Package 'metaVaR'

February 24, 2020

Type Package

Title Population Genomics Analysis from Metagenomic Data

Version 0.3.18

Author Amin Madoui

Maintainer Amin Madoui <amine.madoui@gmail.com>

Description Various functions to reconstruct several species models simultaneously and to perform population genetic anayses from enviromental molecular data. This includes the estimation of genetic differentiation and the identification of loci under selection.

License GPL-3

Encoding UTF-8

LazyData true

Imports stats,MASS,methods,survival,fitdistrplus,dbscan,igraph,ggplot2,reshape2

Depends R (>= 2.10)

RoxygenNote 6.1.1

Collate 'LK.R'
'data.R'
'dbscan2mvc.R'
'fst.R'
'getMWIS.R'
'mvc_class.R'
'mvc.R'
'mvc2AdjMat.R'
'mvc2mvs.R'
'mvs_class.R'
'mvs.R'
'mwis.R'
'plotMvs.R'
'pwFst.R'
'readMvc.R'
'readMvcList.R'
'readMvs.R'
'readMvsList.R'
'setMvc.R'
'setMvs.R'
'tryParam.R'
'varFilterbyCov.R'

    'writeMvc.R'
    'writeMvcList.R'
    'writeMvs.R'
    'writeMvsList.R'

Suggests knitr,
    rmarkdown

VignetteBuilder knitr

R topics documented:

AdjMat2Graph	*AdjMat2Graph*

Description

Create an undirected graph from an adjacency matrix of mvc

Usage

```
AdjMat2Graph(adjMatrix)
```

Arguments

adjMatrix   an adjacency matrix.

Value

a graph object of class `igraph`.

---

dbscan2mvc	*dbscan2mvc*

Description

Convert a dbscan ouput to metavariant clusters

Usage

```
dbscan2mvc(dbscanRes, cov, mvcMinVar = 1000, mvcMinCov = 5)
```

Arguments

dbscanRes   dbscan result.

cov   metavaraint loci depth of coverage dataframe.

mvcMinVar   minimum number of metavariants for mvc, set to 1,000 by default.

mvcMinCov   minimum vamue for the median depth of coverage

Value

a list of object of class `mvc`

---

fst	*fst*

Description

Compute F-statistics from MVS allele frequencies

Usage

```
fst(p)
```

Arguments

p   allele frequencies of class `data.frame`

Details

The F-statistics is computed as follow, $F = \frac{var(p)}{mean(p)(1-mean(p))}$, where `p` is a vector of allele frequencies

Value

F-statistics value

References

Sewall Wright. Genetical Structure of Populations. Nature, 166:247–249, 1950

---

getMWIS   *getMWIS*

Description

getMWIS identifies all the maximum weighted independant sets from a list of `mvc`.

Usage

```
getMWIS(MVC, p = FALSE, prefix = "out")
```

Arguments

MVC	a list of objects of class `mvc`.
p	`logical`, for graphical output, set to TRUE by default.
prefix	name of the graphical output directory, set to "out" by default.

Value

a list of object of class `mvc` candidate for `mvs`.

References

S. Sakai, M. Togasaki, K. Yamazaki. A note on greedy algorithms for the maximum weighted independent set problem. *Discrete applied Mathematics* , 126:313-322, 2003.

Examples

```
## Not run: data("MS5")
e = c(5,6)
p = c(5, 10)
MVC = tryParam(e, p , MS5$cov)
MWIS = getMWIS (MVC)
## End(Not run)
```

graph2mvcComp   *graph2mvcComp*

Description

Extract connected mvs from the mvc graph and attribute the mvc to components

Usage

```
graph2mvcComp(graph, mvcList)
```

Arguments

graph   an adjacency matrix.

mvcList   a list of object of class `mvc`.

Value

a list of object of class `mvc` assigned to graph component.

LK   *LK*

Description

Compute the LK statistics from MVS allele frequencies

Usage

```
LK(p)
```

Arguments

p   allele frequencies of class `data.frame`

Details

The LK is computed as follow, $LK = \frac{(n-1)Fst}{mean(F_{ST})}$, where `F_{ST}` is the wright's F-statistics.

Value

LK value of class `data.frame`

References

"RC Lewontin and J Krakauer. Distribution of gene frequency as a test of the theory of the selective neutrality of polymorphisms. Genetics 74(1):175–195. 1973

---

metaVaR *metavariants data*

Description

Data from discoSNP++ ran on five samples of metagenomic data genrated by the Tara Oceans expedition in the Mediterranean Sea

Usage

```
data(MS5)
```

Format

a list of object of class `data.frame`

pop population names

cov biallelic loci depth of coverage

freq metavariant frequencies

Source

Genoscope and Tara Oceans consortium

---

mvc *mvc*

Description

mvc

Usage

```
mvc(name = "", eps = 0, pts = 0, var = c(), pop = c(),
  cov = data.frame(), fit = list(), comp = 0, deg = 0,
  weight = 0, score = 0, mwis = FALSE)
```

Arguments

name	cluster name.
eps	epsilon value used by dbscan.
pts	minimum points value used by dbscan.
var	metavariants IDs.
pop	population names.
cov	depth of coverage matrix of biallelic loci.
fit	`fitdist` object for fitting between observed and theoritical depth of coverage.
comp	component id of the mvc.
deg	mvc degree in the mvc graph

weight	mvc weight
score	mvc score for maximum weighted independant set
mwis	logical, set to true if the mvc is a maximum weighted independant set

---

mvc-class   *Class mvc*

Description

Class `mvc` defines a metavariants cluster.

Slots

name cluster name.

eps epsilon value used by dbscan.

pts minimum points value used by dbscan.

var metavariants IDs.

pop population names.

cov depth of coverage matrix of biallelic loci.

fit `fitdist` objects for fitting between observed and theoritical depth of coverage in each population.

comp component id of the mvc.

deg mvc degree in mvc graph.

weight mvc weight.

score mvc score for maximum weighted independant set

mwis logical, if true the mvc is a maximum weighted independant set

Author(s)

momosapiens

---

mvc2AdjMat   *mvc2AdjMat*

Description

Create an adjacency matrix mvc based on shared metavariants between `mvc`

Usage

```
mvc2AdjMat(MVC)
```

Arguments

| MVC | a list of objects of class `mvc`. |

Value

a mvc adjacency matrix.

mvc2graph   *mvc2graph*

Description

Create an undirected graph of a list of `mvc`

Usage

```
mvc2graph(mvcList)
```

Arguments

mvcList   a list of object of class `mvc`.

Value

a mvc graph of class `igraph`.

mvc2mvs   *mvc2mvs*

Description

mvc2mvs selects the `mvc` being maximum weighted independent set and passing quality filters to generates the `mvs`.

Usage

```
mvc2mvs(MVC, minPop = 3, minCov = 8, freq, minVarCov = 8, sd = 2,
  minVar = 100)
```

Arguments

MVC   a list of object of class `mvc`.

minPop   minimum number of population with `minCov`, set to 3 by default.

minCov   minimum median depth of coverage, set to 8 by default.

freq   allele frequencies in a `data.frame`

minVarCov   minimum depth of coverage of loci, set to 8 by default.

sd   standard deviation of depth of coverage to select variant, set to 2 by default.

minVar   minimum number of variants in a `mvs`, set to 100 by default.

Value

a list of objects of class `mvs`.

Examples

```
## Not run:
e = c(5,6)
p = c(5, 10)
MVC = tryParam(e, p , MS5$cov)
MWIS = getMWIS (MVC)
MVS = mvc2mvs (MWIS, freq = MS5$freq, minPop = 3, minCov = 6 , minVarCov = 8, sd = 2, minVar = 100)
## End(Not run)
```

---

mvs   *mvs*

Description

mvs

Usage

```
mvs(id = "", freq = data.frame(), gFst = 0, gLK = 0,
  pwFst = data.frame(), name = "", eps = 0, pts = 0, var = c(),
  pop = c(), cov = data.frame(), fit = list(), comp = 0, deg = 0,
  weight = 0, score = 0, mwis = FALSE)
```

Arguments

id	mvs id
freq	allele frequency
gFst	global Fst
gLK	global LK
pwFst	pairwise-Fst
name	cluster name.
eps	epsilon value used by dbscan.
pts	minimum points value used by dbscan.
var	metavariants IDs.
pop	population names.
cov	depth of coverage matrix of biallelic loci.
fit	fitdist object for fitting between observed and theoritical depth of coverage.
comp	component id of the mvc.
deg	mvc degree in the mvc graph
weight	mvc weight
score	mvc score for maximum weighted independant set
mwis	logical, set to true if the mvc is a maximum weighted independant set

---

`mvs-class` *Class mvs*

Description

Class `mvs` defines a metavariants species.

Slots

id mvs id.

freq allele frequencies

gFst global Fst

gLK global LK

pwFst pairwise-fst

Author(s)

momosapiens

`mwis` *mwis*

Description

Calculate the mvc weight and score in a list of object of class `mvc`. Identify some the maximum weighted independant sets and change the slot mwis

Usage

```
mwis(nbComp, MVC)
```

Arguments

nbComp	number of connected components
MVC	a list of object of class `mvc`.

Value

a list of object of class `mvc`.

---

plotMvs   *plotMvs*

Description

plotMvs allows various plotting for `mvs` like depth of coverage and allele frequency distribution, pairwise F-statistics, LK distribution

Usage

```
plotMvs(mvs, type)

## S4 method for signature 'mvs'
plotMvs(mvs, type)
```

Arguments

mvs   an object of class `mvs`.

type   type of plot to produce

Details

type can be "freq" for allele frequencies distribution, "cov", for depth of coverage distribution. "heatFst", for a heatmap of the pairwise F-statistics "LK", for the global LK statistics

---

pwFst   *pwFst*

Description

Calculate pairwise F-statistics between each population

Usage

```
pwFst(p)
```

Arguments

p   allele frequencies (data.frame) with row and column names

Value

a symetric pairwise Fst table of class `data.frame`

References

Sewal Wright. Genetical Structure of Populations. Nature, 166:247–249, 1950

---

readMvc *readMvc*

Description

Read a `mvc` from a directory dreated by `writeMvc`

Usage

```
readMvc(mvcDir)
```

Arguments

mvcDir   output directory

Value

an object of class `mvc`

---

readMvcList *readMvcList*

Description

read `mvc` from a list of directory

Usage

```
readMvcList(dir = "")
```

Arguments

dir   a directory.

Value

a list of `mvc`

---

readMvs   *readMvs*

Description

Read a `mvs` from a directory dreated by `writeMvs`

Usage

```
readMvs(mvsDir)
```

Arguments

mvsDir   output directory

Value

an object of class `mvs`

---

readMvsList   *readMvsList*

Description

read `mvs` from a list of directory

Usage

```
readMvsList(dir = "")
```

Arguments

dir   a directory.

Value

a list of `mvs`

setMvc	*setMvc*

Description

set `mvc` attributes

Usage

```
setMvc(x, name = x@name, eps = x@eps, pts = x@pts, var = x@var,
  pop = x@pop, cov = x@cov, fit = x@fit, comp = x@comp,
  deg = x@deg, weight = x@weight, score = x@score, mwis = x@mwis)

## S4 method for signature 'mvc'
setMvc(x, name = x@name, eps = x@eps, pts = x@pts,
  var = x@var, pop = x@pop, cov = x@cov, fit = x@fit,
  comp = x@comp, deg = x@deg, weight = x@weight, score = x@score,
  mwis = x@mwis)
```

Arguments

x	an object of class `mvc`.
name	cluster name.
eps	epsilon value used by dbscan.
pts	minimum points value used by dbscan.
var	metavariants IDs.
pop	population names.
cov	depth of coverage matrix of biallelic loci.
fit	`fitdist` object for fitting between observed and theoritical depth of coverage.
comp	component id of the mvc.
deg	mvc degree in the mvc graph
weight	mvc weight
score	mvc score for maximum weighted independant set
mwis	logical, if true the mvc is a maximum weighted independant set

Value

an object of class `mvc`.

---

setMvs   *setMvs*

Description

set mvs attributes

Usage

```
setMvs(x, id = x@id, freq = x@freq, gFst = x@gFst, gLK = x@gLK,
  pwFst = x@pwFst, name = x@name, eps = x@eps, pts = x@pts,
  var = x@var, pop = x@pop, cov = x@cov, fit = x@fit,
  comp = x@comp, deg = x@deg, weight = x@weight, score = x@score,
  mwis = x@mwis)

## S4 method for signature 'mvs'
setMvs(x, id = x@id, freq = x@freq, gFst = x@gFst,
  gLK = x@gLK, pwFst = x@pwFst, name = x@name, eps = x@eps,
  pts = x@pts, var = x@var, pop = x@pop, cov = x@cov,
  fit = x@fit, comp = x@comp, deg = x@deg, weight = x@weight,
  score = x@score, mwis = x@mwis)
```

Arguments

x	an object of class mvc.
id	mvs name.
freq	allele frequencies
gFst	global Fst
gLK	global LK
pwFst	pairwise-Fst
name	cluster name.
eps	epsilon value used by dbscan.
pts	minimum points value used by dbscan.
var	metavariants IDs.
pop	population names.
cov	depth of coverage matrix of biallelic loci.
fit	fitdist object for fitting between observed and theoritical depth of coverage.
comp	component id of the mvc.
deg	mvc degree in the mvc graph
weight	mvc weight
score	mvc score for maximum weighted independant set
mwis	logical, if true the mvc is a maximum weighted independant set
mvs	an object of class mvs.

Value

an object of class mvs.

---

tryParam   *tryParam*

Description

tryParam tests several couples of epsilon and minimum points values for density-based clustering of metavariants.

Usage

```
tryParam(eps, pts, cov, mvcMinVar = 1000)
```

Arguments

eps	epsilon value(s).
pts	minimum points value(s).
cov	depth of coverage of biallelic loci in data.frame.
mvcMinVar	minimum of metavariants in mcv, set to 1,000 by default.

Value

a list of objects of class mcv.

Examples

```
## Not run: data("MS5")
e = c(5,6)
p = c(5, 10)
MVC = tryParam(e, p , MS5$cov)
## End(Not run)
```

---

varFilterbyCov   *varFilterbyCov*

Description

varFilterbyCov

Usage

```
varFilterbyCov(cov, dev = 2, minCov = 8)
```

Arguments

cov	depth of coverage
dev	number of standard deviation
minCov	minimum depth of coverage

---

writeMvc *writeMvc*

Description

Write a `mvc` in an output directory

Usage

```
writeMvc(x, prefix = "out")

## S4 method for signature 'mvc'
writeMvc(x, prefix = "out")
```

Arguments

x   an object of class `mvc`.

prefix   output directory

---

writeMvcList *writeMvcList*

Description

Write a list of `mvc` in an output directory

Usage

```
writeMvcList(MVC, prefix = "out")
```

Arguments

MVC   a list of objects of class `mvc`.

prefix   output directory

---

writeMvs   *writeMvs*

Description

Write a mvs in an output directory

Usage

```
writeMvs(x, prefix = "out")

## S4 method for signature 'mvs'
writeMvs(x, prefix = "out")
```

Arguments

x   an object of class mvs.

prefix   output directory

---

writeMvsList   *writeMvsList*

Description

Write a list of mvs in an output directory

Usage

```
writeMvsList(MVS, prefix = "out")
```

Arguments

MVS   a list of objects of class mvs.

prefix   output directory

Index