

Package ‘msdxr’

April 12, 2026

Type Package

Title Read GC-MS Data Files in the Chromatec MSDX Format

Version 0.1.0

Description Tools to read and parse GC-MS data stored in the proprietary MSDX binary file format used by Chromatec instruments. This package provides an independent parser and is not affiliated with, endorsed by, or associated with Chromatec.

License MIT + file LICENSE

URL <https://mass-spec.ru/projects/gcmsdata/msdxr/eng/>

BugReports <https://github.com/andreysamokhin/msdxr/issues>

Encoding UTF-8

LazyData true

NeedsCompilation yes

Suggests mssearchr,
testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

LinkingTo Rcpp

Imports Rcpp

SystemRequirements zlib

Author Andrey Samokhin [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0003-0223-6087>>),
Roman Borisov [dct]

Maintainer Andrey Samokhin <andrey.s.samokhin@gmail.com>

VignetteBuilder knitr

Contents

ExtractSpectrum	2
ExtractTic	3
ExtractXic	4
ReadMsdx	5
Index	7

ExtractSpectrum	<i>Extract a mass spectrum</i>
-----------------	--------------------------------

Description

Extract a single mass spectrum from an object returned by [ReadMsdx](#).

Usage

```
ExtractSpectrum(gcms, scan = NULL, rt = NULL)
```

Arguments

gcms	An object returned by ReadMsdx .
scan	An integer scalar defining the scan number.
rt	A numeric scalar defining the retention time in seconds. The closest scan is selected.

Value

A named list with elements:

- `rt` - retention time in seconds;
- `scan_no` - scan number;
- `mz` - mass-to-charge ratios;
- `intst` - intensities;
- `tic` - total ion current.

Examples

```
# Read an example MSDX file
msdx_file <- system.file("extdata", "ketones.msdx", package = "msdxc")
gcms <- ReadMsdx(msdx_file)

# Plot a mass spectrum at scan 32
ms <- ExtractSpectrum(gcms, scan = 32L)
old_par <- par(yaxs = "i")
plot(
  ms$mz,
  ms$intst,
  main = sprintf("rt = %.1fs; scan_no = %d", ms$rt, ms$scan_no),
  xlim = c(0, 200),
  type = "h",
  xlab = "m/z",
  ylab = "Intensity",
  bty = "n"
)
par(old_par)

# Plot a mass spectrum at retention time of 467.0 s
ms <- ExtractSpectrum(gcms, rt = 467.0)
old_par <- par(yaxs = "i")
```

```

plot(
  ms$ mz,
  ms$intst,
  main = sprintf("rt = %.1fs; scan_no = %d", ms$rt, ms$scan_no),
  xlim = c(0, 200),
  type = "h",
  xlab = "m/z",
  ylab = "Intensity",
  bty = "l"
)
par(old_par)

```

ExtractTic

Extract a total ion chromatogram

Description

Extract a total ion chromatogram from an object returned by [ReadMsdx](#).

Usage

```
ExtractTic(gcms, scan_range = NULL, rt_range = NULL)
```

Arguments

gcms	An object returned by ReadMsdx .
scan_range	An integer vector of length 2 defining the range of scan numbers.
rt_range	A numeric vector of length 2 defining the range of retention times in seconds.

Value

A data frame with columns:

- scan_no - scan number;
- rt - retention time in seconds;
- tic - total ion current.

Examples

```

# Read an example MSDX file
msdx_file <- system.file("extdata", "ketones.msdx", package = "msdxr")
gcms <- ReadMsdx(msdx_file)

# Plot the total ion chromatogram
tic <- ExtractTic(gcms, rt_range = c(420, 480))
plot(
  tic$rt,
  tic$tic,
  main = "Total ion chromatogram",
  type = "l",
  xlab = "Retention time, s",
  ylab = "Intensity",

```

```
    bty = "l"
  )
```

ExtractXic

Extract an extracted ion chromatogram

Description

Extract an extracted ion chromatogram (XIC) from an object returned by [ReadMsdx](#).

Usage

```
ExtractXic(gcms, mz, scan_range = NULL, rt_range = NULL, mz_delta = 0.5)
```

Arguments

gcms	An object returned by ReadMsdx .
mz	A numeric scalar defining the target mass-to-charge ratio.
scan_range	An integer vector of length 2 defining the range of scan numbers.
rt_range	A numeric vector of length 2 defining the range of retention times in seconds.
mz_delta	A numeric scalar defining the half-width of the extraction window around mz. Peaks with m/z values between $mz - mz_delta$ and $mz + mz_delta$ are summed.

Value

A data frame with columns:

- scan_no - scan number;
- rt - retention time in seconds;
- xic - extracted ion chromatogram intensity.

Examples

```
# Read an example MSDX file
msdx_file <- system.file("extdata", "ketones.msdx", package = "msdxr")
gcms <- ReadMsdx(msdx_file)

# Plot extracted ion chromatograms for m/z 85, 184, and 198
xic_85 <- ExtractXic(gcms, mz = 85, rt_range = c(420, 480))
xic_184 <- ExtractXic(gcms, mz = 184, rt_range = c(420, 480))
xic_198 <- ExtractXic(gcms, mz = 198, rt_range = c(420, 480))
plot(
  xic_85$rt,
  xic_85$xic,
  col = "gray",
  main = "Extracted ion chromatograms",
  type = "l",
  xlab = "Retention time, s",
  ylab = "Intensity",
  bty = "l"
)
```

```
lines(xic_184$rt, xic_184$xic, col = "blue3")
lines(xic_198$rt, xic_198$xic, col = "red3")
legend(
  "top",
  legend = paste0("m/z = ", c(85L, 184L, 198L)),
  col = c("gray", "blue3", "red3"),
  lty = 1
)
```

ReadMsdx*Read GC/MS data from an MSDX file*

Description

Read GC/MS data from an MSDX file and return it as a list of mass spectra, where each element corresponds to a single scan.

Each mass spectrum contains the retention time, scan number, mass-to-charge values, ion intensities, and total ion current.

Usage

```
ReadMsdx(file_path)
```

Arguments

file_path A character string specifying the path to the MSDX file.

Value

A list of mass spectra. Each element of the list is itself a named list with the following elements:

- **rt** - Retention time.
- **scan_no** - Scan number.
- **mz** - Numeric vector of mass-to-charge ratios.
- **intst** - Numeric vector of ion intensities.
- **tic** - Total ion current for the scan.

The **settings** attribute is a named list containing extracted mass spectrometer settings and scan parameters. Scan parameters are extracted only for GC/MS data containing a single segment with a single event.

Examples

```
# Read an example MSDX file
msdx_file <- system.file("extdata", "ketones.msdx", package = "msdxr")
gcms <- ReadMsdx(msdx_file)

# Plot a mass spectrum
ms <- gcms[[32L]]
old_par <- par(yaxs = "i")
plot(
```

```
ms$nz,  
ms$intst,  
main = sprintf("rt=%.1fs; scan_no=%d", ms$rt, ms$scan_no),  
xlim = c(0, 200),  
type = "h",  
xlab = "m/z",  
ylab = "Intensity",  
bty = "l"  
)  
par(old_par)  
  
# Show experimental settings  
settings <- attr(gcms, "settings")  
print(settings)  
#> $ion_source_temp  
#> [1] 200  
#>  
#> $transfer_line_temp  
#> [1] 250  
#>  
#> $operation_mode  
#> [1] "segmented"  
#>  
#> $n_segments  
#> [1] 1  
#>  
#> $scan_mode  
#> [1] "scan"  
#>  
#> $mass_range  
#> [1] 10 700  
#>  
#> $scan_time  
#> [1] 0.2
```

Index

ExtractSpectrum, [2](#)

ExtractTic, [3](#)

ExtractXic, [4](#)

ReadMsdx, [2–4](#), [5](#)