

Package ‘beautier’

February 25, 2020

Title 'BEAUti' from R

Version 2.3.5

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.
'BEAUti 2' (which is part of 'BEAST2') is a GUI tool that allows users to specify the many possible setups and generates the XML file 'BEAST2' needs to run.
This package provides a way to create 'BEAST2' input files without active user input, but using R function calls instead.

License GPL-3

LazyData true

RoxygenNote 7.0.2

VignetteBuilder knitr

URL <https://docs.ropensci.org/beautier>,
<https://github.com/ropensci/beautier>

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

Imports ape, assertive, pryr, rappdirs, seqinr, stringr, testit

Suggests spelling, devtools, knitr, ggplot2, hunspell, lintr, rmarkdown, testthat (>= 2.1.0)

Language en-US

Encoding UTF-8

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>),
 Paul Van Els [ctb],
 Raphael Scherrer [ctb],
 Yacine B. Chehida [ctb],
 Jana Riederer [ctb]

Repository CRAN

Date/Publication 2020-02-25 12:20:02 UTC

R topics documented:

are_clock_models	11
are_equal_mcmcs	12
are_equal_screenlogs	13
are_equal_tracelogs	14
are_equal_treelogs	15
are_equal_xml_files	16
are_equal_xml_lines	16
are_equivalent_xml_files	17
are_equivalent_xml_lines	18
are_equivalent_xml_lines_all	18
are_equivalent_xml_lines_loggers	19
are_equivalent_xml_lines_operators	20
are_equivalent_xml_lines_section	20
are_fasta_filenames	21
are_ids	22
are_init_clock_models	22
are_init_mrca_priors	23
are_init_site_models	23
are_init_tree_priors	24
are_mrca_align_ids_in_fasta	24
are_mrca_priors	25
are_mrca_taxon_names_in_fasta	25
are_rln_clock_models	26
are_site_models	26
are_tree_priors	27
bd_tree_prior_to_xml_prior_distr	28
beautier	29
cbs_tree_prior_to_xml_prior_distr	29
ccp_tree_prior_to_xml_prior_distr	30
cep_tree_prior_to_xml_prior_distr	31
check_alignment_id	32
check_beauti_options	32
check_clock_model	33
check_clock_models	34
check_file_and_model_agree	35
check_file_exists	35

check_gamma_site_model	36
check_gamma_site_model_names	37
check_gtr_site_model	38
check_gtr_site_model_names	38
check_inference_model	39
check_inference_models	40
check_is_monophyletic	41
check_log_mode	41
check_log_sort	41
check_mcmc	42
check_mcmc_list_element_names	43
check_mcmc_values	43
check_mrca_prior	44
check_mrca_prior_name	45
check_mrca_prior_names	45
check_mrca_prior_taxa_names	46
check_ns_mcmc	46
check_param	47
check_param_names	48
check_param_types	49
check_phylogeny	49
check_rename_fun	50
check_rln_clock_model	51
check_screenlog	51
check_screenlog_names	52
check_screenlog_values	52
check_site_model	53
check_site_models	54
check_site_model_names	55
check_site_model_types	56
check_store_every	56
check_strict_clock_model	57
check_tn93_site_model	58
check_tn93_site_model_names	58
check_tracelog	59
check_tracelog_names	59
check_tracelog_values	60
check_treelog	60
check_treelog_names	61
check_treelog_values	61
check_tree_prior	62
check_tree_priors	63
clock_models_to_xml_operators	64
clock_models_to_xml_prior_distr	64
clock_models_to_xml_state	65
clock_models_to_xml_tracelog	66
clock_model_to_xml_lh_distr	67
clock_model_to_xml_operators	68

clock_model_to_xml_prior_distr	69
clock_model_to_xml_state	70
clock_model_to_xml_tracelog	70
clock_model_to_xml_treelogger	71
compare_lines	72
count_trailing_spaces	73
create_alpha_param	73
create_bd_tree_prior	75
create_beast2_beast_xml	76
create_beast2_input	76
create_beast2_input_beast	78
create_beast2_input_data	79
create_beast2_input_data_sequences	80
create_beast2_input_distr	81
create_beast2_input_distr_lh	82
create_beast2_input_distr_prior	83
create_beast2_input_file	84
create_beast2_input_file_from_model	86
create_beast2_input_from_model	87
create_beast2_input_init	88
create_beast2_input_map	89
create_beast2_input_operators	89
create_beast2_input_run	90
create_beast2_input_state	91
create_beauti_options	92
create_beta_distr	93
create_beta_param	94
create_cbs_tree_prior	95
create_ccp_tree_prior	96
create_cep_tree_prior	97
create_clock_model	98
create_clock_models	99
create_clock_models_from_names	100
create_clock_model_from_name	101
create_clock_rate_param	102
create_data_xml	103
create_distr	103
create_exp_distr	104
create_gamma_distr	105
create_gamma_site_model	106
create_gtr_site_model	108
create_hky_site_model	110
create_inference_model	111
create_inv_gamma_distr	112
create_jc69_site_model	113
create_kappa_1_param	114
create_kappa_2_param	115
create_lambda_param	115

create_laplace_distr	116
create_loggers_xml	117
create_log_normal_distr	118
create_mcmc	119
create_mean_param	120
create_mrca_prior	121
create_mu_param	123
create_m_param	124
create_normal_distr	125
create_ns_mcmc	126
create_one_div_x_distr	128
create_param	129
create_poisson_distr	130
create_rate_ac_param	131
create_rate_ag_param	132
create_rate_at_param	133
create_rate_cg_param	134
create_rate_ct_param	135
create_rate_gt_param	136
create_rln_clock_model	137
create_scale_param	138
create_screenlog	139
create_screenlog_xml	140
create_sigma_param	141
create_site_model	142
create_site_models	143
create_site_models_from_names	144
create_site_model_from_name	145
create_strict_clock_model	146
create_s_param	147
create_temp_screenlog_filename	148
create_temp_tracelog_filename	149
create_temp_treelog_filename	149
create_test_inference_model	150
create_test_mcmc	151
create_test_ns_inference_model	152
create_test_ns_mcmc	153
create_test_screenlog	154
create_test_tracelog	155
create_test_treelog	156
create_tn93_site_model	156
create_tracelog	158
create_tracelog_xml	158
create_trait_set_string	159
create_treelog	160
create_treelog_xml	160
create_tree_prior	161
create_tree_priors	162

create_uniform_distr	163
create_xml_declaration	164
create_yule_tree_prior	165
default_parameters_doc	166
default_params_doc	166
distr_to_xml	172
distr_to_xml_beta	173
distr_to_xml_exp	173
distr_to_xml_gamma	174
distr_to_xml_inv_gamma	174
distr_to_xml_laplace	175
distr_to_xml_log_normal	175
distr_to_xml_normal	176
distr_to_xml_one_div_x	176
distr_to_xml_poisson	177
distr_to_xml_uniform	177
extract_xml_loggers_from_lines	178
extract_xml_operators_from_lines	178
extract_xml_section_from_lines	179
fasta_file_to_sequences	179
find_clock_model	180
find_first_regex_line	180
find_first_xml_opening_tag_line	181
find_last_regex_line	181
find_last_xml_closing_tag_line	182
freq_equilibrium_to_xml	182
gamma_site_models_to_xml_prior_distr	183
gamma_site_model_to_xml_prior_distr	183
gamma_site_model_to_xml_state	184
get_alignment_id	184
get_alignment_ids	185
get_alignment_ids_from_fasta_filenames	186
get_beautier_path	187
get_beautier_paths	188
get_clock_models_ids	189
get_clock_model_name	189
get_clock_model_names	190
get_crown_age	191
get_distr_names	191
get_distr_n_params	192
get_fasta_filename	193
get_file_base_sans_ext	194
get_freq_equilibrium_names	194
get_gamma_site_model_n_distrs	195
get_gamma_site_model_n_params	196
get_has_non_strict_clock_model	197
get_inference_model_filenames	197
get_log_modes	198

get_log_sorts	199
get_mcmc_filenames	199
get_n_taxa	200
get_operator_id_pre	201
get_param_names	201
get_remove_dir_fun	202
get_remove_hex_fun	203
get_replace_dir_fun	203
get_site_models_n_distrs	204
get_site_models_n_params	204
get_site_model_names	205
get_site_model_n_distrs	206
get_site_model_n_params	207
get_taxa_names	208
get_tree_priors_n_distrs	208
get_tree_priors_n_params	209
get_tree_prior_names	210
get_tree_prior_n_distrs	211
get_tree_prior_n_params	212
get_xml_closing_tag	213
get_xml_opening_tag	214
gtr_site_model_to_xml_subst_model	214
has_mrca_prior	215
has_xml_closing_tag	216
has_xml_opening_tag	217
has_xml_short_closing_tag	217
hky_site_model_to_xml_subst_model	218
indent	218
init_bd_tree_prior	219
init_beta_distr	220
init_ccp_tree_prior	220
init_cep_tree_prior	221
init_clock_models	221
init_distr	222
init_exp_distr	222
init_gamma_distr	223
init_gamma_site_model	224
init_gtr_site_model	225
init_hky_site_model	225
init_inference_model	226
init_inv_gamma_distr	227
init_jc69_site_model	227
init_laplace_distr	228
init_log_normal_distr	229
init_mrca_prior	229
init_mrca_priors	230
init_normal_distr	231
init_one_div_x_distr	231

<code>init_param</code>	232
<code>init_poisson_distr</code>	232
<code>init_rln_clock_model</code>	233
<code>init_site_models</code>	234
<code>init_strict_clock_model</code>	234
<code>init_tn93_site_model</code>	235
<code>init_tree_priors</code>	236
<code>init_uniform_distr</code>	236
<code>init_yule_tree_prior</code>	237
<code>interspace</code>	237
<code>is_alpha_param</code>	238
<code>is_bd_tree_prior</code>	239
<code>is_beauti_options</code>	240
<code>is_beta_distr</code>	241
<code>is_beta_param</code>	242
<code>is_cbs_tree_prior</code>	243
<code>is_ccp_tree_prior</code>	244
<code>is_cep_tree_prior</code>	245
<code>is_clock_model</code>	246
<code>is_clock_model_name</code>	247
<code>is_clock_rate_param</code>	247
<code>is_default_mcmc</code>	248
<code>is_distr</code>	249
<code>is_distr_name</code>	250
<code>is_exp_distr</code>	251
<code>is_freq_equilibrium_name</code>	252
<code>is_gamma_distr</code>	253
<code>is_gamma_site_model</code>	254
<code>is_gtr_site_model</code>	254
<code>is_hky_site_model</code>	255
<code>is_id</code>	256
<code>is_inference_model</code>	257
<code>is_init_bd_tree_prior</code>	257
<code>is_init_beta_distr</code>	258
<code>is_init_cbs_tree_prior</code>	258
<code>is_init_ccp_tree_prior</code>	259
<code>is_init_cep_tree_prior</code>	259
<code>is_init_clock_model</code>	260
<code>is_init_distr</code>	261
<code>is_init_exp_distr</code>	261
<code>is_init_gamma_distr</code>	262
<code>is_init_gamma_site_model</code>	262
<code>is_init_gtr_site_model</code>	263
<code>is_init_hky_site_model</code>	263
<code>is_init_inv_gamma_distr</code>	264
<code>is_init_jc69_site_model</code>	265
<code>is_init_laplace_distr</code>	266
<code>is_init_log_normal_distr</code>	266

is_init_mrca_prior	267
is_init_normal_distr	267
is_init_one_div_x_distr	268
is_init_param	268
is_init_poisson_distr	269
is_init_rln_clock_model	269
is_init_site_model	270
is_init_strict_clock_model	270
is_init_tn93_site_model	271
is_init_tree_prior	271
is_init_uniform_distr	272
is_init_yule_tree_prior	273
is_inv_gamma_distr	273
is_in_patterns	274
is_jc69_site_model	275
is_kappa_1_param	276
is_kappa_2_param	277
is_lambda_param	278
is_laplace_distr	279
is_log_normal_distr	280
is_mcmc	281
is_mcmc_nested_sampling	282
is_mean_param	283
is_mrca_align_ids_in_fastas	284
is_mrca_align_id_in_fasta	284
is_mrca_prior	285
is_mrca_prior_with_distr	286
is_mu_param	286
is_m_param	287
is_normal_distr	288
is_one_bool	289
is_one_div_x_distr	290
is_one_double	291
is_one_int	291
is_one_na	292
is_param	293
is_param_name	294
is_phylo	295
is_poisson_distr	296
is_rate_ac_param	297
is_rate_ag_param	298
is_rate_at_param	299
is_rate_cg_param	300
is_rate_ct_param	301
is_rate_gt_param	303
is_rln_clock_model	304
is_scale_param	305
is_sigma_param	306

is_site_model	307
is_site_model_name	308
is_strict_clock_model	308
is_s_param	309
is_tn93_site_model	310
is_tree_prior	311
is_tree_prior_name	312
is_uniform_distr	313
is_xml	314
is_yule_tree_prior	314
jc69_site_model_to_xml_subst_model	315
mcmc_to_xml_run	316
mcmc_to_xml_run_default	316
mcmc_to_xml_run_nested_sampling	317
mrca_priors_to_xml_prior_distr	318
mrca_priors_to_xml_state	319
mrca_priors_to_xml_tracelog	320
mrca_prior_to_xml_lh_distr	321
mrca_prior_to_xml_prior_distr	322
mrca_prior_to_xml_state	323
mrca_prior_to_xml_taxonset	323
mrca_prior_to_xml_tracelog	324
no_taxa_to_xml_tree	325
parameter_to_xml	325
parameter_to_xml_alpha	326
parameter_to_xml_beta	327
parameter_to_xml_clock_rate	327
parameter_to_xml_kappa_1	328
parameter_to_xml_kappa_2	328
parameter_to_xml_lambda	329
parameter_to_xml_m	329
parameter_to_xml_mean	330
parameter_to_xml_mu	330
parameter_to_xml_rate_ac	331
parameter_to_xml_rate_ag	331
parameter_to_xml_rate_at	332
parameter_to_xml_rate_cg	333
parameter_to_xml_rate_ct	333
parameter_to_xml_rate_gt	334
parameter_to_xml_s	334
parameter_to_xml_scale	335
parameter_to_xml_sigma	336
phylo_to_xml_state	336
remove_empty_lines	337
remove_multiline	337
rename_inference_model_filenames	338
rename_mcmc_filenames	339
rln_clock_model_to_xml_mean_rate_prior	341

rnd_phylo_to_xml_init	341
site_models_to_xml_operators	342
site_models_to_xml_prior_distr	343
site_models_to_xml_state	343
site_models_to_xml_tracelog	344
site_model_to_xml_lh_distr	345
site_model_to_xml_operators	345
site_model_to_xml_prior_distr	346
site_model_to_xml_state	347
site_model_to_xml_subst_model	347
site_model_to_xml_tracelog	348
taxa_to_xml_tree	348
tipdate_taxa_to_xml_tree	349
tn93_site_model_to_xml_subst_model	350
tree_models_to_xml_tracelog	350
tree_priors_to_xml_operators	351
tree_priors_to_xml_prior_distr	352
tree_priors_to_xml_state	352
tree_priors_to_xml_tracelog	353
tree_prior_to_xml_operators	354
tree_prior_to_xml_prior_distr	354
tree_prior_to_xml_state	355
tree_prior_to_xml_tracelog	356
yule_tree_prior_to_xml_prior_distr	356

Index **358**

are_clock_models	<i>Determine if x consists out of clock_models objects</i>
------------------	--

Description

Determine if x consists out of clock_models objects

Usage

are_clock_models(x)

Arguments

x the object to check if it consists out of clock_models objects

Value

TRUE if x, or all elements of x, are clock_model objects

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

rln_clock_model <- create_rln_clock_model()
strict_clock_model <- create_strict_clock_model()
both_clock_models <- list(rln_clock_model, strict_clock_model)
expect_true(are_clock_models(rln_clock_model))
expect_true(are_clock_models(strict_clock_model))
expect_true(are_clock_models(both_clock_models))

expect_false(are_clock_models(NA))
expect_false(are_clock_models(NULL))
expect_false(are_clock_models("nonsense"))
expect_false(are_clock_models(create_jc69_site_model()))
```

are_equal_mcmc	<i>Determine if two MCMCs are equal.</i>
----------------	--

Description

Will [stop](#) if the arguments are not MCMCs.

Usage

```
are_equal_mcmc(mcmc_1, mcmc_2)
```

Arguments

mcmc_1	an MCMC, as created by create_mcmc
mcmc_2	an MCMC, as created by create_mcmc

Value

TRUE if the two MCMCs are equal

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_mcmc](#) to create an MCMC

Examples

```
library(testthat)

mcmc_1 <- create_mcmc(chain_length = 1000)
mcmc_2 <- create_mcmc(chain_length = 314)
expect_true(are_equal_mcmcs(mcmc_1, mcmc_1))
expect_false(are_equal_mcmcs(mcmc_1, mcmc_2))
```

are_equal_screenlogs *Determine if two screenlogs are equal.*

Description

Will [stop](#) if the arguments are not screenlogs.

Usage

```
are_equal_screenlogs(screenlog_1, screenlog_2)
```

Arguments

screenlog_1 an screenlog, as created by [create_screenlog](#)
screenlog_2 an screenlog, as created by [create_screenlog](#)

Value

TRUE if the two screenlogs are equal

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_screenlog](#) to create an screenlog

Examples

```
library(testthat)

screenlog_1 <- create_screenlog(log_every = 1000)
screenlog_2 <- create_screenlog(log_every = 314)
expect_true(are_equal_screenlogs(screenlog_1, screenlog_1))
expect_false(are_equal_screenlogs(screenlog_1, screenlog_2))
```

are_equal_tracelogs *Determine if two tracelogs are equal.*

Description

Will [stop](#) if the arguments are not tracelogs.

Usage

```
are_equal_tracelogs(tracelog_1, tracelog_2)
```

Arguments

tracelog_1 an tracelog, as created by [create_tracelog](#)

tracelog_2 an tracelog, as created by [create_tracelog](#)

Value

TRUE if the two tracelogs are equal

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tracelog](#) to create an tracelog

Examples

```
library(testthat)

tracelog_1 <- create_tracelog(log_every = 1000)
tracelog_2 <- create_tracelog(log_every = 314)
expect_true(are_equal_tracelogs(tracelog_1, tracelog_1))
expect_false(are_equal_tracelogs(tracelog_1, tracelog_2))
```

are_equal_treelogs *Determine if two treelogs are equal.*

Description

Will [stop](#) if the arguments are not treelogs.

Usage

```
are_equal_treelogs(treelog_1, treelog_2)
```

Arguments

treelog_1 an treelog, as created by [create_treelog](#)

treelog_2 an treelog, as created by [create_treelog](#)

Value

TRUE if the two treelogs are equal

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_treelog](#) to create an treelog

Examples

```
library(testthat)

treelog_1 <- create_treelog(log_every = 1000)
treelog_2 <- create_treelog(log_every = 314)
expect_true(are_equal_treelogs(treelog_1, treelog_1))
expect_false(are_equal_treelogs(treelog_1, treelog_2))
```

are_equal_xml_files *Determine if XML files result in equal trees*

Description

Determine if XML files result in equal trees

Usage

```
are_equal_xml_files(filename_1, filename_2, section)
```

Arguments

filename_1	name of a first XML file
filename_2	name of a second XML file
section	name of an XML section. Assumes that there is one line that starts with <section (excluding whitespace) and one line that is </section> (also excluding whitespace)

Value

TRUE if the two sections of the XML files are equal, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

to check for equivalence, use [are_equivalent_xml_files](#)

are_equal_xml_lines *Determine if XML lines result in equal trees*

Description

Determine if XML lines result in equal trees

Usage

```
are_equal_xml_lines(lines_1, lines_2, section)
```


Arguments

lines_1	lines of a first XML file
lines_2	lines of a second XML file
section	name of an XML section. Assumes that there is one line that starts with <section (excluding whitespace) and one line that is </section> (also excluding whitespace)

Value

TRUE if the two sections of the XML files are equal, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_equivalent_xml_files

Determine if XML files result in equivalent trees

Description

Determine if XML files result in equivalent trees

Usage

```
are_equivalent_xml_files(filename_1, filename_2, section = NA)
```

Arguments

filename_1	name of a first XML file
filename_2	name of a second XML file
section	the name of the XML section, use NA to check the whole file

Value

TRUE if the two XML files result in equivalent trees, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

to check for equality, use `are_equal_xml_files`

are_equivalent_xml_lines

Determine if XML lines result in equivalent trees

Description

Determine if XML lines result in equivalent trees

Usage

```
are_equivalent_xml_lines(lines_1, lines_2, section = NA, verbose = FALSE)
```

Arguments

lines_1	lines of a first XML file
lines_2	lines of a second XML file
section	the name of the XML section
verbose	if TRUE, additional information is displayed, that is potentially useful in debugging

Value

TRUE if the two XML lines result in equivalent trees, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_equivalent_xml_lines_all

Determine if XML lines result in equivalent trees

Description

Determine if XML lines result in equivalent trees

Usage

```
are_equivalent_xml_lines_all(lines_1, lines_2, verbose = FALSE)
```

Arguments

lines_1	lines of a first XML file
lines_2	lines of a second XML file
verbose	if TRUE, additional information is displayed, that is potentially useful in debugging

Value

TRUE if the two XML lines result in equivalent trees, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_equivalent_xml_lines_loggers

Determine if XML operator lines result in equivalent trees

Description

Determine if XML operator lines result in equivalent trees

Usage

are_equivalent_xml_lines_loggers(lines_1, lines_2, verbose = FALSE)

Arguments

lines_1	lines of a first XML file
lines_2	lines of a second XML file
verbose	if TRUE, additional information is displayed, that is potentially useful in debugging

Value

TRUE if the two XML lines result in equivalent trees, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_equivalent_xml_lines_operators

Determine if XML operator lines result in equivalent trees

Description

Determine if XML operator lines result in equivalent trees

Usage

```
are_equivalent_xml_lines_operators(lines_1, lines_2, verbose = FALSE)
```

Arguments

lines_1	lines of a first XML file
lines_2	lines of a second XML file
verbose	if TRUE, additional information is displayed, that is potentially useful in debugging

Value

TRUE if the two XML lines result in equivalent trees, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_equivalent_xml_lines_section

Determine if XML lines result in equivalent trees

Description

Determine if XML lines result in equivalent trees

Usage

```
are_equivalent_xml_lines_section(lines_1, lines_2, section, verbose = FALSE)
```

Arguments

lines_1	lines of a first XML file
lines_2	lines of a second XML file
section	the name of the XML section
verbose	if TRUE, additional information is displayed, that is potentially useful in debugging

Value

TRUE if the two XML lines result in equivalent trees, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_fasta_filenames *Checks if all filenames have a FASTA filename extension*

Description

Checks if all filenames have a FASTA filename extension

Usage

```
are_fasta_filenames(filenames)
```

Arguments

filenames filenames

Value

TRUE if all filenames have a FASTA filename extension

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(are_fasta_filenames("1.fas"))
expect_true(are_fasta_filenames("1.fasta"))
expect_true(are_fasta_filenames("1.FAS"))
expect_true(are_fasta_filenames("1.FASTA"))
expect_true(are_fasta_filenames(c("1.fas", "2.fas")))

expect_false(are_fasta_filenames(""))
expect_false(are_fasta_filenames(NA))
expect_false(are_fasta_filenames(NULL))
expect_false(are_fasta_filenames(Inf))
expect_false(are_fasta_filenames("1.fasX"))

expect_false(are_fasta_filenames(c("1.fas", "2.exe")))
expect_false(are_fasta_filenames(c("1.bat", "2.exe")))
```

are_ids *Determine if x consists out of IDs*

Description

Determine if x consists out of IDs

Usage

```
are_ids(x)
```

Arguments

x the object to check if it consists out of IDs

Value

TRUE if x, or all elements of x, are IDs

Author(s)

Richèl J.C. Bilderbeek

See Also

to check one ID, use [is_id](#)

are_init_clock_models *Determine if x consists out of initialized clock_models objects*

Description

Determine if x consists out of initialized clock_models objects

Usage

```
are_init_clock_models(x)
```

Arguments

x the object to check if it consists out of initialized clock_models objects

Value

TRUE if x, or all elements of x, are initialized clock_model objects

Author(s)

Richèl J.C. Bilderbeek

are_init_mrca_priors *Determine if x consists out of initialized MRCA priors*

Description

Determine if x consists out of initialized MRCA priors

Usage

```
are_init_mrca_priors(x)
```

Arguments

x the object to check if it consists out of initialized MRCA priors

Value

TRUE if x, or all elements of x, are initialized MRCA priors

Author(s)

Richèl J.C. Bilderbeek

are_init_site_models *Determine if x consists out of initialized site_models objects*

Description

Determine if x consists out of initialized site_models objects

Usage

```
are_init_site_models(x)
```

Arguments

x the object to check if it consists out of initialized site_models objects

Value

TRUE if x, or all elements of x, are initialized site_model objects

Author(s)

Richèl J.C. Bilderbeek

are_init_tree_priors *Determine if x consists out of initialized tree_priors objects*

Description

Determine if x consists out of initialized tree_priors objects

Usage

```
are_init_tree_priors(x)
```

Arguments

x the object to check if it consists out of initialized tree_priors objects

Value

TRUE if x, or all elements of x, are initialized tree_prior objects

Author(s)

Richèl J.C. Bilderbeek

are_mrca_align_ids_in_fasta
Determine if the MRCA priors' alignment IDs are present in the FASTA files

Description

Determine if the MRCA priors' alignment IDs are present in the FASTA files

Usage

```
are_mrca_align_ids_in_fasta(mrca_prior, fasta_filename)
```

Arguments

mrca_prior a Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)
fasta_filename a FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

Value

TRUE if all the MRCA priors' alignment IDs are present in the FASTA files. Returns FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_mrca_priors *Determine if x consists out of MRCA priors*

Description

Determine if x consists out of MRCA priors

Usage

```
are_mrca_priors(mrca_priors)
```

Arguments

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

Value

TRUE if x, or all elements of x, are MRCA priors. Returns FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_mrca_taxon_names_in_fasta
Determine if the MRCA priors' taxa names are present in the FASTA files

Description

Determine if the MRCA priors' taxa names are present in the FASTA files

Usage

```
are_mrca_taxon_names_in_fasta(mrca_prior, fasta_filename)
```

Arguments

mrca_prior a Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)
 fasta_filename a FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

Value

TRUE if the MRCA priors' taxa names are present in the FASTA files. FALSE otherwise.

Author(s)

Richèl J.C. Bilderbeek

are_rln_clock_models *Are the clock models Relaxed Log-Normal clock models?*

Description

Are the clock models Relaxed Log-Normal clock models?

Usage

```
are_rln_clock_models(clock_models)
```

Arguments

clock_models a list of one or more clock models, as returned by [create_clock_model](#)

Value

vector of booleans with the same length as the number of clock models in clock_models. Each nth element is TRUE if the nth element in clock_models is a relaxed log-normal clock model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

are_site_models *Determine if x consists out of site_models objects*

Description

Determine if x consists out of site_models objects

Usage

```
are_site_models(x)
```

Arguments

x the object to check if it consists out of site_models objects

Value

TRUE if x, or all elements of x, are site_model objects

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a site model

Examples

```
jc69_site_model <- create_jc69_site_model()
gtr_site_model <- create_gtr_site_model()
both_site_models <- list(jc69_site_model, gtr_site_model)
testit::assert(are_site_models(jc69_site_model))
testit::assert(are_site_models(gtr_site_model))
testit::assert(are_site_models(both_site_models))
```

are_tree_priors *Determine if x consists out of tree_priors objects*

Description

Determine if x consists out of tree_priors objects

Usage

```
are_tree_priors(x)
```

Arguments

x the object to check if it consists out of tree_priors objects

Value

TRUE if x, or all elements of x, are tree_prior objects

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_yule_tree_prior](#) to create a Yule tree prior

Examples

```
library(testthat)

yule_tree_prior <- create_yule_tree_prior()
bd_tree_prior <- create_bd_tree_prior()
both_tree_priors <- list(yule_tree_prior, bd_tree_prior)
expect_true(are_tree_priors(yule_tree_prior))
expect_true(are_tree_priors(bd_tree_prior))
expect_true(are_tree_priors(both_tree_priors))
```

bd_tree_prior_to_xml_prior_distr

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Birth-Death tree prior

Description

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Birth-Death tree prior

Usage

```
bd_tree_prior_to_xml_prior_distr(bd_tree_prior)
```

Arguments

bd_tree_prior a Birth-Death tree prior, as created by [create_bd_tree_prior](#)

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
# <distribution id="likelihood" ...>
#   </distribution>
# </distribution>
```

beautier

beautier: *A package to create a BEAST2 input file.*

Description

beautier allows to create a BEAST2 input file, using an R interface. beautier closely follows the interface of BEAUti 2, a GUI tool bundled with BEAST2, including its default settings.

Author(s)

Richèl J.C. Bilderbeek

See Also

These are packages associated with beautier:

- The package `beastier` can run BEAST2 from R
- The package `tracerer` can parse BEAST2 output files from R
- The package `mauricer` manages BEAST2 packages from R
- The package `babette` combines the functionality of `beautier`, `beastier`, `mauricer` and `tracerer` into a single workflow

Examples

```
# Get an example FASTA file
input_filename <- get_fasta_filename()

# The file created by beautier, a BEAST2 input file
output_filename <- tempfile()

# Use the default BEAUti settings to create a BEAST2 input file
create_beast2_input_file_from_model(
  input_filename,
  output_filename,
  inference_model = create_inference_model()
)
testthat::expect_true(file.exists(output_filename))
```

`cbs_tree_prior_to_xml_prior_distr`

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Birth-Death tree prior

Description

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Birth-Death tree prior

Usage

```
cbs_tree_prior_to_xml_prior_distr(cbs_tree_prior)
```

Arguments

`cbs_tree_prior` a Coalescent Bayesian Skyline tree prior, as returned by [create_cbs_tree_prior](#)

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
# <distribution id="likelihood" ...>
# </distribution>
# </distribution>
```

```
ccp_tree_prior_to_xml_prior_distr
```

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Coalescent Constant Population tree prior

Description

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Coalescent Constant Population tree prior

Usage

```
ccp_tree_prior_to_xml_prior_distr(ccp_tree_prior)
```

Arguments

`ccp_tree_prior` a Coalescent Constant Population tree prior, as returned by [create_ccp_tree_prior](#)

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
# <distribution id="likelihood" ...>
# </distribution>
# </distribution>
```

cep_tree_prior_to_xml_prior_distr

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Coalescent Exponential Population tree prior

Description

Creates the tree prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Coalescent Exponential Population tree prior

Usage

```
cep_tree_prior_to_xml_prior_distr(cep_tree_prior)
```

Arguments

cep_tree_prior a Coalescent Exponential Population tree prior, as returned by [create_cep_tree_prior](#)

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
# <distribution id="likelihood" ...>
# </distribution>
# </distribution>
```

check_alignment_id *Check if the alignment_id is valid.*

Description

Will [stop](#) if not.

Usage

```
check_alignment_id(alignment_id)
```

Arguments

alignment_id ID of the alignment, as returned by [get_alignment_id](#). Keep at NA to have it initialized automatically

Examples

```
library(testthat)

# Path need not exist, use UNIX path as example
created <- get_alignment_id("/home/homer/anthus_aco_sub.fas")
expected <- "anthus_aco_sub"
expect_equal(created, expected)
expect_silent(check_alignment_id(created))
```

check_beauti_options *Check if the beauti_options is a valid beauti_options object.*

Description

Calls [stop](#) if the beauti_options object is invalid

Usage

```
check_beauti_options(beauti_options)
```

Arguments

beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_beauti_options](#) to create a valid BEAUti options setup

Examples

```
testthat::expect_silent(check_beauti_options(create_beauti_options()))  
  
# Must stop on nonsense  
testthat::expect_error(check_beauti_options(beauti_options = "nonsense"))  
testthat::expect_error(check_beauti_options(beauti_options = NULL))  
testthat::expect_error(check_beauti_options(beauti_options = NA))
```

check_clock_model	<i>Check if the clock model is a valid clock model.</i>
-------------------	---

Description

Calls stop if the clock model is invalid

Usage

```
check_clock_model(clock_model)
```

Arguments

clock_model a clock model, as returned by [create_clock_model](#)

Value

TRUE if clock_model is a valid clock model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_model](#) to create a valid clock model

Examples

```
testthat::expect_silent(check_clock_model(create_strict_clock_model()))
testthat::expect_silent(check_clock_model(create_rln_clock_model()))

# Must stop on non-clock models
testthat::expect_error(check_clock_model(clock_model = "nonsense"))
testthat::expect_error(check_clock_model(clock_model = NULL))
testthat::expect_error(check_clock_model(clock_model = NA))
```

check_clock_models *Check if the object is a list of one or more clock models.*

Description

Will [stop](#) if the object is not a list of one or more clock models.

Usage

```
check_clock_models(clock_models)
```

Arguments

clock_models the object to be checked if it is a list of one or more valid clock models

Value

nothing. Will [stop](#) if the object is not a list of one or more clock models.

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_model](#) to create a valid clock model

Examples

```
testthat::expect_silent(check_clock_models(create_strict_clock_model()))
testthat::expect_silent(
  check_clock_models(list(create_strict_clock_model()))
)
testthat::expect_silent(
  check_clock_models(
    list(create_strict_clock_model(), create_rln_clock_model())
  )
)

testthat::expect_error(check_clock_models("nonsense"))
```

```
testthat::expect_error(check_clock_models(3.14))
testthat::expect_error(check_clock_models(42))
testthat::expect_error(check_clock_models(NA))
testthat::expect_error(check_clock_models(NULL))
```

check_file_and_model_agree

Checks if the input FASTA file and the inference model agree.

Description

Will [stop](#) if not

Usage

```
check_file_and_model_agree(input_filename, inference_model)
```

Arguments

`input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

`inference_model`

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

check_file_exists

Function to check if a file exists. Calls stop if the file is absent

Description

Function to check if a file exists. Calls stop if the file is absent

Usage

```
check_file_exists(filename, filename_description = NA)
```

Arguments

`filename` name of the file

`filename_description`

description of the filename

Value

nothing. Will stop if the file is absent, with a proper error message

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_silent(
  check_file_exists(
    get_beautier_path("anthus_aco_sub.fas")
  )
)

# Minimal use
expect_error(
  check_file_exists("absent"),
  "File not found. Could not find file with path 'absent'"
)

# Add a description
absent_filename <- "absent"
expect_error(
  check_file_exists(absent_filename, "absent_filename"),
  "File 'absent_filename' not found. Could not find file with path 'absent'"
)
```

check_gamma_site_model

Checks if the parameter is a valid gamma site model

Description

Checks if the parameter is a valid gamma site model

Usage

```
check_gamma_site_model(gamma_site_model)
```

Arguments

gamma_site_model
a site model's gamma site model, as returned by [create_gamma_site_model](#)

Value

nothing. Will call stop if the argument is not a valid gamma site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_silent(
  check_gamma_site_model(
    create_gamma_site_model()
  )
)
expect_error(
  check_gamma_site_model(
    "not a gamma site model"
  )
)
```

check_gamma_site_model_names

Checks if the gamma site model has the right list elements' names

Description

Checks if the gamma site model has the right list elements' names

Usage

```
check_gamma_site_model_names(gamma_site_model)
```

Arguments

gamma_site_model
a site model's gamma site model, as returned by [create_gamma_site_model](#)

Value

nothing. Will call stop if the argument is not a valid gamma site model

Author(s)

Richèl J.C. Bilderbeek

check_gtr_site_model *Check if the gtr_site_model is a valid GTR nucleotide substitution model.*

Description

Use [create_gtr_site_model](#) to create a valid GTR nucleotide substitution model.

Usage

```
check_gtr_site_model(gtr_site_model)
```

Arguments

gtr_site_model a GTR site model, as returned by [create_gtr_site_model](#)

Examples

```
library(testthat)

expect_silent(check_gtr_site_model(create_gtr_site_model()))

expect_error(check_gtr_site_model("nonsense"))
expect_error(check_gtr_site_model(NA))
expect_error(check_gtr_site_model(NULL))
expect_error(check_gtr_site_model(""))
expect_error(check_gtr_site_model(c()))
```

check_gtr_site_model_names *Check if the gtr_site_model has the list elements of a valid gtr_site_model object.*

Description

Calls stop if an element is missing

Usage

```
check_gtr_site_model_names(gtr_site_model)
```

Arguments

gtr_site_model a GTR site model, as returned by [create_gtr_site_model](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_gtr_site_model](#) to create a valid gtr_site_model

check_inference_model *Check if the supplied object is a valid Bayesian phylogenetic inference model.*

Description

Calls stop if the supplied object is not a valid Bayesian phylogenetic inference model.

Usage

```
check_inference_model(inference_model)
```

Arguments

inference_model

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_inference_model](#) to create a valid Bayesian phylogenetic inference model

Examples

```
testthat::expect_silent(check_inference_model(create_inference_model()))

# Must stop on non-MCMCs
testthat::expect_error(check_inference_model(inference_model = "nonsense"))
testthat::expect_error(check_inference_model(inference_model = NULL))
testthat::expect_error(check_inference_model(inference_model = NA))
```

check_inference_models

Check if the inference_model is a valid BEAUti inference model.

Description

Calls stop if not.

Usage

```
check_inference_models(inference_models)
```

Arguments

inference_models
a list of one or more inference models, as can be created by [create_inference_model](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_inference_model](#) to create a valid BEAST2 options object

Examples

```
testthat::expect_silent(  
  check_inference_models(  
    list(create_inference_model()  
  )  
)  
)  
  
# Must stop on nonsense  
testthat::expect_error(check_inference_models("nonsense"))  
testthat::expect_error(check_inference_models(NULL))  
testthat::expect_error(check_inference_models(NA))
```

check_is_monophyletic *Check if is_monophyletic has a valid value.*

Description

Will [stop](#) if not.

Usage

```
check_is_monophyletic(is_monophyletic)
```

Arguments

is_monophyletic
boolean to indicate monophyly is assumed in a Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)

check_log_mode *Check if the supplied mode is a valid logging mode.*

Description

Check if the supplied mode is a valid logging mode.

Usage

```
check_log_mode(mode)
```

Arguments

mode
mode how to log. Valid are tree, autodetect and compound

check_log_sort *Check if the supplied sort is a valid logging sorting option.*

Description

Check if the supplied sort is a valid logging sorting option.

Usage

```
check_log_sort(sort)
```

Arguments

sort
how to sort the entries in a log. Valid are smart, none and alphabetic

`check_mcmc`*Check if the MCMC is a valid MCMC object.*

Description

Calls stop if the MCMC is invalid

Usage

```
check_mcmc(mcmc)
```

Arguments

`mcmc` one MCMC. Use [create_mcmc](#) to create an MCMC. Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_mcmc](#) to create a valid MCMC

Examples

```
library(testthat)

expect_silent(check_mcmc(create_mcmc()))

# Must stop on non-MCMCs
expect_error(check_mcmc(mcmc = "nonsense"))
expect_error(check_mcmc(mcmc = NULL))
expect_error(check_mcmc(mcmc = NA))
```

`check_mcmc_list_element_names`*Check if the MCMC has the list elements of a valid MCMC object.*

Description

Calls stop if an element is missing

Usage

```
check_mcmc_list_element_names(mcmc)
```

Arguments

`mcmc` one MCMC. Use [create_mcmc](#) to create an MCMC. Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_mcmc](#) to create a valid MCMC

`check_mcmc_values`*Check if the MCMC has the list elements with valid values for being a valid MCMC object.*

Description

Calls stop if a value is invalid

Usage

```
check_mcmc_values(mcmc)
```

Arguments

mcmc one MCMC. Use [create_mcmc](#) to create an MCMC. Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_mcmc](#) to create a valid MCMC

check_mrca_prior *Check if the MRCA prior is a valid MRCA prior.*

Description

Calls stop if the MRCA prior is invalid.

Usage

```
check_mrca_prior(mrca_prior)
```

Arguments

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

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_mrca_prior](#) to create a valid MRCA prior

Examples

```

fasta_filename <- get_beautier_path("anthus_aco.fas")
mrca_prior <- create_mrca_prior(
  alignment_id = get_alignment_id(fasta_filename = fasta_filename),
  taxa_names = get_taxa_names(filename = fasta_filename)
)
mrca_prior <- create_mrca_prior(
  alignment_id = get_alignment_id(fasta_filename = fasta_filename),
  taxa_names = get_taxa_names(filename = fasta_filename)
)
testthat::expect_silent(check_mrca_prior(mrca_prior))

# NA is a valid MRCA prior
testthat::expect_silent(check_mrca_prior(mrca_prior = NA))

# Must stop on non-MRCA priors
testthat::expect_error(check_mrca_prior(mrca_prior = "nonsense"))
testthat::expect_error(check_mrca_prior(mrca_prior = NULL))

```

check_mrca_prior_name *Check if mrca_prior_name is a valid MRCA prior name.*

Description

A valid MRCA prior name is either **NA** or one character string. Will **stop** if not.

Usage

```
check_mrca_prior_name(mrca_prior_name)
```

Arguments

mrca_prior_name

the unique name of the MRCA prior, for example a genus, family, order or even class name. Leave at **NA** to have it named automatically.

check_mrca_prior_names

Check if the MRCA prior, which is a list, has all the named elements.

Description

Calls **stop** if not.

Usage

```
check_mrca_prior_names(mrca_prior)
```

Arguments

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

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [check_mrca_prior](#) to check the entire MRCA prior

check_mrca_prior_taxa_names

Check the MRCA prior's taxon names are valid.

Description

Will [stop](#) if not.

Usage

```
check_mrca_prior_taxa_names(taxa_names)
```

Arguments

taxa_names names of the taxa, as returned by [get_taxa_names](#). Keep at NA to have it initialized automatically, using all taxa in the alignment

check_ns_mcmc

Check if this an MCMC that uses Nested Sampling to estimate a marginal likelihood.

Description

Will [stop](#) if not, else will do nothing

Usage

```
check_ns_mcmc(mcmc)
```

Arguments

mcmc one MCMC. Use [create_mcmc](#) to create an MCMC. Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Author(s)

Richèl J.C. Bilderbeek

See Also

use [create_ns_mcmc](#) to create an MCMC that uses Nested Sampling to estimate a marginal likelihood

check_param *Check if the parameter is a valid parameter*

Description

Calls stop if the parameter is invalid

Usage

```
check_param(param)
```

Arguments

param a parameter, as can be created by [create_param](#).

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_param](#) to create a valid parameter

Examples

```
library(testthat)

expect_silent(check_param(create_alpha_param()))
expect_silent(check_param(create_beta_param()))

# List of two parameters is not a/one parameter
expect_error(
  check_param(
    list(create_alpha_param(), create_beta_param())
  )
)

# Must stop on non-parameters
expect_error(check_param("nonsense"))
expect_error(check_param(NULL))
expect_error(check_param(NA))
expect_error(check_param(""))
expect_error(check_param(c()))
```

check_param_names	<i>Check if the param has the list elements of a valid param object.</i>
-------------------	--

Description

Calls stop if an element is missing

Usage

```
check_param_names(param)
```

Arguments

param a parameter, as can be created by [create_param](#).

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_param](#) to create a valid param

check_param_types	<i>Check if the param has the list elements of the right type for a valid param object.</i>
-------------------	---

Description

Calls stop if an element has the incorrect type

Usage

```
check_param_types(param)
```

Arguments

param a parameter, as can be created by [create_param](#).

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_param](#) to create a valid param

check_phylogeny	<i>Check if the phylogeny is a valid phylogeny object.</i>
-----------------	--

Description

Calls stop if the phylogeny is invalid

Usage

```
check_phylogeny(phylogeny)
```

Arguments

phylogeny a phylogeny of type [phylo](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [read.tree](#) to create a phylogeny

Examples

```
library(testthat)

# Must do nothing on phylogenies
phylogeny <- ape::read.tree(text = "(A:1, B:1):1;")
expect_silent(check_phylogeny(phylogeny))

# Must stop on non-phylogenies
expect_error(check_phylogeny("nonsense"))
expect_error(check_phylogeny(NULL))
expect_error(check_phylogeny(NA))
expect_error(check_phylogeny(c()))
expect_error(check_phylogeny(c(3, 1, 4)))
```

check_rename_fun	<i>Check if the rename function is a valid filename rename function</i>
------------------	---

Description

Will [stop](#) if not

Usage

```
check_rename_fun(rename_fun)
```

Arguments

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`

Author(s)

Richèl J.C. Bilderbeek

check_rln_clock_model *Check if the clock model is a valid clock model.*

Description

Calls `stop` if the clock model is invalid

Usage

```
check_rln_clock_model(clock_model)
```

Arguments

clock_model a clock model, as returned by [create_clock_model](#)

Value

TRUE if clock_model is a valid clock model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_model](#) to create a valid clock model

Examples

```
testthat::expect_silent(  
  check_rln_clock_model(create_rln_clock_model())  
)  
testthat::expect_error(  
  check_rln_clock_model(create_strict_clock_model())  
)
```

check_screenlog *Check if a screenlog is valid.*

Description

Will call `stop` if not.

Usage

```
check_screenlog(screenlog)
```

Arguments

screenlog a screenlog, as created by [create_screenlog](#)

check_screenlog_names *Check if the screenlog has the list elements of a valid screenlog object.*

Description

Calls stop if an element is missing

Usage

```
check_screenlog_names(screenlog)
```

Arguments

screenlog a screenlog, as created by [create_screenlog](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_screenlog](#) to create a valid screenlog

check_screenlog_values *Check if the screenlog has the list elements with valid values for being a valid screenlog object.*

Description

Calls stop if a value is invalid

Usage

```
check_screenlog_values(screenlog)
```

Arguments

screenlog a screenlog, as created by [create_screenlog](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_screenlog](#) to create a valid screenlog

check_site_model *Check if the site model is a valid site model*

Description

Calls stop if the site models are invalid

Usage

```
check_site_model(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a valid site model

Examples

```
library(testthat)

expect_silent(check_site_model(create_jc69_site_model()))
expect_silent(check_site_model(create_hky_site_model()))
expect_silent(check_site_model(create_tn93_site_model()))
expect_silent(check_site_model(create_gtr_site_model()))

# Can use list of one site model
expect_silent(check_site_model(list(create_jc69_site_model())))

# List of two site models is not a one site model
expect_error(
  check_site_model(
    list(create_jc69_site_model(), create_jc69_site_model())
  )
)

# Must stop on non-site models
expect_error(check_site_model("nonsense"))
expect_error(check_site_model(NULL))
expect_error(check_site_model(NA))
expect_error(check_site_model(""))
expect_error(check_site_model(c()))
```

check_site_models	<i>Check if the object is a list of one or more site models.</i>
-------------------	--

Description

Will **stop** if the object is not a list of one or more site models.

Usage

```
check_site_models(site_models)
```

Arguments

site_models the object to be checked if it is a list of one or more valid site models

Value

nothing. Will **stop** if the object is not a list of one or more site models.

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a valid site model

Examples

```
testthat::expect_silent(check_site_models(create_jc69_site_model()))
testthat::expect_silent(check_site_models(list(create_jc69_site_model())))
testthat::expect_silent(
  check_site_models(
    list(create_jc69_site_model(), create_gtr_site_model())
  )
)

testthat::expect_error(check_site_models("nonsense"))
testthat::expect_error(check_site_models(3.14))
testthat::expect_error(check_site_models(42))
testthat::expect_error(check_site_models(NA))
testthat::expect_error(check_site_models(NULL))
```

check_site_model_names

Check if the site_model has the list elements of a valid site_model object.

Description

Calls stop if an element is missing

Usage

```
check_site_model_names(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a valid site_model

check_site_model_types

Check if the site_model has the list elements of the right type for a valid site_model object.

Description

Calls stop if an element has the incorrect type

Usage

```
check_site_model_types(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a valid site_model

check_store_every

Check if store_every holds a valid value

Description

Will [stop](#) if not

Usage

```
check_store_every(store_every)
```

Arguments

store_every number of states the MCMC will process before the posterior's state will be saved to file. Use -1 or NA to use the default frequency.

`check_strict_clock_model`*Check if the clock model is a valid clock model.*

Description

Calls stop if the clock model is invalid

Usage

```
check_strict_clock_model(clock_model)
```

Arguments

`clock_model` a clock model, as returned by [create_clock_model](#)

Value

TRUE if `clock_model` is a valid clock model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_model](#) to create a valid clock model

Examples

```
testthat::expect_silent(  
  check_strict_clock_model(create_strict_clock_model())  
)  
testthat::expect_error(  
  check_strict_clock_model(create_rln_clock_model())  
)
```

check_tn93_site_model *Check if the tn93_site_model is a valid TN93 nucleotide substitution model.*

Description

Use [create_tn93_site_model](#) to create a valid TN93 nucleotide substitution model.

Usage

```
check_tn93_site_model(tn93_site_model)
```

Arguments

tn93_site_model
a TN93 site model, as returned by [create_tn93_site_model](#)

Examples

```
library(testthat)

expect_silent(check_tn93_site_model(create_tn93_site_model()))

expect_error(check_tn93_site_model("nonsense"))
expect_error(check_tn93_site_model(NA))
expect_error(check_tn93_site_model(NULL))
expect_error(check_tn93_site_model(""))
expect_error(check_tn93_site_model(c()))
```

check_tn93_site_model_names
Check if the tn93_site_model has the list elements of a valid tn93_site_model object.

Description

Calls stop if an element is missing

Usage

```
check_tn93_site_model_names(tn93_site_model)
```

Arguments

tn93_site_model
a TN93 site model, as returned by [create_tn93_site_model](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tn93_site_model](#) to create a valid tn93_site_model

check_tracelog	<i>Check if a tracelog is valid.</i>
----------------	--------------------------------------

Description

Will call [stop](#) if not.

Usage

```
check_tracelog(tracelog)
```

Arguments

tracelog a tracelog, as created by [create_tracelog](#)

check_tracelog_names	<i>Check if the tracelog has the list elements of a valid tracelog object.</i>
----------------------	--

Description

Calls [stop](#) if an element is missing

Usage

```
check_tracelog_names(tracelog)
```

Arguments

tracelog a tracelog, as created by [create_tracelog](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tracelog](#) to create a valid tracelog

check_tracelog_values *Check if the tracelog has the list elements with valid values for being a valid tracelog object.*

Description

Calls stop if a value is invalid

Usage

```
check_tracelog_values(tracelog)
```

Arguments

tracelog a tracelog, as created by [create_tracelog](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tracelog](#) to create a valid tracelog

check_treelog *Check if a treelog is valid.*

Description

Will call [stop](#) if not.

Usage

```
check_treelog(treelog)
```

Arguments

treelog a treelog, as created by [create_treelog](#)

check_treelog_names *Check if the treelog has the list elements of a valid treelog object.*

Description

Calls stop if an element is missing

Usage

```
check_treelog_names(treelog)
```

Arguments

treelog a treelog, as created by [create_treelog](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_treelog](#) to create a valid treelog

check_treelog_values *Check if the treelog has the list elements with valid values for being a valid treelog object.*

Description

Calls stop if a value is invalid

Usage

```
check_treelog_values(treelog)
```

Arguments

treelog a treelog, as created by [create_treelog](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_treelog](#) to create a valid treelog

check_tree_prior	<i>Check if the tree prior is a valid tree prior</i>
------------------	--

Description

Calls stop if the tree priors are invalid

Usage

```
check_tree_prior(tree_prior)
```

Arguments

tree_prior a tree priors, as returned by [create_tree_prior](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tree_prior](#) to create a valid tree prior

Examples

```
testthat::expect_silent(check_tree_prior(create_yule_tree_prior()))
testthat::expect_silent(check_tree_prior(create_bd_tree_prior()))
testthat::expect_silent(check_tree_prior(create_cbs_tree_prior()))
testthat::expect_silent(check_tree_prior(create_ccp_tree_prior()))
testthat::expect_silent(check_tree_prior(create_cep_tree_prior()))

# Can use list of one tree prior
testthat::expect_silent(check_tree_prior(list(create_yule_tree_prior())))

# List of two tree priors is not a/one tree prior
testthat::expect_error(
  check_tree_prior(
    list(create_yule_tree_prior(), create_yule_tree_prior())
  )
)
```

```

    )
  )

# Must stop on non-tree priors
testthat::expect_error(check_tree_prior(tree_prior = "nonsense"))
testthat::expect_error(check_tree_prior(tree_prior = NULL))
testthat::expect_error(check_tree_prior(tree_prior = NA))

```

check_tree_priors *Check if the object is a list of one or more tree priors.*

Description

Will [stop](#) if the object is not a list of one or more tree priors.

Usage

```
check_tree_priors(tree_priors)
```

Arguments

tree_priors the object to be checked if it is a list of one or more valid tree priors

Value

nothing. Will [stop](#) if the object is not a list of one or more tree priors.

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tree_prior](#) to create a valid tree prior

Examples

```

testthat::expect_silent(check_tree_priors(create_yule_tree_prior()))
testthat::expect_silent(check_tree_priors(list(create_yule_tree_prior())))
testthat::expect_silent(
  check_tree_priors(
    list(create_yule_tree_prior(), create_bd_tree_prior())
  )
)

testthat::expect_error(check_tree_priors("nonsense"))
testthat::expect_error(check_tree_priors(3.14))
testthat::expect_error(check_tree_priors(42))
testthat::expect_error(check_tree_priors(NA))
testthat::expect_error(check_tree_priors(NULL))

```

`clock_models_to_xml_operators`*Create all clock models' operators' XML text*

Description

Create all clock models' operators' XML text

Usage

```
clock_models_to_xml_operators(  
  clock_models,  
  mrca_priors = NA,  
  tipdates_filename = NA  
)
```

Arguments

`clock_models` a list of one or more clock models, as returned by [create_clock_model](#)

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

`tipdates_filename` name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

`clock_models_to_xml_prior_distr`*Represent the clock models as XML*

Description

Represent the clock models as XML

Usage

```
clock_models_to_xml_prior_distr(  
  clock_models,  
  mrca_priors = NA,  
  tipdates_filename = NA  
)
```

Arguments

`clock_models` a list of one or more clock models, as returned by [create_clock_model](#)

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

`tipdates_filename` name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">  
#   <distribution id="prior" spec="util.CompoundDistribution">  
#     HERE, where the ID of the distribution is 'prior'  
#   </distribution>  
# <distribution id="likelihood" ...>  
#   </distribution>  
# </distribution>
```

clock_models_to_xml_state

*Converts one or more clock models to the state section of the XML
as text*

Description

Converts one or more clock models to the state section of the XML as text

Usage

```
clock_models_to_xml_state(  
  clock_models,  
  mrca_priors = NA,  
  has_tip_dating = FALSE  
)
```

Arguments

`clock_models` a list of one or more clock models, as returned by [create_clock_model](#)
`mrca_priors` a list of one or more Most Recent Common Ancestor priors, as returned by [create_mrca_prior](#)
`has_tip_dating` TRUE if the user has supplied tip dates, FALSE otherwise

Value

lines of XML text, without indentation nor state tags

Author(s)

Richèl J.C. Bilderbeek

clock_models_to_xml_tracelog

Creates the clock models' XML for the tracelog section

Description

Creates the clock models' XML for the tracelog section

Usage

```
clock_models_to_xml_tracelog(clock_models, mrca_priors = NA)
```

Arguments

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

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

See Also

the complete tracelog section is created by [create_tracelog_xml](#)

Examples

```
# <logger id="tracelog" ...>
#' # Here
# </logger>
```

clock_model_to_xml_lh_distr

Converts a clock model to the branchRateModel section of the XML as text.

Description

This function will be called only if there are no MRCA priors.

Usage

```
clock_model_to_xml_lh_distr(
  clock_model,
  mrca_priors = NA,
  tipdates_filename = NA
)
```

Arguments

clock_model a clock model, as returned by [create_clock_model](#)

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

tipdates_filename name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     </distribution>
#   <distribution id="likelihood" ...>
#     HERE, where the ID of the distribution is 'likelihood'
#   </distribution>
# </distribution>
```

clock_model_to_xml_operators

Converts a clock model to the operators section of the XML as text

Description

Converts a clock model to the operators section of the XML as text

Usage

```
clock_model_to_xml_operators(clock_model, mrca_priors, tipdates_filename = NA)
```

Arguments

clock_model a clock model, as returned by [create_clock_model](#)

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

tipdates_filename name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

`clock_model_to_xml_prior_distr`*Converts a clock model to the prior section of the XML as text*

Description

Converts a clock model to the prior section of the XML as text

Usage

```
clock_model_to_xml_prior_distr(  
  clock_model,  
  mrca_priors = NA,  
  tipdates_filename = NA  
)
```

Arguments

<code>clock_model</code>	a clock model, as returned by create_clock_model
<code>mrca_priors</code>	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
<code>tipdates_filename</code>	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">  
#   <distribution id="prior" spec="util.CompoundDistribution">  
#     HERE, where the ID of the distribution is 'prior'  
#   </distribution>  
#   <distribution id="likelihood" ...>  
#     </distribution>  
# </distribution>
```

clock_model_to_xml_state

Converts a clock model to the state section of the XML as text

Description

Converts a clock model to the state section of the XML as text

Usage

```
clock_model_to_xml_state(clock_model, has_tip_dating = FALSE)
```

Arguments

clock_model a clock model, as returned by [create_clock_model](#)

has_tip_dating TRUE if the user has supplied tip dates, FALSE otherwise

Value

lines of XML text, without indentation nor state tags

Author(s)

Richèl J.C. Bilderbeek

clock_model_to_xml_tracelog

Creates the clock model's XML for the tracelog section

Description

Creates the clock model's XML for the tracelog section

Usage

```
clock_model_to_xml_tracelog(clock_model, mrca_priors = NA)
```

Arguments

clock_model a clock model, as returned by [create_clock_model](#)

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

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

See Also

all clock models' tracelog section is created by [clock_models_to_xml_tracelog](#)

Examples

```
# <logger id="tracelog" ...>
#' # Here
# </logger>
```

clock_model_to_xml_treelogger

Convert a clock model to the XML of the TreeLogger

Description

Convert a clock model to the XML of the TreeLogger

Usage

```
clock_model_to_xml_treelogger(clock_model)
```

Arguments

clock_model a clock model, as returned by [create_clock_model](#)

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

compare_lines	<i>Internal debug function to compare the actually created lines to expected lines using any diff tool</i>
---------------	--

Description

Internal debug function to compare the actually created lines to expected lines using any diff tool

Usage

```
compare_lines(  
  lines,  
  expected,  
  section = NA,  
  created_lines_filename = "created.xml",  
  expected_lines_filename = "expected.xml"  
)
```

Arguments

lines	the created lines
expected	the expected/goal/target lines
section	the XML section. Leave at NA to compare all lines
created_lines_filename	name of the file where the (section of the) created lines are stored
expected_lines_filename	name of the file where the (section of the) expected lines are stored

Value

nothing. Instead, two files are created, with the names `created_lines_filename` and `expected_lines_filename` that contain the section under investigation, so that a diff tool can compare these

Author(s)

Richèl J.C. Bilderbeek

count_trailing_spaces *Count the number of spaces before the first character*

Description

Count the number of spaces before the first character

Usage

```
count_trailing_spaces(line)
```

Arguments

line line of text

Value

the number of spaces before the first character

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_equal(count_trailing_spaces("x"), 0)
expect_equal(count_trailing_spaces(" y"), 1)
expect_equal(count_trailing_spaces(" <"), 2)
expect_equal(count_trailing_spaces(""), 0)
expect_equal(count_trailing_spaces(" "), 1)
expect_equal(count_trailing_spaces("  "), 2)
```

create_alpha_param *Create a parameter called alpha*

Description

Create a parameter called alpha

Usage

```
create_alpha_param(id = NA, value = 0)
```

Arguments

id	the parameter's ID
value	value of the parameter

Value

a parameter called alpha

Note

this parameter is used in a beta distribution (as returned by [create_beta_distr](#)) and gamma distribution (as returned by [create_gamma_distr](#)) and inverse-gamma distribution (as returned by [create_inv_gamma_distr](#)). It cannot be estimated (as a hyper parameter) yet.

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
alpha_param <- create_alpha_param()

# Use the parameter in a distribution
beta_distr <- create_beta_distr(
  alpha = alpha_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = beta_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_bd_tree_prior *Create a Birth-Death tree prior*

Description

Create a Birth-Death tree prior

Usage

```
create_bd_tree_prior(  
  id = NA,  
  birth_rate_distr = create_uniform_distr(),  
  death_rate_distr = create_uniform_distr()  
)
```

Arguments

`id` the ID of the alignment
`birth_rate_distr` the birth rate distribution, as created by a [create_distr](#) function
`death_rate_distr` the death rate distribution, as created by a [create_distr](#) function

Value

a Birth-Death tree_prior

Author(s)

Richèl J.C. Bilderbeek

See Also

An alignment ID can be extracted from its FASTA filename using [get_alignment_id](#)

Examples

```
bd_tree_prior <- create_bd_tree_prior()  
  
beast2_input_file <- tempfile(fileext = ".xml")  
create_beast2_input_file(  
  input_filename = get_fasta_filename(),  
  beast2_input_file,  
  tree_prior = bd_tree_prior  
)  
testit::assert(file.exists(beast2_input_file))  
  
bd_tree_prior_exp <- create_bd_tree_prior()
```

```

    birth_rate_distr = create_exp_distr()
  )

  beast2_input_file <- tempfile(fileext = ".xml")
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    tree_prior = bd_tree_prior_exp
  )
  testit::assert(file.exists(beast2_input_file))

```

```
create_beast2_beast_xml
```

Create the <beast ...> XML

Description

Create the <beast ...> XML

Usage

```
create_beast2_beast_xml(beast2_version, required = "")
```

Arguments

beast2_version BEAST2 version
 required requirement

Value

the XML

Author(s)

Richèl J.C. Bilderbeek

```
create_beast2_input
```

Create a BEAST2 XML input text

Description

Create a BEAST2 XML input text

Usage

```

create_beast2_input(
  input_filename,
  tipdates_filename = NA,
  site_model = create_jc69_site_model(),
  clock_model = create_strict_clock_model(),
  tree_prior = create_yule_tree_prior(),
  mrca_prior = NA,
  mcmc = create_mcmc(),
  beauti_options = create_beauti_options(),
  input_filenames = "deprecated",
  site_models = "deprecated",
  clock_models = "deprecated",
  tree_priors = "deprecated",
  mrca_priors = "deprecated",
  posterior_crown_age = "deprecated"
)

```

Arguments

input_filename	A FASTA filename. Use get_fasta_filename to obtain a testing FASTA filename.
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.
site_model	a site model, as returned by create_site_model
clock_model	a clock model, as returned by create_clock_model
tree_prior	a tree priors, as returned by create_tree_prior
mrca_prior	a Most Recent Common Ancestor prior, as returned by create_mrca_prior
mcmc	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.
beauti_options	one BEAUti options object, as returned by create_beauti_options
input_filenames	One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename.
site_models	one or more site models, as returned by create_site_model
clock_models	a list of one or more clock models, as returned by create_clock_model
tree_priors	one or more tree priors, as returned by create_tree_prior
mrca_priors	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
posterior_crown_age	deprecated

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_beast2_input_from_model](#) to create the BEAST2 XML input text from an inference model Use [create_beast2_input_file](#) to also save it to file.

[create_beast2_input_file](#) shows more examples

Examples

```
text <- create_beast2_input(  
  input_filename = get_fasta_filename()  
)  
testit::assert(substr(text[1], 1, 5) == "<?xml")  
text[1]  
testit::assert(tail(text, n = 1) == "</beast>")
```

create_beast2_input_beast

Creates the XML text for the beast tag of a BEAST2 parameter file.

Description

The beast tag has these elements:

```
<beast[...]>  
  <data  
  [...] </data>  
  [map names]  
  <run[...]>  
  [...] </run>  
</beast>
```

Usage

```
create_beast2_input_beast(  
  input_filename,  
  inference_model = create_inference_model()  
)
```

Arguments

- `input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.
- `inference_model` a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_beast2_input_from_model](#) to create the complete XML text. Use [create_beast2_input_data](#) to create the XML text for the data tag only. Use [create_beast2_input_map](#) to create the XML text for the [map names] part. Use [create_beast2_input_run](#) to create the XML text for the run tag only.

`create_beast2_input_data`

Creates the data section of a BEAST2 XML parameter file

Description

Creates the data section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_data(
  input_filenames,
  beauti_options = create_beauti_options()
)
```

Arguments

- `input_filenames` One or more FASTA filenames. Use [get_fasta_filename](#) to obtain a testing FASTA filename.
- `beauti_options` one BEAUti options object, as returned by [create_beauti_options](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

create_beast2_input_data_sequences

Creates the data section of a BEAST2 XML parameter file

Description

Creates the data section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_data_sequences(  
  input_fasta_filename,  
  beauti_options = create_beauti_options()  
)
```

Arguments

input_fasta_filename
one FASTA filename

beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

`create_beast2_input_distr`*Creates the distribution section of a BEAST2 XML parameter file.*

Description

Creates the distribution section of a BEAST2 XML parameter file.

Usage

```
create_beast2_input_distr(  
  site_models,  
  clock_models,  
  tree_priors,  
  mrca_priors = NA,  
  tipdates_filename = NA  
)
```

Arguments

<code>site_models</code>	one or more site models, as returned by create_site_model
<code>clock_models</code>	a list of one or more clock models, as returned by create_clock_model
<code>tree_priors</code>	one or more tree priors, as returned by create_tree_prior
<code>mrca_priors</code>	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
<code>tipdates_filename</code>	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

lines of XML text

Note

this function is not intended for regular use, thus its long name length is accepted

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_beast2_input](#)

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
# <distribution id="likelihood" ...>
# </distribution>
# </distribution>
```

```
create_beast2_input_distr_lh
```

Creates the likelihood section in the distribution section of a BEAST2 XML parameter file

Description

Creates the likelihood section in the distribution section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_distr_lh(
  site_models,
  clock_models,
  mrca_priors = NA,
  tipdates_filename = NA
)
```

Arguments

site_models	one or more site models, as returned by create_site_model
clock_models	a list of one or more clock models, as returned by create_clock_model
mrca_priors	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Note

this function is not intended for regular use, thus its long name length is accepted

Author(s)

Richèl J.C. Bilderbeek

See Also

this function is called by create_beast2_input_distr, together with create_beast2_input_distr_prior

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     </distribution>
#   <distribution id="likelihood" ...>
#     HERE, where the ID of the distribution is 'likelihood'
#   </distribution>
# </distribution>
```

```
create_beast2_input_distr_prior
```

Creates the prior section in the distribution section of a BEAST2 XML parameter file

Description

Creates the prior section in the distribution section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_distr_prior(
  site_models,
  clock_models,
  tree_priors,
  mrca_priors = NA,
  tipdates_filename = NA
)
```

Arguments

site_models	one or more site models, as returned by create_site_model
clock_models	a list of one or more clock models, as returned by create_clock_model
tree_priors	one or more tree priors, as returned by create_tree_prior
mrca_priors	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Note

this function is not intended for regular use, thus its long name length is accepted

Author(s)

Richèl J.C. Bilderbeek

See Also

this function is called by create_beast2_input_distr, together with create_beast2_input_distr_lh

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
# <distribution id="likelihood" ...>
#   </distribution>
# </distribution>
```

create_beast2_input_file

Create a BEAST2 input file

Description

Create a BEAST2 input file

Usage

```
create_beast2_input_file(  
  input_filename,  
  output_filename,  
  site_model = create_jc69_site_model(),  
  clock_model = create_strict_clock_model(),  
  tree_prior = create_yule_tree_prior(),  
  mrca_prior = NA,  
  mcmc = create_mcmc(),  
  beauti_options = create_beauti_options(),  
  tipdates_filename = NA,  
  input_filenames = "deprecated",  
  site_models = "deprecated",  
  clock_models = "deprecated",  
  tree_priors = "deprecated",  
  mrca_priors = "deprecated",  
  posterior_crown_age = "deprecated"  
)
```

Arguments

input_filename	A FASTA filename. Use get_fasta_filename to obtain a testing FASTA filename.
output_filename	Name of the XML parameter file created by this function. BEAST2 uses this file as input.
site_model	a site model, as returned by create_site_model
clock_model	a clock model, as returned by create_clock_model
tree_prior	a tree priors, as returned by create_tree_prior
mrca_prior	a Most Recent Common Ancestor prior, as returned by create_mrca_prior
mcmc	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.
beauti_options	one BEAUti options object, as returned by create_beauti_options
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.
input_filenames	One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename.
site_models	one or more site models, as returned by create_site_model
clock_models	a list of one or more clock models, as returned by create_clock_model
tree_priors	one or more tree priors, as returned by create_tree_prior
mrca_priors	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
posterior_crown_age	deprecated

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_beast2_input_file_from_model](#) to do the same with an inference model. See [create_site_model](#) for examples with different site models. See [create_clock_model](#) for examples with clock models. See [create_tree_prior](#) for examples with different tree priors. See [create_mcmc](#) for examples with a different MCMC setup.

Examples

```
library(testthat)

# Get an example FASTA file
input_filename <- get_fasta_filename()

# The file created by beautier, a BEAST2 input file
output_filename <- tempfile(pattern = "beast2", fileext = ".xml")

create_beast2_input_file(
  input_filename,
  output_filename
)
expect_true(file.exists(output_filename))
```

```
create_beast2_input_file_from_model
```

Create a BEAST2 input file from an inference model

Description

Create a BEAST2 input file from an inference model

Usage

```
create_beast2_input_file_from_model(
  input_filename,
  output_filename,
  inference_model = create_inference_model()
)
```

Arguments

`input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

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

`inference_model` a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

use [create_beast2_input_from_model](#) to get the BEAST2 input file as text

See [create_site_model](#) for examples with different site models. See [create_clock_model](#) for examples with clock models. See [create_tree_prior](#) for examples with different tree priors. See [create_mcmc](#) for examples with a different MCMC setup. Use [create_beast2_input_file](#) to do the same with the elements of an inference model.

Examples

```
# Get an example FASTA file
input_filename <- get_fasta_filename()

# The file created by beautier, a BEAST2 input file
output_filename <- tempfile()

create_beast2_input_file_from_model(
  input_filename,
  output_filename
)
testthat::expect_true(file.exists(output_filename))
```

```
create_beast2_input_from_model
```

Create a BEAST2 XML input text from an inference model

Description

The main two XML tags are these:

```
<?xml[...]?><beast[...]>
[...]
</beast>
```

Usage

```
create_beast2_input_from_model(input_filename, inference_model)
```

Arguments

`input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

inference_model

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_beast2_input_file_from_model](#) to also save it to file. Use [create_xml_declaration](#) to create the XML text of the XML declaration. Use [create_beast2_input_beast](#) to create the XML text of the beast tag.

Examples

```
library(testthat)

text <- create_beast2_input_from_model(
  input_filename = get_fasta_filename(),
  inference_model = create_inference_model()
)
expect_true(substr(text[1], 1, 5) == "<?xml")
expect_true(tail(text, n = 1) == "</beast>")
```

create_beast2_input_init

Creates the init section of a BEAST2 XML parameter file

Description

Creates the init section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_init(ids)
```

Arguments

ids one or more alignments' IDs. IDs can be extracted from their FASTA filenames with [get_alignment_ids_from_fasta_filenames](#))

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

create_beast2_input_map

Creates the map section of a BEAST2 XML parameter file

Description

Creates the map section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_map()
```

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

create_beast2_input_operators

Creates the operators section of a BEAST2 XML parameter file

Description

Creates the operators section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_operators(  
  site_models,  
  clock_models,  
  tree_priors,  
  fixed_crown_ages = rep(FALSE, length(site_models)),  
  mrca_priors = NA,  
  tipdates_filename = NA  
)
```

Arguments

site_models	one or more site models, as returned by create_site_model
clock_models	a list of one or more clock models, as returned by create_clock_model
tree_priors	one or more tree priors, as returned by create_tree_prior
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.
mrca_priors	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

create_beast2_input_run

Creates the 'run' section of a BEAST2 XML parameter file

Description

The run tag has these elements:

```

<run[...]>
  <state[...]>
    [...]
  </state>
  <init[...]>
    [...]
  </init>
  <distribution[...]>
    [...]
  </distribution>
  [operator ids]
  [loggers]
</run>

```

Usage

```
create_beast2_input_run(
  input_filename,
  inference_model = create_inference_model()
)
```

Arguments

`input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

`inference_model` a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_beast2_input_state](#) to create the XML text of the state tag. Use [create_beast2_input_init](#) to create the XML text of the init tag. Use [create_beast2_input_distr](#) to create the XML text of the distribution tag. Use [create_beast2_input_operators](#) to create the XML text of the [operator ids] section. Use [create_loggers_xml](#) to create the XML text of the [loggers] part.

create_beast2_input_state

Creates the state section of a BEAST2 XML parameter file

Description

Creates the state section of a BEAST2 XML parameter file

Usage

```
create_beast2_input_state(
  site_models,
  clock_models,
  tree_priors,
  mrca_priors = NA,
  tipdates_filename = NA
)
```

Arguments

site_models	one or more site models, as returned by create_site_model
clock_models	a list of one or more clock models, as returned by create_clock_model
tree_priors	one or more tree priors, as returned by create_tree_prior
mrca_priors	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

create_beauti_options *Function to create a set of BEAUti options.*

Description

BEAUti options are settings that differ between BEAUti version. The use of these options is mostly for testing older versions. Whatever option chosen here, the created XML file will be valid.

Usage

```
create_beauti_options(
  capitalize_first_char_id = FALSE,
  nucleotides_uppercase = FALSE,
  beast2_version = "2.4",
  required = "",
  sequence_indent = 20
)
```

Arguments

capitalize_first_char_id	must the ID of alignment start with a capital? TRUE if yes, FALSE if it can be left lower case (if it is lowercase)
nucleotides_uppercase	must the nucleotides of the DNA sequence be in uppercase?
beast2_version	the BEAST2 version
required	things that may be required, for example BEAST v2.5.0
sequence_indent	the number of spaces the XML sequence lines are indented

Value

a BEAUi options structure

Author(s)

Richèl J.C. Bilderbeek

Examples

```
beauti_options <- create_beauti_options(  
  nucleotides_uppercase = TRUE,  
  beast2_version = "2.5"  
)  
xml <- create_beast2_input(  
  get_fasta_filename(),  
  beauti_options = beauti_options  
)  
testit::assert(is.character(xml))  
testit::assert(length(xml) > 1)
```

create_beta_distr *Create a beta distribution*

Description

Create a beta distribution

Usage

```
create_beta_distr(id = NA, alpha = 0, beta = 1)
```

Arguments

id	the distribution's ID
alpha	the alpha shape parameter, a numeric value. The value of alpha must be at least 0.0. For advanced usage, use the structure as returned by create_alpha_param .
beta	the beta shape parameter, a numeric value. The value of beta must be at least 1.0. For advanced usage, use the structure as returned by create_beta_param .

Value

a beta distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
beta_distr <- create_beta_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = beta_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_beta_param	<i>Create a parameter called beta</i>
-------------------	---------------------------------------

Description

Create a parameter called beta

Usage

```
create_beta_param(id = NA, value = 1)
```

Arguments

id	the parameter's ID
value	value of the parameter

Value

a parameter called beta

Note

this parameter is used in a beta distribution (as returned by [create_beta_distr](#)) and gamma distribution (as returned by [create_gamma_distr](#)) and inverse-gamma distribution (as returned by [create_inv_gamma_distr](#)). It cannot be estimated (as a hyper parameter) yet.

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
beta_param <- create_beta_param()

# Use the parameter in a distribution
gamma_distr <- create_gamma_distr(
  beta = beta_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = gamma_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_cbs_tree_prior *Create a Coalescent Bayesian Skyline tree prior*

Description

Create a Coalescent Bayesian Skyline tree prior

Usage

```
create_cbs_tree_prior(id = NA, group_sizes_dimension = 5)
```

Arguments

id an alignment's IDs. An ID can be extracted from its FASTA filename with [get_alignment_ids_from_fasta_filenames](#))

group_sizes_dimension the group sizes' dimension, as used by the CBS tree prior (see [create_cbs_tree_prior](#))

Value

a Coalescent Bayesian Skyline tree_prior

Author(s)

Richèl J.C. Bilderbeek

See Also

An alignment ID can be extracted from its FASTA filename using [get_alignment_id](#)

Examples

```
cbs_tree_prior <- create_cbs_tree_prior()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_beautier_path("test_output_6.fas"),
  beast2_input_file,
  tree_prior = cbs_tree_prior
)
testit::assert(file.exists(beast2_input_file))
```

create_ccp_tree_prior *Create a Coalescent Constant Population tree prior*

Description

Create a Coalescent Constant Population tree prior

Usage

```
create_ccp_tree_prior(
  id = NA,
  pop_size_distr = beautier::create_one_div_x_distr()
)
```

Arguments

`id` the ID of the alignment
`pop_size_distr` the population distribution, as created by a [create_distr](#) function

Value

a Coalescent Constant Population tree_prior

Author(s)

Richèl J.C. Bilderbeek

See Also

An alignment ID can be extracted from its FASTA filename using [get_alignment_id](#)

Examples

```
ccp_tree_prior <- create_ccp_tree_prior()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = ccp_tree_prior
)
testit::assert(file.exists(beast2_input_file))
```

create_cep_tree_prior *Create a Coalescent Exponential Population tree prior*

Description

Create a Coalescent Exponential Population tree prior

Usage

```
create_cep_tree_prior(
  id = NA,
  pop_size_distr = create_one_div_x_distr(),
  growth_rate_distr = create_laplace_distr()
)
```

Arguments

`id` the ID of the alignment

`pop_size_distr` the population distribution, as created by a [create_distr](#) function

`growth_rate_distr` the growth rate distribution, as created by a [create_distr](#) function

Value

a Coalescent Exponential Population tree_prior

Author(s)

Richèl J.C. Bilderbeek

See Also

An alignment ID can be extracted from its FASTA filename using [get_alignment_id](#)

Examples

```
cep_tree_prior <- create_cep_tree_prior()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = cep_tree_prior
)
testit::assert(file.exists(beast2_input_file))
```

create_clock_model *General function to create a clock model*

Description

General function to create a clock model

Usage

```
create_clock_model(name, id, ...)
```

Arguments

name	the clock model name. Valid names can be found in <code>get_clock_model_names</code>
id	a clock model's ID
...	specific clock model parameters

Value

a valid clock model

Note

Prefer using the named function [create_rln_clock_model](#) and [create_strict_clock_model](#)

Author(s)

Richèl J.C. Bilderbeek

See Also

An alignment ID can be extracted from its FASTA filename using [get_alignment_id](#). For more examples about creating a relaxed log-normal clock model, see [create_rln_clock_model](#). For more examples about creating a strict clock model, see [create_strict_clock_model](#).

Examples

```
rln_clock_model <- create_rln_clock_model()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  clock_model = rln_clock_model
)
testit::assert(file.exists(beast2_input_file))

strict_clock_model <- create_strict_clock_model()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  clock_model = strict_clock_model
)
testit::assert(file.exists(beast2_input_file))
```

`create_clock_models` *Creates all supported clock models, which is a list of the types returned by [create_rln_clock_model](#), and [create_strict_clock_model](#)*

Description

Creates all supported clock models, which is a list of the types returned by [create_rln_clock_model](#), and [create_strict_clock_model](#)

Usage

```
create_clock_models()
```

Value

a list of site_models

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_model](#) to create a clock model

Examples

```
library(testthat)

clock_models <- create_clock_models()
expect_true(is_rln_clock_model(clock_models[[1]]))
expect_true(is_strict_clock_model(clock_models[[2]]))
```

```
create_clock_models_from_names
      Create clock models from their names
```

Description

Create clock models from their names

Usage

```
create_clock_models_from_names(clock_model_names)
```

Arguments

```
clock_model_names
  one or more names of a clock model, must be name among those returned by
  get\_clock\_model\_names
```

Value

a list of one or more clock models

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_models](#) to get all clock models

Examples

```
names <- get_clock_model_names()
clock_models <- create_clock_models_from_names(names)

for (i in seq_along(names)) {
  testthat::expect_equal(names[i], clock_models[[i]]$name)
}
```

create_clock_model_from_name
Create a clock model from name

Description

Create a clock model from name

Usage

```
create_clock_model_from_name(clock_model_name)
```

Arguments

clock_model_name
name of a clock model, must be a name as returned by [get_clock_model_names](#)

Value

a clock model, as can be created by [create_clock_model](#)

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_model](#) to create a clock model

Examples

```
clock_model_names <- get_clock_model_names()
for (clock_model_name in clock_model_names) {
  clock_model <- create_clock_model_from_name(clock_model_name)
  testthat::expect_equal(clock_model_name, clock_model$name)
}
```

`create_clock_rate_param`

Create a parameter called `clock_rate`, as needed by [create_strict_clock_model](#)

Description

Create a parameter called `clock_rate`, as needed by [create_strict_clock_model](#)

Usage

```
create_clock_rate_param(value = "1.0", id = NA)
```

Arguments

<code>value</code>	value of the parameter
<code>id</code>	the parameter's ID

Value

a parameter called `rate`

Note

It cannot be estimated (as a hyper parameter) yet.

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
clock_rate_param <- create_clock_rate_param(  
  id = "anthus_aco", value = 1.0  
)  
  
# Use the parameter in a clock model  
strict_clock_model <- create_strict_clock_model(  
  clock_rate_param = clock_rate_param  
)  
  
# Use the distribution to create a BEAST2 input file  
beast2_input_file <- tempfile(fileext = ".xml")  
create_beast2_input_file(  
  input_filename = get_fasta_filename(),
```

```

    beast2_input_file,
    clock_model = strict_clock_model
)
testit::assert(file.exists(beast2_input_file))

```

create_data_xml *Create the <data ..> XML*

Description

Create the <data ..> XML

Usage

```
create_data_xml(id, beast2_version)
```

Arguments

id an alignment's IDs. An ID can be extracted from its FASTA filename with [get_alignment_ids_from_fasta_filenames](#))

beast2_version BEAST2 version, for example, code"2.5"

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

create_distr *General function to create a distribution.*

Description

General function to create a distribution.

Usage

```
create_distr(name, id, ...)
```

Arguments

name the distribution name. Valid names can be found in [get_distr_names](#)

id the distribution's ID

... specific distribution parameters

Value

a distribution

Note

Prefer using the named functions [create_beta_distr](#), [create_exp_distr](#), [create_gamma_distr](#), [create_inv_gamma_distr](#), [create_laplace_distr](#), [create_log_normal_distr](#), [create_normal_distr](#), [create_one_div_x_distr](#), [create_poisson_distr](#) and [create_uniform_distr](#)

See [create_beta_distr](#), [create_exp_distr](#), [create_gamma_distr](#), [create_inv_gamma_distr](#), [create_laplace_distr](#), [create_log_normal_distr](#), [create_normal_distr](#), [create_one_div_x_distr](#), [create_poisson_distr](#) and [create_uniform_distr](#) for examples how to use those distributions

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# Use any distribution
distr <- create_beta_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_exp_distr	<i>Create an exponential distribution</i>
------------------	---

Description

Create an exponential distribution

Usage

```
create_exp_distr(id = NA, mean = 1)
```

Arguments

id	the distribution's ID
mean	the mean parameter, a numeric value. For advanced usage, use the structure as returned by create_mean_param

Value

an exponential distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
exp_distr <- create_exp_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = exp_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_gamma_distr *Create a gamma distribution*

Description

Create a gamma distribution

Usage

```
create_gamma_distr(id = NA, alpha = 0.5396, beta = 0.3819)
```

Arguments

id	the distribution's ID
alpha	the alpha shape parameter, a numeric value. For advanced usage, use the structure as returned by create_alpha_param
beta	the beta shape parameter, a numeric value. For advanced usage, use the structure as returned by create_beta_param

Value

a gamma distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
gamma_distr <- create_gamma_distr(  
  alpha = 0.05,  
  beta = 10.0  
)  
  
gtr_site_model <- create_gtr_site_model(  
  rate_ac_prior_distr = gamma_distr  
)  
  
beast2_input_file <- tempfile(fileext = ".xml")  
create_beast2_input_file(  
  input_filename = get_fasta_filename(),  
  beast2_input_file,  
  site_model = gtr_site_model  
)  
testit::assert(file.exists(beast2_input_file))
```

create_gamma_site_model

Create a gamma site model, part of a site model

Description

Create a gamma site model, part of a site model

Usage

```
create_gamma_site_model(  
  gamma_cat_count = "0",  
  gamma_shape = "1.0",  
  prop_invariant = "0.0",  
  gamma_shape_prior_distr = NA,  
  freq_equilibrium = "estimated"  
)
```

Arguments

`gamma_cat_count` the number of gamma categories, must be an integer with value zero or more

`gamma_shape` gamma curve shape parameter

`prop_invariant` the proportion invariant, must be a value from 0.0 to 1.0

`gamma_shape_prior_distr` the distribution of the gamma shape prior. `gamma_shape_prior_distr` must be NA for a `gamma_cat_count` of zero or one. For a `gamma_cat_count` of two or more, leaving `gamma_shape_prior_distr` equal to its default value of NA, a default distribution is used. Else `gamma_shape_prior_distr` must be a distribution, as can be created by [create_distr](#)

`freq_equilibrium` the frequency in which the rates are at equilibrium are either estimated, empirical or all_equal. `get_freq_equilibrium_names` returns the possible values for `freq_equilibrium`

Value

a gamma site model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_gamma_site_model](#) to create a gamma site model

Examples

```
gamma_site_model <- create_gamma_site_model(prop_invariant = 0.5)

site_model <- create_hky_site_model(gamma_site_model = gamma_site_model)

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  site_model = site_model
)
testit::assert(file.exists(beast2_input_file))
```

create_gtr_site_model *Create a GTR site model*

Description

Create a GTR site model

Usage

```
create_gtr_site_model(  
  id = NA,  
  gamma_site_model = create_gamma_site_model(),  
  rate_ac_prior_distr = create_gamma_distr(alpha = 0.05, beta = create_beta_param(value  
    = "10.0")),  
  rate_ag_prior_distr = create_gamma_distr(alpha = 0.05, beta = create_beta_param(value  
    = "20.0")),  
  rate_at_prior_distr = create_gamma_distr(alpha = 0.05, beta = create_beta_param(value  
    = "10.0")),  
  rate_cg_prior_distr = create_gamma_distr(alpha = 0.05, beta = create_beta_param(value  
    = "10.0")),  
  rate_gt_prior_distr = create_gamma_distr(alpha = 0.05, beta = create_beta_param(value  
    = "10.0")),  
  rate_ac_param = create_rate_ac_param(),  
  rate_ag_param = create_rate_ag_param(),  
  rate_at_param = create_rate_at_param(),  
  rate_cg_param = create_rate_cg_param(),  
  rate_ct_param = create_rate_ct_param(),  
  rate_gt_param = create_rate_gt_param(),  
  freq_equilibrium = "estimated"  
)
```

Arguments

id the IDs of the alignment (can be extracted from the FASTA filename using [get_alignment_id](#))

gamma_site_model a gamma site model, as created by [create_gamma_site_model](#)

rate_ac_prior_distr the AC rate prior distribution, as returned by [create_distr](#)

rate_ag_prior_distr the AG rate prior distribution, as returned by [create_distr](#)

rate_at_prior_distr the AT rate prior distribution, as returned by [create_distr](#)

rate_cg_prior_distr the CG rate prior distribution, as returned by [create_distr](#)

rate_gt_prior_distr	the GT rate prior distribution, as returned by create_distr
rate_ac_param	the 'rate AC' parameter, a numeric value. For advanced usage, use the structure as returned by create_rate_ac_param
rate_ag_param	the 'rate AG' parameter, a numeric value. For advanced usage, use the structure as returned by create_rate_ag_param
rate_at_param	the 'rate AT' parameter, a numeric value. For advanced usage, use the structure as returned by create_rate_at_param
rate_cg_param	the 'rate CG' parameter, a numeric value. For advanced usage, use the structure as returned by create_rate_cg_param
rate_ct_param	the 'rate CT' parameter, a numeric value. For advanced usage, use the structure as returned by create_rate_ct_param
rate_gt_param	the 'rate GT' parameter, a numeric value. For advanced usage, use the structure as returned by create_rate_gt_param
freq_equilibrium	the frequency in which the rates are at equilibrium are either estimated, empirical or all_equal. <code>get_freq_equilibrium_names</code> returns the possible values for <code>freq_equilibrium</code>

Value

a GTR `site_model`

Author(s)

Richèl J.C. Bilderbeek

Examples

```
gtr_site_model <- create_gtr_site_model(
  rate_ac_param = 1.2,
  rate_ag_param = 2.3,
  rate_at_param = 3.4,
  rate_cg_param = 4.5,
  rate_ct_param = 5.6,
  rate_gt_param = 6.7
)

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  site_model = gtr_site_model
)
testit::assert(file.exists(beast2_input_file))
```

create_hky_site_model *Create an HKY site model*

Description

Create an HKY site model

Usage

```
create_hky_site_model(  
  id = NA,  
  kappa = "2.0",  
  gamma_site_model = create_gamma_site_model(),  
  kappa_prior_distr = create_log_normal_distr(m = create_m_param(value = "1.0"), s =  
    1.25),  
  freq_equilibrium = "estimated"  
)
```

Arguments

id	the IDs of the alignment (can be extracted from the FASTA filename using get_alignment_id)
kappa	the kappa
gamma_site_model	a gamma site model, as created by create_gamma_site_model
kappa_prior_distr	the distribution of the kappa prior, which is a log-normal distribution (as created by create_log_normal_distr) by default
freq_equilibrium	the frequency in which the rates are at equilibrium are either estimated, empirical or all_equal. get_freq_equilibrium_names returns the possible values for freq_equilibrium

Value

an HKY site_model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
hky_site_model <- create_hky_site_model()  
  
create_beast2_input_file(  
  input_filename = get_fasta_filename(),
```

```

    output_filename = tempfile(pattern = "beast", fileext = ".xml"),
    site_model = hky_site_model
)

```

```
create_inference_model
```

Create a Bayesian phylogenetic inference model.

Description

Create a Bayesian phylogenetic inference model, as can be done by BEAUti.

Usage

```

create_inference_model(
  site_model = create_jc69_site_model(),
  clock_model = create_strict_clock_model(),
  tree_prior = create_yule_tree_prior(),
  mrca_prior = NA,
  mcmc = create_mcmc(),
  beauti_options = create_beauti_options(),
  tipdates_filename = NA
)

```

Arguments

site_model	a site model, as returned by create_site_model
clock_model	a clock model, as returned by create_clock_model
tree_prior	a tree priors, as returned by create_tree_prior
mrca_prior	a Most Recent Common Ancestor prior, as returned by create_mrca_prior
mcmc	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.
beauti_options	one BEAUti options object, as returned by create_beauti_options
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

an inference model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_inference_model](#) to create an inference model

Examples

```
# Create an MCMC chain with 50 states
inference_model <- create_inference_model(
  mcmc = create_mcmc(chain_length = 50000, store_every = 1000)
)

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file_from_model(
  get_fasta_filename(),
  beast2_input_file,
  inference_model = inference_model
)
testit::assert(file.exists(beast2_input_file))
```

create_inv_gamma_distr

Create an inverse-gamma distribution

Description

Create an inverse-gamma distribution

Usage

```
create_inv_gamma_distr(id = NA, alpha = 0, beta = 1)
```

Arguments

id	the distribution's ID
alpha	the alpha shape parameter, a numeric value. For advanced usage, use the structure as returned by create_alpha_param
beta	the beta shape parameter, a numeric value. For advanced usage, use the structure as returned by create_beta_param

Value

an inverse-gamma distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
inv_gamma_distr <- create_inv_gamma_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = inv_gamma_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_jc69_site_model

Create a JC69 site model

Description

Create a JC69 site model

Usage

```
create_jc69_site_model(id = NA, gamma_site_model = create_gamma_site_model())
```

Arguments

id the IDs of the alignment (can be extracted from the FASTA filename using [get_alignment_id](#))

gamma_site_model a gamma site model, as created by [create_gamma_site_model](#)

Value

a JC69 site_model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
jc69_site_model <- create_jc69_site_model()

create_beast2_input_file(
  input_filename = get_fasta_filename(),
  output_filename = tempfile(pattern = "beast", fileext = ".xml"),
  site_model = jc69_site_model
)
```

create_kappa_1_param *Create a parameter called kappa 1*

Description

Create a parameter called kappa 1

Usage

```
create_kappa_1_param(id = NA, lower = "0.0", value = "2.0", estimate = TRUE)
```

Arguments

id	the parameter's ID
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value
value	value of the parameter
estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise

Value

a parameter called kappa 1

Author(s)

Richèl J.C. Bilderbeek

create_kappa_2_param *Create a parameter called kappa 2*

Description

Create a parameter called kappa 2

Usage

```
create_kappa_2_param(id = NA, lower = "0.0", value = "2.0", estimate = TRUE)
```

Arguments

id	the parameter's ID
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value
value	value of the parameter
estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise

Value

a parameter called kappa 2

Author(s)

Richèl J.C. Bilderbeek

create_lambda_param *Create a parameter called lambda*

Description

Create a parameter called lambda

Usage

```
create_lambda_param(id = NA, value = 0)
```

Arguments

id	the parameter's ID
value	value of the parameter

Value

a parameter called lambda

Note

this parameter is used in a Poisson distribution (as returned by [create_poisson_distr](#))

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
lambda_param <- create_lambda_param()

# Use the parameter in a distribution
poisson_distr <- create_poisson_distr(
  lambda = lambda_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = poisson_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_laplace_distr *Create a Laplace distribution*

Description

Create a Laplace distribution

Usage

```
create_laplace_distr(id = NA, mu = 0, scale = 1)
```

Arguments

id	the distribution's ID
mu	the mu parameter, a numeric value. For advanced usage, use the structure as returned by create_mu_param
scale	the scale parameter, a numeric value. For advanced usage, use the structure as returned by create_scale_param

Value

a Laplace distribution

Author(s)

Richard J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
laplace_distr <- create_laplace_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = laplace_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_loggers_xml *Creates the three logger sections of a BEAST2 XML parameter file*

Description

The logger section has these elements:

```
<logger id="tracelog" [...]>
  [...]
</logger>
<logger id="screenlog" [...]>
  [...]
</logger>
<logger id="treelog.t:[alignment ID]" [...]>
  [...]
</logger>
```

Usage

```
create_loggers_xml(input_filename, inference_model)
```

Arguments

- `input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.
- `inference_model` a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tracelog_xml](#) to create the XML text of the logger with the tracelog ID. Use [create_screenlog_xml](#) to create the XML text of the logger with the screenlog ID. Use [create_treeelog_xml](#) to create the XML text of the loggers with the treeelog ID.

create_log_normal_distr

Create a log-normal distribution

Description

Create a log-normal distribution

Usage

```
create_log_normal_distr(id = NA, m = 0, s = 0)
```

Arguments

- `id` the distribution's ID
- `m` the m parameter, a numeric value. For advanced usage, use the structure as returned by [create_m_param](#)
- `s` the s parameter, a numeric value. For advanced usage, use the structure as returned by [create_s_param](#)

Value

a log-normal distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
log_normal_distr <- create_log_normal_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = log_normal_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_mcmc

Create an MCMC configuration.

Description

Create an MCMC configuration, as in the BEAUti MCMC tab. The number of states that will be saved equals the chain length (`chain_length`) divided by the number of states between each sampling event (`store_every`)

Usage

```
create_mcmc(
  chain_length = 1e+07,
  store_every = -1,
  pre_burnin = 0,
  n_init_attempts = 10,
  sample_from_prior = FALSE,
  tracelog = create_tracelog(),
  screenlog = create_screenlog(),
  treelog = create_treelog()
)
```

Arguments

<code>chain_length</code>	length of the MCMC chain
<code>store_every</code>	number of states the MCMC will process before the posterior's state will be saved to file. Use -1 or NA to use the default frequency.
<code>pre_burnin</code>	number of burn in samples taken before entering the main loop
<code>n_init_attempts</code>	number of initialization attempts before failing

sample_from_prior set to **TRUE** to sample from the prior

tracelog a tracelog, as created by [create_tracelog](#)

screenlog a screenlog, as created by [create_screenlog](#)

treelog a treelog, as created by [create_treelog](#)

Value

an MCMC configuration

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Examples

```
library(testthat)

# Create an MCMC chain with 50 states
mcmc <- create_mcmc(chain_length = 50000, store_every = 1000)

beast2_input_file <- tempfile()
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  mcmc = mcmc
)
expect_true(file.exists(beast2_input_file))
```

create_mean_param *Create a parameter called mean*

Description

Create a parameter called mean

Usage

```
create_mean_param(id = NA, value = 0)
```

Arguments

id the parameter's ID

value value of the parameter

Value

a parameter called mean

Note

this parameter is used in an exponential distribution (as returned by [create_exp_distr](#)) and normal distribution (as returned by [create_normal_distr](#)). It cannot be estimated (as a hyper parameter) yet.

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
mean_param <- create_mean_param(value = 1.0)

# Use the parameter in a distribution
exp_distr <- create_exp_distr(
  mean = mean_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = exp_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_mrca_prior

Create a Most Recent Common Ancestor prior

Description

Create a Most Recent Common Ancestor prior

Usage

```
create_mrca_prior(
  alignment_id = NA,
  taxa_names = NA,
  is_monophyletic = FALSE,
  mrca_distr = NA,
  name = NA,
  clock_prior_distr_id = NA
)
```

Arguments

<code>alignment_id</code>	ID of the alignment, as returned by get_alignment_id . Keep at NA to have it initialized automatically
<code>taxa_names</code>	names of the taxa, as returned by get_taxa_names . Keep at NA to have it initialized automatically, using all taxa in the alignment
<code>is_monophyletic</code>	boolean to indicate monophyly is assumed in a Most Recent Common Ancestor prior, as returned by create_mrca_prior
<code>mrca_distr</code>	the distribution used by the MRCA prior. Can be NA (the default) or any distribution returned by create_distr
<code>name</code>	the unique name of the MRCA prior, for example a genus, family, order or even class name. Leave at NA to have it named automatically.
<code>clock_prior_distr_id</code>	ID of an MRCA clock model's distribution. Keep at NA to have it initialized automatically

Value

an MRCA prior

Author(s)

Richèl J.C. Bilderbeek

Examples

```
fasta_filename <- get_beautier_path("anthus_aco.fas")

# The first two taxa are sister species
mrca_prior <- create_mrca_prior(
  alignment_id = get_alignment_id(fasta_filename = fasta_filename),
  taxa_names = get_taxa_names(filename = fasta_filename)[1:2]
)

# Set the crown age
mrca_prior <- create_mrca_prior(
  alignment_id = get_alignment_id(fasta_filename = fasta_filename),
```

```
    taxa_names = get_taxa_names(filename = fasta_filename),  
    is_monophyletic = TRUE  
  )
```

create_mu_param	<i>Create a parameter called mu</i>
-----------------	-------------------------------------

Description

Create a parameter called mu

Usage

```
create_mu_param(id = NA, value = 0)
```

Arguments

id	the parameter's ID
value	value of the parameter

Value

a parameter called mu

Note

this parameter is used in a Laplace distribution (as returned by [create_laplace_distr](#)). It cannot be estimated (as a hyper parameter) yet.

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter  
mu_param <- create_mu_param()  
  
# Use the parameter in a distribution  
laplace_distr <- create_laplace_distr(  
  mu = mu_param  
)  
  
# Use the distribution to create a BEAST2 input file  
beast2_input_file <- tempfile(fileext = ".xml")
```

```
create_beast2_input_file(  
  input_filename = get_fasta_filename(),  
  beast2_input_file,  
  tree_prior = create_yule_tree_prior(  
    birth_rate_distr = laplace_distr  
  )  
)  
testit::assert(file.exists(beast2_input_file))
```

create_m_param	<i>Create a parameter called m</i>
----------------	------------------------------------

Description

Create a parameter called m

Usage

```
create_m_param(id = NA, value = 0)
```

Arguments

id	the parameter's ID
value	value of the parameter

Value

a parameter called m

Note

this parameter is used in a log-normal distribution (as returned by [create_log_normal_distr](#)) It cannot be estimated (as a hyper parameter) yet.

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
m_param <- create_m_param()

# Use the parameter in a distribution
log_normal_distr <- create_log_normal_distr(
  m = m_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = log_normal_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_normal_distr *Create an normal distribution*

Description

Create an normal distribution

Usage

```
create_normal_distr(id = NA, mean = 0, sigma = 1)
```

Arguments

id	the distribution's ID
mean	the mean parameter, a numeric value. For advanced usage, use the structure as returned by create_mean_param
sigma	the sigma parameter, a numeric value. For advanced usage, use the structure as returned by create_sigma_param

Value

a normal distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function `create_distr` shows an overview of all supported distributions

Examples

```
normal_distr <- create_normal_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = normal_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_ns_mcmc	<i>Create an MCMC object to estimate the marginal likelihood using Nested Sampling.</i>
----------------	---

Description

This will result in a BEAST run that estimates the marginal likelihood until convergence is achieved. In this context, `chain_length` is only an upper bound to the length of that run.

Usage

```
create_ns_mcmc(
  chain_length = 1e+07,
  store_every = -1,
  pre_burnin = 0,
  n_init_attempts = 3,
  particle_count = 1,
  sub_chain_length = 5000,
  epsilon = "1e-12",
  tracelog = create_tracelog(),
  screenlog = create_screenlog(),
  treelog = create_treelog()
)
```

Arguments

<code>chain_length</code>	upper bound to the length of the MCMC chain
<code>store_every</code>	number of states the MCMC will process before the posterior's state will be saved to file. Use -1 or NA to use the default frequency.
<code>pre_burnin</code>	number of burn in samples taken before entering the main loop

n_init_attempts	number of initialization attempts before failing
particle_count	number of particles
sub_chain_length	sub-chain length
epsilon	epsilon
tracelog	a tracelog, as created by create_tracelog
screenlog	a screenlog, as created by create_screenlog
treelog	a treelog, as created by create_treelog

Value

an MCMC object

Author(s)

Richèl J.C. Bilderbeek

References

* [1] Patricio Maturana Russel, Brendon J Brewer, Steffen Klaere, Remco R Bouckaert; Model Selection and Parameter Inference in Phylogenetics Using Nested Sampling, *Systematic Biology*, 2018, syy050, <https://doi.org/10.1093/sysbio/syy050>

See Also

Use [create_mcmc](#) to create a regular MCMC. Use [create_test_ns_mcmc](#) to create an NS MCMC for testing, with, among others, a short MCMC chain length. Use [check_ns_mcmc](#) to check that an NS MCMC object is valid.

Examples

```
mcmc <- create_ns_mcmc(
  chain_length = 1e7,
  store_every = 1000,
  particle_count = 1,
  sub_chain_length = 1000,
  epsilon = 1e-12
)

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  mcmc = mcmc
)
testit::assert(file.exists(beast2_input_file))
```

`create_one_div_x_distr`*Create a 1/x distribution*

Description

Create a 1/x distribution

Usage

```
create_one_div_x_distr(id = NA)
```

Arguments

`id` the distribution's ID

Value

a 1/x distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
one_div_x_distr <- create_one_div_x_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = one_div_x_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_param	<i>General function to create a parameter.</i>
--------------	--

Description

General function to create a parameter.

Usage

```
create_param(name, id, value, ...)
```

Arguments

name	the parameters' name. Valid names can be found in <code>get_param_names</code>
id	the parameter's ID
value	value of the parameter
...	specific parameter parameters

Value

a parameter

Note

Prefer using the named functions `create_alpha_param`, `create_beta_param`, `create_clock_rate_param`, `create_kappa_1_param`, `create_kappa_2_param`, `create_lambda_param`, `create_m_param`, `create_mean_param`, `create_mu_param`, `create_rate_ac_param`, `create_rate_ag_param`, `create_rate_at_param`, `create_rate_cg_param`, `create_rate_ct_param`, `create_rate_gt_param`, `create_s_param`, `create_scale_param`, and `create_sigma_param`

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# Create an alpha parameter
alpha_param <- create_alpha_param()

# Use the parameter in a distribution
beta_distr <- create_beta_distr(
  alpha = alpha_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
```

```
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = beta_distr  
    )  
  )  
  testit::assert(file.exists(beast2_input_file))
```

create_poisson_distr *Create a Poisson distribution*

Description

Create a Poisson distribution

Usage

```
create_poisson_distr(id = NA, lambda = 0)
```

Arguments

id	the distribution's ID
lambda	the lambda parameter, a numeric value. For advanced usage, use the structure as returned by create_lambda_param

Value

a Poisson distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
poisson_distr <- create_poisson_distr()  
  
beast2_input_file <- tempfile(fileext = ".xml")  
create_beast2_input_file(  
  input_filename = get_fasta_filename(),  
  beast2_input_file,  
  tree_prior = create_yule_tree_prior(  
    birth_rate_distr = poisson_distr  
  )  
)  
testit::assert(file.exists(beast2_input_file))
```

create_rate_ac_param *Create a parameter called 'rate AC'*

Description

Create a parameter called 'rate AC'

Usage

```
create_rate_ac_param(id = NA, estimate = TRUE, value = "1.0", lower = "0.0")
```

Arguments

id	the parameter's ID
estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise
value	value of the parameter
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value

Value

a parameter called 'rate AC'

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create parameter
rate_ac_param <- create_rate_ac_param(value = 1, estimate = FALSE)

# Use the parameter to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  site_model = create_gtr_site_model(
    rate_ac_param = rate_ac_param
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_rate_ag_param *Create a parameter called 'rate AG'*

Description

Create a parameter called 'rate AG'

Usage

```
create_rate_ag_param(id = NA, estimate = TRUE, value = "1.0", lower = "0.0")
```

Arguments

id	the parameter's ID
estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise
value	value of the parameter
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value

Value

a parameter called 'rate AG'

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create parameter
rate_ag_param <- create_rate_ag_param(value = 1, estimate = FALSE)

# Use the parameter to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  site_model = create_gtr_site_model(
    rate_ag_param = rate_ag_param
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_rate_at_param *Create a parameter called 'rate AT'*

Description

Create a parameter called 'rate AT'

Usage

```
create_rate_at_param(id = NA, estimate = TRUE, value = "1.0", lower = "0.0")
```

Arguments

id	the parameter's ID
estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise
value	value of the parameter
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value

Value

a parameter called 'rate AT'

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create parameter
rate_at_param <- create_rate_at_param(value = 1, estimate = FALSE)

# Use the parameter to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  site_model = create_gtr_site_model(
    rate_at_param = rate_at_param
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_rate_cg_param *Create a parameter called 'rate CG'*

Description

Create a parameter called 'rate CG'

Usage

```
create_rate_cg_param(id = NA, estimate = TRUE, value = "1.0", lower = "0.0")
```

Arguments

id	the parameter's ID
estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise
value	value of the parameter
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value

Value

a parameter called 'rate CG'

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create parameter
rate_cg_param <- create_rate_cg_param(value = 1, estimate = FALSE)

# Use the parameter to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  site_model = create_gtr_site_model(
    rate_cg_param = rate_cg_param
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_rate_ct_param *Create a parameter called 'rate CT'*

Description

Create a parameter called 'rate CT'

Usage

```
create_rate_ct_param(id = NA, value = "1.0", lower = "0.0")
```

Arguments

id	the parameter's ID
value	value of the parameter
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value

Value

a parameter called 'rate CT'

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create parameter
rate_ct_param <- create_rate_ct_param(value = 1)

# Use the parameter to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  site_model = create_gtr_site_model(
    rate_ct_param = rate_ct_param
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_rate_gt_param *Create a parameter called 'rate GT'*

Description

Create a parameter called 'rate GT'

Usage

```
create_rate_gt_param(id = NA, estimate = TRUE, value = "1.0", lower = "0.0")
```

Arguments

id	the parameter's ID
estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise
value	value of the parameter
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value

Value

a parameter called 'rate GT'

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create parameter
rate_gt_param <- create_rate_gt_param(value = 1, estimate = FALSE)

# Use the parameter to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  site_model = create_gtr_site_model(
    rate_gt_param = rate_gt_param
  )
)
testit::assert(file.exists(beast2_input_file))
```

```
create_rln_clock_model
```

Create a relaxed log-normal clock model

Description

Create a relaxed log-normal clock model

Usage

```
create_rln_clock_model(
  id = NA,
  mean_rate_prior_distr = create_uniform_distr(),
  ucldstdev_distr = create_gamma_distr(),
  mparam_id = NA,
  mean_clock_rate = "1.0",
  n_rate_categories = -1,
  normalize_mean_clock_rate = FALSE,
  dimension = NA
)
```

Arguments

<code>id</code>	an alignment's IDs. An ID can be extracted from its FASTA filename with get_alignment_ids_from_fasta_filenames
<code>mean_rate_prior_distr</code>	the mean clock rate prior distribution, as created by a create_distr function
<code>ucldstdev_distr</code>	the standard deviation of the uncorrelated log-normal distribution, as created by a create_distr function
<code>mparam_id</code>	the ID of the M parameter in the branchRateModel, set to NA to have it initialized
<code>mean_clock_rate</code>	the mean clock rate, 1.0 by default (is called <code>uclid_stdev</code> in XML, where <code>uclid_stdev</code> is always 0.1)
<code>n_rate_categories</code>	the number of rate categories. -1 is default, 0 denotes as much rates as branches
<code>normalize_mean_clock_rate</code>	normalize the mean clock rate
<code>dimension</code>	the dimensionality of the relaxed clock model. Leave NA to let <code>beautier</code> calculate it. Else, the dimensionality of the clock equals twice the number of taxa minus two.

Value

a relaxed log-normal clock_model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
rln_clock_model <- create_rln_clock_model()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  clock_model = rln_clock_model
)
testit::assert(file.exists(beast2_input_file))

rln_clock_model_exp <- create_rln_clock_model(
  mean_rate_prior_distr = create_exp_distr()
)

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  clock_model = rln_clock_model_exp
)
testit::assert(file.exists(beast2_input_file))
```

create_scale_param *Create a parameter called scale*

Description

Create a parameter called scale

Usage

```
create_scale_param(id = NA, value = 0)
```

Arguments

id	the parameter's ID
value	value of the parameter

Value

a parameter called scale

Note

this parameter is used in a Laplace distribution (as returned by [create_laplace_distr](#))

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
scale_param <- create_scale_param()

# Use the parameter in a distribution
laplace_distr <- create_laplace_distr(
  scale = scale_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = laplace_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_screenlog	<i>Create a screenlog object</i>
------------------	----------------------------------

Description

Create a screenlog object

Usage

```
create_screenlog(
  filename = "",
  log_every = 1000,
  mode = "autodetect",
  sanitise_headers = FALSE,
  sort = "none"
)
```

Arguments

filename	name of the file to store the posterior screens phylogenies to. By default, this is <code>\$(screen).screens</code>
log_every	number of MCMC states between writing to file
mode	mode how to log. Valid values are the ones returned by get_log_modes
sanitise_headers	set to TRUE to sanitise the headers of the log file
sort	how to sort the log. Valid values are the ones returned by get_log_sorts

`create_screenlog_xml` *Creates the screenlog section of the logger section of a BEAST2 XML parameter file*

Description

Creates the screenlog section of the logger section of a BEAST2 XML parameter file

Usage

```
create_screenlog_xml(inference_model = create_inference_model())
```

Arguments

`inference_model`
 a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Value

the XML text

Author(s)

Richèl J.C. Bilderbeek

create_sigma_param *Create a parameter called sigma*

Description

Create a parameter called sigma

Usage

```
create_sigma_param(id = NA, value = 1)
```

Arguments

id	the parameter's ID
value	value of the parameter

Value

a parameter called sigma

Note

this parameter is used in a normal distribution (as returned by [create_normal_distr](#))

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
sigma_param <- create_sigma_param()

# Use the parameter in a distribution
normal_distr <- create_normal_distr(
  sigma = sigma_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = normal_distr
```

```

    )
  )
  testit::assert(file.exists(beast2_input_file))

```

create_site_model *General function to create a site model.*

Description

General function to create a site model.

Usage

```
create_site_model(name, id, gamma_site_model = create_gamma_site_model(), ...)
```

Arguments

name	the site model name. Valid names can be found in <code>get_site_model_names</code>
id	the IDs of the alignment (can be extracted from the FASTA filename using get_alignment_id)
gamma_site_model	a gamma site model, as created by create_gamma_site_model
...	specific site model parameters

Value

a `site_model`

Note

Prefer using the named functions [create_gtr_site_model](#), [create_hky_site_model](#), [create_jc69_site_model](#), and [create_tn93_site_model](#)

Author(s)

Richèl J.C. Bilderbeek

See Also

See [create_gtr_site_model](#) for more examples with a GTR site model. See [create_hky_site_model](#) for more examples with an HKY site model. See [create_jc69_site_model](#) for more examples with a JC69 site model. See [create_tn93_site_model](#) for more examples with a TN93 site model

Examples

```
library(testthat)

input_filename <- get_fasta_filename()

# GTR
output_filename <- tempfile(pattern = "example_gtr", fileext = ".xml")
create_beast2_input_file(
  input_filename = input_filename,
  output_filename = output_filename,
  site_model = create_gtr_site_model()
)
expect_true(file.exists(output_filename))

# HKY
output_filename <- tempfile(pattern = "example_hky", fileext = ".xml")
create_beast2_input_file(
  input_filename = input_filename,
  output_filename = output_filename,
  site_model = create_hky_site_model()
)
expect_true(file.exists(output_filename))

# JC69
output_filename <- tempfile(pattern = "example_jc69", fileext = ".xml")
create_beast2_input_file(
  input_filename = input_filename,
  output_filename = output_filename,
  site_model = create_jc69_site_model()
)
expect_true(file.exists(output_filename))

# TN93
output_filename <- tempfile(pattern = "example_tn93", fileext = ".xml")
create_beast2_input_file(
  input_filename = input_filename,
  output_filename = output_filename,
  site_model = create_tn93_site_model()
)
expect_true(file.exists(output_filename))
```

create_site_models	<i>Creates all supported site models which is a list of the types returned by create_gtr_site_model, create_hky_site_model, create_jc69_site_model and create_tn93_site_model</i>
--------------------	---

Description

Creates all supported site models which is a list of the types returned by [create_gtr_site_model](#), [create_hky_site_model](#), [create_jc69_site_model](#) and [create_tn93_site_model](#)

Usage

```
create_site_models()
```

Value

a list of site_models

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a site model

Examples

```
library(testthat)

# All created site models are a kind of site model
site_models <- create_site_models()
expect_true(is_gtr_site_model(site_models[[1]]))
expect_true(is_hky_site_model(site_models[[2]]))
expect_true(is_jc69_site_model(site_models[[3]]))
expect_true(is_tn93_site_model(site_models[[4]]))

# Names are conformant
for (site_model in site_models) {
  expect_true(site_model$name %in% get_site_model_names())
}
```

create_site_models_from_names

Create site models from their names

Description

Create site models from their names

Usage

```
create_site_models_from_names(site_model_names)
```

Arguments

site_model_names

one or more names of a site model, must be name among those returned by [get_site_model_names](#)

Value

one or more site models

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a site model

Examples

```
names <- get_site_model_names()
site_models <- create_site_models_from_names(names)

for (i in seq_along(names)) {
  testthat::expect_equal(names[i], site_models[[i]]$name)
}
```

create_site_model_from_name

Create a site model from name

Description

Create a site model from name

Usage

```
create_site_model_from_name(site_model_name)
```

Arguments

site_model_name

name of a site model, must be a name as returned by [get_site_model_names](#)

Value

a site model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_model](#) to create a site model

Examples

```
site_model_names <- get_site_model_names()
for (site_model_name in site_model_names) {
  site_model <- create_site_model_from_name(site_model_name)
  testthat::expect_equal(site_model_name, site_model$name)
}
```

create_strict_clock_model

Create a strict clock model

Description

Create a strict clock model

Usage

```
create_strict_clock_model(
  id = NA,
  clock_rate_param = create_clock_rate_param(),
  clock_rate_distr = create_uniform_distr()
)
```

Arguments

id an alignment's IDs. An ID can be extracted from its FASTA filename with [get_alignment_ids_from_fasta_filenames](#)

clock_rate_param the clock rate's parameter, a numeric value. For advanced usage, use the structure as created by the [create_clock_rate_param](#) function

clock_rate_distr the clock rate's distribution, as created by a [create_distr](#) function

Value

a strict clock_model

Author(s)

Richèl J.C. Bilderbeek

Examples

```

strict_clock_model <- create_strict_clock_model(
  clock_rate_param = 1.0,
  clock_rate_distr = create_uniform_distr()
)

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  clock_model = strict_clock_model
)
testit::assert(file.exists(beast2_input_file))

strict_clock_model_gamma <- create_strict_clock_model(
  clock_rate_distr = create_gamma_distr()
)

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  clock_model = strict_clock_model_gamma
)
testit::assert(file.exists(beast2_input_file))

```

create_s_param	<i>Create a parameter called s</i>
----------------	------------------------------------

Description

Create a parameter called s

Usage

```
create_s_param(id = NA, value = 0, lower = 0, upper = Inf)
```

Arguments

id	the parameter's ID
value	value of the parameter
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value
upper	upper value of the parameter

Value

a parameter called s

Note

this parameter is used in a log-normal distribution (as returned by [create_log_normal_distr](#))

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_param](#) contains a list of all parameters that can be created

Examples

```
# Create the parameter
s_param <- create_s_param()

# Use the parameter in a distribution
log_normal_distr <- create_log_normal_distr(
  s = s_param
)

# Use the distribution to create a BEAST2 input file
beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = log_normal_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_temp_screenlog_filename

Create a filename for a temporary screenlog file

Description

Create a filename for a temporary screenlog file

Usage

```
create_temp_screenlog_filename()
```

See Also

use [create_screenlog](#) to create a screenlog.

`create_temp_tracelog_filename`

Create a filename for a temporary tracelog file

Description

Create a filename for a temporary tracelog file

Usage

`create_temp_tracelog_filename()`

See Also

use [create_tracelog](#) to create a tracelog.

`create_temp_treelog_filename`

Create a filename for a temporary treelog file

Description

Create a filename for a temporary treelog file

Usage

`create_temp_treelog_filename()`

See Also

use [create_treelog](#) to create a treelog.

```
create_test_inference_model
```

Create a simple inference model with a short MCMC chain

Description

Create a simple inference model with a short MCMC chain

Usage

```
create_test_inference_model(
  site_model = create_jc69_site_model(),
  clock_model = create_strict_clock_model(),
  tree_prior = create_yule_tree_prior(),
  mrca_prior = NA,
  mcmc = create_test_mcmc(),
  beauti_options = create_beauti_options(),
  tipdates_filename = NA
)
```

Arguments

site_model	a site model, as returned by create_site_model
clock_model	a clock model, as returned by create_clock_model
tree_prior	a tree priors, as returned by create_tree_prior
mrca_prior	a Most Recent Common Ancestor prior, as returned by create_mrca_prior
mcmc	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.
beauti_options	one BEAUti options object, as returned by create_beauti_options
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

an inference model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_inference_model](#) to create the BEAST2 default inference model

Examples

```
library(testthat)

inference_model <- create_test_inference_model()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file_from_model(
  get_fasta_filename(),
  beast2_input_file,
  inference_model = inference_model
)
expect_true(file.exists(beast2_input_file))
```

create_test_mcmc	<i>Create an MCMC configuration for testing.</i>
------------------	--

Description

Create an MCMC configuration for testing.

Usage

```
create_test_mcmc(
  chain_length = 3000,
  store_every = 1000,
  pre_burnin = 0,
  n_init_attempts = 10,
  sample_from_prior = FALSE,
  tracelog = create_test_tracelog(),
  screenlog = create_test_screenlog(),
  treelog = create_test_treelog()
)
```

Arguments

chain_length	length of the MCMC chain
store_every	number of states the MCMC will process before the posterior's state will be saved to file. Use -1 or NA to use the default frequency.
pre_burnin	number of burn in samples taken before entering the main loop
n_init_attempts	number of initialization attempts before failing
sample_from_prior	set to TRUE to sample from the prior

tracelog a tracelog, as created by [create_tracelog](#)
screenlog a screenlog, as created by [create_screenlog](#)
treelog a treelog, as created by [create_treelog](#)

Value

an MCMC configuration

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_mcmc](#) to create a default BEAST2 MCMC

Examples

```
library(testthat)

# Create an MCMC chain with 50 states
mcmc <- create_test_mcmc()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  mcmc = mcmc
)
expect_true(file.exists(beast2_input_file))
```

create_test_ns_inference_model

Create an inference model to be tested by Nested Sampling

Description

Create an inference model to be tested by Nested Sampling

Usage

```
create_test_ns_inference_model(
  site_model = beautier::create_jc69_site_model(),
  clock_model = beautier::create_strict_clock_model(),
  tree_prior = beautier::create_yule_tree_prior(),
  mcmc = beautier::create_test_ns_mcmc()
)
```


Arguments

site_model	a site model, as returned by create_site_model
clock_model	a clock model, as returned by create_clock_model
tree_prior	a tree priors, as returned by create_tree_prior
mcmc	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.

create_test_ns_mcmc *Create an NS MCMC object for testing*

Description

Create an NS MCMC object for testing

Usage

```
create_test_ns_mcmc(
  chain_length = 2000,
  store_every = 1000,
  pre_burnin = 0,
  n_init_attempts = 3,
  particle_count = 1,
  sub_chain_length = 500,
  epsilon = 1e-12,
  tracelog = create_test_tracelog(),
  screenlog = create_test_screenlog(),
  treelog = create_test_treelog()
)
```

Arguments

chain_length	upper bound to the length of the MCMC chain
store_every	number of states the MCMC will process before the posterior's state will be saved to file. Use -1 or NA to use the default frequency.
pre_burnin	number of burn in samples taken before entering the main loop
n_init_attempts	number of initialization attempts before failing
particle_count	number of particles
sub_chain_length	sub-chain length
epsilon	epsilon
tracelog	a tracelog, as created by create_tracelog
screenlog	a screenlog, as created by create_screenlog
treelog	a treelog, as created by create_treelog

Value

an MCMC object

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_ns_mcmc](#) to create a default nested sampling MCMC

Examples

```
library(testthat)

mcmc <- create_test_ns_mcmc()

beast2_input_file <- tempfile(fileext = ".xml")

create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  mcmc = mcmc
)

expect_true(file.exists(beast2_input_file))
```

`create_test_screenlog` *Create a screenlog object*

Description

Create a screenlog object

Usage

```
create_test_screenlog(
  filename = create_temp_screenlog_filename(),
  log_every = 1000,
  mode = "autodetect",
  sanitise_headers = FALSE,
  sort = "none"
)
```

Arguments

filename	name of the file to store the posterior screens phylogenies to. By default, this is <code>\$(screen).screens</code>
log_every	number of MCMC states between writing to file
mode	mode how to log. Valid values are the ones returned by get_log_modes
sanitise_headers	set to TRUE to sanitise the headers of the log file
sort	how to sort the log. Valid values are the ones returned by get_log_sorts

create_test_tracelog *Create a tracelog object*

Description

Create a tracelog object

Usage

```
create_test_tracelog(
  filename = create_temp_tracelog_filename(),
  log_every = 1000,
  mode = "autodetect",
  sanitise_headers = TRUE,
  sort = "smart"
)
```

Arguments

filename	name of the file to store the posterior traces. Use NA to use the filename <code>[alignment_id].log</code> , where <code>alignment_id</code> is obtained using get_alignment_id
log_every	number of MCMC states between writing to file
mode	mode how to log. Valid values are the ones returned by get_log_modes
sanitise_headers	set to TRUE to sanitise the headers of the log file
sort	how to sort the log. Valid values are the ones returned by get_log_sorts

create_test_treelog *Create a treelog object*

Description

Create a treelog object

Usage

```
create_test_treelog(
  filename = create_temp_treelog_filename(),
  log_every = 1000,
  mode = "tree",
  sanitise_headers = FALSE,
  sort = "none"
)
```

Arguments

filename	name of the file to store the posterior trees
log_every	number of MCMC states between writing to file
mode	mode how to log. Valid values are the ones returned by get_log_modes
sanitise_headers	set to TRUE to sanitise the headers of the log file
sort	how to sort the log. Valid values are the ones returned by get_log_sorts

create_tn93_site_model
Create a TN93 site model

Description

Create a TN93 site model

Usage

```
create_tn93_site_model(
  id = NA,
  gamma_site_model = create_gamma_site_model(),
  kappa_1_param = create_kappa_1_param(),
  kappa_2_param = create_kappa_2_param(),
  kappa_1_prior_distr = create_log_normal_distr(m = 1, s = 1.25),
  kappa_2_prior_distr = create_log_normal_distr(m = 1, s = 1.25),
  freq_equilibrium = "estimated"
)
```

Arguments

- `id` the IDs of the alignment (can be extracted from the FASTA filename using [get_alignment_id](#))
- `gamma_site_model` a gamma site model, as created by [create_gamma_site_model](#)
- `kappa_1_param` the 'kappa 1' parameter, a numeric value. For advanced usage, use the structure as returned by [create_kappa_1_param](#)
- `kappa_2_param` the 'kappa 2' parameter, a numeric value. For advanced usage, use the structure as returned by [create_kappa_2_param](#)
- `kappa_1_prior_distr` the distribution of the kappa 1 prior, which is a log-normal distribution (as created by [create_log_normal_distr](#)) by default
- `kappa_2_prior_distr` the distribution of the kappa 2 prior, which is a log-normal distribution (as created by [create_log_normal_distr](#)) by default
- `freq_equilibrium` the frequency in which the rates are at equilibrium are either estimated, empirical or all_equal. [get_freq_equilibrium_names](#) returns the possible values for `freq_equilibrium`

Value

a TN93 `site_model`

Author(s)

Richèl J.C. Bilderbeek

Examples

```
tn93_site_model <- create_tn93_site_model(
  kappa_1_param = 2.0,
  kappa_2_param = 2.0
)

create_beast2_input_file(
  input_filename = get_fasta_filename(),
  output_filename = tempfile(pattern = "beast", fileext = ".xml"),
  site_model = tn93_site_model
)
```

create_tracelog	<i>Create a tracelog object</i>
-----------------	---------------------------------

Description

Create a tracelog object

Usage

```
create_tracelog(
  filename = NA,
  log_every = 1000,
  mode = "autodetect",
  sanitise_headers = TRUE,
  sort = "smart"
)
```

Arguments

filename	name of the file to store the posterior traces. Use NA to use the filename <code>[alignment_id].log</code> , where <code>alignment_id</code> is obtained using get_alignment_id
log_every	number of MCMC states between writing to file
mode	mode how to log. Valid values are the ones returned by get_log_modes
sanitise_headers	set to TRUE to sanitise the headers of the log file
sort	how to sort the log. Valid values are the ones returned by get_log_sorts

create_tracelog_xml	<i>Creates the tracelog section of the logger section of a BEAST2 XML parameter file</i>
---------------------	--

Description

Creates the tracelog section of the logger section of a BEAST2 XML parameter file

Usage

```
create_tracelog_xml(input_filename, inference_model)
```

Arguments

- `input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.
- `inference_model` a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Author(s)

Richèl J.C. Bilderbeek

`create_trait_set_string`

Create a trait set string.

Description

For example, a data frame with row A 1 and another row B 2, the trait set string will be A=1 ,B=2

Usage

```
create_trait_set_string(df)
```

Arguments

`df` a data frame with two columns

Value

the trait set string

Author(s)

Richèl J.C. Bilderbeek

create_treelog	<i>Create a treelog object</i>
----------------	--------------------------------

Description

Create a treelog object

Usage

```
create_treelog(
  filename = "$(tree).trees",
  log_every = 1000,
  mode = "tree",
  sanitise_headers = FALSE,
  sort = "none"
)
```

Arguments

filename	name of the file to store the posterior trees
log_every	number of MCMC states between writing to file
mode	mode how to log. Valid values are the ones returned by get_log_modes
sanitise_headers	set to TRUE to sanitise the headers of the log file
sort	how to sort the log. Valid values are the ones returned by get_log_sorts

create_treelog_xml	<i>Creates the XML text for the logger tag with ID treelog. This section has these elements:</i>
--------------------	--

```
<logger id="treelog.t:test_output_0"
spec="Logger" fileName="my_treelog.trees"
logEvery="345000" mode="tree" sanitiseHeaders="true"
sort="smart"> # nolint indeed long <log
id="TreeWithMetaDataLogger.t:test_output_0"
spec="beast.evolution.tree.TreeWithMetaDataLogger"
tree="@Tree.t:test_output_0"/> # nolint indeed long
</logger>
```

Description

Creates the XML text for the logger tag with ID treelog. This section has these elements:

```
<logger id="treelog.t:test_output_0" spec="Logger" fileName="my_treelog.trees" logEvery="345000" mode="tree"
  <log id="TreeWithMetaDataLogger.t:test_output_0" spec="beast.evolution.tree.TreeWithMetaDataLogger"
  </logger>
```


Usage

```
create_treelog_xml(inference_model)
```

Arguments

inference_model

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Author(s)

Richèl J.C. Bilderbeek

create_tree_prior *Internal function to create a tree prior*

Description

Internal function to create a tree prior

Usage

```
create_tree_prior(name, id, ...)
```

Arguments

name	the tree prior name. Can be any name in <code>get_tree_prior_names</code>
id	the ID of the alignment
...	specific tree prior parameters

Value

a tree_prior

Note

Prefer the use the named functions [create_bd_tree_prior](#), [create_cbs_tree_prior](#), [create_ccp_tree_prior](#), [create_cep_tree_prior](#) and [create_yule_tree_prior](#) instead

Author(s)

Richèl J.C. Bilderbeek

See Also

See [create_bd_tree_prior](#), [create_cbs_tree_prior](#), [create_ccp_tree_prior](#), [create_cep_tree_prior](#) and [create_yule_tree_prior](#) for more examples using those functions

Examples

```

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_bd_tree_prior()
)
testit::assert(file.exists(beast2_input_file))

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_beautier_path("test_output_6.fas"),
  beast2_input_file,
  tree_prior = create_cbs_tree_prior()
)
testit::assert(file.exists(beast2_input_file))

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_ccp_tree_prior()
)
testit::assert(file.exists(beast2_input_file))

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_cep_tree_prior()
)
testit::assert(file.exists(beast2_input_file))

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior()
)
testit::assert(file.exists(beast2_input_file))

```

create_tree_priors	<i>Creates all supported tree priors, which is a list of the types returned by create_bd_tree_prior, create_cbs_tree_prior, create_ccp_tree_prior, create_cep_tree_prior and create_yule_tree_prior</i>
--------------------	---

Description

Creates all supported tree priors, which is a list of the types returned by [create_bd_tree_prior](#), [create_cbs_tree_prior](#), [create_ccp_tree_prior](#), [create_cep_tree_prior](#) and [create_yule_tree_prior](#)

Usage

```
create_tree_priors()
```

Value

a list of tree_priors

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

tree_priors <- create_tree_priors()
expect_true(is_bd_tree_prior(tree_priors[[1]]))
expect_true(is_cbs_tree_prior(tree_priors[[2]]))
expect_true(is_ccp_tree_prior(tree_priors[[3]]))
expect_true(is_cep_tree_prior(tree_priors[[4]]))
expect_true(is_yule_tree_prior(tree_priors[[5]]))
```

create_uniform_distr *Create a uniform distribution*

Description

Create a uniform distribution

Usage

```
create_uniform_distr(id = NA, upper = Inf)
```

Arguments

id	the distribution's ID
upper	an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf.

Value

a uniform distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

the function [create_distr](#) shows an overview of all supported distributions

Examples

```
uniform_distr <- create_uniform_distr()

beast2_input_file <- tempfile(fileext = ".xml")
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = uniform_distr
  )
)
testit::assert(file.exists(beast2_input_file))
```

create_xml_declaration

Create the XML declaration as XML text

Description

Create the XML declaration as XML text

Usage

```
create_xml_declaration()
```

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

create_yule_tree_prior
Create a Yule tree prior

Description

Create a Yule tree prior

Usage

```
create_yule_tree_prior(  
  id = NA,  
  birth_rate_distr = create_uniform_distr()  
)
```

Arguments

`id` the ID of the alignment
`birth_rate_distr` the birth rate distribution, as created by a [create_distr](#) function

Value

a Yule tree_prior

Author(s)

Richèl J.C. Bilderbeek

See Also

An alignment ID can be extracted from its FASTA filename using [get_alignment_id](#)

Examples

```
yule_tree_prior <- create_yule_tree_prior()  
  
beast2_input_file <- tempfile(fileext = ".xml")  
create_beast2_input_file(  
  input_filename = get_fasta_filename(),  
  beast2_input_file,  
  tree_prior = yule_tree_prior  
)  
testit::assert(file.exists(beast2_input_file))
```

default_parameters_doc

Documentation of parameters (for example, create_param. This function does nothing. It is intended to inherit documentation from.

Description

Documentation of parameters (for example, create_param. This function does nothing. It is intended to inherit documentation from.

Usage

```
default_parameters_doc(estimate, id, lower, name, upper, value, ...)
```

Arguments

estimate	TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise
id	the parameter's ID
lower	lowest possible value of the parameter. If the parameter is estimated, lower must be less than value
name	the parameters' name. Valid names can be found in get_param_names
upper	upper value of the parameter
value	value of the parameter
...	specific parameter parameters

Note

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

Author(s)

Richèl J.C. Bilderbeek

default_params_doc

Documentation of general function arguments. This function does nothing. It is intended to inherit function argument documentation.

Description

Documentation of general function arguments. This function does nothing. It is intended to inherit function argument documentation.

Usage

```
default_params_doc(  
    alignment_id,  
    bd_tree_prior,  
    cbs_tree_prior,  
    beast2_version,  
    beauti_options,  
    ccp_tree_prior,  
    cep_tree_prior,  
    chain_length,  
    clock_model,  
    clock_model_name,  
    clock_model_names,  
    clock_models,  
    clock_prior_distr_id,  
    crown_age,  
    crown_ages,  
    distr_id,  
    fasta_filename,  
    fasta_filenames,  
    fixed_crown_age,  
    fixed_crown_ages,  
    gamma_site_model,  
    group_sizes_dimension,  
    gtr_site_model,  
    has_non_strict_clock_model,  
    has_tip_dating,  
    hky_site_model,  
    id,  
    ids,  
    inference_model,  
    inference_models,  
    initial_phylogenies,  
    input_filename,  
    input_filenames,  
    is_monophyletic,  
    jc69_site_model,  
    log_every,  
    mcmc,  
    mode,  
    mrca_prior,  
    mrca_priors,  
    mrca_prior_name,  
    n_init_attempts,  
    output_filename,  
    param,  
    param_id,  
    phylogeny,
```

```

posterior_crown_age,
pre_burnin,
rename_fun,
rln_clock_model,
sample_from_prior,
sanitise_headers,
screenlog,
sequence_length,
site_model,
site_model_name,
site_model_names,
site_models,
sort,
store_every,
strict_clock_model,
taxa_names,
tipdates_filename,
tn93_site_model,
tracelog,
treelog,
tree_prior,
tree_prior_name,
tree_prior_names,
tree_priors,
verbose,
yule_tree_prior
)

```

Arguments

`alignment_id` ID of the alignment, as returned by [get_alignment_id](#). Keep at NA to have it initialized automatically

`bd_tree_prior` a Birth-Death tree prior, as created by [create_bd_tree_prior](#)

`cbs_tree_prior` a Coalescent Bayesian Skyline tree prior, as returned by [create_cbs_tree_prior](#)

`beast2_version` BEAST2 version, for example, code "2.5"

`beauti_options` one BEAUti options object, as returned by [create_beauti_options](#)

`ccp_tree_prior` a Coalescent Constant Population tree prior, as returned by [create_ccp_tree_prior](#)

`cep_tree_prior` a Coalescent Exponential Population tree prior, as returned by [create_cep_tree_prior](#)

`chain_length` length of the MCMC chain

`clock_model` a clock model, as returned by [create_clock_model](#)

`clock_model_name` name of a clock model, must be a name as returned by [get_clock_model_names](#)

`clock_model_names` one or more names of a clock model, must be name among those returned by [get_clock_model_names](#)

`clock_models` a list of one or more clock models, as returned by [create_clock_model](#)

clock_prior_distr_id	ID of an MRCA clock model's distribution. Keep at NA to have it initialized automatically
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
distr_id	a distributions' ID
fasta_filename	a FASTA filename. Use get_fasta_filename to obtain a testing FASTA filename.
fasta_filenames	One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename.
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.
gamma_site_model	a site model's gamma site model, as returned by create_gamma_site_model
group_sizes_dimension	the group sizes' dimension, as used by the CBS tree prior (see create_cbs_tree_prior)
gtr_site_model	a GTR site model, as returned by create_gtr_site_model
has_non_strict_clock_model	boolean to indicate that there is already at least one non-strict (i.e. relaxed log-normal) clock model
has_tip_dating	TRUE if the user has supplied tip dates, FALSE otherwise
hky_site_model	an HKY site model, as returned by create_hky_site_model
id	an alignment's IDs. An ID can be extracted from its FASTA filename with get_alignment_ids_from_fasta_filenames)
ids	one or more alignments' IDs. IDs can be extracted from their FASTA filenames with get_alignment_ids_from_fasta_filenames)
inference_model	a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use create_inference_model to create an inference model. Use check_inference_model to check if an inference model is valid. Use rename_inference_model_filenames to rename the files in an inference model.
inference_models	a list of one or more inference models, as can be created by create_inference_model
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 phylo

<code>input_filename</code>	A FASTA filename. Use get_fasta_filename to obtain a testing FASTA filename.
<code>input_filenames</code>	One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename.
<code>is_monophyletic</code>	boolean to indicate monophyly is assumed in a Most Recent Common Ancestor prior, as returned by create_mrca_prior
<code>jc69_site_model</code>	a JC69 site model, as returned by create_jc69_site_model
<code>log_every</code>	number of MCMC states between writing to file
<code>mcmc</code>	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.
<code>mode</code>	mode how to log. Valid values are the ones returned by get_log_modes
<code>mrca_prior</code>	a Most Recent Common Ancestor prior, as returned by create_mrca_prior
<code>mrca_priors</code>	a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior
<code>mrca_prior_name</code>	the unique name of the MRCA prior, for example a genus, family, order or even class name. Leave at <code>NA</code> to have it named automatically.
<code>n_init_attempts</code>	number of initialization attempts before failing
<code>output_filename</code>	Name of the XML parameter file created by this function. BEAST2 uses this file as input.
<code>param</code>	a parameter, as can be created by create_param .
<code>param_id</code>	a parameter's ID
<code>phylogeny</code>	a phylogeny of type phylo
<code>posterior_crown_age</code>	deprecated
<code>pre_burnin</code>	number of burn in samples taken before entering the main loop
<code>rename_fun</code>	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 <code>NA</code> . The function should return one filename (when passed one filename) or one <code>NA</code> (when passed one <code>NA</code>). Example rename functions are: <ul style="list-style-type: none"> • 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, <code>tracelog_82c1a522040.log</code> becomes <code>tracelog.log</code>

rln_clock_model	a Relaxed Log-Normal clock model, as returned by create_rln_clock_model
sample_from_prior	set to TRUE to sample from the prior
sanitise_headers	set to TRUE to sanitise the headers of the log file
screenlog	a screenlog, as created by create_screenlog
sequence_length	a DNA sequence length, in base pairs
site_model	a site model, as returned by create_site_model
site_model_name	name of a site model, must be a name as returned by get_site_model_names
site_model_names	one or more names of a site model, must be name among those returned by get_site_model_names
site_models	one or more site models, as returned by create_site_model
sort	how to sort the log. Valid values are the ones returned by get_log_sorts
store_every	number of states the MCMC will process before the posterior's state will be saved to file. Use -1 or NA to use the default frequency.
strict_clock_model	a strict clock model, as returned by create_strict_clock_model
taxa_names	names of the taxa, as returned by get_taxa_names . Keep at NA to have it initialized automatically, using all taxa in the alignment
tipdates_filename	name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.
tn93_site_model	a TN93 site model, as returned by create_tn93_site_model
tracelog	a tracelog, as created by create_tracelog
treelog	a treelog, as created by create_treelog
tree_prior	a tree priors, as returned by create_tree_prior
tree_prior_name	name of a tree prior, must be a name as returned by get_tree_prior_names
tree_prior_names	one or more names of a tree prior, must be a name among those returned by get_tree_prior_names
tree_priors	one or more tree priors, as returned by create_tree_prior
verbose	if TRUE , additional information is displayed, that is potentially useful in debugging
yule_tree_prior	a Yule tree_prior, as created by create_yule_tree_prior

Note

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

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml	<i>Converts a distribution to XML</i>
--------------	---------------------------------------

Description

Converts a distribution to XML

Usage

```
distr_to_xml(distr)
```

Arguments

distr a distribution, as created by [create_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

xml <- distr_to_xml(create_uniform_distr(id = 1))
expect_true(is.character(xml))
expect_true(length(xml) == 1)
expect_true(nchar(xml) > 1)
```

distr_to_xml_beta *Converts a beta distribution to XML*

Description

Converts a beta distribution to XML

Usage

```
distr_to_xml_beta(distr)
```

Arguments

distr a beta distribution, as created by [create_beta_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_exp *Converts an exponential distribution to XML*

Description

Converts an exponential distribution to XML

Usage

```
distr_to_xml_exp(distr)
```

Arguments

distr an exponential distribution, as created by [create_exp_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_gamma *Converts a gamma distribution to XML*

Description

Converts a gamma distribution to XML

Usage

```
distr_to_xml_gamma(distr)
```

Arguments

distr a gamma distribution, as created by [create_gamma_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_inv_gamma *Converts an inverse-gamma distribution to XML*

Description

Converts an inverse-gamma distribution to XML

Usage

```
distr_to_xml_inv_gamma(distr)
```

Arguments

distr an inverse-gamma distribution, as created by [create_inv_gamma_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_laplace *Converts a Laplace distribution to XML*

Description

Converts a Laplace distribution to XML

Usage

```
distr_to_xml_laplace(distr)
```

Arguments

distr a Laplace distribution as created by [create_laplace_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_log_normal
Converts a log-normal distribution to XML

Description

Converts a log-normal distribution to XML

Usage

```
distr_to_xml_log_normal(distr)
```

Arguments

distr a log-normal distribution, as created by [create_log_normal_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_normal *Converts a normal distribution to XML*

Description

Converts a normal distribution to XML

Usage

```
distr_to_xml_normal(distr)
```

Arguments

distr a normal distribution, as created by [create_normal_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_one_div_x
Converts a 1/x distribution to XML

Description

Converts a 1/x distribution to XML

Usage

```
distr_to_xml_one_div_x(distr)
```

Arguments

distr a 1/x distribution, as created by [create_one_div_x_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_poisson *Converts a Poisson distribution to XML*

Description

Converts a Poisson distribution to XML

Usage

```
distr_to_xml_poisson(distr)
```

Arguments

distr a Poisson distribution, as created by [create_poisson_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_uniform *Converts a uniform distribution to XML*

Description

Converts a uniform distribution to XML

Usage

```
distr_to_xml_uniform(distr)
```

Arguments

distr a uniform distribution, as created by [create_uniform_distr](#))

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

extract_xml_loggers_from_lines

Extract everything between first loggers and last loggers line

Description

Extract everything between first loggers and last loggers line

Usage

```
extract_xml_loggers_from_lines(lines)
```

Arguments

lines lines of text

Value

lines of text from the first to and including the last operators line

Author(s)

Richèl J.C. Bilderbeek

extract_xml_operators_from_lines

Extract everything between first operators and last operators line

Description

Extract everything between first operators and last operators line

Usage

```
extract_xml_operators_from_lines(lines)
```

Arguments

lines lines of text

Value

lines of text from the first to and including the last operators line

Author(s)

Richèl J.C. Bilderbeek

`extract_xml_section_from_lines`*Get the lines of an XML section, including the section tags*

Description

Get the lines of an XML section, including the section tags

Usage

```
extract_xml_section_from_lines(lines, section)
```

Arguments

<code>lines</code>	lines of the XML text
<code>section</code>	the XML section name

Value

the section's lines of XML text, including the tags

Author(s)

Richèl J.C. Bilderbeek

`fasta_file_to_sequences`*Convert a FASTA file to a table of sequences*

Description

Convert a FASTA file to a table of sequences

Usage

```
fasta_file_to_sequences(fasta_filename)
```

Arguments

`fasta_filename` One existing FASTA filenames

Value

a table of sequences

Author(s)

Richèl J.C. Bilderbeek

find_clock_model *Finds a clock model with a certain ID*

Description

Finds a clock model with a certain ID

Usage

```
find_clock_model(clock_models, id)
```

Arguments

clock_models a list of one or more clock models, as returned by [create_clock_model](#)
id the ID of the clock model

Value

the clock models with the desired ID, NULL if such a clock model is absent

Author(s)

Richèl J.C. Bilderbeek

find_first_regex_line *Find the first line that satisfies a regex*

Description

Find the first line that satisfies a regex

Usage

```
find_first_regex_line(lines, regex)
```

Arguments

lines lines of text
regex the regex as text

Value

index of the line

Author(s)

Richèl J.C. Bilderbeek

`find_first_xml_opening_tag_line`

Find the line number of the first section's opening tag

Description

Find the line number of the first section's opening tag

Usage

`find_first_xml_opening_tag_line(lines, section)`

Arguments

<code>lines</code>	the lines of an XML text
<code>section</code>	the name of the XML section

Value

the line number's index (which is 1 for the first line) if the opening tag is found, else NA

Author(s)

Richèl J.C. Bilderbeek

`find_last_regex_line` *Find the index of the last line that matches a regex*

Description

Find the index of the last line that matches a regex

Usage

`find_last_regex_line(lines, regex)`

Arguments

<code>lines</code>	lines of text
<code>regex</code>	regex string

Value

index of the line

Author(s)

Richèl J.C. Bilderbeek

`find_last_xml_closing_tag_line`*Find the highest line number of a section's closing tag*

Description

Find the highest line number of a section's closing tag

Usage

```
find_last_xml_closing_tag_line(lines, section)
```

Arguments

lines	the lines of an XML text
section	the name of the XML section

Value

the line number's index (which is 1 for the first line) if the opening tag is found, else NA

Author(s)

Richèl J.C. Bilderbeek

`freq_equilibrium_to_xml`*Creates the freq_equilibrium as XML*

Description

Creates the freq_equilibrium as XML

Usage

```
freq_equilibrium_to_xml(freq_equilibrium, id)
```

Arguments

freq_equilibrium	
id	a freq_equilibrium name
	a site model's name

Value

the freq_equilibrium as XML

Author(s)

Richèl J.C. Bilderbeek

`gamma_site_models_to_xml_prior_distr`

Creates the gamma site models section in the distribution section of a BEAST2 XML parameter file

Description

Creates the gamma site models section in the distribution section of a BEAST2 XML parameter file

Usage

`gamma_site_models_to_xml_prior_distr(site_models)`

Arguments

`site_models` one or more site models, as returned by [create_site_model](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

`gamma_site_model_to_xml_prior_distr`

Creates the gamma site models section in the distribution section of a BEAST2 XML parameter file

Description

Creates the gamma site models section in the distribution section of a BEAST2 XML parameter file

Usage

`gamma_site_model_to_xml_prior_distr(site_model)`

Arguments

`site_model` a site model, as returned by [create_site_model](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

gamma_site_model_to_xml_state

Converts a gamma site model to XML, used in the state section

Description

Converts a gamma site model to XML, used in the state section

Usage

```
gamma_site_model_to_xml_state(gamma_site_model, id)
```

Arguments

gamma_site_model	a gamma site model, as created by create_gamma_site_model)
id	the site model's ID

Value

the gamma_site model as XML text

Author(s)

Richèl J.C. Bilderbeek

get_alignment_id

Conclude the ID from a FASTA filename.

Description

This is done in the same way as BEAST2 will do so.

Usage

```
get_alignment_id(fasta_filename, capitalize_first_char_id = FALSE)
```


Arguments

fasta_filename a FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.
capitalize_first_char_id
if TRUE, the first character will be capitalized

Value

an alignment's ID

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [check_alignment_id](#) to check if an alignment ID is valid.

Examples

```
library(testthat)

# Path need not exist, use UNIX path as example
created <- get_alignment_id("/home/homer/anthus_aco_sub.fas")
expected <- "anthus_aco_sub"
expect_equal(created, expected)
expect_silent(check_alignment_id(created))
```

get_alignment_ids *Get the alignment IDs from one or more files.*

Description

This is done in the same way as BEAST2 does by default The file extension will be used to determine which type of file is worked on.

Usage

```
get_alignment_ids(filenamees)
```

Arguments

filenamees names of the files to be checked

Value

the IDs extracted from the one or more files

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [get_alignment_ids_from_fasta_filenames](#) to get the alignment IDs from files known to be FASTA files

Examples

```
created <- get_alignment_ids(  
  get_beautier_paths(c("anthus_aco.fas", "anthus_nd2.fas"))  
)  
expected <- c(  
  get_alignment_id(get_beautier_path("anthus_aco.fas")),  
  get_alignment_id(get_beautier_path("anthus_nd2.fas"))  
)  
testit::assert(created == expected)
```

get_alignment_ids_from_fasta_filenames

Get the alignment ID from one or more FASTA filenames.

Description

This is done in the same way as BEAST2 does by default. The files are assumed to be FASTA. If this is not the case, there may be any kind of error message when calling this function.

Usage

```
get_alignment_ids_from_fasta_filenames(fasta_filenames)
```

Arguments

fasta_filenames

One or more FASTA filenames. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

Value

the IDs from one or more FASTA files

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [get_alignment_ids](#) to get the alignment IDs from multiple kids of files. Use [are_fasta_filenames](#) to see if the filenames all have a common FASTA filename extension.

Examples

```
created <- get_alignment_ids_from_fasta_filenames(  
  get_beautier_paths(c("anthus_aco.fas", "anthus_nd2.fas"))  
)  
expected <- c(  
  get_alignment_id(get_beautier_path("anthus_aco.fas")),  
  get_alignment_id(get_beautier_path("anthus_nd2.fas"))  
)  
testit::assert(created == expected)
```

get_beautier_path	<i>Get the full path of a file in the inst/extdata folder</i>
-------------------	---

Description

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

Usage

```
get_beautier_path(filename)
```

Arguments

filename the file's name, without the path

Value

the full path of the filename

Author(s)

Richèl J.C. Bilderbeek

See Also

for more files, use [get_beautier_paths](#)

Examples

```
library(testthat)  
  
expect_true(is.character(get_beautier_path("test_output_0.fas")))  
expect_true(is.character(get_beautier_path("anthus_aco.fas")))  
expect_true(is.character(get_beautier_path("anthus_nd2.fas")))
```

get_beautier_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_beautier_paths(filenamees)
```

Arguments

filenamees the files' names, without the path

Value

the filenamees' full paths

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [get_beautier_path](#) to get the path of one file
for one file, use [get_beautier_path](#)

Examples

```
testit::assert(  
  length(  
    get_beautier_paths(  
      c("test_output_0.fas", "anthus_aco.fas", "anthus_nd2.fas")  
    )  
  ) == 3  
)
```

`get_clock_models_ids` *Collect the IDs of the list of clock models*

Description

Collect the IDs of the list of clock models

Usage

```
get_clock_models_ids(clock_models)
```

Arguments

`clock_models` a list of one or more clock models, as returned by [create_clock_model](#)

Value

IDs of the clock models

Author(s)

Richèl J.C. Bilderbeek

`get_clock_model_name` *Get the BEAUti name for a clock model*

Description

Will [stop](#) if the clock model is an invalid clock model

Usage

```
get_clock_model_name(clock_model)
```

Arguments

`clock_model` a clock model, as returned by [create_clock_model](#)

Value

name of the clock model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_equal(
  get_clock_model_name(create_strict_clock_model()),
  "StrictClock"
)

expect_equal(
  get_clock_model_name(create_rln_clock_model()),
  "RelaxedClock"
)
```

get_clock_model_names *Get the clock model names*

Description

Get the clock model names

Usage

```
get_clock_model_names()
```

Value

the clock model names

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_clock_models](#) to create a list with all clock models

Examples

```
library(testthat)

names <- get_clock_model_names()
expect_true("relaxed_log_normal" %in% names)
expect_true("strict" %in% names)
```

get_crown_age	<i>Obtain the crown age of a phylogeny.</i>
---------------	---

Description

The crown age of a phylogeny is the time between the present and the moment of at which the first diversification (resulting in two lineages) happened.

Usage

```
get_crown_age(phylogeny)
```

Arguments

phylogeny The phylogeny to obtain the crown age of

Value

the crown age of the phylogeny

Author(s)

Richèl J.C. Bilderbeek

Examples

```
phylogeny <- ape::read.tree(text = "(a:15,b:15):1;")
created <- get_crown_age(phylogeny = phylogeny)
testit::assert(created == 15)
```

get_distr_names	<i>Get the distribution names</i>
-----------------	-----------------------------------

Description

Get the distribution names

Usage

```
get_distr_names()
```

Value

the distribution names

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

names <- get_distr_names()

expect_true("uniform" %in% names)
expect_true("normal" %in% names)
expect_true("one_div_x" %in% names)
expect_true("log_normal" %in% names)
expect_true("exponential" %in% names)
expect_true("gamma" %in% names)
expect_true("beta" %in% names)
expect_true("laplace" %in% names)
expect_true("inv_gamma" %in% names)
expect_true("poisson" %in% names)
```

get_distr_n_params *Get the number of parameters a distribution uses*

Description

Get the number of parameters a distribution uses

Usage

```
get_distr_n_params(distr)
```

Arguments

distr a distribution, as created by [create_distr](#) or (preferable) its named functions

Value

the number of parameters that distribution uses

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)
expect_equal(get_distr_n_params(create_beta_distr()), 2)
expect_equal(get_distr_n_params(create_exp_distr()), 1)
expect_equal(get_distr_n_params(create_gamma_distr()), 2)
expect_equal(get_distr_n_params(create_inv_gamma_distr()), 2)
expect_equal(get_distr_n_params(create_laplace_distr()), 2)
expect_equal(get_distr_n_params(create_log_normal_distr()), 2)
expect_equal(get_distr_n_params(create_normal_distr()), 2)
expect_equal(get_distr_n_params(create_one_div_x_distr()), 0)
expect_equal(get_distr_n_params(create_poisson_distr()), 1)
expect_equal(get_distr_n_params(create_uniform_distr()), 0)
```

get_fasta_filename	<i>Get the path of a FASTA file used in testing</i>
--------------------	---

Description

Get the path of a FASTA file used in testing

Usage

```
get_fasta_filename()
```

Value

the path of a FASTA file used in testing

Author(s)

Richèl J.C. Bilderbeek

Examples

```
filename <- beautier::get_fasta_filename()
testit::assert(file.exists(filename))

create_beast2_input_file(
  input_filename = filename,
  output_filename = tempfile(pattern = "beast", fileext = ".xml")
)
```

get_file_base_sans_ext

Get the base of the filename base without extension

Description

The path need not to actually exist

Usage

```
get_file_base_sans_ext(filename)
```

Arguments

filename A filename

Value

That filename without its full path and extension

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# Path need not exist, use UNIX path as example
expect_equal(
  get_file_base_sans_ext("/home/homer/test.txt"),
  "test"
)
```

get_freq_equilibrium_names

Returns valid values for the freq_equilibrium argument

Description

Returns valid values for the freq_equilibrium argument

Usage

```
get_freq_equilibrium_names()
```

Value

the valid values for the `freq_equilibrium` argument

Author(s)

Richèl J.C. Bilderbeek

See Also

the `freq_equilibrium` argument is used in [create_gtr_site_model](#), [create_hky_site_model](#), and [create_tn93_site_model](#)

Examples

```
library(testthat)

names <- get_freq_equilibrium_names()
expect_true("estimated" %in% names)
expect_true("empirical" %in% names)
expect_true("all_equal" %in% names)
```

`get_gamma_site_model_n_distrs`

Get the number of distributions in a gamma site model

Description

Get the number of distributions in a gamma site model

Usage

```
get_gamma_site_model_n_distrs(gamma_site_model)
```

Arguments

`gamma_site_model`
a site model's gamma site model, as returned by [create_gamma_site_model](#)

Value

the number of distributions a gamma site model has

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_gamma_site_model](#) to create a gamma site model

Examples

```
gamma_site_model <- create_gamma_site_model()
n_distrs <- get_gamma_site_model_n_distrs(
  gamma_site_model
)
testit::assert(n_distrs == 0)

gamma_site_model <- create_gamma_site_model(
  gamma_cat_count = 2,
  gamma_shape_prior_distr = create_exp_distr()
)
n_distrs <- get_gamma_site_model_n_distrs(gamma_site_model)
testit::assert(n_distrs == 1)
```

get_gamma_site_model_n_params

Get the number of distributions a site model has

Description

Get the number of distributions a site model has

Usage

```
get_gamma_site_model_n_params(gamma_site_model)
```

Arguments

gamma_site_model
a site model's gamma site model, as returned by [create_gamma_site_model](#)

Value

the number of parameters a site model has

Author(s)

Richèl J.C. Bilderbeek

Examples

```
testit::assert(
  get_gamma_site_model_n_params(
    create_gamma_site_model(gamma_cat_count = 0)
  ) == 0
)
testit::assert(
  get_gamma_site_model_n_params(
    create_gamma_site_model(gamma_cat_count = 1)
  ) == 1
)
```

```

    ) == 0
  )
  testit::assert(
    get_gamma_site_model_n_params(
      create_gamma_site_model(
        gamma_cat_count = 2,
        gamma_shape_prior_distr = create_exp_distr()
      )
    ) == 1
  )

```

get_has_non_strict_clock_model

Determines if there is at least one non-strict clock model in the list of one or more clock models

Description

Determines if there is at least one non-strict clock model in the list of one or more clock models

Usage

```
get_has_non_strict_clock_model(clock_models)
```

Arguments

clock_models a list of one or more clock models, as returned by [create_clock_model](#)

Value

TRUE if there is at least one non-strict clock model

Author(s)

Richèl J.C. Bilderbeek

get_inference_model_filenames

Get the filenames stored in an inference model.

Description

If there is no name for a tipdates file specified (as done by setting inference_model\$tipdates_filename to [NA](#), there will be one filename less returned

Usage

```
get_inference_model_filenames(inference_model)
```

Arguments

`inference_model`

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Examples

```
library(testthat)

inference_model <- create_inference_model()
inference_model$mcmc$tracelog$filename <- "/home/john/trace.log"
inference_model$mcmc$screenlog$filename <- "/home/john/screen.log"
inference_model$mcmc$treelog$filename <- "/home/john/tree.log"
inference_model$tipdates_filename <- "/home/john/tipdate.csv"

filenames <- get_inference_model_filenames(inference_model)

expect_equal(length(filenames), 4)
expect_true("/home/john/trace.log" %in% filenames)
expect_true("/home/john/screen.log" %in% filenames)
expect_true("/home/john/tree.log" %in% filenames)
expect_true("/home/john/tipdate.csv" %in% filenames)

#' Nope, no tip dates needed in my inference ...
inference_model$tipdates_filename <- NA
filenames <- get_inference_model_filenames(inference_model)

#' ... so one less file gets created
expect_equal(length(filenames), 3)
```

```
get_log_modes
```

```
Get the possible log modes
```

Description

Get the possible log modes

Usage

```
get_log_modes()
```

get_log_sorts	<i>Get the possible log sorts</i>
---------------	-----------------------------------

Description

Get the possible log sorts

Usage

```
get_log_sorts()
```

get_mcmc_filenames	<i>Get the filenames stored in an MCMC.</i>
--------------------	---

Description

If a filename is set to an empty string, to indicate a certain log file need not be created, this (non-)filename will not be returned.

Usage

```
get_mcmc_filenames(mcmc)
```

Arguments

mcmc	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.
------	---

Examples

```
library(testthat)

mcmc <- create_mcmc()
mcmc$tracelog$filename <- "/home/john/trace.log"
mcmc$screenlog$filename <- "/home/john/screen.log"
mcmc$treelog$filename <- "/home/john/tree.log"

filenames <- get_mcmc_filenames(mcmc)

expect_equal(length(filenames), 3)
expect_true("/home/john/trace.log" %in% filenames)
expect_true("/home/john/screen.log" %in% filenames)
expect_true("/home/john/tree.log" %in% filenames)

# If there is no need to write to the screenlog file ...
```

```
mcmc$screenlog$filename <- ""  
  
# ... one file less will be created  
filenames <- get_mcmc_filenames(mcmc)  
expect_equal(length(filenames), 2)
```

get_n_taxa

Extract the number of taxa from a file

Description

Extract the number of taxa from a file

Usage

```
get_n_taxa(filename)
```

Arguments

filename name of a FASTA file

Value

the number of taxa

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)  
  
fasta_filename <- get_beautier_path("test_output_5.fas")  
expect_equal(get_n_taxa(fasta_filename), 5)
```

get_operator_id_pre *Get the prefix of operator IDs*

Description

Get the prefix of operator IDs

Usage

```
get_operator_id_pre(tree_prior)
```

Arguments

tree_prior a tree priors, as returned by [create_tree_prior](#)

Value

the prefix of operator IDs, similar to the name of a tree prior

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

bd_pre <- get_operator_id_pre(
  tree_prior = create_bd_tree_prior()
)
expect_equal(bd_pre, "BirthDeath")
```

get_param_names *Get the parameter names*

Description

Get the parameter names

Usage

```
get_param_names()
```

Value

the parameter names

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

names <- get_param_names()
expect_true("alpha" %in% names)
expect_true("beta" %in% names)
expect_true("clock_rate" %in% names)
expect_true("kappa_1" %in% names)
expect_true("kappa_2" %in% names)
expect_true("lambda" %in% names)
expect_true("m" %in% names)
expect_true("mean" %in% names)
expect_true("mu" %in% names)
expect_true("rate_ac" %in% names)
expect_true("rate_ag" %in% names)
expect_true("rate_at" %in% names)
expect_true("rate_cg" %in% names)
expect_true("rate_ct" %in% names)
expect_true("rate_gt" %in% names)
expect_true("s" %in% names)
expect_true("scale" %in% names)
expect_true("sigma" %in% names)
```

get_remove_dir_fun	<i>Get a function that, from a filename, returns the part without the directory.</i>
--------------------	--

Description

Or: get a function that returns the local version of a filename. Also, the function will return [NA](#) if the filename is [NA](#)

Usage

```
get_remove_dir_fun()
```

Author(s)

Richèl J.C. Bilderbeek

See Also

see [check_rename_fun](#) for an overview of file renaming functions

get_remove_hex_fun *Get a function that removes the hex string from filenames.*

Description

The default filenames created by [beautier](#) are temporary files, such as `/home/john/.cache/tracelog_82c5888db98.log` (on Linux), where `/home/john/.cache` is the location to a temporary folder (on Linux) and `tracelog_82c5888db98.log` the filename. The filename ends with a hex string (as is common for temporary files, as [tempfile](#) does so). Because [beautier](#) puts an underscore between the filename description (tracelog) and the hex string, this function removes both.

Usage

```
get_remove_hex_fun()
```

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

f <- get_remove_hex_fun()
expect_equal(
  f("/home/john/beast2_186c7404208c.xml.state"),
  "/home/john/beast2.xml.state"
)
expect_equal(
  f("beast2_186c7404208c.xml.state"),
  "beast2.xml.state"
)
expect_equal(f(NA), NA)
```

get_replace_dir_fun *Get a function to replace the directory of a filename*

Description

Get a function to replace the directory of a filename

Usage

```
get_replace_dir_fun(new_dir_name = "")
```

Arguments

`new_dir_name` the new directory name

get_site_models_n_distrs

Get the number of distributions a site model has

Description

Get the number of distributions a site model has

Usage

```
get_site_models_n_distrs(site_models)
```

Arguments

site_models one or more site models, as returned by [create_site_model](#)

Value

the number of distributions the site models have

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_equal(get_site_models_n_distrs(list(create_gtr_site_model())), 5)
expect_equal(get_site_models_n_distrs(list(create_hky_site_model())), 1)
expect_equal(get_site_models_n_distrs(list(create_jc69_site_model())), 0)
expect_equal(get_site_models_n_distrs(list(create_tn93_site_model())), 2)
```

get_site_models_n_params

Get the number of distributions one or more site models have

Description

Get the number of distributions one or more site models have

Usage

```
get_site_models_n_params(site_models)
```

Arguments

site_models one or more site models, as returned by [create_site_model](#)

Value

the number of parameters the site models have

Author(s)

Richèl J.C. Bilderbeek

Examples

```
testit::assert(
  get_site_models_n_params(list(create_gtr_site_model())) == 10
)
testit::assert(
  get_site_models_n_params(list(create_hky_site_model())) == 2
)
testit::assert(
  get_site_models_n_params(list(create_jc69_site_model())) == 0
)
testit::assert(
  get_site_models_n_params(list(create_tn93_site_model())) == 4
)
```

get_site_model_names *Get the site models' names*

Description

Get the site models' names

Usage

```
get_site_model_names()
```

Value

the site model names

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_site_models](#) to get all site models

Examples

```
# Check all names
names <- get_site_model_names()
testit::assert("JC69" %in% names)
testit::assert("HKY" %in% names)
testit::assert("TN93" %in% names)
testit::assert("GTR" %in% names)
```

get_site_model_n_distrs

Get the number of distributions a site model has

Description

Get the number of distributions a site model has

Usage

```
get_site_model_n_distrs(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the number of distributions a site model has

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# gamma site model, rates AC, AG, AT, CG and GT
expect_equal(
  get_site_model_n_distrs(create_gtr_site_model()),
  5
)

# gamma site model, kappa
expect_equal(
  get_site_model_n_distrs(create_hky_site_model()),
  1
)

# gamma site model
```

```
expect_equal(  
  get_site_model_n_distrs(create_jc69_site_model()),  
  0  
)  
  
# gamma site model, kappa 1 and kappa 2  
expect_equal(  
  get_site_model_n_distrs(create_tn93_site_model()),  
  2  
)
```

get_site_model_n_params

Get the number of distributions a site model has

Description

Get the number of distributions a site model has

Usage

```
get_site_model_n_params(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the number of parameters a site model has

Author(s)

Richèl J.C. Bilderbeek

Examples

```
testit::assert(  
  get_site_model_n_params(create_gtr_site_model()) == 10  
)  
testit::assert(  
  get_site_model_n_params(create_hky_site_model()) == 2  
)  
testit::assert(  
  get_site_model_n_params(create_jc69_site_model()) == 0  
)  
testit::assert(  
  get_site_model_n_params(create_tn93_site_model()) == 4  
)
```

get_taxa_names *Extract the names of taxa from a file*

Description

Extract the names of taxa from a file

Usage

```
get_taxa_names(filename)
```

Arguments

filename name of a FASTA file

Value

the taxa names

Author(s)

Richèl J.C. Bilderbeek

Examples

```
created <- get_taxa_names(get_beautier_path("anthus_aco_sub.fas"))
expected <- c(
  "61430_aco", "626029_aco", "630116_aco", "630210_aco", "B25702_aco"
)
testit::assert(created == expected)
```

get_tree_priors_n_distrs *Get the number of distributions a tree prior has*

Description

Get the number of distributions a tree prior has

Usage

```
get_tree_priors_n_distrs(tree_priors)
```

Arguments

tree_priors one or more tree priors, as returned by [create_tree_prior](#)

Value

the number of distributions a tree prior has

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_equal(
  get_tree_priors_n_distrs(
    list(
      create_bd_tree_prior(), # has two distributions
      create_ccp_tree_prior() # has one distribution
    )
  ),
  3
)
```

get_tree_priors_n_params

Get the number of parameters a list of tree priors has

Description

Get the number of parameters a list of tree priors has

Usage

```
get_tree_priors_n_params(tree_priors)
```

Arguments

tree_priors one or more tree priors, as returned by [create_tree_prior](#)

Value

the number of parameters the tree priors have

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_equal(
  get_tree_priors_n_params(
    list(
      create_bd_tree_prior(), # zero
      create_cep_tree_prior() # two
    )
  ),
  2
)
```

get_tree_prior_names *Get the tree prior names*

Description

Get the tree prior names

Usage

```
get_tree_prior_names()
```

Value

the tree prior names

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tree_priors](#) to get all tree priors

Examples

```
library(testthat)

names <- get_tree_prior_names()
expect_true("birth_death" %in% names)
expect_true("coalescent_bayesian_skyline" %in% names)
expect_true("coalescent_constant_population" %in% names)
expect_true("coalescent_exp_population" %in% names)
expect_true("yule" %in% names)
```

`get_tree_prior_n_distrs`*Get the number of distributions a tree prior has*

Description

Get the number of distributions a tree prior has

Usage

```
get_tree_prior_n_distrs(tree_prior)
```

Arguments

`tree_prior` a tree priors, as returned by [create_tree_prior](#)

Value

the number of distributions a tree prior has

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# birth_rate_distr and death_rate_distr
expect_equal(get_tree_prior_n_distrs(create_bd_tree_prior()), 2)

# none
expect_equal(get_tree_prior_n_distrs(create_cbs_tree_prior()), 0)

# pop_size_distr
expect_equal(get_tree_prior_n_distrs(create_ccp_tree_prior()), 1)

# pop_size_distr and growth_rate_distr
expect_equal(get_tree_prior_n_distrs(create_cep_tree_prior()), 2)

# birth_rate_distr
expect_equal(get_tree_prior_n_distrs(create_yule_tree_prior()), 1)
```

`get_tree_prior_n_params`*Get the number of parameters a tree prior has*

Description

Get the number of parameters a tree prior has

Usage

```
get_tree_prior_n_params(tree_prior)
```

Arguments

`tree_prior` a `tree_prior`, as created by [create_tree_prior](#)

Value

the number of parameters a tree prior has

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# birth_rate_distr is uniform, which has zero parameters
# death_rate_distr is uniform, which has zero parameters
testit::assert(
  get_tree_prior_n_params(create_bd_tree_prior()) == 0
)

# no distributions, no parameters
testit::assert(
  get_tree_prior_n_params(create_cbs_tree_prior()) == 0
)

# pop_size_distr is 1/x, which has zero parameters
testit::assert(
  get_tree_prior_n_params(create_ccp_tree_prior()) == 0
)

# pop_size_distr is 1/x, which has zero parameters
# growth_rate_distr is Laplace, which has two parameters
testit::assert(
  get_tree_prior_n_params(create_cep_tree_prior()) == 2
)

# birth_rate_distr is uniform, which has zero parameters
```

```
testit::assert(
  get_tree_prior_n_params(create_yule_tree_prior()) == 0
)
```

get_xml_closing_tag *Get the XML closing tag*

Description

Get the XML closing tag

Usage

```
get_xml_closing_tag(text)
```

Arguments

text lines of XML to extract the XML closing tag from

Value

the closing tag if found, else NA

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_equal(
  get_xml_closing_tag("<my_tag text=something></my_tag>"),
  "my_tag"
)
expect_true(
  is_one_na(
    get_xml_closing_tag("<my_tag text=something/>")
  )
)
expect_true(is_one_na(get_xml_closing_tag("no_xml")))
```

get_xml_opening_tag *Get the XML opening tag*

Description

Get the XML opening tag

Usage

```
get_xml_opening_tag(text)
```

Arguments

text text to be determined to be valid

Value

the opening tag if found, else NA

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_equal(
  get_xml_opening_tag("<my_tag text=something/>"),
  "my_tag"
)
expect_true(is_one_na(get_xml_opening_tag("no_xml")))
```

gtr_site_model_to_xml_subst_model
Converts a GTR site model to XML, used in the substModel section

Description

Converts a GTR site model to XML, used in the substModel section

Usage

```
gtr_site_model_to_xml_subst_model(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

has_mrca_prior	<i>Determines if the inference model has an MRCA prior.</i>
----------------	---

Description

Will [stop](#) if the inference model is invalid

Usage

```
has_mrca_prior(inference_model)
```

Arguments

inference_model

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

Value

TRUE if the inference model has an MRCA prior, FALSE otherwise

Note

MRCA: 'Most Recent Common Ancestor'

Author(s)

Richèl J.C. Bilderbeek

See Also

- [create_inference_model](#): create an inference model
- [create_mrca_prior](#): create an MRCA prior

Examples

```
# No MRCA prior
inference_model <- create_inference_model(
  mrca_prior = NA
)
testthat::expect_false(has_mrca_prior(inference_model))

# A default MRCA prior
inference_model <- create_inference_model(
  mrca_prior = create_mrca_prior()
)
testthat::expect_true(has_mrca_prior(inference_model))
```

has_xml_closing_tag *Is an XML closing tag with the value of section present among the lines of the text?*

Description

Is an XML closing tag with the value of section present among the lines of the text?

Usage

```
has_xml_closing_tag(lines, section)
```

Arguments

lines	lines of the XML text
section	the XML section

Value

TRUE if there is an XML closing tag with the value of section present. FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

has_xml_opening_tag *Is an XML opening tag with value 'section' present among the lines of the text?*

Description

Is an XML opening tag with value 'section' present among the lines of the text?

Usage

```
has_xml_opening_tag(lines, section = NA)
```

Arguments

lines	lines of an XML text
section	if NA, this function returns TRUE if there is any XML opening tag. If section is set to a certain word, this function returns TRUE if that tag matches section

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

has_xml_short_closing_tag *Is an XML closing tag with short closing text in one of the lines of the text?*

Description

Is an XML closing tag with short closing text in one of the lines of the text?

Usage

```
has_xml_short_closing_tag(lines)
```

Arguments

lines	lines of an XML text
-------	----------------------

Value

TRUE if there is an XML tag that also closes present in the lines of text, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)
```

```
expect_true(has_xml_short_closing_tag("<my_tag id=1/>"))
```

```
expect_false(has_xml_short_closing_tag("<my_tag id=1>text</my_tag>"))
```

```
hky_site_model_to_xml_subst_model
```

Converts a site model to XML, used in the substModel section

Description

Converts a site model to XML, used in the substModel section

Usage

```
hky_site_model_to_xml_subst_model(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

```
indent
```

Indent text for a certain number of spaces. If the text is only whitespace, leave it as such

Description

Indent text for a certain number of spaces. If the text is only whitespace, leave it as such

Usage

```
indent(text, n_spaces = 4)
```

Arguments

- `text` the text to indent
- `n_spaces` the number of spaces to add before the text. BEAUti uses four spaces by default

Value

the indented text

Author(s)

Richèl J.C. Bilderbeek

`init_bd_tree_prior` *Initializes a Birth-Death tree prior*

Description

Initializes a Birth-Death tree prior

Usage

```
init_bd_tree_prior(bd_tree_prior, distr_id, param_id)
```

Arguments

- `bd_tree_prior` a Birth-Death tree prior, as created by [create_bd_tree_prior](#)
- `distr_id` a distributions' ID
- `param_id` a parameter's ID

Value

an initialized Birth-Death tree prior

Author(s)

Richèl J.C. Bilderbeek

init_beta_distr	<i>Initializes a beta distribution</i>
-----------------	--

Description

Initializes a beta distribution

Usage

```
init_beta_distr(beta_distr, distr_id = 0, param_id = 0)
```

Arguments

beta_distr	a beta distribution, using create_beta_distr
distr_id	the first distribution's ID
param_id	the first parameter's ID

Value

an initialized beta distribution

Author(s)

Richèl J.C. Bilderbeek

init_ccp_tree_prior	<i>Initializes a Coalescent Constant Population tree prior</i>
---------------------	--

Description

Initializes a Coalescent Constant Population tree prior

Usage

```
init_ccp_tree_prior(ccp_tree_prior, distr_id, param_id)
```

Arguments

ccp_tree_prior	a Coalescent Constant Population tree prior, as returned by create_ccp_tree_prior
distr_id	a distributions' ID
param_id	a parameter's ID

Value

an initialized Coalescent Constant Population tree prior

Author(s)

Richèl J.C. Bilderbeek

`init_cep_tree_prior` *Initializes a Coalescent Exponential Population tree prior*

Description

Initializes a Coalescent Exponential Population tree prior

Usage`init_cep_tree_prior(cep_tree_prior, distr_id, param_id)`**Arguments**

<code>cep_tree_prior</code>	a Coalescent Exponential Population tree prior, as returned by create_cep_tree_prior
<code>distr_id</code>	a distributions' ID
<code>param_id</code>	a parameter's ID

Value

an initialized Coalescent Exponential Population tree prior

Author(s)

Richèl J.C. Bilderbeek

`init_clock_models` *Initializes all clock models*

Description

Initializes all clock models

Usage`init_clock_models(fasta_filenames, clock_models, distr_id = 0, param_id = 0)`**Arguments**

<code>fasta_filenames</code>	One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename.
<code>clock_models</code>	a list of one or more clock models, as returned by create_clock_model
<code>distr_id</code>	the first distributions' ID
<code>param_id</code>	the first parameter's ID

Value

a list of initialized clock models

Author(s)

Richèl J.C. Bilderbeek

init_distr	<i>Initializes a distribution</i>
------------	-----------------------------------

Description

Initializes a distribution

Usage

```
init_distr(distr, distr_id = 0, param_id = 0)
```

Arguments

distr	a distribution, using create_distr
distr_id	the first distribution's ID
param_id	the first parameter's ID

Value

an initialized distribution

Author(s)

Richèl J.C. Bilderbeek

init_exp_distr	<i>Initializes an exponential distribution</i>
----------------	--

Description

Initializes an exponential distribution

Usage

```
init_exp_distr(exp_distr, distr_id = 0, param_id = 0)
```

Arguments

- exp_distr a exponential distribution, using [create_exp_distr](#)
- distr_id the first distribution's ID
- param_id the first parameter's ID

Value

an initialized exponential distribution

Author(s)

Richèl J.C. Bilderbeek

init_gamma_distr *Initializes a gamma distribution*

Description

Initializes a gamma distribution

Usage

```
init_gamma_distr(gamma_distr, distr_id = 0, param_id = 0)
```

Arguments

- gamma_distr a gamma distribution, using [create_gamma_distr](#)
- distr_id the first distribution's ID
- param_id the first parameter's ID

Value

an initialized gamma distribution

Author(s)

Richèl J.C. Bilderbeek

init_gamma_site_model *Initializes a gamma site model*

Description

Initializes a gamma site model

Usage

```
init_gamma_site_model(gamma_site_model, distr_id = 0, param_id = 0)
```

Arguments

gamma_site_model	a site model's gamma site model, as returned by create_gamma_site_model
distr_id	the first distributions' ID
param_id	the first parameter's ID

Value

an initialized gamma site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

gamma_site_model <- create_gamma_site_model(
  gamma_cat_count = 2,
  gamma_shape_prior_distr = create_one_div_x_distr(id = NA)
)
expect_false(is_init_gamma_site_model(gamma_site_model))
gamma_site_model <- init_gamma_site_model(gamma_site_model)
expect_true(is_init_gamma_site_model(gamma_site_model))
```

init_gtr_site_model *Initializes a GTR site model*

Description

Initializes a GTR site model

Usage

```
init_gtr_site_model(gtr_site_model, distr_id = 0, param_id = 0)
```

Arguments

gtr_site_model a GTR site model, as returned by [create_gtr_site_model](#)
distr_id a distributions' ID
param_id a parameter's ID

Value

an initialized GTR site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
gtr_site_model <- create_gtr_site_model()
testit::assert(!is_init_gtr_site_model(gtr_site_model))
gtr_site_model <- init_gtr_site_model(gtr_site_model)
testit::assert(is_init_gtr_site_model(gtr_site_model))
```

init_hky_site_model *Initializes an HKY site model*

Description

Initializes an HKY site model

Usage

```
init_hky_site_model(hky_site_model, distr_id = 0, param_id = 0)
```

Arguments

hky_site_model an HKY site model, as returned by [create_hky_site_model](#)
distr_id a distributions' ID
param_id a parameter's ID

Value

an initialized HKY site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

hky_site_model <- create_hky_site_model()
expect_false(is_init_hky_site_model(hky_site_model))
hky_site_model <- init_hky_site_model(hky_site_model)
expect_true(is_init_hky_site_model(hky_site_model))
```

init_inference_model *Initialize an inference model*

Description

Initialize an inference model

Usage

```
init_inference_model(input_filename, inference_model)
```

Arguments

input_filename A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

inference_model

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

init_inv_gamma_distr *Initializes an inverse gamma distribution*

Description

Initializes an inverse gamma distribution

Usage

```
init_inv_gamma_distr(inv_gamma_distr, distr_id = 0, param_id = 0)
```

Arguments

inv_gamma_distr	an inverse gamma distribution, using create_inv_gamma_distr
distr_id	the first distribution's ID
param_id	the first parameter's ID

Value

an initialized inverse gamma distribution

Author(s)

Richèl J.C. Bilderbeek

init_jc69_site_model *Initializes a JC69 site model*

Description

Initializes a JC69 site model

Usage

```
init_jc69_site_model(jc69_site_model, distr_id = 0, param_id = 0)
```

Arguments

jc69_site_model	a JC69 site model, as returned by create_jc69_site_model
distr_id	a distributions' ID
param_id	a parameter's ID

Value

an initialized HKY site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

hky_site_model <- create_hky_site_model()
expect_false(is_init_hky_site_model(hky_site_model))
hky_site_model <- init_hky_site_model(hky_site_model)
expect_true(is_init_hky_site_model(hky_site_model))
```

init_laplace_distr *Initializes an Laplace distribution*

Description

Initializes an Laplace distribution

Usage

```
init_laplace_distr(laplace_distr, distr_id = 0, param_id = 0)
```

Arguments

laplace_distr a Laplace distribution, using [create_laplace_distr](#)
distr_id the first distribution's ID
param_id the first parameter's ID

Value

an initialized Laplace distribution

Author(s)

Richèl J.C. Bilderbeek

init_log_normal_distr *Initializes an log-normal distribution*

Description

Initializes an log-normal distribution

Usage

```
init_log_normal_distr(log_normal_distr, distr_id = 0, param_id = 0)
```

Arguments

log_normal_distr	a log-normal distribution, using create_log_normal_distr
distr_id	the first distribution's ID
param_id	the first parameter's ID

Value

an initialized log-normal distribution

Author(s)

Richèl J.C. Bilderbeek

init_mrca_prior *Initialize the MRCA prior.*

Description

Initialized by

- if no alignment ID is set, it is set by reading it from the alignment file
- if no taxa names are set, these are set by reading these from the alignment file

Usage

```
init_mrca_prior(input_filename, inference_model)
```

Arguments

- `input_filename` A FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.
- `inference_model` a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

<code>init_mrca_priors</code>	<i>Initializes all MRCA priors</i>
-------------------------------	------------------------------------

Description

Initializes all MRCA priors

Usage

```
init_mrca_priors(mrca_priors, distr_id = 0, param_id = 0)
```

Arguments

- `mrca_priors` a list of one or more Most Recent Common Ancestor priors, as returned by [create_mrca_prior](#)
- `distr_id` the first distributions' ID
- `param_id` the first parameter's ID

Value

a list of initialized MRCA priors

Author(s)

Richèl J.C. Bilderbeek

init_normal_distr *Initializes an normal distribution*

Description

Initializes an normal distribution

Usage

```
init_normal_distr(normal_distr, distr_id = 0, param_id = 0)
```

Arguments

normal_distr a normal distribution, using [create_normal_distr](#)
distr_id the first distribution's ID
param_id the first parameter's ID

Value

an initialized normal distribution

Author(s)

Richèl J.C. Bilderbeek

init_one_div_x_distr *Initializes an one-divided-by-x distribution*

Description

Initializes an one-divided-by-x distribution

Usage

```
init_one_div_x_distr(one_div_x_distr, distr_id = 0)
```

Arguments

one_div_x_distr a one-divided-by-x distribution, using [create_one_div_x_distr](#)
distr_id the first distribution's ID

Value

an initialized one-divided-by-x distribution

Author(s)

Richèl J.C. Bilderbeek

init_param	<i>Initializes a parameter</i>
------------	--------------------------------

Description

Initializes a parameter

Usage

```
init_param(param, id)
```

Arguments

param	a parameter, using create_param
id	the parameter's ID. Will be ignored if the parameter already has an ID

Value

an initialized parameter

Author(s)

Richèl J.C. Bilderbeek

init_poisson_distr	<i>Initializes an Poisson distribution</i>
--------------------	--

Description

Initializes an Poisson distribution

Usage

```
init_poisson_distr(poisson_distr, distr_id = 0, param_id = 0)
```

Arguments

poisson_distr	a Poisson distribution, using create_poisson_distr
distr_id	the first distribution's ID
param_id	the first parameter's ID

Value

an initialized Poisson distribution

Author(s)

Richèl J.C. Bilderbeek

init_rln_clock_model *Initializes a Relaxed Log-Normal clock model*

Description

Initializes a Relaxed Log-Normal clock model

Usage

```
init_rln_clock_model(rln_clock_model, distr_id = 0, param_id = 0)
```

Arguments

rln_clock_model	a Relaxed Log-Normal clock model, as returned by create_rln_clock_model
distr_id	a distributions' ID
param_id	a parameter's ID

Value

an initialized Relaxed Log-Normal clock model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

rln_clock_model <- create_rln_clock_model()
expect_false(is_init_rln_clock_model(rln_clock_model))
rln_clock_model <- init_rln_clock_model(rln_clock_model)
# Dimension is set to NA by default, for unknown reasons.
# Because 'init_rln_clock_model' does not initialize it (for
# unknown reasons), set it manually
rln_clock_model$dimension <- 42
expect_true(is_init_rln_clock_model(rln_clock_model))
```

init_site_models *Initializes all site models*

Description

Initializes all site models

Usage

```
init_site_models(site_models, ids, distr_id = 0, param_id = 0)
```

Arguments

site_models	one or more site models, as returned by create_site_model
ids	one or more alignments' IDs. IDs can be extracted from their FASTA filenames with get_alignment_ids_from_fasta_filenames)
distr_id	the first distributions' ID
param_id	the first parameter's ID

Value

a list of initialized site models

Author(s)

Richèl J.C. Bilderbeek

init_strict_clock_model
Initializes a strict clock model

Description

Initializes a strict clock model

Usage

```
init_strict_clock_model(strict_clock_model, distr_id = 0, param_id = 0)
```

Arguments

strict_clock_model	a strict clock model, as returned by create_strict_clock_model
distr_id	a distributions' ID
param_id	a parameter's ID

Value

an initialized strict clock model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

strict_clock_model <- create_strict_clock_model()
expect_false(is_init_strict_clock_model(strict_clock_model))
strict_clock_model <- init_strict_clock_model(strict_clock_model)
expect_true(is_init_strict_clock_model(strict_clock_model))
```

init_tn93_site_model *Initializes a TN93 site model*

Description

Initializes a TN93 site model

Usage

```
init_tn93_site_model(tn93_site_model, distr_id = 0, param_id = 0)
```

Arguments

tn93_site_model	a TN93 site model, as returned by create_tn93_site_model
distr_id	a distributions' ID
param_id	a parameter's ID

Value

an initialized TN93 site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

tn93_site_model <- create_tn93_site_model()
expect_false(is_init_tn93_site_model(tn93_site_model))
tn93_site_model <- init_tn93_site_model(tn93_site_model)
expect_true(is_init_tn93_site_model(tn93_site_model))
```

init_tree_priors *Initializes all tree priors*

Description

Initializes all tree priors

Usage

```
init_tree_priors(tree_priors, ids, distr_id = 0, param_id = 0)
```

Arguments

tree_priors	one or more tree priors, as returned by create_tree_prior
ids	one or more alignments' IDs. IDs can be extracted from their FASTA filenames with get_alignment_ids_from_fasta_filenames)
distr_id	the first distributions' ID
param_id	the first parameter's ID

Value

a list of initialized tree priors

Author(s)

Richèl J.C. Bilderbeek

init_uniform_distr *Initializes a uniform distribution*

Description

Initializes a uniform distribution

Usage

```
init_uniform_distr(uniform_distr, distr_id = 0)
```

Arguments

uniform_distr	a uniform distribution, using create_uniform_distr
distr_id	the first distribution's ID

Value

an initialized uniform distribution

Author(s)

Richèl J.C. Bilderbeek

init_yule_tree_prior *Initializes a Yule tree prior*

Description

Initializes a Yule tree prior

Usage

```
init_yule_tree_prior(yule_tree_prior, distr_id, param_id)
```

Arguments

yule_tree_prior	a Yule tree_prior, as created by create_yule_tree_prior
distr_id	a distributions' ID
param_id	a parameter's ID

Value

an initialized Yule tree prior

Author(s)

Richèl J.C. Bilderbeek

interspace *Puts spaces in between the lines*

Description

Puts spaces in between the lines

Usage

```
interspace(lines)
```

Arguments

lines	lines of text
-------	---------------

Value

interspaced lines of text

Author(s)

Richèl J.C. Bilderbeek

is_alpha_param

Determine if the object is a valid alpha parameter

Description

Determine if the object is a valid alpha parameter

Usage

```
is_alpha_param(x)
```

Arguments

x an object, to be determined if it is a valid alpha parameter

Value

TRUE if x is a valid alpha parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_alpha_param(create_alpha_param()))
expect_false(is_alpha_param(create_beta_param()))
expect_false(is_alpha_param(create_clock_rate_param()))
expect_false(is_alpha_param(create_kappa_1_param()))
expect_false(is_alpha_param(create_kappa_2_param()))
expect_false(is_alpha_param(create_lambda_param()))
expect_false(is_alpha_param(create_m_param()))
expect_false(is_alpha_param(create_mean_param()))
expect_false(is_alpha_param(create_mu_param()))
expect_false(is_alpha_param(create_rate_ac_param()))
expect_false(is_alpha_param(create_rate_ag_param()))
expect_false(is_alpha_param(create_rate_at_param()))
expect_false(is_alpha_param(create_rate_cg_param()))
expect_false(is_alpha_param(create_rate_ct_param()))
expect_false(is_alpha_param(create_rate_gt_param()))
```

```

expect_false(is_alpha_param(create_s_param()))
expect_false(is_alpha_param(create_scale_param()))
expect_false(is_alpha_param(create_sigma_param()))

expect_false(is_alpha_param(NA))
expect_false(is_alpha_param(NULL))
expect_false(is_alpha_param("nonsense"))
expect_false(is_alpha_param(create_jc69_site_model()))
expect_false(is_alpha_param(create_strict_clock_model()))
expect_false(is_alpha_param(create_yule_tree_prior()))
expect_false(is_alpha_param(create_mcmc()))

```

is_bd_tree_prior	<i>Determine if the object is a valid Birth Death tree prior</i>
------------------	--

Description

Determine if the object is a valid Birth Death tree prior

Usage

```
is_bd_tree_prior(x)
```

Arguments

x an object, to be determined if it is a valid birth death tree prior

Value

TRUE if x is a valid birth death tree prior, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_bd_tree_prior](#) to create a valid Birth-Death tree prior

Examples

```

testit::assert(is_bd_tree_prior(create_bd_tree_prior()))
testit::assert(!is_bd_tree_prior(create_cbs_tree_prior()))
testit::assert(!is_bd_tree_prior(create_ccp_tree_prior()))
testit::assert(!is_bd_tree_prior(create_cep_tree_prior()))
testit::assert(!is_bd_tree_prior(create_yule_tree_prior()))

```

is_beauti_options	<i>Determine if the object is a valid beauti_options</i>
-------------------	--

Description

Determine if the object is a valid beauti_options

Usage

```
is_beauti_options(x)
```

Arguments

x an object, to be determined if it is a beauti_options

Value

TRUE if the object is a valid beauti_options, **FALSE** otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [create_beauti_options](#) to create a valid beauti_options object

Examples

```
library(testthat)

expect_true(is_beauti_options(create_beauti_options()))

expect_false(is_beauti_options("nonsense"))
expect_false(is_beauti_options(NA))
expect_false(is_beauti_options(NULL))
expect_false(is_beauti_options(""))
expect_false(is_beauti_options(c()))
```

is_beta_distr	<i>Determine if the object is a valid beta distribution, as created by create_beta_distr</i>
---------------	--

Description

Determine if the object is a valid beta distribution, as created by [create_beta_distr](#)

Usage

```
is_beta_distr(x)
```

Arguments

x an object, to be determined if it is a valid beta distribution,

Value

TRUE if x is a valid beta distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_beta_distr(create_beta_distr()))
expect_false(is_beta_distr(create_exp_distr()))

expect_false(is_beta_distr(NA))
expect_false(is_beta_distr(NULL))
expect_false(is_beta_distr("nonsense"))
```

is_beta_param	<i>Determine if the object is a valid beta parameter</i>
---------------	--

Description

Determine if the object is a valid beta parameter

Usage

```
is_beta_param(x)
```

Arguments

x an object, to be determined if it is a valid beta parameter

Value

TRUE if x is a valid beta parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_false(is_beta_param(create_alpha_param()))
expect_true(is_beta_param(create_beta_param()))
expect_false(is_beta_param(create_clock_rate_param()))
expect_false(is_beta_param(create_kappa_1_param()))
expect_false(is_beta_param(create_kappa_2_param()))
expect_false(is_beta_param(create_lambda_param()))
expect_false(is_beta_param(create_m_param()))
expect_false(is_beta_param(create_mean_param()))
expect_false(is_beta_param(create_mu_param()))
expect_false(is_beta_param(create_rate_ac_param()))
expect_false(is_beta_param(create_rate_ag_param()))
expect_false(is_beta_param(create_rate_at_param()))
expect_false(is_beta_param(create_rate_cg_param()))
expect_false(is_beta_param(create_rate_ct_param()))
expect_false(is_beta_param(create_rate_gt_param()))
expect_false(is_beta_param(create_s_param()))
expect_false(is_beta_param(create_scale_param()))
expect_false(is_beta_param(create_sigma_param()))

expect_false(is_beta_param(NA))
expect_false(is_beta_param(NULL))
expect_false(is_beta_param("nonsense"))
expect_false(is_beta_param(create_jc69_site_model()))
```

```
expect_false(is_beta_param(create_strict_clock_model()))
expect_false(is_beta_param(create_yule_tree_prior()))
expect_false(is_beta_param(create_mcmc()))
```

is_cbs_tree_prior	<i>Determine if the object is a valid constant coalescent Bayesian skyline prior</i>
-------------------	--

Description

Determine if the object is a valid constant coalescent Bayesian skyline prior

Usage

```
is_cbs_tree_prior(x)
```

Arguments

x	an object, to be determined if it is a valid constant coalescent Bayesian skyline prior
---	---

Value

TRUE if x is a valid constant coalescent Bayesian skyline prior, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_cbs_tree_prior](#) to create a valid coalescent Bayes skyline tree prior

Examples

```
testit::assert(!is_cbs_tree_prior(create_bd_tree_prior()))
testit::assert( is_cbs_tree_prior(create_cbs_tree_prior()))
testit::assert(!is_cbs_tree_prior(create_ccp_tree_prior()))
testit::assert(!is_cbs_tree_prior(create_cep_tree_prior()))
testit::assert(!is_cbs_tree_prior(create_yule_tree_prior()))
```

is_ccp_tree_prior	<i>Determine if the object is a valid constant coalescence population tree prior</i>
-------------------	--

Description

Determine if the object is a valid constant coalescence population tree prior

Usage

```
is_ccp_tree_prior(x)
```

Arguments

x an object, to be determined if it is a valid constant coalescence population tree prior

Value

TRUE if x is a valid constant coalescence population tree prior, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_ccp_tree_prior](#) to create a valid constant coalescence population tree prior

Examples

```
testit::assert(!is_ccp_tree_prior(create_bd_tree_prior()))
testit::assert(!is_ccp_tree_prior(create_cbs_tree_prior()))
testit::assert( is_ccp_tree_prior(create_ccp_tree_prior()))
testit::assert(!is_ccp_tree_prior(create_cep_tree_prior()))
testit::assert(!is_ccp_tree_prior(create_yule_tree_prior()))
```

is_cep_tree_prior	<i>Determine if the object is a valid coalescent exponential population tree prior</i>
-------------------	--

Description

Determine if the object is a valid coalescent exponential population tree prior

Usage

```
is_cep_tree_prior(x)
```

Arguments

x	an object, to be determined if it is a valid constant coalescent exponential population tree prior
---	--

Value

TRUE if x is a valid coalescent exponential population tree prior, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_cep_tree_prior](#) to create a valid coalescent exponential population tree prior

Examples

```
testit::assert(!is_cep_tree_prior(create_bd_tree_prior()))
testit::assert(!is_cep_tree_prior(create_cbs_tree_prior()))
testit::assert(!is_cep_tree_prior(create_ccp_tree_prior()))
testit::assert( is_cep_tree_prior(create_cep_tree_prior()))
testit::assert(!is_cep_tree_prior(create_yule_tree_prior()))
```

is_clock_model	<i>Determine if the object is a valid clock_model</i>
----------------	---

Description

Determine if the object is a valid clock_model

Usage

```
is_clock_model(x)
```

Arguments

x an object, to be determined if it is a clock_model

Value

TRUE if the clock_model is a valid clock_model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

see [create_clock_model](#) for an overview of functions to create valid clock model

Examples

```
library(testthat)

expect_true(is_clock_model(create_strict_clock_model()))
expect_true(is_clock_model(create_rln_clock_model()))

expect_false(is_clock_model(NA))
expect_false(is_clock_model(NULL))
expect_false(is_clock_model("nonsense"))
expect_false(is_clock_model(create_jc69_site_model()))
expect_false(is_clock_model(create_mcmc()))
```

is_clock_model_name *Determines if the name is a valid clock model name*

Description

Determines if the name is a valid clock model name

Usage

```
is_clock_model_name(name)
```

Arguments

name the name to be tested

Value

TRUE if the name is a valid clock_model name, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_clock_model_name("relaxed_log_normal"))
expect_true(is_clock_model_name("strict"))
```

is_clock_rate_param *Determine if the object is a valid clock_rate parameter*

Description

Determine if the object is a valid clock_rate parameter

Usage

```
is_clock_rate_param(x)
```

Arguments

x an object, to be determined if it is a valid clock_rate parameter

Value

TRUE if x is a valid clock_rate parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_false(is_clock_rate_param(create_alpha_param()))
expect_false(is_clock_rate_param(create_beta_param()))
expect_true(is_clock_rate_param(create_clock_rate_param()))
expect_false(is_clock_rate_param(create_kappa_1_param()))
expect_false(is_clock_rate_param(create_kappa_2_param()))
expect_false(is_clock_rate_param(create_lambda_param()))
expect_false(is_clock_rate_param(create_m_param()))
expect_false(is_clock_rate_param(create_mean_param()))
expect_false(is_clock_rate_param(create_mu_param()))
expect_false(is_clock_rate_param(create_rate_ac_param()))
expect_false(is_clock_rate_param(create_rate_ag_param()))
expect_false(is_clock_rate_param(create_rate_at_param()))
expect_false(is_clock_rate_param(create_rate_cg_param()))
expect_false(is_clock_rate_param(create_rate_ct_param()))
expect_false(is_clock_rate_param(create_rate_gt_param()))
expect_false(is_clock_rate_param(create_s_param()))
expect_false(is_clock_rate_param(create_scale_param()))
expect_false(is_clock_rate_param(create_sigma_param()))

expect_false(is_clock_rate_param(NA))
expect_false(is_clock_rate_param(NULL))
expect_false(is_clock_rate_param("nonsense"))
expect_false(is_clock_rate_param(create_jc69_site_model()))
expect_false(is_clock_rate_param(create_strict_clock_model()))
expect_false(is_clock_rate_param(create_yule_tree_prior()))
expect_false(is_clock_rate_param(create_mcmc()))
```

is_default_mcmc

Determine if the MCMC is a default MCMC

Description

Determine if the MCMC is a default MCMC

Usage

```
is_default_mcmc(mcmc)
```


Arguments

mcmc one MCMC. Use [create_mcmc](#) to create an MCMC. Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Value

TRUE if the MCMC is a default MCMC

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# An MCMC created by 'create_mcmc' is default
expect_true(is_default_mcmc(create_mcmc()))

# An MCMC created by 'create_ns_mcmc' is not
expect_false(is_default_mcmc(create_ns_mcmc()))
```

is_distr *Determine if the object is a valid distribution*

Description

Determine if the object is a valid distribution

Usage

```
is_distr(x)
```

Arguments

x an object, to be determined if it is a valid distribution

Value

TRUE if x is a valid distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_beta_distr](#), [is_exp_distr](#), [is_gamma_distr](#), [is_inv_gamma_distr](#), [is_laplace_distr](#), [is_log_normal_distr](#), [is_normal_distr](#), [is_one_div_x_distr](#), [is_poisson_distr](#), or [is_uniform_distr](#), to check for more specific distribution

Examples

```
library(testthat)

expect_true(is_distr(create_beta_distr()))
expect_true(is_distr(create_exp_distr()))
expect_true(is_distr(create_gamma_distr()))
expect_true(is_distr(create_inv_gamma_distr()))
expect_true(is_distr(create_laplace_distr()))
expect_true(is_distr(create_log_normal_distr()))
expect_true(is_distr(create_normal_distr()))
expect_true(is_distr(create_one_div_x_distr()))
expect_true(is_distr(create_poisson_distr()))
expect_true(is_distr(create_uniform_distr()))

expect_false(is_distr(NA))
expect_false(is_distr(NULL))
expect_false(is_distr("nonsense"))
```

is_distr_name	<i>Determines if the name is a valid distribution name</i>
---------------	--

Description

Determines if the name is a valid distribution name

Usage

```
is_distr_name(name)
```

Arguments

name	the name to be tested
------	-----------------------

Value

TRUE if the name is a valid distribution name, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_distr_name("uniform"))
expect_true(is_distr_name("normal"))
expect_true(is_distr_name("one_div_x"))
expect_true(is_distr_name("log_normal"))
expect_true(is_distr_name("exponential"))
expect_true(is_distr_name("gamma"))
expect_true(is_distr_name("beta"))
expect_true(is_distr_name("laplace"))
expect_true(is_distr_name("inv_gamma"))
expect_true(is_distr_name("poisson"))
```

is_exp_distr	<i>Determine if the object is a valid exponential distribution as created by create_exp_distr</i>
--------------	---

Description

Determine if the object is a valid exponential distribution as created by [create_exp_distr](#)

Usage

```
is_exp_distr(x)
```

Arguments

x an object, to be determined if it is a valid exponential distribution

Value

TRUE if x is a valid exponential distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_exp_distr(create_exp_distr()))
expect_false(is_exp_distr(create_gamma_distr()))
expect_false(is_exp_distr(NA))
expect_false(is_exp_distr(NULL))
expect_false(is_exp_distr("nonsense"))
```

is_freq_equilibrium_name

Checks if name is a valid freq_equilibrium argument value

Description

Checks if name is a valid freq_equilibrium argument value

Usage

```
is_freq_equilibrium_name(name)
```

Arguments

name the name to check if it is a valid freq_equilibrium argument value

Value

TRUE if the name is a valid freq_equilibrium value

Author(s)

Richèl J.C. Bilderbeek

See Also

the freq_equilibrium argument is used by [create_gtr_site_model](#), [create_hky_site_model](#), and [create_tn93_site_model](#)

Examples

```
library(testthat)

expect_true(is_freq_equilibrium_name("estimated"))
expect_true(is_freq_equilibrium_name("empirical"))
expect_true(is_freq_equilibrium_name("all_equal"))
```

is_gamma_distr	<i>Determine if the object is a valid gamma distribution, as created by create_gamma_distr</i>
----------------	--

Description

Determine if the object is a valid gamma distribution, as created by [create_gamma_distr](#)

Usage

```
is_gamma_distr(x)
```

Arguments

x an object, to be determined if it is a valid gamma distribution

Value

TRUE if x is a valid gamma distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_gamma_distr(create_gamma_distr()))

expect_false(is_gamma_distr(create_inv_gamma_distr()))
expect_false(is_gamma_distr(NA))
expect_false(is_gamma_distr(NULL))
expect_false(is_gamma_distr("nonsense"))
```

is_gamma_site_model *Is object x a gamma site model?*

Description

Is object x a gamma site model?

Usage

```
is_gamma_site_model(x)
```

Arguments

x the object to be determined if it is a valid gamma site object

Value

TRUE if x is a valid gamma site object, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_gamma_site_model(create_gamma_site_model()))

expect_false(is_gamma_site_model("nonsense"))
expect_false(is_gamma_site_model(NA))
expect_false(is_gamma_site_model(NULL))
expect_false(is_gamma_site_model(""))
expect_false(is_gamma_site_model(c()))
```

is_gtr_site_model *Determine if the object is a valid GTR site model, as created by [create_gtr_site_model](#)*

Description

Determine if the object is a valid GTR site model, as created by [create_gtr_site_model](#)

Usage

```
is_gtr_site_model(x)
```

Arguments

x an object, to be determined if it is a valid GTR site model

Value

TRUE if x is a valid GTR site model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# site models
expect_true(is_gtr_site_model(create_gtr_site_model()))
expect_false(is_gtr_site_model(create_hky_site_model()))
expect_false(is_gtr_site_model(create_jc69_site_model()))
expect_false(is_gtr_site_model(create_tn93_site_model()))

# other models
expect_false(is_gtr_site_model(NA))
expect_false(is_gtr_site_model(NULL))
expect_false(is_gtr_site_model("nonsense"))
expect_false(is_gtr_site_model(create_strict_clock_model()))
expect_false(is_gtr_site_model(create_bd_tree_prior()))
expect_false(is_gtr_site_model(create_mcmc()))
```

is_hky_site_model *Determine if the object is a valid HKY site model, as created by*
[create_hky_site_model](#)

Description

Determine if the object is a valid HKY site model, as created by [create_hky_site_model](#)

Usage

```
is_hky_site_model(x)
```

Arguments

x an object, to be determined if it is a valid HKY site model

Value

TRUE if x is a valid HKY site model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# site models
expect_true(is_hky_site_model(create_hky_site_model()))
expect_false(is_hky_site_model(create_gtr_site_model()))
expect_false(is_hky_site_model(create_jc69_site_model()))
expect_false(is_hky_site_model(create_tn93_site_model()))

# other models
expect_false(is_hky_site_model(NA))
expect_false(is_hky_site_model(NULL))
expect_false(is_hky_site_model("nonsense"))
expect_false(is_hky_site_model(create_strict_clock_model()))
expect_false(is_hky_site_model(create_bd_tree_prior()))
expect_false(is_hky_site_model(create_mcmc()))
```

is_id

Determine if the object is a valid ID

Description

Determine if the object is a valid ID

Usage

```
is_id(x)
```

Arguments

x an object, to be determined if it is a valid ID

Value

TRUE if x is a valid ID, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

to check multiple IDs, use [are_ids](#)

Examples

```
library(testthat)

expect_true(is_id("anthus_aco"))
expect_true(is_id(3))
expect_false(is_id(ape::rcoal(3)))
expect_false(is_id(NULL))
expect_false(is_id(NA))
```

is_inference_model *Determine if the input is an inference model*

Description

Determine if the input is an inference model

Usage

```
is_inference_model(x)
```

Arguments

x object to be determined of if it is an inference model

Value

TRUE if the object is an inference model

is_init_bd_tree_prior *Determine if x is an initialized Birth-Death tree_prior object*

Description

Determine if x is an initialized Birth-Death tree_prior object

Usage

```
is_init_bd_tree_prior(x)
```

Arguments

x the object to check if it is an initialized Birth-Death tree prior object

Value

TRUE if x is an initialized Birth-Death tree_prior object

Author(s)

Richèl J.C. Bilderbeek

is_init_beta_distr *Determine if x is an initialized beta distribution object as created by [create_beta_distr](#)*

Description

Determine if x is an initialized beta distribution object as created by [create_beta_distr](#)

Usage

```
is_init_beta_distr(x)
```

Arguments

x the object to check if it is an initialized beta distribution object

Value

TRUE if x is an initialized beta distribution object

Author(s)

Richèl J.C. Bilderbeek

is_init_cbs_tree_prior *Determine if x is an initialized Coalescent Bayesian Skyline tree_prior object*

Description

Determine if x is an initialized Coalescent Bayesian Skyline tree_prior object

Usage

```
is_init_cbs_tree_prior(x)
```

Arguments

x the object to check if it is an initialized Coalescent Bayesian Skyline tree prior object

Value

TRUE if *x* is an initialized Coalescent Bayesian Skyline tree prior object

Author(s)

Richèl J.C. Bilderbeek

is_init_ccp_tree_prior

Determine if x is an initialized Coalescent Constant Population tree_prior object

Description

Determine if *x* is an initialized Coalescent Constant Population tree_prior object

Usage

`is_init_ccp_tree_prior(x)`

Arguments

x the object to check if it is an initialized Coalescent Constant Population tree prior object

Value

TRUE if *x* is an initialized Coalescent Constant Population tree prior object

Author(s)

Richèl J.C. Bilderbeek

is_init_cep_tree_prior

Determine if x is an initialized Coalescent Exponential Population tree_prior object

Description

Determine if *x* is an initialized Coalescent Exponential Population tree_prior object

Usage

`is_init_cep_tree_prior(x)`

Arguments

x the object to check if it is an initialized Coalescent Exponential Population tree prior object

Value

TRUE if x is an initialized Coalescent Exponential Population tree prior object

Author(s)

Richèl J.C. Bilderbeek

`is_init_clock_model` *Determine if x is an initialized clock_model object, as created by [create_clock_model](#)*

Description

Determine if x is an initialized clock_model object, as created by [create_clock_model](#)

Usage

```
is_init_clock_model(x)
```

Arguments

x the object to check if it is an initialized clock_models object

Value

TRUE if x is an initialized clock_model object

Author(s)

Richèl J.C. Bilderbeek

is_init_distr	<i>Determine if x is an initialized distribution object as created by create_distr</i>
---------------	--

Description

Determine if x is an initialized distribution object as created by [create_distr](#)

Usage

```
is_init_distr(x)
```

Arguments

x the object to check if it is an initialized distribution object

Value

TRUE if x is an initialized distribution object

Author(s)

Richèl J.C. Bilderbeek

is_init_exp_distr	<i>Determine if x is an initialized exponential distribution object as created by create_exp_distr</i>
-------------------	--

Description

Determine if x is an initialized exponential distribution object as created by [create_exp_distr](#)

Usage

```
is_init_exp_distr(x)
```

Arguments

x the object to check if it is an initialized exponential distribution object

Value

TRUE if x is an initialized exponential distribution object

Author(s)

Richèl J.C. Bilderbeek

is_init_gamma_distr *Determine if x is an initialized gamma distribution object*

Description

Determine if x is an initialized gamma distribution object

Usage

```
is_init_gamma_distr(x)
```

Arguments

x the object to check if it is an initialized gamma distribution object

Value

TRUE if x is an initialized gamma distribution object

Author(s)

Richèl J.C. Bilderbeek

is_init_gamma_site_model
Determine if x is an initialized gamma site model, as created by
[create_gamma_site_model](#)

Description

Determine if x is an initialized gamma site model, as created by [create_gamma_site_model](#)

Usage

```
is_init_gamma_site_model(x)
```

Arguments

x the object to check if it is an initialized gamma site_models object

Value

TRUE if x is an initialized gamma site model

Author(s)

Richèl J.C. Bilderbeek

```
is_init_gtr_site_model
```

Determine if x is an initialized GTR site model as created by [create_gtr_site_model](#)

Description

Determine if x is an initialized GTR site model as created by [create_gtr_site_model](#)

Usage

```
is_init_gtr_site_model(x)
```

Arguments

x the object to check if it is an initialized GTR site model

Value

TRUE if x is an initialized GTR site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

gtr_site_model <- create_gtr_site_model()
expect_false(is_init_gtr_site_model(gtr_site_model))
gtr_site_model <- init_gtr_site_model(gtr_site_model)
expect_true(is_init_gtr_site_model(gtr_site_model))
```

```
is_init_hky_site_model
```

Determine if x is an initialized HKY site model as created by [create_hky_site_model](#)

Description

Determine if x is an initialized HKY site model as created by [create_hky_site_model](#)

Usage

```
is_init_hky_site_model(x)
```

Arguments

x the object to check if it is an initialized HKY site model

Value

TRUE if x is an initialized HKY site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

hky_site_model <- create_hky_site_model()
expect_false(is_init_hky_site_model(hky_site_model))
hky_site_model <- init_hky_site_model(hky_site_model)
expect_true(is_init_hky_site_model(hky_site_model))
```

is_init_inv_gamma_distr

*Determine if x is an initialized inverse-gamma distribution as created
by [create_inv_gamma_distr](#)*

Description

Determine if x is an initialized inverse-gamma distribution as created by [create_inv_gamma_distr](#)

Usage

```
is_init_inv_gamma_distr(x)
```

Arguments

x the object to check if it is an initialized inverse-gamma distribution

Value

TRUE if x is an initialized inverse-gamma distribution

Author(s)

Richèl J.C. Bilderbeek

```
is_init_jc69_site_model
```

Determine if x is an initialized JC69 site model as created by [create_jc69_site_model](#)

Description

Determine if x is an initialized JC69 site model as created by [create_jc69_site_model](#)

Usage

```
is_init_jc69_site_model(x)
```

Arguments

x the object to check if it is an initialized JC69 site model

Value

TRUE if x is an initialized JC69 site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

jc69_site_model <- create_jc69_site_model(
  gamma_site_model = create_gamma_site_model(
    gamma_cat_count = 2,
    gamma_shape_prior_distr = create_normal_distr()
  )
)
expect_false(is_init_jc69_site_model(jc69_site_model))
jc69_site_model <- init_jc69_site_model(jc69_site_model)
expect_true(is_init_jc69_site_model(jc69_site_model))
```

`is_init_laplace_distr` *Determine if x is an initialized Laplace distribution as created by [create_laplace_distr](#)*

Description

Determine if x is an initialized Laplace distribution as created by [create_laplace_distr](#)

Usage

```
is_init_laplace_distr(x)
```

Arguments

x the object to check if it is an initialized Laplace distribution

Value

TRUE if x is an initialized Laplace distribution

Author(s)

Richèl J.C. Bilderbeek

`is_init_log_normal_distr` *Determine if x is an initialized log_normal distribution object as created by [create_log_normal_distr](#)*

Description

Determine if x is an initialized log_normal distribution object as created by [create_log_normal_distr](#)

Usage

```
is_init_log_normal_distr(x)
```

Arguments

x the object to check if it is an initialized log_normal distribution object

Value

TRUE if x is an initialized log_normal distribution object

Author(s)

Richèl J.C. Bilderbeek

is_init_mrca_prior *Determine if x is an initialized MRCA prior*

Description

Determine if x is an initialized MRCA prior

Usage

```
is_init_mrca_prior(x)
```

Arguments

x the object to check if it is an initialized MRCA prior

Value

TRUE if x is an initialized MRCA prior

Author(s)

Richèl J.C. Bilderbeek

is_init_normal_distr *Determine if x is an initialized normal distribution object as created by [create_normal_distr](#)*

Description

Determine if x is an initialized normal distribution object as created by [create_normal_distr](#)

Usage

```
is_init_normal_distr(x)
```

Arguments

x the object to check if it is an initialized normal distribution object

Value

TRUE if x is an initialized normal distribution object

Author(s)

Richèl J.C. Bilderbeek

is_init_one_div_x_distr

Determine if x is an initialized one_div_x distribution object as created by [create_one_div_x_distr](#)

Description

Determine if x is an initialized one_div_x distribution object as created by [create_one_div_x_distr](#)

Usage

```
is_init_one_div_x_distr(x)
```

Arguments

x the object to check if it is an initialized one_div_x distribution object

Value

TRUE if x is an initialized one_div_x distribution object

Author(s)

Richèl J.C. Bilderbeek

is_init_param

Determine if x is an initialized parameter, as created by [create_param](#)

Description

Determine if x is an initialized parameter, as created by [create_param](#)

Usage

```
is_init_param(x)
```

Arguments

x the object to check if it is an initialized parameter

Value

[TRUE](#) if x is an initialized parameter, [FALSE](#) otherwise

Author(s)

Richèl J.C. Bilderbeek

`is_init_poisson_distr` *Determine if x is an initialized Poisson distribution object as created by [create_poisson_distr](#)*

Description

Determine if x is an initialized Poisson distribution object as created by [create_poisson_distr](#)

Usage

```
is_init_poisson_distr(x)
```

Arguments

x the object to check if it is an initialized Poisson distribution object

Value

TRUE if x is an initialized Poisson distribution object

Author(s)

Richèl J.C. Bilderbeek

`is_init_rln_clock_model` *Determine if x is an initialized relaxed log-normal clock_model object*

Description

Determine if x is an initialized relaxed log-normal clock_model object

Usage

```
is_init_rln_clock_model(rln_clock_model)
```

Arguments

`rln_clock_model`
a Relaxed Log-Normal clock model, as returned by [create_rln_clock_model](#)

Value

TRUE if x is an initialized relaxed log-normal clock_model object, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

`is_init_site_model` *Determine if x is an initialized site model, as created by [create_site_model](#)*

Description

Determine if x is an initialized site model, as created by [create_site_model](#)

Usage

```
is_init_site_model(x)
```

Arguments

x the object to check if it is an initialized site_models object

Value

TRUE if x is an initialized site model

Author(s)

Richèl J.C. Bilderbeek

`is_init_strict_clock_model`
Determine if x is an initialized strict clock_model object

Description

Determine if x is an initialized strict clock_model object

Usage

```
is_init_strict_clock_model(strict_clock_model)
```

Arguments

`strict_clock_model`
a strict clock model, as returned by [create_strict_clock_model](#)

Value

TRUE if x is an initialized strict clock_model object

Author(s)

Richèl J.C. Bilderbeek

`is_init_tn93_site_model`

Determine if x is an initialized tn93 site model as created by [create_tn93_site_model](#)

Description

Determine if x is an initialized tn93 site model as created by [create_tn93_site_model](#)

Usage

```
is_init_tn93_site_model(x)
```

Arguments

x the object to check if it is an initialized TN93 site model

Value

TRUE if x is an initialized TN93 site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

tn93_site_model <- create_tn93_site_model()
expect_false(is_init_tn93_site_model(tn93_site_model))
tn93_site_model <- init_tn93_site_model(tn93_site_model)
expect_true(is_init_tn93_site_model(tn93_site_model))
```

`is_init_tree_prior` *Determine if x is an initialized tree_prior objects*

Description

Determine if x is an initialized tree_prior objects

Usage

```
is_init_tree_prior(x)
```

Arguments

x the object to check if it is an initialized tree_priors object

Value

TRUE if x is an initialized tree_prior object

Author(s)

Richèl J.C. Bilderbeek

is_init_uniform_distr *Determine if x is an initialized uniform distribution object as created by [create_uniform_distr](#)*

Description

Determine if x is an initialized uniform distribution object as created by [create_uniform_distr](#)

Usage

```
is_init_uniform_distr(x)
```

Arguments

x the object to check if it is an initialized uniform distribution object

Value

TRUE if x is an initialized uniform distribution object

Author(s)

Richèl J.C. Bilderbeek

`is_init_yule_tree_prior`*Determine if x is an initialized Yule tree_prior object*

Description

Determine if x is an initialized Yule tree_prior object

Usage

```
is_init_yule_tree_prior(x)
```

Arguments

x the object to check if it is an initialized Yule tree prior object

Value

TRUE if x is an initialized Yule tree_prior object

Author(s)

Richèl J.C. Bilderbeek

`is_inv_gamma_distr`*Determine if the object is a valid inverse-gamma distribution as created by [create_inv_gamma_distr](#)*

Description

Determine if the object is a valid inverse-gamma distribution as created by [create_inv_gamma_distr](#)

Usage

```
is_inv_gamma_distr(x)
```

Arguments

x an object, to be determined if it is a valid inverse-gamma distribution

Value

TRUE if x is a valid inverse-gamma distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use `is_distr` to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_inv_gamma_distr(create_inv_gamma_distr()))

expect_false(is_inv_gamma_distr(create_laplace_distr()))
expect_false(is_inv_gamma_distr(NA))
expect_false(is_inv_gamma_distr(NULL))
expect_false(is_inv_gamma_distr("nonsense"))
```

is_in_patterns	<i>Is there at least one regular expression having a match with the line?</i>
----------------	---

Description

Is there at least one regular expression having a match with the line?

Usage

```
is_in_patterns(line, patterns)
```

Arguments

line	a line of text
patterns	one or more regular expression patterns

Value

TRUE if there is at least one match found

Author(s)

Richèl J.C. Bilderbeek

is_jc69_site_model *Determine if the object is a valid JC69 site model*

Description

Determine if the object is a valid JC69 site model

Usage

```
is_jc69_site_model(x)
```

Arguments

x an object, to be determined if it is a valid JC69 site model

Value

TRUE if x is a valid JC69 site model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# site models
expect_false(is_jc69_site_model(create_gtr_site_model()))
expect_false(is_jc69_site_model(create_hky_site_model()))
expect_true(is_jc69_site_model(create_jc69_site_model()))
expect_false(is_jc69_site_model(create_tn93_site_model()))

# other models
expect_false(is_jc69_site_model(NA))
expect_false(is_jc69_site_model(NULL))
expect_false(is_jc69_site_model("nonsense"))
expect_false(is_jc69_site_model(create_strict_clock_model()))
expect_false(is_jc69_site_model(create_bd_tree_prior()))
expect_false(is_jc69_site_model(create_mcmc()))
```

is_kappa_1_param	<i>Determine if the object is a valid kappa 1 parameter</i>
------------------	---

Description

Determine if the object is a valid kappa 1 parameter

Usage

```
is_kappa_1_param(x)
```

Arguments

x an object, to be determined if it is a valid kappa 1 parameter

Value

TRUE if x is a valid kappa 1 parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

kappa 1 parameters are returned by [create_kappa_1_param](#)

Examples

```
library(testthat)

expect_false(is_kappa_1_param(create_alpha_param()))
expect_false(is_kappa_1_param(create_beta_param()))
expect_false(is_kappa_1_param(create_clock_rate_param()))
expect_true(is_kappa_1_param(create_kappa_1_param()))
expect_false(is_kappa_1_param(create_kappa_2_param()))
expect_false(is_kappa_1_param(create_lambda_param()))
expect_false(is_kappa_1_param(create_m_param()))
expect_false(is_kappa_1_param(create_mean_param()))
expect_false(is_kappa_1_param(create_mu_param()))
expect_false(is_kappa_1_param(create_rate_ac_param()))
expect_false(is_kappa_1_param(create_rate_ag_param()))
expect_false(is_kappa_1_param(create_rate_at_param()))
expect_false(is_kappa_1_param(create_rate_cg_param()))
expect_false(is_kappa_1_param(create_rate_ct_param()))
expect_false(is_kappa_1_param(create_rate_gt_param()))
expect_false(is_kappa_1_param(create_s_param()))
expect_false(is_kappa_1_param(create_scale_param()))
expect_false(is_kappa_1_param(create_sigma_param()))
```

```

expect_false(is_kappa_1_param(NA))
expect_false(is_kappa_1_param(NULL))
expect_false(is_kappa_1_param("nonsense"))
expect_false(is_kappa_1_param(create_jc69_site_model()))
expect_false(is_kappa_1_param(create_strict_clock_model()))
expect_false(is_kappa_1_param(create_yule_tree_prior()))
expect_false(is_kappa_1_param(create_mcmc()))

```

is_kappa_2_param	<i>Determine if the object is a valid kappa 2 parameter</i>
------------------	---

Description

Determine if the object is a valid kappa 2 parameter

Usage

```
is_kappa_2_param(x)
```

Arguments

x an object, to be determined if it is a valid kappa 2 parameter

Value

TRUE if x is a valid kappa_2 parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

kappa 2 parameters are returned by [create_kappa_2_param](#)

Examples

```

library(testthat)

expect_false(is_kappa_2_param(create_alpha_param()))
expect_false(is_kappa_2_param(create_beta_param()))
expect_false(is_kappa_2_param(create_clock_rate_param()))
expect_false(is_kappa_2_param(create_kappa_1_param()))
expect_true(is_kappa_2_param(create_kappa_2_param()))
expect_false(is_kappa_2_param(create_lambda_param()))
expect_false(is_kappa_2_param(create_m_param()))
expect_false(is_kappa_2_param(create_mean_param()))
expect_false(is_kappa_2_param(create_mu_param()))
expect_false(is_kappa_2_param(create_rate_ac_param()))

```

```
expect_false(is_kappa_2_param(create_rate_ag_param()))
expect_false(is_kappa_2_param(create_rate_at_param()))
expect_false(is_kappa_2_param(create_rate_cg_param()))
expect_false(is_kappa_2_param(create_rate_ct_param()))
expect_false(is_kappa_2_param(create_rate_gt_param()))
expect_false(is_kappa_2_param(create_s_param()))
expect_false(is_kappa_2_param(create_scale_param()))
expect_false(is_kappa_2_param(create_sigma_param()))

expect_false(is_kappa_2_param(NA))
expect_false(is_kappa_2_param(NULL))
expect_false(is_kappa_2_param("nonsense"))
expect_false(is_kappa_2_param(create_jc69_site_model()))
expect_false(is_kappa_2_param(create_strict_clock_model()))
expect_false(is_kappa_2_param(create_yule_tree_prior()))
expect_false(is_kappa_2_param(create_mcmc()))
```

is_lambda_param

Determine if the object is a valid lambda parameter

Description

Determine if the object is a valid lambda parameter

Usage

```
is_lambda_param(x)
```

Arguments

x an object, to be determined if it is a valid lambda parameter

Value

TRUE if x is a valid lambda parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

lambda parameters are returned by [create_lambda_param](#)

Examples

```
library(testthat)

expect_false(is_lambda_param(create_alpha_param()))
expect_false(is_lambda_param(create_beta_param()))
expect_false(is_lambda_param(create_clock_rate_param()))
expect_false(is_lambda_param(create_kappa_1_param()))
expect_false(is_lambda_param(create_kappa_2_param()))
expect_true(is_lambda_param(create_lambda_param()))
expect_false(is_lambda_param(create_m_param()))
expect_false(is_lambda_param(create_mean_param()))
expect_false(is_lambda_param(create_mu_param()))
expect_false(is_lambda_param(create_rate_ac_param()))
expect_false(is_lambda_param(create_rate_ag_param()))
expect_false(is_lambda_param(create_rate_at_param()))
expect_false(is_lambda_param(create_rate_cg_param()))
expect_false(is_lambda_param(create_rate_ct_param()))
expect_false(is_lambda_param(create_rate_gt_param()))
expect_false(is_lambda_param(create_s_param()))
expect_false(is_lambda_param(create_scale_param()))
expect_false(is_lambda_param(create_sigma_param()))

expect_false(is_lambda_param(NA))
expect_false(is_lambda_param(NULL))
expect_false(is_lambda_param("nonsense"))
expect_false(is_lambda_param(create_jc69_site_model()))
expect_false(is_lambda_param(create_strict_clock_model()))
expect_false(is_lambda_param(create_yule_tree_prior()))
expect_false(is_lambda_param(create_mcmc()))
```

is_laplace_distr	<i>Determine if the object is a valid Laplace distribution, as created by create_laplace_distr</i>
------------------	--

Description

Determine if the object is a valid Laplace distribution, as created by [create_laplace_distr](#)

Usage

```
is_laplace_distr(x)
```

Arguments

x an object, to be determined if it is a valid Laplace distribution

Value

TRUE if x is a valid Laplace distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_laplace_distr(create_laplace_distr()))

expect_false(is_laplace_distr(create_log_normal_distr()))
expect_false(is_laplace_distr(NA))
expect_false(is_laplace_distr(NULL))
expect_false(is_laplace_distr("nonsense"))
```

is_log_normal_distr	<i>Determine if the object is a valid log-normal distribution, as created by create_log_normal_distr</i>
---------------------	--

Description

Determine if the object is a valid log-normal distribution, as created by [create_log_normal_distr](#)

Usage

```
is_log_normal_distr(x)
```

Arguments

x an object, to be determined if it is a valid log-normal distribution

Value

TRUE if x is a valid log-normal distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_log_normal_distr(create_log_normal_distr()))

expect_false(is_log_normal_distr(create_normal_distr()))
expect_false(is_distr(NA))
expect_false(is_distr(NULL))
expect_false(is_distr("nonsense"))
```

is_mcmc

Determine if the object is a valid MCMC

Description

Determine if the object is a valid MCMC

Usage

```
is_mcmc(x)
```

Arguments

x an object, to be determined if it is a valid MCMC

Value

TRUE if x is a valid MCMC, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_mcmc](#) to create an MCMC

Examples

```
library(testthat)

expect_true(is_mcmc(create_mcmc()))
expect_true(is_mcmc(create_ns_mcmc()))

expect_false(is_mcmc("nonsense"))
expect_false(is_mcmc(NULL))
expect_false(is_mcmc(NA))
expect_false(is_mcmc(""))
expect_false(is_mcmc(c()))
```

is_mcmc_nested_sampling

Determine if the object is a valid Nested-Sampling MCMC, as used in [1]

Description

Determine if the object is a valid Nested-Sampling MCMC, as used in [1]

Usage

```
is_mcmc_nested_sampling(x)
```

Arguments

x an object, to be determined if it is a valid MCMC

Value

TRUE if x is a valid Nested-Sampling MCMC, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

References

* [1] Patricio Maturana Russel, Brendon J Brewer, Steffen Klaere, Remco R Bouckaert; Model Selection and Parameter Inference in Phylogenetics Using Nested Sampling, Systematic Biology, 2018, syy050, <https://doi.org/10.1093/sysbio/syy050>

See Also

Use [create_ns_mcmc](#) to create an NS MCMC

Examples

```
testthat::expect_false(is_nested_sampling_mcmc(create_mcmc()))
testthat::expect_true(
  is_nested_sampling_mcmc(create_ns_mcmc())
)
testthat::expect_false(is_nested_sampling_mcmc("nonsense"))
```

is_mean_param	<i>Determine if the object is a valid mean parameter</i>
---------------	--

Description

Determine if the object is a valid mean parameter

Usage

```
is_mean_param(x)
```

Arguments

x an object, to be determined if it is a valid mean parameter, as created by [create_mean_param](#))

Value

TRUE if x is a valid mean parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_false(is_mean_param(create_alpha_param()))
expect_false(is_mean_param(create_beta_param()))
expect_false(is_mean_param(create_clock_rate_param()))
expect_false(is_mean_param(create_kappa_1_param()))
expect_false(is_mean_param(create_kappa_2_param()))
expect_false(is_mean_param(create_lambda_param()))
expect_false(is_mean_param(create_m_param()))
expect_true(is_mean_param(create_mean_param()))
expect_false(is_mean_param(create_mu_param()))
expect_false(is_mean_param(create_rate_ac_param()))
expect_false(is_mean_param(create_rate_ag_param()))
expect_false(is_mean_param(create_rate_at_param()))
expect_false(is_mean_param(create_rate_cg_param()))
expect_false(is_mean_param(create_rate_ct_param()))
expect_false(is_mean_param(create_rate_gt_param()))
expect_false(is_mean_param(create_s_param()))
expect_false(is_mean_param(create_scale_param()))
expect_false(is_mean_param(create_sigma_param()))

expect_false(is_mean_param(NA))
expect_false(is_mean_param(NULL))
expect_false(is_mean_param("nonsense"))
expect_false(is_mean_param(create_jc69_site_model()))
```

```
expect_false(is_mean_param(create_strict_clock_model()))
expect_false(is_mean_param(create_yule_tree_prior()))
expect_false(is_mean_param(create_mcmc()))
```

is_mrca_align_ids_in_fastas

Determine if an MRCA prior's alignment IDs are present in the FASTA files

Description

Determine if an MRCA prior's alignment IDs are present in the FASTA files

Usage

```
is_mrca_align_ids_in_fastas(mrca_prior, fasta_filenames)
```

Arguments

mrca_prior a Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)
 fasta_filenames One or more FASTA filenames. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

Value

TRUE if the MRCA prior's alignment IDs is present in the FASTA files. Returns FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

is_mrca_align_id_in_fasta

Determine if an MRCA prior's alignment IDs is present in the FASTA file

Description

Determine if an MRCA prior's alignment IDs is present in the FASTA file

Usage

```
is_mrca_align_id_in_fasta(mrca_prior, fasta_filename)
```

Arguments

mrca_prior a Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)
 fasta_filename a FASTA filename. Use [get_fasta_filename](#) to obtain a testing FASTA filename.

Value

TRUE if the MRCA prior's alignment IDs is present in the FASTA file. Returns FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

is_mrca_prior	<i>Determine of the object is an empty (NA) or valid MRCA prior.</i>
---------------	--

Description

Determine of the object is an empty (NA) or valid MRCA prior.

Usage

```
is_mrca_prior(mrca_prior)
```

Arguments

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

Value

TRUE if x is an MRCA prior, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_mrca_prior(create_mrca_prior()))
# Also 'NA' is a valid MRCA prior,
# denoting that there no MRCA priors
expect_true(is_mrca_prior(NA))

expect_false(is_mrca_prior(NULL))
expect_false(is_mrca_prior("nonsense"))
```

is_mrca_prior_with_distr

See if x is one MRCA prior with a distribution

Description

See if x is one MRCA prior with a distribution

Usage

is_mrca_prior_with_distr(x)

Arguments

x the object to be tested

Value

TRUE if x is one MRCA prior with a distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

is_mu_param

Determine if the object is a valid mu parameter

Description

Determine if the object is a valid mu parameter

Usage

is_mu_param(x)

Arguments

x an object, to be determined if it is a valid mu parameter

Value

TRUE if x is a valid mu parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_mu_param](#) creates a mu parameter

Examples

```
library(testthat)

expect_false(is_mu_param(create_alpha_param()))
expect_false(is_mu_param(create_beta_param()))
expect_false(is_mu_param(create_clock_rate_param()))
expect_false(is_mu_param(create_kappa_1_param()))
expect_false(is_mu_param(create_kappa_2_param()))
expect_false(is_mu_param(create_lambda_param()))
expect_false(is_mu_param(create_m_param()))
expect_false(is_mu_param(create_mean_param()))
expect_true(is_mu_param(create_mu_param()))
expect_false(is_mu_param(create_rate_ac_param()))
expect_false(is_mu_param(create_rate_ag_param()))
expect_false(is_mu_param(create_rate_at_param()))
expect_false(is_mu_param(create_rate_cg_param()))
expect_false(is_mu_param(create_rate_ct_param()))
expect_false(is_mu_param(create_rate_gt_param()))
expect_false(is_mu_param(create_s_param()))
expect_false(is_mu_param(create_scale_param()))
expect_false(is_mu_param(create_sigma_param()))

expect_false(is_mu_param(NA))
expect_false(is_mu_param(NULL))
expect_false(is_mu_param("nonsense"))
expect_false(is_mu_param(create_jc69_site_model()))
expect_false(is_mu_param(create_strict_clock_model()))
expect_false(is_mu_param(create_yule_tree_prior()))
expect_false(is_mu_param(create_mcmc()))
```

is_m_param

Determine if the object is a valid m parameter

Description

Determine if the object is a valid m parameter

Usage

```
is_m_param(x)
```

Arguments

x an object, to be determined if it is a valid m parameter

Value

TRUE if x is a valid m parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_false(is_m_param(create_alpha_param()))
expect_false(is_m_param(create_beta_param()))
expect_false(is_m_param(create_clock_rate_param()))
expect_false(is_m_param(create_kappa_1_param()))
expect_false(is_m_param(create_kappa_2_param()))
expect_false(is_m_param(create_lambda_param()))
expect_true(is_m_param(create_m_param()))
expect_false(is_m_param(create_mean_param()))
expect_false(is_m_param(create_mu_param()))
expect_false(is_m_param(create_rate_ac_param()))
expect_false(is_m_param(create_rate_ag_param()))
expect_false(is_m_param(create_rate_at_param()))
expect_false(is_m_param(create_rate_cg_param()))
expect_false(is_m_param(create_rate_ct_param()))
expect_false(is_m_param(create_rate_gt_param()))
expect_false(is_m_param(create_s_param()))
expect_false(is_m_param(create_scale_param()))
expect_false(is_m_param(create_sigma_param()))

expect_false(is_m_param(NA))
expect_false(is_m_param(NULL))
expect_false(is_m_param("nonsense"))
expect_false(is_m_param(create_jc69_site_model()))
expect_false(is_m_param(create_strict_clock_model()))
expect_false(is_m_param(create_yule_tree_prior()))
expect_false(is_m_param(create_mcmc()))
```

is_normal_distr	<i>Determine if the object is a valid normal distribution as created by</i> create_normal_distr
-----------------	--

Description

Determine if the object is a valid normal distribution as created by [create_normal_distr](#)

Usage

```
is_normal_distr(x)
```


Arguments

x an object, to be determined if it is a valid normal distribution

Value

TRUE if x is a valid normal distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_normal_distr(create_normal_distr()))

expect_false(is_normal_distr(create_one_div_x_distr()))
expect_false(is_normal_distr(NA))
expect_false(is_normal_distr(NULL))
expect_false(is_normal_distr("nonsense"))
```

is_one_bool	<i>Check if the argument is one boolean</i>
-------------	---

Description

Check if the argument is one boolean

Usage

```
is_one_bool(x)
```

Arguments

x the argument to be tested to be boolean

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)
expect_true(is_one_bool(TRUE))
expect_true(is_one_bool(FALSE))
expect_false(is_one_bool(NULL))
expect_false(is_one_bool(NA))
expect_false(is_one_bool(c()))
expect_false(is_one_bool("nonsense"))
expect_false(is_one_bool(is_one_bool))
expect_false(is_one_bool(c(TRUE, FALSE)))
```

is_one_div_x_distr	<i>Determine if the object is a valid 1/x distribution, as created by create_one_div_x_distr</i>
--------------------	--

Description

Determine if the object is a valid 1/x distribution, as created by [create_one_div_x_distr](#)

Usage

```
is_one_div_x_distr(x)
```

Arguments

x an object, to be determined if it is a valid 1/x distribution

Value

TRUE if x is a valid 1/x distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_one_div_x_distr(create_one_div_x_distr()))

expect_false(is_one_div_x_distr(create_poisson_distr()))
expect_false(is_one_div_x_distr(NA))
expect_false(is_one_div_x_distr(NULL))
expect_false(is_one_div_x_distr("nonsense"))
```

is_one_double	<i>Determines if the argument is a double</i>
---------------	---

Description

Determines if the argument is a double

Usage

```
is_one_double(x)
```

Arguments

x the object to be determined of if it is one double

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_one_double(314))
expect_true(is_one_double(0))
expect_true(is_one_double(-314))
expect_true(is_one_double(3.14))
expect_false(is_one_double(NULL))
expect_false(is_one_double(NA))
expect_false(is_one_double(Inf))
expect_false(is_one_double("nonsense"))
expect_false(is_one_double(is_one_double))
expect_false(is_one_double(c()))
expect_false(is_one_double(c(1, 2)))
```

is_one_int	<i>Determines if the argument is a whole number</i>
------------	---

Description

Determines if the argument is a whole number

Usage

```
is_one_int(x, tolerance = .Machine$double.eps^0.5)
```

Arguments

x the object to be determined of if it is one integer
tolerance tolerance to rounding errors

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_one_int(314))
expect_true(is_one_int(0))
expect_true(is_one_int(-314))
expect_false(is_one_int(3.14))
expect_false(is_one_int(NULL))
expect_false(is_one_int(NA))
expect_false(is_one_int(Inf))
expect_false(is_one_int("nonsense"))
expect_false(is_one_int(c()))
expect_false(is_one_int(c(1, 2)))
```

is_one_na	<i>Determines if x is one NA</i>
-----------	----------------------------------

Description

Determines if x is one NA

Usage

```
is_one_na(x)
```

Arguments

x the object to be determined if it is one NA

Value

TRUE if x is one NA, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```

testit::assert(is_one_na(NA))
testit::assert(!is_one_na(NULL))
testit::assert(!is_one_na(42))
testit::assert(!is_one_na("Hello"))
testit::assert(!is_one_na(3.14))
testit::assert(!is_one_na(c(NA, NA)))

```

is_param

*Determine if the object is a valid parameter***Description**

Determine if the object is a valid parameter

Usage

```
is_param(x)
```

Arguments

x an object, to be determined if it is a valid parameter, as created by [create_param](#))

Value

TRUE if x is a valid parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```

library(testthat)

expect_true(is_param(create_alpha_param()))
expect_true(is_param(create_beta_param()))
expect_true(is_param(create_clock_rate_param()))
expect_true(is_param(create_kappa_1_param()))
expect_true(is_param(create_kappa_2_param()))
expect_true(is_param(create_lambda_param()))
expect_true(is_param(create_m_param()))
expect_true(is_param(create_mean_param()))
expect_true(is_param(create_mu_param()))
expect_true(is_param(create_rate_ac_param()))
expect_true(is_param(create_rate_ag_param()))
expect_true(is_param(create_rate_at_param()))
expect_true(is_param(create_rate_cg_param()))
expect_true(is_param(create_rate_ct_param()))

```

```

expect_true(is_param(create_rate_gt_param()))
expect_true(is_param(create_s_param()))
expect_true(is_param(create_scale_param()))
expect_true(is_param(create_sigma_param()))

expect_false(is_param(NA))
expect_false(is_param(NULL))
expect_false(is_param("nonsense"))
expect_false(is_param(create_jc69_site_model()))
expect_false(is_param(create_strict_clock_model()))
expect_false(is_param(create_yule_tree_prior()))
expect_false(is_param(create_mcmc()))

```

is_param_name	<i>Determines if the name is a valid parameter name</i>
---------------	---

Description

Determines if the name is a valid parameter name

Usage

```
is_param_name(name)
```

Arguments

name	the name to be tested
------	-----------------------

Value

TRUE if the name is a valid parameter name, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```

library(testthat)

expect_true(is_param_name("alpha"))
expect_true(is_param_name("beta"))
expect_true(is_param_name("clock_rate"))
expect_true(is_param_name("kappa_1"))
expect_true(is_param_name("kappa_2"))
expect_true(is_param_name("lambda"))
expect_true(is_param_name("m"))
expect_true(is_param_name("mean"))
expect_true(is_param_name("mu"))
expect_true(is_param_name("rate_ac"))

```

```
expect_true(is_param_name("rate_ag"))
expect_true(is_param_name("rate_at"))
expect_true(is_param_name("rate_cg"))
expect_true(is_param_name("rate_ct"))
expect_true(is_param_name("rate_gt"))
expect_true(is_param_name("s"))
expect_true(is_param_name("scale"))
expect_true(is_param_name("sigma"))

expect_false(is_param_name("nonsense"))
expect_false(is_param_name(NA))
expect_false(is_param_name(NULL))
expect_false(is_param_name(""))
expect_false(is_param_name(c()))
```

is_phylo	<i>Checks if the input is a phylogeny</i>
----------	---

Description

Checks if the input is a phylogeny

Usage

```
is_phylo(x)
```

Arguments

x input to be checked

Value

TRUE or FALSE

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [check_phylogeny](#) to check for a phylogeny

Examples

```
phylogeny <- ape::read.tree(text = "(a:15,b:15):1;")
testit::assert(is_phylo(phylogeny))

testit::assert(!is_phylo("nonsense"))
testit::assert(!is_phylo(NA))
testit::assert(!is_phylo(NULL))
```

is_poisson_distr	<i>Determine if the object is a valid Poisson distribution as created by create_poisson_distr</i>
------------------	---

Description

Determine if the object is a valid Poisson distribution as created by [create_poisson_distr](#)

Usage

```
is_poisson_distr(x)
```

Arguments

x an object, to be determined if it is a valid Poisson distribution

Value

TRUE if x is a valid Poisson distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_poisson_distr(create_poisson_distr()))

expect_false(is_poisson_distr(create_uniform_distr()))
expect_false(is_distr(NA))
expect_false(is_distr(NULL))
expect_false(is_distr("nonsense"))
```

is_rate_ac_param	<i>Determine if the object is a valid 'rate AC' parameter</i>
------------------	---

Description

Determine if the object is a valid 'rate AC' parameter

Usage

```
is_rate_ac_param(x)
```

Arguments

x an object, to be determined if it is a valid 'rate AC' parameter

Value

TRUE if x is a valid 'rate AC' parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_rate_ac_param](#) creates a 'rate AC' parameter

Examples

```
library(testthat)

expect_false(is_rate_ac_param(create_alpha_param()))
expect_false(is_rate_ac_param(create_beta_param()))
expect_false(is_rate_ac_param(create_clock_rate_param()))
expect_false(is_rate_ac_param(create_kappa_1_param()))
expect_false(is_rate_ac_param(create_kappa_2_param()))
expect_false(is_rate_ac_param(create_lambda_param()))
expect_false(is_rate_ac_param(create_m_param()))
expect_false(is_rate_ac_param(create_mean_param()))
expect_false(is_rate_ac_param(create_mu_param()))
expect_true(is_rate_ac_param(create_rate_ac_param()))
expect_false(is_rate_ac_param(create_rate_ag_param()))
expect_false(is_rate_ac_param(create_rate_at_param()))
expect_false(is_rate_ac_param(create_rate_cg_param()))
expect_false(is_rate_ac_param(create_rate_ct_param()))
expect_false(is_rate_ac_param(create_rate_gt_param()))
expect_false(is_rate_ac_param(create_s_param()))
expect_false(is_rate_ac_param(create_scale_param()))
expect_false(is_rate_ac_param(create_sigma_param()))
```

```

expect_false(is_rate_ac_param(NA))
expect_false(is_rate_ac_param(NULL))
expect_false(is_rate_ac_param("nonsense"))
expect_false(is_rate_ac_param(create_jc69_site_model()))
expect_false(is_rate_ac_param(create_strict_clock_model()))
expect_false(is_rate_ac_param(create_yule_tree_prior()))
expect_false(is_rate_ac_param(create_mcmc()))

```

is_rate_ag_param	<i>Determine if the object is a valid 'rate AG' parameter</i>
------------------	---

Description

Determine if the object is a valid 'rate AG' parameter

Usage

```
is_rate_ag_param(x)
```

Arguments

x an object, to be determined if it is a valid 'rate AG' parameter

Value

TRUE if x is a valid 'rate AG' parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_rate_ag_param](#) creates a 'rate AG' parameter

Examples

```

library(testthat)

expect_false(is_rate_ag_param(create_alpha_param()))
expect_false(is_rate_ag_param(create_beta_param()))
expect_false(is_rate_ag_param(create_clock_rate_param()))
expect_false(is_rate_ag_param(create_kappa_1_param()))
expect_false(is_rate_ag_param(create_kappa_2_param()))
expect_false(is_rate_ag_param(create_lambda_param()))
expect_false(is_rate_ag_param(create_m_param()))
expect_false(is_rate_ag_param(create_mean_param()))
expect_false(is_rate_ag_param(create_mu_param()))
expect_false(is_rate_ag_param(create_rate_ac_param()))

```

```
expect_true(is_rate_ag_param(create_rate_ag_param()))
expect_false(is_rate_ag_param(create_rate_at_param()))
expect_false(is_rate_ag_param(create_rate_cg_param()))
expect_false(is_rate_ag_param(create_rate_ct_param()))
expect_false(is_rate_ag_param(create_rate_gt_param()))
expect_false(is_rate_ag_param(create_s_param()))
expect_false(is_rate_ag_param(create_scale_param()))
expect_false(is_rate_ag_param(create_sigma_param()))

expect_false(is_rate_ag_param(NA))
expect_false(is_rate_ag_param(NULL))
expect_false(is_rate_ag_param("nonsense"))
expect_false(is_rate_ag_param(create_jc69_site_model()))
expect_false(is_rate_ag_param(create_strict_clock_model()))
expect_false(is_rate_ag_param(create_yule_tree_prior()))
expect_false(is_rate_ag_param(create_mcmc()))
```

is_rate_at_param	<i>Determine if the object is a valid 'rate AT' parameter</i>
------------------	---

Description

Determine if the object is a valid 'rate AT' parameter

Usage

```
is_rate_at_param(x)
```

Arguments

x an object, to be determined if it is a valid 'rate AT' parameter

Value

TRUE if x is a valid 'rate AT' parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_rate_at_param](#) creates a 'rate AT' parameter

Examples

```

library(testthat)

expect_false(is_rate_at_param(create_alpha_param()))
expect_false(is_rate_at_param(create_beta_param()))
expect_false(is_rate_at_param(create_clock_rate_param()))
expect_false(is_rate_at_param(create_kappa_1_param()))
expect_false(is_rate_at_param(create_kappa_2_param()))
expect_false(is_rate_at_param(create_lambda_param()))
expect_false(is_rate_at_param(create_m_param()))
expect_false(is_rate_at_param(create_mean_param()))
expect_false(is_rate_at_param(create_mu_param()))
expect_false(is_rate_at_param(create_rate_ac_param()))
expect_false(is_rate_at_param(create_rate_ag_param()))
expect_true(is_rate_at_param(create_rate_at_param()))
expect_false(is_rate_at_param(create_rate_cg_param()))
expect_false(is_rate_at_param(create_rate_ct_param()))
expect_false(is_rate_at_param(create_rate_gt_param()))
expect_false(is_rate_at_param(create_s_param()))
expect_false(is_rate_at_param(create_scale_param()))
expect_false(is_rate_at_param(create_sigma_param()))

expect_false(is_rate_at_param(NA))
expect_false(is_rate_at_param(NULL))
expect_false(is_rate_at_param("nonsense"))
expect_false(is_rate_at_param(create_jc69_site_model()))
expect_false(is_rate_at_param(create_strict_clock_model()))
expect_false(is_rate_at_param(create_yule_tree_prior()))
expect_false(is_rate_at_param(create_mcmc()))

```

is_rate_cg_param	<i>Determine if the object is a valid 'rate CG' parameter</i>
------------------	---

Description

Determine if the object is a valid 'rate CG' parameter

Usage

```
is_rate_cg_param(x)
```

Arguments

x an object, to be determined if it is a valid 'rate CG' parameter

Value

TRUE if x is a valid 'rate CG' parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_rate_cg_param](#) creates a 'rate CG' parameter

Examples

```
library(testthat)

expect_false(is_rate_cg_param(create_alpha_param()))
expect_false(is_rate_cg_param(create_beta_param()))
expect_false(is_rate_cg_param(create_clock_rate_param()))
expect_false(is_rate_cg_param(create_kappa_1_param()))
expect_false(is_rate_cg_param(create_kappa_2_param()))
expect_false(is_rate_cg_param(create_lambda_param()))
expect_false(is_rate_cg_param(create_m_param()))
expect_false(is_rate_cg_param(create_mean_param()))
expect_false(is_rate_cg_param(create_mu_param()))
expect_false(is_rate_cg_param(create_rate_ac_param()))
expect_false(is_rate_cg_param(create_rate_ag_param()))
expect_false(is_rate_cg_param(create_rate_at_param()))
expect_true(is_rate_cg_param(create_rate_cg_param()))
expect_false(is_rate_cg_param(create_rate_ct_param()))
expect_false(is_rate_cg_param(create_rate_gt_param()))
expect_false(is_rate_cg_param(create_s_param()))
expect_false(is_rate_cg_param(create_scale_param()))
expect_false(is_rate_cg_param(create_sigma_param()))

expect_false(is_rate_cg_param(NA))
expect_false(is_rate_cg_param(NULL))
expect_false(is_rate_cg_param("nonsense"))
expect_false(is_rate_cg_param(create_jc69_site_model()))
expect_false(is_rate_cg_param(create_strict_clock_model()))
expect_false(is_rate_cg_param(create_yule_tree_prior()))
expect_false(is_rate_cg_param(create_mcmc()))
```

is_rate_ct_param	<i>Determine if the object is a valid 'rate CT' parameter</i>
------------------	---

Description

Determine if the object is a valid 'rate CT' parameter

Usage

```
is_rate_ct_param(x)
```

Arguments

x an object, to be determined if it is a valid 'rate CT' parameter

Value

TRUE if x is a valid 'rate CG' parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_rate_ct_param](#) creates a 'rate CT' parameter

Examples

```
library(testthat)

expect_false(is_rate_ct_param(create_alpha_param()))
expect_false(is_rate_ct_param(create_beta_param()))
expect_false(is_rate_ct_param(create_clock_rate_param()))
expect_false(is_rate_ct_param(create_kappa_1_param()))
expect_false(is_rate_ct_param(create_kappa_2_param()))
expect_false(is_rate_ct_param(create_lambda_param()))
expect_false(is_rate_ct_param(create_m_param()))
expect_false(is_rate_ct_param(create_mean_param()))
expect_false(is_rate_ct_param(create_mu_param()))
expect_false(is_rate_ct_param(create_rate_ac_param()))
expect_false(is_rate_ct_param(create_rate_ag_param()))
expect_false(is_rate_ct_param(create_rate_at_param()))
expect_false(is_rate_ct_param(create_rate_cg_param()))
expect_true(is_rate_ct_param(create_rate_ct_param()))
expect_false(is_rate_ct_param(create_rate_gt_param()))
expect_false(is_rate_ct_param(create_s_param()))
expect_false(is_rate_ct_param(create_scale_param()))
expect_false(is_rate_ct_param(create_sigma_param()))

expect_false(is_rate_ct_param(NA))
expect_false(is_rate_ct_param(NULL))
expect_false(is_rate_ct_param("nonsense"))
expect_false(is_rate_ct_param(create_jc69_site_model()))
expect_false(is_rate_ct_param(create_strict_clock_model()))
expect_false(is_rate_ct_param(create_yule_tree_prior()))
expect_false(is_rate_ct_param(create_mcmc()))
```

is_rate_gt_param	<i>Determine if the object is a valid 'rate GT' parameter</i>
------------------	---

Description

Determine if the object is a valid 'rate GT' parameter

Usage

```
is_rate_gt_param(x)
```

Arguments

x an object, to be determined if it is a valid 'rate GT' parameter

Value

TRUE if x is a valid 'rate GT' parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_rate_gt_param](#) creates a 'rate GT' parameter

Examples

```
library(testthat)

expect_false(is_rate_gt_param(create_alpha_param()))
expect_false(is_rate_gt_param(create_beta_param()))
expect_false(is_rate_gt_param(create_clock_rate_param()))
expect_false(is_rate_gt_param(create_kappa_1_param()))
expect_false(is_rate_gt_param(create_kappa_2_param()))
expect_false(is_rate_gt_param(create_lambda_param()))
expect_false(is_rate_gt_param(create_m_param()))
expect_false(is_rate_gt_param(create_mean_param()))
expect_false(is_rate_gt_param(create_mu_param()))
expect_false(is_rate_gt_param(create_rate_ac_param()))
expect_false(is_rate_gt_param(create_rate_ag_param()))
expect_false(is_rate_gt_param(create_rate_at_param()))
expect_false(is_rate_gt_param(create_rate_cg_param()))
expect_false(is_rate_gt_param(create_rate_ct_param()))
expect_true(is_rate_gt_param(create_rate_gt_param()))
expect_false(is_rate_gt_param(create_s_param()))
expect_false(is_rate_gt_param(create_scale_param()))
expect_false(is_rate_gt_param(create_sigma_param()))
```

```

expect_false(is_rate_gt_param(NA))
expect_false(is_rate_gt_param(NULL))
expect_false(is_rate_gt_param("nonsense"))
expect_false(is_rate_gt_param(create_jc69_site_model()))
expect_false(is_rate_gt_param(create_strict_clock_model()))
expect_false(is_rate_gt_param(create_yule_tree_prior()))
expect_false(is_rate_gt_param(create_mcmc()))

```

is_rln_clock_model *Determine if the object is a valid relaxed log normal clock model*

Description

Determine if the object is a valid relaxed log normal clock model

Usage

```
is_rln_clock_model(x)
```

Arguments

x an object, to be determined if it is a valid relaxed log normal clock model, as created by [create_rln_clock_model](#))

Value

TRUE if x is a valid relaxed log normal clock model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_clock_model](#) shows an overview of functions to create a clock model

Examples

```

library(testthat)

expect_false(is_rln_clock_model(create_strict_clock_model()))
expect_true(is_rln_clock_model(create_rln_clock_model()))

expect_false(is_rln_clock_model(NA))
expect_false(is_rln_clock_model(NULL))
expect_false(is_rln_clock_model("nonsense"))
expect_false(is_rln_clock_model(create_jc69_site_model()))
expect_false(is_rln_clock_model(create_mcmc()))

```

is_scale_param	<i>Determine if the object is a valid scale parameter</i>
----------------	---

Description

Determine if the object is a valid scale parameter

Usage

```
is_scale_param(x)
```

Arguments

x an object, to be determined if it is a valid scale parameter

Value

TRUE if x is a valid scale parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_false(is_scale_param(create_alpha_param()))
expect_false(is_scale_param(create_beta_param()))
expect_false(is_scale_param(create_clock_rate_param()))
expect_false(is_scale_param(create_kappa_1_param()))
expect_false(is_scale_param(create_kappa_2_param()))
expect_false(is_scale_param(create_lambda_param()))
expect_false(is_scale_param(create_m_param()))
expect_false(is_scale_param(create_mean_param()))
expect_false(is_scale_param(create_mu_param()))
expect_false(is_scale_param(create_rate_ac_param()))
expect_false(is_scale_param(create_rate_ag_param()))
expect_false(is_scale_param(create_rate_at_param()))
expect_false(is_scale_param(create_rate_cg_param()))
expect_false(is_scale_param(create_rate_ct_param()))
expect_false(is_scale_param(create_rate_gt_param()))
expect_false(is_scale_param(create_s_param()))
expect_true(is_scale_param(create_scale_param()))
expect_false(is_scale_param(create_sigma_param()))

expect_false(is_scale_param(NA))
expect_false(is_scale_param(NULL))
expect_false(is_scale_param("nonsense"))
expect_false(is_scale_param(create_jc69_site_model()))
```

```
expect_false(is_scale_param(create_strict_clock_model()))
expect_false(is_scale_param(create_yule_tree_prior()))
expect_false(is_scale_param(create_mcmc()))
```

is_sigma_param	<i>Determine if the object is a valid sigma parameter</i>
----------------	---

Description

Determine if the object is a valid sigma parameter

Usage

```
is_sigma_param(x)
```

Arguments

x an object, to be determined if it is a valid sigma parameter

Value

TRUE if x is a valid sigma parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_false(is_sigma_param(create_alpha_param()))
expect_false(is_sigma_param(create_beta_param()))
expect_false(is_sigma_param(create_clock_rate_param()))
expect_false(is_sigma_param(create_kappa_1_param()))
expect_false(is_sigma_param(create_kappa_2_param()))
expect_false(is_sigma_param(create_lambda_param()))
expect_false(is_sigma_param(create_m_param()))
expect_false(is_sigma_param(create_mean_param()))
expect_false(is_sigma_param(create_mu_param()))
expect_false(is_sigma_param(create_rate_ac_param()))
expect_false(is_sigma_param(create_rate_ag_param()))
expect_false(is_sigma_param(create_rate_at_param()))
expect_false(is_sigma_param(create_rate_cg_param()))
expect_false(is_sigma_param(create_rate_ct_param()))
expect_false(is_sigma_param(create_rate_gt_param()))
expect_false(is_sigma_param(create_s_param()))
expect_false(is_sigma_param(create_scale_param()))
expect_true(is_sigma_param(create_sigma_param()))
```

```
expect_false(is_sigma_param(NA))
expect_false(is_sigma_param(NULL))
expect_false(is_sigma_param("nonsense"))
expect_false(is_sigma_param(create_jc69_site_model()))
expect_false(is_sigma_param(create_strict_clock_model()))
expect_false(is_sigma_param(create_yule_tree_prior()))
expect_false(is_sigma_param(create_mcmc()))
```

is_site_model	<i>Determine if the object is a valid site_model</i>
---------------	--

Description

Determine if the object is a valid site_model

Usage

```
is_site_model(x)
```

Arguments

x an object, to be determined if it is a site_model

Value

TRUE if the site_model is a valid site_model, FALSE otherwise

See Also

A site model can be created using [create_site_model](#)

Examples

```
library(testthat)

# site models
expect_true(is_site_model(create_gtr_site_model()))
expect_true(is_site_model(create_hky_site_model()))
expect_true(is_site_model(create_jc69_site_model()))
expect_true(is_site_model(create_tn93_site_model()))

# other models
expect_false(is_site_model(NA))
expect_false(is_site_model(NULL))
expect_false(is_site_model("nonsense"))
expect_false(is_site_model(create_strict_clock_model()))
expect_false(is_site_model(create_bd_tree_prior()))
expect_false(is_site_model(create_mcmc()))
```

is_site_model_name *Determines if the name is a valid site_model name*

Description

Determines if the name is a valid site_model name

Usage

```
is_site_model_name(name)
```

Arguments

name the name to be tested

Value

TRUE if the name is a valid site_model name, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_site_model_name("JC69"))
expect_true(is_site_model_name("HKY"))
expect_true(is_site_model_name("TN93"))
expect_true(is_site_model_name("GTR"))
expect_false(is_site_model_name("nonsense"))
```

is_strict_clock_model *Determine if the object is a valid strict clock model, as returned by*
[create_strict_clock_model](#)

Description

Determine if the object is a valid strict clock model, as returned by [create_strict_clock_model](#)

Usage

```
is_strict_clock_model(x)
```

Arguments

x an object, to be determined if it is a valid strict clock model

Value

TRUE if x is a valid strict clock model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

[create_clock_model](#) shows an overview of functions to create a clock model

Examples

```
library(testthat)

expect_true(is_strict_clock_model(create_strict_clock_model()))
expect_false(is_strict_clock_model(create_rln_clock_model()))

expect_false(is_strict_clock_model(NA))
expect_false(is_strict_clock_model(NULL))
expect_false(is_strict_clock_model("nonsense"))
expect_false(is_strict_clock_model(create_jc69_site_model()))
expect_false(is_strict_clock_model(create_mcmc()))
```

is_s_param

Determine if the object is a valid s parameter

Description

Determine if the object is a valid s parameter

Usage

```
is_s_param(x)
```

Arguments

x an object, to be determined if it is a valid s parameter

Value

TRUE if x is a valid s parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_false(is_s_param(create_alpha_param()))
expect_false(is_s_param(create_beta_param()))
expect_false(is_s_param(create_clock_rate_param()))
expect_false(is_s_param(create_kappa_1_param()))
expect_false(is_s_param(create_kappa_2_param()))
expect_false(is_s_param(create_lambda_param()))
expect_false(is_s_param(create_m_param()))
expect_false(is_s_param(create_mean_param()))
expect_false(is_s_param(create_mu_param()))
expect_false(is_s_param(create_rate_ac_param()))
expect_false(is_s_param(create_rate_ag_param()))
expect_false(is_s_param(create_rate_at_param()))
expect_false(is_s_param(create_rate_cg_param()))
expect_false(is_s_param(create_rate_ct_param()))
expect_false(is_s_param(create_rate_gt_param()))
expect_true(is_s_param(create_s_param()))
expect_false(is_s_param(create_scale_param()))
expect_false(is_s_param(create_sigma_param()))

expect_false(is_s_param(NA))
expect_false(is_s_param(NULL))
expect_false(is_s_param("nonsense"))
expect_false(is_s_param(create_jc69_site_model()))
expect_false(is_s_param(create_strict_clock_model()))
expect_false(is_s_param(create_yule_tree_prior()))
expect_false(is_s_param(create_mcmc()))
```

is_tn93_site_model *Determine if the object is a valid TN93 site model,*

Description

Determine if the object is a valid TN93 site model,

Usage

```
is_tn93_site_model(x)
```

Arguments

x an object, to be determined if it is a valid TN93 site model, as created by [create_tn93_site_model](#)

Value

TRUE if x is a valid TN93 site model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

# site models
expect_false(is_tn93_site_model(create_gtr_site_model()))
expect_false(is_tn93_site_model(create_hky_site_model()))
expect_false(is_tn93_site_model(create_jc69_site_model()))
expect_true(is_tn93_site_model(create_tn93_site_model()))

# other models
expect_false(is_tn93_site_model(NA))
expect_false(is_tn93_site_model(NULL))
expect_false(is_tn93_site_model("nonsense"))
expect_false(is_tn93_site_model(""))
expect_false(is_tn93_site_model(c()))
expect_false(is_tn93_site_model(create_strict_clock_model()))
expect_false(is_tn93_site_model(create_bd_tree_prior()))
expect_false(is_tn93_site_model(create_mcmc()))
```

is_tree_prior

Determine if an object is a valid tree prior

Description

Determine if an object is a valid tree prior

Usage

```
is_tree_prior(x)
```

Arguments

x an object

Value

TRUE if x is a valid tree_prior, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

tree priors can be created by [create_tree_prior](#)

Examples

```
testit::assert(is_tree_prior(create_bd_tree_prior()))
testit::assert(is_tree_prior(create_yule_tree_prior()))
testit::assert(!is_tree_prior("nonsense"))
```

is_tree_prior_name *Determines if the name is a valid tree prior name*

Description

Determines if the name is a valid tree prior name

Usage

```
is_tree_prior_name(name)
```

Arguments

name the name to be tested

Value

TRUE if the name is a valid tree_prior name, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_tree_prior_name("birth_death"))
expect_true(is_tree_prior_name("coalescent_bayesian_skyline"))
expect_true(is_tree_prior_name("coalescent_constant_population"))
expect_true(is_tree_prior_name("coalescent_exp_population"))
expect_true(is_tree_prior_name("yule"))
```

is_uniform_distr	<i>Determine if the object is a valid uniform distribution as created by create_uniform_distr</i>
------------------	---

Description

Determine if the object is a valid uniform distribution as created by [create_uniform_distr](#)

Usage

```
is_uniform_distr(x)
```

Arguments

x an object, to be determined if it is a valid uniform distribution

Value

TRUE if x is a valid uniform distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

use [is_distr](#) to see if x is any distribution

Examples

```
library(testthat)

expect_true(is_uniform_distr(create_uniform_distr()))
expect_false(is_uniform_distr(create_beta_distr()))

expect_false(is_uniform_distr(NA))
expect_false(is_uniform_distr(NULL))
expect_false(is_uniform_distr("nonsense"))
```

is_xml	<i>Checks if the text is a valid XML node, that is, it has a opening and matching closing tag</i>
--------	---

Description

Checks if the text is a valid XML node, that is, it has a opening and matching closing tag

Usage

```
is_xml(text)
```

Arguments

text	text to be determined to be valid
------	-----------------------------------

Value

TRUE if the text is valid XML, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

is_yule_tree_prior	<i>Determine if the object is a valid Yule tree prior,</i>
--------------------	--

Description

Determine if the object is a valid Yule tree prior,

Usage

```
is_yule_tree_prior(x)
```

Arguments

x	an object, to be determined if it is a valid Yule tree prior
---	--

Value

TRUE if x is a valid Yule tree prior, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_yule_tree_prior](#) to create a valid Yule tree prior

Examples

```
testit::assert(!is_yule_tree_prior(create_bd_tree_prior()))
testit::assert(!is_yule_tree_prior(create_cbs_tree_prior()))
testit::assert(!is_yule_tree_prior(create_ccp_tree_prior()))
testit::assert(!is_yule_tree_prior(create_cep_tree_prior()))
testit::assert( is_yule_tree_prior(create_yule_tree_prior()))
```

jc69_site_model_to_xml_subst_model

Converts a JC69 site model to XML, used in the substModel section

Description

Converts a JC69 site model to XML, used in the substModel section

Usage

```
jc69_site_model_to_xml_subst_model(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

mcmc_to_xml_run	<i>Converts an MCMC object to the run section's XML</i>
-----------------	---

Description

Converts an MCMC object to the run section's XML

Usage

```
mcmc_to_xml_run(mcmc)
```

Arguments

mcmc	one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC.
------	---

Value

the XML as text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

xml <- mcmc_to_xml_run(create_mcmc())
expect_equal(
  xml,
  "<run id=\"mcmc\" spec=\"MCMC\" chainLength=\"1e+07\">"
)
```

mcmc_to_xml_run_default	<i>Converts an MCMC object to the run section's XML for a default MCMC</i>
-------------------------	--

Description

Converts an MCMC object to the run section's XML for a default MCMC

Usage

```
mcmc_to_xml_run_default(mcmc)
```

Arguments

mcmc one MCMC. Use [create_mcmc](#) to create an MCMC. Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Value

the XML as text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

xml <- mcmc_to_xml_run_default(create_mcmc())

expect_equal(
  xml,
  "<run id=\"mcmc\" spec=\"MCMC\" chainLength=\"1e+07\">"
)
```

mcmc_to_xml_run_nested_sampling

Converts an MCMC object to the run section's XML for a Nested-Sampling MCMC

Description

Converts an MCMC object to the run section's XML for a Nested-Sampling MCMC

Usage

```
mcmc_to_xml_run_nested_sampling(mcmc)
```

Arguments

mcmc one MCMC. Use [create_mcmc](#) to create an MCMC. Use [create_ns_mcmc](#) to create an MCMC for a Nested Sampling run. Use [check_mcmc](#) to check if an MCMC is valid. Use [rename_mcmc_filenames](#) to rename the filenames in an MCMC.

Value

the XML as text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

xml <- mcmc_to_xml_run_nested_sampling(
  create_ns_mcmc()
)

expect_equal(
  xml,
  paste0(
    "<run id=\"mcmc\" spec=\"beast.gss.NS\" chainLength=\"1e+07\" ",
    "particleCount=\"1\" subChainLength=\"5000\" epsilon=\"1e-12\">"
  )
)
```

mrca_priors_to_xml_prior_distr

Creates the the distribution's prior section (which is part of a posterior distribution section) of a BEAST2 XML parameter file.

Description

These lines start with '<distribution id="prior"'

Usage

```
mrca_priors_to_xml_prior_distr(mrca_priors, has_non_strict_clock_model)
```

Arguments

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

has_non_strict_clock_model
 boolean to indicate that there is already at least one non-strict (i.e. relaxed log-normal) clock model

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
# <distribution id="likelihood" ...>
#   </distribution>
# </distribution>
```

mrca_priors_to_xml_state

*Converts one or more MRCA priors to the state section of the XML
as text*

Description

Converts one or more MRCA priors to the state section of the XML as text

Usage

```
mrca_priors_to_xml_state(mrca_priors, has_non_strict_clock_model = FALSE)
```

Arguments

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

has_non_strict_clock_model boolean to indicate that there is already at least one non-strict (i.e. relaxed log-normal) clock model

Value

lines of XML text, without indentation nor state tags

Author(s)

Richèl J.C. Bilderbeek

mrca_priors_to_xml_tracelog

Creates the MRCA priors' XML for the tracelog section

Description

Creates the MRCA priors' XML for the tracelog section

Usage

```
mrca_priors_to_xml_tracelog(clock_models, mrca_priors, tipdates_filename = NA)
```

Arguments

clock_models a list of one or more clock models, as returned by [create_clock_model](#)

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

tipdates_filename name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

the complete tracelog section is created by [create_tracelog_xml](#)

Examples

```
# <logger id="tracelog" ...>
# ' # Here
# </logger>
```

mrca_prior_to_xml_lh_distr

Converts an MRCA prior to the branchRateModel section of the XML as text.

Description

This function will be called if and only if there are MRCA priors and only supports strict clocks at the moment.

Usage

```
mrca_prior_to_xml_lh_distr(mrca_prior, has_non_strict_clock_model = FALSE)
```

Arguments

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

`has_non_strict_clock_model` boolean to indicate that there is already at least one non-strict (i.e. relaxed log-normal) clock model

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     </distribution>
#   <distribution id="likelihood" ...>
#     HERE, where the ID of the distribution is 'likelihood'
#   </distribution>
# </distribution>
```

mrca_prior_to_xml_prior_distr

Creates the distribution section in the prior section of the distribution section of a BEAST2 XML parameter file.

Description

These lines start with '<distribution id='

Usage

```
mrca_prior_to_xml_prior_distr(  
  mrca_prior,  
  has_non_strict_clock_model = FALSE,  
  taxa_names_with_ids = NULL  
)
```

Arguments

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

`has_non_strict_clock_model` boolean to indicate that there is already at least one non-strict (i.e. relaxed log-normal) clock model

`taxa_names_with_ids` taxa names that already have received an ID. Causes the XML to idref these

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">  
#   <distribution id="prior" spec="util.CompoundDistribution">  
#     HERE, where the ID of the distribution is 'prior'  
#   </distribution>  
#   <distribution id="likelihood" ...>  
#     </distribution>  
# </distribution>
```

 mrca_prior_to_xml_state

Creates the XML of an MRCA prior, as used in the state section

Description

Creates the XML of an MRCA prior, as used in the state section

Usage

```
mrca_prior_to_xml_state(mrca_prior, has_non_strict_clock_model = FALSE)
```

Arguments

mrca_prior a Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)
 has_non_strict_clock_model
 boolean to indicate that there is already at least one non-strict (i.e. relaxed log-normal) clock model

Value

the tree prior as XML text

Author(s)

Richèl J.C. Bilderbeek

mrca_prior_to_xml_taxonset

Creates the taxonset section in the prior section of the distribution section of a BEAST2 XML parameter file.

Description

Creates the taxonset section in the prior section of the distribution section of a BEAST2 XML parameter file.

Usage

```
mrca_prior_to_xml_taxonset(mrca_prior, taxa_names_with_ids = NULL)
```

Arguments

mrca_prior a Most Recent Common Ancestor prior, as returned by [create_mrca_prior](#)
 taxa_names_with_ids
 taxa names that already have received an ID. Causes the XML to idref these

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <taxonset id="all" spec="TaxonSet">
#   <taxon id="626029_aco" spec="Taxon"/>
#   <taxon id="630116_aco" spec="Taxon"/>
#   <taxon id="630210_aco" spec="Taxon"/>
#   <taxon id="B25702_aco" spec="Taxon"/>
#   <taxon id="61430_aco" spec="Taxon"/>
# </taxonset>
```

mrca_prior_to_xml_tracelog

Creates the MRCA prior's XML for the tracelog section

Description

Creates the MRCA prior's XML for the tracelog section

Usage

```
mrca_prior_to_xml_tracelog(clock_models, mrca_prior, tipdates_filename = NA)
```

Arguments

`clock_models` a list of one or more clock models, as returned by [create_clock_model](#)

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

`tipdates_filename` name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

all MRCA priors' tracelog section is created by [mrca_priors_to_xml_tracelog](#)

Examples

```
# <logger id="tracelog" ...>
#' # Here
# </logger>
```

no_taxa_to_xml_tree	<i>Creates the tree section (part of the state section) when there is no tip-dating</i>
---------------------	---

Description

Creates the tree section (part of the state section) when there is no tip-dating

Usage

```
no_taxa_to_xml_tree(id)
```

Arguments

id	an alignment's IDs. An ID can be extracted from its FASTA filename with get_alignment_ids_from_fasta_filenames)
----	--

Value

the random phylogeny as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml	<i>Converts a parameter to XML</i>
------------------	------------------------------------

Description

Converts a parameter to XML

Usage

```
parameter_to_xml(parameter)
```

Arguments

parameter a parameter, as created by `create_param`)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

xml <- parameter_to_xml(create_alpha_param(id = 1))
expect_equal(length(xml), 1)
expect_true(nchar(xml) > 1)
```

parameter_to_xml_alpha

Converts an alpha parameter to XML

Description

Converts an alpha parameter to XML

Usage

```
parameter_to_xml_alpha(parameter)
```

Arguments

parameter an alpha parameter, a numeric value. For advanced usage, use the structure as created by `create_alpha_param`)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_beta *Converts a beta parameter to XML*

Description

Converts a beta parameter to XML

Usage

```
parameter_to_xml_beta(parameter)
```

Arguments

parameter a beta parameter, a numeric value. For advanced usage, use the structure as created by [create_beta_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_clock_rate
Converts a clockRate parameter to XML

Description

Converts a clockRate parameter to XML

Usage

```
parameter_to_xml_clock_rate(parameter)
```

Arguments

parameter a clockRate parameter, a numeric value. For advanced usage, use the structure as created by [create_clock_rate_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_kappa_1

Converts a kappa 1 parameter to XML

Description

Converts a kappa 1 parameter to XML

Usage

parameter_to_xml_kappa_1(parameter)

Arguments

parameter a kappa 1 parameter, a numeric value. For advanced usage, use the structure as created by [create_kappa_1_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_kappa_2

Converts a kappa 2 parameter to XML

Description

Converts a kappa 2 parameter to XML

Usage

parameter_to_xml_kappa_2(parameter)

Arguments

parameter a kappa 2 parameter, a numeric value. For advanced usage, use the structure as created by [create_kappa_2_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_lambda

Converts a lambda parameter to XML

Description

Converts a lambda parameter to XML

Usage

parameter_to_xml_lambda(parameter)

Arguments

parameter a lambda parameter, a numeric value. For advanced usage, use the structure as created by [create_lambda_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_m

Converts a m parameter to XML

Description

Converts a m parameter to XML

Usage

parameter_to_xml_m(parameter)

Arguments

parameter a m parameter, a numeric value. For advanced usage, use the structure as created by [create_m_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_mean *Converts a mean parameter to XML*

Description

Converts a mean parameter to XML

Usage

```
parameter_to_xml_mean(parameter)
```

Arguments

parameter a mean parameter, a numeric value. For advanced usage, use the structure as created by [create_mean_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_mu *Converts a mu parameter to XML*

Description

Converts a mu parameter to XML

Usage

```
parameter_to_xml_mu(parameter)
```

Arguments

parameter a mu parameter, a numeric value. For advanced usage, use the structure as created by [create_mu_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_rate_ac

Converts a 'rate AC' parameter to XML

Description

Converts a 'rate AC' parameter to XML

Usage

```
parameter_to_xml_rate_ac(parameter, which_name = "state_node")
```

Arguments

parameter	a 'rate AC' parameter, a numeric value. For advanced usage, use the structure as created by create_rate_ac_param)
which_name	the name, can be state_node or rate_name

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_rate_ag

Converts a 'rate AG' parameter to XML

Description

Converts a 'rate AG' parameter to XML

Usage

```
parameter_to_xml_rate_ag(parameter, which_name = "state_node")
```

Arguments

parameter	a 'rate AG' parameter, a numeric value. For advanced usage, use the structure as created by create_rate_ag_param)
which_name	the name, can be state_node or rate_name

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_rate_at

Converts a 'rate AT' parameter to XML

Description

Converts a 'rate AT' parameter to XML

Usage

```
parameter_to_xml_rate_at(parameter, which_name = "state_node")
```

Arguments

parameter	a 'rate AT' parameter, a numeric value. For advanced usage, use the structure as created by create_rate_at_param)
which_name	the name, can be state_node or rate_name

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_rate_cg

Converts a 'rate CG' parameter to XML

Description

Converts a 'rate CG' parameter to XML

Usage

```
parameter_to_xml_rate_cg(parameter, which_name = "state_node")
```

Arguments

parameter	a 'rate CG' parameter, a numeric value. For advanced usage, use the structure as created by create_rate_cg_param)
which_name	the name, can be state_node or rate_name

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_rate_ct

Converts a 'rate CT' parameter to XML

Description

Converts a 'rate CT' parameter to XML

Usage

```
parameter_to_xml_rate_ct(parameter, which_name = "state_node")
```

Arguments

parameter	a 'rate CT' parameter, a numeric value. For advanced usage, use the structure as created by create_rate_ct_param)
which_name	the name, can be state_node or rate_name

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_rate_gt

Converts a 'rate GT' parameter to XML

Description

Converts a 'rate GT' parameter to XML

Usage

parameter_to_xml_rate_gt(parameter, which_name = "state_node")

Arguments

parameter	a 'rate GT' parameter, a numeric value. For advanced usage, use the structure as created by create_rate_gt_param)
which_name	the name, can be state_node or rate_name

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_s

Converts a s parameter to XML

Description

Converts a s parameter to XML

Usage

parameter_to_xml_s(parameter)

Arguments

parameter a s parameter, a numeric value. For advanced usage, use the structure as created by [create_s_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_scale

Converts a scale parameter to XML

Description

Converts a scale parameter to XML

Usage

parameter_to_xml_scale(parameter)

Arguments

parameter a scale parameter, a numeric value. For advanced usage, use the structure as created by [create_scale_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_sigma

Converts a sigma parameter to XML

Description

Converts a sigma parameter to XML

Usage

```
parameter_to_xml_sigma(parameter)
```

Arguments

parameter a sigma parameter, a numeric value. For advanced usage, use the structure as created by [create_sigma_param](#))

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

phylo_to_xml_state

Creates the XML of a phylogeny, as used in the state section

Description

Creates the XML of a phylogeny, as used in the state section

Usage

```
phylo_to_xml_state(id, tipdates_filename = NA)
```

Arguments

id the ID of the alignment

tipdates_filename

name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

the random phylogeny as XML text

Author(s)

Richèl J.C. Bilderbeek

remove_empty_lines *Remove all lines that are only whitespace*

Description

Remove all lines that are only whitespace

Usage

```
remove_empty_lines(lines, trim = FALSE)
```

Arguments

lines	character vector with text
trim	FALSE if indentation must be preserved, TRUE will remove all surrounding whitespace

Value

the lines with text

Author(s)

Richèl J.C. Bilderbeek

remove_multiline *Remove consecutive lines*

Description

Remove consecutive lines

Usage

```
remove_multiline(text, lines_to_remove)
```

Arguments

text	lines of characters
lines_to_remove	lines of character that need to be removed from text

Value

lines of text

Author(s)

Richèl J.C. Bilderbeek

rename_inference_model_filenames

Rename the filenames in an inference model

Description

Rename the filenames in an inference model

Usage

```
rename_inference_model_filenames(inference_model, rename_fun)
```

Arguments

inference_model

a Bayesian phylogenetic inference model. An inference model is the complete model setup in which a site model, clock model, tree prior and more are specified. Use [create_inference_model](#) to create an inference model. Use [check_inference_model](#) to check if an inference model is valid. Use [rename_inference_model_filenames](#) to rename the files in an inference model.

rename_fun

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

- [get_remove_dir_fun](#) get a function that removes the directory paths from the filenames, in effect turning these into local files
- [get_replace_dir_fun](#) get a function that replaces the directory paths from the filenames
- [get_remove_hex_fun](#) get a function that removes the hex string from filenames. For example, `tracelog_82c1a522040.log` becomes `tracelog.log`

Value

an inference model with the renamed filenames

Examples

```

library(testthat)

inference_model <- create_inference_model()
inference_model$mcmc$tracelog$filename <- "trace.log"
inference_model$mcmc$screenlog$filename <- "screen.log"
inference_model$mcmc$treelog$filename <- "tree.log"
inference_model$tipdates_filename <- "tipdates.csv"

# Nah, put the files in a folder
inference_model <- rename_inference_model_filenames(
  inference_model = inference_model,
  rename_fun = get_replace_dir_fun("/home/john")
)

expect_equal(inference_model$mcmc$tracelog$filename, "/home/john/trace.log")
expect_equal(
  inference_model$mcmc$screenlog$filename, "/home/john/screen.log"
)
expect_equal(inference_model$mcmc$treelog$filename, "/home/john/tree.log")
expect_equal(inference_model$tipdates_filename, "/home/john/tipdates.csv")

# Nah, put the files in another folder
inference_model <- rename_inference_model_filenames(
  inference_model = inference_model,
  rename_fun = get_replace_dir_fun("/home/doi")
)

expect_equal(inference_model$mcmc$tracelog$filename, "/home/doi/trace.log")
expect_equal(inference_model$mcmc$screenlog$filename, "/home/doi/screen.log")
expect_equal(inference_model$mcmc$treelog$filename, "/home/doi/tree.log")
expect_equal(inference_model$tipdates_filename, "/home/doi/tipdates.csv")

# Nah, store the files locally
inference_model <- rename_inference_model_filenames(
  inference_model = inference_model,
  rename_fun = get_remove_dir_fun()
)

expect_equal(inference_model$mcmc$tracelog$filename, "trace.log")
expect_equal(inference_model$mcmc$screenlog$filename, "screen.log")
expect_equal(inference_model$mcmc$treelog$filename, "tree.log")
expect_equal(inference_model$tipdates_filename, "tipdates.csv")

```

rename_mcmc_filenames *Rename the filenames within an MCMC*

Description

Rename the filenames within an MCMC

Usage

```
rename_mcmc_filenames(mcmc, rename_fun)
```

Arguments

- | | |
|------------|--|
| mcmc | one MCMC. Use create_mcmc to create an MCMC. Use create_ns_mcmc to create an MCMC for a Nested Sampling run. Use check_mcmc to check if an MCMC is valid. Use rename_mcmc_filenames to rename the filenames in an MCMC. |
| 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 <code>NA</code> . The function should return one filename (when passed one filename) or one <code>NA</code> (when passed one <code>NA</code>). Example rename functions are: <ul style="list-style-type: none"> • 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, <code>tracelog_82c1a522040.log</code> becomes <code>tracelog.log</code> |

Examples

```
library(testthat)

# Create an MCMC with local filenames
mcmc <- create_mcmc()
mcmc$tracelog$filename <- "trace.log"
mcmc$screenlog$filename <- "screen.log"
mcmc$treelog$filename <- "tree.log"

# Nah, files should be put in '/home/john' folder
mcmc <- rename_mcmc_filenames(
  mcmc = mcmc,
  rename_fun = get_replace_dir_fun("/home/john")
)

expect_equal(mcmc$tracelog$filename, "/home/john/trace.log")
expect_equal(mcmc$screenlog$filename, "/home/john/screen.log")
expect_equal(mcmc$treelog$filename, "/home/john/tree.log")

# Nah, files should be put in '/home/doe' folder instead
mcmc <- rename_mcmc_filenames(
  mcmc = mcmc,
  rename_fun = get_replace_dir_fun("/home/doe")
)

expect_equal(mcmc$tracelog$filename, "/home/doe/trace.log")
expect_equal(mcmc$screenlog$filename, "/home/doe/screen.log")
expect_equal(mcmc$treelog$filename, "/home/doe/tree.log")
```

```
# Nah, files should be put in local folder instead
mcmc <- rename_mcmc_filenames(
  mcmc = mcmc,
  rename_fun = get_remove_dir_fun()
)

expect_equal(mcmc$tracelog$filename, "trace.log")
expect_equal(mcmc$screenlog$filename, "screen.log")
expect_equal(mcmc$treelog$filename, "tree.log")
```

rln_clock_model_to_xml_mean_rate_prior

Used by [clock_models_to_xml_prior_distr](#)

Description

Used by [clock_models_to_xml_prior_distr](#)

Usage

```
rln_clock_model_to_xml_mean_rate_prior(rln_clock_model)
```

Arguments

rln_clock_model

a Relaxed Log-Normal clock model, as returned by [create_rln_clock_model](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

`rnd_phylo_to_xml_init` *Creates the XML of a random phylogeny, as used in the init section*

Description

Creates the XML of a random phylogeny, as used in the init section

Usage

```
rnd_phylo_to_xml_init(id)
```

Arguments

id an alignment's IDs. An ID can be extracted from its FASTA filename with [get_alignment_ids_from_fasta_filenames](#))

Value

the phylogeny as XML text

Author(s)

Richèl J.C. Bilderbeek

site_models_to_xml_operators

Write the XML operators section from the site models.

Description

Write the XML operators section from the site models.

Usage

```
site_models_to_xml_operators(site_models)
```

Arguments

site_models one or more site models, as returned by [create_site_model](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

`site_models_to_xml_prior_distr`*Represent the site models as XML*

Description

Represent the site models as XML

Usage

```
site_models_to_xml_prior_distr(site_models)
```

Arguments

`site_models` one or more site models, as returned by [create_site_model](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
#   <distribution id="likelihood" ...>
#   </distribution>
# </distribution>
```

`site_models_to_xml_state`*Converts one or more clock models to the state section of the XML as text*

Description

Converts one or more clock models to the state section of the XML as text

Usage

```
site_models_to_xml_state(site_models)
```

Arguments

site_models one or more site models, as returned by [create_site_model](#)

Value

lines of XML text, without indentation nor state tags

Author(s)

Richèl J.C. Bilderbeek

site_models_to_xml_tracelog

Creates the site models' XML for the tracelog section

Description

Creates the site models' XML for the tracelog section

Usage

```
site_models_to_xml_tracelog(site_models)
```

Arguments

site_models one or more site models, as returned by [create_site_model](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

the complete tracelog section is created by [create_tracelog_xml](#)

Examples

```
# <logger id="tracelog" ...>
#' # Here
# </logger>
```

`site_model_to_xml_lh_distr`*Converts a site model to XML, used in the siteModel section*

Description

Converts a site model to XML, used in the siteModel section

Usage

```
site_model_to_xml_lh_distr(site_model)
```

Arguments

`site_model` a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     </distribution>
#   <distribution id="likelihood" ...>
#     HERE, where the ID of the distribution is 'likelihood'
#   </distribution>
# </distribution>
```

`site_model_to_xml_operators`*Converts a site model to XML, used in the operators section*

Description

Converts a site model to XML, used in the operators section

Usage

```
site_model_to_xml_operators(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

site_model_to_xml_prior_distr

Converts a site model to XML, used in the prior section

Description

Converts a site model to XML, used in the prior section

Usage

```
site_model_to_xml_prior_distr(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

`site_model_to_xml_state`*Converts a site model to XML, used in the state section*

Description

Converts a site model to XML, used in the state section

Usage

```
site_model_to_xml_state(site_model)
```

Arguments

`site_model` a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

`site_model_to_xml_subst_model`*Converts a site model to XML, used in the substModel section*

Description

Converts a site model to XML, used in the substModel section

Usage

```
site_model_to_xml_subst_model(site_model)
```

Arguments

`site_model` a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

site_model_to_xml_tracelog

Creates the site model's XML for the tracelog section

Description

Creates the site model's XML for the tracelog section

Usage

```
site_model_to_xml_tracelog(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

all site models' tracelog section is created by [site_models_to_xml_tracelog](#)

Examples

```
# <logger id="tracelog" ...>
#' # Here
# </logger>
```

taxa_to_xml_tree

Creates the tree section (part of the state section) of a phylogeny and/or taxa

Description

Creates the tree section (part of the state section) of a phylogeny and/or taxa

Usage

```
taxa_to_xml_tree(id, tipdates_filename = NA)
```

Arguments

`id` an alignment's IDs. An ID can be extracted from its FASTA filename with [get_alignment_ids_from_fasta_filenames](#))

`tipdates_filename` name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

the random phylogeny as XML text

Author(s)

Richèl J.C. Bilderbeek

tipdate_taxa_to_xml_tree

Creates the tree section (part of the state section) when there is tip-dating

Description

Creates the tree section (part of the state section) when there is tip-dating

Usage

```
tipdate_taxa_to_xml_tree(id, tipdates_filename)
```

Arguments

`id` an alignment's IDs. An ID can be extracted from its FASTA filename with [get_alignment_ids_from_fasta_filenames](#))

`tipdates_filename` name of the file containing the tip dates. This file is assumed to have two columns, separated by a tab. The first column contains the taxa names, the second column contains the date.

Value

the random phylogeny as XML text

Author(s)

Richèl J.C. Bilderbeek

tn93_site_model_to_xml_subst_model

Converts a TN93 site model to XML, used in the substModel section

Description

Converts a TN93 site model to XML, used in the substModel section

Usage

```
tn93_site_model_to_xml_subst_model(site_model)
```

Arguments

site_model a site model, as returned by [create_site_model](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

tree_models_to_xml_tracelog

Creates the tree models' XML for the tracelog section

Description

Creates the tree models' XML for the tracelog section

Usage

```
tree_models_to_xml_tracelog(site_models)
```

Arguments

site_models one or more site models, as returned by [create_site_model](#)

Value

lines of XML text

Note

use site_models just because it contains all IDs

Author(s)

Richèl J.C. Bilderbeek

See Also

the complete tracelog section is created by [create_tracelog_xml](#)

Examples

```
# <logger id="tracelog" ...>
#' # Here
# </logger>
```

tree_priors_to_xml_operators

Creates the XML of a list of one or more tree priors, as used in the operators section

Description

Creates the XML of a list of one or more tree priors, as used in the operators section

Usage

```
tree_priors_to_xml_operators(
  tree_priors,
  fixed_crown_ages = rep(FALSE, times = length(tree_priors))
)
```

Arguments

tree_priors one or more tree priors, as returned by [create_tree_prior](#)

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.

Value

the tree priors as XML text

Author(s)

Richèl J.C. Bilderbeek

tree_priors_to_xml_prior_distr

Creates the distribution section in the prior section of the distribution section of a BEAST2 XML parameter file.

Description

These lines start with '<distribution id='

Usage

```
tree_priors_to_xml_prior_distr(tree_priors)
```

Arguments

tree_priors one or more tree priors, as returned by [create_tree_prior](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
#   <distribution id="likelihood" ...>
#   </distribution>
# </distribution>
```

tree_priors_to_xml_state

Converts one or more tree priors to the state section of the XML as text

Description

Converts one or more tree priors to the state section of the XML as text

Usage

```
tree_priors_to_xml_state(tree_priors)
```


Arguments

tree_priors one or more tree priors, as returned by [create_tree_prior](#)

Value

lines of XML text, without indentation nor state tags

Author(s)

Richèl J.C. Bilderbeek

tree_priors_to_xml_tracelog

Creates the tree priors' XML for the tracelog section

Description

Creates the tree priors' XML for the tracelog section

Usage

```
tree_priors_to_xml_tracelog(tree_priors)
```

Arguments

tree_priors one or more tree priors, as returned by [create_tree_prior](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

the complete tracelog section is created by [create_tracelog_xml](#)

Examples

```
# <logger id="tracelog" ...>
#' # Here
# </logger>
```

tree_prior_to_xml_operators

Creates the XML of a tree prior, as used in the operators section

Description

Creates the XML of a tree prior, as used in the operators section

Usage

```
tree_prior_to_xml_operators(tree_prior, fixed_crown_age = FALSE)
```

Arguments

tree_prior a tree priors, as returned by [create_tree_prior](#)

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.

Value

the tree prior as XML text

Author(s)

Richèl J.C. Bilderbeek

tree_prior_to_xml_prior_distr

Creates the distribution section in the prior section of the distribution section of a BEAST2 XML parameter file.

Description

These lines start with '<distribution id='

Usage

```
tree_prior_to_xml_prior_distr(tree_prior)
```

Arguments

tree_prior a tree priors, as returned by [create_tree_prior](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">
#   <distribution id="prior" spec="util.CompoundDistribution">
#     HERE, where the ID of the distribution is 'prior'
#   </distribution>
#   <distribution id="likelihood" ...>
#     </distribution>
# </distribution>
```

tree_prior_to_xml_state

Creates the XML of a tree prior, as used in the state section

Description

Creates the XML of a tree prior, as used in the state section

Usage

```
tree_prior_to_xml_state(tree_prior)
```

Arguments

tree_prior a tree priors, as returned by [create_tree_prior](#)

Value

the tree prior as XML text

Author(s)

Richèl J.C. Bilderbeek

```
tree_prior_to_xml_tracelog
```

Creates the tree prior's XML for the tracelog section

Description

Creates the tree prior's XML for the tracelog section

Usage

```
tree_prior_to_xml_tracelog(tree_prior)
```

Arguments

tree_prior a tree priors, as returned by [create_tree_prior](#)

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

See Also

all tree priors' tracelog section is created by [tree_priors_to_xml_tracelog](#)

Examples

```
# <logger id="tracelog" ...>
#   # Here
# </logger>
```

```
yule_tree_prior_to_xml_prior_distr
```

Creates the prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Yule tree prior

Description

Creates the prior section in the prior section of the prior section of the distribution section of a BEAST2 XML parameter file for a Yule tree prior

Usage

```
yule_tree_prior_to_xml_prior_distr(yule_tree_prior)
```

Arguments

```
yule_tree_prior  
    a Yule tree_prior, as created by create\_yule\_tree\_prior
```

Author(s)

Richèl J.C. Bilderbeek

Examples

```
# <distribution id="posterior" spec="util.CompoundDistribution">  
#   <distribution id="prior" spec="util.CompoundDistribution">  
#     HERE, where the ID of the distribution is 'prior'  
#   </distribution>  
#   <distribution id="likelihood" ...>  
#   </distribution>  
# </distribution>
```

Index

are_clock_models, 11
are_equal_mcmcs, 12
are_equal_screenlogs, 13
are_equal_tracelogs, 14
are_equal_treelogs, 15
are_equal_xml_files, 16
are_equal_xml_lines, 16
are_equivalent_xml_files, 16, 17
are_equivalent_xml_lines, 18
are_equivalent_xml_lines_all, 18
are_equivalent_xml_lines_loggers, 19
are_equivalent_xml_lines_operators, 20
are_equivalent_xml_lines_section, 20
are_fasta_filenames, 21, 187
are_ids, 22, 256
are_init_clock_models, 22
are_init_mrca_priors, 23
are_init_site_models, 23
are_init_tree_priors, 24
are_mrca_align_ids_in_fasta, 24
are_mrca_priors, 25
are_mrca_taxon_names_in_fasta, 25
are_rln_clock_models, 26
are_site_models, 26
are_tree_priors, 27

bd_tree_prior_to_xml_prior_distr, 28
beautier, 29, 203

cbs_tree_prior_to_xml_prior_distr, 29
ccp_tree_prior_to_xml_prior_distr, 30
cep_tree_prior_to_xml_prior_distr, 31
check_alignment_id, 32, 185
check_beauti_options, 32
check_clock_model, 33
check_clock_models, 34
check_file_and_model_agree, 35
check_file_exists, 35
check_gamma_site_model, 36
check_gamma_site_model_names, 37

check_gtr_site_model, 38
check_gtr_site_model_names, 38
check_inference_model, 35, 39, 39, 79, 86,
88, 91, 118, 140, 159, 161, 169, 198,
215, 226, 230, 338
check_inference_models, 40
check_is_monophyletic, 41
check_log_mode, 41
check_log_sort, 41
check_mcmc, 42, 42, 43, 44, 47, 77, 85, 111,
120, 150, 153, 170, 199, 249, 316,
317, 340
check_mcmc_list_element_names, 43
check_mcmc_nested_sampling
(check_ns_mcmc), 46
check_mcmc_values, 43
check_mrca_prior, 44, 46
check_mrca_prior_name, 45
check_mrca_prior_names, 45
check_mrca_prior_taxa_names, 46
check_nested_sampling_mcmc
(check_ns_mcmc), 46
check_ns_mcmc, 46, 127
check_param, 47
check_param_names, 48
check_param_types, 49
check_phylogeny, 49, 295
check_rename_fun, 50, 50, 170, 202, 338, 340
check_rln_clock_model, 51
check_screenlog, 51
check_screenlog_names, 52
check_screenlog_values, 52
check_site_model, 53
check_site_model_names, 55
check_site_model_types, 56
check_site_models, 54
check_store_every, 56
check_strict_clock_model, 57
check_tn93_site_model, 58

- check_tn93_site_model_names, 58
- check_tracelog, 59
- check_tracelog_names, 59
- check_tracelog_values, 60
- check_tree_prior, 62
- check_tree_priors, 63
- check_treelog, 60
- check_treelog_names, 61
- check_treelog_values, 61
- clock_model_to_xml_lh_distr, 67
- clock_model_to_xml_operators, 68
- clock_model_to_xml_prior_distr, 69
- clock_model_to_xml_state, 70
- clock_model_to_xml_tracelog, 70
- clock_model_to_xml_treelogger, 71
- clock_models_to_xml_operators, 64
- clock_models_to_xml_prior_distr, 64, 341
- clock_models_to_xml_state, 65
- clock_models_to_xml_tracelog, 66, 71
- compare_lines, 72
- count_trailing_spaces, 73
- create_alpha_param, 73, 93, 105, 112, 129, 326
- create_bd_tree_prior, 28, 75, 161–163, 168, 219, 239
- create_beast2_beast_xml, 76
- create_beast2_input, 76, 81
- create_beast2_input_beast, 78, 88
- create_beast2_input_data, 79, 79
- create_beast2_input_data_sequences, 80
- create_beast2_input_distr, 81, 91
- create_beast2_input_distr_lh, 82
- create_beast2_input_distr_prior, 83
- create_beast2_input_file, 78, 84, 87
- create_beast2_input_file_from_model, 85, 86, 88
- create_beast2_input_from_model, 78, 79, 87, 87
- create_beast2_input_init, 88, 91
- create_beast2_input_map, 79, 89
- create_beast2_input_operators, 89, 91
- create_beast2_input_run, 79, 90
- create_beast2_input_state, 91, 91
- create_beauti_options, 32, 33, 77, 79, 80, 85, 92, 111, 150, 168, 240
- create_beta_distr, 74, 93, 94, 104, 173, 220, 241, 258
- create_beta_param, 93, 94, 105, 112, 129, 327
- create_cbs_tree_prior, 30, 95, 95, 161–163, 168, 169, 243
- create_ccp_tree_prior, 30, 96, 161–163, 168, 220, 244
- create_cep_tree_prior, 31, 97, 161–163, 168, 221, 245
- create_clock_model, 26, 33, 34, 51, 57, 64–71, 77, 81–83, 85, 87, 90, 92, 98, 99, 101, 111, 150, 153, 168, 180, 189, 197, 221, 246, 260, 304, 309, 320, 324
- create_clock_model_from_name, 101
- create_clock_model_rln
(create_rln_clock_model), 137
- create_clock_model_strict
(create_strict_clock_model), 146
- create_clock_models, 99, 100, 190
- create_clock_models_from_names, 100
- create_clock_rate_param, 102, 129, 146, 327
- create_data_xml, 103
- create_distr, 75, 94, 96, 97, 103, 105–109, 113, 117, 119, 122, 126, 128, 130, 137, 146, 164, 165, 172, 192, 222, 261
- create_distr_beta (create_beta_distr), 93
- create_distr_exp (create_exp_distr), 104
- create_distr_gamma
(create_gamma_distr), 105
- create_distr_inv_gamma
(create_inv_gamma_distr), 112
- create_distr_laplace
(create_laplace_distr), 116
- create_distr_log_normal
(create_log_normal_distr), 118
- create_distr_normal
(create_normal_distr), 125
- create_distr_one_div_x
(create_one_div_x_distr), 128
- create_distr_poisson
(create_poisson_distr), 130
- create_distr_uniform
(create_uniform_distr), 163
- create_exp_distr, 104, 104, 121, 173, 223,

- 251, 261
 create_gamma_distr, 74, 94, 104, 105, 174, 223, 253
 create_gamma_site_model, 36, 37, 106, 107, 108, 110, 113, 142, 157, 169, 184, 195, 196, 224, 262
 create_gtr_site_model, 38, 39, 108, 142, 143, 169, 195, 225, 252, 254, 263
 create_hky_site_model, 110, 142, 143, 169, 195, 226, 252, 255, 263
 create_inference_model, 35, 39, 40, 79, 86, 88, 91, 111, 112, 118, 140, 151, 159, 161, 169, 198, 215, 226, 230, 338
 create_inv_gamma_distr, 74, 94, 104, 112, 174, 227, 264, 273
 create_jc69_site_model, 113, 142, 143, 170, 227, 265
 create_kappa_1_param, 114, 129, 157, 276, 328
 create_kappa_2_param, 115, 129, 157, 277, 328
 create_lambda_param, 115, 129, 130, 278, 329
 create_laplace_distr, 104, 116, 123, 139, 175, 228, 266, 279
 create_log_normal_distr, 104, 110, 118, 124, 148, 157, 175, 229, 266, 280
 create_loggers_xml, 91, 117
 create_m_param, 118, 124, 129, 329
 create_mcmc, 12, 42–44, 47, 77, 85, 87, 111, 119, 127, 150, 152, 153, 170, 199, 249, 281, 316, 317, 340
 create_mcmc_nested_sampling (create_ns_mcmc), 126
 create_mean_param, 104, 120, 125, 129, 283, 330
 create_mrca_prior, 24, 25, 41, 44, 46, 64–70, 77, 81–83, 85, 90, 92, 111, 121, 122, 150, 170, 215, 230, 284, 285, 318–324
 create_mu_param, 116, 123, 129, 287, 330
 create_normal_distr, 104, 121, 125, 141, 176, 231, 267, 288
 create_ns_mcmc, 42–44, 47, 77, 85, 111, 120, 126, 150, 153, 154, 170, 199, 249, 282, 316, 317, 340
 create_one_div_x_distr, 104, 128, 176, 231, 268, 290
 create_param, 47–49, 74, 95, 102, 116, 121, 123, 124, 129, 131–136, 139, 141, 148, 170, 232, 268, 293, 326
 create_param_alpha (create_alpha_param), 73
 create_param_beta (create_beta_param), 94
 create_param_clock_rate (create_clock_rate_param), 102
 create_param_kappa_1 (create_kappa_1_param), 114
 create_param_kappa_2 (create_kappa_2_param), 115
 create_param_lambda (create_lambda_param), 115
 create_param_m (create_m_param), 124
 create_param_mean (create_mean_param), 120
 create_param_mu (create_mu_param), 123
 create_param_rate_ac (create_rate_ac_param), 131
 create_param_rate_ag (create_rate_ag_param), 132
 create_param_rate_at (create_rate_at_param), 133
 create_param_rate_cg (create_rate_cg_param), 134
 create_param_rate_ct (create_rate_ct_param), 135
 create_param_rate_gt (create_rate_gt_param), 136
 create_param_s (create_s_param), 147
 create_param_scale (create_scale_param), 138
 create_param_sigma (create_sigma_param), 141
 create_poisson_distr, 104, 116, 130, 177, 232, 269, 296
 create_rate_ac_param, 109, 129, 131, 297, 331
 create_rate_ag_param, 109, 129, 132, 298, 332
 create_rate_at_param, 109, 129, 133, 299, 332
 create_rate_cg_param, 109, 129, 134, 301, 333
 create_rate_ct_param, 109, 129, 135, 302, 333

- create_rate_gt_param, [109](#), [129](#), [136](#), [303](#), [334](#)
- create_rln_clock_model, [98](#), [99](#), [137](#), [171](#), [233](#), [269](#), [304](#), [341](#)
- create_s_param, [118](#), [129](#), [147](#), [335](#)
- create_scale_param, [116](#), [129](#), [138](#), [335](#)
- create_screenlog, [13](#), [52](#), [53](#), [120](#), [127](#), [139](#), [148](#), [152](#), [153](#), [171](#)
- create_screenlog_xml, [118](#), [140](#)
- create_sigma_param, [125](#), [129](#), [141](#), [336](#)
- create_site_model, [27](#), [53](#), [55](#), [56](#), [77](#), [81–83](#), [85](#), [87](#), [90](#), [92](#), [111](#), [142](#), [144](#), [145](#), [150](#), [153](#), [171](#), [183](#), [204–207](#), [215](#), [218](#), [234](#), [270](#), [307](#), [315](#), [342–348](#), [350](#)
- create_site_model_from_name, [145](#)
- create_site_model_gtr
(create_gtr_site_model), [108](#)
- create_site_model_hky
(create_hky_site_model), [110](#)
- create_site_model_jc69
(create_jc69_site_model), [113](#)
- create_site_model_tn93
(create_tn93_site_model), [156](#)
- create_site_models, [143](#), [205](#)
- create_site_models_from_names, [144](#)
- create_strict_clock_model, [98](#), [99](#), [102](#), [146](#), [171](#), [234](#), [270](#), [308](#)
- create_temp_screenlog_filename, [148](#)
- create_temp_tracelog_filename, [149](#)
- create_temp_treelog_filename, [149](#)
- create_test_inference_model, [150](#)
- create_test_mcmc, [151](#)
- create_test_ns_inference_model, [152](#)
- create_test_ns_mcmc, [127](#), [153](#)
- create_test_screenlog, [154](#)
- create_test_tracelog, [155](#)
- create_test_treelog, [156](#)
- create_tn93_site_model, [58](#), [59](#), [142](#), [143](#), [156](#), [171](#), [195](#), [235](#), [252](#), [271](#), [310](#)
- create_tracelog, [14](#), [59](#), [60](#), [120](#), [127](#), [149](#), [152](#), [153](#), [158](#), [171](#)
- create_tracelog_xml, [67](#), [118](#), [158](#), [320](#), [344](#), [351](#), [353](#)
- create_trait_set_string, [159](#)
- create_tree_prior, [62](#), [63](#), [77](#), [81](#), [83](#), [85](#), [87](#), [90](#), [92](#), [111](#), [150](#), [153](#), [161](#), [171](#), [201](#), [208](#), [209](#), [211](#), [212](#), [236](#), [312](#), [351–356](#)
- create_tree_prior_bd
(create_bd_tree_prior), [75](#)
- create_tree_prior_cbs
(create_cbs_tree_prior), [95](#)
- create_tree_prior_ccp
(create_ccp_tree_prior), [96](#)
- create_tree_prior_cep
(create_cep_tree_prior), [97](#)
- create_tree_prior_yule
(create_yule_tree_prior), [165](#)
- create_tree_priors, [162](#), [210](#)
- create_treelog, [15](#), [60–62](#), [120](#), [127](#), [149](#), [152](#), [153](#), [160](#), [171](#)
- create_treelog_xml, [118](#), [160](#)
- create_uniform_distr, [104](#), [163](#), [177](#), [236](#), [272](#), [313](#)
- create_xml_declaration, [88](#), [164](#)
- create_yule_tree_prior, [27](#), [161–163](#), [165](#), [171](#), [237](#), [315](#), [357](#)
- default_parameters_doc, [166](#)
- default_params_doc, [166](#)
- distr_to_xml, [172](#)
- distr_to_xml_beta, [173](#)
- distr_to_xml_exp, [173](#)
- distr_to_xml_gamma, [174](#)
- distr_to_xml_inv_gamma, [174](#)
- distr_to_xml_laplace, [175](#)
- distr_to_xml_log_normal, [175](#)
- distr_to_xml_normal, [176](#)
- distr_to_xml_one_div_x, [176](#)
- distr_to_xml_poisson, [177](#)
- distr_to_xml_uniform, [177](#)
- extract_xml_loggers_from_lines, [178](#)
- extract_xml_operators_from_lines, [178](#)
- extract_xml_section_from_lines, [179](#)
- FALSE, [240](#), [268](#)
- fasta_file_to_sequences, [179](#)
- find_clock_model, [180](#)
- find_first_regex_line, [180](#)
- find_first_xml_opening_tag_line, [181](#)
- find_last_regex_line, [181](#)
- find_last_xml_closing_tag_line, [182](#)
- freq_equilibrium_to_xml, [182](#)
- gamma_site_model_to_xml_prior_distr, [183](#)

- gamma_site_model_to_xml_state, 184
- gamma_site_models_to_xml_prior_distr, 183
- get_alignment_id, 32, 75, 96–98, 108, 110, 113, 122, 142, 155, 157, 158, 165, 168, 184
- get_alignment_ids, 185, 187
- get_alignment_ids_from_fasta_filenames, 88, 95, 103, 137, 146, 169, 186, 186, 234, 236, 325, 342, 349
- get_beautier_path, 187, 188
- get_beautier_paths, 187, 188
- get_clock_model_name, 189
- get_clock_model_names, 100, 101, 168, 190
- get_clock_models_ids, 189
- get_crown_age, 191
- get_distr_n_params, 192
- get_distr_names, 191
- get_fasta_filename, 24, 25, 35, 77, 79, 85–87, 91, 118, 159, 169, 170, 185, 186, 193, 221, 226, 230, 284, 285
- get_file_base_sans_ext, 194
- get_freq_equilibrium_names, 194
- get_gamma_site_model_n_distrs, 195
- get_gamma_site_model_n_params, 196
- get_has_non_strict_clock_model, 197
- get_inference_model_filenames, 197
- get_log_modes, 140, 155, 156, 158, 160, 170, 198
- get_log_sorts, 140, 155, 156, 158, 160, 171, 199
- get_mcmc_filenames, 199
- get_n_taxa, 200
- get_operator_id_pre, 201
- get_param_names, 201
- get_remove_dir_fun, 50, 170, 202, 338, 340
- get_remove_hex_fun, 50, 170, 203, 338, 340
- get_replace_dir_fun, 50, 170, 203, 338, 340
- get_site_model_n_distrs, 206
- get_site_model_n_params, 207
- get_site_model_names, 144, 145, 171, 205
- get_site_models_n_distrs, 204
- get_site_models_n_params, 204
- get_taxa_names, 46, 122, 171, 208
- get_tree_prior_n_distrs, 211
- get_tree_prior_n_params, 212
- get_tree_prior_names, 171, 210
- get_tree_priors_n_distrs, 208
- get_tree_priors_n_params, 209
- get_xml_closing_tag, 213
- get_xml_opening_tag, 214
- gtr_site_model_to_xml_subst_model, 214
- has_mrca_prior, 215
- has_xml_closing_tag, 216
- has_xml_opening_tag, 217
- has_xml_short_closing_tag, 217
- hky_site_model_to_xml_subst_model, 218
- indent, 218
- init_bd_tree_prior, 219
- init_beta_distr, 220
- init_ccp_tree_prior, 220
- init_cep_tree_prior, 221
- init_clock_models, 221
- init_distr, 222
- init_exp_distr, 222
- init_gamma_distr, 223
- init_gamma_site_model, 224
- init_gtr_site_model, 225
- init_hky_site_model, 225
- init_inference_model, 226
- init_inv_gamma_distr, 227
- init_jc69_site_model, 227
- init_laplace_distr, 228
- init_log_normal_distr, 229
- init_mrca_prior, 229
- init_mrca_priors, 230
- init_normal_distr, 231
- init_one_div_x_distr, 231
- init_param, 232
- init_poisson_distr, 232
- init_rln_clock_model, 233
- init_site_models, 234
- init_strict_clock_model, 234
- init_tn93_site_model, 235
- init_tree_priors, 236
- init_uniform_distr, 236
- init_yule_tree_prior, 237
- interspace, 237
- is_alpha_param, 238
- is_bd_tree_prior, 239
- is_beauti_options, 240
- is_beta_distr, 241, 250
- is_beta_param, 242
- is_cbs_tree_prior, 243

- is_ccp_tree_prior, 244
- is_cep_tree_prior, 245
- is_clock_model, 246
- is_clock_model_name, 247
- is_clock_rate_param, 247
- is_default_mcmc, 248
- is_distr, 241, 249, 251, 253, 274, 280, 289, 290, 296, 313
- is_distr_name, 250
- is_exp_distr, 250, 251
- is_freq_equilibrium_name, 252
- is_gamma_distr, 250, 253
- is_gamma_site_model, 254
- is_gtr_site_model, 254
- is_hky_site_model, 255
- is_id, 22, 256
- is_in_patterns, 274
- is_inference_model, 257
- is_init_bd_tree_prior, 257
- is_init_beta_distr, 258
- is_init_cbs_tree_prior, 258
- is_init_ccp_tree_prior, 259
- is_init_cep_tree_prior, 259
- is_init_clock_model, 260
- is_init_distr, 261
- is_init_exp_distr, 261
- is_init_gamma_distr, 262
- is_init_gamma_site_model, 262
- is_init_gtr_site_model, 263
- is_init_hky_site_model, 263
- is_init_inv_gamma_distr, 264
- is_init_jc69_site_model, 265
- is_init_laplace_distr, 266
- is_init_log_normal_distr, 266
- is_init_mrca_prior, 267
- is_init_normal_distr, 267
- is_init_one_div_x_distr, 268
- is_init_param, 268
- is_init_poisson_distr, 269
- is_init_rln_clock_model, 269
- is_init_site_model, 270
- is_init_strict_clock_model, 270
- is_init_tn93_site_model, 271
- is_init_tree_prior, 271
- is_init_uniform_distr, 272
- is_init_yule_tree_prior, 273
- is_inv_gamma_distr, 250, 273
- is_jc69_site_model, 275
- is_kappa_1_param, 276
- is_kappa_2_param, 277
- is_lambda_param, 278
- is_laplace_distr, 250, 279
- is_log_normal_distr, 250, 280
- is_m_param, 287
- is_mcmc, 281
- is_mcmc_nested_sampling, 282
- is_mean_param, 283
- is_mrca_align_id_in_fasta, 284
- is_mrca_align_ids_in_fastas, 284
- is_mrca_prior, 285
- is_mrca_prior_with_distr, 286
- is_mu_param, 286
- is_nested_sampling_mcmc
(is_mcmc_nested_sampling), 282
- is_normal_distr, 250, 288
- is_one_bool, 289
- is_one_div_x_distr, 250, 290
- is_one_double, 291
- is_one_int, 291
- is_one_na, 292
- is_param, 293
- is_param_name, 294
- is_phylo, 295
- is_poisson_distr, 250, 296
- is_rate_ac_param, 297
- is_rate_ag_param, 298
- is_rate_at_param, 299
- is_rate_cg_param, 300
- is_rate_ct_param, 301
- is_rate_gt_param, 303
- is_rln_clock_model, 304
- is_s_param, 309
- is_scale_param, 305
- is_sigma_param, 306
- is_site_model, 307
- is_site_model_name, 308
- is_strict_clock_model, 308
- is_tn93_site_model, 310
- is_tree_prior, 311
- is_tree_prior_name, 312
- is_uniform_distr, 250, 313
- is_xml, 314
- is_yule_tree_prior, 314
- jc69_site_model_to_xml_subst_model, 315

- mcmc_to_xml_run, 316
- mcmc_to_xml_run_default, 316
- mcmc_to_xml_run_nested_sampling, 317
- mrca_prior_to_xml_lh_distr, 321
- mrca_prior_to_xml_prior_distr, 322
- mrca_prior_to_xml_state, 323
- mrca_prior_to_xml_taxonset, 323
- mrca_prior_to_xml_tracelog, 324
- mrca_priors_to_xml_prior_distr, 318
- mrca_priors_to_xml_state, 319
- mrca_priors_to_xml_tracelog, 320, 325
- NA, 45, 50, 122, 155, 158, 170, 197, 202, 338, 340
- no_taxa_to_xml_tree, 325
- parameter_to_xml, 325
- parameter_to_xml_alpha, 326
- parameter_to_xml_beta, 327
- parameter_to_xml_clock_rate, 327
- parameter_to_xml_kappa_1, 328
- parameter_to_xml_kappa_2, 328
- parameter_to_xml_lambda, 329
- parameter_to_xml_m, 329
- parameter_to_xml_mean, 330
- parameter_to_xml_mu, 330
- parameter_to_xml_rate_ac, 331
- parameter_to_xml_rate_ag, 331
- parameter_to_xml_rate_at, 332
- parameter_to_xml_rate_cg, 333
- parameter_to_xml_rate_ct, 333
- parameter_to_xml_rate_gt, 334
- parameter_to_xml_s, 334
- parameter_to_xml_scale, 335
- parameter_to_xml_sigma, 336
- phylo, 49, 169, 170
- phylo_to_xml_state, 336
- read.tree, 50
- remove_empty_lines, 337
- remove_multiline, 337
- rename_inference_model_filenames, 35, 39, 79, 86, 88, 91, 118, 140, 159, 161, 169, 198, 215, 226, 230, 338, 338
- rename_mcmc_filenames, 42–44, 47, 77, 85, 111, 120, 150, 153, 170, 199, 249, 316, 317, 339, 340
- return, 50, 170, 338, 340
- rln_clock_model_to_xml_mean_rate_prior, 341
- rnd_phylo_to_xml_init, 341
- site_model_to_xml_lh_distr, 345
- site_model_to_xml_operators, 345
- site_model_to_xml_prior_distr, 346
- site_model_to_xml_state, 347
- site_model_to_xml_subst_model, 347
- site_model_to_xml_tracelog, 348
- site_models_to_xml_operators, 342
- site_models_to_xml_prior_distr, 343
- site_models_to_xml_state, 343
- site_models_to_xml_tracelog, 344, 348
- stop, 12–15, 32, 34, 35, 41, 45, 46, 50, 51, 54, 56, 59, 60, 63, 189, 215
- taxa_to_xml_tree, 348
- tempfile, 203
- tipdate_taxa_to_xml_tree, 349
- tn93_site_model_to_xml_subst_model, 350
- tree_models_to_xml_tracelog, 350
- tree_prior_to_xml_operators, 354
- tree_prior_to_xml_prior_distr, 354
- tree_prior_to_xml_state, 355
- tree_prior_to_xml_tracelog, 356
- tree_priors_to_xml_operators, 351
- tree_priors_to_xml_prior_distr, 352
- tree_priors_to_xml_state, 352
- tree_priors_to_xml_tracelog, 353, 356
- TRUE, 120, 140, 151, 155, 156, 158, 160, 171, 240, 268
- yule_tree_prior_to_xml_prior_distr, 356