Package ‘pergola’

April 3, 2016

Type Package
Title Toolbox for Polyploid Genetic Data
Version 1.0
Date 2016-03-31
Description Provides tools for linkage mapping in polyploids.
    It implements the method PERGOLA, which is a fast, deterministic method to
    calculate the order of markers in a linkage group.
License GPL-3
Imports seriation, utils, stats
LazyData true
Suggests knitr, gclus, dendextend, dendextendRcpp, Matrix, rmarkdown,
grDevices
VignetteBuilder knitr
RoxygenNote 5.0.1
URL http://github.com/grafab/pergola
BugReports http://github.com/grafab/pergola/issues
NeedsCompilation no
Author Fabian Grandke [aut, cre],
    Dirk Metzler [ths]
Maintainer Fabian Grandke <grafabian@gmail.com>
Repository CRAN
Date/Publication 2016-04-03 16:29:32

R topics documented:

add_offset .................................................. 2
bases2genotypes .......................................... 3
calcRec ..................................................... 3
calcSarf ..................................................... 4
add_offset

Description
Add offset to zero distance markers to allow computation of correlation between maps.

Usage
add_offset(map, offset = 0.1)

Arguments
map
One map. Required.
offset
Numeric value for offset.

Value
Map object.

Examples
data(simTetra)
simTetrageno <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetrageno, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
map <- pullMap(rfMat, split = split)
map <- add_offset(map)
bases2genotypes  

Transform bases into genotypes

Description
Preprocess the input data in case bases are provided instead of genotypes

Usage
bases2genotypes(input, ploidy)

Arguments
- **input**: Matrix of genotype bases. Rows represent the individual markers. Columns represent samples, dependending on the ploidy.
- **ploidy**: Ploidy level of the organism. Influences how many columns are collapsed into one.

Value
Matrix of genotypes. The number of columns is 1/ploidy of the input.

Examples
```r
data(simTetra)
bases2genotypes(simTetra, 4)
```

calcRec  

Recombination frequencies computation

Description
Calculate recombination frequencies for a whole matrix

Usage
```r
calcRec(input, ploidy, sparse = FALSE, ...)
```

Arguments
- **input**: Matrix of genotypes. Rows represent markers. Columns represent samples.
- **ploidy**: Ploidy level of the organism.
- **sparse**: Logical, if the matrix is a sparse matrix or not.
- **...**: arguments are forwarded to pairwRF.
Value

Matrix of pairwise recombination frequencies.

Examples

data(simTetra)
simTetraGen <- bases2genotypes(simTetra, ploidy = 4)
calcRec(simTetraGen, 4)

calcSarf  Calculates the SARP value of given input.

Description

The sum of adjacent recombination frequency (SARP) is a measure of how well the marker order is. This function calculates it for a given matrix of pairwise recombination frequencies and marker order. The SARP criterion can be extended to a neighborhood > 1.

Usage

calcSarf(rf, ord = 1:(ncol(rf)), n = 1)

Arguments

rf          Matrix of pairwise recombination frequencies.
ord         Vector with marker order.
n           Number of neighbors, which are included in the calculation.

Value

Single numeric value, which is the result of the SARP calculation.

References


Examples

data(simTetra)
simTetraGeno <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetraGeno, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
calcSarf(rfMat, split$order, n = 1)
calcSarf(rfMat, split$order, n = 2)
calcSarf(rfMat, split$order, n = 3)
maketangle

Create a gray scale tanglegram

Description

Create tanglegram. Removes markers, that are not in both trees. Calculates alternating light and dark shades of grey.

Usage

maketangle(dend1, dend2, cutheight, k = NULL, ncol = k, ...)

Arguments

dend1 First dendrogram. Required.
dend2 Second dendrogram. Required.
cutheight The height, at which dend1 is cut. Influences number of colors.
k Number of desired linkage groups.
col Number of desired colors.
... Other parameters are forwarded to the tanglegram command.

Value

None. Plotting only.

Examples

data(simTetra)
simTetraGen <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetraGen, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
map <- pullMap(rfMat, split = split)
dend <- map2dend(map)
maketangle(dend, dend, cutheight = 500, k = 7, ncol = 7)
map2dend  

Transforming a map into a dendrogram

Description

Create dendrogram object. The map specific distance are ignored and only the grouping and ordering is maintained. Allows for comparison of whole map with package 'dendextend'.

Usage

map2dend(map, mergeoff = 0L)

Arguments

map  

One map. Required.

mergeoff  

Numeric, offset between chromosomes, to avoid equal heights in dendrogram. Equal heights lead to problems in cor_bakers_gamma().

Value

Dendogram object.

Examples

data(simTetra)
simTetrageno <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetrageno, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
map <- pullMap(rfMat, split = split)
dend <- map2dend(map)
plot(dend)

pergola  

Toolbox for polyploid genetic data

Description

This package provides multiple tools to work with polyploid data.

Details

Load the dataset simTetra and analyse it according to the vignette.
plotChr

Plotting one or two linkage maps

Description

Visualization of one or two linkage maps. Used as comparison between two different maps (e.g. different parameters or linkage mapping tools).

Usage

plotChr(map1, map2 = NULL, cex = 1, labels = c("Map 1", "Map 2"), ...)

Arguments

map1 Numeric vector with marker positions.
map2 Optional second map for comparison.
cex Font size in the figure.
labels Labels for the two blocks
... arguments are forwarded to plot.

Value

None. Plotting only.

Examples

data(simTetra)
simTetraGen <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetraGen, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
map <- pullMap(rfMat, split = split)
plotChr(map[[1]])

plotRf

Plot recombination frequencies

Description

Graphical representation of recombination frequencies to support supervised estimation of the numbers of clusters

Usage

plotRf(rf, plottype = "dendrogram", method = "single", cex.axis = 1, ...)
Arguments

- rf: Matrix of pairwise recombination frequencies.
- plottype: Default is "dendrogram". Any other value will plot the recombination frequencies.
- method: Default is "single", which is used for the hierarchical clustering.
- cex.axis: Size of axis labels in image plot.
- ... arguments are forwarded to image.

Value

None.

Examples

```r
data(simTetra)
simTetraGen <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetraGen, 4)
plotRf(rfMat)
```

pullMap

Creates map object

Description

Creates map object from matrix of pairwise recombination frequencies.

Usage

```r
pullMap(rf, split, fun = "haldane", corr = 1)
```

Arguments

- rf: Matrix of pairwise recombination frequencies.
- split: Split object.
- fun: Function to space the markers on the map. Default is "haldane". Alternatives are "kosambi", "carter" and "none.
- corr: Corrector, if recombinations are overestimated. Allows to multiply all spaces by a fixed factor.

Value

Ordered vector of marker locations. Each marker has a name attribute.
shuffleInput

Examples

data(simTetra)
simTetrageno <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetrageno, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
pullMap(rfMat, split = split)

shuffleInput

Randomize marker order and alleles within samples In simulated datasets, the order or markers and alleles within samples is often given. To remove any prior knowledge, that would not be available, the data should be randomized. Thus, the performance of our tool can be validated unbiased.

Description

Randomize marker order and alleles within samples

In simulated datasets, the order or markers and alleles within samples is often given. To remove any prior knowledge, that would not be available, the data should be randomized. Thus, the performance of our tool can be validated unbiased.

Usage

shuffleInput(input, ploidy = 4, ignore = 0)

Arguments

input Matrix of genotypes. Rows represent markers. Columns represent samples.
ploidy Ploidy level of the organism. Default is 4.
ignore In case of unnecessary frontstanding columns (e.g. parental genotypes or row-names), these can be excluded from the randomization.

Value

Matrix of the same size as the input matrix. The markers are in a random order and the alleles within the samples are in a random order.

Examples

data(simTetra)
shuffleInput(simTetra, 4)
**simHexa**  
*Hexaploid F2 population*

**Description**

100 offspring genotypes from an F2 crossing population. Generated with PedigreeSim (Voorips et al, 2012).

**Usage**

simHexa

**Format**

A data frame with 131 rows and 600 variables:

**Source**

https://github.com/PBR/pedigreeSim

---

**simTetra**  
*Tetraploid F2 population*

**Description**

100 offspring genotypes from an F2 crossing population. Generated with PedigreeSim (Voorips et al, 2012).

**Usage**

simTetra

**Format**

A data frame with 131 rows and 400 variables:

**Source**

https://github.com/PBR/pedigreeSim
sortLeafs

Chromosome wise leaf ordering

Description

Calculates the optimal leaf ordering pairwise for all linkage groups.

Usage

sortLeafs(rf, df, method = "seriation", maxSarf = NULL)

Arguments

rf Matrix of pairwise recombination frequencies.
df Vector of cluster numbers, created by splitChr(). Zeros indicated filtered markers and will be ignored.
method Name of method. Default: seriation (uses the optimal leaf ordering algorithm from the seriation package). Alternatives endlink (order.endlink from gclus) and endlink-global (ignores linkage groups).
maxSarf Maximum number of neighbor to include into SARF extension.

Value

Vector of global marker order.

Examples

data(simTetra)
simTetrageno <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetrageno, 4)
split <- splitChr(rfMat, nchr = 7)
sortLeafs(rfMat, split)

splitChr

Split markers into chromosomes

Description

This function splits markers into linkage groups (LG), which ideally represent chromosomes. The split is based on hierarchical clustering with a single linkage distance.

Usage

splitChr(rf, height = 0.4, nchr = NULL, method = "single",
filter = FALSE, thresh = 0.05, rm.dup = TRUE)
### Arguments

- `rf`  
  Matrix of pairwise recombination frequencies.

- `height`  
  Threshold value for grouping the markers.

- `nchr`  
  Expected number of chromosomes.

- `method`  
  Default is "single", which is used for the hierarchical clustering.

- `filter`  
  Logical, if the result should be filtered or not. Default is FALSE. Creates zeros for the markers below the threshold.

- `thresh`  
  Threshold for filtering. Default is 0.05, i.e. linkage groups with less than 5% of markers, are filtered out.

- `rm.dup`  
  Logical, if the duplicated markers should be filtered out. TRUE is highly recommended because the markers have no added value for the linkage map.

### Value

Vector of cluster relationship. Same length and order as the matrix of recombination frequencies.

### Examples

```r
data(simTetra)
simTetragenoc2genotypes(simTetra, 4)
rfMat<-calcRec(simTetragenoc, 4)
splitChr(rfMat, nchr = 7)
```

---

### swapChrs

**Swap chromosomes**

Find best matching chromosome for each chromosome and brings them into the same order.

### Usage

```r
swapChrs(map, comp)
```

### Arguments

- `map`  
  Map to switch.

- `comp`  
  Other map for comparison.

### Value

`map`
**switchChrs**

**Examples**

```r
data(simTetra)
simTetrageno <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetrageno, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
map <- pullMap(rfMat, split = split)
split <- sortLeafs(rfMat, split, method = "endlink")
map2 <- pullMap(rfMat, split = split)
map <- switchChrs(map, map2)
```

**Description**

Wrapper function to switch chromosomes for the whole map

**Usage**

```r
switchChrs(map, comp)
```

**Arguments**

- `map` Map to switch.
- `comp` Other map for comparison.

**Value**

`map`

**Examples**

```r
data(simTetra)
simTetrageno <- bases2genotypes(simTetra, 4)
rfMat <- calcRec(simTetrageno, 4)
split <- splitChr(rfMat, nchr = 7)
split <- sortLeafs(rfMat, split)
map <- pullMap(rfMat, split = split)
split <- sortLeafs(rfMat, split, method = "endlink")
map2 <- pullMap(rfMat, split = split)
map <- switchChrs(map, map2)
```
Index

*Topic **datasets**
  - simHexa, 10
  - simTetra, 10

add_offset, 2

bases2genotypes, 3

calcRec, 3
calcSarF, 4

maketangle, 5
map2dend, 6

pergola, 6
pergola-package (pergola), 6
plotChr, 7
plotRf, 7
pullMap, 8

shuffleInput, 9
simHexa, 10
simTetra, 10
sortLeafs, 11
splitChr, 11
swapChrs, 12
switchChrs, 13