

Package ‘ecipex’

March 14, 2020

Type Package

Title Efficient Calculation of Fine Structure Isotope Patterns via
Fourier Transforms of Simplex-Based Elemental Models

Author Andreas Ipsen

Maintainer Andreas Ipsen <andreas.b.ipsen@gmail.com>

Imports CHNOSZ

Description Provides a function that quickly computes the fine structure isotope patterns of a set of chemical formulas to a given degree of accuracy (up to the limit set by errors in floating point arithmetic). A data-set comprising the masses and isotopic abundances of individual elements is also provided and calculation of isotopic gross structures is also supported.

License GPL (>= 2)

Repository CRAN

LazyData true

Version 1.1

Date 2020-03-12

NeedsCompilation no

Date/Publication 2020-03-13 23:00:10 UTC

R topics documented:

ecipex-package	2
ecipex	2
nistiso	5

Index	6
--------------	----------

ecipex-package	<i>Efficient Calculation of Fine Structure Isotope Patterns via Fourier Transforms of Simplex-Based Elemental Models</i>
----------------	--

Description

Provides a function that quickly computes the fine structure isotope patterns of a set of chemical formulas to a given degree of accuracy (up to the limit set by errors in floating point arithmetic). A data-set comprising the masses and isotopic abundances of individual elements is also provided and calculation of isotopic gross structures is also supported.

Details

Package:	ecipex
Type:	Package
Version:	1.1
Date:	2020-03-12
License:	GPL (>= 2)
LazyLoad:	true

Author(s)

Andreas Ipsen

ecipex	<i>Calculate fine structure isotope pattern</i>
--------	---

Description

This function can calculate either the fine structure isotope patterns or the gross structure isotope patterns for a given set of chemical formulas. It returns the isotope patterns as a list of data frames which may be sorted by mass or abundance.

Usage

```
ecipex(formulas, isoinfo = ecipex::nistiso, limit = 1e-12,  
       id = FALSE, sortby = "abundance", gross = FALSE, groupby = "mass")
```

Arguments

<code>formulas</code>	a character vector specifying the chemical formulas whose isotope patterns are to be calculated. The elements specified must be present in <code>isoinfo</code> and must be compatible with <code>count.elements</code> .
<code>isoinfo</code>	a data frame that specifies the masses and isotopic abundances of the elements to be used in the calculations. Must include the variables <code>element</code> (the chemical symbol), <code>mass</code> and <code>abundance</code> so that each row specifies a unique isotope. Defaults to <code>nistiso</code> .
<code>limit</code>	isotopologues with abundances below this value are ignored.
<code>id</code>	determines whether the full isotopic composition of each isotopic variant should be specified in the output. Setting this to <code>TRUE</code> can increase memory requirements substantially. This argument has no effect if <code>gross</code> is set to <code>TRUE</code> .
<code>sortby</code>	should be one of "abundance" or "mass", depending on whether the output should be sorted by abundance (high to low) or mass (low to high).
<code>gross</code>	if <code>TRUE</code> the gross isotope patterns will be returned instead of the fine structure ones.
<code>groupby</code>	determines how the fine structure isotope patterns are grouped together when the gross structure isotope patterns are calculated. If it is equal to "mass" then isotopologues whose masses get rounded to the same integer number are grouped together. If it is equal to "nucleons" then isotopologues with the same number of nucleons get grouped together. This argument is only used if <code>gross</code> is set to <code>TRUE</code> .

Details

The fine structure isotope pattern of each formula is calculated by applying the multi-dimensional fast Fourier transform to a simplex-based representation of each element's isotopic abundance. The algorithm is most efficient when the atom counts of the same elements in different formulas are roughly similar. Performance can also be improved by increasing `limit` although this should be done with care. It is generally not advisable to reduce `limit` below its default value, as this can increase memory requirements significantly and is unlikely to provide information of any value, since the natural isotopic abundances can be quite variable. If `gross` is set to `TRUE` then the gross structure isotope patterns are calculated directly from the fine structure isotope patterns. Note that for centroids with extremely low abundances (say less than one billionth of the total) the centroided masses can be somewhat inaccurate due to floating point errors.

Value

A list of data frames containing the fine structure isotope patterns of each formula in `formulas`. The list names are determined by `formulas`, so that the isotope pattern of any given formula is easily extracted. If `id` is set to `TRUE`, the output will include additional columns listing the counts of each distinct isotope of each element. If `gross` is set to `TRUE`, the output will instead contain the gross structure isotope patterns of each formula in `formulas`.

References

Ipsen, A., Efficient Calculation of Exact Fine Structure Isotope Patterns via the Multidimensional Fourier Transform, *Anal. Chem.*, 2014, 86 (11), pp 5316-5322

<http://pubs.acs.org/doi/abs/10.1021/ac500108n>

See Also

[nistiso](#)

Examples

```
# a simple molecule
iso_H2O <- ecipex("H2O")[[1]]
iso_H2O

# reduce limit for larger molecule and sort output by mass
iso_C254H338N65075S6 <- ecipex("C254H338N65075S6", limit=1e-8, sortby="mass")[[1]]
head(iso_C254H338N65075S6)

# check that sum of all abundances is still close to 1
sum(iso_C254H338N65075S6$abundance)

# inspect the full isotope pattern, the fine structure, and the full pattern on a log scale
par(mfrow=c(1,3))
plot(iso_C254H338N65075S6, t="h")
plot(iso_C254H338N65075S6, t="h", xlim=c(5691.29, 5691.31))
plot(iso_C254H338N65075S6, t="h", log="y")

# calculate isotopic abundances with enriched Carbon-13
modifiediso <- nistiso
modifiediso[modifiediso$element=="C",3] <- c(0.9, 0.1)
ecipex("C2", isoinfo=modifiediso)

# the isotope pattern can be calculated quickly if the elements only have 2 stable isotopes
system.time(iso_C10000H10000 <- ecipex("C10000H10000", limit=1e-8)[[1]])

# this is typically a more demanding calculation because S has 4 stable isotopes
system.time(iso_S50 <- ecipex("S50", limit=1e-8)[[1]])

# if the limit is greater than the most abundant isotopologue the output is uninformative
iso_C10000H10000_useless <- ecipex("C10000H10000", limit=0.015)

# calculate the isotope patterns of multiple formulas, and include the detailed isotopic composition
multisopatterns <- ecipex(c("H2O", "CO2", "O2", "C8H18", "C60"), sortby="mass", id=TRUE)

# inspect C8H18 in particular
multisopatterns$C8H18

# make sure all abundances are close to 1
sapply(multisopatterns, function(x){sum(x$abundance)})

# due to floating point errors, the following are not identical
iso_C60_almostComplete <- ecipex("C60", limit= 0)[[1]]
iso_C60_reallyComplete <- ecipex("C60", limit= -1)[[1]]
# the latter includes negative isotopic abundances because the floating point errors are orders of
# magnitude greater than the "true" abundances. The variations in natural isotopic abundances will
```

```
# typically be much greater than floating point errors.  
  
# calculate the gross structure isotope pattern, grouping the fine structure isotopologues by mass  
ecipex("C6H14N4O2", sortby="mass", gross=TRUE, groupby="mass")
```

nistiso

Masses, isotope patterns and nucleon numbers of the elements

Description

A data frame giving the masses, standard isotopic abundances and nucleon numbers of most chemical elements as provided by the Physical Measurement Laboratory of the National Institute of Standards and Technology (but taken from separate publications). It includes the four variables element (the chemical symbol), mass, abundance and nucleons, arranged so that each isotope is uniquely specified by one row.

References

<http://www.nist.gov/pml/data/comp.cfm>

Index

- *Topic **datasets**
 - nistiso, [5](#)
- *Topic **data**
 - nistiso, [5](#)
- *Topic **fft**
 - ecipex, [2](#)
- *Topic **fine**
 - ecipex, [2](#)
- *Topic **isotope**
 - ecipex, [2](#)
- *Topic **package**
 - ecipex-package, [2](#)
- *Topic **pattern**
 - ecipex, [2](#)
- *Topic **structure**
 - ecipex, [2](#)

count.elements, [3](#)

ecipex, [2](#)

ecipex-package, [2](#)

nistiso, [3](#), [4](#), [5](#)