Mass decomposition with the Rdisop package

Steffen Neumann[‡], Anton Pervukhin[¶], Sebastian Böcker[¶]

May 3, 2016

[‡]Leibniz Institute of Plant Biochemistry, Department of Stress and Developmental Biology, sneumann@IPB-Halle.DE
[¶]Bioinformatics, Friedrich-Schiller-University Jena, {apervukh|boecker}@minet.uni-jena.de

Contents

1 Introduction

The BioConductor *Rdisop* package is designed to determine the sum formula of metabolites solely from their exact mass and isotope pattern as obtained from high resolution mass spectrometry measurements. Algorithms are described in ????.

It is designed with compatibility to the Bioconductor packages *XCMS*, *MassSpecWavelet* and *rpubchem* in mind.

2 Decomposing isotope patterns

After preprocessing, the output of a mass spectrometer is a list of peaks which corresponds to the masses of the sample molecules and their abundance, i.e., the amount of sample compounds with a certain mass. In fact, sum formulas of small molecules can be identified using only accurate output masses. However, even with very high mass accuracy (< 1 ppm), many chemically possible formulas are found in higher mass regions. It has been shown that applying only this data therefore does not suffice to identify a compound, and more information, such as isotopic abundances, needs to be taken into account. High resolution mass spectrometry allows us to obtain the isotope pattern of sample molecule with outstanding accuracy.

2.1 Chemical background

Atoms are composed of electrons, protons, and neutrons. The number of protons (the atomic number) is fixed and defines what element the atom is. The number of neutrons, on the other hand, can vary: Atoms with the same number of protons but different numbers of neutrons are called *isotopes* of the element. Each of these isotopes occurs in nature with a certain abundance. The *nominal mass* of a molecule is the sum of protons and neutrons of the constituting atoms. The *mass* of the molecule is the sum of masses of these atoms. The *monoisotopic* (*nominal*) *mass* of a molecule is the sum of (nominal) masses of the constituting atoms where for every element its most abundant natural isotope is chosen. Clearly, nominal mass and mass depend on the isotopes the molecule consists of, thus on the *isotope species* of the molecule.

No present-day analysis technique is capable of resolving isotope species with identical nominal mass. Instead, these isotope species appear as one single peak in the mass spectrometry output. For this reason, we merge isotope species with identical nominal mass and refer to the resulting distribution as the molecule's *isotope pattern*.

2.2 Identification schema

Obtaining an accurate isotope pattern from a high resolution mass spectrometer, we apply this information to identify the elemental composition of the sample molecule. Our input is a list of masses with normalized abundances that corresponds to the isotope pattern of the sample molecule. We want to find that molecule's elemental composition whose isotope pattern best matches the input.

Solving this task is divided into the following parts: First, all elemental compositions are calculated that share some property, for example monoisotopic mass, with the input spectrum. Second, to remove those compositions that do not exist in nature, chemical bonding rules are applied, discarding formulas that have negative or non-integer degree of unsaturation. And third, for every remaining composition, its theoretical isotope pattern is calculated and compared to the measured isotope pattern. Candidate patterns are ranked using Bayesian Statistics, and the one with the highest score is chosen.

3 Working with molecules and isotope peaklists

The central object in Rdisop is the molecule, which is a list containing the (sum-)formula, its isotope pattern, a score and other information. Molecules can either be created explicitely through `getMolecule()` or `initializeXXX()`, or through `decomposeMass()` and `decomposeIsotopes()`. Most functions operate only on a subset of the periodic system of elements (PSE) given as "elements" argument.

3.1 Handling of Molecules

The `getMolecule` returns a list object containing the information for a named single atom or a more complex molecule.

```
> library(Rdisop)
> molecule <- getMolecule("C2H5OH")
> getFormula(molecule)

[1] "C2H6O"

> getMass(molecule)

[1] 46.04186
```

Note that the formula is in a canonical form, and the mass includes the decimals (the nominal mass for ethanol would be just 46).

Without further arguments only the elements C, H, N, O, P and S are available. For metabolomics research, these are the most relevant ones. A different subset of the PSE can be returned and passed to the functions, but keep in mind that a larger set of elements yields a (much) larger result set when decomposing masses later.

```
> essentialElements <- initializeCHNOPSMgKCaFe()
> chlorophyll <- getMolecule("C55H72MgN4O5H", z=1,
+   elements=essentialElements)
> isotopes <- getIsotope(chlorophyll, seq(1,4))
> isotopes

            [,1]        [,2]       [,3]         [,4]
[1,] 893.5431390 894.5459934 895.546247 896.54752708
[2,]   0.4140648   0.3171228   0.178565   0.06773657
```

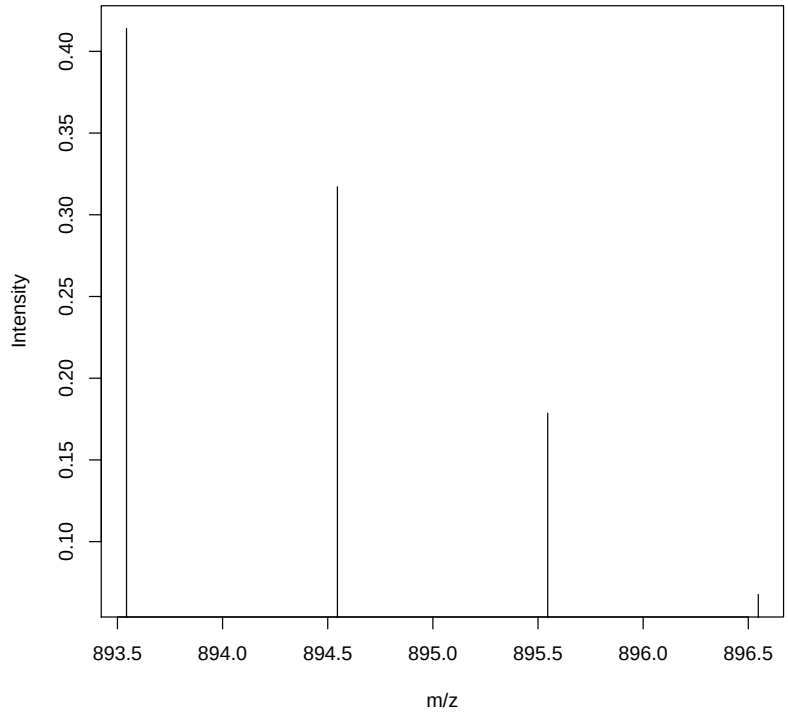

Figure 1: Isotope pattern for a protonated chlorophyll ion, which could be observed on a high-resolution mass spectrometer in positive mode.

```
> plot(t(isotopes), type="h", xlab="m/z", ylab="Intensity")
```

In this case we have created a complex molecule with a charge (z= +1) containing a metal ion and check its first four isotope peaks. For a visual inspection the isotope pattern can be plotted, see figure **??**.

3.2 decomposeMass and decomposeIsotopes

The `decomposeMass` returns a list molecules which have a given exact mass (within an error window in ppm):

```
> molecules <- decomposeMass(46.042, ppm=20)
> molecules

$formula
[1] "C2H6O"
```

```
$score
[1] 1

$exactmass
[1] 46.04186

$charge
[1] 0

$parity
[1] "e"

$valid
[1] "Valid"

$DBE
[1] 0

$isotopes
$isotopes[[1]]
           [,1]        [,2]        [,3]         [,4]         [,5]         [,6]
[1,] 46.0418648 47.04534542 48.04631711 4.904960e+01 5.005324e+01 5.105929e+01
[2,]  0.9749152  0.02293559  0.00210353 4.540559e-05 2.816635e-07 2.324709e-10
             [,7]         [,8]         [,9]        [,10]
[1,] 5.206548e+01 5.307171e+01 5.407796e+01 5.508422e+01
[2,] 8.458096e-14 1.665930e-17 1.857005e-21 1.107409e-25
```

This call produces a list of potential molecules (with a single element in this case). The larger the masses, the allowed ppm deviation and the allowed elements list, the larger the result list will grow. For each hypothesis there is its formula and weight and score. The parity, validity (using the nitrogen rule) and double bond equivalents (DBE) are simple, yet commonly used hints for the plausibility of a solution and can be used for filtering the results list. For an amino acid this simple method guesses already eight hypotheses:

```
> length(decomposeMass(147.053))

[1] 8
```

On modern mass spectrometers a full isotope pattern can be obtained for a molecule, and the masses and intensities improve the accuracy of the sum

formula prediction. Accessor functions return only subsets of the molecule data structure:

```
> # glutamic acid (C5H9NO4)
> masses <- c(147.053, 148.056)
> intensities <- c(93, 5.8)
> molecules <- decomposeIsotopes(masses, intensities)
> cbind(getFormula(molecules), getScore(molecules), getValid(molecules))
```

```
     [,1]       [,2]                   [,3]
[1,] "C5H9NO4"  "0.999999998064664"    "Valid"
[2,] "C3H17P2S" "1.93533578193563e-09" "Invalid"
```

The first ranked solution already has a score close to one, and if using an N-rule filter, only one solution would remain. These cases are not removed by default, because a few compound classes do not obey the N-rule, which after all is just a simple heuristic.

I the masses were obtained by an LC-ESI-MS, it is likely that the measured mass signal actually resembles an adduct ion, such as $[M+H]^+$. The sum formula obtained through `decomposeIsotopes` will have one H too much, and will not be found in PubChem or other libraries, unless the adduct has been removed:

```
> querymolecule <- subMolecules("C5H10NO4", "H")
> getFormula(querymolecule)
```

```
[1] "C5H9NO4"
```

Similarly, if during ionisation an in-source fragmentation occurred, the lost fragment can be added before querying using `addMolecules`.

3.3 Interaction with other BioConductor packages

This section will give some suggestions how the Rdisop functionality can be combined with other BioConductor packages.

Usually the masses and intensities will be obtained from a high-resolution mass spectrometer such as an FTICR-MS or QTOF-MS. BioConductor currently has two packages dealing with peak picking on raw machine data, *MassSpecWavelet* and *XCMS*. The latter contains a wrapper for MassSpecWavelet, so we need to deal with XCMS peak lists only. The ESI package[1] can extract a set of isotope clusters from peak lists.

[1] not part of BioConductor, see `http://msbi.ipb-halle.de/`

After Rdisop has created a set of candidate molecular formulae, the open-access compound databases PubChem or ChEBI can be queried whether any information about this compound exists. Nota bene: a hit or non-hit does not indicate a correct or incorrect formula, but merely helps in further verification or structure elucidation steps. For other cheminformatics functionality in BioConductor see e.g. *RCDK*.

Acknowledgments

AP supported by Deutsche Forschungsgemeinschaft (BO 1910/1), additional programming by Marcel Martin, whom we thank for his unfailing support, and by Marco Kortkamp.