Package ‘mQTL.NMR’

March 28, 2017

Type Package
Title Metabolomic Quantitative Trait Locus Mapping for 1H NMR data
Version 1.8.0
Date 2015-04-09
Author Lyamine Hedjazi and Jean-Baptiste Cazier
Maintainer Lyamine Hedjazi <mqtl@ican-institute.org>

Description mQTL.NMR provides a complete mQTL analysis pipeline for 1H NMR data. Distinctive features include normalisation using most-used approaches, peak alignment using RSPA approach, dimensionality reduction using SRV and binning approaches, and mQTL analysis for animal and human cohorts.

License Artistic-2.0
URL http://www.ican-institute.org/tools/

LazyLoad yes
LazyData yes
NeedsCompilation yes

biocViews Cheminformatics, Metabolomics, Genetics, SNP

Depends R (>= 2.15.0)
Imports qtl, GenABEL, MASS, outliers, graphics, stats, utils
Suggests BiocStyle

R topics documented:

mQTL.NMR-package .................................................. 2
alignSp ............................................................. 4
align_mQTL .......................................................... 5
attachSegments ..................................................... 7
circle_mQTL ......................................................... 8
configureRSPA ...................................................... 9
format_mGWA ........................................................ 10
format_mQTL ......................................................... 11
load_datafiles ....................................................... 12
load_demo_data .................................................... 12
matchSegments ...................................................... 13
normalise .......................................................... 15
mQTL.NMR-package

Description

mQTL.NMR provides a complete mQTL analysis pipeline for 1H NMR data. Distinctive features include normalisation using most-used approaches, peak alignment using RSPA approach, dimensionality reduction using SRV and binning approaches, and mQTL analysis for animal and human cohorts.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>mQTL.NMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>0.99.2</td>
</tr>
<tr>
<td>Link:</td>
<td><a href="http://www.ican-institute.org/tools">http://www.ican-institute.org/tools</a></td>
</tr>
<tr>
<td>Date:</td>
<td>2014-05-19</td>
</tr>
<tr>
<td>License:</td>
<td>Artistic-2.0</td>
</tr>
</tbody>
</table>

Main functions:

- `format_mQTL`: generates the proper format of animal crosses data
- `format_mGWA`: generates the proper format of human data
- `align_mQTL`: peak alignment
- `normalise_mQTL`: normalisation of metabolomic data using different approaches (Probabilistic quotient, constant sum,...)
- `pre_mQTL`: dimension reduction by statistical recoupling of variables or binning
• `process_mQTL`: computes LODs using extended Haley-Knott method for animal crosses
• `process_mGWA`: computes p-values using a standard linear regression approach for human
• `post_mQTL`: plots the results of a given run
• `summary_mQTL`: provides the results as a table
• `simple.plot`: Plots a region of NMR profile
• `SRV.plot`: Plots the regions identified by SRV in NMR profiles
• `ppersp`: Plot 3-D profile of LODs as function of genomic position and chemical shift
• `pplot`: Plot a color scale layer
• `Top_SRV.plot`: Plot top SRV clusters for structural assignment
• `circle_mQTL`: Plot a circular genome-metabolome plot

Author(s)
Lyamine Hedjazi and Jean-Baptiste Cazier
Maintainer: Lyamine Hedjazi <mqtl@ican-institute.org>

References

Examples

```r
# Download data files
load_datafiles()

# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleanngen)

# Constant Sum normalisation
nmeth<-'CS'
normalise_mQTL(cleandat, CSnorm, nmeth)

# Alignment
align_mQTL(CSnorm, aligdat)

# Dimensionality reduction
met="rectangle" # choose the statistical summarizing measure ("max","sum","trapez",...)
RedMet="SRV" # reduction method ("SRV" or "bin")
pre_mQTL(aligdat, reducedF, RedMet="SRV", met, corrT=0.9)

# mQTL mapping
results<- list() # a list to stock the mQTL mapping results
nperm< 0 # number of permutations if required
results<-process_mQTL(reducedF, cleanngen, nperm)

## Post-Process
```
```r
post_mQTL(results)

## Summarize
redfile<="rectangle_SRV.ppm"
summary_mQTL(results,redfile,T=8)

#plot circular genome
circle_mQTL(results, Th=8, spacing=0)

## visualisation and metabolite identification
#plot NMR profile
simple.plot(file=cleandat,lo=3.02,hi=3.08,k=1:20,title="NMR profile")

#plot SRV regions
SRV.plot(file1=cleandat,file2=rectangle_SRV,lo=3.02,hi=3.08,k=1:20,title="Cluster plot")

#plot lod for the region of interest
SRV_lod.plot(results,rectangle_SRV,Th=1)

#plot top lod SRV regions
Top_SRV.plot(file1=cleandat,file2=rectangle_SRV,results=results,met=met,intMeth="mean")
```

---

**alignSp**

*Base function for Spectrum Alignment*

**Description**

Alignment of spectrum segment to the spectrum of interest

**Usage**

```r
alignSp(refSp, refSegments, intSp, intSegments, recursion, MAX_DIST_FACTOR, MIN_RC)
```

**Arguments**

- `refSp` a vector specifying the reference spectrum
- `refSegments` a list characterizing the reference segments (start, end, peaks, ...)
- `intSp` a vector specifying the spectrum of interest
- `intSegments` a list characterizing the segment of interest (start, end, peaks, ...)
- `recursion` A list defining default values of the parameters of recursive alignment (minimal segment width, recursion step, resemblance, acceptance, ...)
- `MAX_DIST_FACTOR` distance matching parameter (0.5*peak width)
- `MIN_RC` minimum resemblance coefficient

**Value**

`alignedSpectrum` aligned spectrum as a vector
**align_mQTL**

**Peak alignment and normalisation of metabolomic data**

**Description**

Recursive Segment-Wise Peak Alignment (RSPA) for accounting peak position variation across metabolomic data

**Usage**

`align_mQTL(datafile, outdat, idx)`

**Examples**

```r
## Data
load_datafiles()
Sp<-t(read.table(phenofile))
ppm<-as.numeric(colnames(Sp))

## Normalization
normSp<-normalise(abs(Sp),'CS')

##Segmentation and matching parameters
setupRSPA(ppm)

##reference spectrum selection
attach(normSp)
index<-selectRefSp(Sp,recursion$step)
refSp<-Sp[index,]

##segmentate a reference spectrum
refSegments<- segmentateSp(refSp, peakParam)

##segmentate a test spectrum
testSegments<- segmentateSp(Sp[1,], peakParam)

##attach test and reference segments
attachedSegs<-attachSegments(refSegments,testSegments)

##Match test and reference segments
attach(attachedSegs)
Segs<-matchSegments(refSp,Sp[1,],testSegmentsNew,refSegmentsNew,MAX_DIST_FACTOR, MIN_RC)

##Align test spectrum
attach(Segs)
SpAlg<- alignSp(refSp,refSegs,Sp[1,],testSegs,recursion,MAX_DIST_FACTOR,MN_RC)
```
Arguments

datafile  The main input file of raw spectra in the csvs format
outdat   The output file of aligned spectra in the csvs format
idx      index of reference spectrum

Details

The algorithm is based on the following workflow:

1. Automatic selection of a reference spectrum (if required).
2. Segmentate a reference spectrum.
3. Then for each test spectrum:
   • segmentate a test spectrum.
   • match test and reference segments.
   • align a test spectrum.

Value

It returns a file with aligned data in the csvs format.

Author(s)

Lyamine Hedjazi

References


See Also

alignSp, attachSegments, matchSegments, segmentateSp, format_mQTL, format_mQTL

Examples

# Download data files
load_datafiles()

# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleangen)

# Constant Sum normalisation
nmeth<-'CS'
normalise_mQTL(cleandat, CSnorm, nmeth)

# Alignment
align_mQTL(CSnorm, aligndat)
### Description

Concatenation of test and reference segments to ensure one-to-one correspondence.

### Usage

```r
attachSegments(refSegments, testSegments)
```

### Arguments

- `refSegments` : a list characterizing the segments of the reference spectrum (start, end, peaks, center)
- `testSegments` : a list characterizing the segments of the test spectrum (start, end, peaks, center)

### Details

The algorithm:

1. For each reference segment within segment boundaries, i.e. between initial and final positions, find all centre (middle) positions of test segments and merge those segments, if more than one centre position is found
2. Apply the same procedure for each test segment

### Value

A list:

- `segments$start` : a vector specifying the starting of each concatenated test segment
- `segments$PeakLeftBoundary` : a list defining the peak left boundary of each concatenated test segment
- `segments$PeakRightBoundary` : a list defining the peak right boundary of each concatenated test segment
- `segments$Peaks` : a list specifying the peaks information of each concatenated test segment (max position, start position, end position,...)
- `segments$end` : a vector specifying the end of each concatenated test segment
- `segments$center` : a vector specifying the center of each concatenated test segment

### Author(s)

Lyamine Hedjazi

### References

See Also

matchSegments

Examples

```r
## Data
load_datafiles()
Sp<-t(read.table(phenofile))
ppm<-as.numeric(colnames(Sp))

## Normalization
normSp<-normalise(abs(Sp),'CS')

##Segmentation and matching parameters
setupRSPA(ppm)

##reference spectrum selection
attach(normSp)
index<-selectRefSp(Sp,recursion$step)
refSp<-Sp[index,]

##segmentate a reference spectrum
refSegments<- segmentateSp(refSp, peakParam) # segmentate reference spectrum

##segmentate a test spectrum
testSegments<- segmentateSp(Sp[,1], peakParam) # segmentate test spectrum (1st sample)

##attach test and reference segments
attachedSegs<-attachSegments(refSegments,testSegments)
```

circle_mQTL

Circular genome-metabolome plot for mQTL.NMR

Description

shows mQTL locations and relations with the metabolome on a central chemical axis

Usage

circle_mQTL(results, Th = 0, chr = 9, spacing = 25)

Arguments

results a list containing mQTL mapping results generated by mQTL.NMR package
Th a numerical parameter specifying LOD threshold
chr a numerical value defining the chromosomes to show if necessary
spacing a numerical parameter specifying the spacing between chromosomes on the circular genome
Value
A circuler plot where the central horizontal line corresponds to the NMR chemical axis, the circle represents the chromosomal positions, and the colored lines significant association between a shift and genomic location.

Author(s)
Lyamine Hedjazi

See Also
pplot

Examples
load_datafiles()
load(results)
circle_mQTL(results, Th=8, spacing=0)

configureRSPA  
segmentation and recursive alignment parameters

Description
The routine used to change and improve the RSPA algorithm performance

Usage
configureRSPA(ppm)

Arguments
ppm  
a numerical vector defining the chemical shift scale

Author(s)
Jean-Baptiste Cazier

See Also
setupRSPA

Examples
load_datafiles()
load(results)
ppm<-results$ppm
configureRSPA(ppm)
format_mGWA

Routine to reformat the data into the required format to perform mG-WAS

Description

This function enables to reformat data into the proper format. The user should provide input metabolomic file, Genotype file, map file and a file containing sex, age and individual IDs.

Usage

format_mGWA(datafile, genofile1, genofile2, covarfile, outdat, outgeno)

Arguments

datafile metabolomic data file

genofile1 genotype file in the "ped" format

genofile2 map file containing more information on SNP marker (position, ...)
covarfile a text file contains covariates such as age or sex
outdat output data file with formatted phenotype data in csvs format
outgeno output data file with formatted genotype data in csvs format

Value

formatted phenotype and genotype data files (in format csvs) are written to the user working directory (it is therefore preferable that the user create a new directory to be used throughout the study)

Author(s)

Lyamine Hedjazi

See Also

format_mQTL, process_mGWA

Examples

load_datafiles()
format_mGWA(human.pheno, human.geno, humanMap, covarFile, cleandat, cleangen)
format_mQTL

Routine to reformat the data of animal crosses into the required format to perform mQTL mapping

Description

This function enables to reformat data into the proper format. The user should provides in input metabolomic file, Genotype file and a file containing sex and pgm (parental grandmother).

Usage

format_mQTL(datafile, genofile, physdat, outdat, outgeno)

Arguments

datafile metabolomic data file in text format
ogenofile genotype data file in text format
physdat a file containing sex and pgm in text format
outdat Output data file with formatted phenotype data (metabolomic data + sex + pgm) in the format csvs
outgeno Output data file with formatted genotype data in the csvs format

Value

formatted phenotype and genotype data files (in format csvs) are written to the user working directory (it is therefore preferable that the user create a new directory to be used throughout the study)

Author(s)

Lyamine Hedjazi

See Also

align_mQTL,

Examples

# Download data files
load_datafiles()

# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleangen)
load_datafiles  
*Load data files for examples*

Description
Data files are downloaded from the extdata directory to the user's working directory.

Usage
```r
load_datafiles()
```

Value
Loaded data files concern four datasets: raw metabolomic data ('phenofile.txt'), genomic data ('genofile.txt'), additional data ('physiodat.txt').

Author(s)
Lyamine Hedjazi

See Also
- `format_mQTL`

Examples
```r
# Load data files
load_datafiles()
```

load_demo_data  
*Load demo data files*

Description
Data files are downloaded from the sourceforge.net website to the user's working directory.

Usage
```r
load_demo_data()
```

Value
Loaded data files concern four datasets: raw metabolomic data (Metabofile.txt), genomic data (Genofile.txt), additional data (physiodat.txt), formatted metabolomic data (met.clean.txt) and formatted genomic data (gen.clean.txt). Data files specifying additional information and results are also provided such as: result of SRV clustering (ur.rectangle.alig.txt), aligned data (ur.alig.txt), normalized data by CS and PQN methods (cs.norm.txt and pqn.norm.txt) and SRV clusters parameters (rectangle_SRV.txt)
matchSegments

Author(s)
Lyamine Hedjazi

See Also
format_mQTL

Examples

```r
## Not run:
# Load demo data files
load_demo_data()

## End(Not run)
```

matchSegments Matching the segment of interest to the corresponding reference

Description
The algorithm makes use of a fuzzy logic approach to match the segment of interest to the corresponding reference.

Usage
`matchSegments(refSp, intSp, intSegments, refSegments, MAX_DIST_FACTOR, MIN_RC)`

Arguments
- `refSp`: a vector specifying the spectrum of reference
- `intSp`: a vector specifying the spectrum of interest (test spectrum)
- `intSegments`: a list characterizing the segments of the spectrum of interest
- `refSegments`: a list characterizing the segments of the reference spectrum (start, end, peaks, center)
- `MAX_DIST_FACTOR`: distance matching parameter (0.5*peak_width)
- `MIN_RC`: minimum resamblance coefficient

Details
Algorithm:
1. pick-up segment of interest
2. pick-up reference segments
3. calculate relative distance between them
4. calculate relative resamblance between them
5. find min value of relative distance and resamblance
6. use it as representative of similarity between target and reference segments
7. find the segment that has the highest value of both relative distance and resamblance
## Data
load_datafiles()
Sp<-t(read.table(phenofile))
ppm<-as.numeric(colnames(Sp))

## Normalization
normSp<-normalise(abs(Sp),'CS')

##Segmentation and matching parameters
setupRSPA(ppm)

##reference spectrum selection
attach(normSp)
index<-selectRefSp(Sp,recursion$step)
refSp<-Sp[index,]

##segmentate a reference spectrum
refSegments<- segmentateSp(refSp, peakParam) # segmentate reference spectrum

##segmentate a test spectrum
testSegments<- segmentateSp(Sp[1,], peakParam) # segmentate test spectrum (1st sample)

##attach test and reference segments
attachedSegs<-attachSegments(refSegments,testSegments)

##Match test and reference segments
attach(attachedSegs)
Segs<-matchSegments(refSp,Sp[1,],testSegmentsNew,refSegmentsNew,MAX_DIST_FACTOR, MIN_RC)
Description

Removing dilutions between biofluid samples (normalisation of spectra)

Usage

`normalise(X, method, refIdx, noiseInt)`

Arguments

- `X` A matrix specifying metabolomic data
- `method` A character defining the normalization method. Constant sum normalisation (method<-'CS'), Constant noise normalisation (method<-'CN'), Quotient probabilistic method (method<-'PQN'), Linear baseline normalisation (method<-'LBN'), Auto scaling (method<-'AS'), Pareto scaling (method<-'PS').
- `refIdx` index of reference individual (set by the user if necessary)
- `noiseInt` noise region on the resonance axis as an interval (ex. [11,12] ppm)

Value

A matrix defining normalised spectrum

Author(s)

Lyamine Hedjazi

References


See Also

`normalise_mQTL`
normalise_mQTL

Normalisation of metabolomic data

Description

Takes use of the base function `normalise` to provide a normalised metabolomic data file.

Usage

```r
normalise_mQTL(infile, outfile, method, refIdx=1, noiseInt=c(11,12))
```

Arguments

- `infile` a text file with non-normalised spectra profiles
- `outfile` a text file with normalised spectra profiles
- `method` a character defining the normalization method: - Constant sum normalisation (method='CS') - Constant noise normalisation (method='CN') - Quotient probabilistic method (method='PQN') - Linear baseline normalisation (method='LBN') - Auto-scaling (method='AS') - Pareto scaling (method='PS')
- `refIdx` index of reference individual (set by the user)
- `noiseInt` noise region on the resonance axis as an interval (ex. [11,12] ppm)

Value

a file containing normalised spectra profiles

Author(s)

Lyamine Hedjazi

See Also

`normalise`
**peakPeaks**

**Examples**

```r
# Download data files
load_datafiles()

# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleangen)

# Constant Sum normlisation
nmeth<- 'CS'
normalise_mQTL(cleandat, CSnorm, nmeth)
```

---

**peakPeaks**  
**Peak picking algorithm**

**Description**

Identification of peaks in metabolomic data based on the calculation of smoothed derivates using Savitzky-Golay filter. The peak is identified if derivative crosses zero, i.e. sign(X'(i))>sing(X'(i+1)).

**Usage**

```r
peakPeaks(SpSmooth, dpDerivs, Sp)
```

**Arguments**

- `SpSmooth` a vector specifying smoothed spectrum
- `dpDerivs` a vector specifying smoothed derivative of the spectrum
- `Sp` a vector specifying the spectrum of interest

**Value**

identified peaks

**Author(s)**

Lyamine Hedjazi

**References**


**See Also**

`s golayDeriv`
Examples

```r
load_datafiles()
Sp<-t(read.table(phenofile))

## Peak picking
Spectrum<-Sp[1,]
iOrder <- 3
iFrameLen<- 11

SpDerivs<-sgolayDeriv(Spectrum,iOrder,iFrameLen,2)
SpSmooth<-sgolayDeriv(Spectrum,iOrder,iFrameLen,1)
peaks<-peakPeaks(SpSmooth,SpDerivs,Spectrum)
```

post_mQTL

Plot top LOD results

Description

plot the results of a given run

Usage

```r
post_mQTL(results, probs = c(0.95, 0.99, 0.999, 0.9999))
```

Arguments

- `results`: a list containing the results of mQTL analysis.
- `probs`: a numerical vector of probabilities with values in [0,1]. (Values up to 2e-14 outside that range are accepted and moved to the nearby endpoint).

Details

This function plots different results corresponding to top LOD marker

Value

It returns one window gathering all figures of the mQTL analysis. Each figure is also saved separately in the user’s working space.

Author(s)

Hedjazi Lyamine

See Also

- `pre_mQTL`
ppersp

Example

# Download data files
load_datafiles()

# mQTL mapping results
load(results)

# Plot mQTL mapping results
post_mQTL(results)

---

ppersp

Plot a 3-D profile of LODs

Description

Plot 3-D profile of LODs as function of genomic position and chemical shift

Usage

ppersp(z, ppm, title, theta=-15, phi=15, r=50)

Arguments

- `z`: a matrix specifying metabolome genome-wide mQTL mapping results
- `ppm`: a vector of chemical shift
- `title`: plot title
- `theta`: angle defining the viewing direction (azimuthal direction)
- `phi`: angle defining the viewing direction (colatitude direction)
- `r`: the distance of the eyepoint from the centre of the plotting box.

Value

plot 2D-profile

Author(s)

Jean-Baptiste Cazier

See Also

ppplot

Examples

# Download data files
load_datafiles()

# mQTL mapping results
load(results)
## Plot 3D profile

dev.new(width=5,height=5,pointsize=5)
ppersp(results$res, results$ppm, title="Example plot")

---

### Plot a color scale layer

**Description**  
Plot the results with a color scale y layer over 3 in 2D

**Usage**  

```r
pplot(z, title, ppm, res, LT = c(5,10,15,20))
```

**Arguments**

- `z`: a matrix specifying metabolome genome-wide mQTL mapping results
- `title`: figure title
- `ppm`: a vector of chemical shift
- `res`: mQTL results to be plotted (scanone object)
- `LT`: `quantile(res,probs)`, `res`: matrix of mQTL mapping results and `probs`: vector of probabilities

**Value**  
plot of 2-D profile

**Author(s)**  
Jean-Baptiste Cazier

**See Also**

`ppersp`

**Examples**

```r
# Download data files
load_datafiles()

# mQTL mapping results
load(results)

## Plot 3D profile

dev.new(width=5,height=5,pointsize=5)
probs=c(0.95,0.99,0.999,0.9999) ## probabilities
pplot(results$res,"Full 2D Profile", results$ppm, results$best, quantile(results$res,probs=probs))
```
**pre_mQTL**

*Statistical Recoupling of variables for mQTL analysis*

**Description**

Makes use of SRV to preprocess metabolomic data for dimensionality reduction by statistical recoupling of variables

**Usage**

```r
pre_mQTL(infile, outfile, RedMet="SRV", met="sum", corrT = 0.9, BinWidth=0.01)
```

**Arguments**

- `infile`: metabolomic datafile in csv format
- `outfile`: reduced metabolomic datafile in csv format
- `met`: a character specifying the used statistical summary
- `RedMet`: a character indicating the used dimensionality reduction method: Redmet="SRV" for statistical recoupling of variables and Redmet="bin" to apply the bining approach
- `corrT`: a numerical parameter indicating correlation threshold
- `BinWidth`: a numerical parameter indicating the bining width

**Details**

mQTL-NMR package implements two dimensionality reduction methods. The first one concerns the SRV algorithm which forms clusters of variables using a measure of a local spectral dependency. The second one concerns the classical bining method which divides the spectra into evenly spaced windows (bins) whose width commonly ranges between 0.001 and 0.05 ppm.

**Value**

variables are associated into a series of clusters (or bins). This function provides in output the parameters of the clusters (min and max borders, mean, ...)

**Author(s)**

Lyamine Hedjazi

**References**


**See Also**

`SRV`, `post_mQTL`
Examples

# Download data files
load_datafiles()

# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleangen)

# Constant Sum normalisation
nmeth<-'CS'
normalise_mQTL(cleandat, CSnorm, nmeth)

# Alignment
align_mQTL(CSnorm, aligdat)

# Dimensionality reduction
met="rectangle" # choose the statistical summarizing measure ("max", "sum", "trapez", ...)
RedMet="SRV" # reduction method ("SRV" or "bin")
pre_mQTL(aligdat, reducedF, RedMet="SRV", met, corrT=0.9)

Description

Test for association between a trait and genetic polymorphism

Usage

process_mGWA(phenofile = phenofile, genofile = genofile, nperm = 0, gtmodel = "overdominant", covarList = c("sex", "age"))

Arguments

phenofile a text file with phenotype data
genofile a text file with genotype data
nperm number of permutations
gtmodel genetic model ("additive", "recessive", "dominant", "overdominant")
covarList covariate variables ("sex" and/or "age")

Details

This function makes use of metabolomic and genotype data to perform genome-wide association analysis using a standard regression method based on the GenABEL package.

Value

2D score tables (-log10(p-value))
process_mQTL

Author(s)
Lyamine Hedjazi

References

See Also
format_mGWA

Examples

load_datafiles()
format_mGWA(human.pheno, human.geno, humanMap, covarFile,hcleandat, hcleangen)

# mQTL mapping
results<- list() # a list to stock the mQTL mapping results
nperm<- 0 # number of permutations if required
results<-process_mGWA(phenofile=hreducedF, genofile=hcleangen,nperm=0, gtmodel="additive")

---

process_mQTL  mQTL mapping

Description
Function to process the tissue extract of the individuals for QTL analysis

Usage
process_mQTL(datfile, genfile, nperm = 0)

Arguments
datfile  a text file with phenotype data
genfile  a text file with genotype data
nperm   nperm

Details
This function makes use of metabolomic and genotype data to perform QTL analysis based on the R/QTL package, for mapping quantitative trait loci. In particular, it makes use of the extended Haley-Knott method to optimize the LOD score evaluation and avoid problems with missing genotypes.

Value
2D LOD score table
segmentateSp

**Description**

Determination of highly intensive peaks in the spectrum of interest and subsequent concatenation of closely located peaks into larger segments

**Usage**

`segmentateSp(Sp, peakParam)`

**Arguments**

- **Sp**
  - a vector defining the spectrum

- **peakParam**
  - a list:
    - `ampThr`: amplitude threshold [default 2*median(peaksMaxValues)]
    - `iFrameLen`: Savitzky-Golay frame length
    - `iOrder`: polynomial order of Savitzky-Golay filter
    - `iFrameLen`: Savitzky-Golay frame length
    - `minPeakWidth`: min peak size
    - `ppmDist`: distance to concatenate adjacent peaks
selectRefSp

Value
A list:
- `testSegmentsNew` a list specifying the new test segments
- `refSegmentsNew` a list specifying the new reference segments

Author(s)
Lyamine Hedjazi

References

See Also
`attachSegments`, `matchSegments`

Examples
```r
# Data
load_datafiles()
Spc <- t(read.table(phenofile))
ppm <- as.numeric(colnames(Spc))

# Normalization
normSp <- normalise(abs(Sp), 'CS')

# Segmentation and matching parameters
setupRSPA(ppm)

# reference spectrum selection
attach(normSp)
index <- selectRefSp(Sp, recursion$step)
refSp <- Sp[index,]

# segmentate a reference spectrum
refSegments <- segmentateSp(refSp, peakParam) # segmentate reference spectrum
```

---

selectRefSp  
Automated selection of a reference spectrum

Description
The selection of reference spectrum among all spectrums is based on the highest similarity to all other spectra

Usage
`selectRefSp(X, step)`
**Arguments**

- **X**: matrix of spectra
- **step**: a numerical parameter used to scale spectral regions down to specific bin size

**Value**

returns the index of selected spectrum

**Author(s)**

Lyamine Hedjazi

**See Also**

alignSp

**Examples**

```r
# Data
Sp=matrix(rnorm(10*5000,mean=0,sd=1), nrow=10,ncol=5000)
# Reference spectrum selection
step=0.02 # Recursion step (default 0.02)
index<-selectRefSp(Sp,step)
```

### setupRSPA

**setup of alignment parameters**

**Description**

Configuration of the RSPA algorithm invariant parameters

**Usage**

```
setupRSPA(ppm)
```

**Arguments**

- **ppm**: a vector defining chemical shift scale

**Author(s)**

Jean-Baptiste Cazier

**See Also**

configureRSPA
Examples

load_datafiles()
load(results)
ppm<-results$ppm
setupRSPA(ppm)

sgolay

Find the matrix of differentiation filters

Description

designs a Savitzky-Golay (polynomial) FIR smoothing filter. The polynomial order must be less than the frame size which must be odd.

Usage

sgolay(k,F,W)

Arguments

k
a numerical value of polynomial order
F
a numerical value of frame size
W
weighting matrix

Value

matrix of differentiators

Author(s)

Lyamine Hedjazi

References

Sophocles J. Orfanidis, INTRODUCTION TO SIGNAL PROCESSING, Prentice-Hall, 1995, Chapter 8

See Also

sgolayDeriv

Examples

k <- 3
F <- 11
Sg=sgolay(k,F)
sgolayDeriv  

*Calculate smoothed derivates*

**Description**

Calculate smoothed derivates using Savitzky-Golay filter

**Usage**

```R
sgolayDeriv(dpSpectr, iOrder, iFrameLen, j)
```

**Arguments**

- `dpSpectr`: a vector specifying the input spectrum
- `iOrder`: polynomial order of Savitzky-Golay filter
- `iFrameLen`: Savitzky-Golay frame length in ppm scale
- `j`: order of derivative

**Value**

jth derivative of the spectrum

**Author(s)**

Lyamine Hedjazi

**See Also**

`sgolay`

**Examples**

```R
## Data
Sp=matrix(rnorm(10*13454,mean=0,sd=1), nrow=10,ncol=13454)

## Peak picking
Spectrum<-Sp[10,]
iOrder <- 3
iFrameLen<- 11
j<-2

SpDerivs<-sgolayDeriv(Spectrum,iOrder,iFrameLen,j)
```
simple.plot

Plot NMR profile plus SRV regions

Description
Plot NMR profile plus SRV regions and consensus across the various statistics

Usage
simple.plot(file, lo, hi, k, title)

Arguments
- file: a text file containing NMR data
- lo: starting point on the chemical axis
- hi: ending point on the chemical axis
- k: number of samples
- title: title of the plot

Value
NMR profile and SRV region plot with peak calling consensus

Author(s)
Jean-Baptiste Cazier

See Also
SRV.plot

Examples
# Load data files
load_datafiles()
# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleangen)
# Plot NMR profile
simple.plot(file=cleandat, lo=3.02, hi=3.08, k=1:20, title="NMR profile")
SRV

Statistical Recoupling of Variables

Description

Base function for dimensionality reduction by statistical recoupling of variables

Usage

SRV(X, minsize, correl, clustf = median)

Arguments

- X: matrix of metabolomic data
- minsize: a numerical value defining the singlet size
- correl: a numerical value defining the bucketting resolution
- clustf: a numerical value defining the correlation threshold

Value

A list:
- indicesdeb: a vector indicating the starting border of superclusters
- indicesfinf: a vector indicating the ending border of superclusters
- Xcluster: matrix of reduced data

Author(s)

Jean-Baptiste Cazier

References


See Also

pre_mQTL

Examples

# Load data files
load_datafiles()

Sp<-read.table(phenofile, as.is=TRUE, header=TRUE, sep='\t')

# Perform the SRV analysis to reduce the number of dimension of Spectra #data (Sp)

corrT=0.9 # correlation threshold
minsize=10 # singlet size
SRV.plot

met = "rectangle"  # summary measure
SRV <- SRV(t(Sp), minsize, corrT, clustf = met)

---

**SRV.plot**  
*Plot SRV clusters*

**Description**

Plot arrows defined by SRV on data

**Usage**

SRV.plot(file1, file2, lo, hi, k, title)

**Arguments**

- `file1`: a text file with NMR data
- `file2`: a text file with SRV results
- `lo`: starting point on chemical shift
- `hi`: ending point on chemical shift
- `k`: number of samples
- `title`: title of the plot

**Author(s)**

Lyamine Hedjazi

**See Also**

simple.plot

**Examples**

# Load data files
load_datafiles()

# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleaneng)

## Plot SRV profile
SRV.plot(file1 = cleandat, file2 = rectangle_SRV, lo = 3.02, hi = 3.08, k = 1:168, title = "Cluster plot")
SRV_lod.plot  
Plot top lod SRV clusters

Description
Plot all SRV clusters associated with the top lod locus

Usage
SRV_lod.plot(results, file, Th)

Arguments
results  a list specifying the results of mQTL mapping
file     a text file contains resulting clusters
Th       a numerical value of LOD threshold

Author(s)
Lyamine Hedjazi

See Also
SRV.plot

Examples
load_datafiles()
load(results)
## Plot LOD profile
SRV_lod.plot(results, rectangle_SRV, T=1)

summary_mQTL  
Function to summarize the mQTL mapping results of all the runs and their differences

Description
This function generates a table containing the genetic markers and their associated metabolomic variables and estimated LOD score.

Usage
summary_mQTL(results, redfile, Th = 5)
Top_SRV.plot

Arguments
- `results`: a list specifying the mQTL mapping results
- `redfile`: a text file containing the parameters of identified clusters (.PPM file)
- `Th`: a numerical parameter indicating the threshold of top accepted score (LOD or -log10(p-value))

Details
Generates a text file containing a table of summary of mQTL mapping results

Value
returns Summaries

Author(s)
Jean-Baptiste Cazier and Lyamine Hedjazi

See Also
pre_mQTL

Examples
```
load_datafiles()
load(results)
Th<10 ## LOD threshold
summary_mQTL(results,rectangle_SRV,Th)## summarizes mQTL results in a table
```

---

**Top_SRV.plot**

*Plot top SRV clusters*

Description
Plot lines defined by SRV on top SRV clusters

Usage
```
Top_SRV.plot(file1,file2,results,met,intMeth,clustidx)
```

Arguments
- `file1`: a text file with NMR data
- `file2`: a text file with SRV clusters
- `results`: a list containing results of mQTL mapping
- `met`: a character specifying the summarizing statistical measure of peaks
- `intMeth`: a character specifying summarizing method across samples (“mean” or “max”)
- `clustidx`: index specifying the SRV cluster of interest (optional)
Author(s)
Lyamine Hedjazi

See Also
SRV.plot

Examples

load_datafiles()
load(results)

# Format data
format_mQTL(phenofile, genofile, physiodat, cleandat, cleangen)

## Plot SRV profile
Top_SRV.plot(file1=cleandat, file2=rectangle_SRV, results=results, met=met, intMeth="mean")
Index

∗Topic function
alignSp, 4
attachSegments, 7
circle_mQTL, 8
configureRSPA, 9
format_mGWA, 10
format_mQTL, 11
load_datafiles, 12
load_demo_data, 12
matchSegments, 13
post_mQTL, 18
ppersp, 19
pplot, 20
process_mGWA, 22
selectRefSp, 25
setupRSPA, 26
sgolay, 27
sgolayDeriv, 28
simple.plot, 29
SRV.plot, 31
SRV_lod.plot, 32
Top_SRV.plot, 33

∗Topic methods
align_mQTL, 5
normalise, 15
normalise_mQTL, 16
peakPeaks, 17
pre_mQTL, 21
process_mQTL, 23
segmentateSp, 24
SRV, 30
summary_mQTL, 32

align_mQTL, 5, 5, 11
alignSp, 4, 6, 26
attachSegments, 6, 7, 14, 25

circle_mQTL, 8
configureRSPA, 9, 26

format_mGWA, 10, 23
format_mQTL, 6, 10, 11, 12, 13

load_datafiles, 12

matchSegments, 6, 8, 13, 25
mQTL.NMR (mQTL.NMR-package), 2
mQTL.NMR-package, 2

normalise, 15, 16
normalise_mQTL, 15, 16

peakPeaks, 17
post_mQTL, 18, 21, 24
ppersp, 19, 20
pplot, 9, 19, 20
pre_mQTL, 18, 21, 30, 33
process_mGWA, 10, 22
process_mQTL, 23

segmentateSp, 6, 24
selectRefSp, 25
setupRSPA, 9, 26
sgolay, 27, 28
sgolayDeriv, 17, 27, 28
simple.plot, 29, 31
SRV, 21, 30
SRV.plot, 29, 31, 32, 34
SRV_lod.plot, 32
summary_mQTL, 32

Top_SRV.plot, 33