Package 'deisotoper'

December 19, 2017

Type Package

Title Detection of Isotope Pattern of a Mass Spectrometric Measurement

Version 0.0.3

Author Christian Panse [cre, aut] (0000-0003-1975-3064),
Lucas Schmidt [ctb, aut] (0000-0003-4496-0487),
Witold E. Wolski [ctb, aut] (0000-0002-6468-120X)

Maintainer Christian Panse <cp@fgcz.ethz.ch>

Depends R (>= 3.0), rJava (>= 0.9)

Suggests DiagrammeR, lattice, roxygen2, protViz (>= 0.2.45), shiny, testthat, knitr, rmarkdown

SystemRequirements Java JDK 1.8 or higher, GNU make

Description Provides a low-level interface for a deisotoper container implemented in the 'Java' programming language and means of S3 helper functions for plotting and debugging isotopes of mass spectrometric data. The deisotoper algorithm detects and aggregates peaks which belong to the same isotopic cluster of a given mass spectrum.

License GPL-3

URL https://github.com/protViz/deisotoper/

BugReports https://github.com/protViz/deisotoper/issues
deisotoper.R

LazyData true

NeedsCompilation yes

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2017-12-19 11:40:28 UTC

R topics documented:

`deisotope` *Deisotope a Mass Spectrum*

Description

Deisotope a Mass Spectrum

Usage

```
deisotope(deisotoper, massspectrum, algorithm = "features-based")
```

Arguments

`deisotoper`	a `deisotoper` object.
`massspectrum`	a list of numeric vectors `mZ` and `intensity` where `mZ` is ordered and `mZ` and `intensity` have the same length.
`algorithm`	the supported deisotope algorithms, default is `method="features-based"`.

Author(s)

Lucas Schmidt, Christian Panse, Witold E. Wolski

References

- Features-Based Deisotoping Method for Tandem Mass Spectra, http://dx.doi.org/10.1155/2011/210805.

See Also

`deisotoper`

Examples

```
x <- list(mZ = c(1, 2, 2.5, 3), intensity = rep(1, 4), pepmass=600, charge=2)

xd <- deisotope(dtoper <- deisotoper(), x)
plot.deisotoper(x, xd)
summary(dtoper)
```

deisotoper *Construct a Deisotoper Object*

Description

deisotoper returns a deisotoper object.

deisotope detects and aggregates peaks which belong to the same isotopic cluster of a given mass spectrum.

Usage

```
deisotoper(amino_acid_masses = list(A = 71.03711, R = 156.10111, N =
  114.04293, D = 115.02694, C = 103.00919, E = 129.04259, Q = 128.05858, G =
  57.02146, H = 137.05891, I = 113.08406, L = 113.08406, K = 128.09496, M =
  131.04049, F = 147.06841, P = 97.05276, S = 87.03203, T = 101.04768, W =
  186.07931, Y = 163.06333, V = 99.06841), F1 = 0.8, F2 = 0.5, F3 = 0.1,
  F4 = 0.1, F5 = 0.1, delta = 0.003, errortolerance = 0.3,
  distance = 1.00048, noise = 0, decharge = FALSE, modus = "first",
  comment = "")
```

Arguments

amino_acid_masses	List of amino acid masses used for scoring.
F1	F1 multiplier used for scoring.
F2	F2 multiplier used for scoring.
F3	F3 multiplier used for scoring.
F4	F4 multiplier used for scoring.
F5	F5 multiplier used for scoring.
delta	Delta value used for clustering.
errortolerance	Errortolerance used for scoring.
distance	Distance between two peaks used by clustering.
noise	Noise value for noise filtering (in percent).
decharge	De- and activates decharging.
modus	Modus of aggregation ('first' or 'highest').
comment	default is empty word.

Details

Input: a peak peaked mass spectrum.

The algorithm: The deisotoper algorithm detects and aggregates peaks which belong to the same isotopic cluster of a given mass spectrum.

Output:

Value

deisotoper as list of JavaRef

Author(s)

Lucas Schmidt, Christian Panse

References

Features-Based Deisotoping Method for Tandem Mass Spectra http://dx.doi.org/10.1155/2011/210805.

See Also

deisotope

Examples

```
# EXAMPLE 1
# standart configurated deisotoper
dtoper <- deisotoper()

# return the configuration of dtoper
config <- config.deisotoper(dtoper)

# example data
x <- list(mZ = c(110.07172, 111.07504, 129.10249, 130.08649, 147.11302,
   149.04506, 167.05571, 175.11923, 181.06099, 199.07158, 216.09814,
   223.15556, 239.09503, 251.15036, 261.15579, 262.13, 280.14053,
   281.14398, 285.00974, 299.06165, 303.02039, 328.11386, 332.20789,
   344.97641, 345.14056, 350.21866, 355.06995, 360.22412, 368.17529,
   373.08078, 415.03656, 418.99521, 430.2774, 431.28107, 464.26218,
   473.3085, 476.18176, 479.20718, 481.25989, 497.21811, 521.27063,
   521.77087, 540.7804, 559.31946, 560.32391, 580.32739, 582.30688,
   592.28766, 592.35815, 593.34113, 608.25214, 610.30243, 610.36755,
   611.30554, 611.37042, 630.35724, 631.36115, 642.572, 643.054,
   643.569, 644.062, 644.557, 681.37762, 684.31494, 691.36011,
   709.37109, 709.4353, 710.44037, 712.3092, 721.33459, 754.33899,
   774.36261, 790.38892, 791.39124, 792.39221, 813.4679, 820.40479,
   823.41522, 824.40546, 825.39423, 826.39734, 840.47681, 841.43036,
   841.47949, 896.4137, 903.47253, 904.47565, 905.47632, 906.47607,
   924.46271, 951.51038, 969.52002, 970.52283, 1038.50195, 1041.53308,
   1042.53845, 1080.55493, 1081.54773),
          intensity = c(378322, 32496.6, 85689.6, 46440.3, 49645.2,
          25102.5, 32516.2, 83497.2, 74653.1, 37228, 196053, 83826.4,
          112718, 114812, 88089.5, 61521.3, 220054, 58888.5, 280334,
          122311, 14953.2, 26959.6, 24854, 27122.9, 86216.1, 63360.3,
          358968, 47393.5, 37893.2, 16532.9, 17259, 37250.4, 33679.8,
          21243.6, 17854.9, 51232.4, 12738.8, 19515.4, 31560.1, 48772.3,
          66481.2, 23353.6, 11994, 15211, 9883.29, 14753.7, 17304.7,
          51575.9, 10917.6, 40518.3, 15107.3, 62106.4, 72496.1, 9430.4,
          10289.3, 34831.3, 41981.1, 17000, 25000, 12000, 9000, 4000,
```

```
            9579.9, 10392.3, 13507.4, 38200.9, 29990.5, 9304.39, 19849,
            10678.6, 8452.85, 14519.3, 111717, 185030, 56020.8, 3387.69,
            9478.08, 7878.29, 3167.8, 20670.7, 2774.25, 31114.4, 3385.92,
            4656.8, 3687.15, 65332.5, 207097, 68080.9, 11934.3, 3630.86,
            9201.02, 47579.2, 19125.8, 3439.48, 15082.1, 8280.57, 4170.47,
            2603.17),
            title = "TP filtered inserted example 2 of protViz::deisotoper.",
            rtinseconds = 1581,
            charge = 2,
            scan = 1,
            id = 1,
            pepmass = 592.8093)

  # deisotope the data
  xd <- deisotope(dtoper, x)
  summary.deisotoper(dtoper)

  # plot the example data and the deisotoped data
  op <- par(mfrow=c(2,2))
  plot.deisotoper(x, xd)
  plot.deisotoper(x, xd, xlim=c(275,285))
  plot.deisotoper(x, xd, xlim=c(790,795))
  plot.deisotoper(x, xd, xlim=c(901,910))
  par(op)

  # return the annotated spectrum of the above deisotoped data
  print.deisotoper(dtoper)



  # EXAMPLE 2
  # standart configurated deisotoper with changed delta and decharging
  dtoper2 <- deisotoper(delta = 0.005, decharge = TRUE)

  # return the configuration of dtoper2
  config2 <- config.deisotoper(dtoper2)

  ## Not run:
  # return the GraphViz dot graphs of the above deisotoped data
  xdot <- dot.deisotoper(dtoper)

  # draws the isotopic cluster graphs in the browser (html)
    if(require(DiagrammeR)){
      lapply(xdot, DiagrammeR::grViz)
    }

  ## End(Not run)
```

---

findNN *find index of nearest neighbor.*

Description

Given a vector of sorted double values vec of size n and a vector of m query objects q.

findNN determines for each element q[i] in q the nearest neighbor index o so that the following remains true:

there is no element k with $1 \leq k \leq n$ and k is not o so that

abs(vec[k] - q[i]) < abs(vec[o] - q[i]).

The internal algorithm of findNN is implemented as binary search. findNN has $O(m * log(n))$ time complexity.

Usage

```
findNN(q, vec)
```

Arguments

q	a double vector which can be considered as query objects.
vec	a sorted double vector which can be considered as a data base.

Value

an integer vector

Author(s)

Lucas Schmidt, Christian Panse

See Also

protViz::findNN's cpluplus implementation.

Examples

```
(NNidx <- findNN(q<-c(1, 1.0001, 1.24, 1.26), DB<-seq(1,5,by=0.25)))
(NNidx == c(1,1,2,2))

# should be 0
unique(DB[findNN(DB,DB)] - DB)
```

Index