Package ‘speaq’

January 14, 2015

Type Package

Title Tools for Nuclear Magnetic Resonance (NMR) spectrum alignment and quantitative analysis.

Version 1.2.0

Date 2015-10-01

Author Trung Nghia Vu, Kris Laukens and Dirk Valkenborg

Maintainer Trung Nghia Vu <nghiavtr@gmail.com>

Description We introduce a novel suite of informatics tools for the quantitative analysis of NMR metabolomic profile data. The core of the processing cascade is a novel peak alignment algorithm, called hierarchical Cluster-based Peak Alignment (CluPA). The algorithm aligns a target spectrum to the reference spectrum in a top-down fashion by building a hierarchical cluster tree from peak lists of reference and target spectra and then dividing the spectra into smaller segments based on the most distant clusters of the tree. To reduce the computational time to estimate the spectral misalignment, the method makes use of Fast Fourier Transformation (FFT) cross-correlation. Since the method returns a high-quality alignment, we can propose a simple methodology to study the variability of the NMR spectra. For each aligned NMR data point the ratio of the between-group and within-group sum of squares (BW-ratio) is calculated to quantify the difference in variability between and within predefined groups of NMR spectra. This differential analysis is related to the calculation of the F-statistic or a one-way ANOVA, but without distributional assumptions. Statistical inference based on the BW-ratio is achieved by bootstrapping the null distribution from the experimental data. Related publication is available at http://www.biomedcentral.com/1471-2105/12/405/.

Depends R (>= 3.0.0), MassSpecWavelet

Imports graphics,stats

License Apache License 2.0

URL https://github.com/nghiavtr/speaq

R topics documented:

speaq-package 2
BWR 3
createNullSampling 4
detectSpecPeaks 5
dohCluster 7
dohClusterCustommedSegments 8

doShift 10
drawBW 11
drawSpec 13
findRef 14
findSegPeakList 16
findShiftStepFFT 17
hClustAlign 18
makeSimulatedData 19
returnLocalMaxima 20

Index **22**

`speaq-package` *Tools for Nuclear Magnetic Resonance (NMR) spectrum alignment and quantitative analysis.*

Description

We introduce a novel suite of informatics tools for the quantitative analysis of NMR metabolomic profile data. The core of the processing cascade is a novel peak alignment algorithm, called hierarchical Cluster-based Peak Alignment (CluPA). The algorithm aligns a target spectrum to the reference spectrum in a top-down fashion by building a hierarchical cluster tree from peak lists of reference and target spectra and then dividing the spectra into smaller segments based on the most distant clusters of the tree. To reduce the computational time to estimate the spectral misalignment, the method makes use of Fast Fourier Transformation (FFT) cross-correlation. Since the method returns a high-quality alignment, we can propose a simple methodology to study the variability of the NMR spectra. For each aligned NMR data point the ratio of the between-group and within-group sum of squares (BW-ratio) is calculated to quantify the difference in variability between and within predefined groups of NMR spectra. This differential analysis is related to the calculation of the F-statistic or a one-way ANOVA, but without distributional assumptions. Statistical inference based on the BW-ratio is achieved by bootstrapping the null distribution from the experimental data. Related publication is available at http://www.biomedcentral.com/1471-2105/12/405/.

Details

Package:	speaq
Type:	Package
Version:	1.2.0
Date:	2015-10-01
License:	Apache License 2.0
LazyLoad:	yes

Author(s)

Trung Nghia Vu, Kris Laukens and Dirk Valkenborg

Maintainer: Trung Nghia Vu <nghiavtr@gmail.com>

References

Vu TN, Valkenborg D, Smets K, Verwaest KA, Dommisse R, Lemie‘re F, Verschoren A, Goethals B, Laukens K. (2011) An integrated workflow for robust alignment and simplified quantitative analysis of NMR spectrometry data. BMC Bioinformatics. 2011 Oct 20;12:405.

Examples

```

#load testing data
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;

# read manual to see how to run detectSpecPeaks, findRef, dohCluster, etc
```

BWR *BW ratio calculation*

Description

Compute the BW ratios from data groups

Usage

```
BWR(X,
    groupLabel)
```

Arguments

`X`	The spectral dataset in the matrix format in which each row contains a single sample
`groupLabel`	Group label of samples in the dataset.

Details

Compute the BW ratios from data groups

Value

Return BW ratio

Author(s)

Trung Nghia Vu

See Also

`createNullSampling`

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;



peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);


resFindRef<- findRef(peakList);
refInd <- resFindRef$refInd;


maxShift = 50;
Y <- dohCluster(X,
                peakList = peakList,
                refInd = refInd,
                maxShift  = maxShift,
                acceptLostPeak = TRUE, verbose=FALSE);

# find the BW-statistic
BW = BWR(Y, groupLabel);
```

createNullSampling *Building a null hypothesis data*

Description

Create a null sampling data (N times) and write them to a file

Usage

```
createNullSampling(X, groupLabel, N = 100, verbose=TRUE)
```

Arguments

X	The spectral dataset in the matrix format in which each row contains a single sample
groupLabel	Group label of samples in the dataset.
N	The number of iteration for creating null sample distribution
verbose	A boolean value to allow print out process information.

Details

Create a null sampling data (N times) and write them to a file

Value

A matrix with N rows containing the null distribution.

Author(s)

Trung Nghia Vu

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;



peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);


resFindRef<- findRef(peakList);
refInd <- resFindRef$refInd;


maxShift = 50;
Y <- dohCluster(X,
                peakList = peakList,
                refInd = refInd,
                maxShift  = maxShift,
                acceptLostPeak = TRUE, verbose=FALSE);


# find the BW-statistic
BW = BWR(Y, groupLabel);

H0 = createNullSampling(Y, groupLabel, N = 100,verbose=FALSE)
```

detectSpecPeaks *Peak detection for spectra*

Description

Divide the whole spectra into smaller segments and detect peaks by using MassSpecWavelet package. Note that, the peak lists could be found by using other methods, this function is just a choice.

Usage

```
detectSpecPeaks(X, nDivRange = 128, scales = seq(1, 16, 2),
        baselineThresh = 50000, SNR.Th = -1, verbose=TRUE)
```

Arguments

`X` The spectral dataset in matrix format in which each row contains a single sample

`nDivRange` The size of a single small segment after division of spectra

`scales` The parameter of peakDetectionCWT function of MassSpecWavelet package, look it up in the original function.

`baselineThresh` It will remove all peaks under an intensity set by baselineThresh.

`SNR.Th` The parameter of peakDetectionCWT function of MassSpecWavelet package, look it up in the original function. If you set -1, the function will itseff recompute this value.

`verbose` A boolean value to allow print out process information.

Details

Divide the whole spectra into smaller segments and detect peaks by using MassSpecWavelet package. Note that, the peak lists could be found by using other methods, this function is just a choice.

Value

The peak lists of the spectra

Author(s)

Trung Nghia Vu

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;



peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);
```

dohCluster *CluPA function for multiple spectra.*

Description

Use CluPA for alignment for multiple spectra.

Usage

```
dohCluster(X,
           peakList,
           refInd = 0,
           maxShift = 100,
           acceptLostPeak = TRUE,
           verbose=TRUE)
```

Arguments

X	The spectral dataset in the matrix format in which each row contains a single sample
peakList	The peak lists of the spectra
refInd	The index of the reference spectrum.
maxShift	The maximum number of the points for a shift step.
acceptLostPeak	This is an option for users, TRUE is the default value. If the users believe that all the peaks in the peak list are true positive, change it to FALSE.
verbose	A boolean value to allow print out process information.

Details

Use CluPA for alignment for multiple spectra.

Value

The aligned spectra.

Author(s)

Trung Nghia Vu

See Also

dohClusterCustommedSegments

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;
```

```
peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);


resFindRef<- findRef(peakList);
refInd <- resFindRef$refInd;


maxShift = 50;
Y <- dohCluster(X,
                peakList = peakList,
                refInd = refInd,
                maxShift  = maxShift,
                acceptLostPeak = TRUE, verbose=FALSE);
```

dohClusterCustommedSegments

Use CluPA for alignment with additional information

Description

This function integrates some additional information from user such as references for each specific segment, segment ignorance, maximum step size.. to align spectra using CluPA.

Usage

```
dohClusterCustommedSegments(X,
                               peakList,
                               refInd,
                               maxShift,
                               acceptLostPeak = TRUE,
                               segmentInfoMat,
                               minSegSize = 128,
                               verbose=TRUE)
```

Arguments

X	The spectral dataset in matrix format in which each row contains a single sample.
peakList	The peak lists of the spectra.
refInd	The index of the reference spectrum.
maxShift	The maximum number of points for a shift step.
acceptLostPeak	This is an option for users, TRUE is the default value. If the users believe that all the peaks in the peak list are true positive, change it to FALSE.
segmentInfoMat	The matrix containing the additional information for segments from the users. This parameter must be a matrix.

`minSegSize`	The minimum size of the segments which could be considered for alignment.
`verbose`	A boolean value to allow print out process information.

Details

Each row of the segmentInfoMat matrix includes 5 values. For example, it could be imported from a CSV file consisting of following content:

#

begin,end,forAlign,ref,maxShift

100,200,0,0,0

450,680,1,0,50

#

Each column could be explained as the following:

- begin: the starting point of the segment.

- end: the end point of the segment.

- forAlign: the segment is aligned (1) or not (0).

- ref: the index of the reference spectrum. If 0, the algorithm will select the reference found by the reference finding step.

- maxShift: the maximum number of points of a shift to left/right.

It is worth to note that only segments with forAlign=1 (column 3) will be taken into account for spectral alignment.

Value

The aligned spectral segments.

Author(s)

Trung Nghia Vu

See Also

`dohCluster`

Examples

```
 cat("\n Please see more examples in the vignettes file.")

res=makeSimulatedData();
X=res$data;
groupLabel=res$label;



peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);
```

```
resFindRef<- findRef(peakList);
refInd <- resFindRef$refInd;


segmentInfoMat=matrix(data=c(100,200,0,0,0,
                      50,680,1,0,50),nrow=2,ncol=5,byrow=TRUE
                      )
colnames(segmentInfoMat)=c("begin","end","forAlign","ref","maxShift")
segmentInfoMat

maxShift = 50;
Yc <- dohClusterCustommedSegments(X,
                                 peakList = peakList,
                                 refInd = refInd,
                                 maxShift  = maxShift,
                                 acceptLostPeak = TRUE,
                                 segmentInfoMat = segmentInfoMat,
                                 minSegSize = 128,
                                 verbose=FALSE)
```

doShift *Segment shift*

Description

Move a spectral segment of a sample shiftStep points to right or left

Usage

```
doShift(specSeg,
    shiftStep)
```

Arguments

specSeg	The segment which needs to be shifted
shiftStep	The shift step for moving. If it is a negative (positive) value, the segment is moved to left (right).

Details

Move a spectral segment of a sample shiftStep points to right or left

Value

The new segment after shifting.

Author(s)

Trung Nghia Vu

See Also

hClustAlign,findShiftStepFFT

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;


maxShift=50;
refSpec=X[1,];
tarSpec=X[2,];
adj=findShiftStepFFT(refSpec, tarSpec,maxShift=maxShift);
newTarSpec=doShift(tarSpec,adj$stepAdj);
```

---

drawBW *BW and percentile ratios plot*

Description

This function is used to plot BW and percentile ratios

Usage

```
drawBW(BW,
      perc,
      X,
      startP = -1,
      endP = -1,
      groupLabel = NULL,
      highBound = -1,
      lowBound = -1,
      nAxisPos = 4,
      offside = 0)
```

Arguments

BW	An array of the BW ratios.
perc	An array of the percentile ratios.
X	The spectral dataset in matrix format in which each row contains a single sample.
startP	The starting point of the segment. If it is -1, the starting point is from begining of the spectra.
endP	The ending point of the segment. If it is -1, the ending point is the last point of the spectra.
groupLabel	The default value is NULL, it means that a single spectrum has a distinct color. Otherwise, the spectra is colored by their label.

`highBound` Default value is -1, that means the plot covers also the highest intesity peaks in the figure. If the users want to limit the upper height of the figure, set this parameter by the limited value.

`lowBound` Default value is -1, that means the plot covers also the lowest intesity peaks in the figure. If the users want to limit the under height of the figure, set this parameter by the limited value.

`nAxisPos` The number of ticks that will be displayed in the horizontal axis.

`offside` The offside of values in x-axis for display.

Details

This function is used to plot BW and percentile ratios

Value

Return a plot containing both the BW and the spectra.

Author(s)

Trung Nghia Vu

See Also

`drawSpec`

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;


peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);


resFindRef<- findRef(peakList);
refInd <- resFindRef$refInd;


maxShift = 50;
Y <- dohCluster(X,
                peakList = peakList,
                refInd = refInd,
                maxShift  = maxShift,
                acceptLostPeak = TRUE, verbose=FALSE);

# find the BW-statistic
BW = BWR(Y, groupLabel);
```

```
N = 100;
alpha = 0.05;

# create sampled H0 and export to file
H0 = createNullSampling(Y, groupLabel, N = N,verbose=FALSE)

#compute percentile of alpha
perc = double(ncol(Y));
alpha_corr = alpha/sum(returnLocalMaxima(Y[2,])$pkMax>50000);
for (i in 1 : length(perc)){
    perc[i] = quantile(H0[,i],1-alpha_corr, type = 3);
}

drawBW(BW, perc,Y, groupLabel = groupLabel)
```

drawSpec *Spectral plot*

Description

This function allows to draw a segment or the whole spectra with limited high/low bounds of intensity.

Usage

```
drawSpec(X,
          startP = -1,
            endP = -1,
            groupLabel = NULL,
            useLog = -1,
            highBound = -1,
            lowBound = -1,
            xlab = NULL,
            ylab = NULL,
            main = NULL,
            nAxisPos = 4,
            offside = 0)
```

Arguments

X	The spectral dataset in matrix format in which each row contains a single sample.
startP	The starting point of the segment. If it is -1, the starting point is from begining of the spectra.
endP	The ending point of the segment. If it is -1, the ending point is the last point of the spectra.
groupLabel	The default value is NULL, it means that a single spectrum has a distinct color. Otherwise, the spectra is colored by their label.
useLog	The default value is -1, that means do not use a logarit transformation. If users want to transform the intensities to logarit values before ploting, set it to 1.

highBound	Default value is -1, that means the plot covers also the highest intesity peaks in the figure. If the users want to limit the upper height of the figure, set this parameter by the limited value.
lowBound	Default value is -1, that means the plot covers also the lowest intesity peaks in the figure. If the users want to limit the under height of the figure, set this parameter by the limited value.
xlab	The default value is NULL, if so, "index" is displayed at the horizontal axis.
ylab	The default value is NULL, if so, "intensity" is displayed at the vertical axis.
main	The default value is NULL, if so, the title shows the values of startP and endP
nAxisPos	The number of ticks that you want to display in horizontal axis.
offside	The offside of values in x-axis for display.

Details

This function allows to draw a segment or the whole spectra with limited high/low bounds of intensity.

Value

Return a plot of the spectra.

Author(s)

Trung Nghia Vu

See Also

drawBW

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;

drawSpec(X)
```

findRef *Reference finding*

Description

This function is to heuristically detect a reference spectrum.

Usage

```
findRef(peakList)
```

Arguments

`peakList`	The peak lists of the spectra.

Details

This function is to heuristically detect a reference spectrum, see the reference for more details.

Value

`refInd`	The index of the reference spectrum found by the algorithm
`orderSpec`	A sorted array of the spectra by their goodness values

Author(s)

Trung Nghia Vu

References

Vu TN, Valkenborg D, Smets K, Verwaest KA, Dommisse R, Lemie‘re F, Verschoren A, Goethals B, Laukens K. (2011) An integrated workflow for robust alignment and simplified quantitative analysis of NMR spectrometry data. BMC Bioinformatics. 2011 Oct 20;12:405.

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;


peakList <- detectSpecPeaks(X,
    nDivRange = c(128),
    scales = seq(1, 16, 2),
    baselineThresh = 50000,
    SNR.Th = -1,
    verbose=FALSE
);


cat("\n Find the spectrum reference...")
resFindRef<- findRef(peakList);
refInd <- resFindRef$refInd;

cat("\n Order of spectrum for reference  \n");
for (i in 1:length(resFindRef$orderSpec))
    cat(paste(i, ":",resFindRef$orderSpec[i],sep=""), " ");

cat("\n The reference is: ", refInd);
```

findSegPeakList *Selecting the peaks in a segment*

Description

This function is to find out which peaks belonging to a segment which ranges from startP to endP

Usage

```
findSegPeakList(peakList,
                startP,
                endP)
```

Arguments

peakList	The peak lists of the spectra.
startP	The starting point of the segment.
endP	The ending point of the segment.

Details

This function is to find out which peaks belonging to a segment which ranges from startP to endP

Value

The list of indices of the peaks in the segment.

Author(s)

Trung Nghia Vu

See Also

dohClusterCustommedSegments

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;


peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);

cat("\n ", peakList[[1]])
segmentpeakList= findSegPeakList(peakList[[1]],400,600);
```

```
cat("\n ", segmentpeakList)
```

findShiftStepFFT *Finding the shift-step by using Fast Fourier Transform cross-correlation*

Description

This function uses Fast Fourier Transform cross-correlation to find out the shift step between two spectra.

Usage

```
findShiftStepFFT(refSpec,
        tarSpec,
        maxShift = 0)
```

Arguments

refSpec	The reference spectrum.
tarSpec	The target spectrum which needs to be aligned.
maxShift	The maximum number of points for a shift step. If this value is zero, the algorithm will check on the whole length of the spectra.

Details

Finding the shift-step by using Fast Fourier Transform cross-correlation

Value

corValue	The best correlation value
stepAdj	The shift step found by the algorithm

Author(s)

Trung Nghia Vu

See Also

hClustAlign

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;


maxShift=50;
refSpec=X[1,];
```

```
tarSpec=X[2,];
adj=findShiftStepFFT(refSpec, tarSpec,maxShift=maxShift);
```

hClustAlign *CluPA function for two spectra.*

Description

This function implements the idea of the CluPA algorithm to align the target spectrum against the reference spectrum.

Usage

```
hClustAlign(refSpec,
        tarSpec,
        peakList,
        peakLabel,
        startP,
        endP,
        distanceMethod = "average",
        maxShift = 0,
        acceptLostPeak = FALSE)
```

Arguments

`refSpec` The reference spectrum.

`tarSpec` The target spectrum.

`peakList` List of peaks of the both reference and target spectra

`peakLabel` The list of the labels of the peaks

`startP` The starting point of the segment.

`endP` The ending point of the segment.

`distanceMethod` The distance method for the hierarchial clustering algorithm.

`maxShift` The maximum number of points for a shift step.

`acceptLostPeak` This is an option for users, TRUE is the default value. If the users believe that all the peaks in the peak list are true positive, change it to FALSE.

Details

This function implements the idea of the CluPA algorithm to align the target spectrum against the reference spectrum.

Value

`tarSpec` The target spectrum after alignment

`peakList` The peak list after alignment

Author(s)

Trung Nghia VU

References

Vu TN, Valkenborg D, Smets K, Verwaest KA, Dommisse R, Lemie're F, Verschoren A, Goethals B, Laukens K. (2011) An integrated workflow for robust alignment and simplified quantitative analysis of NMR spectrometry data. BMC Bioinformatics. 2011 Oct 20;12:405.

See Also

dohCluster

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;


peakList <- detectSpecPeaks(X,
  nDivRange = c(128),
  scales = seq(1, 16, 2),
  baselineThresh = 50000,
  SNR.Th = -1,
    verbose=FALSE
);


resFindRef<- findRef(peakList);
refInd <- resFindRef$refInd;

tarInd=1;
refSpec=X[refInd,];
tarSpec=X[tarInd,];

mergedPeakList=c(peakList[[refInd]],peakList[[tarInd]]);
mergedPeakLabel=double(length(mergedPeakList));
for (i in 1:length(peakList[[refInd]]) ) mergedPeakLabel[i]=1;
startP=1;
endP=length(tarSpec);
res=hClustAlign(refSpec,tarSpec,mergedPeakList,mergedPeakLabel,startP,endP,
        maxShift=50,acceptLostPeak=TRUE)
```

makeSimulatedData *Create a simulated NMR spectral data*

Description

Generate an NMR spectral data for testing.

Usage

```
makeSimulatedData()
```

Details

We generate a NMR spectral data sets that contains two group A and group B. One at around 300 has a single tip and the other at around 600 has double tips that intentionally contains biological variation. First, a single spectrum is created based on statistic information (mean, standard deviation of intensity) achieved from real NMR spectra. Then, we randomely shift the spectrum to maximum 50 data points and add some biological and technical variations to each point intensity to the spectrum to create a new spectrum. The collection of spectra from each group is the final dataset.

Value

`data`	The simulated NMR spectral data matrix
`label`	Group label of each spectrum

Author(s)

Trung Nghia Vu

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;
```

`returnLocalMaxima` *Local maximum detection*

Description

Find and return local maximum of a single spectrum.

Usage

```
returnLocalMaxima(spectrum)
```

Arguments

`spectrum`	A spectral sample in the vector format.

Details

Find and return local maximum of a single spectrum.

Value

`locMax`	Locations of the found local maximum peaks
`pkMax`	Intensities of the found local maxumum peaks

Author(s)

Trung Nghia Vu

Examples

```
res=makeSimulatedData();
X=res$data;
groupLabel=res$label;


returnLocalMaxima(X[2,])
```

Index

∗Topic **Alignment**
hClustAlign, 18
∗Topic **BW**
BWR, 3
drawBW, 11
∗Topic **CluPA**
dohCluster, 7
dohClusterCustommedSegments, 8
hClustAlign, 18
∗Topic **Data**
makeSimulatedData, 19
∗Topic **FFT**
findShiftStepFFT, 17
∗Topic **Simulated**
makeSimulatedData, 19
∗Topic **\textasciitildekwd1**
doShift, 10
∗Topic **\textasciitildekwd2**
doShift, 10
∗Topic **alignment**
dohCluster, 7
dohClusterCustommedSegments, 8
∗Topic **cross-correlation**
findShiftStepFFT, 17
∗Topic **null hypothesis**
createNullSampling, 4
∗Topic **package**
speaq-package, 2
∗Topic **peak detection**
detectSpecPeaks, 5
∗Topic **peak list**
findSegPeakList, 16
∗Topic **plot**
drawBW, 11
drawSpec, 13
∗Topic **reference**
findRef, 14
∗Topic **segment**
dohClusterCustommedSegments, 8
drawSpec, 13
findSegPeakList, 16
∗Topic **spectra**
drawSpec, 13
∗Topic **spectrum shift**
findShiftStepFFT, 17

BWR, 3

createNullSampling, *3*, 4

detectSpecPeaks, 5
dohCluster, 7, *9*, *19*
dohClusterCustommedSegments, *7*, 8, *16*
doShift, 10
drawBW, 11, *14*
drawSpec, *12*, 13

findRef, 14
findSegPeakList, 16
findShiftStepFFT, *11*, 17

hClustAlign, *11*, *17*, 18

makeSimulatedData, 19

returnLocalMaxima, 20

speaq *(speaq-package)*, 2
speaq-package, 2