Package 'rhierbaps'

November 18, 2022

Type Package

Title Clustering Genetic Sequence Data Using the HierBAPS Algorithm

Version 1.1.4

Description Implements the hierarchical Bayesian analysis of populations structure (hierBAPS) algorithm of Cheng et al. (2013) <doi:10.1093/molbev/mst028> for clustering DNA sequences from multiple sequence alignments in FASTA format.
The implementation includes improved defaults and plotting capabilities and unlike the original 'MATLAB' version removes singleton SNPs by default.

License MIT + file LICENSE

Encoding UTF-8

Imports ape, purrr, utils, ggplot2, matrixStats, patchwork, methods

RoxygenNote 7.2.1

Suggests knitr, rmarkdown, ggtree, phytools, testthat, formatR

VignetteBuilder knitr

URL `https://github.com/gtonkinhill/rhierbaps`

BugReports `https://github.com/gtonkinhill/rhierbaps/issues`

NeedsCompilation no

Author Gerry Tonkin-Hill [cre, aut]

Maintainer Gerry Tonkin-Hill <`g.tonkinhill@gmail.com`>

Repository CRAN

Date/Publication 2022-11-18 14:50:07 UTC

R topics documented:

`calc_change_in_ml` *calc_change_in_ml*

Description

Calculate the change in the log marginal likelihood after moving index to each possible cluster

Usage

```
calc_change_in_ml(snp.object, partition, indexes)
```

Arguments

`snp.object`	A snp.object containing the processed SNP data.
`partition`	An integer vector indicating a partition of the isolates.
`indexes`	Indexes of the isolates to be moved (must come from one cluster.)

Value

the best cluster to move indexes to.

`calc_log_ml` *calc_log_ml*

Description

Calculate the log marginal likelihood assuming a Multinomial-Dirichlet distribution

Usage

```
calc_log_ml(snp.object, partition)
```

Arguments

`snp.object`	A snp.object containing the processed SNP data.
`partition`	An integer vector indicating a partition of the isolates.

Value

The log marginal likelihood of the given partition.

`hierBAPS` *hierBAPS*

Description

Runs the hierBAPS algorithm of Cheng et al. 2013

Usage

```
hierBAPS(
  snp.matrix,
  max.depth = 2,
  n.pops = floor(nrow(snp.matrix)/5),
  quiet = FALSE,
  n.extra.rounds = 0,
  assignment.probs = FALSE,
  n.cores = 1
)
```

Arguments

`snp.matrix`	Character matrix of aligned sequences produced by load_fasta.
`max.depth`	Maximum depth of hierarchical search (default = 2).
`n.pops`	Maximum number of populations in the data (default = number of isolates/5)
`quiet`	Whether to suppress progress information (default=FALSE).
`n.extra.rounds`	The number of additional rounds to perform after the default hierBAPS settings (default=0). If set to Inf it will run until a local optimum is reached (this might take a long time).
`assignment.probs`	whether or not to calculate the assignment probabilities to each cluster (default=FALSE)
`n.cores`	The number of cores to use.

Value

A list containing a dataframe indicating an assignment of each sequence to hierarchical clusters as well as the log marginal likelihoods for each level.

Author(s)

Gerry Tonkin-Hill

References

Cheng, Lu, Thomas R. Connor, Jukka Sirén, David M. Aanensen, and Jukka Corander. 2013. “Hierarchical and Spatially Explicit Clustering of DNA Sequences with BAPS Software.” Molecular Biology and Evolution 30 (5): 1224–28.

Examples

```
snp.matrix <- load_fasta(system.file("extdata", "small_seqs.fa", package = "rhierbaps"))
hb <- hierBAPS(snp.matrix, max.depth=2, n.pops=20, quiet=FALSE)


snp.matrix <- load_fasta(system.file("extdata", "seqs.fa", package = "rhierbaps"))
system.time({hb <- hierBAPS(snp.matrix, max.depth=2, n.pops=20, quiet=FALSE)})
```

`join_units_2` *join_units_2*

Description

Peform an iteration of the second move in the algorithm. That is combine two clusters to improve the marginal likelihood.

Usage

```
join_units_2(
  snp.object,
  partition,
  threshold = 1e-05,
  n.cores = 1,
  comb.chache = NULL
)
```

Arguments

`snp.object`	A snp.object containing the processed SNP data.
`partition`	An integer vector indicating an initial partition of the isolates.
`threshold`	The increase in marginal log likelihood required to accept a move.
`n.cores`	The number of cores to use.
`comb.chache`	a matrix recording previous marginal llks of combining clusters

Value

The best partition after combining two clusters as well as a boolean value indicating whether a move increased the marginal likelihood.

`load_fasta` *load_fasta*

Description

Loads a fasta file into matrix format ready for running the hierBAPS algorithm.

Usage

```
load_fasta(msa, keep.singletons = FALSE)
```

Arguments

`msa`	Either the location of a fasta file or ape DNAbin object containing the multiple sequence alignment data to be clustered
`keep.singletons`	A logical indicating whether to consider singleton mutations in calculating the clusters

Value

A character matrix with filtered SNP data

Examples

```
msa <- system.file("extdata", "seqs.fa", package = "rhierbaps")
snp.matrix <- load_fasta(msa)
```

`log_stirling2` *log_stirling2*

Description

log_stirling2

Usage

```
log_stirling2(n, k)
```

Arguments

`n`	number of objects
`k`	number of partitions

Value

log of the Stirling number of the second kind

model_search_parallel *model_search_parallel*

Description

Clusters DNA alignment using independent loci model

Usage

```
model_search_parallel(
  snp.object,
  partition,
  round.types,
  quiet = FALSE,
  n.extra.rounds = 0,
  n.cores = 1
)
```

Arguments

snp.object	A snp.object containing the processed SNP data.
partition	An integer vector indicating an initial starting partition.
round.types	A vector indicating which series of moves to make.
quiet	Whether to suppress progress information (default=FALSE).
n.extra.rounds	The number of additional rounds to perform after the default hierBAPS settings (default=0). If set to Inf it will run until a local optimum is reached (this might take a long time).
n.cores	The number of cores to use.

Value

an optimised partition and marginal llk

move_units_1 *move_units_1*

Description

Peform an iteration of the first move in the algorithm. That is move units from one cluster to another to improve the marginal likelihood

Usage

```
move_units_1(
  snp.object,
  partition,
  threshold = 1e-05,
  frac.clust.searched = 0.3,
  min.clust.size = 20,
  n.cores = 1
)
```

Arguments

`snp.object`	A snp.object containing the processed SNP data.
`partition`	An integer vector indicating an initial partition of the isolates.
`threshold`	The increase in marginal log likelihood required to accept a move.
`frac.clust.searched`	The percentage of a large cluster that will be moved.
`min.clust.size`	All isolates in clusters less than or equal to min.clus.size will be searched.
`n.cores`	The number of cores to use.

Value

The best partition after moving units from one cluster to another as well as a boolean value indicating whether a move increased the marginal likelihood.

`plot_sub_cluster` *plot_sub_cluster*

Description

Creates a zoom plot using ggtree focusing on a cluster.

Usage

```
plot_sub_cluster(hb.object, tree, level, sub.cluster)
```

Arguments

`hb.object`	The resulting object from running hierBAPS
`tree`	A phylo tree object to plot
`level`	The level of the subcluster to be considered.
`sub.cluster`	An integer representing the subcluster to be considered.

Examples

```
snp.matrix <- load_fasta(system.file("extdata", "seqs.fa", package = "rhierbaps"))
newick.file.name <- system.file("extdata", "seqs.fa.treefile", package = "rhierbaps")
tree <- phytools::read.newick(newick.file.name)
hb.result <- hierBAPS(snp.matrix, max.depth=2, n.pops=20)
plot_sub_cluster(hb.result, tree, level = 1, sub.cluster = 9)
```

`preproc_alignment` *preproc_alignment*

Description

Preprocessed the snp matrix for hierBAPS.

Usage

```
preproc_alignment(snp.matrix)
```

Arguments

`snp.matrix`	A matrix containing SNP data. Rows indicate isolates and columns loci.

Value

an snp.object

`reallocate_units_4` *reallocate_units_4*

Description

Peform an iteration of the fourth move in the algorithm. That is split cluster into n subclusters and re-allocate one sub-cluster.

Usage

```
reallocate_units_4(
  snp.object,
  partition,
  threshold = 1e-05,
  min.clust.size = 20,
  split = FALSE,
  n.cores = 1
)
```

Arguments

`snp.object` A snp.object containing the processed SNP data.

`partition` An integer vector indicating an initial partition of the isolates.

`threshold` The increase in marginal log likelihood required to accept a move.

`min.clust.size` Clusters smaller than min.clust.size will not be split.

`split` Whether to split only into two clusters (for move type 3).

`n.cores` The number of cores to use.

Value

The best partition after splitting a cluster and re-allocating as well as a boolean value indicating whether a move increased the marginal likelihood.

`save_lml_logs` *save_lml_logs*

Description

Saves the log marginal likelihoods to a text file.

Usage

```
save_lml_logs(hb.object, file)
```

Arguments

`hb.object` The resulting object from runnign hierBAPS

`file` The file you would like to save the log output to.

Examples

```
snp.matrix <- load_fasta(system.file(”extdata”, ”small_seqs.fa”, package = ”rhierbaps”))
hb.result <- hierBAPS(snp.matrix, max.depth=2, n.pops=20)
save_lml_logs(hb.result,  file.path(tempdir(), ”output_file.txt”))
```

split_clusters_3 *split_clusters_3*

Description

Peform an iteration of the third move in the algorithm. That is split cluster in two and re-allocate one sub-cluster.

Usage

```
split_clusters_3(
  snp.object,
  partition,
  threshold = 1e-05,
  min.clust.size = 20,
  n.cores = 1
)
```

Arguments

snp.object	A snp.object containing the processed SNP data.
partition	An integer vector indicating an initial partition of the isolates.
threshold	The increase in marginal log likelihood required to accept a move.
min.clust.size	Clusters smaller than min.clust.size will not be split.
n.cores	The number of cores to use.

Value

The best partition after splitting a cluster and re-allocating as well as a boolean value indicating whether a move increased the marginal likelihood.

Index