Package 'Rtropical'

November 9, 2021

Title Data Analysis Tools over Space of Phylogenetic Trees Using Tropical Geometry

Version 1.2.1

Maintainer Houjie Wang <wanghoujie6688@gmail.com>

Description Process phylogenetic trees with tropical support vector machine and principal component analysis defined with tropical geometry. Details about tropical support vector machine are available in : Tang, X., Wang, H. & Yoshida, R. (2020) <arXiv:2003.00677>. Details about tropical principle component analysis are available in : Page, R., Yoshida, R. & Zhang L. (2020) <doi:10.1093/bioinformatics/btaa564> and Yoshida, R., Zhang, L. & Zhan 018-0493-4>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports ape, lpSolve, lpSolveAPI, parallel, Rfast, RcppAlgos, caret

Depends R (>= 3.5.0)

Suggests rmarkdown, knitr, e1071, testthat (>= 3.0.0)

URL https://github.com/HoujieWang/Rtropical

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Houjie Wang [aut, cre],
Kaizhang Wang [aut],
Grady Weyenberg [aut],
Xiaoxian Tang [aut],
Ruriko Yoshida [aut]

Repository CRAN

Date/Publication 2021-11-09 18:50:08 UTC

R topics documented:

`all_trees` *Simulated Tree Data with Different Proximity Parameter Value*

Description

Simulated Tree Data with Different Proximity Parameter Value

Usage

```
data(all_trees)
```

Format

A list of length 12 with each element a sublist containing an ape `multiPhylo` object with 300 rooted trees on 5 tips and the tree categories.

apicomplexa *Apicomplexa gene trees sample data set.*

Description

Apicomplexa gene trees sample data set.

Usage

```
apicomplexa
```

Format

An ape `multiPhylo` object with 268 rooted trees on 8 tips.

Source

Chih-Horng Kuo, John P. Wares, Jessica C. Kissinger. *The Apicomplexan Whole-Genome Phylogeny: An Analysis of Incongruence among Gene Trees Molecular Biology and Evolution, Volume 25, Issue 12, December 2008, Pages 2689–2698.*

as.matrix.multiPhylo *Vectorize a Set of Phylognetic Trees*

Description

Unifies tip labels of all phylogenetic trees in `multiPhylo` object the same as the first tree and returns the cophenetic distance of their corresponding chronogram.

Usage

```
## S3 method for class 'multiPhylo'
as.matrix(x, tipOrder = x[[1]]$tip.label, parallel = FALSE, ncores = 2, ...)
```

Arguments

`x`	an object of class `multiPhylo`
`tipOrder`	a numeric vector of order of leaf names to which all trees in the `multiPhylo` object will unified. If not specified on purpose, the tip order of the first tree will be used.
`parallel`	a logical value indicating if parallel computing should be used. (default: FALSE)
`ncores`	a numeric value indicating the number of threads utilized for multi-cored CPUs. (default: 2)
`...`	Not used. Other arguments to as.vector

Value

A data matrix with each row a vector representation of a chronogram. Each element of the vector is the distance between two leaves.

Examples

```
data(apicomplexa)
data <- as.matrix(apicomplexa[1: 10]) # matrixize first ten trees
```

`as.vector.phylo` *Vectorize a Phylogenetic Tree*

Description

Computes the cophenetic distance and outputs them in a vector of a phylogenetic tree in `phylo` object

Usage

```
## S3 method for class 'phylo'
as.vector(x, mode = "any")
```

Arguments

`x`	A object of class phylo
`mode`	The same as `base::as.vector`. But only numeric output in vector form is accepted for other functions in `Rtropical`

Value

A vector with its elements the distance between two leaves of the tree.

Examples

```
library(ape)
tree <- rcoal(5)
tree_vec <- as.vector(tree)
```

`coef.cv.tropsvm` *Extract Optimal Tropical Hyperplane from a cv.tropsvm object*

Description

Obtain the optimal tropical hyperplane in the form of vectors from a cv.tropsvm object.

Usage

```
## S3 method for class 'cv.tropsvm'
coef(object, ...)
```

Arguments

`object`	a fitted `"cv.tropsvm"` object.
`...`	Not used. Other arguments.

Value

An output of the apex of the fitted optimal tropical hyperplane.

`coef.tropsvm` *Extract Optimal Tropical Hyperplane from a tropsvm object*

Description

Obtain the optimal tropical hyperplane in the form of vectors from a tropsvm object.

Usage

```
## S3 method for class 'tropsvm'
coef(object, ...)
```

Arguments

`object`	a fitted `"tropsvm"` object.
`...`	Not used. Other arguments.

Value

An output of the apex of the fitted optimal tropical hyperplane.

`cv.tropsvm` *Cross-Validation for Tropical Support Vector Machines*

Description

Conduct k-fold cross validation for tropsvm and return an object `"cv.tropsvm"`.

Usage

```
cv.tropsvm(x, y, parallel = FALSE, nfold = 10, nassignment = 10, ncores = 2)
```

Arguments

`x`	a data matrix, of dimension nobs x nvars; each row is an observation vector.
`y`	a response vector with one label for each row/component of x.
`parallel`	a logical value indicating if parallel computing should be used. (default: FALSE)
`nfold`	a numeric value of the number of data folds for cross-validation. (default: 10)
`nassignment`	a numeric value indicating the size of the parameter grid of assignments. (default: 10)
`ncores`	a numeric value indicating the number of threads utilized for multi-cored CPUs. (default: 2)

Value

object with S3 class `cv.tropsvm` containing the fitted model, including:

`apex`	The negative apex of the fitted optimal tropical hyperplane.
`assignment`	The best `assignment` tuned by cross-validation.
`index`	The best classification method tuned by cross-validation.
`levels`	The name of each category, consistent with categories in `y`.
`accuracy`	The validation accuracy for each fold.
`nfold`	The number of folds used in cross-validation.

See Also

`summary`, `predict`, `coef` and the `tropsvm` function.

Examples

```
# data generation
library(Rfast)
set.seed(101)
e <- 20
n <- 10
```

```
N <- 10
s <- 5
x <- rbind(
  rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(n, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
y <- as.factor(c(rep(1, n), rep(2, n)))
newx <- rbind(
  rmvnorm(N, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(N, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
newy <- as.factor(rep(c(1, 2), each = N))

# train the tropical svm
cv_tropsvm_fit <- cv.tropsvm(x, y, parallel = FALSE)

summary(cv_tropsvm_fit)
coef(cv_tropsvm_fit)

# test with new data
pred <- predict(cv_tropsvm_fit, newx)

# check with accuracy
table(pred, newy)

# compute testing accuracy
sum(pred == newy) / length(newy)
```

`lungfish` *Coelacanths genome and transcriptome data*

Description

Coelacanths genome and transcriptome data

Usage

```
lungfish
```

Format

An ape `multiPhylo` object with 1193 rooted trees on 10 tips.

Source

Tom M. W. Nye, Xiaoxian Tang, Grady Weyenberg and Ruriko Yoshida. *Principal component analysis and the locus of the Fréchet mean in the space of phylogenetic trees, Biometrika, Volume 104, Issue 4, December 2017, Pages 901–922.*

plot.troppca *Plot the Tropical Principal Components with Data Projections*

Description

Visualize the second order tropical principle components in troppca as a tropical triangle with projections on a two-dimensional plot via tropical isometry.

Usage

```
## S3 method for class 'troppca'
plot(x, plab = NULL, fw = FALSE, ...)
```

Arguments

x	a fitted troppca object.
plab	a vector of labels of all points in the given data matrix. Not needed for unlabeled data. (default: NULL)
fw	a logical variable to determine if to add Fermat-Weber point of the data projection. (default: FALSE)
...	Not used. Other arguments to plot

Value

plot.troppca does not return anything other than the plot.

predict.cv.tropsvm *Predict Method for Tropical Support Vector Machines based on Cross-Validation*

Description

Predicts values based upon a model trained by cv.tropsvm.

Usage

```
## S3 method for class 'cv.tropsvm'
predict(object, newx, ...)
```

Arguments

object	a fitted "cv.tropsvm" object.
newx	a data matrix, of dimension nobs x nvars used as testing data.
...	Not used. Other arguments to predict.

Value

A vector of predicted values of a vector of labels.

See Also

summary, coef and the cv.tropsvm function.

Examples

```
# data generation
library(Rfast)
e <- 20
n <- 10
N <- 10
s <- 5
x <- rbind(
  rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(n, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
y <- as.factor(c(rep(1, n), rep(2, n)))
newx <- rbind(
  rmvnorm(N, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(N, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
newy <- as.factor(rep(c(1, 2), each = N))

# train the tropical svm
cv_tropsvm_fit <- cv.tropsvm(x, y, parallel = FALSE)

# test with new data
pred <- predict(cv_tropsvm_fit, newx)

# check with accuracy
table(pred, newy)

# compute testing accuracy
sum(pred == newy) / length(newy)
```

---

predict.tropsvm *Predict Method for Tropical Support Vector Machines*

Description

Predicts values based upon a model trained by tropsvm.

Usage

```
## S3 method for class 'tropsvm'
predict(object, newx, ...)
```

Arguments

object	a fitted tropsvm object.
newx	a data matrix, of dimension nobs x nvars used as testing data.
...	Not used. Other arguments to predict.

Value

A vector of predicted values of a vector of labels.

See Also

summary, coef and the tropsvm function.

Examples

```
# data generation
library(Rfast)
e <- 100
n <- 10
N <- 10
s <- 5
x <- rbind(
  rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(n, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
y <- as.factor(c(rep(1, n), rep(2, n)))
newx <- rbind(
  rmvnorm(N, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(N, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
newy <- as.factor(rep(c(1, 2), each = N))

# train the tropical svm
tropsvm_fit <- tropsvm(x, y, auto.assignment = TRUE, ind = 1)

# test with new data
pred <- predict(tropsvm_fit, newx)

# check with accuracy
table(pred, newy)

# compute testing accuracy
sum(pred == newy) / length(newy)
```

read.nexus.to.data.matrix

Read NEXUS-formatted trees from two categories into a data matrix

Description

Read NEXUS-formatted trees from two categories into a data matrix

Usage

```
read.nexus.to.data.matrix(data.file1, data.file2)
```

Arguments

data.file1	A data set with trees from one category.
data.file2	A data set with trees from the other category.

Value

A data matrix with the first x rows corresponding the x trees in the first file and the last y rows are the trees from the second file.

read.tree.to.data.matrix

Read Newick-formatted trees in two categories into a data matrix

Description

Read Newick-formatted trees in two categories into a data matrix

Usage

```
read.tree.to.data.matrix(data.file1, data.file2)
```

Arguments

data.file1	A file containing trees in Newick form in a category.
data.file2	A file containing trees in Newick form in an assumed different category.

Value

`read.tree.to.data.matrix` has the same return as `read.nexus.to.data.matrix`.

sim_trees *Simulated Tree Data*

Description

Simulated Tree Data

Usage

```
data(sim_trees)
```

Format

An ape `multiPhylo` object with 300 rooted trees on 5 tips.

summary.cv.tropsvm *Summarize an Analysis of Cross-Validated Tropical Support Vector Machine*

Description

Return a summary with a more detailed explanation of the object `"cv.tropsvm"`.

Usage

```
## S3 method for class 'cv.tropsvm'
summary(object, ...)
```

Arguments

`object`	a fitted `"cv.tropsvm"` object.
`...`	Not used. Other arguments to summary.

Value

A summary of the crucial information of a tropical support vector machine is printed, including the selected best assignment and classification methods and the validation accuracy of each data fold. The summary section of classification methods specifies the sectors and their intersections used to classify points of two different categories.

See Also

`predict`, `coef` and the `cv.tropsvm` function.

Examples

```
# data generation
library(Rfast)
e <- 20
n <- 10
N <- 10
s <- 5
x <- rbind(
  rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(n, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
y <- as.factor(c(rep(1, n), rep(2, n)))
newx <- rbind(
  rmvnorm(N, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(N, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
newy <- as.factor(rep(c(1, 2), each = N))

# train the tropical svm with cross-validation
cv_tropsvm_fit <- cv.tropsvm(x, y, parallel = FALSE)

summary(cv_tropsvm_fit)
```

---

tropdet *Tropical Determinant of a Matrix*

Description

Compute the tropical determinant for a given matrix. This is equivalent to solving an assignment problem.

Usage

```
tropdet(x)
```

Arguments

x	a square matrix

Value

The determinant of the given matrix,

Examples

```
R <- matrix(sample(1:9, 9), nrow = 3)
tropdet(R)
```

tropFW *Tropical Fermat-Weber Point*

Description

Compute the tropical Fermat-Weber (FW) point for a given data matrix. The FW point minimizes the summed tropical distance to the trees described in the data matrix.

Usage

```
tropFW(x)
```

Arguments

x	a data matrix, of dimension nobs x nvars; each row is an observation vector.

Value

A list containing:

fw	The fermat-weber point.
distsum	The sum of distance from each observation to the fermat-weber point.

Examples

```
x <- matrix(rnorm(100), ncol = 10)
tropFW(x)
```

troppca.linsp *Tropical Principal Component Analysis by Tropical Linear Space*

Description

Approximate the principal component as a tropical linear space for a given data matrix and returns the results as an object of class troppca.

Usage

```
troppca.linsp(x, pcs = 2, iteration = list(), ncores = 2)
```

Arguments

x	a data matrix, of size n x e, with each row an observation vector. e is the dimension of the tropical space
pcs	a numeric value indicating the order of principal component. (default: 2)
iteration	a list with arguments controlling the iteration of the algorithm.
	exhaust a logical variable indicating if to iterate over all possible combinations of the linear space based on the given data matrix x. If FALSE, please input a number of iteration for niter. If TRUE, please enter 0 for niter and this function will iterate over all possible combinations of linear space. This could be time consuming when x is large. (default: FALSE)
	niter a numeric variable indicating the number of iterations. (default: 100)
ncores	a numeric value indicating the number of threads utilized for multi-cored CPUs. (default: 2)

Value

A list of S3 class "troppca", including:

pc	The principal component as a tropical linear space
obj	The tropical PCA objective, the sum of tropical distance from each point to the projection.
projection	The projections of all data points.
type	The geometry of principal component.

Examples

```
library(Rfast)
n <- 100
e <- 10
sig2 <- 1
x <- rbind(rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(sig2, e)))
troppca_fit <- troppca.linsp(x)
```

---

troppca.linsp2poly *Tropical Principal Component Analysis by Polytope Converted from Linear Space*

Description

Approximate the principal component as a tropical polytope converted from tropical linear space for a given data matrix via MCMC and return the results as an object of class troppca.

Usage

```
troppca.linsp2poly(x, pcs = 2, nsample = 1000, ncores = 2)
```

Arguments

x	a data matrix, of size n x e, with each row an observation vector. e is the dimension of the tropical space#'
pcs	a numeric value indicating the order of principal component. (default: 2)
nsample	a numeric value indicating the number of samples of MCMC. (default: 1000)
ncores	a numeric value indicating the number of threads utilized for multi-cored CPUs. (default: 2)

Value

A list of S3 class "troppca", including:

pc	The principal component as a tropical linear space
obj	The tropical PCA objective, the sum of tropical distance from each point to the projection.
projection	The projections of all data points.
type	The geometry of principal component.

Examples

```
library(Rfast)
n <- 50
e <- 50
s <- 5
x <- rbind(
  rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(n, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
troppca_fit <- troppca.linsp2poly(x)
```

---

troppca.poly *Tropical Principal Component Analysis by Tropical Polytope*

Description

Approximates the principal component as a tropical polytope for a given data matrix via MCMC and return the results as an object of class troppca.

Usage

```
troppca.poly(x, pcs = 2, nsample = 1000, ncores = 2)
```

Arguments

x	a data matrix, of size n x e, with each row an observation vector. e is the dimension of the tropical space#'
pcs	a numeric value indicating the order of principal component. (default: 2)
nsample	a numeric value indicating the number of samples of MCMC. (default: 1000)
ncores	a numeric value indicating the number of threads utilized for multi-cored CPUs. (default: 2)

Value

A list of S3 class "troppca", including:

pc	The principal component as a tropical linear space
obj	The tropical PCA objective, the sum of tropical distance from each point to the projection.
projection	The projections of all data points.
type	The geometry of principal component.

Examples

```
library(Rfast)
n <- 50
e <- 50
s <- 5
x <- rbind(
  rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(n, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
troppca_fit <- troppca.poly(x)
plot(troppca_fit)
```

---

tropproj.linsp *Projection on Tropical Linear Space*

Description

Compute projection of data points on a given tropical linear space.

Usage

```
tropproj.linsp(x, V)
```

Arguments

x	a data matrix, of size n x e, with each row an observation.
V	a data matrix, of dimension s x e, with each row a basis of tropical linear space. e is the dimension of the tropical space and s is the dimension of the linear space.

Value

A matrix of projections of all data points.

Examples

```
library(Rfast)
n <- 100
e <- 10
sig2 <- 1
s <- 3
x <- rbind(rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(sig2, e)))
V <- matrix(runif(s * e, -10, 10), nrow = s, ncol = e)
x_proj <- tropproj.linsp(x, V)
head(x_proj)
```

---

tropproj.poly *Projection on Tropical Polytope*

Description

Project a point onto a given tropical polytope.

Usage

```
tropproj.poly(x, tconv)
```

Arguments

x	a data vector, of length e.
tconv	a data matrix, of size e x s, with each column a vertex of the tropical polytope. e is the dimension of the tropical space and s is the number of vertices of the polytope

Value

A projected vector on the given tropical polytope.

Examples

```
# Generate a tropical polytope consisting of three trees each with 5 leaves
library(ape)
pltp <- sapply(1:3, function(i) {
  as.vector(rcoal(5))
})
# Generate an observation and vectorize it
tree <- rcoal(5)
tree_vec <- as.vector(tree)
tropproj.poly(tree_vec, pltp)
```

`tropsvm` *Tropical Support Vector Machines*

Description

Fit a discriminative two-class classifier via linear programming defined by the tropical hyperplane which maximizes the minimum tropical distance from data points to itself in order to separate the data points into sectors (half-spaces) in the tropical projective torus.

Usage

```
tropsvm(x, y, auto.assignment = FALSE, assignment = NULL, ind = 1)
```

Arguments

`x`	a data matrix, of dimension nobs x nvars; each row is an observation vector.
`y`	a response vector with one label for each row/component of x.
`auto.assignment`	a logical value indicating if to provide an `assignment` manually. If `FALSE`, an input is required, otherwise the function automatically finds a good assignment.(default: FALSE)
`assignment`	a numeric vector of length 4 indicating the sectors of tropical hyperplane that the data will be assigned to. The first and third elements in the `assignment` are the coordinates of an observed point in data matrix `x` believed from the first category where the maximum and second maximum of the vector addition between the fitted optimal tropical hyperplane and the point itself are achieved. The meanings for the second and the fourth element in the `assignment` are the same but for the points in the second category. Namely, the first and second values in the `assignment` are the indices of sectors where the two point clouds are assigned. Not needed when `auto.assignment = TRUE`. (default: NULL)
`ind`	a numeric value or a numeric vector ranging from 1 to 70 indicating which classification method to be used. There are 70 different classification methods. Details of a given method can be retrieved by `summary`. The different classification methods are proposed to resolve the issue when points fall on the intersections of sectors. Users can have personal choices if better knowledge is assumed. (default: 1)

Value

An object with S3 class `tropsvm` containing the fitted model, including:

`apex`	The negative apex of the fitted optimal tropical hyperplane.
`assignment`	The user-input or auto-found `assignment`.
`index`	The user-input classification method.
`levels`	The name of each category, consistent with categories in `y`.

See Also

`predict`, `coef` and the `cv.tropsvm` function.

Examples

```
# data generation
library(Rfast)
e <- 100
n <- 10
N <- 100
s <- 10
x <- rbind(
  rmvnorm(n, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(n, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
y <- as.factor(c(rep(1, n), rep(2, n)))
newx <- rbind(
  rmvnorm(N, mu = c(5, -5, rep(0, e - 2)), sigma = diag(s, e)),
  rmvnorm(N, mu = c(-5, 5, rep(0, e - 2)), sigma = diag(s, e))
)
newy <- as.factor(rep(c(1, 2), each = N))

# train the tropical svm
tropsvm_fit <- tropsvm(x, y, auto.assignment = TRUE, ind = 1)

coef(tropsvm_fit)

# test with new data
pred <- predict(tropsvm_fit, newx)

# check with accuracy
table(pred, newy)

# compute testing accuracy
sum(pred == newy) / length(newy)
```

Index