Package 'abctools'

July 2, 2014

Type Package

Title Tools for ABC analyses

Version 0.3-2

Date 25/6/14

Description Tools for approximate Bayesian computation including summary statistic selection and assessing coverage

Depends R (>= 2.10), abc, abind, parallel, pls, plyr

Suggests ggplot2

License GPL (>= 2)

LazyLoad yes

SystemRequirements ABCtoolbox

Author Matt Nunes [aut, cre],Dennis Prangle [aut]

Maintainer Matt Nunes <m.nunes@lancaster.ac.uk>

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-06-25 11:11:13

R topics documented:

`abctools-package` *Tools for ABC analyses*

Description

Tools for approximate Bayesian computation including summary statistic selection and assessing coverage

Details

Package:	abctools
Type:	Package
Version:	0.3-2
Date:	25-6-2014
License:	GPL-2
LazyLoad:	yes

Author(s)

Matt Nunes and Dennis Prangle

Maintainer: Matthew Nunes <m.nunes@lancaster.ac.uk>

References

For details on methods for summary statistics selection, see

Blum, M. G. B, Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci.* **28**, Issue 2, 189–208.

Fearnhead, P. and Prangle, D. (2012) Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. B* **74**, Part 3, 1–28.

Joyce, P. and P. Marjoram (2008) Approximately sufficient statistics and Bayesian computation.

Stat. Appl. Gen. Mol. Biol. **7** Article 26.

Nunes, M. A. and Balding, D. J. (2010) On Optimal Selection of Summary Statistics for Approximate Bayesian Computation. *Stat. Appl. Gen. Mol. Biol.* **9**, Iss. 1, Art. 34.

Wegmann, D. et al. (2009) Efficient approximate Bayesian computation coupled With Markov chain Monte Carlo Without Likelihood. *Genetics* **182** (4), 1207–1218.

For extra scripts used for `pls.abc`, see:

Wegmann, D. et al. (2010) ABCtoolbox: A versatile toolkit for approximate Bayesian computations. *BMC Bioinf.* **11**, 116–123.

Details of ABC coverage methods appear in:

Prangle D., Blum M. G. B., Popovic G., Sisson S. A. (2013) Diagnostic tools of approximate Bayesian computation using the coverage property. *(preprint)* `arxiv.org/abs/1301.3166`

See Also

abc

`AS.select` *Summary statistics selection using approximate sufficiency.*

Description

This function uses approximate sufficiency to assess subsets of summary statistics for ABC inference.

Usage

```
AS.select(obs, param, sumstats, obspar=NULL, abcmethod=abc, grid=10, inturn=FALSE,
limit=ncol(sumstats), allow.none=TRUE, do.err=FALSE, final.dens=FALSE,
errfn=rsse,...)
```

Arguments

`obs`	(matrix of) observed summary statistics.
`param`	matrix of simulated model parameter values.
`sumstats`	matrix of simulated summary statistics.
`obspar`	optional observed parameters (for use to assess simulation performance).
`abcmethod`	a function to perform ABC inference, e.g. the `abc` function from package *abc*.
`grid`	the number of bins into which to divide the posterior sample for the approximate sufficiency calculation.
`inturn`	a boolean value indicating whether "bad" statistics should be dropped and tested sequentially (`inturn=TRUE`) or all at the end (`inturn=FALSE`).

limit	an optional integer indicating whether to limit summary selection to subsets of a maximum size.
allow.none	a boolean values indicating whether an empty subset of statistics is considered in the selection procedure.
do.err	a boolean value indicating whether the simulation error should be returned. Note: if do.err=TRUE, obspar must be supplied.
final.dens	a boolean value indicating whether the posterior sample should be returned.
errfn	an error function to assess ABC inference performance.
...	any other optional arguments to the ABC inference procedure (e.g. arguments to the abc function).

Details

The summary selection procedure works by sequentially testing randomly chosen statistics for inclusion, using the ratio of ABC posterior samples to determine whether a statistic is added. Since adding a statistic may result in a suboptimal subset of summaries, the included statistics are then individually dropped and retested, to determine whether a smaller subset of statistics is equally / more informative than the accepted set of statistics.

Value

A list with the following components:

best	the final subset of included statistics.
err	simulation error (if obspar is supplied and do.err=TRUE).
post.sample	an array of dimension nacc x npar x ndatasets giving the posterior sample for each observed dataset. Not returned if final.dens=FALSE.

Note

The approximate sufficiency techniques described here are only suitable for single parameters only.

Author(s)

Matt Nunes

References

Blum, M. G. B, Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci. (to appear)*.

Joyce, P. and P. Marjoram (2008) Approximately sufficient statistics and Bayesian computation. *Stat. Appl. Gen. Mol. Biol.* **7** Article 26.

See Also

AS.test

Examples

```
# load example data:

data(coal)
data(coalobs)

param<-coal[,2]
simstats<-coal[,4:6]

# use matrix below just in case to preserve dimensions.

obsstats<-matrix(coalobs[1,4:6],nrow=1)

# example of AS.select:

## Not run:
tmp<-AS.select(obsstats, param, simstats,tol=.1,method="neuralnet",nument=5,
allow.none=FALSE,inturn=TRUE)

tmp$best

## End(Not run)
```

`AS.test` *Test for relative approximate sufficiency between two posterior samples.*

Description

The function tests to determine adding a (set of) statistics is informative in ABC inference.

Usage

```
AS.test(grid = 10, x1, x2, supp)
```

Arguments

`grid`	the number of bins into which to divide the posterior sample for the approximate sufficiency calculation.
`x1`	the posterior sample using the first set of summary statistics.
`x2`	the posterior sample using the second (alternative) set of summary statistics.
`supp`	the "support" of the prior (e.g. uniform bounds).

Details

After dividing each posterior sample into a number of bins (specified by `grid`), the function computes the ratio of the posterior densities. This is seen as a measure of information added (sufficiency) by using the alternative posterior sample instead of the first posterior sample. If the ratio exceeds a particular threshold (a number of standard deviations away from the expected counts in each bin), then the alternative set of summaries is seen as being more informative.

Value

`extreme`	a boolean value indicating whether the alternative posterior sample is more informative than the first (i.e. the extra summary statistics add information.

Author(s)

Matt Nunes

References

Blum, M. G. B, Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci. (to appear)*.

Joyce, P. and P. Marjoram (2008) Approximately sufficient statistics and Bayesian computation. *Stat. Appl. Gen. Mol. Biol.* **7** Article 26.

See Also

`AS.select`

Examples

```
#create two fake posterior samples:

x1<-runif(10000)
x2<-rnorm(10000)

AS.test(x1=x1,x2=x2,supp=range(x2))
```

---

coal *Examples of coalescent data*

Description

Data generated from a coalescent model for genetic variation.

Usage

```
data(coal)
```

Format

Matrix of of parameters and summary statistics from a coalescent model

---

`combmat` *table of combinations*

Description

This function creates a table of binary masks representing combinations of statistics.

Usage

```
combmat(n,limit = NULL)
```

Arguments

`n`	number of statistics
`limit`	an optional (integer) value indicating whether to limit the table to subsets up to a certain size.

Value

`m`	The matrix of binary masks.

Author(s)

Matt Nunes

Examples

```
#
# Find all binary masks of a set of statistics {C1,C2,C3,C4},
# listing all singlets, pairs, triples and then the whole set:

combmat(4,TRUE)
```

cov.pi *Coverage property diagnostics*

Description

These functions produce diagnostic statistics for an ABC analysis to judge when the tolerance level is small enough to produce roughly no approximation error. This is done by running analyses for many test data sets and assessing whether the results satisfy the "coverage property" (roughly speaking: credible intervals have the claimed coverage levels).

Usage

```
cov.pi(param, sumstat, testsets, tol, eps, diagnostics = c(), multicore =
FALSE, cores, method = "rejection", nacc.min=20, ...)

cov.mc(index, sumstat, testsets, tol, eps, diagnostics = c(), multicore =
FALSE, cores, method = "rejection", nacc.min=20, ...)

covstats.pi(raw, diagnostics = c("KS", "CGR"), nacc.min = 20)

covstats.mc(raw, index, diagnostics = c("freq", "loglik.binary", "loglik.multi"),
nacc.min = 20)
```

Arguments

param	A data frame of parameter values. It must have the same number of rows as sumstat and contain numeric values only.
index	A vector of model indices. Any value which can be convert to factor is ok (e.g. character or numeric entries). It must have the same length as nrow(sumstat).
sumstat	A data frame of summary statistic values whose the ith row has been simulated using param[i,] or index[i].
testsets	A numeric vector giving the rows of sumstat to be used as pseudo-observed data to test the coverage property.
tol	A vector of proportions of ABC acceptances which will be investigated.
eps	A vector of ABC thresholds which will be investigated. These are used when tol is missing. One of eps and tol must be supplied.
diagnostics	A character vector containing diagnostics to be calculated. Allowable values for parameter inference are "KL" (Kullback-Leibler based test) or "CGR" (Cook, Gelman and Rubin test). Allowable values for model choice are "freq" (a separate frequency-based test for each model), "loglik.binary" (a separate log-likelihood based test for each model) or "loglik.multi" (single log-likelihood based test). If diagnostics is empty only raw results will be returned.
multicore	Whether to use the parallel package to perform analyses of test datasets in parallel.
cores	Number of cores to use when multicore==TRUE.

method	Method used for ABC analysis. The default is "rejection". For alternatives see abc (parameter inference) or postpr (model choice).
nacc.min	Minimum number of ABC acceptances required to compute diagnostics. See Values for details of how this is used.
...	Extra arguments to be supplied to the function performing abc analysis i.e. abc (parameter inference) or postpr (model choice).
raw	Raw output component from cov.pi or cov.mc for which diagnostics are to be calculated.

Details

These functions are intended to be applied as follows (i) random models/parameters are generated, data sets simulated for each and summary statistics calculated (ii) these are input to cov.pi (parameter inference) or cov.mc (model choice) which outputs raw results and diagnostics (see below) (iii) the output can be passed to covstats.pi or covstats.mc if further diagnostics are required later (or to find diagnostics for a subset of test sets).

The cov.pi and cov.mc functions operate by performing many ABC analyses. The user specifies which datasets amongst those simulated will be analysed. The results of each analysis are compared to the known model/parameters which produced the data to see whether they are consistent in a particular sense (i.e. if the coverage property is satisfied). Various diagnostics are provided to judge this easily, and determine what happens as the ABC threshold is varied. Raw results are also returned which can be investigated in more detail

All ABC analyses use a rejection sampling algorithm implemented by the abc package. The user may specify regression post-processing as part of this analysis.

Value

Output of cov.pi or cov.mc is a list of two data frames, raw and diag. The covstats.pi and covstats.mc functions just return the latter data frame.

For parameter inference, raw contains estimated cdfs (referred to as p0 estimates in Prangle et al 2013) of the true parameter values for each input configuration (i.e. for every tol/eps value at every test dataset). diag is a data frame of tol/eps value, parameter name, diagnostic name and p-value. Here the p-value relates to the test statistic used as a diagnostic. It is NA if any analyses had fewer than nacc.min acceptances (Diagnostics based on a small number of acceptances can be misleading.)

For model choice, raw contains estimated model weights for each input configuration, and diag is a data frame of tol/eps value, model, diagnostic name and p-value (NA under the same conditions as before.)

In both cases, raw also reports the number of acceptances. Note that raw contains p0 estimates/weights of NA if regression correction is requested but there are too few acceptances to compute it.

Author(s)

Dennis Prangle

References

Prangle D., Blum M. G. B., Popovic G., Sisson S. A. (2013) Diagnostic tools of approximate Bayesian computation using the coverage property. *(preprint)* arxiv.org/abs/1301.3166

See Also

mc.ci for a diagnostic plot of raw model choice results

abc and postpr to perform ABC for a given dataset

Examples

```
##The examples below are chosen to run relatively quickly (<5 mins)
##and do not represent recommended tuning choices.
## Not run:
data(musigma2)
library(ggplot2)
##Parameter inference example
parameters <- data.frame(par.sim)
sumstats <- data.frame(stat.sim)
covdiag <- cov.pi(param=parameters, sumstat=sumstats, testsets=1:100,
tol=seq(0.1,1,by=0.1), diagnostics=c("KS"))

qplot(x=tol, y=pvalue, facets=.~parameter, data=covdiag$diag) #Plot of diagnostic results
qplot(x=mu, data=subset(covdiag$raw, tol==0.5)) #Plot of raw results for tol=0.5
qplot(x=sigma2, data=subset(covdiag$raw, tol==0.5)) #Plot of raw results for tol=0.5

cgrout <- covstats.pi(covdiag$raw, diagnostics="CGR") #Compute CGR statistic as well
qplot(x=tol, y=pvalue, facets=.~parameter, data=cgrout) #Plot CGR diagnostic

##Model choice example, based on simple simulated data
index <- sample(1:2, 1E4, replace=TRUE)
sumstat <- ifelse(index==1, rnorm(1E4,0,1), rnorm(1E4,0,rexp(1E4,1)))
sumstat <- data.frame(ss=sumstat)
covdiag <- cov.mc(index=index, sumstat=sumstat, testsets=1:100, tol=seq(0.1,1,by=0.1),
diagnostics=c("freq"))
qplot(x=tol, y=pvalue, data=covdiag$diag)
llout <- covstats.mc(covdiag$raw, index=index, diagnostics="loglik.binary")
qplot(x=tol, y=pvalue, data=llout)
mc.ci(covdiag$raw, tol=0.5, modname=1, modtrue=index[1:200])

## End(Not run)
```

---

mc.ci *Diagnostic plots for model choice coverage output*

Description

Plots credible interval estimates for raw model choice output from cov.mc. This is used to investigate whether the coverage property holds and validate whether diagnostic statistics are acting as intended.

Usage

```
  mc.ci(raw, tol, eps, modname, modtrue, nbins=5, bintype=c("interval",
"quantile"), bw=FALSE, ...)
```

Arguments

raw	The raw item from the list output by cov.mc.
tol	The value of tol to test.
eps	The value of eps to test. This is used when tol is missing. One of eps and tol must be supplied.
modname	The name of the model to test.
modtrue	Vector containing the true models generating the pseudo-observed data. i.e. modtrue[i] is the model generating dataset i.
nbins	Number of bins to display.
bintype	How to choose the bins (see Details).
bw	Whether to produce a black and white image. Default is FALSE. Colour is used to make different bins stand out.
...	Additional plotting arguments - anything that can be used by plot.

Details

This function provides a plot which can be used as an informal test of the model choice coverage hypothesis for a particular value of eps or tol and choice of model. The plot is more flexible than the diagnostics, but not suitable as the basis of a formal test.

For each pseudo-observed data set, the ABC probability that the model is modname is taken from raw, and the true model is taken from modtrue. The probabilities are binned into nbins intervals, either of equal length or based on nbins+1 equally spaced empirical quantiles. The function estimates the observed probability of modname within each bin using Bayesian inference for a binomial proportion using a uniform prior. The plot shows the mean and 95% credible interval plotted against predicted probabilities. Informally, the coverage property should be rejected if predicted values are too unlikely given the observed values.

Author(s)

Dennis Prangle

References

Prangle D., Blum M. G. B., Popovic G., Sisson S. A. (2013) Diagnostic tools of approximate Bayesian computation using the coverage property. *(preprint)* arxiv.org/abs/1301.3166

See Also

cov.mc to produce the input for this function

Examples

```
##The examples below are chosen to run relatively quickly (<5 mins)
##and do not represent recommended tuning choices.
## Not run:
index <- sample(1:2, 1E4, replace=TRUE)
sumstat <- ifelse(index==1, rnorm(1E4,0,1), rnorm(1E4,0,rexp(1E4,1)))
sumstat <- data.frame(ss=sumstat)
covdiag <- cov.mc(index=index, sumstat=sumstat, testsets=1:100, tol=seq(0.1,1,by=0.1),
diagnostics=c("freq"))
mc.ci(covdiag$raw, tol=0.5, modname=1, modtrue=index[1:100])

## End(Not run)
```

`mincrit` *Summary statistics selection by minimizing a posterior sample measure.*

Description

The function cycles through all possible subsets of summary statistics and computes a criterion from the posterior sample. The subset which achieves the minimum is chosen as the most informative subset.

Usage

```
mincrit(obs,param, sumstats, obspar=NULL, abcmethod=abc,crit=nn.ent,
sumsubs=1:ncol(sumstats), limit = length(sumsubs), do.only = NULL,
verbose = TRUE, do.crit = TRUE, do.err=FALSE,final.dens=FALSE,errfn=rsse, ...)
```

Arguments

`obs`	(matrix of) observed summary statistics.
`param`	matrix of simulated model parameter values.
`sumstats`	matrix of simulated summary statistics.
`obspar`	optional observed parameters (for use to assess simulation performance).
`abcmethod`	a function to perform ABC inference, e.g. the `abc` function from package *abc*.
`crit`	a function to minimize to measure information from a posterior sample, e.g. `nn.ent`.
`sumsubs`	an optional index into the summary statistics to limit summary selection to a specific subset of summaries.
`limit`	an optional integer indicating whether to limit summary selection to subsets of a maximum size.
`do.only`	an optional index into the summary statistics combination table. Can be used to limit entropy calculations to certain summary statistics subsets only.

verbose	a boolean value indicating whether informative statements should be printed to screen.
do.crit	a boolean value indicating whether the measure on the posterior sample should be returned.
do.err	a boolean value indicating whether the simulation error should be returned. Note: if do.err=TRUE, obspar must be supplied.
final.dens	a boolean value indicating whether the posterior sample should be returned.
errfn	an error function to assess ABC inference performance.
...	any other optional arguments to the ABC inference procedure (e.g. arguments to the abc function).

Details

The function uses a criterion (e.g.sample entropy) as a proxy for information in a posterior sample. The criterion for each possible subset of statistics is computed, and the best subset is judged as the one which minimises this vector of values.

Value

A list with the following components:

best	the best subset(s) of statistics.
critvals	the calculated criterion values (if do.crit=TRUE).
err	simulation error (if obspar is supplied and do.err=TRUE).
order	the subsets considered during the algorithm (same as the input do.only.
post.sample	an array of dimension nacc x npar x ndatasets giving the posterior sample for each observed dataset. Not returned if final.dens=FALSE.
sumsubs	an index into the subsets considered during the algorithm.

Warning

These functions are computationally intensive due to the cyclic ABC inference procedure.

Author(s)

Matt Nunes

References

Nunes, M. A. and Balding, D. J. (2010) On Optimal Selection of Summary Statistics for Approximate Bayesian Computation. *Stat. Appl. Gen. Mol. Biol.* **9**, Iss. 1, Art. 34.

See Also

nn.ent

Examples

```
# load example data:

data(coal)
data(coalobs)

param<-coal[,2]
simstats<-coal[,4:6]

# use matrix below just in case to preserve dimensions.

obsstats<-matrix(coalobs[1,4:6],nrow=1)
obsparam<-matrix(coalobs[1,1])

# example of entropy minimization algorithm:

tmp<-mincrit(obsstats, param, simstats,tol=.01,method="rejection",do.crit=TRUE)

tmp$critvals
```

`nn.ent` *Works out entropy of a sample.*

Description

The function computes the k nearest neighbour sample entropy.

Usage

```
nn.ent(th, k=4)
```

Arguments

`th`	The sample from which to compute the entropy.
`k`	The order (number of neighbours) of the sample entropy calculation.

Details

The sample entropy gives a measure of information in a (posterior) sample, or lack of it.

Value

The k nearest neighbour entropy from the sample.

Warning

For high-dimensional posterior samples, the `nn.ent` calculation is quite computationally intensive.

Author(s)

Matt Nunes

References

Singh, H. et al. (2003) Nearest neighbor estimates of entropy. *Am. J. Math. Man. Sci.*,**23**, 301–321.

Shannon, C. E. and Weaver, W. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.*, **27**, 379–423.

See Also

mincrit

Examples

```
# create a dummy sample to calculate an entropy measure:

theta<-rnorm(10000)

nn.ent(theta)
```

---

pls.abc *ABC inference using PLS components computed from data.*

Description

The function performs ABC inference using PLS transformed summary statistics.

Usage

```
pls.abc(obs, param, sumstats, obspar=NULL, abcmethod=abc,transfile = "Routput_test",
bc=FALSE, err.only=TRUE, errfn=rsse,...)
```

Arguments

obs	(matrix of) observed summary statistics.
param	matrix of simulated model parameter values.
sumstats	matrix of simulated summary statistics.
obspar	optional observed parameters (for use to assess simulation performance).
abcmethod	a function to perform ABC inference, e.g. the abc function from package *abc*.
transfile	path to file containing the PLS transformation (see Details section and documentation for *ABCtoolbox*).

bc	a boolean value indicating whether the Box-Cox transformation should be applied to the statistics prior to transformation.
err.only	a boolean value indicating whether only the simulation error should be returned. Note: if err.only=TRUE, obspar must be supplied.
errfn	an error function to assess ABC inference performance.
...	any other optional arguments to the ABC inference procedure (e.g. arguments to the abc function).

Details

The function uses a precomputed PLS transformation file to construct new summary statistics, being linear combinations of the original summary statistics. This is achieved using the transformer command line script supplied with the *ABCtoolbox* software package. Note that the precomputed transformation file can be created for example, with the *find_pls.r* R script supplied in *ABCtoolbox*. An example Routput_test for three components is supplied as a dataset. See the package documentation for more information on these two script files.

Value

A list with the following components:

post.sample	an array of dimension nacc x npar x ndatasets giving the posterior sample for each observed dataset. Not returned if err.only=TRUE.
err	simulation error (if obspar is supplied).

Note

This function requires that the transformer script from ABCtoolbox is in the system path.

Author(s)

Matt Nunes

References

Blum, M. G. B, Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci. (to appear)*.

Wegmann, D. et al. (2010) ABCtoolbox: A versatile toolkit for approximate Bayesian computations. *BMC Bioinformatics* **11**, 116–123.

Wegmann, D. et al. (2009) Efficient approximate Bayesian computation coupled With Markov chain Monte Carlo Without Likelihood. *Genetics* **182** (4), 1207–1218.

Examples

```
# example transformation file:

data(Routput_test)
write.table(Routput_test,file="Routput_test",quote=FALSE,row.names=FALSE,col.names=FALSE)

data(coal)

## Not run:
pls.abc(coal[1:3,3:7], coal[,1:2], coal[,3:7], transfile = "Routput_test",tol=.1,
method="rejection")

## End(Not run)
```

`rsse` *Simulation error measures.*

Description

Computes error measures between two sets of data.

Usage

```
rsse(a, b, v = 1)
sse(a, b, v = 1)
```

Arguments

a	Dataset 1, of dimension (n x p)
b	Dataset 2: of dimension (1 x p) or a vector.
v	an optional factor to normalise the data before computation of the RSSE.

Value

The RSSE between dataset a and b.

Author(s)

Matt Nunes

Examples

```
a<-matrix(rnorm(1000),ncol=2)
b<-runif(2)

rsse(a,b)
```

saABC *Summary statistic construction by semi-automatic ABC*

Description

saABC fits parameter estimators based on simulated data to be used as summary statistics within ABC. Fitting is by linear regression. Some simple diagnostics are provided for assistance.

Usage

```
saABC(theta, X, plot = TRUE)
```

Arguments

theta	A n x d matrix or data frame of simulated parameter values. theta[i,j] is the ith simulated value of parameter j.
X	A n x p matrix or data frame of simulated data and/or associated transformations. X[i,] is a vector of the data for parameter values theta[i,]. A constant term should not be included.
plot	When plot==TRUE, a plot of parameter values against fitted values is produced for each parameter as a side-effect.

Details

The semi-automatic ABC method of Fearnhead and Prangle (2012) is as follows:

1) Simulate parameter vectors θ_i and corresponding data sets x_i for i=1,2,...,N.

2) Use the simulations to fit an estimator of each parameter as a linear combination of f(x), where f(x) is a vector of transformations of x (including a constant term).

3) Run ABC using these simulations.

The saABC function automates step 2 of this process. The user must supply simulated parameter values theta and corresponding f(x) values x (n.b. excluding the constant term). The function returns weights for the linear combinations which can easily be used for step 3. In particular, fitted weights are returned as a matrix of weights for the columns of x and a vector of constants. The vector can usually be discarded, as it is not needed to find differences between summary statistics.

The function also returns BIC values for each parameter so that the user can judge the quality of the fits, and compare different choices of f(x). Diagnostic plots of supplied parameter values against fitted values are also optionally provided. These are useful for exploratory purposes when there are a small number of parameters, but provide less protection from overfitting than BIC values.

Value

B0	Vector of constant terms from fitted regressions.
B	Matrix of weights from fitted regressions.
BICs	Vector of BIC values for each fitted regression.

Author(s)

Dennis Prangle

References

Blum, M. G. B., Nunes M. A., Prangle D. and Sisson S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science (to appear)*.

Fearnhead, P. and Prangle, D. (2012) Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. B* **74**, Part 3, 1–28.

Examples

```
set.seed(1)
theta <- matrix(runif(2E3),ncol=2)
colnames(theta) <- c("Mean", "Variance")

X <- replicate(5, rnorm(1E3, theta[,1], theta[,2]))

saABC(theta, X)$BICs
saABC(theta, cbind(X, X^2))$BICs ##Variance parameter estimated better
```

---

selectsumm *Generic function for selecting summary statistics in ABC inference.*

Description

The function implements functions which implement summary statistics selection methods.

Usage

```
selectsumm(obs, param, sumstats, obspar = NULL, ssmethod = mincrit, verbose = TRUE,
final.dens = FALSE, ...)
```

Arguments

obs	(matrix of) observed summary statistics.
param	matrix of simulated model parameter values.
sumstats	matrix of simulated summary statistics.
obspar	optional observed parameters (for use to assess simulation performance).
ssmethod	a function to perform summary statistics selection. Current methods are `AS.select` and `mincrit`.
verbose	a boolean value indicating whether informative statements should be printed to screen.
final.dens	a boolean value indicating whether the posterior sample should be returned.
...	any other optional arguments to the summary selection procedure.

Details

The function is essentially a wrapper for more specific summary selection methods, and is designed to be flexible for future additions and minimization criteria. See the help files for each summary selection method for more details.

Value

A list with the following components:

`best`	the best subset(s) of statistics.
`critvals`	the calculated criterion values (if `do.crit=TRUE`).
`err`	simulation error (if `obspar` is supplied and `do.err=TRUE`).
`order`	the subsets considered during the algorithm (same as the input `do.only`.
`post.sample`	an array of dimension `nacc x npar x ndatasets` giving the posterior sample for each observed dataset. Not returned if `final.dens=FALSE`.
`sumsubs`	an index into the subsets considered during the algorithm.

Author(s)

Matt Nunes

References

Blum, M. G. B, Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci. (to appear)*.

Fearnhead, P. and Prangle, D. (2012) Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. B* **74**, Part 3, 1–28.

Joyce, P. and P. Marjoram (2008) Approximately sufficient statistics and Bayesian computation. *Stat. Appl. Gen. Mol. Biol.* **7** Article 26.

Nunes, M. A. and Balding, D. J. (2010) On Optimal Selection of Summary Statistics for Approximate Bayesian Computation. *Stat. Appl. Gen. Mol. Biol.* **9**, Iss. 1, Art. 34.

Wegmann, D. et al. (2009) Efficient approximate Bayesian computation coupled With Markov chain Monte Carlo Without Likelihood. *Genetics* **182** (4), 1207–1218.

Wegmann, D. et al. (2010) ABCtoolbox: A versatile toolkit for approximate Bayesian computations. *BMC Bioinf.* **11**, 116–123.

See Also

`mincrit`, `AS.select`, `pls.abc`

Examples

```
# load example data:

data(coal)
data(coalobs)

param<-coal[,2]
simstats<-coal[,4:6]

# use matrix below just in case to preserve dimensions.

obsstats<-matrix(coalobs[1,4:6],nrow=1)

tmp<-selectsumm(obsstats, param, simstats,ssmethod=AS.select,tol=.1,method="rejection",
allow.none=FALSE,inturn=TRUE,hcorr=TRUE)

tmp$best
```

`semiauto.abc` *Performs semi-automatic ABC based on summary statistics regression.*

Description

Performs semi-automatic ABC based on summary statistics regression.

Usage

```
semiauto.abc(obs, param, sumstats, obspar = NULL, abcmethod = abc, saprop = 0.5,
abcprop = 0.5, overlap = FALSE, satr = list(), plot = FALSE, verbose = TRUE,
do.err = FALSE, final.dens = FALSE, errfn = rsse, ...)
```

Arguments

`obs`	(matrix of) observed summary statistics.
`param`	matrix of simulated model parameter values.
`sumstats`	matrix of simulated summary statistics.
`obspar`	optional observed parameters (for use to assess simulation performance).
`abcmethod`	a function to perform ABC inference, e.g. the `abc` function from package *abc*.
`saprop`	a proportion, denoting the proportion of simulated datasets with which to perform semi-automatic ABC regression.
`abcprop`	a proportion, denoting the proportion of simulated datasets with which to perform ABC using `abcmethod`.
`overlap`	a boolean value indicating whether the simulated datasets specified by `saprop` and `abcprop` are disjoint (`overlap=FALSE`) or not.

satr	a list of functions indicating transformations of the summary statistics sumstats. These must be *suitable* functions, and must each return a vector, matrix or array with the number of elements being a multiple of the rows of sumstats. See details and examples sections for more information
plot	When plot==TRUE, a plot of parameter values against fitted values is produced for each parameter as a side-effect. This is most useful when the number of parameters is reasonably small.
verbose	a boolean value indicating whether informative statements should be printed to screen.
do.err	a boolean value indicating whether the simulation error should be returned. Note: if do.err=TRUE, obspar must be supplied.
final.dens	a boolean value indicating whether the posterior sample should be returned.
errfn	an error function to assess ABC inference performance.
...	any other optional arguments to the ABC inference procedure (e.g. arguments to the abc function).

Details

This function is essentially a wrapper for saABC. See the details section of saABC for more details on the implementation. The argument satr can be almost anything sensible in function form, see Examples section for example specifications.

Value

A list with the following components:

err	simulation error (if obspar is supplied and do.err=TRUE).
post.sample	an array of dimension nacc x npar x ndatasets giving the posterior sample for each observed dataset. Not returned if final.dens=FALSE.
sainfo	A list with the following information about the semi-automatic ABC run: saprop, abcprop ,overlap, satr. See arguments for more details.

Warning

The argument satr must be supplied with valid functions. Whilst there are checks, these are minimal, since doing sophisticated checks is quite difficult.

Author(s)

Matt Nunes and Dennis Prangle

References

Blum, M. G. B, Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci. (to appear)*.

Fearnhead, P. and Prangle, D. (2012) Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. B* **74**, Part 3, 1–28.

See Also

saABC, selectsumm

Examples

```
data(coal)
data(coalobs)

param<-coal[,2]
simstats<-coal[,4:6]

# use matrix below just in case to preserve dimensions.

obsstats<-matrix(coalobs[1,4:6],nrow=1)
obsparam<-matrix(coalobs[1,1])

# perform semi-automatic ABC with summary statistics defined by X, X^2,X^3,X^4:
# other alternative specifications for this could be:
# list(function(x){ cbind(x,x^2,x^3,x^4) })
# list(as.function(alist(x=,cbind(x,x^2,x^3)))) etc

tmp<-semiauto.abc(obsstats, param, simstats,tol=.01,method="rejection",
satr=list(function(x){outer(x,Y=1:4,"^")}))

tmp$sa.info

# both these functions may be problematic:

## Not run:
 tmp<-semiauto.abc(obsstats, param, simstats,tol=.01,method="rejection",satr=list(unique,sum))

## End(Not run)
```

---

stage2 *stage2*

Description

Summary statistics selection for ABC inference using estimated posterior error.

Usage

```
stage2(obs, param, sumstats, obspar = NULL, chosen, dsets = 100,
sumsubs = 1:ncol(sumstats), limit = length(sumsubs), do.only = NULL, do.err = FALSE,
final.dens = FALSE, verbose = TRUE, ...)
```

Arguments

obs	observed summary statistics.
param	matrix of simulated model parameter values.
sumstats	matrix of simulated summary statistics.
obspar	optional observed parameters (for use to assess simulation performance).
chosen	an initial estimate of the best summary statistics subset. Can be either an index into the summaries combination table (see combmat) or a vector of indices into 1:nstats. See details.
dsets	the number of simulated datasets to treat as observed when estimating the posterior error. See details.
sumsubs	an optional index into the summary statistics to limit summary selection to a specific subset of summaries.
limit	an optional integer indicating whether to limit summary selection to subsets of a maximum size.
do.only	an optional index into the summary statistics combination table. Can be used to limit entropy calculations to certain summary statistics subsets only.
do.err	a boolean value indicating whether the simulation error should be returned. Note: if do.err=TRUE, obspar must be supplied.
final.dens	a boolean value indicating whether the posterior sample should be returned.
verbose	a boolean value indicating whether informative statements should be printed to screen.
...	any other optional arguments to the ABC inference procedure (e.g. arguments to the abc function).

Details

The function uses the chosen set of summaries to determine the dsets simulated datasets which are closest (in Euclidean norm) to the observed dataset. Since the model parameters generating the summary statistics are known for these simulated datasets, for each candidate subset of summary statistics, we can compute the error under ABC inference for each of these datasets. The best subset of summary statistics is that which minimizes this (average) error over all dsets datasets.

Value

A list with the following components:

best	the best subset of statistics.
closest	the indices of the dsets simulated datasets closest to the oberved dataset as measured by the chosen subset of summaries.
err	simulation error (if obspar is supplied and do.err=TRUE).
order	the subsets considered during the algorithm (same as the input do.only.
post.sample	an array of dimension nacc x npar x ndatasets giving the posterior sample for each observed dataset. Not returned if final.dens=FALSE.
sumsubs	an index into the subsets considered during the algorithm.

Warning

This function is computationally intensive due to its cyclic ABC inference procedure.

Author(s)

Matt Nunes

References

Blum, M. G. B, Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) A comparative review of dimension reduction methods in approximate Bayesian computation. *Stat. Sci.* **28**, Issue 2, 189–208.

Nunes, M. A. and Balding, D. J. (2010) On Optimal Selection of Summary Statistics for Approximate Bayesian Computation. *Stat. Appl. Gen. Mol. Biol.* **9**, Iss. 1, Art. 34.

Examples

```
# load example data:

data(coal)
data(coalobs)

param<-coal[,2]
simstats<-coal[,5:8]

# use matrix below just in case to preserve dimensions.

obsstats<-matrix(coalobs[1,5:8],nrow=1)
obsparam<-matrix(coalobs[1,1])

## Not run:
tmp<-stage2(obsstats, param, simstats, chosen=c(1,3), dsets = 10)
tmp$best

## End(Not run)
```

Index