Package: TSDFGS (via r-universe)

September 11, 2024

Type Package

Title Training Set Determination For Genomic Selection

Version 2.4.2

Encoding UTF-8

Date 2023-04-09

URL `https://oumarkme.github.io/TSDFGS`

Depends R (>= 4.0)

Maintainer Jen-Hsiang Ou `<jen-hsiang.ou@imbim.uu.se>`

Description We propose an optimality criterion to determine the required training set, r-score, which is derived directly from Pearson's correlation between the genomic estimated breeding values and phenotypic values of the test set <doi:10.1007/s00122-019-03387-0>. This package provides two main functions to determine a good training set and its size.

License GPL (>= 3)

Imports dplyr, ggplot2, latex2exp, parallel, Rcpp, RcppEigen

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.2.3

LazyData true

Repository https://oumarkme.r-universe.dev

RemoteUrl https://github.com/oumarkme/tsdfgs

RemoteRef HEAD

RemoteSha a802b8120154f3517b68b69cc0e5a65b1ae92119

Contents

cd_score *CD-score*

Description

This function calculate CD-score <doi:10.1186/1297-9686-28-4-359> by given training set and test set.

Usage

```
cd_score(X, X0)
```

Arguments

`X` A numeric matrix. The training set genotypic information matrix can be given as genotype matrix (coded as -1, 0, 1) or principle component matrix (row: sample; column: marker).

`X0` A numeric mareix. The test set genotypic information matrix can be given as genotype matrix (coded as -1, 0, 1) or principle component matrix (row: sample; column: marker).

Value

A floating-point number, CD score.

Author(s)

Jen-Hsiang Ou

Examples

```
data(geno)
## Not run: cd_score(geno[1:50, ], geno[51:100])
```

FGCM *Fit logistic growth curve model*

Description

A function for fitting logisti growth model

Usage

```
FGCM(geno, nt = NULL, n_iter = NULL, multi.threads = FALSE)
```

Arguments

`geno`	Genotype information saved as a dataframe. Columns represent variants (SNPs or PCs).
`nt`	A numerical vector of training set sample size for estimating logistic growth curve parameters
`n_iter`	Number of simulation of each training set size. Automatically gave a suitable number by default.
`multi.threads`	Default: FALSE. Multi-thread function is only avalyble for mac or linux systems.

Value

Estimation of parameters.

Examples

```
data(geno)
## Not run: FGCM(geno)
```

geno *Genotype information*

Description

A PCA matrix of rice genotype information. This data was published by Zhao et al. (2011) <doi:10.1038/ncomms1467>

Usage

```
geno
```

Format

A numeric matrix (PCA) with 404 rows (sample) and 404 columns (PCs).

Source

http://www.ricediversity.org/data/

Examples

```
data(geno)
```

nt2r *Simulate r-scores of each training set size*

Description

Calculate r-scores (un-target) by in parallel.

Usage

```
nt2r(geno, nt, n_iter = 30, multi.threads = FALSE)
```

Arguments

`geno`	A numeric dataframe of genotype, column represent sites (genotype coding as 1, 0, -1)
`nt`	Numeric. Number of training set size
`n_iter`	Times of iteration. (default = 30)
`multi.threads`	Default: FALSE. Multi-thread function is only avalyble for mac or linux systems.

Value

A vector of r-scores of each iteration

Examples

```
data(geno)
## Not run: nt2r(geno, 50)
```

optTrain *Optimal training set determination*

Description

This function is designed for determining optimal training set.

Usage

```
optTrain(
  geno,
  cand,
  n.train,
  subpop = NULL,
  test = NULL,
  method = "rScore",
  min.iter = NULL,
  console = TRUE
)
```

Arguments

`geno`	A numeric matrix of principal components (rows: individuals; columns: PCs).
`cand`	An integer vector of which rows of individuals are candidates of the training set in the geno matrix.
`n.train`	The size of the target training set. This could be determined with the help of the ssdfgp function provided in this package.
`subpop`	A character vector of sub-population's group name. The algorithm will ignore the population structure if it remains NULL.
`test`	An integer vector of which rows of individuals are in the test set in the geno matrix. The algorithm will use an un-target method if it remains NULL.
`method`	Choices are rScore, PEV and CD. rScore will be used by default.
`min.iter`	Minimum iteration of all methods can be appointed. One should always check if the algorithm is converged or not. A minimum iteration will set by considering the candidate and test set size if it remains NULL.
`console`	Default: TRUE. Set it to FALSE if you don't want the function printing out the number count of each iteration.

Value

This function will return 3 information including OPTtrain (a vector of chosen optimal training set), TOPscore (highest scores of before iteration), and ITERscore (criteria scores of each iteration).

Author(s)

Jen-Hsiang Ou

Examples

```
data(geno)
## Not run: optTrain(geno, cand = 1:404, n.train = 100)
```

pev_score *PEV score*

Description

This function calculate prediction error variance (PEV) score <doi:10.1186/s12711-015-0116-6> by given training set and test set.

Usage

```
pev_score(X, X0)
```

Arguments

`X` A numeric matrix. The training set genotypic information matrix can be given as genotype matrix (coded as -1, 0, 1) or principle component matrix (row: sample; column: marker).

`X0` A numeric mareix. The test set genotypic information matrix can be given as genotype matrix (coded as -1, 0, 1) or principle component matrix (row: sample; column: marker).

Value

A floating-point number, PEV score.

Author(s)

Jen-Hsiang Ou

Examples

```
data(geno)
## Not run: pev_score(geno[1:50, ], geno[51:100])
```

r_score *r-score*

Description

This function calculate r-score <doi:10.1007/s00122-019-03387-0> by given training set and test set.

Usage

```
r_score(X, X0)
```

Arguments

`X` A numeric matrix. The training set genotypic information matrix can be given as genotype matrix (coded as -1, 0, 1) or principle component matrix (row: sample; column: marker).

`X0` A numeric mareix. The test set genotypic information matrix can be given as genotype matrix (coded as -1, 0, 1) or principle component matrix (row: sample; column: marker).

Value

A floating-point number, r-score.

Author(s)

Jen-Hsiang Ou

Examples

```
data(geno)
## Not run: r_score(geno[1:50, ], geno[51:100])
```

SSDFGS *Sample size determination for genomic selection*

Description

This function is designed to generate an operating curve for sample size determination

Usage

```
SSDFGS(geno, nt = NULL, n_iter = NULL, multi.threads = FALSE)
```

Arguments

geno	A numeric data frame carried genotype information (column: PCs, row: sample)
nt	A numeric vector carried training set sizes for r-score simulation.
n_iter	Number of iterations for estimating parameters.
multi.threads	Default: False. If TRUE, this function will use 75% of threads if the computer has more than 4 threads. Multi-thread computing is only avalible in mac and linux environments.

Value

An operating curve and its information.

Author(s)

Jen-Hsiang Ou & Po-Ya Wu

Examples

```
data(geno)
## Not run: SSDFGS(geno)
```

---

subpop *Sub-population information*

Description

Sub-population information of samples. This data was published by Zhao et al. (2011) <doi:10.1038/ncomms1467>

Usage

```
subpop
```

Format

A character vector.

Source

http://www.ricediversity.org/data/

Examples

```
data(subpop)
```

TSDFGS *TSDFGS*

Description

We propose an optimality criterion to determine the required training set, r-score, which is derived directly from Pearson's correlation between the genomic estimated breeding values and phenotypic values of the test set <doi:10.1007/s00122-019-03387-0>. This package provides two main functions to determine a good training set and its size.

Index