Package: abcpp (via r-universe)

November 2, 2026

Type Package

Title Shared C++ Backend for Approximate Bayesian Computation

Version 1.0.0

Author Mengzhen Hu [aut, cre], Katalin Csillery [ctb, cph], Louisiane Lemaire [ctb, cph], Olivier Francois [ctb, cph], Michael Blum [ctb, cph]

Maintainer Mengzhen Hu <hmz1969a@gmail.com>

Description Provides a compact C++ backend for Approximate Bayesian Computation (ABC) with a thin R frontend. The current implementation is primarily a C++ reimplementation of offline ABC workflows provided by the R 'abc' package <doi:10.32614/CRAN.package.abc>, with the public R interface intentionally kept small and centered on abc() and summary() methods. The computational work is performed by shared C++ code. In addition to reproducing common 'abc' workflows, the package adds optional dimensionality reduction of summary statistics through Principal Component Analysis (PCA) and Partial Least Squares (PLS), following related ideas described by Bazin et al. (2010) <doi:10.1534/genetics.109.112391> and Wegmann et al. (2009) <doi:10.1534/genetics.109.102509>.

License GPL-3

URL https://github.com/yuki-961004/abcpp

BugReports https://github.com/yuki-961004/abcpp/issues

Encoding UTF-8

ByteCompile true

Depends R (>= 4.1.0)

Imports Rcpp

LinkingTo Rcpp

SystemRequirements C++17

RoxygenNote 8.0.0

Suggests abc, testthat

Repository https://yuki-961004.r-universe.dev

Date/Publication 2026-06-29 09:29:57 UTC

RemoteUrl https://github.com/yuki-961004/abcpp

RemoteRef HEAD

RemoteSha f520da3eaaafa92846735ad6892e9b7d96399d19

RemoteSubdir R

Contents

abc 2
control 4
summary.abcpp 5

Index **7**

---

abc *Parameter Estimation With Approximate Bayesian Computation (ABC)*

Description

Runs Approximate Bayesian Computation (ABC) through the shared C++ backend. The R interface is intentionally small: users provide ‘target‘, ‘params‘, ‘sumstats‘, and an optional nested ‘control‘ list.

Usage

```
abc(target, params, sumstats, control = list())
```

Arguments

`target`	A vector, one-row matrix, or two-dimensional matrix of observed summary statistics.
`params`	A vector, one-row matrix, matrix, or data frame of simulated parameter values. Each row corresponds to one simulation.
`sumstats`	A vector, matrix, data frame, or list of matrices of simulated summary statistics. Rows, or list elements for matrix-valued summaries, correspond to rows of ‘params‘.
`control`	Nested list of algorithm controls. See [control] for the full schema. Common fields include ‘method‘, ‘tol‘, ‘kernel‘, ‘hcorr‘, ‘transf‘, ‘prior.weights‘, ‘seed‘, ‘reduction‘, ‘n_comp‘, and ‘nnet‘.

Details

The algorithmic implementation lives in C++. The R layer only prepares the input matrices, merges the `control` list with defaults, calls the compiled backend, and attaches the `"abcpp"` class to the output.

The core ABC procedure calculates the Euclidean distance between the simulated summary statistics and the target summary statistics. Optionally, dimensionality reduction (Principal Component Analysis, PCA, or Partial Least Squares, PLS) can be applied to the summary statistics before calculating distances. A predefined proportion (`tol`) of the simulations closest to the target are retained. The `method` parameter in the `control` list specifies whether the accepted parameters are returned as-is (`"rejection"`) or adjusted using a regression model (`"loclinear"`, `"ridge"`, or `"neuralnet"`) to account for the discrepancy between the simulated and observed summary statistics.

`summary()` computes statistical summaries from the returned object.

Value

An object of class `"abcpp"` containing complete algorithm output. This object is a list containing the following components:

- `unadj.values`: A matrix of unadjusted accepted posterior samples.
- `adj.values`: A matrix of adjusted posterior samples when a regression method (e.g., `"loclinear"`, `"ridge"`, `"neuralnet"`) is used; otherwise `NULL`.
- `ss`: A matrix of accepted summary statistics.
- `weights`: A numeric vector of regression weights used for adjustments.
- `dist`: A numeric vector of Euclidean distances between accepted simulated and target summary statistics.
- `region`: A logical vector indicating which simulations were accepted (unadjusted).
- `method`: The ABC method used.
- `call`: The original matched function call.
- `names`: A list containing `parameter.names` and `statistics.names`.

See Also

[control], [summary.abcpp()]

Examples

```
set.seed(1004)
n <- 200L
params <- cbind(theta = stats::runif(n))
sumstats <- cbind(s = params[, 1] + stats::rnorm(n, sd = 0.05))
fit <- abc(
  target = c(0.5),
  params = params,
  sumstats = sumstats,
  control = list(tol = 0.1)
)
summary(fit)
```

`control` *ABC control options*

Description

Documents the nested ‘control‘ list accepted by [abc()]. The control list is merged with defaults, so users may provide only the fields they want to override.

Details

The default control is:

“‘r list(method = "rejection", tol = 0.01, kernel = "epanechnikov", hcorr = TRUE, transf = "none", logit.bounds = NULL, subset = NULL, prior.weights = NULL, seed = 1004, reduction = "none", n_comp = 0L, nnet = list(numnet = 10L, sizenet = 5L, lambda = c(0.0001, 0.001, 0.01), maxit = 500L, rang = 0.7, abstol = 1e-4, reltol = 1e-8, verbose = FALSE, skip = FALSE)) “‘

Control List Fields

`method` Character. Selects the ABC method. Supported values are `"rejection"`, `"loclinear"`, `"ridge"`, and `"neuralnet"`.

`tol` Numeric. The tolerance proportion in `(0, 1]`. It determines the proportion of simulations accepted based on distance to the target summary statistics.

`kernel` Character. Selects the regression weight kernel. Supported values are `"epanechnikov"` (default), `"rectangular"`, `"gaussian"`, `"triangular"`, `"biweight"`, and `"cosine"`.

`hcorr` Logical. Enables conditional heteroscedastic correction in regression adjustment methods (default is `TRUE`).

`transf` Character vector. Controls parameter transformation before adjustment. Use `"none"`, `"log"`, or `"logit"`. A single value is recycled across all parameters.

`logit.bounds` Matrix. Used when `transf = "logit"` to specify the bounds for each parameter. Each row corresponds to a parameter, with the first column being the lower bound and the second being the upper bound.

`subset` Logical vector. An optional mask to apply to simulations, selecting a subset to be considered.

`prior.weights` Numeric vector. Optional prior weights for simulated samples. When supplied, its length must match the number of rows in `params`. For regression adjustment methods, final regression weights are the distance-kernel weights multiplied by these prior weights.

`seed` Integer. The seed used by stochastic C++ components (default is `1004L`).

`reduction` Character. Optionally reduces summary statistics prior to ABC. Supported values are `"none"`, `"pca"` (Principal Component Analysis), and `"pls"` (Partial Least Squares).

`n_comp` Integer. The number of PCA or PLS components to retain when `reduction` is active. Use `0L` to let the backend decide the default.

`nnet` A nested list of settings for the neural network method (`method = "neuralnet"`):

- `numnet`: Integer. The number of neural networks to train (default `10L`).

- `sizenet`: Integer. The number of units in the hidden layer (default `5L`).
- `lambda`: Numeric vector. Weight decay parameters sampled uniformly across neural network fits.
- `maxit`: Integer. The maximum number of BFGS (Broyden-Fletcher-Goldfarb-Shanno) iterations per fit (default `500L`).
- `rang`: Numeric. Initial weights are randomized in `[-rang, rang]` (default `0.7`).
- `abstol`: Numeric. Absolute convergence tolerance (default `1e-4`).
- `reltol`: Numeric. Relative convergence tolerance (default `1e-8`).
- `verbose`: Logical. If `TRUE`, requests training progress from the backend.
- `skip`: Logical. If `TRUE`, adds direct input-to-output skip-layer connections.

See Also

[abc()]

Examples

```
target <- c(0.5)
params <- matrix(runif(100), ncol = 1)
sumstats <- matrix(runif(100), ncol = 1)
fit <- abc(
  target = target,
  params = params,
  sumstats = sumstats,
  control = list(tol = 0.1)
)

fit <- abc(
  target = target,
  params = params,
  sumstats = sumstats,
  control = list(
    method = "neuralnet",
    tol = 0.1,
    nnet = list(sizenet = 8, maxit = 1000)
  )
)
```

`summary.abcpp` *Summarize an abcpp Posterior*

Description

Summarizes posterior samples returned by [abc()].

Usage

```
## S3 method for class 'abcpp'
summary(object, unadj = FALSE, intvl = 0.95, ...)
```

Arguments

object	An object of class ‘"abcpp"‘ returned by [abc()].
unadj	Logical. If ‘TRUE‘, summarize unadjusted rejection samples even when regression-adjusted values are available.
intvl	Credible interval width used for lower and upper posterior summaries.
...	Additional arguments for S3 compatibility.

Details

The summary extracts the relevant posterior samples and calculates the minimum, lower interval endpoint, median, mean, upper interval endpoint, maximum, and standard deviation for each parameter.

For `method = "rejection"`, or when adjusted values are not available, unadjusted samples are always summarized. When an adjusted method is used (e.g., `"loclinear"`), the adjusted samples are summarized by default unless `unadj = TRUE` is specified.

Value

A list of class `"summary.abcpp"` containing the following components:

- `statistics`: A matrix of posterior summary statistics for each parameter. The rows include Minimum, Lower endpoint, Median, Mean, Upper endpoint, Maximum, and Standard Deviation.
- `unadjusted`: A logical value indicating whether the unadjusted samples were summarized.
- `interval`: The numeric credible interval width specified by the user.
- `method`: The Approximate Bayesian Computation (ABC) method inherited from the fitted `abcpp` object.

Examples

```
set.seed(1004)
n <- 200L
param <- cbind(theta = stats::runif(n))
sumstat <- cbind(s = param[, 1] + stats::rnorm(n, sd = 0.05))
fit <- abc(
  target = c(0.5),
  params = param,
  sumstats = sumstat,
  control = list(tol = 0.1)
)
summary(fit)
```

Index

abc, 2

control, 4

summary.abcpp, 5