Package ‘PCMBaseCpp’

July 21, 2025

Type Package

Title Fast Likelihood Calculation for Phylogenetic Comparative Models

Version 0.1.11

Maintainer Venelin Mitov <`vmitov@gmail.com`>

Description Provides a C++ backend for multivariate phylogenetic comparative models implemented in the R-package 'PCMBase'. Can be used in combination with 'PCMBase' to enable fast and parallel likelihood calculation. Implements the pruning likelihood calculation algorithm described in Mitov et al. (2020) <doi:10.1016/j.tpb.2019.11.005>. Uses the 'SPLITT' C++ library for parallel tree traversal described in Mitov and Stadler (2018) <doi:10.1111/2041-210X.13136>.

Encoding UTF-8

License GPL (>= 3.0)

LazyData true

Depends R (>= 3.1.0), Rcpp, methods

Imports PCMBase, data.table, abind

Suggests testthat, knitr, rmarkdown, covr

LinkingTo Rcpp, RcppArmadillo

ByteCompile yes

NeedsCompilation yes

URL `https://github.com/venelin/PCMBaseCpp`, `https://venelin.github.io`

BugReports `https://github.com/venelin/PCMBaseCpp/issues`

Repository CRAN

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.2.3

Author Venelin Mitov [aut, cre, cph] (venelin.github.io)

Date/Publication 2025-05-04 10:40:02 UTC

Contents

`benchmarkData` *Data for performing a benchmark*

Description

A dataset containing three triplets trees, trait-values and models to evaluate the likelihood calculation times for R and C++ implementations.

Usage

```
benchmarkData
```

Format

A data frame with 4 rows and 8 variables:

tree phylogenetic tree (phylo) with set edge.regimes member

model MGPM model used to simulate the data in X

X trait values

ll log-likelihood value

modelBM a random BM model

llBM log-likelihood value form modelBM

modelOU a random OU model

llOU log-likelihood value for modelOU

`benchmarkResults` *Results from running a performance benchmark on a personal computer including the time for parameter transformation*

Description

Results from running a performance benchmark on a personal computer including the time for parameter transformation

Usage

```
benchmarkResults
```

Format

A data.table

`benchmarkResultsNoTransform` *Results from running a performance benchmark on a personal computer excluding the time for parameter transformation*

Description

Results from running a performance benchmark on a personal computer excluding the time for parameter transformation

Usage

```
benchmarkResultsNoTransform
```

Format

A data.table

`BenchmarkRvsCpp` *A log-likelihood calculation time comparison for different numbers of traits and option-sets*

Description

A log-likelihood calculation time comparison for different numbers of traits and option-sets

Usage

```
BenchmarkRvsCpp(
  ks = c(1, 2, 4, 8),
  includeR = TRUE,
  includeTransformationTime = TRUE,
  optionSets = NULL,
  includeParallelMode = TRUE,
  doProf = FALSE,
  RprofR.out = "RprofR.out",
  RprofCpp.out = "RprofCpp.out",
  verbose = FALSE
)
```

Arguments

`ks`	a vector of positive integers, denoting different numbers of traits. Default: `c(1, 2, 4, 8)`.
`includeR`	logical (default TRUE) indicating if likelihood calculations in R should be included in the benchmark (can be slow).
`includeTransformationTime`	logical (default TRUE) indicating if the time for `PCMApplyTransformation` should be included in the benchmark.
`optionSets`	a named list of lists of PCM-options. If NULL (the default) the option set is set to `DefaultBenchmarkOptions(k, includeParallelMode)` for each `k` in `ks` (see the code in `PCMBaseCpp:::DefaultBenchmarkOptions`).
`includeParallelMode`	logical (default TRUE) indicating if the default optionSet should include parallel execution modes, i.e. setting the option PCMBase.Lmr.mode to 21 instead of 11. This argument is taken into account only with the argument `optionSets` set to NULL (the default).
`doProf`	logical indicating if profiling should be activated (see Rprof from the utils R-package). Default: FALSE. Additional arguments to Rprof can be specified by assigning lists of arguments to the options 'PCMBaseCpp.ArgsRprofR' and 'PCMBaseCpp.ArgsRprofCpp'. The default values for both options is `list(append = TRUE, line.profiling = TRUE)`.

`RprofR.out`, `RprofCpp.out`	character strings indicating Rprof.out files for the R and Cpp implementations; ignored if doProf is FALSE. Default values: 'RprofR.out' and 'Rprofcpp.out'.
`verbose`	logical indicating if log-messages should be printed to the console during the benchmark. Default FALSE.

Value

a data.table for results similar to the data.table returned from `MiniBenchmarkRvsCpp` with additional columns for k, option-set and the type of model.

`MiniBenchmarkRvsCpp` *Evaluate the likelihood calculation times for example trees and data*

Description

Evaluate the likelihood calculation times for example trees and data

Usage

```
MiniBenchmarkRvsCpp(
  data = PCMBaseCpp::benchmarkData,
  includeR = TRUE,
  includeTransformationTime = TRUE,
  nRepsCpp = 10L,
  listOptions = list(PCMBase.Lmr.mode = 11, PCMBase.Threshold.EV = 0,
    PCMBase.Threshold.SV = 0),
  doProf = FALSE,
  RprofR.out = "RprofR.out",
  RprofCpp.out = "RprofCpp.out"
)
```

Arguments

`data`	a 'data.frame' with at least the following columns: **tree:** a list column of phylo objects with an edge.part member set. **X:** a list column of k x N numerical matrices. **model:** a list column of PCM objects. Defaults: to 'benchmarkData', which is small data.table included with the PCMBaseCpp package.
`includeR`	logical (default TRUE) indicating if likelihood calculations in R should be included in the benchmark (can be slow).
`includeTransformationTime`	logical (default TRUE) indicating if the time for `PCMApplyTransformation` should be included in the benchmark.

`nRepsCpp`	: number of repetitions for the cpp likelihood calculation calls: a bigger value increases the precision of time estimation at the expense of longer running time for the benchmark. Defaults to 10.
`listOptions`	options to set before measuring the calculation times. Defaults to ‘list(PCMBase.Lmr.mode = 11, PCMBase.Threshold.EV = 0, PCMBase.Threshold.SV = 0)‘. ‘PCMBase.Lmr.mode‘ corresponds to the parallel traversal mode for the tree traversal algorithm (see this page for possible values).
`doProf`	logical indicating if profiling should be activated (see Rprof from the utils R-package). Default: FALSE. Additional arguments to Rprof can be specified by assigning lists of arguments to the options ’PCMBaseCpp.ArgsRprofR’ and ’PCMBaseCpp.ArgsRprofCpp’. The default values for both options is `list(append = TRUE, line.profiling = TRUE)`.
`RprofR.out`, `RprofCpp.out`	character strings indicating Rprof.out files for the R and Cpp implementations; ignored if doProf is FALSE. Default values: ’RprofR.out’ and ’Rprofcpp.out’.

Value

a data.frame.

Examples

```
library(PCMBase)
library(PCMBaseCpp)
library(data.table)

testData <- PCMBaseCpp::benchmarkData[1]
# original MGPM model
MiniBenchmarkRvsCpp(data = testData)

# original MGPM model and parallel mode
MiniBenchmarkRvsCpp(
data = testData,
listOptions = list(PCMBase.Lmr.mode = 21, PCMBase.Threshold.EV = 1e-9,
PCMBase.Threshold.SV = 1e-9))

# single-trait data, original MGPM model and single mode and enabled option
# PCMBase.Use1DClasses
MiniBenchmarkRvsCpp(
data = PCMBaseCpp::benchmarkData[1, list(
 tree,
 X = lapply(X, function(x) x[1,, drop=FALSE]),
 model = lapply(model, function(m) PCMExtractDimensions(m, dims = 1)))],
listOptions = list(
  PCMBase.Lmr.mode = 11,
  PCMBase.Threshold.EV = 1e-9,
  PCMBase.Threshold.SV = 1e-9,
  PCMBase.Use1DClasses = FALSE))
```

`PCListInt` *Converts the logical matrix pc into a list of vectors denoting the (0-based) TRUE-indices in each column*

Description

Converts the logical matrix pc into a list of vectors denoting the (0-based) TRUE-indices in each column

Usage

```
PCListInt(pc)
```

Arguments

`pc`	a logical matrix.

Value

a list

`PCMBaseCppIsADevRelease` *Check if the PCMBaseCpp version correpsonds to a dev release*

Description

This function is used during unit-testing, to disable some unit- tests which run extremely long or are consistently failing on some systems.

Usage

```
PCMBaseCppIsADevRelease()
```

Value

a logical

`PCMInfoCpp` *A S3 generic for creating C++ backend objects given a model, data and a tree.*

Description

Replace calls to PCMInfo() with this method in order to use C++ for likelihood calculation.

Usage

```
PCMInfoCpp(
  X,
  tree,
  model,
  SE = matrix(0, PCMNumTraits(model), PCMTreeNumTips(tree)),
 metaI = PCMInfo(X = X, tree = tree, model = model, SE = SE, verbose = verbose, preorder
    = PCMTreePreorderCpp(tree)),
  verbose = FALSE,
  ...
)
```

Arguments

`X`	a k x N numerical matrix with possible `NA` and `NaN` entries. Each column of X contains the measured trait values for one species (tip in tree). Missing values can be either not-available (`NA`) or not existing (`NaN`). These two values have are treated differently when calculating likelihoods: see `PCMPresentCoordinates`.
`tree`	a phylo object with N tips.
`model`	an S3 object specifying both, the model type (class, e.g. "OU") as well as the concrete model parameter values at which the likelihood is to be calculated (see also Details).
`SE`	a k x N matrix specifying the standard error for each measurement in X. Alternatively, a k x k x N cube specifying an upper triangular k x k Choleski factor of the variance covariance matrix for the measurement error for each node i=1, ..., N. Default: `matrix(0.0, PCMNumTraits(model), PCMTreeNumTips(tree))`.
`metaI`	a list returned from a call to `PCMInfo(X, tree, model, SE)`, containing meta-data such as N, M and k. Default: `PCMInfo(X, tree, model, verbose, preorder=PCMTreePreorderCp`
`verbose`	logical indicating if some debug-messages should be printed. Default: FALSE
`...`	passed to methods.

Value

a list to be passed to PCMLik as argument metaI.

Examples

```
metaICpp <- PCMInfoCpp(
PCMBase::PCMBaseTestObjects$traits.a.123,
PCMBase::PCMBaseTestObjects$tree.a,
PCMBase::PCMBaseTestObjects$model.a.123)
PCMBase::PCMLik(
PCMBase::PCMBaseTestObjects$traits.a.123,
PCMBase::PCMBaseTestObjects$tree.a,
PCMBase::PCMBaseTestObjects$model.a.123,
metaI = metaICpp)
```

`PCMParamGetFullVector` *Get a vector with all model parameters unrolled*

Description

Get a vector with all model parameters unrolled

Usage

```
PCMParamGetFullVector(model, ...)
```

Arguments

`model`	a PCM model object
`...`	passed to methods

Value

a numerical vector

Examples

```
PCMParamGetFullVector(PCMBase::PCMBaseTestObjects$model.a.123)
```

`PCMTreePreorderCpp` *Fast preorder of the edges in a tree*

Description

Fast preorder of the edges in a tree

Usage

```
PCMTreePreorderCpp(tree)
```

Arguments

`tree`	a phylo object

Value

an integer vector containing indices of rows in `tree$edge` in their preorder order.

Examples

```
PCMTreePreorderCpp(PCMBase::PCMBaseTestObjects$tree.a)
```

Index