

Package ‘beautier’

October 12, 2022

Title 'BEAUti' from R

Version 2.6.5

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

RoxygenNote 7.2.1

VignetteBuilder knitr

URL <https://docs.ropensci.org/beautier/> (website)
<https://github.com/ropensci/beautier/>

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

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

Suggests spelling, devtools, knitr, ggplot2, hunspell, lintr, markdown, readr, 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] (<<https://orcid.org/0000-0002-9499-8873>>),
 Raphael Scherrer [ctb] (<<https://orcid.org/0000-0002-1447-7630>>),
 Yacine B. Chehida [ctb] (<<https://orcid.org/0000-0001-7269-9082>>),
 Katharine S. Walter [ctb] (<<https://orcid.org/0000-0003-0065-2204>>),
 Gary Napier [ctb] (<<https://orcid.org/0000-0002-1077-0055>>),
 Jason Griffiths [ctb] (<<https://orcid.org/0000-0002-1667-8233>>),
 Thijs Janzen [ctb] (<<https://orcid.org/0000-0002-4162-1140>>),
 Jana Riederer [ctb] (<<https://orcid.org/0000-0001-6951-9984>>)

Repository CRAN

Date/Publication 2022-08-11 10:40:07 UTC

R topics documented:

| | |
|--|----|
| alpha_parameter_to_xml | 12 |
| are_clock_models | 13 |
| are_equal_mcmcs | 14 |
| are_equal_screenlogs | 15 |
| are_equal_tracelogs | 16 |
| are_equal_treelogs | 17 |
| are_equal_xml_files | 18 |
| are_equal_xml_lines | 19 |
| are_equivalent_xml_files | 20 |
| are_equivalent_xml_lines | 21 |
| are_equivalent_xml_lines_all | 21 |
| are_equivalent_xml_lines_loggers | 22 |
| are_equivalent_xml_lines_operators | 23 |
| are_equivalent_xml_lines_section | 23 |
| are_fasta_filenames | 24 |
| are_ids | 25 |
| are_init_clock_models | 26 |
| are_init_mrca_priors | 26 |
| are_init_site_models | 27 |
| are_init_tree_priors | 27 |
| are_mrca_align_ids_in_fasta | 28 |
| are_mrca_priors | 28 |
| are_mrca_taxon_names_in_fasta | 29 |
| are_rln_clock_models | 30 |
| are_site_models | 30 |
| are_tree_priors | 31 |
| bd_tree_prior_to_xml_prior_distr | 32 |
| beautier | 33 |
| beta_parameter_to_xml | 34 |
| cbs_tree_prior_to_xml_prior_distr | 34 |
| ccp_tree_prior_to_xml_prior_distr | 35 |
| ccp_tree_prior_to_xml_state | 36 |
| cep_tree_prior_to_xml_prior_distr | 37 |
| check_alignment_id | 38 |

| | |
|---|----|
| check_beauti_options | 38 |
| check_clock_model | 39 |
| check_clock_models | 40 |
| check_empty_beautier_folder | 41 |
| check_filename | 42 |
| check_file_and_model_agree | 42 |
| check_file_exists | 43 |
| check_gamma_site_model | 44 |
| check_gamma_site_model_names | 44 |
| check_gtr_site_model | 45 |
| check_gtr_site_model_names | 46 |
| check_inference_model | 46 |
| check_inference_models | 47 |
| check_is_monophyletic | 48 |
| check_log_mode | 48 |
| check_log_sort | 49 |
| check_mcmc | 49 |
| check_mcmc_list_element_names | 50 |
| check_mcmc_values | 51 |
| check_mrca_prior | 51 |
| check_mrca_prior_name | 52 |
| check_mrca_prior_names | 53 |
| check_mrca_prior_taxa_names | 53 |
| check_ns_mcmc | 54 |
| check_param | 54 |
| check_param_names | 55 |
| check_param_types | 56 |
| check_phylogeny | 56 |
| check_rename_fun | 57 |
| check_rln_clock_model | 58 |
| check_screenlog | 58 |
| check_screenlog_names | 59 |
| check_screenlog_values | 60 |
| check_site_model | 60 |
| check_site_models | 61 |
| check_site_model_names | 62 |
| check_site_model_types | 63 |
| check_store_every | 63 |
| check_strict_clock_model | 64 |
| check_tn93_site_model | 64 |
| check_tn93_site_model_names | 65 |
| check_tracelog | 66 |
| check_tracelog_names | 66 |
| check_tracelog_values | 67 |
| check_treelog | 67 |
| check_treelog_names | 68 |
| check_treelog_values | 68 |
| check_tree_prior | 69 |

| | |
|--|-----|
| check_tree_priors | 70 |
| clock_model_to_xml_operators | 71 |
| clock_model_to_xml_prior_distr | 71 |
| clock_model_to_xml_state | 72 |
| clock_model_to_xml_tracelog | 73 |
| clock_model_to_xml_treelogger | 74 |
| clock_rate_param_to_xml | 74 |
| compare_lines | 75 |
| count_trailing_spaces | 76 |
| create_alpha_param | 77 |
| create_bd_tree_prior | 78 |
| create_beast2_beast_xml | 79 |
| create_beast2_input | 80 |
| create_beast2_input_beast | 81 |
| create_beast2_input_data | 82 |
| create_beast2_input_data_sequences | 83 |
| create_beast2_input_distr | 84 |
| create_beast2_input_distr_lh | 85 |
| create_beast2_input_distr_prior | 86 |
| create_beast2_input_file | 87 |
| create_beast2_input_file_from_model | 89 |
| create_beast2_input_from_model | 90 |
| create_beast2_input_init | 92 |
| create_beast2_input_map | 93 |
| create_beast2_input_operators | 93 |
| create_beast2_input_run | 94 |
| create_beast2_input_state | 95 |
| create_beautier_tempfolder | 96 |
| create_beauti_options | 97 |
| create_beauti_options_v2_4 | 98 |
| create_beauti_options_v2_6 | 99 |
| create_beta_distr | 99 |
| create_beta_param | 101 |
| create_branch_rate_model_rln_xml | 102 |
| create_branch_rate_model_sc_xml | 103 |
| create_branch_rate_model_stuff_xml | 103 |
| create_branch_rate_model_xml | 104 |
| create_cbs_tree_prior | 105 |
| create_ccp_tree_prior | 106 |
| create_cep_tree_prior | 107 |
| create_clock_model | 108 |
| create_clock_models | 109 |
| create_clock_models_from_names | 110 |
| create_clock_model_from_name | 111 |
| create_clock_rate_param | 112 |
| create_clock_rate_state_node_parameter_xml | 113 |
| create_data_xml | 114 |
| create_distr | 115 |

| | |
|---|-----|
| <code>create_exp_distr</code> | 116 |
| <code>create_gamma_distr</code> | 117 |
| <code>create_gamma_site_model</code> | 118 |
| <code>create_gtr_site_model</code> | 120 |
| <code>create_gtr_subst_model_xml</code> | 122 |
| <code>create_hky_site_model</code> | 122 |
| <code>create_hky_subst_model_xml</code> | 123 |
| <code>create_inference_model</code> | 124 |
| <code>create_inv_gamma_distr</code> | 125 |
| <code>create_jc69_site_model</code> | 127 |
| <code>create_jc69_subst_model_xml</code> | 128 |
| <code>create_kappa_1_param</code> | 128 |
| <code>create_kappa_2_param</code> | 129 |
| <code>create_lambda_param</code> | 129 |
| <code>create_laplace_distr</code> | 130 |
| <code>create_loggers_xml</code> | 132 |
| <code>create_log_normal_distr</code> | 133 |
| <code>create_mcmc</code> | 134 |
| <code>create_mean_param</code> | 136 |
| <code>create_mrca_prior</code> | 137 |
| <code>create_mu_param</code> | 138 |
| <code>create_m_param</code> | 139 |
| <code>create_normal_distr</code> | 141 |
| <code>create_ns_inference_model</code> | 142 |
| <code>create_ns_mcmc</code> | 143 |
| <code>create_one_div_x_distr</code> | 145 |
| <code>create_param</code> | 146 |
| <code>create_poisson_distr</code> | 147 |
| <code>create_rate_ac_param</code> | 148 |
| <code>create_rate_ag_param</code> | 149 |
| <code>create_rate_at_param</code> | 150 |
| <code>create_rate_categories_state_node_xml</code> | 151 |
| <code>create_rate_cg_param</code> | 152 |
| <code>create_rate_ct_param</code> | 153 |
| <code>create_rate_gt_param</code> | 155 |
| <code>create_rln_clock_model</code> | 156 |
| <code>create_scale_param</code> | 157 |
| <code>create_screenlog</code> | 158 |
| <code>create_screenlog_xml</code> | 159 |
| <code>create_sigma_param</code> | 160 |
| <code>create_site_model</code> | 161 |
| <code>create_site_models</code> | 162 |
| <code>create_site_models_from_names</code> | 163 |
| <code>create_site_model_from_name</code> | 164 |
| <code>create_site_model_parameters_xml</code> | 165 |
| <code>create_site_model_xml</code> | 166 |
| <code>create_strict_clock_model</code> | 167 |
| <code>create_strict_clock_rate_scaler_operator_xml</code> | 168 |

| | |
|---|-----|
| create_subst_model_xml | 169 |
| create_s_param | 170 |
| create_temp_screenlog_filename | 171 |
| create_temp_tracelog_filename | 172 |
| create_temp_treelog_filename | 173 |
| create_test_inference_model | 173 |
| create_test_mcmc | 175 |
| create_test_ns_inference_model | 176 |
| create_test_ns_mcmc | 177 |
| create_test_screenlog | 179 |
| create_test_tracelog | 179 |
| create_test_treelog | 180 |
| create_tn93_site_model | 181 |
| create_tn93_subst_model_xml | 182 |
| create_tracelog | 183 |
| create_tracelog_xml | 183 |
| create_trait_set_string | 184 |
| create_treelog | 185 |
| create_treelog_xml | 186 |
| create_tree_likelihood_distr_xml | 186 |
| create_tree_prior | 188 |
| create_tree_priors | 189 |
| create_uclid_mean_state_node_param_xml | 190 |
| create_uclid_stdev_state_node_param_xml | 191 |
| create_uniform_distr | 192 |
| create_xml_declaration | 193 |
| create_yule_tree_prior | 193 |
| default_parameters_doc | 194 |
| default_params_doc | 195 |
| distr_to_xml | 201 |
| distr_to_xml_beta | 202 |
| distr_to_xml_exp | 202 |
| distr_to_xml_inv_gamma | 203 |
| distr_to_xml_laplace | 203 |
| distr_to_xml_log_normal | 204 |
| distr_to_xml_normal | 204 |
| distr_to_xml_one_div_x | 205 |
| distr_to_xml_poisson | 205 |
| distr_to_xml_uniform | 206 |
| extract_xml_loggers_from_lines | 206 |
| extract_xml_operators_from_lines | 207 |
| extract_xml_section_from_lines | 207 |
| fasta_file_to_sequences | 208 |
| find_clock_model | 208 |
| find_first_regex_line | 209 |
| find_first_xml_opening_tag_line | 209 |
| find_last_regex_line | 210 |
| find_last_xml_closing_tag_line | 211 |

| | |
|--|-----|
| freq_equilibrium_to_xml | 211 |
| gamma_distr_to_xml | 212 |
| gamma_site_models_to_xml_prior_distr | 213 |
| gamma_site_model_to_xml_prior_distr | 213 |
| gamma_site_model_to_xml_state | 214 |
| get_alignment_id | 214 |
| get_alignment_ids | 215 |
| get_alignment_ids_from_fasta_filenames | 216 |
| get_beautier_folder | 217 |
| get_beautier_path | 217 |
| get_beautier_paths | 218 |
| get_beautier_tempfilename | 219 |
| get_clock_models_ids | 220 |
| get_clock_model_name | 220 |
| get_clock_model_names | 221 |
| get_crown_age | 222 |
| get_distr_names | 222 |
| get_distr_n_params | 223 |
| get_fasta_filename | 224 |
| get_file_base_sans_ext | 224 |
| get_freq_equilibrium_names | 225 |
| get_gamma_site_model_n_distrs | 226 |
| get_gamma_site_model_n_params | 227 |
| get_has_non_strict_clock_model | 228 |
| get_inference_model_filenames | 228 |
| get_log_modes | 229 |
| get_log_sorts | 229 |
| get_mcmc_filenames | 230 |
| get_n_taxa | 230 |
| get_operator_id_pre | 231 |
| get_param_names | 232 |
| get_remove_dir_fun | 233 |
| get_remove_hex_fun | 233 |
| get_replace_dir_fun | 234 |
| get_site_models_n_distrs | 234 |
| get_site_models_n_params | 235 |
| get_site_model_names | 236 |
| get_site_model_n_distrs | 237 |
| get_site_model_n_params | 238 |
| get_taxa_names | 239 |
| get_tree_priors_n_distrs | 239 |
| get_tree_priors_n_params | 240 |
| get_tree_prior_names | 241 |
| get_tree_prior_n_distrs | 242 |
| get_tree_prior_n_params | 243 |
| get_xml_closing_tag | 244 |
| get_xml_opening_tag | 244 |
| has_mrca_prior | 245 |

| | |
|-------------------------------------|-----|
| has_mrca_prior_with_distr | 246 |
| has_rln_clock_model | 247 |
| has_strict_clock_model | 248 |
| has_tip_dating | 249 |
| has_xml_closing_tag | 250 |
| has_xml_opening_tag | 250 |
| has_xml_short_closing_tag | 251 |
| indent | 252 |
| init_bd_tree_prior | 252 |
| init_beta_distr | 253 |
| init_ccp_tree_prior | 253 |
| init_cep_tree_prior | 254 |
| init_clock_models | 254 |
| init_distr | 255 |
| init_exp_distr | 256 |
| init_gamma_distr | 256 |
| init_gamma_site_model | 257 |
| init_gtr_site_model | 258 |
| init_hky_site_model | 259 |
| init_inference_model | 259 |
| init_inv_gamma_distr | 260 |
| init_jc69_site_model | 261 |
| init_laplace_distr | 262 |
| init_log_normal_distr | 262 |
| init_mrca_prior | 263 |
| init_mrca_priors | 263 |
| init_normal_distr | 264 |
| init_one_div_x_distr | 264 |
| init_param | 265 |
| init_poisson_distr | 266 |
| init_rln_clock_model | 266 |
| init_site_models | 267 |
| init_strict_clock_model | 268 |
| init_tn93_site_model | 269 |
| init_tree_priors | 270 |
| init_uniform_distr | 270 |
| init_yule_tree_prior | 271 |
| interspace | 271 |
| is_alpha_param | 272 |
| is_bd_tree_prior | 273 |
| is_beauti_options | 274 |
| is_beta_distr | 275 |
| is_beta_param | 276 |
| is_cbs_tree_prior | 277 |
| is_ccp_tree_prior | 278 |
| is_cep_tree_prior | 279 |
| is_clock_model | 280 |
| is_clock_model_name | 281 |

| | |
|--------------------------------------|-----|
| is_clock_rate_param | 281 |
| is_default_mcmc | 282 |
| is_distr | 283 |
| is_distr_name | 284 |
| is_exp_distr | 285 |
| is_freq_equilibrium_name | 286 |
| is_gamma_distr | 287 |
| is_gamma_site_model | 288 |
| is_gtr_site_model | 289 |
| is_hky_site_model | 290 |
| is_id | 291 |
| is_inference_model | 292 |
| is_init_bd_tree_prior | 292 |
| is_init_beta_distr | 293 |
| is_init_cbs_tree_prior | 293 |
| is_init_ccp_tree_prior | 294 |
| is_init_cep_tree_prior | 294 |
| is_init_clock_model | 295 |
| is_init_distr | 295 |
| is_init_exp_distr | 296 |
| is_init_gamma_distr | 296 |
| is_init_gamma_site_model | 297 |
| is_init_gtr_site_model | 297 |
| is_init_hky_site_model | 298 |
| is_init_inv_gamma_distr | 299 |
| is_init_jc69_site_model | 300 |
| is_init_laplace_distr | 301 |
| is_init_log_normal_distr | 301 |
| is_init_mrca_prior | 302 |
| is_init_normal_distr | 302 |
| is_init_one_div_x_distr | 303 |
| is_init_param | 303 |
| is_init_poisson_distr | 304 |
| is_init_rln_clock_model | 304 |
| is_init_site_model | 305 |
| is_init_strict_clock_model | 305 |
| is_init_tn93_site_model | 306 |
| is_init_tree_prior | 306 |
| is_init_uniform_distr | 307 |
| is_init_yule_tree_prior | 308 |
| is_inv_gamma_distr | 308 |
| is_in_patterns | 309 |
| is_jc69_site_model | 310 |
| is_kappa_1_param | 311 |
| is_kappa_2_param | 312 |
| is_lambda_param | 313 |
| is_laplace_distr | 314 |
| is_log_normal_distr | 315 |

| | |
|---|-----|
| is_mcmc | 316 |
| is_mcmc_nested_sampling | 317 |
| is_mean_param | 318 |
| is_mrca_align_ids_in_fastas | 319 |
| is_mrca_align_id_in_fasta | 320 |
| is_mrca_prior | 320 |
| is_mrca_prior_with_distr | 321 |
| is_mu_param | 322 |
| is_m_param | 323 |
| is_normal_distr | 324 |
| is_one_bool | 325 |
| is_one_div_x_distr | 326 |
| is_one_double | 327 |
| is_one_int | 328 |
| is_one_na | 329 |
| is_on_appveyor | 330 |
| is_on_ci | 330 |
| is_on_github_actions | 331 |
| is_on_travis | 332 |
| is_param | 332 |
| is_param_name | 333 |
| is_phylo | 334 |
| is_poisson_distr | 335 |
| is_rate_ac_param | 336 |
| is_rate_ag_param | 337 |
| is_rate_at_param | 338 |
| is_rate_cg_param | 340 |
| is_rate_ct_param | 341 |
| is_rate_gt_param | 342 |
| is_rln_clock_model | 343 |
| is_scale_param | 344 |
| is_sigma_param | 345 |
| is_site_model | 346 |
| is_site_model_name | 347 |
| is_strict_clock_model | 348 |
| is_s_param | 349 |
| is_tn93_site_model | 350 |
| is_tree_prior | 351 |
| is_tree_prior_name | 352 |
| is_uniform_distr | 352 |
| is_xml | 353 |
| is_yule_tree_prior | 354 |
| mcmc_to_xml_run | 355 |
| mcmc_to_xml_run_default | 355 |
| mcmc_to_xml_run_nested_sampling | 356 |
| mrca_priors_to_xml_prior_distr | 357 |
| mrca_prior_to_xml_prior_distr | 358 |
| mrca_prior_to_xml_state | 359 |

| | |
|--|-----|
| mrca_prior_to_xml_taxonset | 360 |
| mrca_prior_to_xml_tracelog | 360 |
| m_param_to_xml | 361 |
| no_taxa_to_xml_tree | 362 |
| parameter_to_xml | 363 |
| parameter_to_xml_kappa_1 | 363 |
| parameter_to_xml_kappa_2 | 364 |
| parameter_to_xml_lambda | 365 |
| parameter_to_xml_mean | 365 |
| parameter_to_xml_mu | 366 |
| parameter_to_xml_rate_ac | 366 |
| parameter_to_xml_rate_ag | 367 |
| parameter_to_xml_rate_at | 368 |
| parameter_to_xml_rate_cg | 368 |
| parameter_to_xml_rate_ct | 369 |
| parameter_to_xml_rate_gt | 370 |
| parameter_to_xml_s | 370 |
| parameter_to_xml_scale | 371 |
| parameter_to_xml_sigma | 372 |
| remove_beautier_folder | 372 |
| remove_empty_lines | 373 |
| remove_multiline | 374 |
| rename_inference_model_filenames | 374 |
| rename_mcmc_filenames | 375 |
| rln_clock_model_to_xml_mean_rate_prior | 377 |
| rln_clock_model_to_xml_prior_distr | 377 |
| rnd_phylo_to_xml_init | 379 |
| site_models_to_xml_operators | 380 |
| site_models_to_xml_prior_distr | 380 |
| site_models_to_xml_tracelog | 381 |
| site_model_to_xml_operators | 382 |
| site_model_to_xml_prior_distr | 382 |
| site_model_to_xml_state | 383 |
| site_model_to_xml_tracelog | 383 |
| taxa_to_xml_tree | 384 |
| tipdate_taxa_to_xml_trait | 385 |
| tipdate_taxa_to_xml_tree | 386 |
| tree_model_to_tracelog_xml | 386 |
| tree_priors_to_xml_prior_distr | 387 |
| tree_priors_to_xml_tracelog | 388 |
| tree_prior_to_xml_operators | 389 |
| tree_prior_to_xml_prior_distr | 390 |
| tree_prior_to_xml_state | 391 |
| tree_prior_to_xml_tracelog | 391 |
| unindent | 392 |
| yule_tree_prior_to_xml_operators | 393 |
| yule_tree_prior_to_xml_prior_distr | 393 |

alpha_parameter_to_xml

Internal function

Description

Converts an alpha parameter to XML

Usage

```
alpha_parameter_to_xml(  
  alpha_parameter,  
  beauti_options = create_beauti_options()  
)
```

Arguments

alpha_parameter

an alpha parameter, as created by [create_alpha_param](#)

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

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
remove_beautier_folder()  
check_empty_beautier_folder()  
  
# The alpha parameter must be initialized, i.e. have an ID  
alpha_parameter_to_xml(alpha_parameter = create_alpha_param(id = "1"))  
  
check_empty_beautier_folder()
```

| | |
|------------------|--|
| 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

```
check_empty_beautier_folder()

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

# FALSE
are_clock_models(NA)
are_clock_models(NULL)
are_clock_models("nonsense")
are_clock_models(create_jc69_site_model())

check_empty_beautier_folder()
```

are_equal_mcmc *Determine if two MCMCs are equal.*

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

```
if (is_on_ci()) {  
  check_empty_beautier_folder()  
  
  mcmc_1 <- create_mcmc(chain_length = 1000)  
  mcmc_2 <- create_mcmc(chain_length = 314)  
  # TRUE  
  are_equal_mcmc(mcmc_1, mcmc_1)  
  # FALSE  
  are_equal_mcmc(mcmc_1, mcmc_2)  
  
  check_empty_beautier_folder()  
}
```

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

```
check_empty_beautier_folder()

screenlog_1 <- create_screenlog(log_every = 1000)
screenlog_2 <- create_screenlog(log_every = 314)
# TRUE
are_equal_screenlogs(screenlog_1, screenlog_1)
# FALSE
are_equal_screenlogs(screenlog_1, screenlog_2)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

tracelog_1 <- create_tracelog(log_every = 1000)
tracelog_2 <- create_tracelog(log_every = 314)
# TRUE
are_equal_tracelogs(tracelog_1, tracelog_1)
# FALSE
are_equal_tracelogs(tracelog_1, tracelog_2)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

treelog_1 <- create_treelog(log_every = 1000)
treelog_2 <- create_treelog(log_every = 314)
# TRUE
are_equal_treelogs(treelog_1, treelog_1)
# FALSE
are_equal_treelogs(treelog_1, treelog_2)

check_empty_beautier_folder()
```

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](#)

Examples

```
check_empty_beautier_folder()

are_equal_xml_files(
  filename_1 = get_beautier_path("2_4.xml"),
  filename_2 = get_beautier_path("2_6_0.xml"),
  section = "taxonset"
)

check_empty_beautier_folder()
```

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

Examples

```
check_empty_beautier_folder()

are_equal_xml_lines(
  lines_1 = readLines(get_beautier_path("2_4.xml")),
  lines_2 = readLines(get_beautier_path("2_6_0.xml")),
  section = "taxonset"
)

check_empty_beautier_folder()
```

`are_equivalent_xml_files`*Internal function*

Description

Internal function used for debugging to determine if XML files result in equivalent trees

Usage

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

Arguments

| | |
|-------------------------|---|
| <code>filename_1</code> | name of a first XML file |
| <code>filename_2</code> | name of a second XML file |
| <code>section</code> | 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`

Examples

```
check_empty_beautier_folder()

are_equivalent_xml_files(
  filename_1 = get_beautier_path("2_4.xml"),
  filename_2 = get_beautier_path("2_6_0.xml")
)

check_empty_beautier_folder()
```

`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

| | |
|----------------------|---|
| <code>lines_1</code> | lines of a first XML file |
| <code>lines_2</code> | lines of a second XML file |
| <code>section</code> | the name of the XML section |
| <code>verbose</code> | 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

| | |
|----------------------|---|
| <code>lines_1</code> | lines of a first XML file |
| <code>lines_2</code> | lines of a second XML file |
| <code>verbose</code> | 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

| | |
|----------------------|---|
| <code>lines_1</code> | lines of a first XML file |
| <code>lines_2</code> | lines of a second XML file |
| <code>verbose</code> | 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

| | |
|----------------------|---|
| <code>lines_1</code> | lines of a first XML file |
| <code>lines_2</code> | lines of a second XML file |
| <code>section</code> | the name of the XML section |
| <code>verbose</code> | 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

```
check_empty_beautier_folder()

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

# FALSE
are_fasta_filenames("")
are_fasta_filenames(NA)
are_fasta_filenames(NULL)
are_fasta_filenames(Inf)
are_fasta_filenames("1.fasX")
are_fasta_filenames(c("1.fas", "2.exe"))
are_fasta_filenames(c("1.bat", "2.exe"))
```



```
check_empty_beautier_folder()
```

| | |
|---------|---|
| are_ids | <i>Determine if x consists out of IDs</i> |
|---------|---|

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](#)

Examples

```
check_empty_beautier_folder()

# TRUE
are_ids("anthus_aco")
are_ids(c("anthus_aco", "anthus_nd2"))
are_ids(list("anthus_aco", "anthus_nd2"))
are_ids(c(1, 2))
are_ids(1)

# FALSE
are_ids(NULL)
are_ids(NA)
are_ids(c())
are_ids(ape::rcoal(3))
are_ids(c(ape::rcoal(3), ape::rcoal(4)))

check_empty_beautier_folder()
```

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. Note that BEAST2 also supports missing data, by using a dash (-) or question mark (?) as a sequence.

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. Note that BEAST2 also supports missing data, by using a dash (-) or question mark (?) as a sequence.

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

```
check_empty_beautier_folder()

jc69_site_model <- create_jc69_site_model()
gtr_site_model <- create_gtr_site_model()
both_site_models <- list(jc69_site_model, gtr_site_model)

# TRUE
are_site_models(jc69_site_model)

# TRUE
are_site_models(gtr_site_model)

# TRUE
are_site_models(both_site_models)

check_empty_beautier_folder()
```

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

```

check_empty_beautier_folder()

yule_tree_prior <- create_yule_tree_prior()
bd_tree_prior <- create_bd_tree_prior()
both_tree_priors <- list(yule_tree_prior, bd_tree_prior)
# TRUE
are_tree_priors(yule_tree_prior)
# TRUE
are_tree_priors(bd_tree_prior)
# TRUE
are_tree_priors(both_tree_priors)

check_empty_beautier_folder()

```

```
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

```

check_empty_beautier_folder()

# <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_empty_beautier_folder()
```

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.

Details

See the documentation of `create_inference_model` to see the features of BEAST2 that beautier supports.

Author(s)

Richèl J.C. Bilderbeek

See Also

These are packages associated with beautier:

- The package `beastier` can run BEAST2 from R
- The package `tracrer` 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 tracrer into a single workflow

Examples

```
if (is_on_ci()) {  
  
  check_empty_beautier_folder()  
  
  # Get an example FASTA file  
  input_filename <- get_fasta_filename()  
  
  # The file created by beautier, a BEAST2 input file  
  output_filename <- get_beautier_tempfilename()  
  
  # 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()  
  )  
}
```

```
# Remove the folder that contains the beautier temporary files
remove_beautier_folder()
check_empty_beautier_folder()
}
```

beta_parameter_to_xml *Internal function*

Description

Converts a beta parameter to XML

Usage

```
beta_parameter_to_xml(beta_parameter, beauti_options = create_beauti_options())
```

Arguments

beta_parameter a beta parameter, as created by [create_beta_param](#)

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

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

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

```
check_empty_beautier_folder()

# <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_empty_beautier_folder()
```

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

```

check_empty_beautier_folder()

# <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_empty_beautier_folder()

```

```
ccp_tree_prior_to_xml_state
```

Convert a CCP tree prior to the XML as part of the state section

Description

Convert a CCP tree prior to the XML as part of the state section

Usage

```
ccp_tree_prior_to_xml_state(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

XML as text

Examples

```

check_empty_beautier_folder()

# Need an ID and initial value
inference_model <- create_inference_model(
  tree_prior = create_ccp_tree_prior(
    id = "anthus_nd2_sub",
    pop_size_distr = create_normal_distr(
      id = 123,
      value = 3.14
    )
  )
)

```

```

    )
  )
)

ccp_tree_prior_to_xml_state(inference_model)

check_empty_beautier_folder()

```

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

```

check_empty_beautier_folder()

# <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_empty_beautier_folder()

```

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

```
check_empty_beautier_folder()

# anthus_aco_sub
alignment_id <- get_alignment_id("/home/homer/anthus_aco_sub.fas")
check_alignment_id(alignment_id)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

check_beauti_options(create_beauti_options())

check_empty_beautier_folder()
```

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

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

```
check_empty_beautier_folder()

check_clock_model(create_strict_clock_model())
check_clock_model(create_rln_clock_model())

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

check_clock_models(create_strict_clock_model())
check_clock_models(list(create_strict_clock_model()))
check_clock_models(
  list(create_strict_clock_model(), create_rln_clock_model())
)

check_empty_beautier_folder()
```

`check_empty_beautier_folder`*Internal function*

Description

Internal function to verify that, if there are ‘beautier’ temporary files created, these are also cleaned up, as by CRAN policy.

Usage

```
check_empty_beautier_folder(beautier_folder = get_beautier_folder())
```

Arguments

`beautier_folder`

the path to the [beautier](#) temporary files folder

Details

If the ‘beautier’ folder does not exist, this function does nothing. If there are folder and/or files in the ‘beautier’ folder, an error is given.

Value

Nothing.

Author(s)

Richèl J.C. Bilderbeek

See Also

use [remove_beautier_folder](#) to remove the default ‘beautier’ folder

Examples

```
remove_beautier_folder()
```

```
check_empty_beautier_folder()
```

```
remove_beautier_folder()
```

| | |
|----------------|---|
| check_filename | <i>Check if the 'filename' is valid</i> |
|----------------|---|

Description

Calls stop if the filename is invalid

Usage

```
check_filename(filename, allow_empty_str = FALSE, allow_na = FALSE)
```

Arguments

| | |
|-----------------|---|
| filename | a filename, as can be checked by check_filename |
| allow_empty_str | allow a string to be empty |
| allow_na | allow NA |

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

check_filename("trace.log")
check_filename("my.trees")

check_empty_beautier_folder()
```

| | |
|----------------------------|--|
| check_file_and_model_agree | <i>Checks if the input FASTA file and the inference model agree.</i> |
|----------------------------|--|

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

```
check_empty_beautier_folder()

check_file_exists(get_beautier_path("anthus_aco_sub.fas"))

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()
```

```
check_gamma_site_model(create_gamma_site_model())
```

```
check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()  
  
check_gtr_site_model(create_gtr_site_model())  
  
check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()
```

```
check_inference_model(create_inference_model())
```

```
check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()
```

```
check_inference_models(list(create_inference_model()))
```

```
check_empty_beautier_folder()
```

`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 | <i>Check if the supplied sort is a valid logging sorting option.</i> |
|----------------|--|

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 | <i>Check if the MCMC is a valid MCMC object.</i> |
|------------|--|

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

```
check_empty_beautier_folder()

check_mcmc(create_mcmc())

check_empty_beautier_folder()
```

```
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 | <i>Check if the MCMC has the list elements with valid values for being a valid MCMC object.</i> |
|-------------------|---|

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 | <i>Check if the MRCA prior is a valid MRCA prior.</i> |
|------------------|---|

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

```
check_empty_beautier_folder()

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)
)
check_mrca_prior(mrca_prior)

check_empty_beautier_folder()
```

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 | <i>Check if this an MCMC that uses Nested Sampling to estimate a marginal likelihood.</i> |
|---------------|---|

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 | <i>Check if the parameter is a valid parameter</i> |
|-------------|--|

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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

check_param_names *Check if the param has the list elements of a valid param object.*

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 from the ape package

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See AlsoUse `ape::read.tree` to create a phylogeny**Examples**

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

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

DescriptionWill [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: <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> |
|------------|--|

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

```
check_empty_beautier_folder()

check_rln_clock_model(create_rln_clock_model())

check_empty_beautier_folder()
```

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](#)

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()  
check_screenlog(create_test_screenlog())  
check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

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

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

check_empty_beautier_folder()
```

check_site_models *Check if the object is a list of one or more site models.*

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

```
check_empty_beautier_folder()

check_site_models(create_jc69_site_model())
check_site_models(list(create_jc69_site_model()))
check_site_models(
  list(create_jc69_site_model(), create_gtr_site_model())
)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

check_strict_clock_model(create_strict_clock_model())

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

check_tn93_site_model(create_tn93_site_model())

check_empty_beautier_folder()
```

```
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 *Check if a tracelog is valid.*

Description

Will call [stop](#) if not.

Usage

```
check_tracelog(tracelog)
```

Arguments

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

check_tracelog_names *Check if the tracelog has the list elements of a valid tracelog object.*

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](#)

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

check_treelog(create_test_treelog())

check_empty_beautier_folder()
```

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

tree_log a tree_log, as created by [create_tree_log](#)

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_tree_log](#) to create a valid tree_log

check_tree_prior *Check if the tree prior is a valid tree prior*

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

```
check_empty_beautier_folder()

check_tree_prior(create_yule_tree_prior())
check_tree_prior(create_bd_tree_prior())
check_tree_prior(create_cbs_tree_prior())
check_tree_prior(create_ccp_tree_prior())
check_tree_prior(create_cep_tree_prior())

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

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

check_tree_priors(create_yule_tree_prior())
check_tree_priors(list(create_yule_tree_prior()))
check_tree_priors(list(create_yule_tree_prior(), create_bd_tree_prior()))

check_empty_beautier_folder()
```

`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(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

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

`clock_model_to_xml_prior_distr`*Internal function*

Description

Internal function to converts a clock model to the prior section of the XML as text

Usage

```
clock_model_to_xml_prior_distr(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

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()
# <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_empty_beautier_folder()
```

clock_model_to_xml_state

Internal function

Description

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

Usage

```
clock_model_to_xml_state(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

lines of XML text, without indentation nor state tags

Author(s)

Richèl J.C. Bilderbeek

clock_model_to_xml_tracelog

Internal function

Description

Creates the clock model's XML for the tracelog section

Usage

```
clock_model_to_xml_tracelog(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

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

`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

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

`clock_rate_param_to_xml`*Internal function*

Description

Converts a clockRate parameter to XML

Usage

```
clock_rate_param_to_xml(  
  clock_rate_param,  
  beauti_options = create_beauti_options()  
)
```

Arguments

clock_rate_param
 a clockRate parameter, a numeric value, as created by [create_clock_rate_param](#)
 beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

| | |
|---------------|--------------------------|
| compare_lines | <i>Internal function</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 = get_beautier_tempfilename(pattern = "created", fileext =
    ".xml"),
  expected_lines_filename = get_beautier_tempfilename(pattern = "expected", fileext =
    ".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

Examples

```
check_empty_beautier_folder()

# Creates temporary files in beautier folder
compare_lines(
  lines = readLines(beautier::get_beautier_path("bd_2_4.xml")),
  expected = readLines(beautier::get_beautier_path("bd_2_4.xml"))
)

remove_beautier_folder()
check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# 0
count_trailing_spaces("x")
# 1
count_trailing_spaces(" y")
# 2
count_trailing_spaces(" <")
```

```
# 0
count_trailing_spaces("")
# 1
count_trailing_spaces(" ")
# 2
count_trailing_spaces("  ")

check_empty_beautier_folder()
```

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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = beta_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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

```
if (is_on_ci()) {  
  
  bd_tree_prior <- create_bd_tree_prior()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = bd_tree_prior  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

```
create_beast2_beast_xml
```

Create the <beast ...> XML

Description

The <beast ...> XML is the XML at the start of a BEAST2 XML input file, directly after the general XML declaration (as created by [create_xml_declaration](#)).

Usage

```
create_beast2_beast_xml(beauti_options)
```

Arguments

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

Value

the XML

Author(s)

Richèl J.C. Bilderbeek

Examples

```
remove_beautier_folder()

create_beast2_beast_xml(
  beauti_options = create_beauti_options_v2_6()
)

check_empty_beautier_folder()
```

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 = beautier::create_jc69_site_model(),
  clock_model = beautier::create_strict_clock_model(),
  tree_prior = beautier::create_yule_tree_prior(),
  mrca_prior = NA,
  mcmc = beautier::create_mcmc(),
  beauti_options = beautier::create_beauti_options()
)
```

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](#)

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

```
if (is_on_ci()) {
  create_beast2_input(
    input_filename = get_fasta_filename()
  )
}
```

`create_beast2_input_beast`

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

Description

Creates the XML text for the beast tag of a BEAST2 parameter file, which is directly after the XML declaration (created by [create_xml_declaration](#)).

Usage

```
create_beast2_input_beast(
  input_filename,
  inference_model = beautier::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.

Details

The beast tag has these elements:

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

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_filename,  
  beauti_options = beautier::create_beauti_options()  
)
```

Arguments

`input_filename` A FASTA filename. 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

Examples

```
check_empty_beautier_folder()  
  
create_beast2_input_data(  
  input_filename = get_fasta_filename(),  
  beauti_options = create_beauti_options_v2_4()  
)  
  
check_empty_beautier_folder()
```

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 = beautier::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(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

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**

```

check_empty_beautier_folder()

inference_model <- init_inference_model(
  input_filename = get_fasta_filename(),
  inference_model = create_inference_model(
    beauti_options = create_beauti_options_v2_4()
  )
)
create_beast2_input_distr(
  inference_model = inference_model
)
# <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_empty_beautier_folder()

```

```
create_beast2_input_distr_lh
```

Creates the XML text for the distribution tag with the likelihood ID, of a BEAST2 parameter file.

Description

Creates the XML text for the distribution tag with the likelihood ID, of a BEAST2 parameter file, in an unindented form

Usage

```
create_beast2_input_distr_lh(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.

Details

The distribution tag (with ID equals likelihood) has these elements:

```
<distribution id="likelihood" [...]>
  <distribution id="treeLikelihood" [...]>
    [...]
  </distribution>
</distribution>
```

The distribution section with ID treeLikelihood is created by [create_tree_likelihood_distr_xml](#).

Zooming out:

```
<beast [...]>
  <run [...]>
    <distribution id="posterior" [...]>
      <distribution id="likelihood" [...]>
        [this section]
      </distribution>
    </distribution>
  </run>
</beast>
```

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

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(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.

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

```
check_empty_beautier_folder()

# <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_empty_beautier_folder()
```

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 = beautier::create_jc69_site_model(),
```

```

clock_model = beautier::create_strict_clock_model(),
tree_prior = beautier::create_yule_tree_prior(),
mrca_prior = NA,
mcmc = beautier::create_mcmc(),
beauti_options = beautier::create_beauti_options(),
tipdates_filename = NA
)

```

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.

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

```
if (is_on_ci()) {  
  
  check_empty_beautier_folder()  
  
  # Get an example FASTA file  
  input_filename <- get_fasta_filename()  
  
  # The file created by beautier, a BEAST2 input file  
  output_filename <- get_beautier_tempfilename()  
  
  create_beast2_input_file(  
    input_filename,  
    output_filename  
  )  
  file.remove(output_filename)  
  
  remove_beautier_folder()  
  check_empty_beautier_folder()  
}
```

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 = beautier::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

```
if (is_on_ci()) {
  check_empty_beautier_folder()

  output_filename <- get_beautier_tempfilename()
  create_beast2_input_file_from_model(
    input_filename = get_fasta_filename(),
    output_filename = output_filename,
    inference_model = create_inference_model()
  )
  file.remove(output_filename)

  remove_beautier_folder()
  check_empty_beautier_folder()
}
```

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

```
if (is_on_ci()) {  
  check_empty_beautier_folder()  
  
  text <- create_beast2_input_from_model(  
    input_filename = get_fasta_filename(),  
    inference_model = create_inference_model()  
  )  
  
  check_empty_beautier_folder()  
}
```

`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(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

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

inference_model <- init_inference_model(
  input_filename = get_fasta_filename(),
  inference_model = create_test_inference_model()
)
xml <- create_beast2_input_init(
  inference_model = inference_model
)

check_empty_beautier_folder()
```

`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(beauti_options)
```

Arguments

`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_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(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

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

Creates the 'run' section of a BEAST2 XML parameter file, without being indented.

Usage

```
create_beast2_input_run(
  input_filename,
  inference_model = beautier::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.

Details

The run tag has these elements:

```
<run[...]>
  <state[...]>
  [...]
</state>
  <init[...]>
  [...]
</init>
  <distribution[...]>
  [...]
</distribution>
```

```

    [operator ids]
    [loggers]
  </run>

```

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, without being indented.

Usage

```
create_beast2_input_state(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.

Details

The state tag has these elements:

```

<state[...]>
  <tree[...]>
  [...]
  </tree>
  [parameters]
</run>

```

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 tree tag. to create the XML text of the [parameters] section.

create_beautier_tempfolder

Create the default 'beautier' temporary folder

Description

Create the default 'beautier' temporary folder

Usage

```
create_beautier_tempfolder()
```

Value

nothing

Author(s)

Richèl J.C. Bilderbeek

Examples

```
create_beautier_tempfolder()
```

```
remove_beautier_folder()
```

```
check_empty_beautier_folder()
```

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

Details

Available BEAUti options are:

- [create_beauti_options_v2_4](#)
- [create_beauti_options_v2_6](#)

Value

a BEAUti options structure

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (is_on_ci()) {  
  
  check_empty_beautier_folder()  
  
  beauti_options <- create_beauti_options_v2_4()  
  xml <- create_beast2_input(  
    get_fasta_filename(),  
    beauti_options = beauti_options  
  )  
  
  check_empty_beautier_folder()  
}
```

create_beauti_options_v2_4

Function to create the BEAUti options for version 2.4.

Description

Function to create the BEAUti options for version 2.4, by calling [create_beauti_options](#).

Usage

```
create_beauti_options_v2_4()
```

Value

a BEAUti options structure

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (is_on_ci()) {  
  check_empty_beautier_folder()  
  
  beauti_options <- create_beauti_options_v2_4()  
  xml <- create_beast2_input(  
    get_fasta_filename(),  
    beauti_options = beauti_options  
  )  
  
  check_empty_beautier_folder()  
}
```

`create_beauti_options_v2_6`*Function to create the BEAUti options for version 2.6.*

Description

Function to create the BEAUti options for version 2.6, by calling [create_beauti_options](#).

Usage

```
create_beauti_options_v2_6()
```

Value

a BEAUti options structure

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (is_on_ci()) {  
  check_empty_beautier_folder()  
  
  beauti_options <- create_beauti_options_v2_6()  
  xml <- create_beast2_input(  
    get_fasta_filename(),  
    beauti_options = beauti_options  
  )  
  
  check_empty_beautier_folder()  
}
```

`create_beta_distr`*Create a beta distribution*

Description

Create a beta distribution

Usage

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

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

Value

a beta distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
if (is_on_ci()) {  
  
  beta_distr <- create_beta_distr()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = beta_distr  
    )  
  )  
  file.remove(beast2_input_file)
```

```
    remove_beautier_folder()
  }
```

| | |
|-------------------|---------------------------------------|
| 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

```
if (is_on_ci()) {
  # 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 <- get_beautier_tempfilename()
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = create_yule_tree_prior(
    birth_rate_distr = gamma_distr
  )
)
file.remove(beast2_input_file)

remove_beautier_folder()
}
```

create_branch_rate_model_rln_xml
Internal function

Description

Internal function to call [create_branch_rate_model_xml](#) for a relaxed log-normal clock.

Usage

```
create_branch_rate_model_rln_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.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

`create_branch_rate_model_sc_xml`

Internal function to call [create_branch_rate_model_xml](#) for a strict clock.

Description

Internal function to call [create_branch_rate_model_xml](#) for a strict clock.

Usage

```
create_branch_rate_model_sc_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.

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

`create_branch_rate_model_stuff_xml`

Internal function called by [create_branch_rate_model_xml](#)

Description

It generates the desired XML for some circumstances. Yes, that is a vague description. Would be nice if someone would untangle this :-)

Usage

```
create_branch_rate_model_stuff_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.

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

 create_branch_rate_model_xml

Internal function to create the branchRateModel section of the XML as text.

Description

Creates the branchRateModel section of the XML as text.

Usage

```
create_branch_rate_model_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.

Details

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

The distribution tag (with ID equals treeLikelihood) has these elements:

```
<branchRateModel[...]>
  [...]
</branchRateModel>
```


When there is a strict clock, [create_branch_rate_model_sc_xml](#) is called. When there is an RLN clock, [create_branch_rate_model_rln_xml](#) is called.

Zooming out:

```
<beast[...]>
  <run[...]>
    <distribution id="posterior"[...]>
      <distribution id="likelihood"[...]>
        <distribution id="treeLikelihood"[...]>
          [...]

          [this section]
        </distribution>
      </distribution>
    </distribution>
  </run>
</beast>
```

Value

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

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

```
if (is_on_ci()) {  
  
  cbs_tree_prior <- create_cbs_tree_prior()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_beautier_path("test_output_6.fas"),  
    beast2_input_file,  
    tree_prior = cbs_tree_prior  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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(value = 0.3)  
)
```

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

```
if (is_on_ci()) {
  ccp_tree_prior <- create_ccp_tree_prior()

  beast2_input_file <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    tree_prior = ccp_tree_prior
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}
```

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

```
if (is_on_ci()) {  
  
  cep_tree_prior <- create_cep_tree_prior()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = cep_tree_prior  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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 get_clock_model_names |
| 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

```
if (is_on_ci()) {  
  # Can use any of these models  
  strict_clock_model <- create_strict_clock_model()  
  rln_clock_model <- create_rln_clock_model()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    get_fasta_filename(),  
    beast2_input_file,  
    clock_model = strict_clock_model  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

create_clock_models_from_names(get_clock_model_names())

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

create_clock_model_from_name(get_clock_model_names()[1])

check_empty_beautier_folder()
```

`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", estimate = FALSE, id = NA)
```

Arguments

| | |
|-----------------------|--|
| <code>value</code> | value of the parameter |
| <code>estimate</code> | TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise |
| <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

```
if (is_on_ci()) {  
  
  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 <- get_beautier_tempfilename()
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  clock_model = strict_clock_model
)
file.remove(beast2_input_file)

remove_beautier_folder()
}
```

create_clock_rate_state_node_parameter_xml

Internal function

Description

Creates the clockRate parameter with the name stateNode, such as: `<parameter id="uclDStdev.c:[id]" [...] name="stateNode">0.1</parameter>`

Usage

```
create_clock_rate_state_node_parameter_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.

Value

the following XML: `<parameter id="uclDStdev.c:[id]" lower="0.0" name="stateNode"> 0.1 </parameter>`

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

create_ucld_stdev_state_node_param_xml(
  create_inference_model(
    clock_model = create_rln_clock_model(id = 314)
  )
)

check_empty_beautier_folder()
```

| | |
|-----------------|--|
| create_data_xml | <i>Create the <data . .> XML</i> |
|-----------------|--|

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 | <i>General function to create a distribution.</i> |
|--------------|---|

Description

General function to create a distribution.

Usage

```
create_distr(name, id, value = NA, lower = NA, upper = NA, ...)
```

Arguments

| | |
|-------|---|
| name | the distribution name. Valid names can be found in <code>get_distr_names</code> |
| id | the distribution's ID |
| value | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to <code>Inf</code> . |
| ... | 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

```
if (is_on_ci()) {  
  
  # Use any distribution  
  distr <- create_beta_distr()  
  
  beast2_input_file <- get_beautier_tempfilename()  
}
```

```
create_beast2_input_file(  
  input_filename = get_fasta_filename(),  
  beast2_input_file,  
  tree_prior = create_yule_tree_prior(  
    birth_rate_distr = distr  
  )  
)  
file.remove(beast2_input_file)  
  
remove_beautier_folder()  
}
```

create_exp_distr *Create an exponential distribution*

Description

Create an exponential distribution

Usage

```
create_exp_distr(id = NA, mean = 1, value = NA, lower = NA, upper = NA)
```

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

Value

an exponential distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
if (is_on_ci()) {  
  
  exp_distr <- create_exp_distr()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = exp_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

create_gamma_distr *Create a gamma distribution*

Description

Create a gamma distribution

Usage

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

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

Value

a gamma distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
if (is_on_ci()) {  
  
  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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    site_model = gtr_site_model  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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

```

if (is_on_ci()) {
  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 <- get_beautier_tempfilename()
  create_beast2_input_file(
    get_fasta_filename(),
    beast2_input_file,
    site_model = site_model
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}

```

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. `get_freq_equilibrium_names` returns the possible values for `freq_equilibrium`

Value

a GTR site_model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (is_on_ci()) {

  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 <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    site_model = gtr_site_model
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}
```

```
create_gtr_subst_model_xml
```

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

```
create_gtr_subst_model_xml(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

```
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

```
if (is_on_ci()) {
  hky_site_model <- create_hky_site_model()
  output_filename <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    output_filename = output_filename,
    site_model = hky_site_model
  )
  file.remove(output_filename)

  remove_beautier_folder()
}
```

```
create_hky_subst_model_xml
```

Converts a site model to XML, used in the substModel section

Description

Converts a site model to XML, used in the substModel section

Usage

```
create_hky_subst_model_xml(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

```
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 = beautier::create_jc69_site_model(),
  clock_model = beautier::create_strict_clock_model(),
  tree_prior = beautier::create_yule_tree_prior(),
  mrca_prior = NA,
  mcmc = beautier::create_mcmc(),
  beauti_options = beautier::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_test_inference_model](#) to create an inference model with a short MCMC, to be used in testing. Use [create_ns_inference_model](#) to create an inference model to estimate the marginal likelihood (aka evidence) using a nested sampling approach.

Examples

```
if (is_on_ci()) {  
  
  check_empty_beautier_folder()  
  
  # Create an MCMC chain with 50 states  
  inference_model <- create_inference_model(  
    mcmc = create_mcmc(chain_length = 50000, store_every = 1000)  
  )  
  
  output_filename <- get_beautier_tempfilename()  
  create_beast2_input_file_from_model(  
    input_filename = get_fasta_filename(),  
    output_filename = output_filename,  
    inference_model = inference_model  
  )  
  file.remove(output_filename)  
  
  remove_beautier_folder()  
  check_empty_beautier_folder()  
}
```

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,  
  value = NA,  
  lower = NA,  
  upper = NA  
)
```

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

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

```
if (is_on_ci()) {  
  
  inv_gamma_distr <- create_inv_gamma_distr()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = inv_gamma_distr  
    )  
  )  
  file.remove(beast2_input_file)
```

```
    remove_beautier_folder()
  }
```

```
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

```
if (is_on_ci()) {
  jc69_site_model <- create_jc69_site_model()

  output_filename <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    output_filename = output_filename,
    site_model = jc69_site_model
  )
  file.remove(output_filename)

  remove_beautier_folder()
}
```

```
create_jc69_subst_model_xml
```

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

```
create_jc69_subst_model_xml(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

```
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**

| | |
|-----------------------|--|
| <code>id</code> | the parameter's ID |
| <code>lower</code> | lowest possible value of the parameter. If the parameter is estimated, lower must be less than value |
| <code>value</code> | value of the parameter |
| <code>estimate</code> | 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**

| | |
|--------------------|------------------------|
| <code>id</code> | the parameter's ID |
| <code>value</code> | 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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = poisson_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

create_laplace_distr *Create a Laplace distribution*

Description

Create a Laplace distribution

Usage

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

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

Value

a Laplace distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
if (is_on_ci()) {  
  
  laplace_distr <- create_laplace_distr()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = laplace_distr  
    )  
  )  
  file.remove(beast2_input_file)
```

```

    remove_beautier_folder()
}

```

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_treelog_xml](#) to create the XML text of the loggers with the treelog 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,  
  value = NA,  
  lower = NA,  
  upper = NA  
)
```

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

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

```

if (is_on_ci()) {

  log_normal_distr <- create_log_normal_distr()

  beast2_input_file <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    tree_prior = create_yule_tree_prior(
      birth_rate_distr = log_normal_distr
    )
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}

```

create_mcmc

Create an MCMC configuration.

Description

Create an MCMC configuration, as in the BEAUti MCMC tab.

Usage

```

create_mcmc(
  chain_length = 1e+07,
  store_every = -1,
  pre_burnin = 0,
  n_init_attempts = 10,
  sample_from_prior = FALSE,
  tracelog = beautier::create_tracelog(),
  screenlog = beautier::create_screenlog(),
  treelog = beautier::create_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 |

Details

There are four things that can be saved: * `store_every`: saves the state of the MCMC to file, as a `.state.xml` file * `tracelog`: stores the trace of the state of the MCMC to file. See [create_tracelog](#) how to specify the filename * `screenlog`: stores the screen output to file. See [create_screenlog](#) how to specify the filename * `treelog`: stores the estimated phylogenies to file. See [create_treelog](#) how to specify the filename

Value

an MCMC configuration

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_test_mcmc](#) to create a short regular MCMC, that can be used for testing runs. 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

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

  beast2_input_file <- get_beautier_tempfilename()
  create_beast2_input_file(
    get_fasta_filename(),
    beast2_input_file,
    mcmc = mcmc
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}
```

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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    exp_distr, beast2_input_file
```



```

    input_filename = get_fasta_filename(),
    beast2_input_file,
    tree_prior = create_yule_tree_prior(
      birth_rate_distr = exp_distr
    )
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}

```

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

| | |
|----------------------|---|
| alignment_id | ID of the alignment, as returned by get_alignment_id . Keep at NA to have it initialized automatically |
| 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 |
| is_monophyletic | boolean to indicate monophyly is assumed in a Most Recent Common Ancestor prior, as returned by create_mrca_prior |
| mrca_distr | the distribution used by the MRCA prior. Can be NA (the default) or any distribution returned by create_distr |
| 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. |
| clock_prior_distr_id | 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

```
check_empty_beautier_folder()

fasta_filename <- get_beautier_path("anthus_aco.fas")

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

# The taxa are monophyletic
mrca_prior <- create_mrca_prior(
  taxa_names = get_taxa_names(filename = fasta_filename),
  is_monophyletic = TRUE
)

# Set the crown age to 10
mrca_prior <- create_mrca_prior(
  taxa_names = get_taxa_names(fasta_filename),
  mrca_distr = create_normal_distr(mean = 10, sigma = 0.1)
)

check_empty_beautier_folder()
```

create_mu_param

Create a parameter called mu

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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = laplace_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

create_m_param

Create a parameter called m

Description

Create a parameter called m

Usage

```
create_m_param(id = NA, estimate = FALSE, lower = NA, upper = NA, value = 0)
```

Arguments

| | |
|----------|--|
| id | the parameter's ID |
| estimate | TRUE if this parameter is to be estimated by BEAST2, FALSE otherwise |
| 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 | 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

```
if (is_on_ci()) {

  # 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 <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    tree_prior = create_yule_tree_prior(
      birth_rate_distr = log_normal_distr
    )
  )
}
```

```
    file.remove(beast2_input_file)

    remove_beautier_folder()

}
```

create_normal_distr *Create an normal distribution*

Description

Create an normal distribution

Usage

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

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

Value

a normal distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```

if (is_on_ci()) {

  normal_distr <- create_normal_distr()

  beast2_input_file <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    tree_prior = create_yule_tree_prior(
      birth_rate_distr = normal_distr
    )
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}

```

```
create_ns_inference_model
```

Create an inference model to measure the evidence of.

Description

Create an inference model to measure the evidence of. To do so, the inference model is created as usual (see [create_inference_model](#)), except for using a Nested Sampling MCMC (see [create_ns_mcmc](#))

Usage

```

create_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_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. |

Value

an inference model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_inference_model](#) to create a regular inference model. Use [create_test_ns_inference_model](#) to create an inference model to estimate the marginal likelihood with a short MCMC, to be used in testing.

Examples

```
check_empty_beautier_folder()

inference_model <- create_ns_inference_model()

check_empty_beautier_folder()
```

| | |
|----------------|---|
| 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 = beautier::create_tracelog(),
  screenlog = beautier::create_screenlog(),
  treelog = beautier::create_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

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

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



```
beast2_input_file <- get_beautier_tempfilename()
create_beast2_input_file(
  get_fasta_filename(),
  beast2_input_file,
  mcmc = mcmc
)
file.remove(beast2_input_file)

remove_beautier_folder()
}
```

create_one_div_x_distr

Create a 1/x distribution

Description

Create a 1/x distribution

Usage

```
create_one_div_x_distr(id = NA, value = NA, lower = NA, upper = NA)
```

Arguments

| | |
|-------|---|
| id | the distribution's ID |
| value | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

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

```

if (is_on_ci()) {

  one_div_x_distr <- create_one_div_x_distr()

  beast2_input_file <- get_beautier_tempfilename()
  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
    )
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}

```

create_param

General function to create a parameter.

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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = beta_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

create_poisson_distr *Create a Poisson distribution*

Description

Create a Poisson distribution

Usage

```
create_poisson_distr(id = NA, lambda = 0, value = NA, lower = NA, upper = NA)
```

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 | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| upper | an upper limit of the uniform distribution. If the upper limits needs to be infinity, set upper to Inf. |

Value

a Poisson distribution

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
if (is_on_ci()) {  
  
  poisson_distr <- create_poisson_distr()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = poisson_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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

```
if (is_on_ci()) {
  # 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 <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    site_model = create_gtr_site_model(
      rate_ac_param = rate_ac_param
    )
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}
```

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

```
if (is_on_ci()) {
  # 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 <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    site_model = create_gtr_site_model(
      rate_ag_param = rate_ag_param
    )
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}
```

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

```
if (is_on_ci()) {  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    site_model = create_gtr_site_model(  
      rate_at_param = rate_at_param  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

create_rate_categories_state_node_xml

Internal function

Description

Creates the rateCategories state node, such as: "<stateNode id=\"rateCategories.c:[id]\" spec=\"parameter.IntegerParameter\" dimension=\"[dimension]\">1 </stateNode>"

Usage

```
create_rate_categories_state_node_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.

Value

the following XML: "<stateNode id=\"rateCategories.c:[id]\" spec=\"parameter.IntegerParameter\" dimension=\"[dimension]\"> 1 </stateNode>"

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

create_rate_categories_state_node_xml(
  create_inference_model(
    clock_model = create_rln_clock_model(
      id = 314,
      dimension = 1
    )
  )
)

check_empty_beautier_folder()
```

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

```
if (is_on_ci()) {
  # 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 <- get_beautier_tempfilename()
  create_beast2_input_file(
    input_filename = get_fasta_filename(),
    beast2_input_file,
    site_model = create_gtr_site_model(
      rate_cg_param = rate_cg_param
    )
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}
```

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

```
if (is_on_ci()) {  
  
  # Create parameter  
  rate_ct_param <- create_rate_ct_param(value = 1)  
  
  # Use the parameter to create a BEAST2 input file  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    site_model = create_gtr_site_model(  
      rate_ct_param = rate_ct_param  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    site_model = create_gtr_site_model(  
      rate_gt_param = rate_gt_param  
    )  
  )  
  file.remove(beast2_input_file)  
}
```

```

    remove_beautier_folder()
}

```

```
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 <code>branchRateModel</code> , 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 <- get_beautier_tempfilename()
## Not run:
  create_beast2_input_file(
    get_fasta_filename(),
    beast2_input_file,
    clock_model = rln_clock_model
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()

## End(Not run)
```

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

```
if (is_on_ci()) {  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = laplace_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

| | |
|------------------|----------------------------------|
| 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 |

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()
create_screenlog()
check_empty_beautier_folder()
```

`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 = beautier::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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
  )  
}
```



```
        tree_prior = create_yule_tree_prior(
            birth_rate_distr = normal_distr
        )
    )
    file.remove(beast2_input_file)

    remove_beautier_folder()
}
```

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

```
if (is_on_ci()) {  
  
  check_empty_beautier_folder()  
  
  input_filename <- get_fasta_filename()  
  gtr_site_model <- create_gtr_site_model()  
  hk_site_model <- create_hky_site_model()  
  jc69_site_model <- create_jc69_site_model()  
  tn93_site_model <- create_tn93_site_model()  
  
  # Can use any site model  
  output_filename <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = input_filename,  
    output_filename = output_filename,  
    site_model = jc69_site_model  
  )  
  file.remove(output_filename)  
  
  remove_beautier_folder()  
  check_empty_beautier_folder()  
}
```

| | |
|--------------------|---|
| 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

```
check_empty_beautier_folder()

# All created site models are a kind of site model
site_models <- create_site_models()

# TRUE
is_gtr_site_model(site_models[[1]])
is_hky_site_model(site_models[[2]])
is_jc69_site_model(site_models[[3]])
is_tn93_site_model(site_models[[4]])

check_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

create_site_models_from_names(get_site_model_names())

check_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

site_model <- create_site_model_from_name(get_site_model_names()[1])

check_empty_beautier_folder()
```

`create_site_model_parameters_xml`

Internal function to creates the XML text for the parameters within the siteModel section of a BEAST2 parameter file.

Description

Internal function to creates the XML text for the parameters within the siteModel section, which is part of the siteModel section of a BEAST2 parameter file.

Usage

```
create_site_model_parameters_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.

Details

The parameters sections has these elements:

```
[parameters]
```

[parameters] can be a combination of these:

```
<parameter id="mutationRate.s[...]>  
<parameter id="gammaShape.s[...]>  
<parameter id="proportionInvariant.s[...]>
```

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()
```

create_site_model_xml *Internal function to creates the XML text for the siteModel tag of a BEAST2 parameter file.*

Description

Creates the XML text for the siteModel tag of a BEAST2 parameter file, which is part of the distribution node for the treeLikelihood ID.

Usage

```
create_site_model_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.

Details

The siteModel tag has these elements:

```
<siteModel[...]>
  [parameters]
  <substModel[...]>
    [...]
  </substModel>
</siteModel>
```

The parameter section is created by [create_site_model_parameters_xml](#) The substModel section is created by [create_subst_model_xml](#)

Value

the site model as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()
# <distribution id="posterior"[...]>
#   <distribution id="likelihood" [...]>
#     <siteModel...>
#       [parameters]
#     </siteModel>
#   </distribution>
# </distribution>

check_empty_beautier_folder()
```

```
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

```
if (is_on_ci()) {
  strict_clock_model <- create_strict_clock_model(
    clock_rate_param = 1.0,
    clock_rate_distr = create_uniform_distr()
  )

  beast2_input_file <- get_beautier_tempfilename()
  create_beast2_input_file(
    get_fasta_filename(),
    beast2_input_file,
    clock_model = strict_clock_model
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}
```

create_strict_clock_rate_scaler_operator_xml

Internal function

Description

Creates the StrictClockRateScaler operator such as: ...

Usage

```
create_strict_clock_rate_scaler_operator_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.

Value

the following XML: ...

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

create_strict_clock_rate_scaler_operator_xml(
  create_inference_model(
    clock_model = create_strict_clock_model(id = 314)
  )
)

check_empty_beautier_folder()
```

```
create_subst_model_xml
```

Internal function to create the substModel section

Description

Internal function to create the substModel section

Usage

```
create_subst_model_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.

Value

the substModel section as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

# Inference model must be initialized
inference_model <- create_inference_model(
  site_model = create_jc69_site_model(id = 123)
)
create_subst_model_xml(
```

```
    inference_model = inference_model
  )
  check_empty_beautier_folder()
```

| | |
|----------------|------------------------------------|
| 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

```
if (is_on_ci()) {  
  
  # 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 <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = log_normal_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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()
```

Value

a filename for a temporary 'screenlog' file

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
check_empty_beautier_folder()

create_temp_screenlog_filename()

check_empty_beautier_folder()
```

```
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()
```

Value

a filename for a temporary 'tracelog' file

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
check_empty_beautier_folder()

create_temp_tracelog_filename()

check_empty_beautier_folder()
```

`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()
```

Value

a filename for a temporary 'treelog' file

Author(s)

Richèl J.C. Bilderbeek

See Also

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

Examples

```
check_empty_beautier_folder()
create_temp_treelog_filename()
check_empty_beautier_folder()
```

`create_test_inference_model`

Create a testing inference model.

Description

Creates a simple inference model with a short MCMC chain, to be used in testing.

Usage

```
create_test_inference_model(
  site_model = beautier::create_jc69_site_model(),
  clock_model = beautier::create_strict_clock_model(),
  tree_prior = beautier::create_yule_tree_prior(),
  mrca_prior = NA,
  mcmc = beautier::create_test_mcmc(),
  beauti_options = beautier::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 a regular inference model. Use [create_test_ns_inference_model](#) to create an inference model to estimate the marginal likelihood with a short MCMC, to be used in testing

Examples

```
if (is_on_ci()) {
  check_empty_beautier_folder()

  inference_model <- create_test_inference_model()
```

```

beast2_input_file <- get_beautier_tempfilename()
create_beast2_input_file_from_model(
  get_fasta_filename(),
  beast2_input_file,
  inference_model = inference_model
)
file.remove(beast2_input_file)

remove_beautier_folder()
check_empty_beautier_folder()
}

```

create_test_mcmc *Create an MCMC configuration for testing.*

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 = beautier::create_test_tracelog(),
  screenlog = beautier::create_test_screenlog(),
  treelog = beautier::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

```
if (is_on_ci()) {  
  
  # Create an MCMC chain with 50 states  
  mcmc <- create_test_mcmc()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    get_fasta_filename(),  
    beast2_input_file,  
    mcmc = mcmc  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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. |

Value

an inference model

Author(s)

Richèl J.C. Bilderbeek

See Also

Use [create_test_inference_model](#) to create a regular inference model with a short MCMC, to be used in testing. Use [create_ns_inference_model](#) to create an inference model to estimate the marginal likelihood.

Examples

```
check_empty_beautier_folder()

inference_model <- create_test_ns_inference_model()

check_empty_beautier_folder()
```

```
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

```

if (is_on_ci()) {

  mcmc <- create_test_ns_mcmc()
  beast2_input_file <- get_beautier_tempfilename()
  create_beast2_input_file(
    get_fasta_filename(),
    beast2_input_file,
    mcmc = mcmc
  )
  file.remove(beast2_input_file)

  remove_beautier_folder()
}

```

create_test_screenlog *Create a screenlog object*

Description

Create a screenlog object

Usage

```
create_test_screenlog(  
  filename = beautier::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 <code>TRUE</code> 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

```
if (is_on_ci()) {  
  
  tn93_site_model <- create_tn93_site_model(  
    kappa_1_param = 2.0,  
    kappa_2_param = 2.0  
  )  
  
  output_filename <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    output_filename = output_filename,  
    site_model = tn93_site_model  
  )  
  file.remove(output_filename)  
  
  remove_beautier_folder()  
}
```

create_tn93_subst_model_xml

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

```
create_tn93_subst_model_xml(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

| | |
|-----------------|---------------------------------|
| 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>Internal function</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 |

Value

a 'treelog', as can be checked by [check_treelog](#)

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()  
  
create_treelog()  
  
check_empty_beautier_folder()
```

```

create_treelog_xml    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" sanitiseHeaders="true" sort="smart"> # nolint in-
                      deed 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" 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> ““

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_likelihood_distr_xml    Creates the XML text for the distribution tag with the
                                    treeLikelihood ID, of a BEAST2 parameter file.

```

Description

Creates the XML text for the distribution tag with the treeLikelihood ID, of a BEAST2 parameter file, in an unindented form

Usage

```
create_tree_likelihood_distr_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.

Details

The distribution tag (with ID equals treeLikelihood) has these elements:

```
<distribution id="treeLikelihood" [...]>
  <siteModel [...]>
    [...]
  </siteModel>
  <branchRateModel [...]>
    [...]
  </branchRateModel>
</distribution>
```

The siteModel section is created by [create_site_model_xml](#). The branchRateModel section is created by [create_branch_rate_model_xml](#).

Zooming out:

```
<beast [...]>
  <run [...]>
    <distribution id="posterior" [...]>
      <distribution id="likelihood" [...]>
        [this section]
      </distribution>
    </distribution>
  </run>
</beast>
```

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

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

```
if (is_on_ci()) {  
  
  check_empty_beautier_folder()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  
  bd_tree_prior <- create_bd_tree_prior()  
  cbs_tree_prior <- create_cbs_tree_prior()  
  ccp_tree_prior <- create_ccp_tree_prior()  
  cep_tree_prior <- create_cep_tree_prior()  
  yule_tree_prior <- create_yule_tree_prior()  
}
```

```

# Use any of the above tree priors
create_beast2_input_file(
  input_filename = get_fasta_filename(),
  beast2_input_file,
  tree_prior = bd_tree_prior
)
file.remove(beast2_input_file)

remove_beautier_folder()
check_empty_beautier_folder()
}

```

| | |
|--------------------|---|
| 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

```

check_empty_beautier_folder()

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

check_empty_beautier_folder()

```

`create_uclid_mean_state_node_param_xml`*Internal function*

Description

Creates the `uclidMean.c` parameter with the name `stateNode`, such as: `<parameter id="uclidMean.c:[id]" spec="parameter.RealParameter" name="stateNode">1.0</parameter>`

Usage

```
create_uclid_mean_state_node_param_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.

Value

the XML `<parameter id="uclidMean.c:[id]" spec="parameter.RealParameter" name="stateNode">1.0</parameter>`

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

create_uclid_mean_state_node_param_xml(
  create_inference_model(
    clock_model = create_rln_clock_model(id = 314),
    beauti_options = create_beauti_options_v2_6()
  )
)

check_empty_beautier_folder()
```

```
create_ucld_stdev_state_node_param_xml
```

Internal function

Description

Creates the ucldStdev parameter with the name stateNode, such as: `<parameter id="ucldStdev.c:[id]" [...] name="stateNode">0.1</parameter>`

Usage

```
create_ucld_stdev_state_node_param_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.

Value

the following XML: `<parameter id="ucldStdev.c:[id]" lower="0.0" name="stateNode"> 0.1 </parameter>`

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

create_ucld_stdev_state_node_param_xml(
  create_inference_model(
    clock_model = create_rln_clock_model(id = 314)
  )
)

check_empty_beautier_folder()
```

create_uniform_distr *Create a uniform distribution*

Description

Create a uniform distribution

Usage

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

Arguments

| | |
|-------|---|
| id | the distribution's ID |
| value | the initial value for the MCMC |
| lower | the lower bound, the lowest possible value |
| 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

```
if (is_on_ci()) {  
  
  uniform_distr <- create_uniform_distr()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = create_yule_tree_prior(  
      birth_rate_distr = uniform_distr  
    )  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

`create_xml_declaration`*Create the XML declaration of the BEAST2 XML input file*

Description

Create the XML declaration of the BEAST2 XML input file

Usage

```
create_xml_declaration()
```

Value

one line of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()
```

```
create_xml_declaration()
```

```
check_empty_beautier_folder()
```

`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

```
if (is_on_ci()) {  
  
  yule_tree_prior <- create_yule_tree_prior()  
  
  beast2_input_file <- get_beautier_tempfilename()  
  create_beast2_input_file(  
    input_filename = get_fasta_filename(),  
    beast2_input_file,  
    tree_prior = yule_tree_prior  
  )  
  file.remove(beast2_input_file)  
  
  remove_beautier_folder()  
}
```

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,
  allow_empty_str,
  allow_na,
  alpha_parameter,
  bd_tree_prior,
  beautier_folder,
  cbs_tree_prior,
  beast2_version,
  beauti_options,
  beta_parameter,
  ccp_tree_prior,
  cep_tree_prior,
  chain_length,
  clock_model,
```

clock_model_name,
clock_model_names,
clock_models,
clock_prior_distr_id,
clock_rate_param,
crown_age,
crown_ages,
distr_id,
fasta_filename,
fasta_filenames,
filename,
fixed_crown_age,
fixed_crown_ages,
gamma_distr,
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,
m_param,
mcmc,
mode,
mrca_prior,
mrca_priors,
mrca_prior_name,
n_init_attempts,
output_filename,
param,
param_id,
phylogeny,
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,
    trace_log,
    tree_log,
    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

`allow_empty_str` allow a string to be empty

`allow_na` allow NA

`alpha_parameter` an alpha parameter, as created by [create_alpha_param](#)

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

`beautier_folder` the path to the [beautier](#) temporary files folder

`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](#)

`beta_parameter` a beta parameter, as created by [create_beta_param](#)

`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 |
| clock_rate_param | a <code>clockRate</code> parameter, a numeric value, as created by create_clock_rate_param |
| 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. Note that BEAST2 also supports missing data, by using a dash (-) or question mark (?) as a sequence. |
| fasta_filenames | One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename. |
| filename | a filename, as can be checked by check_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_distr | a gamma distribution, as created by create_gamma_distr |
| 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 from the ape package |
| input_filename | A FASTA filename. Use get_fasta_filename to obtain a testing FASTA filename. |
| input_filenames | One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename. |
| is_monophyletic | boolean to indicate monophyly is assumed in a Most Recent Common Ancestor prior, as returned by create_mrca_prior |
| jc69_site_model | a JC69 site model, as returned by create_jc69_site_model |
| log_every | number of MCMC states between writing to file |
| m_param | an m parameter, as created by create_m_param |
| 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. |
| mode | mode how to log. Valid values are the ones returned by get_log_modes |
| mrca_prior | a Most Recent Common Ancestor prior, as returned by create_mrca_prior |
| mrca_priors | a list of one or more Most Recent Common Ancestor priors, as returned by create_mrca_prior |
| 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. |
| n_init_attempts | number of initialization attempts before failing |
| output_filename | Name of the XML parameter file created by this function. BEAST2 uses this file as input. |
| param | a parameter, as can be created by create_param . |
| param_id | a parameter's ID |
| phylogeny | a phylogeny of type phylo from the ape package |
| pre_burnin | number of burn in samples taken before entering the main loop |
| 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`

`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>Internal function</i> |
|--------------|--------------------------|

Description

Converts a distribution to XML

Usage

```
distr_to_xml(distr, beauti_options = create_beauti_options())
```

Arguments

distr a distribution, as created by [create_distr](#))
 beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

distr_to_xml(create_uniform_distr(id = 1))

check_empty_beautier_folder()
```

distr_to_xml_beta *Internal function*

Description

Converts a beta distribution to XML

Usage

```
distr_to_xml_beta(distr, beauti_options = create_beauti_options())
```

Arguments

distr a beta distribution, as created by [create_beta_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_exp *Internal function*

Description

Converts an exponential distribution to XML

Usage

```
distr_to_xml_exp(distr, beauti_options = create_beauti_options())
```

Arguments

distr an exponential distribution, as created by [create_exp_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_inv_gamma

Internal function

Description

Converts an inverse-gamma distribution to XML

Usage

```
distr_to_xml_inv_gamma(distr, beauti_options = create_beauti_options())
```

Arguments

distr an inverse-gamma distribution, as created by [create_inv_gamma_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_laplace *Internal function*

Description

Converts a Laplace distribution to XML

Usage

```
distr_to_xml_laplace(distr, beauti_options = create_beauti_options())
```

Arguments

distr a Laplace distribution as created by [create_laplace_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_log_normal

Internal function

Description

Converts a log-normal distribution to XML

Usage

```
distr_to_xml_log_normal(distr, beauti_options = create_beauti_options())
```

Arguments

distr a log-normal distribution, as created by [create_log_normal_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_normal *Internal function*

Description

Converts a normal distribution to XML

Usage

```
distr_to_xml_normal(distr, beauti_options = create_beauti_options())
```

Arguments

distr a normal distribution, as created by [create_normal_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_one_div_x

Internal function

Description

Converts a $1/x$ distribution to XML

Usage

```
distr_to_xml_one_div_x(distr, beauti_options = create_beauti_options())
```

Arguments

distr a $1/x$ distribution, as created by [create_one_div_x_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_poisson *Internal function*

Description

Converts a Poisson distribution to XML

Usage

```
distr_to_xml_poisson(distr, beauti_options = create_beauti_options())
```

Arguments

distr a Poisson distribution, as created by [create_poisson_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

distr_to_xml_uniform *Internal function*

Description

Converts a uniform distribution to XML

Usage

```
distr_to_xml_uniform(distr, beauti_options = create_beauti_options())
```

Arguments

distr a uniform distribution, as created by [create_uniform_distr](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

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

lines lines of the XML text
section 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

Examples

```
check_empty_beautier_folder()
```

```
fasta_file_to_sequences(fasta_filename = get_fasta_filename())
```

```
check_empty_beautier_folder()
```

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

| | |
|--------------------|-------------------|
| <code>lines</code> | lines of text |
| <code>regex</code> | 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

| | |
|---------|-----------------------------|
| 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

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

| | |
|-------|---------------|
| lines | lines of text |
| regex | 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

| | |
|----------------------|-----------------------------|
| <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

`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

| | |
|-------------------------------|-------------------------|
| <code>freq_equilibrium</code> | a freq_equilibrium name |
| <code>id</code> | a site model's name |

Value

the freq_equilibrium as XML

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

freq_equilibrium_to_xml(freq_equilibrium = "estimated", id = "my_id")

check_empty_beautier_folder()
```

gamma_distr_to_xml *Internal function*

Description

Converts a gamma distribution to XML

Usage

```
gamma_distr_to_xml(gamma_distr, beauti_options = create_beauti_options())
```

Arguments

`gamma_distr` a gamma distribution, as created by [create_gamma_distr](#))
`beauti_options` one BEAUti options object, as returned by [create_beauti_options](#)

Value

the distribution as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

# gamma distribution must be initialized
gamma_distr_to_xml(
  gamma_distr = create_gamma_distr(
    id = "0",
    alpha = create_alpha_param(id = "2", value = "0.5396"),
    beta = create_beta_param(id = "3", value = "0.3819")
  )
)

check_empty_beautier_folder()
```

`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. Note that BEAST2 also supports missing data, by using a dash (-) or question mark (?) as a sequence. |
| 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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

get_alignment_ids(
    get_beautier_paths(c("anthus_aco.fas", "anthus_nd2.fas"))
)

check_empty_beautier_folder()
```

`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 kinds of files. Use [are_fasta_filenames](#) to see if the filenames all have a common FASTA filename extension.

Examples

```
check_empty_beautier_folder()

get_alignment_ids_from_fasta_filenames(
    get_beautier_paths(c("anthus_aco.fas", "anthus_nd2.fas"))
)

check_empty_beautier_folder()
```

`get_beautier_folder` *Get the path to the **beautier** temporary files folder*

Description

Get the path to the **beautier** temporary files folder

Usage

```
get_beautier_folder()
```

Value

the path to the **beautier** temporary files folder

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

get_beautier_folder()

check_empty_beautier_folder()
```

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

Description

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

Usage

```
get_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

```
check_empty_beautier_folder()

get_beautier_path("test_output_0.fas")
get_beautier_path("anthus_aco.fas")
get_beautier_path("anthus_nd2.fas")

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

get_beautier_paths(
  c("test_output_0.fas", "anthus_aco.fas", "anthus_nd2.fas")
)

check_empty_beautier_folder()
```

get_beautier_tempfilename

Get a temporary filename

Description

Get a temporary filename, similar to [tempfile](#), except that it always writes to a temporary folder named [beautier](#).

Usage

```
get_beautier_tempfilename(pattern = "file", fileext = "")
```

Arguments

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

Value

name for a temporary file

Note

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

get_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

```
check_empty_beautier_folder()

# StrictClock
get_clock_model_name(create_strict_clock_model())

# RelaxedClock
get_clock_model_name(create_rln_clock_model())

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

get_clock_model_names()

check_empty_beautier_folder()
```

get_crown_age *Obtain the crown age of a phylogeny.*

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

```
check_empty_beautier_folder()

phylogeny <- ape::read.tree(text = "(a:15,b:15):1;")
get_crown_age(phylogeny = phylogeny)

check_empty_beautier_folder()
```

get_distr_names *Get the distribution names*

Description

Get the distribution names

Usage

```
get_distr_names()
```

Value

the distribution names

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

get_distr_names()

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

get_distr_n_params(create_beta_distr())
get_distr_n_params(create_exp_distr())
get_distr_n_params(create_gamma_distr())
get_distr_n_params(create_inv_gamma_distr())
get_distr_n_params(create_laplace_distr())
get_distr_n_params(create_log_normal_distr())
get_distr_n_params(create_normal_distr())
get_distr_n_params(create_one_div_x_distr())
get_distr_n_params(create_poisson_distr())
get_distr_n_params(create_uniform_distr())

check_empty_beautier_folder()
```

get_fasta_filename *Get the path of a FASTA file used in testing*

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

```
if (is_on_ci()) {  
  
  input_filename <- get_fasta_filename()  
  output_filename <- get_beautier_tempfilename()  
  
  create_beast2_input_file(  
    input_filename = input_filename,  
    output_filename = output_filename  
  )  
  file.remove(output_filename)  
  
  remove_beautier_folder()  
}
```

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

```
check_empty_beautier_folder()

# Path need not exist, use UNIX path as example
# test
get_file_base_sans_ext("/home/homer/test.txt")

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

get_freq_equilibrium_names()

check_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

# zero distributions
gamma_site_model <- create_gamma_site_model()
get_gamma_site_model_n_distrs(
  gamma_site_model
)

gamma_site_model <- create_gamma_site_model(
  gamma_cat_count = 2,
  gamma_shape_prior_distr = create_exp_distr()
```

```
)  
  
# one distribution  
get_gamma_site_model_n_distrs(gamma_site_model)  
  
check_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()  
  
# 0 parameters  
get_gamma_site_model_n_params(  
  create_gamma_site_model(gamma_cat_count = 0)  
)  
  
# 0 parameters  
get_gamma_site_model_n_params(  
  create_gamma_site_model(gamma_cat_count = 1)  
)  
  
# 1 parameter  
get_gamma_site_model_n_params(  
  create_gamma_site_model(  
    gamma_cat_count = 2,
```

```

    gamma_shape_prior_distr = create_exp_distr()
  )
)

check_empty_beautier_folder()

```

```
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

```
check_empty_beautier_folder()

inference_model <- create_inference_model()
filenames <- get_inference_model_filenames(inference_model)

check_empty_beautier_folder()
```

get_log_modes *Get the possible log modes*

Description

Get the possible log modes

Usage

```
get_log_modes()
```

get_log_sorts *Get the possible log sorts*

Description

Get the possible log sorts

Usage

```
get_log_sorts()
```

get_mcmc_filenames *Get the filenames stored in an MCMC.*

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

```
check_empty_beautier_folder()

mcmc <- create_mcmc()
mcmc$tracelog$filename <- "/home/john/trace.log"
mcmc$screenlog$filename <- "/home/john/screen.log"
mcmc$treelog$filename <- "/home/john/tree.log"

# 3 filenames
filenames <- get_mcmc_filenames(mcmc)

# If there is no need to write to the screenlog file ...
mcmc$screenlog$filename <- ""

# 2 filenames
# ... one file less will be created
filenames <- get_mcmc_filenames(mcmc)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

fasta_filename <- get_beautier_path("test_output_5.fas")
# 5
get_n_taxa(fasta_filename)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# BirthDeath
get_operator_id_pre(
  tree_prior = create_bd_tree_prior()
)

check_empty_beautier_folder()
```

| | |
|-----------------|--------------------------------|
| get_param_names | <i>Get the parameter names</i> |
|-----------------|--------------------------------|

Description

Get the parameter names

Usage

```
get_param_names()
```

Value

the parameter names

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

names <- get_param_names()

check_empty_beautier_folder()
```

get_remove_dir_fun *Get a function that, from a filename, returns the part without the directory.*

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

```
check_empty_beautier_folder()

f <- get_remove_hex_fun()
# /home/john/beast2.xml.state
f("/home/john/beast2_186c7404208c.xml.state")

# beast2.xml.state
f("beast2_186c7404208c.xml.state")

# NA
f(NA)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# 5
get_site_models_n_distrs(list(create_gtr_site_model()))
# 1
get_site_models_n_distrs(list(create_hky_site_model()))
# 0
get_site_models_n_distrs(list(create_jc69_site_model()))
# 2
get_site_models_n_distrs(list(create_tn93_site_model()))

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# Ten
get_site_models_n_params(list(create_gtr_site_model()))

# Two
get_site_models_n_params(list(create_hky_site_model()))

# Zero
get_site_models_n_params(list(create_jc69_site_model()))

# Four
get_site_models_n_params(list(create_tn93_site_model()))

check_empty_beautier_folder()
```

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_empty_beautier_folder()

get_site_model_names()

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

# 5: rates AC, AG, AT, CG and GT
get_site_model_n_distrs(create_gtr_site_model())

# 1: kappa
get_site_model_n_distrs(create_hky_site_model())

# 0: npne
get_site_model_n_distrs(create_jc69_site_model())

# 2: kappa 1 and kappa 2
get_site_model_n_distrs(create_tn93_site_model())

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

# Ten
get_site_model_n_params(create_gtr_site_model())

# Two
get_site_model_n_params(create_hky_site_model())

# Zero
get_site_model_n_params(create_jc69_site_model())

# Four
get_site_model_n_params(create_tn93_site_model())

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

get_taxa_names(get_beautier_path("anthus_aco_sub.fas"))

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# Three distributions
get_tree_priors_n_distrs(
  list(
    create_bd_tree_prior(), # has two distributions
    create_ccp_tree_prior() # has one distribution
  )
)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# Two
get_tree_priors_n_params(
  list(
    create_bd_tree_prior(), # zero
    create_cep_tree_prior() # two
  )
)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

get_tree_prior_names()

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

# 2: birth_rate_distr and death_rate_distr
get_tree_prior_n_distrs(create_bd_tree_prior())

# 0: none
get_tree_prior_n_distrs(create_cbs_tree_prior())

# 1: pop_size_distr
get_tree_prior_n_distrs(create_ccp_tree_prior())

# 2: pop_size_distr and growth_rate_distr
get_tree_prior_n_distrs(create_cep_tree_prior())

# 1: birth_rate_distr
get_tree_prior_n_distrs(create_yule_tree_prior())

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

# birth_rate_distr is uniform, which has zero parameters
# death_rate_distr is uniform, which has zero parameters
get_tree_prior_n_params(create_bd_tree_prior())

# no distributions, no parameters
get_tree_prior_n_params(create_cbs_tree_prior())

# pop_size_distr is 1/x, which has zero parameters
get_tree_prior_n_params(create_ccp_tree_prior())

# pop_size_distr is 1/x, which has zero parameters
# growth_rate_distr is Laplace, which has two parameters
get_tree_prior_n_params(create_cep_tree_prior())

# birth_rate_distr is uniform, which has zero parameters
get_tree_prior_n_params(create_yule_tree_prior())

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# my_tag
get_xml_closing_tag("<my_tag text=something></my_tag>")

# Will return NA
get_xml_closing_tag("<my_tag text=something/>")
get_xml_closing_tag("no_xml")

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# my_tag
get_xml_opening_tag("<my_tag text=something/>")

# NA when there is no opening tag
get_xml_opening_tag("no_xml")

check_empty_beautier_folder()
```

has_mrca_prior *Determines if the inference model has an MRCA prior.*

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

```
check_empty_beautier_folder()

# No MRCA prior
inference_model <- create_inference_model(
  mrca_prior = NA
)
has_mrca_prior(inference_model) # Returns FALSE

# A default MRCA prior
inference_model <- create_inference_model(
  mrca_prior = create_mrca_prior()
)
has_mrca_prior(inference_model) # Returns TRUE

check_empty_beautier_folder()
```

```
has_mrca_prior_with_distr
```

See if the inference model has one MRCA prior with a distribution

Description

See if the inference model has one MRCA prior with a distribution

Usage

```
has_mrca_prior_with_distr(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 one MRCA prior with a distribution, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

has_rln_clock_model *Determine if the inference_model uses a relaxed log-normal clock model.*

Description

Determine if the inference_model uses a relaxed log-normal clock model.

Usage

```
has_rln_clock_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

TRUE if the inference_model uses a relaxed log-normal clock model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

# Yes, has a RLN clock model
has_rln_clock_model(
  create_inference_model(clock_model = create_rln_clock_model())
)

# No RLN clock model
has_rln_clock_model(
  create_inference_model(clock_model = create_strict_clock_model())
)
```

```
check_empty_beautier_folder()
```

```
has_strict_clock_model
```

Determine if the inference_model uses a strict clock model.

Description

Determine if the inference_model uses a strict clock model

Usage

```
has_strict_clock_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

TRUE if the inference_model uses a strict clock model, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

# Yes, has a strict clock model
has_strict_clock_model(
  create_inference_model(clock_model = create_strict_clock_model())
)

# No strict clock model
has_strict_clock_model(
  create_inference_model(clock_model = create_rln_clock_model())
)

check_empty_beautier_folder()
```

| | |
|----------------|--|
| has_tip_dating | <i>Determine if the inference_model uses tip dating.</i> |
|----------------|--|

Description

Determine if the inference_model uses tip dating

Usage

```
has_tip_dating(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 uses tip dating, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

# Yes, has tip dating
has_strict_clock_model(
  create_inference_model(
    tipdates_filename = get_beautier_path("test_output_0_tipdates.tsv")
  )
)

# No tip dating
has_strict_clock_model(
  create_inference_model()
)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# TRUE
has_xml_short_closing_tag("<my_tag id=1/>")
# FALSE
has_xml_short_closing_tag("<my_tag id=1>text</my_tag>")

check_empty_beautier_folder()
```

| | |
|--------|---|
| indent | <i>Indent text for a certain number of spaces. If the text is only whitespace, leave it as such</i> |
|--------|---|

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 | <i>Initializes a Birth-Death tree prior</i> |
|--------------------|---|

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 *Initializes a beta distribution*

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 *Initializes a Coalescent Constant Population tree prior*

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

cep_tree_prior a Coalescent Exponential Population tree prior, as returned by [create_cep_tree_prior](#)
distr_id a distributions' ID
param_id 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

| | |
|-----------------|---|
| fasta_filenames | One or more FASTA filenames. Use get_fasta_filename to obtain a testing FASTA filename. |
| clock_models | a list of one or more clock models, as returned by create_clock_model |
| distr_id | the first distributions' ID |
| param_id | 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 | <i>Initializes a gamma distribution</i> |
|------------------|---|

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

| | |
|-------------------------------|---|
| <code>gamma_site_model</code> | a site model's gamma site model, as returned by create_gamma_site_model |
| <code>distr_id</code> | the first distributions' ID |
| <code>param_id</code> | the first parameter's ID |

Value

an initialized gamma site model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

gamma_site_model <- create_gamma_site_model(
  gamma_cat_count = 2,
  gamma_shape_prior_distr = create_one_div_x_distr(id = NA)
)
# FALSE: not yet initialized
is_init_gamma_site_model(gamma_site_model)
gamma_site_model <- init_gamma_site_model(gamma_site_model)
# TRUE: now it is initialized
is_init_gamma_site_model(gamma_site_model)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

gtr_site_model <- create_gtr_site_model()
# FALSE
is_init_gtr_site_model(gtr_site_model)
gtr_site_model <- init_gtr_site_model(gtr_site_model)
# TRUE
is_init_gtr_site_model(gtr_site_model)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

hky_site_model <- create_hky_site_model()
is_init_hky_site_model(hky_site_model)
hky_site_model <- init_hky_site_model(hky_site_model)
is_init_hky_site_model(hky_site_model)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

hky_site_model <- create_hky_site_model()
is_init_hky_site_model(hky_site_model)
hky_site_model <- init_hky_site_model(hky_site_model)
is_init_hky_site_model(hky_site_model)

check_empty_beautier_folder()
```

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.

init_mrca_priors *Initializes all MRCA priors*

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

| | |
|-------------------------|--------------------------------|
| <code>init_param</code> | <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 *Initializes an Poisson distribution*

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

```
check_empty_beautier_folder()

rln_clock_model <- create_rln_clock_model()
# FALSE: not yet initialized
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
# TRUE: now it is initialized
is_init_rln_clock_model(rln_clock_model)

check_empty_beautier_folder()
```

| | |
|------------------|------------------------------------|
| init_site_models | <i>Initializes all site models</i> |
|------------------|------------------------------------|

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

| | |
|---------------------------------|--|
| <code>strict_clock_model</code> | a strict clock model, as returned by create_strict_clock_model |
| <code>distr_id</code> | a distributions' ID |
| <code>param_id</code> | a parameter's ID |

Value

an initialized strict clock model

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

strict_clock_model <- create_strict_clock_model()
# FALSE: not yet initialized
is_init_strict_clock_model(strict_clock_model)
strict_clock_model <- init_strict_clock_model(strict_clock_model)
# TRUE: initialized
is_init_strict_clock_model(strict_clock_model)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

tn93_site_model <- create_tn93_site_model()
is_init_tn93_site_model(tn93_site_model)
tn93_site_model <- init_tn93_site_model(tn93_site_model)
is_init_tn93_site_model(tn93_site_model)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

is_alpha_param(create_alpha_param())
is_alpha_param(create_beta_param())
is_alpha_param(create_clock_rate_param())
is_alpha_param(create_kappa_1_param())
is_alpha_param(create_kappa_2_param())
is_alpha_param(create_lambda_param())
is_alpha_param(create_m_param())
is_alpha_param(create_mean_param())
is_alpha_param(create_mu_param())
is_alpha_param(create_rate_ac_param())
is_alpha_param(create_rate_ag_param())
is_alpha_param(create_rate_at_param())
is_alpha_param(create_rate_cg_param())
is_alpha_param(create_rate_ct_param())
is_alpha_param(create_rate_gt_param())
```



```
is_alpha_param(create_s_param())
is_alpha_param(create_scale_param())
is_alpha_param(create_sigma_param())

is_alpha_param(NA)
is_alpha_param(NULL)
is_alpha_param("nonsense")
is_alpha_param(create_jc69_site_model())
is_alpha_param(create_strict_clock_model())
is_alpha_param(create_yule_tree_prior())
is_alpha_param(create_mcmc())

check_empty_beautier_folder()
```

| | |
|------------------|--|
| 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

```
check_empty_beautier_folder()

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()))
```

```
check_empty_beautier_folder()
```

is_beauti_options *Determine if the object is a valid beauti_options*

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

```
check_empty_beautier_folder()

# TRUE
is_beauti_options(create_beauti_options())

# FALSE
is_beauti_options("nonsense")
is_beauti_options(NA)
is_beauti_options(NULL)
is_beauti_options("")
is_beauti_options(c())

check_empty_beautier_folder()
```

| | |
|---------------|--|
| 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

```
check_empty_beautier_folder()

# TRUE
is_beta_distr(create_beta_distr())
# FALSE
is_beta_distr(create_exp_distr())
is_beta_distr(NA)
is_beta_distr(NULL)
is_beta_distr("nonsense")

check_empty_beautier_folder()
```

| | |
|---------------|--|
| 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

```
check_empty_beautier_folder()

is_beta_param(create_alpha_param())
is_beta_param(create_beta_param())
is_beta_param(create_clock_rate_param())
is_beta_param(create_kappa_1_param())
is_beta_param(create_kappa_2_param())
is_beta_param(create_lambda_param())
is_beta_param(create_m_param())
is_beta_param(create_mean_param())
is_beta_param(create_mu_param())
is_beta_param(create_rate_ac_param())
is_beta_param(create_rate_ag_param())
is_beta_param(create_rate_at_param())
is_beta_param(create_rate_cg_param())
is_beta_param(create_rate_ct_param())
is_beta_param(create_rate_gt_param())
is_beta_param(create_s_param())
is_beta_param(create_scale_param())
is_beta_param(create_sigma_param())

is_beta_param(NA)
is_beta_param(NULL)
is_beta_param("nonsense")
is_beta_param(create_jc69_site_model())
```

```
is_beta_param(create_strict_clock_model())
is_beta_param(create_yule_tree_prior())
is_beta_param(create_mcmc())

check_empty_beautier_folder()
```

is_cbs_tree_prior *Determine if the object is a valid constant coalescent Bayesian skyline prior*

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

```
check_empty_beautier_folder()

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()))

check_empty_beautier_folder()
```

| | |
|-------------------|--|
| 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

```
check_empty_beautier_folder()
  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()))
check_empty_beautier_folder()
```

| | |
|-------------------|--|
| 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

```
check_empty_beautier_folder()

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()))

check_empty_beautier_folder()
```

| | |
|----------------|---|
| 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

```
check_empty_beautier_folder()

# TRUE
is_clock_model(create_strict_clock_model())
is_clock_model(create_rln_clock_model())

# FALSE
is_clock_model(NA)
is_clock_model(NULL)
is_clock_model("nonsense")
is_clock_model(create_jc69_site_model())
is_clock_model(create_mcmc())

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# TRUE
is_clock_model_name("relaxed_log_normal")
is_clock_model_name("strict")

check_empty_beautier_folder()
```

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

```

check_empty_beautier_folder()

is_clock_rate_param(create_alpha_param())
is_clock_rate_param(create_beta_param())
is_clock_rate_param(create_clock_rate_param())
is_clock_rate_param(create_kappa_1_param())
is_clock_rate_param(create_kappa_2_param())
is_clock_rate_param(create_lambda_param())
is_clock_rate_param(create_m_param())
is_clock_rate_param(create_mean_param())
is_clock_rate_param(create_mu_param())
is_clock_rate_param(create_rate_ac_param())
is_clock_rate_param(create_rate_ag_param())
is_clock_rate_param(create_rate_at_param())
is_clock_rate_param(create_rate_cg_param())
is_clock_rate_param(create_rate_ct_param())
is_clock_rate_param(create_rate_gt_param())
is_clock_rate_param(create_s_param())
is_clock_rate_param(create_scale_param())
is_clock_rate_param(create_sigma_param())

is_clock_rate_param(NA)
is_clock_rate_param(NULL)
is_clock_rate_param("nonsense")
is_clock_rate_param(create_jc69_site_model())
is_clock_rate_param(create_strict_clock_model())
is_clock_rate_param(create_yule_tree_prior())
is_clock_rate_param(create_mcmc())

check_empty_beautier_folder()

```

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

```
if (is_on_ci()) {  
  check_empty_beautier_folder()  
  
  # TRUE: An MCMC created by 'create_mcmc' is default.  
  is_default_mcmc(create_mcmc())  
  
  # FALSE: An MCMC created by 'create_ns_mcmc' is not  
  is_default_mcmc(create_ns_mcmc())  
  
  check_empty_beautier_folder()  
}
```

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

```
check_empty_beautier_folder()

# TRUE
is_distr(create_beta_distr())
is_distr(create_exp_distr())
is_distr(create_gamma_distr())
is_distr(create_inv_gamma_distr())
is_distr(create_laplace_distr())
is_distr(create_log_normal_distr())
is_distr(create_normal_distr())
is_distr(create_one_div_x_distr())
is_distr(create_poisson_distr())
is_distr(create_uniform_distr())

# FALSE
is_distr(NA)
is_distr(NULL)
is_distr("nonsense")

check_empty_beautier_folder()
```

| | |
|---------------|--|
| 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

```
check_empty_beautier_folder()

# TRUE
is_distr_name("uniform")
is_distr_name("normal")
is_distr_name("one_div_x")
is_distr_name("log_normal")
is_distr_name("exponential")
is_distr_name("gamma")
is_distr_name("beta")
is_distr_name("laplace")
is_distr_name("inv_gamma")
is_distr_name("poisson")
# FALSE
is_distr_name("nonsense")

check_empty_beautier_folder()
```

| | |
|--------------|---|
| 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

```
check_empty_beautier_folder()

# TRUE
is_exp_distr(create_exp_distr())
# FALSE
is_exp_distr(create_gamma_distr())
is_exp_distr(NA)
is_exp_distr(NULL)
is_exp_distr("nonsense")

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# TRUE
is_freq_equilibrium_name("estimated")
is_freq_equilibrium_name("empirical")
is_freq_equilibrium_name("all_equal")
# FALSE
is_freq_equilibrium_name("nonsense")

check_empty_beautier_folder()
```

| | |
|----------------|--|
| 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

```
check_empty_beautier_folder()

# TRUE
is_gamma_distr(create_gamma_distr())
# FALSE
is_gamma_distr(create_inv_gamma_distr())
is_gamma_distr(NA)
```

```
is_gamma_distr(NULL)
is_gamma_distr("nonsense")

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# TRUE
is_gamma_site_model(create_gamma_site_model())

# FALSE
is_gamma_site_model("nonsense")
is_gamma_site_model(NA)
is_gamma_site_model(NULL)
is_gamma_site_model("")
is_gamma_site_model(c())

check_empty_beautier_folder()
```

| | |
|-------------------|--|
| 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

```
check_empty_beautier_folder()

# site models
is_gtr_site_model(create_gtr_site_model())
is_gtr_site_model(create_hky_site_model())
is_gtr_site_model(create_jc69_site_model())
is_gtr_site_model(create_tn93_site_model())

# other models
is_gtr_site_model(NA)
is_gtr_site_model(NULL)
is_gtr_site_model("nonsense")
is_gtr_site_model(create_strict_clock_model())
is_gtr_site_model(create_bd_tree_prior())
is_gtr_site_model(create_mcmc())

check_empty_beautier_folder()
```

| | |
|-------------------|---|
| is_hky_site_model | <i>Determine if the object is a valid HKY site model, as created by create_hky_site_model</i> |
|-------------------|---|

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

```
check_empty_beautier_folder()

# site models
is_hky_site_model(create_hky_site_model())
is_hky_site_model(create_gtr_site_model())
is_hky_site_model(create_jc69_site_model())
is_hky_site_model(create_tn93_site_model())

# other models
is_hky_site_model(NA)
is_hky_site_model(NULL)
is_hky_site_model("nonsense")
is_hky_site_model(create_strict_clock_model())
is_hky_site_model(create_bd_tree_prior())
is_hky_site_model(create_mcmc())

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

# TRUE
is_id("anthus_aco")
is_id(3)

# FALSE
is_id(ape::rcoal(3))
is_id(NULL)
is_id(NA)

check_empty_beautier_folder()
```

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 *Determine if *x* is an initialized distribution object as created by [create_distr](#)*

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 *Determine if x is an initialized exponential distribution object as created by [create_exp_distr](#)*

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

```
check_empty_beautier_folder()

gtr_site_model <- create_gtr_site_model()
# FALSE: not yet initialized
is_init_gtr_site_model(gtr_site_model)
gtr_site_model <- init_gtr_site_model(gtr_site_model)
# TRUE: now it is initialized
is_init_gtr_site_model(gtr_site_model)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

hky_site_model <- create_hky_site_model()
# FALSE: not yet initialized
is_init_hky_site_model(hky_site_model)
hky_site_model <- init_hky_site_model(hky_site_model)
# TRUE: now it is initialized
is_init_hky_site_model(hky_site_model)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

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()
  )
)
# FALSE: not yet initialized
is_init_jc69_site_model(jc69_site_model)
jc69_site_model <- init_jc69_site_model(jc69_site_model)
# TRUE: now it is initialized
is_init_jc69_site_model(jc69_site_model)

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

tn93_site_model <- create_tn93_site_model()
# FALSE: not yet initialized
is_init_tn93_site_model(tn93_site_model)
tn93_site_model <- init_tn93_site_model(tn93_site_model)
# TRUE: now it is initialized
is_init_tn93_site_model(tn93_site_model)

check_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

# TRUE
is_inv_gamma_distr(create_inv_gamma_distr())
# FALSE
is_inv_gamma_distr(create_laplace_distr())
is_inv_gamma_distr(NA)
is_inv_gamma_distr(NULL)
is_inv_gamma_distr("nonsense")

check_empty_beautier_folder()
```

| | |
|----------------|---|
| 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

```
check_empty_beautier_folder()

# site models
is_jc69_site_model(create_gtr_site_model())
is_jc69_site_model(create_hky_site_model())
is_jc69_site_model(create_jc69_site_model())
is_jc69_site_model(create_tn93_site_model())

# other models
is_jc69_site_model(NA)
is_jc69_site_model(NULL)
is_jc69_site_model("nonsense")
is_jc69_site_model(create_strict_clock_model())
is_jc69_site_model(create_bd_tree_prior())
is_jc69_site_model(create_mcmc())

check_empty_beautier_folder()
```

| | |
|------------------|---|
| 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

```
check_empty_beautier_folder()

is_kappa_1_param(create_alpha_param())
is_kappa_1_param(create_beta_param())
is_kappa_1_param(create_clock_rate_param())
is_kappa_1_param(create_kappa_1_param())
is_kappa_1_param(create_kappa_2_param())
is_kappa_1_param(create_lambda_param())
is_kappa_1_param(create_m_param())
is_kappa_1_param(create_mean_param())
is_kappa_1_param(create_mu_param())
is_kappa_1_param(create_rate_ac_param())
is_kappa_1_param(create_rate_ag_param())
is_kappa_1_param(create_rate_at_param())
is_kappa_1_param(create_rate_cg_param())
is_kappa_1_param(create_rate_ct_param())
is_kappa_1_param(create_rate_gt_param())
is_kappa_1_param(create_s_param())
is_kappa_1_param(create_scale_param())
is_kappa_1_param(create_sigma_param())
```

```

is_kappa_1_param(NA)
is_kappa_1_param(NULL)
is_kappa_1_param("nonsense")
is_kappa_1_param(create_jc69_site_model())
is_kappa_1_param(create_strict_clock_model())
is_kappa_1_param(create_yule_tree_prior())
is_kappa_1_param(create_mcmc())

check_empty_beautier_folder()

```

is_kappa_2_param *Determine if the object is a valid kappa 2 parameter*

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

```

check_empty_beautier_folder()

is_kappa_2_param(create_alpha_param())
is_kappa_2_param(create_beta_param())
is_kappa_2_param(create_clock_rate_param())
is_kappa_2_param(create_kappa_1_param())
is_kappa_2_param(create_kappa_2_param())
is_kappa_2_param(create_lambda_param())
is_kappa_2_param(create_m_param())
is_kappa_2_param(create_mean_param())

```



```
is_kappa_2_param(create_mu_param())
is_kappa_2_param(create_rate_ac_param())
is_kappa_2_param(create_rate_ag_param())
is_kappa_2_param(create_rate_at_param())
is_kappa_2_param(create_rate_cg_param())
is_kappa_2_param(create_rate_ct_param())
is_kappa_2_param(create_rate_gt_param())
is_kappa_2_param(create_s_param())
is_kappa_2_param(create_scale_param())
is_kappa_2_param(create_sigma_param())

is_kappa_2_param(NA)
is_kappa_2_param(NULL)
is_kappa_2_param("nonsense")
is_kappa_2_param(create_jc69_site_model())
is_kappa_2_param(create_strict_clock_model())
is_kappa_2_param(create_yule_tree_prior())
is_kappa_2_param(create_mcmc())

check_empty_beautier_folder()
```

| | |
|-----------------|--|
| is_lambda_param | <i>Determine if the object is a valid lambda parameter</i> |
|-----------------|--|

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

```

check_empty_beautier_folder()

is_lambda_param(create_alpha_param())
is_lambda_param(create_beta_param())
is_lambda_param(create_clock_rate_param())
is_lambda_param(create_kappa_1_param())
is_lambda_param(create_kappa_2_param())
is_lambda_param(create_lambda_param())
is_lambda_param(create_m_param())
is_lambda_param(create_mean_param())
is_lambda_param(create_mu_param())
is_lambda_param(create_rate_ac_param())
is_lambda_param(create_rate_ag_param())
is_lambda_param(create_rate_at_param())
is_lambda_param(create_rate_cg_param())
is_lambda_param(create_rate_ct_param())
is_lambda_param(create_rate_gt_param())
is_lambda_param(create_s_param())
is_lambda_param(create_scale_param())
is_lambda_param(create_sigma_param())

is_lambda_param(NA)
is_lambda_param(NULL)
is_lambda_param("nonsense")
is_lambda_param(create_jc69_site_model())
is_lambda_param(create_strict_clock_model())
is_lambda_param(create_yule_tree_prior())
is_lambda_param(create_mcmc())

check_empty_beautier_folder()

```

| | |
|------------------|--|
| 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

```
# TRUE
is_laplace_distr(create_laplace_distr())
# FALSE
is_laplace_distr(create_log_normal_distr())
is_laplace_distr(NA)
is_laplace_distr(NULL)
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

```
check_empty_beautier_folder()

# TRUE
is_log_normal_distr(create_log_normal_distr())
# FALSE
is_log_normal_distr(create_normal_distr())
is_distr(NA)
is_distr(NULL)
is_distr("nonsense")

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# Returns TRUE
is_mcmc(create_mcmc())
is_mcmc(create_ns_mcmc())

# Returns FALSE
```

```
is_mcmc("nonsense")
is_mcmc(NULL)
is_mcmc(NA)
is_mcmc("")
is_mcmc(c())

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# TRUE
is_nested_sampling_mcmc(create_ns_mcmc())
# FALSE
is_nested_sampling_mcmc(create_mcmc())
is_nested_sampling_mcmc("nonsense")

check_empty_beautier_folder()
```

is_mean_param*Determine if the object is a valid mean parameter*

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

```
check_empty_beautier_folder()

is_mean_param(create_alpha_param())
is_mean_param(create_beta_param())
is_mean_param(create_clock_rate_param())
is_mean_param(create_kappa_1_param())
is_mean_param(create_kappa_2_param())
is_mean_param(create_lambda_param())
is_mean_param(create_m_param())
is_mean_param(create_mean_param())
is_mean_param(create_mu_param())
is_mean_param(create_rate_ac_param())
is_mean_param(create_rate_ag_param())
```

```

is_mean_param(create_rate_at_param())
is_mean_param(create_rate_cg_param())
is_mean_param(create_rate_ct_param())
is_mean_param(create_rate_gt_param())
is_mean_param(create_s_param())
is_mean_param(create_scale_param())
is_mean_param(create_sigma_param())

is_mean_param(NA)
is_mean_param(NULL)
is_mean_param("nonsense")
is_mean_param(create_jc69_site_model())
is_mean_param(create_strict_clock_model())
is_mean_param(create_yule_tree_prior())
is_mean_param(create_mcmc())

check_empty_beautier_folder()

```

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. Note that BEAST2 also supports missing data, by using a dash (-) or question mark (?) as a sequence.

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

Determine if the object is an empty (NA) or valid MRCA prior.

Description

Determine if 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

```
check_empty_beautier_folder()

# TRUE
is_mrca_prior(create_mrca_prior())
# Also 'NA' is a valid MRCA prior,
# denoting that there no MRCA priors
is_mrca_prior(NA)

# FALSE
is_mrca_prior(NULL)
is_mrca_prior("nonsense")

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

is_mu_param(create_alpha_param())
is_mu_param(create_beta_param())
is_mu_param(create_clock_rate_param())
is_mu_param(create_kappa_1_param())
is_mu_param(create_kappa_2_param())
is_mu_param(create_lambda_param())
is_mu_param(create_m_param())
is_mu_param(create_mean_param())
is_mu_param(create_mu_param())
is_mu_param(create_rate_ac_param())
is_mu_param(create_rate_ag_param())
is_mu_param(create_rate_at_param())
is_mu_param(create_rate_cg_param())
is_mu_param(create_rate_ct_param())
is_mu_param(create_rate_gt_param())
is_mu_param(create_s_param())
is_mu_param(create_scale_param())
is_mu_param(create_sigma_param())
```

```

is_mu_param(NA)
is_mu_param(NULL)
is_mu_param("nonsense")
is_mu_param(create_jc69_site_model())
is_mu_param(create_strict_clock_model())
is_mu_param(create_yule_tree_prior())
is_mu_param(create_mcmc())

check_empty_beautier_folder()

```

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(m_param)
```

Arguments

m_param an m parameter, as created by [create_m_param](#)

Value

TRUE if x is a valid m parameter, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```

check_empty_beautier_folder()

is_m_param(create_alpha_param())
is_m_param(create_beta_param())
is_m_param(create_clock_rate_param())
is_m_param(create_kappa_1_param())
is_m_param(create_kappa_2_param())
is_m_param(create_lambda_param())
is_m_param(create_m_param())
is_m_param(create_mean_param())
is_m_param(create_mu_param())
is_m_param(create_rate_ac_param())
is_m_param(create_rate_ag_param())
is_m_param(create_rate_at_param())

```

```
is_m_param(create_rate_cg_param())
is_m_param(create_rate_ct_param())
is_m_param(create_rate_gt_param())
is_m_param(create_s_param())
is_m_param(create_scale_param())
is_m_param(create_sigma_param())

is_m_param(NA)
is_m_param(NULL)
is_m_param("nonsense")
is_m_param(create_jc69_site_model())
is_m_param(create_strict_clock_model())
is_m_param(create_yule_tree_prior())
is_m_param(create_mcmc())

check_empty_beautier_folder()
```

| | |
|-----------------|---|
| is_normal_distr | <i>Determine if the object is a valid normal distribution as created by create_normal_distr</i> |
|-----------------|---|

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

```
check_empty_beautier_folder()

# TRUE
is_normal_distr(create_normal_distr())
# FALSE
is_normal_distr(create_one_div_x_distr())
is_normal_distr(NA)
is_normal_distr(NULL)
is_normal_distr("nonsense")

check_empty_beautier_folder()
```

is_one_bool

Check if the argument is one boolean

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

```
check_empty_beautier_folder()

# TRUE
is_one_bool(TRUE)
is_one_bool(FALSE)

# FALSE
is_one_bool(NULL)
is_one_bool(NA)
is_one_bool(c())
is_one_bool("nonsense")
is_one_bool(is_one_bool)
is_one_bool(c(TRUE, FALSE))

check_empty_beautier_folder()
```

| | |
|--------------------|--|
| 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

```
check_empty_beautier_folder()

# TRUE
is_one_div_x_distr(create_one_div_x_distr())
# FALSE
is_one_div_x_distr(create_poisson_distr())
is_one_div_x_distr(NA)
is_one_div_x_distr(NULL)
is_one_div_x_distr("nonsense")

check_empty_beautier_folder()
```

| | |
|---------------|---|
| 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

```
check_empty_beautier_folder()

# TRUE
is_one_double(314)
is_one_double(0)
is_one_double(-314)
is_one_double(3.14)

# FALSE
is_one_double(NULL)
is_one_double(NA)
is_one_double(Inf)
is_one_double("nonsense")
is_one_double(is_one_double)
is_one_double(c())
is_one_double(c(1, 2))

check_empty_beautier_folder()
```

| | |
|------------|---|
| 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

```
check_empty_beautier_folder()

# TRUE
is_one_int(314)
is_one_int(0)
is_one_int(-314)
# FALSE
is_one_int(3.14)
is_one_int(NULL)
is_one_int(NA)
is_one_int(Inf)
is_one_int("nonsense")
is_one_int(c())
is_one_int(c(1, 2))

check_empty_beautier_folder()
```

| | |
|-----------|----------------------------------|
| 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

```
check_empty_beautier_folder()

# TRUE
testit::assert(is_one_na(NA))

# FALSE
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)))

check_empty_beautier_folder()
```

is_on_appveyor *Determines if the environment is AppVeyor*

Description

Determines if the environment is AppVeyor

Usage

```
is_on_appveyor()
```

Value

TRUE if run on AppVeyor, **FALSE** otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

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

is_on_ci *Determines if the environment is a continuous integration service*

Description

Determines if the environment is a continuous integration service

Usage

```
is_on_ci()
```

Value

TRUE if run on AppVeyor or Travis CI, **FALSE** otherwise

Note

It is possible to fake being on continuous integration service, in this case GitHub Actions, using:
““r Sys.setenv(GITHUB_ACTIONS = "I fake being on GitHub Actions") is_on_ci() # Will be true
““

To undo this, do

```
““r Sys.setenv(GITHUB_ACTIONS = "") is_on_ci() # Will be false ““
```

Author(s)

Richèl J.C. Bilderbeek

Examples

```
is_on_ci()
```

is_on_github_actions *Determines if the environment is GitHub Actions*

Description

Determines if the environment is GitHub Actions

Usage

```
is_on_github_actions()
```

Value

TRUE if run on GitHub Actions, **FALSE** otherwise

Note

It is possible to fake being on GitHub Actions, using:

```
““r Sys.setenv(GITHUB_ACTIONS = "I fake being on GitHub Actions") is_on_github_actions() # Will be true ““
```

To undo this, do

```
““r Sys.setenv(GITHUB_ACTIONS = "") is_on_github_actions() # Will be false ““
```

Author(s)

Richèl J.C. Bilderbeek

Examples

```
if (is_on_github_actions()) {  
  message("Running on GitHub Actions")  
}
```

is_on_travis *Determines if the environment is Travis CI*

Description

Determines if the environment is Travis CI

Usage

```
is_on_travis()
```

Value

TRUE if run on Travis CI, **FALSE** otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

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

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

```
check_empty_beautier_folder()

# TRUE
is_param(create_alpha_param())
is_param(create_beta_param())
is_param(create_clock_rate_param())
is_param(create_kappa_1_param())
is_param(create_kappa_2_param())
is_param(create_lambda_param())
is_param(create_m_param())
is_param(create_mean_param())
is_param(create_mu_param())
is_param(create_rate_ac_param())
is_param(create_rate_ag_param())
is_param(create_rate_at_param())
is_param(create_rate_cg_param())
is_param(create_rate_ct_param())
is_param(create_rate_gt_param())
is_param(create_s_param())
is_param(create_scale_param())
is_param(create_sigma_param())

# FALSE
is_param(NA)
is_param(NULL)
is_param("nonsense")
is_param(create_jc69_site_model())
is_param(create_strict_clock_model())
is_param(create_yule_tree_prior())
is_param(create_mcmc())

check_empty_beautier_folder()
```

| | |
|---------------|---|
| 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

```
check_empty_beautier_folder()

# TRUE
is_param_name("alpha")
is_param_name("beta")
is_param_name("clock_rate")
is_param_name("kappa_1")
is_param_name("kappa_2")
is_param_name("lambda")
is_param_name("m")
is_param_name("mean")
is_param_name("mu")
is_param_name("rate_ac")
is_param_name("rate_ag")
is_param_name("rate_at")
is_param_name("rate_cg")
is_param_name("rate_ct")
is_param_name("rate_gt")
is_param_name("s")
is_param_name("scale")
is_param_name("sigma")

# FALSE
is_param_name("nonsense")
is_param_name(NA)
is_param_name(NULL)
is_param_name("")
is_param_name(c())

check_empty_beautier_folder()
```

is_phylo

Checks if the input is a phylogeny

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

```
check_empty_beautier_folder()

# TRUE
phylogeny <- ape::read.tree(text = "(a:15,b:15):1;")
is_phylo(phylogeny)

# FALSE
is_phylo("nonsense")
is_phylo(NA)
is_phylo(NULL)

check_empty_beautier_folder()
```

| | |
|------------------|---|
| 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

```
check_empty_beautier_folder()

# TRUE
is_poisson_distr(create_poisson_distr())
# FALSE
is_poisson_distr(create_uniform_distr())
is_distr(NA)
is_distr(NULL)
is_distr("nonsense")

check_empty_beautier_folder()
```

is_rate_ac_param

Determine if the object is a valid 'rate AC' parameter

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

```

check_empty_beautier_folder()

is_rate_ac_param(create_alpha_param())
is_rate_ac_param(create_beta_param())
is_rate_ac_param(create_clock_rate_param())
is_rate_ac_param(create_kappa_1_param())
is_rate_ac_param(create_kappa_2_param())
is_rate_ac_param(create_lambda_param())
is_rate_ac_param(create_m_param())
is_rate_ac_param(create_mean_param())
is_rate_ac_param(create_mu_param())
is_rate_ac_param(create_rate_ac_param())
is_rate_ac_param(create_rate_ag_param())
is_rate_ac_param(create_rate_at_param())
is_rate_ac_param(create_rate_cg_param())
is_rate_ac_param(create_rate_ct_param())
is_rate_ac_param(create_rate_gt_param())
is_rate_ac_param(create_s_param())
is_rate_ac_param(create_scale_param())
is_rate_ac_param(create_sigma_param())

is_rate_ac_param(NA)
is_rate_ac_param(NULL)
is_rate_ac_param("nonsense")
is_rate_ac_param(create_jc69_site_model())
is_rate_ac_param(create_strict_clock_model())
is_rate_ac_param(create_yule_tree_prior())
is_rate_ac_param(create_mcmc())

check_empty_beautier_folder()

```

| | |
|------------------|---|
| 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

```

check_empty_beautier_folder()

is_rate_ag_param(create_alpha_param())
is_rate_ag_param(create_beta_param())
is_rate_ag_param(create_clock_rate_param())
is_rate_ag_param(create_kappa_1_param())
is_rate_ag_param(create_kappa_2_param())
is_rate_ag_param(create_lambda_param())
is_rate_ag_param(create_m_param())
is_rate_ag_param(create_mean_param())
is_rate_ag_param(create_mu_param())
is_rate_ag_param(create_rate_ac_param())
is_rate_ag_param(create_rate_ag_param())
is_rate_ag_param(create_rate_at_param())
is_rate_ag_param(create_rate_cg_param())
is_rate_ag_param(create_rate_ct_param())
is_rate_ag_param(create_rate_gt_param())
is_rate_ag_param(create_s_param())
is_rate_ag_param(create_scale_param())
is_rate_ag_param(create_sigma_param())

is_rate_ag_param(NA)
is_rate_ag_param(NULL)
is_rate_ag_param("nonsense")
is_rate_ag_param(create_jc69_site_model())
is_rate_ag_param(create_strict_clock_model())
is_rate_ag_param(create_yule_tree_prior())
is_rate_ag_param(create_mcmc())

check_empty_beautier_folder()

```

| | |
|------------------|---|
| 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

```
check_empty_beautier_folder()

is_rate_at_param(create_alpha_param())
is_rate_at_param(create_beta_param())
is_rate_at_param(create_clock_rate_param())
is_rate_at_param(create_kappa_1_param())
is_rate_at_param(create_kappa_2_param())
is_rate_at_param(create_lambda_param())
is_rate_at_param(create_m_param())
is_rate_at_param(create_mean_param())
is_rate_at_param(create_mu_param())
is_rate_at_param(create_rate_ac_param())
is_rate_at_param(create_rate_ag_param())
is_rate_at_param(create_rate_at_param())
is_rate_at_param(create_rate_cg_param())
is_rate_at_param(create_rate_ct_param())
is_rate_at_param(create_rate_gt_param())
is_rate_at_param(create_s_param())
is_rate_at_param(create_scale_param())
is_rate_at_param(create_sigma_param())

is_rate_at_param(NA)
is_rate_at_param(NULL)
is_rate_at_param("nonsense")
is_rate_at_param(create_jc69_site_model())
is_rate_at_param(create_strict_clock_model())
is_rate_at_param(create_yule_tree_prior())
is_rate_at_param(create_mcmc())

check_empty_beautier_folder()
```

| | |
|------------------|---|
| 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

```
check_empty_beautier_folder()

is_rate_cg_param(create_alpha_param())
is_rate_cg_param(create_beta_param())
is_rate_cg_param(create_clock_rate_param())
is_rate_cg_param(create_kappa_1_param())
is_rate_cg_param(create_kappa_2_param())
is_rate_cg_param(create_lambda_param())
is_rate_cg_param(create_m_param())
is_rate_cg_param(create_mean_param())
is_rate_cg_param(create_mu_param())
is_rate_cg_param(create_rate_ac_param())
is_rate_cg_param(create_rate_ag_param())
is_rate_cg_param(create_rate_at_param())
is_rate_cg_param(create_rate_cg_param())
is_rate_cg_param(create_rate_ct_param())
is_rate_cg_param(create_rate_gt_param())
is_rate_cg_param(create_s_param())
is_rate_cg_param(create_scale_param())
is_rate_cg_param(create_sigma_param())
```

```

is_rate_cg_param(NA)
is_rate_cg_param(NULL)
is_rate_cg_param("nonsense")
is_rate_cg_param(create_jc69_site_model())
is_rate_cg_param(create_strict_clock_model())
is_rate_cg_param(create_yule_tree_prior())
is_rate_cg_param(create_mcmc())

check_empty_beautier_folder()

```

| | |
|------------------|---|
| 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

```

check_empty_beautier_folder()

is_rate_ct_param(create_alpha_param())
is_rate_ct_param(create_beta_param())
is_rate_ct_param(create_clock_rate_param())
is_rate_ct_param(create_kappa_1_param())
is_rate_ct_param(create_kappa_2_param())
is_rate_ct_param(create_lambda_param())
is_rate_ct_param(create_m_param())
is_rate_ct_param(create_mean_param())

```

```

is_rate_ct_param(create_mu_param())
is_rate_ct_param(create_rate_ac_param())
is_rate_ct_param(create_rate_ag_param())
is_rate_ct_param(create_rate_at_param())
is_rate_ct_param(create_rate_cg_param())
is_rate_ct_param(create_rate_ct_param())
is_rate_ct_param(create_rate_gt_param())
is_rate_ct_param(create_s_param())
is_rate_ct_param(create_scale_param())
is_rate_ct_param(create_sigma_param())

is_rate_ct_param(NA)
is_rate_ct_param(NULL)
is_rate_ct_param("nonsense")
is_rate_ct_param(create_jc69_site_model())
is_rate_ct_param(create_strict_clock_model())
is_rate_ct_param(create_yule_tree_prior())
is_rate_ct_param(create_mcmc())

check_empty_beautier_folder()

```

| | |
|------------------|---|
| 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

```

check_empty_beautier_folder()

is_rate_gt_param(create_alpha_param())
is_rate_gt_param(create_beta_param())
is_rate_gt_param(create_clock_rate_param())
is_rate_gt_param(create_kappa_1_param())
is_rate_gt_param(create_kappa_2_param())
is_rate_gt_param(create_lambda_param())
is_rate_gt_param(create_m_param())
is_rate_gt_param(create_mean_param())
is_rate_gt_param(create_mu_param())
is_rate_gt_param(create_rate_ac_param())
is_rate_gt_param(create_rate_ag_param())
is_rate_gt_param(create_rate_at_param())
is_rate_gt_param(create_rate_cg_param())
is_rate_gt_param(create_rate_ct_param())
is_rate_gt_param(create_rate_gt_param())
is_rate_gt_param(create_s_param())
is_rate_gt_param(create_scale_param())
is_rate_gt_param(create_sigma_param())

is_rate_gt_param(NA)
is_rate_gt_param(NULL)
is_rate_gt_param("nonsense")
is_rate_gt_param(create_jc69_site_model())
is_rate_gt_param(create_strict_clock_model())
is_rate_gt_param(create_yule_tree_prior())
is_rate_gt_param(create_mcmc())

check_empty_beautier_folder()

```

| | |
|--------------------|--|
| is_rln_clock_model | <i>Determine if the object is a valid relaxed log normal clock model</i> |
|--------------------|--|

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

```
check_empty_beautier_folder()

is_rln_clock_model(create_strict_clock_model())
is_rln_clock_model(create_rln_clock_model())

is_rln_clock_model(NA)
is_rln_clock_model(NULL)
is_rln_clock_model("nonsense")
is_rln_clock_model(create_jc69_site_model())
is_rln_clock_model(create_mcmc())

check_empty_beautier_folder()
```

is_scale_param

Determine if the object is a valid scale parameter

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

```

check_empty_beautier_folder()

is_scale_param(create_alpha_param())
is_scale_param(create_beta_param())
is_scale_param(create_clock_rate_param())
is_scale_param(create_kappa_1_param())
is_scale_param(create_kappa_2_param())
is_scale_param(create_lambda_param())
is_scale_param(create_m_param())
is_scale_param(create_mean_param())
is_scale_param(create_mu_param())
is_scale_param(create_rate_ac_param())
is_scale_param(create_rate_ag_param())
is_scale_param(create_rate_at_param())
is_scale_param(create_rate_cg_param())
is_scale_param(create_rate_ct_param())
is_scale_param(create_rate_gt_param())
is_scale_param(create_s_param())
is_scale_param(create_scale_param())
is_scale_param(create_sigma_param())

is_scale_param(NA)
is_scale_param(NULL)
is_scale_param("nonsense")
is_scale_param(create_jc69_site_model())
is_scale_param(create_strict_clock_model())
is_scale_param(create_yule_tree_prior())
is_scale_param(create_mcmc())

check_empty_beautier_folder()

```

| | |
|----------------|---|
| 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

```

check_empty_beautier_folder()

is_sigma_param(create_alpha_param())
is_sigma_param(create_beta_param())
is_sigma_param(create_clock_rate_param())
is_sigma_param(create_kappa_1_param())
is_sigma_param(create_kappa_2_param())
is_sigma_param(create_lambda_param())
is_sigma_param(create_m_param())
is_sigma_param(create_mean_param())
is_sigma_param(create_mu_param())
is_sigma_param(create_rate_ac_param())
is_sigma_param(create_rate_ag_param())
is_sigma_param(create_rate_at_param())
is_sigma_param(create_rate_cg_param())
is_sigma_param(create_rate_ct_param())
is_sigma_param(create_rate_gt_param())
is_sigma_param(create_s_param())
is_sigma_param(create_scale_param())
is_sigma_param(create_sigma_param())

is_sigma_param(NA)
is_sigma_param(NULL)
is_sigma_param("nonsense")
is_sigma_param(create_jc69_site_model())
is_sigma_param(create_strict_clock_model())
is_sigma_param(create_yule_tree_prior())
is_sigma_param(create_mcmc())

check_empty_beautier_folder()

```

is_site_model

Determine if the object is a valid site_model

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

```
check_empty_beautier_folder()

# TRUE
is_site_model(create_gtr_site_model())
is_site_model(create_hky_site_model())
is_site_model(create_jc69_site_model())
is_site_model(create_tn93_site_model())

# FALSE
is_site_model(NA)
is_site_model(NULL)
is_site_model("nonsense")
is_site_model(create_strict_clock_model())
is_site_model(create_bd_tree_prior())
is_site_model(create_mcmc())

check_empty_beautier_folder()
```

| | |
|--------------------|--|
| is_site_model_name | <i>Determines if the name is a valid site_model name</i> |
|--------------------|--|

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

```
check_empty_beautier_folder()

# TRUE
is_site_model_name("JC69")
is_site_model_name("HKY")
is_site_model_name("TN93")
is_site_model_name("GTR")
# FALSE
is_site_model_name("nonsense")

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

is_strict_clock_model(create_strict_clock_model())
is_strict_clock_model(create_rln_clock_model())

is_strict_clock_model(NA)
```

```
is_strict_clock_model(NULL)
is_strict_clock_model("nonsense")
is_strict_clock_model(create_jc69_site_model())
is_strict_clock_model(create_mcmc())

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

is_s_param(create_alpha_param())
is_s_param(create_beta_param())
is_s_param(create_clock_rate_param())
is_s_param(create_kappa_1_param())
is_s_param(create_kappa_2_param())
is_s_param(create_lambda_param())
is_s_param(create_m_param())
is_s_param(create_mean_param())
is_s_param(create_mu_param())
is_s_param(create_rate_ac_param())
is_s_param(create_rate_ag_param())
is_s_param(create_rate_at_param())
is_s_param(create_rate_cg_param())
is_s_param(create_rate_ct_param())
is_s_param(create_rate_gt_param())
is_s_param(create_s_param())
```

```

is_s_param(create_scale_param())
is_s_param(create_sigma_param())

is_s_param(NA)
is_s_param(NULL)
is_s_param("nonsense")
is_s_param(create_jc69_site_model())
is_s_param(create_strict_clock_model())
is_s_param(create_yule_tree_prior())
is_s_param(create_mcmc())

check_empty_beautier_folder()

```

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

```

check_empty_beautier_folder()

# site models
is_tn93_site_model(create_gtr_site_model())
is_tn93_site_model(create_hky_site_model())
is_tn93_site_model(create_jc69_site_model())
is_tn93_site_model(create_tn93_site_model())

# other models
is_tn93_site_model(NA)
is_tn93_site_model(NULL)

```

```
is_tn93_site_model("nonsense")
is_tn93_site_model("")
is_tn93_site_model(c())
is_tn93_site_model(create_strict_clock_model())
is_tn93_site_model(create_bd_tree_prior())
is_tn93_site_model(create_mcmc())

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

testit::assert(is_tree_prior(create_bd_tree_prior()))
testit::assert(is_tree_prior(create_yule_tree_prior()))
testit::assert(!is_tree_prior("nonsense"))

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# TRUE
is_tree_prior_name("birth_death")
is_tree_prior_name("coalescent_bayesian_skyline")
is_tree_prior_name("coalescent_constant_population")
is_tree_prior_name("coalescent_exp_population")
is_tree_prior_name("yule")
# FALSE
is_tree_prior_name("nonsense")

check_empty_beautier_folder()
```

is_uniform_distr *Determine if the object is a valid uniform distribution as created by*
[create_uniform_distr](#)

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

```
check_empty_beautier_folder()

# TRUE
is_uniform_distr(create_uniform_distr())
# FALSE
is_uniform_distr(create_beta_distr())
is_uniform_distr(NA)
is_uniform_distr(NULL)
is_uniform_distr("nonsense")

check_empty_beautier_folder()
```

| | |
|--------|---|
| 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 *Determine if the object is a valid Yule tree prior,*

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

```
check_empty_beautier_folder()

# TRUE
is_yule_tree_prior(create_yule_tree_prior())

# FALSE
is_yule_tree_prior(create_bd_tree_prior())
is_yule_tree_prior(create_cbs_tree_prior())
is_yule_tree_prior(create_ccp_tree_prior())
is_yule_tree_prior(create_cep_tree_prior())

check_empty_beautier_folder()
```

| | |
|-----------------|---|
| 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

```
if (is_on_ci()) {
  check_empty_beautier_folder()

  # <run id="mcmc" spec="MCMC" chainLength="1e+07">
  mcmc_to_xml_run(create_mcmc())

  check_empty_beautier_folder()
}
```

| | |
|-------------------------|--|
| 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

```
check_empty_beautier_folder()

# <run id="mcmc" spec="MCMC" chainLength="1e+07">
xml <- mcmc_to_xml_run_default(create_mcmc())

check_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

# "<run id=\"mcmc\" spec=\"beast.gss.NS\" chainLength=\"1e+07\" \"
# \"particleCount=\"1\" subChainLength=\"5000\" epsilon=\"1e-12\">"
mcmc_to_xml_run_nested_sampling(create_ns_mcmc())

check_empty_beautier_folder()
```

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(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.

Details

```
<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>
```

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

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(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

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

# <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_empty_beautier_folder()
```

`mrca_prior_to_xml_state`

Internal function to create the XML of an MRCA prior, as used in the state section

Description

Internal function to create the XML of an MRCA prior, as used in the state section

Usage

```
mrca_prior_to_xml_state(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 tree prior as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

mrca_prior_to_xml_state(
  inference_model = create_inference_model(
    mrca_prior = create_mrca_prior(
      alignment_id = "test_output_0",
      mrca_distr = create_normal_distr(id = 42)
    ),
    clock_model = create_strict_clock_model()
  )
)

check_empty_beautier_folder()
```

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

Details

```
<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>
```

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

mrca_prior_to_xml_tracelog

Internal function

Description

Internal function to creates the MRCA prior's XML for the tracelog section.

Usage

```
mrca_prior_to_xml_tracelog(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.

Details

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

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

| | |
|----------------|--------------------------|
| m_param_to_xml | <i>Internal function</i> |
|----------------|--------------------------|

Description

Converts an m parameter to XML

Usage

```
m_param_to_xml(m_param, beauti_options = create_beauti_options())
```

Argumentsm_param an m parameter, as created by [create_m_param](#)beauti_options one BEAUti options object, as returned by [create_beauti_options](#)**Value**

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

no_taxa_to_xml_tree *Internal function*

Description

Creates the 'tree' section of a BEAST2 XML parameter file, which is part of a 'state' section, without being indented, when there is no tip-dating

Usage

```
no_taxa_to_xml_tree(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.

Details

The tree tag has these elements:

```
<tree[...]>
  <taxonset[...]>
  [...]
  </taxonset>
</run>
```

Value

the random phylogeny as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml *Internal function*

Description

Converts a parameter to XML

Usage

```
parameter_to_xml(parameter, beauti_options = create_beauti_options())
```

Arguments

parameter a parameter, as created by [create_param](#))
beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()  
parameter_to_xml(create_alpha_param(id = 1))  
check_empty_beautier_folder()
```

parameter_to_xml_kappa_1
Internal function

Description

Converts a kappa 1 parameter to XML

Usage

```
parameter_to_xml_kappa_1(parameter, beauti_options = create_beauti_options())
```

Arguments

- parameter a kappa 1 parameter, a numeric value. For advanced usage, use the structure as created by [create_kappa_1_param](#))
- beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_kappa_2

Internal function

Description

Converts a kappa 2 parameter to XML

Usage

```
parameter_to_xml_kappa_2(parameter, beauti_options = create_beauti_options())
```

Arguments

- parameter a kappa 2 parameter, a numeric value. For advanced usage, use the structure as created by [create_kappa_2_param](#))
- beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_lambda
Internal function

Description

Converts a lambda parameter to XML

Usage

```
parameter_to_xml_lambda(parameter, beauti_options = create_beauti_options())
```

Arguments

`parameter` a lambda parameter, a numeric value. For advanced usage, use the structure as created by [create_lambda_param](#))
`beauti_options` one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_mean *Internal function*

Description

Converts a mean parameter to XML

Usage

```
parameter_to_xml_mean(parameter, beauti_options = create_beauti_options())
```

Arguments

`parameter` a mean parameter, a numeric value. For advanced usage, use the structure as created by [create_mean_param](#))
`beauti_options` one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_mu *Internal function*

Description

Converts a mu parameter to XML

Usage

```
parameter_to_xml_mu(parameter, beauti_options = create_beauti_options())
```

Arguments

parameter a mu parameter, a numeric value. For advanced usage, use the structure as created by [create_mu_param](#))

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

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_rate_ac
Internal function

Description

Converts a 'rate AC' parameter to XML

Usage

```
parameter_to_xml_rate_ac(  
  parameter,  
  beauti_options = create_beauti_options(),  
  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) |
| beauti_options | one BEAUti options object, as returned by create_beauti_options |
| 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
Internal function

Description

Converts a 'rate AG' parameter to XML

Usage

```
parameter_to_xml_rate_ag(
  parameter,
  beauti_options = create_beauti_options(),
  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) |
| beauti_options | one BEAUti options object, as returned by create_beauti_options |
| 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
Internal function

Description

Converts a 'rate AT' parameter to XML

Usage

```
parameter_to_xml_rate_at(  
  parameter,  
  beauti_options = create_beauti_options(),  
  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](#))

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

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
Internal function

Description

Converts a 'rate CG' parameter to XML

Usage

```
parameter_to_xml_rate_cg(  
  parameter,  
  beauti_options = create_beauti_options(),  
  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) |
| beauti_options | one BEAUti options object, as returned by create_beauti_options |
| 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
Internal function

Description

Converts a 'rate CT' parameter to XML

Usage

```
parameter_to_xml_rate_ct(
  parameter,
  beauti_options = create_beauti_options(),
  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) |
| beauti_options | one BEAUti options object, as returned by create_beauti_options |
| 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

Internal function

Description

Converts a 'rate GT' parameter to XML

Usage

```
parameter_to_xml_rate_gt(  
  parameter,  
  beauti_options = create_beauti_options(),  
  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](#))

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

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 *Internal function*

Description

Converts a s parameter to XML

Usage

```
parameter_to_xml_s(parameter, beauti_options = create_beauti_options())
```

Arguments

- parameter a s parameter, a numeric value. For advanced usage, use the structure as created by [create_s_param](#))
- beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_scale
Internal function

Description

Converts a scale parameter to XML

Usage

```
parameter_to_xml_scale(parameter, beauti_options = create_beauti_options())
```

Arguments

- parameter a scale parameter, a numeric value. For advanced usage, use the structure as created by [create_scale_param](#))
- beauti_options one BEAUti options object, as returned by [create_beauti_options](#)

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

parameter_to_xml_sigma

Internal function

Description

Converts a sigma parameter to XML

Usage

```
parameter_to_xml_sigma(parameter, beauti_options = create_beauti_options())
```

Arguments

parameter a sigma parameter, a numeric value. For advanced usage, use the structure as created by [create_sigma_param](#))

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

Value

the parameter as XML text

Author(s)

Richèl J.C. Bilderbeek

remove_beautier_folder

Check there are no files in the default [beautier](#) folder

Description

Check there are no files in the default [beautier](#) folder. The goal is to make sure no temporary files are left undeleted. Will [stop](#) if there are files in the [beautier](#) folder.

Usage

```
remove_beautier_folder()
```

Value

Nothing.

Author(s)

Richèl J.C. Bilderbeek

See Also

use [remove_beautier_folder](#) to remove the default ‘beautier’ folder

Examples

```
check_empty_beautier_folder()
```

```
remove_beautier_folder()
```

```
check_empty_beautier_folder()
```

| | |
|--------------------|--|
| remove_empty_lines | <i>Remove all lines that are only whitespace</i> |
|--------------------|--|

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 | <i>Remove consecutive lines</i> |
|------------------|---------------------------------|

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 | <i>Rename the filenames in an inference model</i> |
|----------------------------------|---|

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

```
check_empty_beautier_folder()

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")
)

# Nah, put the files in anoth folder
inference_model <- rename_inference_model_filenames(
  inference_model = inference_model,
  rename_fun = get_replace_dir_fun("/home/does")
)

# Nah, store the files locally
rename_inference_model_filenames(
  inference_model = inference_model,
  rename_fun = get_remove_dir_fun()
)

check_empty_beautier_folder()
```

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

```
check_empty_beautier_folder()

# 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")
)

# Nah, files should be put in '/home/doe' folder instead
mcmc <- rename_mcmc_filenames(
  mcmc = mcmc,
  rename_fun = get_replace_dir_fun("/home/doe")
)

# Nah, files should be put in local folder instead
mcmc <- rename_mcmc_filenames(
  mcmc = mcmc,
  rename_fun = get_remove_dir_fun()
```



```
)  
check_empty_beautier_folder()
```

rln_clock_model_to_xml_mean_rate_prior
Internal function

Description

Internal function

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

rln_clock_model_to_xml_prior_distr
Internal function

Description

Internal function to converts a relaxed log-normal clock model to the `prior` section of the XML as text

Usage

```
rln_clock_model_to_xml_prior_distr(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

a character vector of XML strings

Author(s)

Richèl J.C. Bilderbeek

Examples

```
check_empty_beautier_folder()

# <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>

# Must be an initialized inference model
inference_model <- create_inference_model(
  clock_model = create_rln_clock_model(
    id = "test_output_0",
    ucldstdev_distr = create_gamma_distr(
      id = 0,
      alpha = create_alpha_param(id = 2, value = "0.5396"),
      beta = create_beta_param(id = 3, value = "0.3819")
    ),
    mean_rate_prior_distr = create_uniform_distr(id = 1),
    mparam_id = 1
  )
)

rln_clock_model_to_xml_prior_distr(inference_model)

check_empty_beautier_folder()
```

`rnd_phylo_to_xml_init` *Creates the XML of a random phylogeny, as used in the init section*

Description

Creates the XML text for the `beast` tag of a BEAST2 parameter file, which is directly after the XML declaration (created by [create_xml_declaration](#)).

Usage

```
rnd_phylo_to_xml_init(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.

Details

The `init` tag has these elements:

```
<init id=\"RandomTree.t:[...]\">
  <populationModel[...]>
    [...]
  </populationModel>
</init>
```

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

```
check_empty_beautier_folder()

# <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_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

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_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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

taxa_to_xml_tree *Internal function*

Description

Internal function to creates the 'tree' section of a BEAST2 XML parameter file, which is part of a 'state' section, without being indented.

Usage

```
taxa_to_xml_tree(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.

Details

The tree tag has these elements:

```
<tree[...]>
  <taxonset[...]>
  [...]
  </taxonset>
</tree>
```

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

`tipdate_taxa_to_xml_trait`*Internal function*

Description

Internal function to creates the 'trait' section of a BEAST2 XML parameter file, which is part of a 'tree' section, without being indented.

Usage

```
tipdate_taxa_to_xml_trait(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.

Details

The tree tag has these elements:

```
<run[...]>
  <state[...]>
    <tree[...]>
      <trait[...]>
        This part
      </trait>
    </tree>
  </run>
</state>
```

Value

lines of XML text

Author(s)

Richèl J.C. Bilderbeek

tipdate_taxa_to_xml_tree

Internal function

Description

Creates the tree section (part of the state section) when there is tip-dating

Usage

```
tipdate_taxa_to_xml_tree(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 random phylogeny as XML text

Author(s)

Richèl J.C. Bilderbeek

tree_model_to_tracelog_xml

Internal function

Description

Creates the tree models' XML for the tracelog section. That is, all XML tags that have the word 'tree' in them.

Usage

```
tree_model_to_tracelog_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.

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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

`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

```
check_empty_beautier_folder()

# <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_empty_beautier_folder()
```

```
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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

```
tree_prior_to_xml_operators
      Internal function
```

Description

Creates the XML of a tree prior, as used in the operators section

Usage

```
tree_prior_to_xml_operators(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 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

```
check_empty_beautier_folder()

# <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_empty_beautier_folder()
```

`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(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 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

```
check_empty_beautier_folder()

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

check_empty_beautier_folder()
```

unindent

Unindents text

Description

Unindents text

Usage

```
unindent(text)
```

Arguments

text one or more lines of text

Value

unindented lines of text

Author(s)

Richèl J.C. Bilderbeek

yule_tree_prior_to_xml_operators
Internal function

Description

Creates the XML of a Yule tree prior, as used in the operators section

Usage

```
yule_tree_prior_to_xml_operators(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 tree prior as XML text

Author(s)

Richèl J.C. Bilderbeek

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

```
check_empty_beautier_folder()
```

```
# <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_empty_beautier_folder()
```

Index

- alpha_parameter_to_xml, 12
- are_clock_models, 13
- are_equal_mcmcs, 14
- are_equal_screenlogs, 15
- are_equal_tracelogs, 16
- are_equal_treelogs, 17
- are_equal_xml_files, 18
- are_equal_xml_lines, 19
- are_equivalent_xml_files, 18, 20
- are_equivalent_xml_lines, 21
- are_equivalent_xml_lines_all, 21
- are_equivalent_xml_lines_loggers, 22
- are_equivalent_xml_lines_operators, 23
- are_equivalent_xml_lines_section, 23
- are_fasta_filenames, 24, 216
- are_ids, 25, 291
- are_init_clock_models, 26
- are_init_mrca_priors, 26
- are_init_site_models, 27
- are_init_tree_priors, 27
- are_mrca_align_ids_in_fasta, 28
- are_mrca_priors, 28
- are_mrca_taxon_names_in_fasta, 29
- are_rln_clock_models, 30
- are_site_models, 30
- are_tree_priors, 31

- bd_tree_prior_to_xml_prior_distr, 32
- beautier, 33, 41, 197, 217, 219, 233, 372
- beta_parameter_to_xml, 34

- cbs_tree_prior_to_xml_prior_distr, 34
- ccp_tree_prior_to_xml_prior_distr, 35
- ccp_tree_prior_to_xml_state, 36
- cep_tree_prior_to_xml_prior_distr, 37
- check_alignment_id, 38, 215
- check_beauti_options, 38
- check_clock_model, 39
- check_clock_models, 40
- check_empty_beautier_folder, 41
- check_file_and_model_agree, 42
- check_file_exists, 43
- check_filename, 42, 42, 198
- check_gamma_site_model, 44
- check_gamma_site_model_names, 44
- check_gtr_site_model, 45
- check_gtr_site_model_names, 46
- check_inference_model, 36, 43, 46, 47, 71–73, 82, 84, 85, 87, 90–95, 102–104, 113, 132, 152, 159, 165, 166, 168, 169, 184, 186, 187, 190, 191, 199, 229, 245–249, 260, 263, 357–359, 361, 362, 374, 378, 379, 384–387, 389, 391, 393
- check_inference_models, 47
- check_is_monophyletic, 48
- check_log_mode, 48
- check_log_sort, 49
- check_mcmc, 49, 49, 50, 51, 54, 81, 88, 124, 135, 142, 174, 177, 199, 230, 283, 355, 356, 376
- check_mcmc_list_element_names, 50
- check_mcmc_nested_sampling (check_ns_mcmc), 54
- check_mcmc_values, 51
- check_mrca_prior, 51, 53
- check_mrca_prior_name, 52
- check_mrca_prior_names, 53
- check_mrca_prior_taxa_names, 53
- check_nested_sampling_mcmc (check_ns_mcmc), 54
- check_ns_mcmc, 54, 144
- check_param, 54
- check_param_names, 55
- check_param_types, 56
- check_phylogeny, 56, 335
- check_rename_fun, 57, 57, 199, 233, 375, 376
- check_rln_clock_model, 58
- check_screenlog, 58

- check_screenlog_names, 59
- check_screenlog_values, 60
- check_site_model, 60
- check_site_model_names, 62
- check_site_model_types, 63
- check_site_models, 61
- check_store_every, 63
- check_strict_clock_model, 64
- check_tn93_site_model, 64
- check_tn93_site_model_names, 65
- check_tracelog, 66
- check_tracelog_names, 66
- check_tracelog_values, 67
- check_tree_prior, 69
- check_tree_priors, 70
- check_treelog, 67, 185
- check_treelog_names, 68
- check_treelog_values, 68
- clock_model_to_xml_operators, 71
- clock_model_to_xml_prior_distr, 71
- clock_model_to_xml_state, 72
- clock_model_to_xml_tracelog, 73
- clock_model_to_xml_treelogger, 74
- clock_rate_param_to_xml, 74
- compare_lines, 75
- count_trailing_spaces, 76
- create_alpha_param, 12, 77, 100, 117, 126, 146, 197
- create_bd_tree_prior, 32, 78, 188, 189, 197, 252, 273
- create_beast2_beast_xml, 79
- create_beast2_input, 80, 85
- create_beast2_input_beast, 81, 91
- create_beast2_input_data, 82, 82
- create_beast2_input_data_sequences, 83
- create_beast2_input_distr, 84, 95
- create_beast2_input_distr_lh, 85
- create_beast2_input_distr_prior, 86
- create_beast2_input_file, 81, 87, 90
- create_beast2_input_file_from_model, 88, 89, 91
- create_beast2_input_from_model, 81, 82, 90, 90
- create_beast2_input_init, 92, 95
- create_beast2_input_map, 82, 93
- create_beast2_input_operators, 93, 95
- create_beast2_input_run, 82, 94
- create_beast2_input_state, 95, 95, 96
- create_beauti_options, 12, 34, 38, 39, 75, 79, 81, 83, 84, 88, 93, 97, 98, 99, 124, 174, 197, 201–206, 212, 274, 361, 363–372
- create_beauti_options_v2_4, 97, 98
- create_beauti_options_v2_6, 97, 99
- create_beautier_tempfolder, 96
- create_beta_distr, 77, 99, 101, 115, 202, 253, 275, 293
- create_beta_param, 34, 100, 101, 117, 126, 146, 197
- create_branch_rate_model_rln_xml, 102, 105
- create_branch_rate_model_sc_xml, 103, 105
- create_branch_rate_model_stuff_xml, 103
- create_branch_rate_model_xml, 102, 103, 104, 187
- create_cbs_tree_prior, 35, 105, 105, 188, 189, 197, 198, 277
- create_ccp_tree_prior, 35, 106, 188, 189, 197, 253, 278
- create_cep_tree_prior, 37, 107, 188, 189, 197, 254, 279
- create_clock_model, 30, 39, 40, 58, 64, 74, 80, 88, 90, 108, 110, 111, 124, 142, 174, 177, 197, 198, 208, 220, 228, 255, 280, 295, 344, 348
- create_clock_model_from_name, 111
- create_clock_model_rln (create_rln_clock_model), 156
- create_clock_model_strict (create_strict_clock_model), 167
- create_clock_models, 109, 110, 221
- create_clock_models_from_names, 110
- create_clock_rate_param, 75, 112, 146, 167, 198
- create_clock_rate_state_node_parameter_xml, 113
- create_data_xml, 114
- create_distr, 78, 100, 106, 107, 115, 116, 118–121, 126, 131, 133, 137, 141, 145, 148, 156, 167, 192, 194, 201, 223, 255, 295
- create_distr_beta (create_beta_distr), 99

- create_distr_exp (create_exp_distr), 116
- create_distr_gamma
 - (create_gamma_distr), 117
- create_distr_inv_gamma
 - (create_inv_gamma_distr), 125
- create_distr_laplace
 - (create_laplace_distr), 130
- create_distr_log_normal
 - (create_log_normal_distr), 133
- create_distr_normal
 - (create_normal_distr), 141
- create_distr_one_div_x
 - (create_one_div_x_distr), 145
- create_distr_poisson
 - (create_poisson_distr), 147
- create_distr_uniform
 - (create_uniform_distr), 192
- create_exp_distr, 115, 116, 136, 202, 256, 285, 296
- create_gamma_distr, 77, 101, 115, 117, 198, 212, 256, 287
- create_gamma_site_model, 44, 45, 118, 119, 120, 123, 127, 161, 181, 198, 214, 226, 227, 257, 297
- create_gtr_site_model, 45, 46, 120, 161, 162, 198, 225, 258, 286, 289, 297
- create_gtr_subst_model_xml, 122
- create_hky_site_model, 122, 161, 162, 198, 225, 259, 286, 290, 298
- create_hky_subst_model_xml, 123
- create_inference_model, 36, 43, 47, 48, 71–73, 82, 84, 85, 87, 89, 91–95, 102–104, 113, 124, 132, 142, 143, 152, 159, 165, 166, 168, 169, 174, 184, 186, 187, 190, 191, 199, 229, 245–249, 260, 263, 357–359, 361, 362, 374, 378, 379, 384–387, 389, 391, 393
- create_inv_gamma_distr, 77, 101, 115, 125, 203, 260, 299, 308
- create_jc69_site_model, 127, 161, 162, 199, 261, 300
- create_jc69_subst_model_xml, 128
- create_kappa_1_param, 128, 146, 181, 311, 364
- create_kappa_2_param, 129, 146, 181, 312, 364
- create_lambda_param, 129, 146, 147, 313, 365
- create_laplace_distr, 115, 130, 139, 157, 203, 262, 301, 314
- create_log_normal_distr, 115, 123, 133, 140, 170, 181, 204, 262, 301, 315
- create_loggers_xml, 95, 132
- create_m_param, 133, 139, 146, 199, 323, 361
- create_mcmc, 14, 49–51, 54, 81, 88, 90, 124, 134, 142, 144, 174, 176, 177, 199, 230, 283, 316, 355, 356, 376
- create_mcmc_nested_sampling
 - (create_ns_mcmc), 143
- create_mean_param, 116, 136, 141, 146, 318, 365
- create_mrca_prior, 28, 29, 48, 51–53, 80, 88, 124, 137, 137, 174, 199, 246, 263, 319, 320, 360
- create_mu_param, 131, 138, 146, 322, 366
- create_normal_distr, 115, 136, 141, 160, 204, 264, 302, 324
- create_ns_inference_model, 125, 142, 177
- create_ns_mcmc, 49–51, 54, 81, 88, 124, 135, 142, 143, 174, 177, 178, 199, 230, 283, 317, 355, 356, 376
- create_one_div_x_distr, 115, 145, 205, 265, 303, 326
- create_param, 54–56, 77, 101, 112, 130, 136, 139, 140, 146, 149–151, 153–155, 158, 160, 170, 199, 265, 303, 332, 363
- create_param_alpha
 - (create_alpha_param), 77
- create_param_beta (create_beta_param), 101
- create_param_clock_rate
 - (create_clock_rate_param), 112
- create_param_kappa_1
 - (create_kappa_1_param), 128
- create_param_kappa_2
 - (create_kappa_2_param), 129
- create_param_lambda
 - (create_lambda_param), 129
- create_param_m (create_m_param), 139
- create_param_mean (create_mean_param), 136
- create_param_mu (create_mu_param), 138
- create_param_rate_ac
 - (create_rate_ac_param), 148

- create_param_rate_ag
(create_rate_ag_param), 149
- create_param_rate_at
(create_rate_at_param), 150
- create_param_rate_cg
(create_rate_cg_param), 152
- create_param_rate_ct
(create_rate_ct_param), 153
- create_param_rate_gt
(create_rate_gt_param), 155
- create_param_s (create_s_param), 170
- create_param_scale
(create_scale_param), 157
- create_param_sigma
(create_sigma_param), 160
- create_poisson_distr, 115, 130, 147, 205,
266, 304, 335
- create_rate_ac_param, 121, 146, 148, 336,
367
- create_rate_ag_param, 121, 146, 149, 338,
367
- create_rate_at_param, 121, 146, 150, 339,
368
- create_rate_categories_state_node_xml,
151
- create_rate_cg_param, 121, 146, 152, 340,
369
- create_rate_ct_param, 121, 146, 153, 341,
369
- create_rate_gt_param, 121, 146, 155, 342,
370
- create_rln_clock_model, 108, 109, 156,
200, 266, 304, 343, 377
- create_s_param, 133, 146, 170, 371
- create_scale_param, 131, 146, 157, 371
- create_screenlog, 15, 59, 60, 135, 144, 158,
171, 175, 178, 200
- create_screenlog_xml, 132, 159
- create_sigma_param, 141, 146, 160, 372
- create_site_model, 31, 60–63, 80, 88, 90,
122, 124, 128, 142, 161, 162–164,
174, 177, 182, 200, 213, 234, 235,
237, 238, 267, 305, 347, 380–383
- create_site_model_from_name, 164
- create_site_model_gtr
(create_gtr_site_model), 120
- create_site_model_hky
(create_hky_site_model), 122
- create_site_model_jc69
(create_jc69_site_model), 127
- create_site_model_parameters_xml, 165,
166
- create_site_model_tn93
(create_tn93_site_model), 181
- create_site_model_xml, 166, 187
- create_site_models, 162, 236
- create_site_models_from_names, 163
- create_strict_clock_model, 108, 109, 112,
167, 200, 268, 305, 348
- create_strict_clock_rate_scaler_operator_xml,
168
- create_subst_model_xml, 166, 169
- create_temp_screenlog_filename, 171
- create_temp_tracelog_filename, 172
- create_temp_treelog_filename, 173
- create_test_inference_model, 125, 173,
177
- create_test_mcmc, 135, 175
- create_test_ns_inference_model, 143,
174, 176
- create_test_ns_mcmc, 144, 177
- create_test_screenlog, 179
- create_test_tracelog, 179
- create_test_treelog, 180
- create_tn93_site_model, 64, 65, 161, 162,
181, 200, 225, 269, 286, 306, 350
- create_tn93_subst_model_xml, 182
- create_tracelog, 16, 66, 67, 135, 144, 172,
175, 178, 183, 200
- create_tracelog_xml, 132, 183, 381, 387,
389
- create_trait_set_string, 184
- create_tree_likelihood_distr_xml, 86,
186
- create_tree_prior, 69, 70, 80, 88, 90, 124,
142, 174, 177, 188, 200, 201, 231,
239, 240, 242, 243, 270, 351, 388,
390, 391
- create_tree_prior_bd
(create_bd_tree_prior), 78
- create_tree_prior_cbs
(create_cbs_tree_prior), 105
- create_tree_prior_ccp
(create_ccp_tree_prior), 106
- create_tree_prior_cep
(create_cep_tree_prior), 107

- create_tree_prior_yule
 - (create_yule_tree_prior), 193
- create_tree_priors, 189, 241
- create_treelog, 17, 67–69, 135, 144, 173, 175, 178, 185, 200
- create_treelog_xml, 132, 186
- create_uclid_mean_state_node_param_xml, 190
- create_uclid_stdev_state_node_param_xml, 191
- create_uniform_distr, 115, 192, 206, 270, 307, 352
- create_xml_declaration, 79, 81, 91, 193, 379
- create_yule_tree_prior, 31, 188, 189, 193, 201, 271, 354, 394

- default_parameters_doc, 194
- default_params_doc, 195
- distr_to_xml, 201
- distr_to_xml_beta, 202
- distr_to_xml_exp, 202
- distr_to_xml_inv_gamma, 203
- distr_to_xml_laplace, 203
- distr_to_xml_log_normal, 204
- distr_to_xml_normal, 204
- distr_to_xml_one_div_x, 205
- distr_to_xml_poisson, 205
- distr_to_xml_uniform, 206

- extract_xml_loggers_from_lines, 206
- extract_xml_operators_from_lines, 207
- extract_xml_section_from_lines, 207

- FALSE, 274, 303, 330–332
- fasta_file_to_sequences, 208
- find_clock_model, 208
- find_first_regex_line, 209
- find_first_xml_opening_tag_line, 209
- find_last_regex_line, 210
- find_last_xml_closing_tag_line, 211
- freq_equilibrium_to_xml, 211

- gamma_distr_to_xml, 212
- gamma_site_model_to_xml_prior_distr, 213
- gamma_site_model_to_xml_state, 214
- gamma_site_models_to_xml_prior_distr, 213

- get_alignment_id, 38, 79, 106–109, 120, 123, 127, 137, 161, 180, 181, 183, 194, 197, 214
- get_alignment_ids, 215, 216
- get_alignment_ids_from_fasta_filenames, 105, 114, 156, 167, 198, 216, 216, 267, 270
- get_beautier_folder, 217
- get_beautier_path, 217, 219
- get_beautier_paths, 218, 218
- get_beautier_tempfilename, 219
- get_clock_model_name, 220
- get_clock_model_names, 110, 111, 197, 221
- get_clock_models_ids, 220
- get_crown_age, 222
- get_distr_n_params, 223
- get_distr_names, 222
- get_fasta_filename, 28, 29, 43, 80, 82, 83, 88, 89, 91, 94, 132, 184, 198, 199, 214, 216, 224, 255, 260, 263, 319, 320
- get_file_base_sans_ext, 224
- get_freq_equilibrium_names, 225
- get_gamma_site_model_n_distrs, 226
- get_gamma_site_model_n_params, 227
- get_has_non_strict_clock_model, 228
- get_inference_model_filenames, 228
- get_log_modes, 159, 179, 180, 183, 185, 199, 229
- get_log_sorts, 159, 179, 180, 183, 185, 200, 229
- get_mcmc_filenames, 230
- get_n_taxa, 230
- get_operator_id_pre, 231
- get_param_names, 232
- get_remove_dir_fun, 57, 200, 233, 375, 376
- get_remove_hex_fun, 57, 200, 233, 375, 376
- get_replace_dir_fun, 57, 200, 234, 375, 376
- get_site_model_n_distrs, 237
- get_site_model_n_params, 238
- get_site_model_names, 163, 164, 200, 236
- get_site_models_n_distrs, 234
- get_site_models_n_params, 235
- get_taxa_names, 53, 137, 200, 239
- get_tree_prior_n_distrs, 242
- get_tree_prior_n_params, 243
- get_tree_prior_names, 200, 241

- get_tree_priors_n_distrs, 239
- get_tree_priors_n_params, 240
- get_xml_closing_tag, 244
- get_xml_opening_tag, 244

- has_mrca_prior, 245
- has_mrca_prior_with_distr, 246
- has_rln_clock_model, 247
- has_strict_clock_model, 248
- has_tip_dating, 249
- has_xml_closing_tag, 250
- has_xml_opening_tag, 250
- has_xml_short_closing_tag, 251

- indent, 252
- init_bd_tree_prior, 252
- init_beta_distr, 253
- init_ccp_tree_prior, 253
- init_cep_tree_prior, 254
- init_clock_models, 254
- init_distr, 255
- init_exp_distr, 256
- init_gamma_distr, 256
- init_gamma_site_model, 257
- init_gtr_site_model, 258
- init_hky_site_model, 259
- init_inference_model, 259
- init_inv_gamma_distr, 260
- init_jc69_site_model, 261
- init_laplace_distr, 262
- init_log_normal_distr, 262
- init_mrca_prior, 263
- init_mrca_priors, 263
- init_normal_distr, 264
- init_one_div_x_distr, 264
- init_param, 265
- init_poisson_distr, 266
- init_rln_clock_model, 266
- init_site_models, 267
- init_strict_clock_model, 268
- init_tn93_site_model, 269
- init_tree_priors, 270
- init_uniform_distr, 270
- init_yule_tree_prior, 271
- interspace, 271
- is_alpha_param, 272
- is_bd_tree_prior, 273
- is_beauti_options, 274
- is_beta_distr, 275, 284
- is_beta_param, 276
- is_cbs_tree_prior, 277
- is_ccp_tree_prior, 278
- is_cep_tree_prior, 279
- is_clock_model, 280
- is_clock_model_name, 281
- is_clock_rate_param, 281
- is_default_mcmc, 282
- is_distr, 275, 283, 285, 287, 309, 315, 324, 326, 336, 353
- is_distr_name, 284
- is_exp_distr, 284, 285
- is_freq_equilibrium_name, 286
- is_gamma_distr, 284, 287
- is_gamma_site_model, 288
- is_gtr_site_model, 289
- is_hky_site_model, 290
- is_id, 25, 291
- is_in_patterns, 309
- is_inference_model, 292
- is_init_bd_tree_prior, 292
- is_init_beta_distr, 293
- is_init_cbs_tree_prior, 293
- is_init_ccp_tree_prior, 294
- is_init_cep_tree_prior, 294
- is_init_clock_model, 295
- is_init_distr, 295
- is_init_exp_distr, 296
- is_init_gamma_distr, 296
- is_init_gamma_site_model, 297
- is_init_gtr_site_model, 297
- is_init_hky_site_model, 298
- is_init_inv_gamma_distr, 299
- is_init_jc69_site_model, 300
- is_init_laplace_distr, 301
- is_init_log_normal_distr, 301
- is_init_mrca_prior, 302
- is_init_normal_distr, 302
- is_init_one_div_x_distr, 303
- is_init_param, 303
- is_init_poisson_distr, 304
- is_init_rln_clock_model, 304
- is_init_site_model, 305
- is_init_strict_clock_model, 305
- is_init_tn93_site_model, 306
- is_init_tree_prior, 306
- is_init_uniform_distr, 307
- is_init_yule_tree_prior, 308

- is_inv_gamma_distr, 284, 308
- is_jc69_site_model, 310
- is_kappa_1_param, 311
- is_kappa_2_param, 312
- is_lambda_param, 313
- is_laplace_distr, 284, 314
- is_log_normal_distr, 284, 315
- is_m_param, 323
- is_mcmc, 316
- is_mcmc_nested_sampling, 317
- is_mean_param, 318
- is_mrca_align_id_in_fasta, 320
- is_mrca_align_ids_in_fastas, 319
- is_mrca_prior, 320
- is_mrca_prior_with_distr, 321
- is_mu_param, 322
- is_nested_sampling_mcmc
(is_mcmc_nested_sampling), 317
- is_normal_distr, 284, 324
- is_on_appveyor, 330
- is_on_ci, 330
- is_on_github_actions, 331
- is_on_travis, 332
- is_one_bool, 325
- is_one_div_x_distr, 284, 326
- is_one_double, 327
- is_one_int, 328
- is_one_na, 329
- is_param, 332
- is_param_name, 333
- is_phylo, 334
- is_poisson_distr, 284, 335
- is_rate_ac_param, 336
- is_rate_ag_param, 337
- is_rate_at_param, 338
- is_rate_cg_param, 340
- is_rate_ct_param, 341
- is_rate_gt_param, 342
- is_rln_clock_model, 343
- is_s_param, 349
- is_scale_param, 344
- is_sigma_param, 345
- is_site_model, 346
- is_site_model_name, 347
- is_strict_clock_model, 348
- is_tn93_site_model, 350
- is_tree_prior, 351
- is_tree_prior_name, 352
- is_uniform_distr, 284, 352
- is_xml, 353
- is_yule_tree_prior, 354
- m_param_to_xml, 361
- mcmc_to_xml_run, 355
- mcmc_to_xml_run_default, 355
- mcmc_to_xml_run_nested_sampling, 356
- mrca_prior_to_xml_prior_distr, 358
- mrca_prior_to_xml_state, 359
- mrca_prior_to_xml_taxonset, 360
- mrca_prior_to_xml_tracelog, 360
- mrca_priors_to_xml_prior_distr, 357
- NA, 42, 52, 57, 137, 180, 183, 197, 199, 228,
233, 375, 376
- no_taxa_to_xml_tree, 362
- parameter_to_xml, 363
- parameter_to_xml_kappa_1, 363
- parameter_to_xml_kappa_2, 364
- parameter_to_xml_lambda, 365
- parameter_to_xml_mean, 365
- parameter_to_xml_mu, 366
- parameter_to_xml_rate_ac, 366
- parameter_to_xml_rate_ag, 367
- parameter_to_xml_rate_at, 368
- parameter_to_xml_rate_cg, 368
- parameter_to_xml_rate_ct, 369
- parameter_to_xml_rate_gt, 370
- parameter_to_xml_s, 370
- parameter_to_xml_scale, 371
- parameter_to_xml_sigma, 372
- remove_beautier_folder, 41, 372, 373
- remove_empty_lines, 373
- remove_multiline, 374
- rename_inference_model_filenames, 36,
43, 47, 71–73, 82, 84, 85, 87, 90–95,
102–104, 113, 132, 152, 159, 165,
166, 168, 169, 184, 186, 187, 190,
191, 199, 229, 245–249, 260, 263,
357–359, 361, 362, 374, 374, 378,
379, 384–387, 389, 391, 393
- rename_mcmc_filenames, 49–51, 54, 81, 88,
124, 135, 142, 174, 177, 199, 230,
283, 355, 356, 375, 376
- return, 57, 199, 375, 376
- rln_clock_model_to_xml_mean_rate_prior,
377

rln_clock_model_to_xml_prior_distr,
 377
rnd_phylo_to_xml_init, 379

site_model_to_xml_operators, 382
site_model_to_xml_prior_distr, 382
site_model_to_xml_state, 383
site_model_to_xml_tracelog, 383
site_models_to_xml_operators, 380
site_models_to_xml_prior_distr, 380
site_models_to_xml_tracelog, 381, 384
stop, 14–17, 38, 40, 42, 48, 52–54, 57, 58, 61,
 63, 66, 67, 70, 220, 245, 372

taxa_to_xml_tree, 384
tempfile, 219, 233
tipdate_taxa_to_xml_trait, 385
tipdate_taxa_to_xml_tree, 386
tree_model_to_tracelog_xml, 386
tree_prior_to_xml_operators, 389
tree_prior_to_xml_prior_distr, 390
tree_prior_to_xml_state, 391
tree_prior_to_xml_tracelog, 391
tree_priors_to_xml_prior_distr, 387
tree_priors_to_xml_tracelog, 388, 392
TRUE, 135, 159, 175, 179, 180, 183, 185, 200,
 274, 303, 330–332

unindent, 392

yule_tree_prior_to_xml_operators, 393
yule_tree_prior_to_xml_prior_distr,
 393