename   a FASTA filename

tipdates_filename

              name of the file containing tip dates

site_model   one site model, see [create_site_models](#)

clock_model   one clock model, see [create_clock_model](#)

tree_prior   one tree priors, as created by [create_tree_prior](#)

mrca_prior   one Most Recent Common Ancestor prior, as returned by create_mrca_prior

mcmc   the MCMC options, see create_mcmc

beast2_input_filename

                  path of the 'BEAST2' configuration file. By default, this file is put in a tempo-
                  rary folder with a random filename, as the user needs not read it: it is used as
                  input of 'BEAST2'. Specifying a beast2_input_filename allows to store that
                  file in a more permanently stored location.

rng_seed   the random number generator seed. Must be either NA or a positive non-zero
                  value. An RNG seed of NA results in 'BEAST2' picking a random seed.

beast2_output_state_filename

                  name of the final state file created by 'BEAST2', containing the operator accep-
                  tances. By default, this file is put a temporary folder with a random filename,
                  as the user needs not read it: its content is parsed and returned by this func-
                  tion. Specifying a beast2_output_state_filename allows to store that file in
                  a more permanently stored location.

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

overwrite   will 'BEAST2' overwrite files? Like 'BEAST2', this is set to TRUE by default.
                  If TRUE, 'BEAST2' will overwrite the beast2_options$output_state_filename
                  if its present. If FALSE, 'BEAST2' will not overwrite the beast2_options$output_state_filename
                  if its present and babette will give an error message. Note that if overwrite
                  is set to FALSE when a tracelog (see create_tracelog), screenlog (see [cre-
                  ate_screenlog](create_screenlog)) or treelog (see create_treelog) file already exists, 'BEAST2'
                  (and thus babette) will freeze.

verbose   set to TRUE for more output

Details

Prefer using bbt_run_from_model, as it has a cleaner interface.

Value

a list with the following elements:

- estimates: a data frame with 'BEAST2' parameter estimates
- [alignment_id]_trees: a multiPhylo containing the phylogenies in the 'BEAST2' poste-
  rior. [alignment_id] is the ID of the alignment. For example, when running [bbt_run](bbt_run) with
  anthus_aco.fas, this element will have name anthus_aco_trees
- operators: a data frame with the 'BEAST2' MCMC operator acceptances
- output: a numeric vector with the output sent to standard output and error streams
- ns: (optional) the results of a marginal likelihood estimation, will exist only when create_ns_mcmc
  was used for the MCMC. This structure will contain the following elements:
    - marg_log_lik the marginal log likelihood estimate

  – `marg_log_lik_sd` the standard deviation around the estimate

  – `estimates` the parameter estimates created during the marginal likelihood estimation

  – `trees` the trees created during the marginal likelihood estimation

Author(s)

Richèl J.C. Bilderbeek

See Also

Use `remove_burn_ins` to remove the burn-ins from the posterior's estimates (`posterior$estimates`)

Examples

```
if (beautier::is_on_ci() && is_beast2_installed()) {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  # Setup for a short run
  mcmc <- create_test_mcmc()

  # Store filenames for cleanup.
  # Note that 'bbt_run_from_model allows for easier cleanup
  mcmc$tracelog$filename <- tempfile()
  mcmc$treelog$filename <- tempfile()
  mcmc$screenlog$filename <- tempfile()
  beast2_input_filename <- tempfile()
  beast2_output_state_filename <- tempfile()

  bbt_run(
    fasta_filename = get_babette_path("anthus_aco.fas"),
    beast2_input_filename = beast2_input_filename,
    beast2_output_state_filename = beast2_output_state_filename,
    mcmc = mcmc
  )

  # Cleanup
  # Again, note that 'bbt_run_from_model allows for easier cleanup
  file.remove(mcmc$tracelog$filename)
  file.remove(mcmc$treelog$filename)
  file.remove(mcmc$screenlog$filename)
  file.remove(beast2_input_filename)
  file.remove(beast2_output_state_filename)
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

---

bbt_run_from_model   *Run BEAST2*

Description

Do a full run: create a 'BEAST2' configuration file (like 'BEAUti 2'), run 'BEAST2', parse results (like 'Tracer')

Usage

```
bbt_run_from_model(
  fasta_filename,
  inference_model = beautier::create_inference_model(),
  beast2_options = beastier::create_beast2_options()
)
```

Arguments

fasta_filename a FASTA filename

inference_model

                a Bayesian phylogenetic inference model, as returned by create_inference_model

beast2_options 'BEAST2' options, as can be created by create_beast2_options

Value

a list with the following elements:

- estimates: a data frame with 'BEAST2' parameter estimates
- [alignment_id]_trees: a multiPhylo containing the phylogenies in the 'BEAST2' posterior. [alignment_id] is the ID of the alignment. For example, when running bbt_run_from_model with anthus_aco.fas, this element will have name anthus_aco_trees
- operators: a data frame with the 'BEAST2' MCMC operator acceptances
- output: a numeric vector with the output sent to standard output and error streams
- ns: (optional) the results of a marginal likelihood estimation, will exist only when create_ns_mcmc was used for mcmc. This structure will contain the following elements:
    - marg_log_lik the marginal log likelihood estimate
    - marg_log_lik_sd the standard deviation around the estimate
    - estimates the parameter estimates created during the marginal likelihood estimation
    - trees the trees created during the marginal likelihood estimation

Author(s)

Richèl J.C. Bilderbeek

See Also

Use remove_burn_ins to remove the burn-ins from the posterior's estimates (`posterior$estimates`)

Examples

```
if (beautier::is_on_ci() && is_beast2_installed()) {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  # Simple short inference
  inference_model <- create_test_inference_model()

  # Default BEAST2 options
  beast2_options <- create_beast2_options()

  bbt_run_from_model(
    fasta_filename = get_babette_path("anthus_aco.fas"),
    inference_model = inference_model,
    beast2_options = beast2_options
  )

  # Cleanup
  bbt_delete_temp_files(
    inference_model = inference_model,
    beast2_options = beast2_options
  )
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

---

bbt_self_test   *Do a self test to verify babette that works correctly.*

Description

Do a self test to verify babette that works correctly.

Usage

```
bbt_self_test(beast2_options = beastier::create_beast2_options())
```

Arguments

`beast2_options`   'BEAST2' options, as can be created by create_beast2_options

Value

Nothing. Will raise an exception if something is wrong.

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (beautier::is_on_ci() && is_beast2_installed()) {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  bbt_self_test()

  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

---

check_beast2_pkgs   *Checks if* bbt_run *has the 'BEAST2' packages needed to process its*
                           arguments. Will stop *if not.*

Description

For example, to use a Nested Sampling MCMC, the 'BEAST2' 'NS' package needs to be installed.

Usage

```
check_beast2_pkgs(mcmc, beast2_path = get_default_beast2_bin_path())
```

Arguments

mcmc   the MCMC options, see create_mcmc

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.

Examples

```
if (beautier::is_on_ci() && is_beast2_installed()) {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  # Minimal BEAST2 setup
  check_beast2_pkgs(mcmc = create_mcmc())
```

```
  # BEAST2 with NS package installed
  if (is_beast2_ns_pkg_installed()) {
    check_beast2_pkgs(mcmc = create_ns_mcmc())
  }

  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

---

create_test_bbt_run_output

Get an example output of bbt_run *or* bbt_run_from_model.

Description

This output is used in testing.

Usage

```
create_test_bbt_run_output()
```

Value

the same results as bbt_run or bbt_run_from_model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()

create_test_bbt_run_output()

beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()
```

`create_test_ns_output`   *Create NS testing output*

Description

Create testing output similar to when running a 'BEAST2' run with nested sampling

Usage

```
create_test_ns_output()
```

Value

a text of type character vector.

Author(s)

Richèl J.C. Bilderbeek

See Also

Use parse_beast2_output_to_ns to parse this output to a Nested Sampling result. See create_ns_mcmc to see how to do a marginal likelihood estimation using Nested Sampling.

Examples

```
beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()

create_test_ns_output()

beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()
```

`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.

Usage

```
default_params_doc(
  beast2_input_filename,
  beast2_options,
  beast2_output_log_filename,
  beast2_output_state_filename,
  beast2_output_trees_filenames,
  beast2_path,
  beast2_working_dir,
  cleanup,
  clock_model,
  clock_models,
  fasta_filename,
  fasta_filenames,
  inference_model,
  mcmc,
  mrca_prior,
  mrca_priors,
  overwrite,
  rng_seed,
  site_model,
  site_models,
  tipdates_filename,
  tree_prior,
  tree_priors,
  verbose
)
```

Arguments

beast2_input_filename

> path of the 'BEAST2' configuration file. By default, this file is put in a tempo-
> rary folder with a random filename, as the user needs not read it: it is used as
> input of 'BEAST2'. Specifying a `beast2_input_filename` allows to store that
> file in a more permanently stored location.

beast2_options 'BEAST2' options, as can be created by [create_beast2_options](#)

beast2_output_log_filename

> name of the log file created by 'BEAST2', containing the parameter estimates
> in time. By default, this file is put a temporary folder with a random filename,
> as the user needs not read it: its content is parsed and returned by this function.
> Specifying a `beast2_output_log_filename` allows to store that file in a more
> permanently stored location.

beast2_output_state_filename

> name of the final state file created by 'BEAST2', containing the operator accep-
> tances. By default, this file is put a temporary folder with a random filename,
> as the user needs not read it: its content is parsed and returned by this func-
> tion. Specifying a `beast2_output_state_filename` allows to store that file in
> a more permanently stored location.

beast2_output_trees_filenames

    name of the one or more trees files created by 'BEAST2', one per alignment. By default, these files are put a temporary folder with a random filename, as the user needs not read it: their content is parsed and returned by this function. Specifying `beast2_output_trees_filenames` allows to store these one or more files in a more permanently stored location.

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_working_dir

    the folder 'BEAST2' will work in. This is an (empty) temporary folder by default. This allows to call 'BEAST2' in multiple parallel processes, as each process can have its own working directory

cleanup   set to FALSE to keep all temporary files

clock_model   one clock model, see [create_clock_model](#)

clock_models   one or more clock models, see [create_clock_models](#)

fasta_filename a FASTA filename

fasta_filenames

    one or more FASTA filename, each with one alignment

inference_model

    a Bayesian phylogenetic inference model, as returned by [create_inference_model](#)

mcmc   the MCMC options, see [create_mcmc](#)

mrca_prior   one Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)

mrca_priors   a list of one or more Most Recent Common Ancestor priors, as returned by [create_mrca_prior](#)

overwrite   will 'BEAST2' overwrite files? Like 'BEAST2', this is set to [TRUE](#) by default. If [TRUE](#), 'BEAST2' will overwrite the `beast2_options$output_state_filename` if its present. If [FALSE](#), 'BEAST2' will not overwrite the `beast2_options$output_state_filename` if its present and [babette](#) will give an error message. Note that if overwrite is set to [FALSE](#) when a tracelog (see [create_tracelog](#)), screenlog (see [create_screenlog](#)) or treelog (see [create_treelog](#)) file already exists, 'BEAST2' (and thus [babette](#)) will freeze.

rng_seed   the random number generator seed. Must be either NA or a positive non-zero value. An RNG seed of NA results in 'BEAST2' picking a random seed.

site_model   one site model, see [create_site_models](#)

site_models   one or more site models, see [create_site_models](#)

tipdates_filename

    name of the file containing tip dates

tree_prior   one tree priors, as created by [create_tree_prior](#)

tree_priors   one or more tree priors, see [create_tree_priors](#)

verbose   set to TRUE for more output

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

---

get_alignment_ids_from_xml

Get the alignment IDs from one or more 'BEAST2' XML input files.

Description

Get the alignment IDs from one or more 'BEAST2' XML input files.

Usage

```
get_alignment_ids_from_xml(xml_filename)
```

Arguments

xml_filename   name of a 'BEAST2' XML input file.

Value

a character vector with one or more alignment IDs.

Author(s)

Richèl J.C. Bilderbeek

Examples

```
beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()

alignment_ids <- get_alignment_ids_from_xml(
  get_babette_path("anthus_2_4.xml")
)

beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()
```

---

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

Description

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

Usage

```
get_babette_path(filename)
```

Arguments

filename   the file's name, without the path

Value

the full path of the filename, if and only if the file is present. Will stop otherwise.

Author(s)

Richèl J.C. Bilderbeek

See Also

for more files, use get_babette_paths

Examples

```
beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()

get_babette_path("anthus_aco.fas")

beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()
```

get_babette_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_babette_paths(filenames)
```

Arguments

filenames   the files' names, without the path

Value

the filenames' full paths, if and only if all files are present. Will stop otherwise.

Author(s)

Richèl J.C. Bilderbeek

See Also

for one file, use get_babette_path

Examples

```
beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()

get_babette_paths(c("anthus_aco.fas", "anthus_nd2.fas"))

beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()
```

---

parse_beast2_output   *Process the 'BEAST2' output dependent on 'BEAST2' package specifics*

Description

Process the 'BEAST2' output dependent on 'BEAST2' package specifics

Usage

```
parse_beast2_output(out, inference_model)
```

Arguments

out   a list with the complete babette output, with elements:

  • output textual output of a 'BEAST2' run

inference_model

  a Bayesian phylogenetic inference model, as returned by create_inference_model

Value

complete babette output with added attributes, which depends on the 'BEAST2' package.

  • `marg_log_lik` the marginal log likelihood estimate
  • `marg_log_lik_sd` the standard deviation around the estimate
  • `estimates` the parameter estimates created during the marginal likelihood estimation
  • `trees` the trees created during the marginal likelihood estimation

Author(s)

Richèl J.C. Bilderbeek

---

parse_beast2_output_to_ns
                              Parse BEAST2 NS output

Description

Parse the BEAST2 output when run with the BEAST2 NS ('Nested Sampling') package.

Usage

```
parse_beast2_output_to_ns(output)
```

Arguments

output   screen output

Value

a list with the following elements:

- marg_log_lik the marginal log likelihood estimate
- marg_log_lik_sd the standard deviation around the estimate

Author(s)

Richèl J.C. Bilderbeek

See Also

use create_test_ns_output to obtain a test screen output.

Examples

```
beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()

parse_beast2_output_to_ns(
  output = create_test_ns_output()
)

beastier::remove_beaustier_folders()
beastier::check_empty_beaustier_folders()
```

---

plot_densitree   *Draw multiple trees on top of one another.*

Description

Draw multiple trees on top of one another.

Usage

```
plot_densitree(phylos, ...)
```

Arguments

phylos   one or more phylogenies, must be of class multiPhylo
...   options to be passed to phangorn's densiTree function

Value

nothing. Will produce a plot.

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (beautier::is_on_ci() && is_beast2_installed()) {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  inference_model <- create_test_inference_model()
  beast2_options <- create_beast2_options()

   out <- bbt_run_from_model(
    get_babette_path("anthus_aco.fas"),
    inference_model = inference_model,
    beast2_options = beast2_options
  )
  bbt_delete_temp_files(
    inference_model = inference_model,
    beast2_options = beast2_options
  )

  plot_densitree(out$anthus_aco_trees)

  # Clean up temporary files created by babette
  bbt_delete_temp_files(
    inference_model = inference_model,
    beast2_options = beast2_options
  )
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

---

prepare_file_creation   *Internal function to prepare for 'BEAST2' creating files*

Description

The inference model and 'BEAST2' options contain paths that may point to sub-sub-sub folders.
Create those folders and test if these folders can be written to

Usage

```
prepare_file_creation(inference_model, beast2_options)
```

Arguments

inference_model

                a Bayesian phylogenetic inference model, as returned by create_inference_model

beast2_options   'BEAST2' options, as can be created by create_beast2_options

Value

Nothing.

Examples

```
# This example will fail on the CRAN
# r-oldrel-macos-x86_64 platform
if (rappdirs::app_dir()$os != "mac") {
  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()

  # For a test inference model, the files can be prepared
  inference_model <- create_test_inference_model()
  beast2_options <- create_beast2_options()
  prepare_file_creation(inference_model, beast2_options)

  beastier::remove_beaustier_folders()
  beastier::check_empty_beaustier_folders()
}
```

---

update_babette   *Deprecated function.*

Description

Update all babette dependencies, by installing their latest versions.

Usage

```
update_babette(upgrade = "default")
```

Arguments

upgrade   Deprecated.

Details

See https://github.com/richelbilderbeek/babetteinstall how to do this.

Value

Nothing.

Author(s)

Giovanni Laudanno, Richèl J.C. Bilderbeek

Index