# Package: cliqueMS (via r-universe)

November 25, 2026

**Type** Package

**Title** Annotation of Isotopes, Adducts and Fragmentation Adducts for in-Source LC/MS Metabolomics Data

**Version** 1.28.0

**Date** 2024-25-04

**Description** Annotates data from liquid chromatography coupled to mass spectrometry (LC/MS) metabolomics experiments. Based on a network algorithm (O.Senan, A. Aguilar- Mogas, M. Navarro, O. Yanes, R.Guimerà and M. Sales-Pardo, Bioinformatics, 35(20), 2019), 'CliqueMS' builds a weighted similarity network where nodes are features and edges are weighted according to the similarity of this features. Then it searches for the most plausible division of the similarity network into cliques (fully connected components). Finally it annotates metabolites within each clique, obtaining for each annotated metabolite the neutral mass and their features, corresponding to isotopes, ionization adducts and fragmentation adducts of that metabolite.

**License** GPL (>= 2)

**Depends** R (>= 4.3.0)

**Imports** Rcpp (>= 0.12.15), xcms(>= 3.0.0), MSnbase, igraph, coop, slam, matrixStats, methods

**LinkingTo** Rcpp, BH, RcppArmadillo

**URL** http://cliquems.seeslab.net

**BugReports** https://github.com/osenan/cliqueMS/issues

**biocViews** Metabolomics, MassSpectrometry, Network, NetworkInference

**SystemRequirements** C++11

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Collate** 'allGenerics.R' 'allClasses.R' 'buildNetwork.R' 'cliqueMS.R' 'ex.cliqueGroups.R' 'findAnnotation.R' 'findCliques.R' 'findIsotopes.R' 'negative.adinfo.R' 'positive.adinfo.R' 'RcppExports.R'

**Suggests** BiocParallel, knitr, rmarkdown, testthat, CAMERA

**VignetteBuilder** knitr

**Config/pak/sysreqs**
cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev libnetcdf-dev libssl-dev zlib1g-dev

**Repository** Bioconductor 3.24

**Date/Publication** 2026-11-23 22:52:55 UTC

**RemoteUrl** https://github.com/bioc/cliqueMS

**RemoteRef** RELEASE_3_24

**RemoteSha** 2ade4384acd807efb9585cdbe779ceb3ede5c38c

# Contents

anClique . . . . . 2
anClique-class . . . . . 4
cliqueMS . . . . . 6
computeCliques . . . . . 7
createanClique . . . . . 8
createNetwork . . . . . 9
ex.cliqueGroups . . . . . 11
getAnnotation . . . . . 11
getCliques . . . . . 13
getIsotopes . . . . . 15
negative.adinfo . . . . . 16
positive.adinfo . . . . . 16

**Index** **17**

---

`anClique` *'anClique' class constructor*

---

## Description

S4 Class `anClique` for annotating isotopes and adducts in processed m/z data. Features are first grouped based on a similarity network algorithm and then annotation of isotopes and adducts is performed in each group.

## Usage

```
anClique(
  peaklist = data.frame(),
  network = igraph::make_empty_graph(directed = FALSE),
  cliques = list(),
  cliquesFound = FALSE,
  isotopes = data.frame(),
  isoFound = FALSE,
  anFound = FALSE
)
```

## Arguments

| | |
|---|---|
| peaklist | 'data.frame' with feature and annotation information. |
| network | 'igraph' undirected network of similarity. |
| cliques | list with the groups of features |
| cliquesFound | 'TRUE' if cliques have been computed. |
| isotopes | 'data.frame' with isotope annotation. |
| isoFound | 'TRUE' if isotopes have been computed. |
| anFound | 'TRUE' if annotation has been computed. |

## Details

See `help("anClique-class")` for information about the slots and methods of the S4 class 'anClique'.

## Value

A new 'anClique' object with variable values set by the user.

## See Also

createanClique

anClique-class

## Examples

```
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
library(xcms)
mzraw <- MSnbase::readMSData(files = mzfile, mode = "onDisk")
cpw <- CentWaveParam(ppm = 15, peakwidth = c(5,20), snthresh = 10)
mzData <- findChromPeaks(object = mzraw, param = cpw)
ex.anClique <- createanClique(mzdata = mzData)
show(ex.anClique)
```

anClique-class *'anClique' S4 class for annotating isotopes and adducts*

## Description

S4 Class anClique-class for annotating isotopes and adducts in processed m/z data. Features are first grouped based on a similarity network algorithm and then annotation of isotopes and adducts is performed in each group. The class contains the following slots.

## Usage

```
## S4 method for signature 'anClique'
show(object)

## S4 method for signature 'anClique'
getPeaklistanClique(object)

## S4 method for signature 'anClique'
getNetanClique(object)

## S4 method for signature 'anClique'
getIsolistanClique(object)

## S4 method for signature 'anClique'
getlistofCliques(object)

## S4 method for signature 'anClique'
hasAnnotation(object)

## S4 method for signature 'anClique'
hasCliques(object)

## S4 method for signature 'anClique'
hasIsotopes(object)

## S4 replacement method for signature 'anClique'
getIsolistanClique(object) <- value

## S4 replacement method for signature 'anClique'
getNetanClique(object) <- value

## S4 replacement method for signature 'anClique'
getlistofCliques(object) <- value

## S4 replacement method for signature 'anClique'
getPeaklistanClique(object) <- value
```

```
## S4 replacement method for signature 'anClique'
hasAnnotation(object) <- value

## S4 replacement method for signature 'anClique'
hasCliques(object) <- value

## S4 replacement method for signature 'anClique'
hasIsotopes(object) <- value
```

## Arguments

| | |
|---|---|
| `object` | 'anClique' S4 object. |
| `value` | Is the new variable which can be a 'peaklist', a 'network', a 'isotopes' a 'cliques' a 'cliquesFound' a 'isoFound' or 'anFound' and it's set by the user. |

## Value

An 'anClique' object with annotation of isotopes, adducts and fragments, and information about the annotation process.

## Methods (by generic)

- `show`: show information about the object
- `getPeaklistanClique`: get the list of features with current annotation
- `getNetanClique`: get the correlation network
- `getIsolistanClique`: get the table of isotopes
- `getlistofCliques`: get the list of the clique groups
- `hasAnnotation`: is 'TRUE' if annotation has been computed
- `hasCliques`: is 'TRUE' if cliques have been computed
- `hasIsotopes`: is 'TRUE' if isotopes have been computed
- `getIsolistanClique<-`: set the table of isotopes
- `getNetanClique<-`: set the network of correlation
- `getlistofCliques<-`: set the list of clique groups
- `getPeaklistanClique<-`: set the list of features
- `hasAnnotation<-`: set if annotation has been computed
- `hasCliques<-`: set if cliques have been computed
- `hasIsotopes<-`: set if isotopes have been computed

## Slots

`'peaklist'` Is a data.frame with m/z, retention time and intensity information for each feature. It also contains adduct and isotope information if annotation has been performed.

`'network'` Is an igraph undirected network of similarity used to compute groups of features before annotation.

'cliques' Is a list that contains the groups of features. Each id corresponds to a row in the peaklist.

'isotopes' Is a data.frame with the column 'feature' for feature id, column 'charge' for the charge, column 'grade' that starts with 0 and it is 1 for the first isotope, 2 for the second and so on and column 'cluster' which labels each group of features that are isotopes.

'cliquesFound' is TRUE if clique groups have been computed,

'isoFound' is TRUE if isotopes have been annotated,

'anFound' is TRUE if annotation of adducts have been computed.

## See Also

createanClique

## Examples

```
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
library(xcms)
mzraw <- MSnbase::readMSData(files = mzfile, mode = "onDisk")
cpw <- CentWaveParam(ppm = 15, peakwidth = c(5,20), snthresh = 10)
mzData <- findChromPeaks(object = mzraw, param = cpw)
ex.anClique <- createanClique(mzdata = mzData)
show(ex.anClique)
```

---

cliqueMS *'cliqueMS' annotates isotopes and adducts in m/z data*

---

## Description

'cliqueMS' first separates features in the data into different groups. To do this it computes a similarity weighted network from the data, and searches clique groups. This cliques are fully connected components that have higher similarity in inner edges than edges outside cliques.

Once clique groups are computed, annotation of isotopes is first performed. After isotope annotation, adducts are annotated within each group

## Author(s)

**Maintainer**: Oriol Senan Campos <oriol.senan@praenoscere.com>

Authors:

- Antoni Aguilar-Mogas
- Jordi Capellades
- Miriam Navarro
- Oscar Yanes
- Roger Guimera
- Marta Sales-Pardo

## See Also

Useful links:

- http://cliquems.seeslab.net
- Report bugs at https://github.com/osenan/cliqueMS/issues

---

computeCliques *Computes clique groups from a similarity network*

---

## Description

This function splits the features in the network in clique groups. The cliques are fully connected components that have high similarity for inner edges and low similarity for edges outside the clique. This function finds the clique groups that better fit this criteria, moving nodes to different groups until we find the groups that have the best log-likelihood.

## Usage

```
computeCliques(anclique, tol = 1e-05, silent = TRUE)
```

## Arguments

| | |
|---|---|
| anclique | This function uses S4 'anClique' object. Gives warning if clique groups have already been computed. |
| tol | Minimum relative increase in log-likelihood to do a new round of log-likelihood maximisation. |
| silent | If 'FALSE' print on the console the log-likelihood maximization progress. Default is 'TRUE'. |

## Value

It returns an 'anClique' object with the computed clique groups. It adds the column 'cliqueGroup' to the 'peaklist' in the 'anClique' object.

## See Also

getCliques

## Examples

```
library(BiocParallel)
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
msSet <- xcms::xcmsSet(files = mzfile, method = "centWave",
ppm = 15, peakwidth = c(5,20), snthresh = 10,
BPPARAM = BiocParallel::SerialParam())
ex.anClique <- createanClique(msSet)
show(ex.anClique)
```

```
netlist <- createNetwork(msSet, xcms::peaks(msSet), filter = TRUE)
getNetanClique(ex.anClique) <- netlist$network
computeCliques(ex.anClique)
```

---

`createanClique` *'createanClique' generic function to create an object of class 'anClique'.*

---

## Description

`createanClique` creates an 'anClique' object from processed m/z data.e

## Usage

```
createanClique(mzdata)

## S4 method for signature 'xcmsSet'
createanClique(mzdata)

## S4 method for signature 'XCMSnExp'
createanClique(mzdata)
```

## Arguments

| | |
|---|---|
| `mzdata` | An object with processed m/z data. See methods for valid class types. |

## Value

An 'anClique' S4 object with all elements to perform clique grouping, isotope annotation and adduct annotation.

## Functions

- `createanClique(xcmsSet)`: Method for 'xcmsSet' object
- `createanClique(XCMSnExp)`: Method for 'XCMSnExp' object

## See Also

`anClique`

## Examples

```
## Using a 'XCMSnExp' object
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
library(xcms)
mzraw <- MSnbase::readMSData(files = mzfile, mode = "onDisk")
cpw <- CentWaveParam(ppm = 15, peakwidth = c(5,20), snthresh = 10)
mzData <- findChromPeaks(object = mzraw, param = cpw)
```

```
ex.anClique <- createanClique(mzdata = mzData)
show(ex.anClique)

## Using a 'xcmsSet' object
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
msSet <- xcms::xcmsSet(files = mzfile, method = "centWave",
ppm = 15, peakwidth = c(5,20), snthresh = 10)
ex.anClique <- createanClique(msSet)
```

---

createNetwork *Generic function to create a similarity network from processed m/z data*

---

## Description

This function creates a similarity network with nodes as features and weighted edges as the cosine similarity between those nodes. Edges with weights = 0 are not included in the network. Nodes without edges are not included in the network. This network will be used to define clique groups and find annotation within this groups.

## Usage

```
createNetwork(
  mzdata,
  peaklist,
  filter = TRUE,
  mzerror = 5e-06,
  intdiff = 1e-04,
  rtdiff = 1e-04
)

## S4 method for signature 'xcmsSet'
createNetwork(
  mzdata,
  peaklist,
  filter = TRUE,
  mzerror = 5e-06,
  intdiff = 1e-04,
  rtdiff = 1e-04
)

## S4 method for signature 'XCMSnExp'
createNetwork(
  mzdata,
  peaklist,
  filter = TRUE,
  mzerror = 5e-06,
  intdiff = 1e-04,
```

```
  rtdiff = 1e-04
)
```

## Arguments

| | |
|---|---|
| mzdata | An object of class 'xcmsSet' or 'XCMSnExp' with processed m/z data. |
| peaklist | Is a data.frame feature info for m/z data. put each feature in a row and a column 'mz' for mass data, retention time column 'rt' and intensity in column 'maxo'. |
| filter | If TRUE, filter out very similar features that have a correlation similarity > 0.99 and equal values of m/z, retention time and intensity. |
| mzerror | Relative error for m/z, if relative error between two features is below that value that features are considered with similar m/z value. |
| intdiff | Relative error for intensity, if relative error between two features is below that value that features are considered with similar intensity |
| rtdiff | Relative error for retention time, if relative error between two features is below that value that features are considered with similar retention time |

## Details

Signal processing algorithms may output artefact features. Sometimes they produce two artefact features which are almost identical This artefacts may lead to errors in the computation of the clique groups, so it is recommended to set 'filter' = TRUE to drop repeated features.

## Value

This function returns a list with the similarity network and the filtered peaklist if 'filter' = TRUE. If filter = FALSE the peaklist is returned unmodified.

## Functions

- `createNetwork(xcmsSet)`: To use with 'xcmsSet' class
- `createNetwork(XCMSnExp)`: To use with 'XCMSnExp' class

## See Also

`getCliques`

## Examples

```
## Using a 'xcmsSet' object
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
require(xcms)
rawMS <- MSnbase::readMSData(files = mzfile, mode = "onDisk")
cpw <- CentWaveParam(ppm = 15, peakwidth = c(5,20), snthresh = 10)
mzData <- findChromPeaks(rawMS, cpw)
peaklist = as.data.frame(chromPeaks(mzData))
netlist = createNetwork(mzData, peaklist, filter = TRUE)

## Using a 'XCMSnExp' object
```

```
require(xcms)
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
rawMS <- MSnbase::readMSData(files = mzfile, mode = "onDisk")
cpw <- CentWaveParam(ppm = 15, peakwidth = c(5,20), snthresh = 10)
mzData <- findChromPeaks(rawMS, cpw)
peaklist = as.data.frame(chromPeaks(mzData))
netlist = createNetwork(mzData, peaklist, filter = TRUE)
```

---

`ex.cliqueGroups` *Example m/z processed data*

---

## Description

This dataset contains a mass sprectrometry data of metabolite standards MS1 analyses were performed using an UHPLC system (1290 series, Agilent Technologies) coupled to a 6550 ESI-QTOF MS (Agilent Technologies) operated in positive (ESI+) electrospray ionization mode.

The original mzdata, which can be found at "CliqueMS: a computational tool for annotating in-source metabolite ions from LC-MS untargeted metabolomics data based on a coelution similarity network" Senan et al, 2019 Bioinformatics https://doi.org/10.1093/bioinformatics/btz207

The raw data was filtered from scan 0 to 700 with Proteowizard mzconvert in order to have an smaller file

The metabolites in this example set are the following: thymine and uracil

## Usage

```
data(ex.cliqueGroups)
```

## Format

It is an 'xcmsSet' object of one sample with 126 features. Has been obtained with parameters ppm = 15, method = "centWave", peakwidth = c(5,20), snthresh = 10. then features have been splitted into cliques with getCliques, with default parameters and filter = T. Before getCliques it was used set.seed(2).

---

`getAnnotation` *Annotate adducts and fragments*

---

## Description

This function annotates adducts after isotope annotation. For each clique group, it searches for combinations of two or more features compatible with the same neutral mass and two or more adducts in 'adinfo' list. For clique groups than have more than one annotation solution, it scores all possibilities and returns the top five solutions.

## Usage

```
getAnnotation(
  anclique,
  adinfo,
  polarity,
  topmasstotal = 10,
  topmassf = 1,
  sizeanG = 20,
  ppm = 10,
  filter = 1e-04,
  emptyS = -6,
  normalizeScore = TRUE
)
```

## Arguments

| | |
|---|---|
| anclique | Object of class 'anClique' with isotope annotation |
| adinfo | data.frame with columns 'adduct' with adduct name, column 'log10freq' with the log10 frequency of each adduct in the list, column 'massdiff' with the adduct mass diff, column 'nummol' with the number of metabolite's molecule necessary for that adduct and column 'charge' with the charge of that adduct. |
| polarity | Polarity of the adducts, choose between 'positive' or 'negative' |
| topmasstotal | All neutral masses in the group are ordered based on their adduct log-frequencies and their number of adducts. From that list, a number of "topmasstotal" masses are selected for the final annotation. |
| topmassf | In addition to 'topmasstotal', for each feature the list of ordered neutral masses is subsetted to the masses with an adduct in that particular feature. For each sublist, a number 'topmassf' neutral masses are also selected for the final annotation. |
| sizeanG | After neutral mass selection, if a clique group has a number of monoisotopic features bigger than 'sizeanG', the annotation group is divided into non-overlapping annotation groups. Each subdivision is annotated independently. |
| ppm | Relative error in ppm in which we consider two or more features compatible with a neutral mass and two or more adducts in 'adinfo'. |
| filter | This parameter removes redundant annotations. If two neutral masses in the same annotation group have a relative mass difference smaller than 'filter' and the same features and adducts, drop the neutral mass with less adducts |
| emptyS | Score given to non annotated features. If you use your own 'adinfo', do not set 'emptyS' bigger than any adduct log frequency in your list. |
| normalizeScore | If 'TRUE', the reported score is normalized and scaled. Normalized score goes from 0, when it means that the raw score is close to the minimum score (all features with empty annotations), up to 100, which is the score value of the theoretical maximum annotation (all the adducts of the list with the minimum number of neutral masses). |

## Details

The default 'adinfo' lists are 'positive.adinfo' and 'negative.adinfo'. For use load them with 'data(positive.adinfo)' or data(negative.adinfo) commands. Reported scores do not always refer to the entire clique group. There might be features whose annotation is independent from other features of the clique group. This occurs when there are no neutral masses with adducts in both groups of features. Therefore, the clique group is divided in non overlapping regions, called annotation groups. Scores report for these annotation groups. To compare scores between different groups use 'normalizeScore' = TRUE.

If clique groups have a lot of features, there are many combinations of neutral masses and adducts. This could lead to long running times to score the top annotations. Parameters 'topmassf' and 'topmasstotal' are relevant in those cases to drop the less likely neutral masses to speed up the time of computation and still obtain the most plausible annotation. If the clique group is small usually no neutral masses are discarded for the scoring.

## Value

An 'anClique' object with annotation columns added to the peaklist

## Examples

```
data(ex.cliqueGroups)
show(ex.cliqueGroups)
ex.isoAn <- getIsotopes(ex.cliqueGroups)
show(ex.isoAn)
data(positive.adinfo)
ex.adductAn <- getAnnotation(ex.isoAn, positive.adinfo, 'positive')
```

---

# getCliques *Compute clique groups from processed m/z data*

---

## Description

This function splits features in groups to find isotope and adduct annotation within each group. To find them it uses a similarity network. This similarity network has nodes as features and weighted edges as the cosine similarity between features. Once the network is obtained we find clique groups in this network. The clique groups are fully connected components with high similarity in inner edges and lower similarity in edges outside the clique. We move nodes to different groups until we find the groups with the maximum log-likelihood.

## Usage

```
getCliques(
  mzdata,
  filter = TRUE,
  mzerror = 5e-06,
  intdiff = 1e-04,
  rtdiff = 1e-04,
```

```
  tol = 1e-05,
  silent = TRUE
)
```

## Arguments

| | |
|---|---|
| mzdata | An 'object with processed m/z data. Currently supported class types are 'xcmsSet' or 'XCMSnExp. |
| filter | If TRUE, filter out very similar features that have a correlation similarity > 0.99 and equal values of m/z, retention time and intensity. |
| mzerror | Relative error for m/z, if relative error between two features is below that value that features are considered with similar m/z value. |
| intdiff | Relative error for intensity, if relative error between two features is below that value that features are considered with similar intensity. |
| rtdiff | Relative error for retention time, if relative error between two features is below that value that features are considered with similar retention time. |
| tol | Minimum relative increase in log-likelihood to do a new round of log-likelihood maximisation. |
| silent | If 'FALSE' print on the console the log-likelihood maximization progress. Default is 'TRUE'. |

## Details

Signal processing algorithms may output artefact features. Sometimes they produce two artefact features which are almost identical This artefacts may lead to errors in the computation of the clique groups, so it is recommended to set 'filter' = TRUE to drop repeated. features.

## Value

It returns an 'anClique' object with the computed clique groups. It adds the column 'cliqueGroup' to the 'peaklist' in the 'anClique' object.

## See Also

computeCliques createNetwork anClique

## Examples

```
library(BiocParallel)
mzfile <- system.file("standards.mzXML", package = "cliqueMS")
msSet <- xcms::xcmsSet(files = mzfile, method = "centWave",
ppm = 15, peakwidth = c(5,20), snthresh = 10,
BPPARAM = BiocParallel::SerialParam())
ex.cliqueGroups <- getCliques(msSet)
```

getIsotopes *Annotate isotopes*

## Description

This function annotates features that are carbon isotopes based on m/z and intensity data. The monoisotopic mass has to be more intense than the first isotope, the first isotope more intense than the second isotope and so one so forth. Isotopes are annotated within each clique group.

## Usage

```
getIsotopes(anclique, maxCharge = 3, maxGrade = 2, ppm = 10, isom = 1.003355)
```

## Arguments

| | |
|---|---|
| anclique | An 'anClique' object with clique groups computed |
| maxCharge | Maximum charge considered when we test two features to see whether they are isotopes |
| maxGrade | The maximum number of isotopes apart from the monoisotopic mass. A 'maxGrade' = 2 means than we have the monoisotopic mass, first isotope and second isotope |
| ppm | Relative error in ppm to consider that two features have the mass difference of an isotope |
| isom | The mass difference of the isotope |

## Value

It returns an 'anClique' object with isotope annotation. it adds the column 'isotope' to the peaklist in the anClique object

## See Also

getCliques

## Examples

```
data(ex.cliqueGroups)
show(ex.cliqueGroups)
ex.isoAn <- getIsotopes(ex.cliqueGroups)
show(ex.isoAn)
```

## negative.adinfo *Default list of negative charged adducts*

### Description

This is a sorted list of adducts ordered by adducts with charge < -1, adducts with charge = -1 and number of molecules = 1, and adducts with number of molecules > 1. Each of the tree groups of adducts is sorted from smaller to bigger mass difference

### Usage

```
data(negative.adinfo)
```

### Format

This is a dataset of 11 rows and 5 columns, corresponding to 11 different adducts. Column 'adduct' contains adduct names, column 'log10freq' contains the log10 frequency of that adduct in the list, column 'massdiff' contains the mass difference of that adduct, column 'nummol' has the number of molecules of that adduct and column 'charge' has the charge of that adduct.

## positive.adinfo *Default list of positive charged adducts*

### Description

This is a sorted list of adducts ordered by adducts with charge > 1, adducts with charge = 1 and number of molecules = 1, and adducts with number of molecules > 1. Each of the tree groups of adducts is sorted from smaller to bigger mass difference

### Usage

```
data(positive.adinfo)
```

### Format

This is a dataset of 39 rows and 5 columns, corresponding to 39 different adducts. Column 'adduct' contains adduct names, column 'log10freq' contains the log10 frequency of that adduct in the list, column 'massdiff' contains the mass difference of that adduct, column 'nummol' has the number of molecules of that adduct and column 'charge' has the charge of that adduct.

# Index


∗ **datasets**
ex.cliqueGroups, 11
negative.adinfo, 16
positive.adinfo, 16

anClique, 2, *8*, *14*
anClique-class, 4

cliqueMS, 6
cliqueMS-package *(*cliqueMS*)*, 6
computeCliques, 7, *14*
createanClique, *3*, *6*, 8
createanClique,XCMSnExp-method *(*createanClique*)*, 8
createanClique,xcmsSet-method *(*createanClique*)*, 8
createNetwork, 9, *14*
createNetwork,XCMSnExp-method *(*createNetwork*)*, 9
createNetwork,xcmsSet-method *(*createNetwork*)*, 9

ex.cliqueGroups, 11

getAnnotation, 11
getCliques, *7*, *10*, 13, *15*
getIsolistanClique *(*anClique-class*)*, 4
getIsolistanClique,anClique-method *(*anClique-class*)*, 4
getIsolistanClique<- *(*anClique-class*)*, 4
getIsolistanClique<-,anClique-method *(*anClique-class*)*, 4
getIsotopes, 15
getlistofCliques *(*anClique-class*)*, 4
getlistofCliques,anClique-method *(*anClique-class*)*, 4
getlistofCliques<- *(*anClique-class*)*, 4
getlistofCliques<-,anClique-method *(*anClique-class*)*, 4
getNetanClique *(*anClique-class*)*, 4
getNetanClique,anClique-method *(*anClique-class*)*, 4
getNetanClique<- *(*anClique-class*)*, 4
getNetanClique<-,anClique-method *(*anClique-class*)*, 4
getPeaklistanClique *(*anClique-class*)*, 4
getPeaklistanClique,anClique-method *(*anClique-class*)*, 4
getPeaklistanClique<- *(*anClique-class*)*, 4
getPeaklistanClique<-,anClique-method *(*anClique-class*)*, 4

hasAnnotation *(*anClique-class*)*, 4
hasAnnotation,anClique-method *(*anClique-class*)*, 4
hasAnnotation<- *(*anClique-class*)*, 4
hasAnnotation<-,anClique-method *(*anClique-class*)*, 4
hasCliques *(*anClique-class*)*, 4
hasCliques,anClique-method *(*anClique-class*)*, 4
hasCliques<- *(*anClique-class*)*, 4
hasCliques<-,anClique-method *(*anClique-class*)*, 4
hasIsotopes *(*anClique-class*)*, 4
hasIsotopes,anClique-method *(*anClique-class*)*, 4
hasIsotopes<- *(*anClique-class*)*, 4
hasIsotopes<-,anClique-method *(*anClique-class*)*, 4

negative.adinfo, 16

positive.adinfo, 16

show,anClique-method *(*anClique-class*)*, 4