# Package: CorMID (via r-universe)

November 3, 2026

**Type** Package

**Title** Correct Mass Isotopologue Distribution Vectors

**Version** 0.3.1

**Date** 2025-01-03

**Maintainer** Jan Lisec <jan.lisec@bam.de>

**Description** In metabolic flux experiments tracer molecules (often glucose containing labelled carbon) are incorporated in compounds measured using mass spectrometry. The mass isotopologue distributions of these compounds needs to be corrected for natural abundance of labelled carbon and other effects, which are specific on the compound and ionization technique applied. This package provides functions to correct such effects in gas chromatography atmospheric pressure chemical ionization mass spectrometry analyses.

**License** GPL (>= 3)

**URL** https://github.com/janlisec/CorMID

**Depends** R (>= 3.5.0)

**Imports** Rcpp (>= 1.0.13)

**Suggests** knitr, Rdisop, rmarkdown, testthat, vdiffr

**VignetteBuilder** knitr

**Encoding** UTF-8

**LinkingTo** Rcpp

**Language** en-US

**RoxygenNote** 7.3.2

**LazyData** true

**Repository** https://janlisec.r-universe.dev

**Date/Publication** 2025-01-17 12:50:28 UTC

**RemoteUrl** https://github.com/janlisec/cormid

**RemoteRef** HEAD

**RemoteSha** 4dc1bb256e8f6b62740b7e1a4a51a67b93b7eb65

# Contents

CalcTheoreticalMDV . . . . . 2
CorMID . . . . . 3
CountChemicalElements . . . . . 5
getMID . . . . . 6
isotopes . . . . . 7
known_frags . . . . . 8
precalc_idx . . . . . 8
prep . . . . . 9
recMID . . . . . 9

**Index** **11**

---

`CalcTheoreticalMDV` *CalcTheoreticalMDV.*

---

## Description

`CalcTheoreticalMDV` will compute the Mass Distribution Vectors of isotopologues as it is used for correction matrix in CorMID computations.

## Usage

```
CalcTheoreticalMDV(
  fml = NULL,
  nbio = NULL,
  nmz = NULL,
  algo = c("CorMID", "Rdisop")
)
```

## Arguments

| | |
|---|---|
| `fml` | The chemical formula of the compound. |
| `nbio` | Provide the number of biological carbon within `fml` explicitly. |
| `nmz` | Provide the number of measured isotopes of `fml` explicitly. |
| `algo` | algo. |

## Details

`CalcTheoreticalMDV` basically is a convenience function using Rdisop to generate the isotopologue distribution at natural abundance of $^{13}C$ for a given formula. It will break this down into a matrix where the components of the MID constitute the rows and the expected relative ion intensities are within the columns. The number of exported ion intensities and MID components can be limited if numeric values for `nmz` and/or `nbio` are provided as parameters.

## Value

A matrix of theoretical mass distribution vectors.

## Examples

```
# standard distribution matrix
fml <- "C5H6Si1"
CalcTheoreticalMDV(fml = fml)

# extend to more columns (number of measured ions) if required
CalcTheoreticalMDV(fml = fml, nmz = 4)

# limit to a smaller number of biological carbon (i.e. if compounds are silylated)
CalcTheoreticalMDV(fml = fml, nmz = 4, nbio = 2)
#CalcTheoreticalMDV(fml = fml, nmz = 4, nbio = 2, algo="Rdisop")

# from Vignette
fml <- "C21Si5"
round(CalcTheoreticalMDV(fml = fml, nbio = 21, nmz = 21)[-(5:19), -(5:19)], 4)
```

---

CorMID *CorMID.*

---

## Description

CorMID will compute a MID (Mass Isotopologues Distribution) based on measured ion intensities in GC-APCI-MS.

## Usage

```
CorMID(
  int = NULL,
  fml = "",
  r = NULL,
  penalize = 7,
  mid_fix = NULL,
  trace_steps = FALSE,
  prec = 0.01,
  algo = c("CorMID", "Rdisop")
)

## S3 method for class 'CorMID'
plot(x, ...)

## S3 method for class 'CorMID'
print(x, ...)
```

## Arguments

| | |
|---|---|
| int | Named numeric vector of measured ion intensities of a fragment. Names will give position of values relative to M+H (see details). |
| fml | Chemical formula of the fragment as string. |

| | |
|---|---|
| r | Either a character vector giving fragments to be considered OR a named numeric giving relative amounts of fragments OR NULL (all known fragments will be estimated) OR a 2-row matrix giving the lower and upper allowed ratio (see examples). |
| penalize | Numeric exponent penalizing solutions with low M+H occurrence. Formula is 1+3*(1-x)^penalty. Set to NA to omit penalizing. |
| mid_fix | May provide a numeric vector used as a given MID. Allows to estimate *r* individually. |
| trace_steps | For testing purposes. Print the results of intermediate steps to console. |
| prec | Precision of the estimation of MID, set to 1% as default. |
| algo | The algorithm used to estimate the isotopic distribution of a chemical formula. |
| x | Object of class CorMID. |
| ... | Further plotting parameters. |

## Details

Let's assume we measured the ion intensities of all 3 isotopes of an individual compound containing 2 carbons and observe a vector of {978,22,0}. We may calculate the enrichment **E** out of this data, i.e. the relative proportion of $^{1}3^{C}$ vs total carbon which will amount to about 1.1% (the natural $^{1}3^{C}$ abundance) under standard conditions. The equivalent corMID vector would be {1,0,0}, indicating that the non-labeled isotopologue (where non-labeled means non-labeled above the natural 1.1%) is the only component observed. During a labeling experiment we may change the measurement values in different ways (either labeling only one carbon or both), which potentially can translate into similar values for **E** being larger 1.1%. The MIDs will provide additional information about the isotopologue fraction which gave rise to the observed **E**'s (cf. examples). The *r* parameter indicates an overlay of chemical rearrangements which may occur.

## Value

Estimated percent representation of each isotopologue measured (corMID).

## References

<doi:10.3390/metabo12050408>

## Examples

```
# make up some fake measurement data for Pyruvic acid 2TMS with 3 biological carbon
# assuming 10% labeling at M3 and 2 fragments
fml <- "C9H20O3Si2"
mid <- c(0.9, 0, 0, 0.1)
r <- unlist(list("M+H" = 0.8, "M+H2O-CH4" = 0.2))
int <- CorMID::recMID(mid = mid, r = r, fml = fml)
plot(int)

# full estimation of M and r
out <- CorMID::CorMID(int = int, fml = fml)
out
```

```
plot(out)

# get an improved result setting r to the correct values
CorMID::CorMID(int = int, fml = fml, r = r, prec = 0.0001)

# provoke a wrong estimation using a fixed r
CorMID::CorMID(int = int, fml = fml, r = unlist(list("M+H" = 1)))

# calculate r if you know the true corMID for a compound
r <- attr(CorMID::CorMID(int = int, fml = fml, mid_fix = c(0.9, 0, 0, 0.1)), "ratio")
round(CorMID::CorMID(int = int, fml = fml, r = r, prec = 0.0001), 3)

# deal with missing intensity values
CorMID::CorMID(int = int[-3], fml = fml)


# perform estimation with banded r and observation of optimization steps
r <- matrix(c(0.5, 1, 0, 0.5, 0, 0.5), nrow = 2, dimnames = list(NULL, c("M+H", "M+", "M+H2O-CH4")))
CorMID::CorMID(int = int, fml = fml, r = r, trace = TRUE)

# process Gln data from publication
utils::data("prep", package = "CorMID")
int <- prep[[24]][["int"]][, 6]
fml <- prep[[24]]$fml
CorMID::CorMID(int = int, fml = fml, trace = TRUE)

# check the effect of the penalize parameter on selection of adducts
int <- c(1560, 119203, 41927, 16932, 4438)
names(int) <- c(-2, 0, 1, 2, 3)
fml <- "C19H37NO4Si3"
CorMID::CorMID(int = int, fml = fml, r = NULL, trace = TRUE)
CorMID::CorMID(int = int, fml = fml, r = NULL, trace = TRUE, penalize = 7)
```

---

`CountChemicalElements` *CountChemicalElements.*

---

## Description

`CountChemicalElements` will split a character (chemical formula) into its elements and count their occurrence.

## Usage

```
CountChemicalElements(x = NULL, ele = NULL)
```

## Arguments

| | |
|---|---|
| `x` | Chemical formula. |
| `ele` | Character vector of elements to count particularly or counting all contained in string if NULL. |

## Details

No testing for any chemical alphabet is performed. Elements may occur several times and will be summed up in this case without a warning. Information within brackets, i.e. [13]C will be removed prior to counting together with other symbols not contained in the R set 'alnum'. The result is filtered and ordered according to parameter 'ele' if provided.

## Value

A named numeric with counts for all contained or specified elements.

## Examples

```
# count every element
CountChemicalElements("C3H7Cl")

# remove additional symbols and sum up redundant elements (here 'C')
CountChemicalElements("[13]CC2H8Cl+")

# count specific elements and return in specified order
CountChemicalElements("[13]CC2H8Cl+", ele=c("Cl","O","H"))

# apply on a vector of formulas using sapply
sapply(c("C3H7Cl", "[13]CC2H8Cl+"), CountChemicalElements, ele=c("Cl","O","H"))
```

---

`getMID` *getMID.*

---

## Description

`getMID` will determine the measurable isotopic spectrum for a chemical formula.

## Usage

```
getMID(
  fml,
  resolution = 20000,
  cutoff = 1e-04,
  isotopes = NULL,
  prec = 4,
  step = 0
)
```

## Arguments

| | |
|---|---|
| `fml` | Chemical formula. |
| `resolution` | Currently fixed to 20000 (might be made changeable in the future). |
| `cutoff` | Discard peaks below this threshold (relative to highest peak). |

| | |
|---|---|
| isotopes | Specify explicitly or keep NULL to use internally provided list. |
| prec | Rounding precision of returned mz and int values. |
| step | Can be used to return intermediate results (might be deprecated in the future). |

## Details

The computation yields similar results that would be obtained by packages 'Rdisop' or 'enviPat' but is completely in R (no C++ dependencies). However, it is approx. 7-fold slower than 'Rdisop'. Where processing speed is of importance, please use the 'algo' parameter of the 'CorMID' function.

## Value

A two column matrix for mz and int values of the calculated spectrum.

## Examples

```
fml <- "C3H7Cl1"
getMID(fml)
## Not run:
bench::mark(
 CorMID = dim(getMID(fml, prec=5)),
 Rdisop = dim(round(t(Rdisop::getMolecule(fml)$isotopes[[1]])[1:4,],5))
)

## End(Not run)

## Not run:
data(chemforms, package = "enviPat")
chemforms <- chemforms[-grep("[[]", chemforms)]
bench::mark(
 CorMID = length(lapply(chemforms, getMID)),
 Rdisop = length(lapply(chemforms, Rdisop::getMolecule))
)

## End(Not run)
```

---

# isotopes *A data frame containing isotope information.*

---

## Description

A data frame containing isotope information.

## Usage

```
isotopes
```

### Format

A data frame of 5 columns for 308 chemical isotopes.

**element** The element name.

**isotope** The isotope name.

**mass** The absolute mass of this isotope in Dalton.

**abundance** The absolute abundance of this isotope.

**ratioC** The C ratio of this element.

### Source

Imported from the enviPat package.

---

## known_frags — *The known fragments and their relative position to* [M+H].

---

### Description

The known fragments and their relative position to [M+H].

### Usage

```
known_frags
```

### Format

A named vector of the currently evaluated fragments.

### Source

Calculated using the internal CorMID function ‘get_idx_mat‘.

---

## precalc_idx — *Pre-calculated index matrices to speed up calculations in function* getMID.

---

### Description

Pre-calculated index matrices to speed up calculations in function getMID.

### Usage

```
precalc_idx
```

### Format

A list of 36 example metabolites.

### Source

Calculated using the internal CorMID function ‘get_idx_mat‘.

---

`prep` *Example data as used in function* `CorMID`.

---

### Description

Example data as used in function `CorMID`.

### Usage

```
data(prep)
```

### Format

A list of 36 example metabolites.

**name** The compound name.

**int** A numeric matrix providing peak intensities of isotopologues containing biological carbon (rows) over ten samples (columns).

**fml** A character vector of the chemical formula of the compound amended by attributes for the number of biological carbons ’nbio’ and the number of measured mass isotopologues ’nmz’.

### Source

A flux experiment on SW480/SW620 cell lines by Inna Zaimenko (IZ_Exp05).

---

`recMID` *recMID.*

---

### Description

`recMID` will reconstruct a measured GC-APCI-MS spectrum of a compound given its true MID and the fragment ratio.

## Usage

```
recMID(
  mid = c(1, 0, 0),
  r = list(`M+H` = 1),
  fml = NULL,
  cutoff = 0.001,
  algo = c("CorMID", "Rdisop")
)

## S3 method for class 'recMID'
plot(x, ...)
```

## Arguments

| | |
|---|---|
| mid | A numeric vector with sum=1 and length of C atoms +1. |
| r | Fragment ratios. A numeric vector with sum=1. |
| fml | A compound formula. |
| cutoff | Remove values below this threshold from output vector. |
| algo | The algorithm used to estimate the isotopic distribution of a chemical formula. |
| x | Object of class recMID. |
| ... | Further plotting parameters. |

## Details

recMID is basically the inverse function to CorMID. Providing a specific chemical formula together with information regarding the true MID and r, this function will compute a vector of ion intensities which can be expected in a GC-APCI-MS analysis for this compound.

## Value

A reconstructed MID, i.e. the relative intensity values that would be measured for the specified molecule ‘fml‘ with respect to parameters“mid‘ and ‘r‘. The default values for ‘mid‘ and ‘r‘ are set to reflect natural abundance and no fragmentation.

## Examples

```
fml <- "C9H20O3Si2"
mid <- c(0.9,0,0,0.1)
r <- list("M+H"=0.8, "M-H"=0.1, "M+H2O-CH4"=0.1)
(rMID <- CorMID::recMID(mid=mid, r=r, fml=fml))
plot(rMID)
plot(x = rMID, ylim=c(0,max(rMID)))
plot(x = rMID, xlim=c(-2,12), ylim=NULL, col=2, lwd=12, las=2, xlab="label")

CorMID::CorMID(int = rMID, fml=fml, prec=0.001, r=unlist(r), trace_steps = TRUE)
```

# Index

∗ **datasets**
isotopes, 7
known_frags, 8
precalc_idx, 8
prep, 9

CalcTheoreticalMDV, 2
CorMID, 2, 3
CountChemicalElements, 5

getMID, 6

isotopes, 7

known_frags, 8

plot.CorMID *(*CorMID*)*, 3
plot.recMID *(*recMID*)*, 9
precalc_idx, 8
prep, 9
print.CorMID *(*CorMID*)*, 3

recMID, 9