

Package ‘rawDiag’

May 2, 2026

Type Package

Title Brings Orbitrap Mass Spectrometry Data to Life; Fast and Colorful

Version 1.9.0

Depends R (>= 4.4)

Imports dplyr, ggplot2 (>= 3.4), grDevices, hexbin, htmltools, BiocManager, BiocParallel, rawrr (>= 1.15.5), rlang, reshape2, scales, shiny (>= 1.5), stats, utils

Suggests BiocStyle (>= 2.28), ExperimentHub, tartare, knitr, testthat

Description Optimizing methods for liquid chromatography coupled to mass spectrometry (LC-MS) poses a nontrivial challenge. The rawDiag package facilitates rational method optimization by generating MS operator-tailored diagnostic plots of scan-level metadata. The package is designed for use on the R shell or as a Shiny application on the Orbitrap instrument PC.

License GPL-3

URL <https://github.com/fgcz/rawDiag/>

BugReports <https://github.com/fgcz/rawDiag/issues>

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.3.2

VignetteBuilder knitr

biocViews MassSpectrometry, Proteomics, Metabolomics, Infrastructure, Software, ShinyApps

git_url <https://git.bioconductor.org/packages/rawDiag>

git_branch devel

git_last_commit d2d26ca

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-01

Author Christian Panse [aut, cre] (ORCID: <https://orcid.org/0000-0003-1975-3064>),
Christian Trachsel [aut],
Tobias Kockmann [aut]

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

Contents

rawDiag-package	2
.calculatioMasterScan	3
.cycleTime	3
.fillNAgaps	4
buildRawDiagShinyApp	4
checkRawrr	5
is.rawDiag	6
plotChargeState	6
plotCycleLoad	7
plotCycleTime	8
plotInjectionTime	8
plotLockMassCorrection	9
plotMassDistribution	10
plotMzDistribution	10
plotPrecursorHeatmap	11
plotScanTime	12
plotTicBasepeak	13
rawDiagServer	13
rawDiagUI	14
readRaw	14

Index	16
--------------	-----------

rawDiag-package	<i>rawDiag: Brings Orbitrap Mass Spectrometry Data to Life; Fast and Colorful</i>
-----------------	---

Description

Optimizing methods for liquid chromatography coupled to mass spectrometry (LC-MS) poses a nontrivial challenge. The rawDiag package facilitates rational method optimization by generating MS operator-tailored diagnostic plots of scan-level metadata. The package is designed for use on the R shell or as a Shiny application on the Orbitrap instrument PC.

Author(s)

Maintainer: Christian Panse <cp@fgcz.ethz.ch> ([ORCID](#))

Authors:

- Christian Trachsel <Christian.Trachsel@fgcz.ethz.ch>
- Tobias Kockmann <Tobias.Kockmann@fgcz.ethz.ch>

See Also

Useful links:

- <https://github.com/fgcz/rawDiag/>
- Report bugs at <https://github.com/fgcz/rawDiag/issues>

.calculatioMasterScan *Calculate Master Scan Number*

Description

calculates the MS1 master scan number of an MS2 scan and populates the MasterScanNumber with it

Usage

```
.calculatioMasterScan(x)
```

Arguments

x a data.frame object adhering to the specified criteria for the is.rawDiag function.

Value

a data.frame containing a MasterScanNumber column.

Author(s)

Christian Trachsel

.cycleTime *Calculate MS Cycle Time*

Description

calculates the lock mass deviations along RT.

Usage

```
.cycleTime(x)
```

Arguments

x a data.frame object adhering to the specified criteria for the is.rawDiag function.

Value

calculates the time of all ms cycles and the 95 the cycle time is defined as the time between two consecutive MS1 scans

Note

TODO: quantile part needed? If no MS1 scan is present? E.g., DIA take lowest window as cycle indicator?

Author(s)

Christian Trachsel (2017), Christian Panse (20231201) refactored

<code>.fillNAgaps</code>	<i>Fill NA values with last previous value</i>
--------------------------	--

Description

Fill NA values with last previous value

Usage

```
.fillNAgaps(x)
```

Arguments

`x` a vector of values

Value

a vector with any NA values replaced with the last previous actual value

Author(s)

Christian Trachsel

Examples

```
c(NA, 1, 2, 3, NA, 4, 5, NA, NA, NA, 6) |>
  rawDiag:::fillNAgaps()
```

<code>buildRawDiagShinyApp</code>	<i>Build the rawDiag shiny application</i>
-----------------------------------	--

Description

Build the rawDiag shiny application

Usage

```
buildRawDiagShinyApp(rawDir = (dirname(rawrr::sampleFilePath())))
```

Arguments

`rawDir` A directory containing the input raw files, default is set to the \$HOME/Downloads directory.

Value

returns the rawDiag shiny apps

Note

launch the shiny application by embracing your command line while expecting the raw file in \$HOME/Downloads

- MacOSX and Linux: `R -q -e "library(rawDiag); buildRawDiagShinyApp() |> shiny::runApp(launch.br`
- Microsoft Windows: `R.exe -e "library(rawDiag); buildRawDiagShinyApp() |> shiny::runApp(launch.b`

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |>
  dirname() |>
  rawDiag::buildRawDiagShinyApp() |>
  shiny::runApp()

# or use your 'Download' folder
(Sys.getenv('HOME') |>
  file.path("Downloads")) |>
  rawDiag::buildRawDiagShinyApp() |>
  shiny::runApp()
```

checkRawrr

Checks Bioconductor installation instructions

Description

Checks Bioconductor installation instructions

Usage

```
checkRawrr()
```

Value

TRUE if everything is installed correctly

`is.rawDiag`*Is an Object an rawDiag Object?*

Description

Is an Object an rawDiag Object?

Usage

```
is.rawDiag(object)
```

Arguments

`object` any R object.

Value

a boolean

Author(s)

Christian Panse 2018

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() |> rawDiag::is.rawDiag()
```

`plotChargeState`*Charge State Overview Plot*

Description

graphs the number of occurrences of all selected precursor charge states.

Usage

```
plotChargeState(x, method = "trellis")
```

Arguments

`x` a `data.frame` object adhering to the specified criteria for the `is.rawDiag` function.

`method` specifying the plot method 'trellis' | 'violin' | 'overlay'. The default is 'trellis'.

Value

a `ggplot` object.

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() -> S  
S |> rawDiag::plotLockMassCorrection()
```

plotCycleLoad

Cycle Load Plot

Description

plotting the number of MS2 per MS1 (the duty cycle) scan versus retention time. The deepskyblue colored loess curve shows the trend.

Usage

```
plotCycleLoad(x, method = "trellis")
```

Arguments

- | | |
|--------|--|
| x | a data.frame object adhering to the specified criteria for the is.rawDiag function. |
| method | specifying the plot method 'trellis' 'violin' 'overlay'. The default is 'trellis'. |

Value

a `ggplot` object.

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() -> S  
S |> rawDiag::plotCycleLoad()
```

plotCycleTime *Plot Cycle Time*

Description

graphs the time difference between two consecutive MS1 scans (cycle time) with respect to RT (scatter plots) or its density (violin). A smooth curve graphs the trend. The 95th percentile is indicated by a red dashed line.

Usage

```
plotCycleTime(x, method = "trellis")
```

Arguments

`x` a `data.frame` object adhering to the specified criteria for the `is.rawDiag` function.

`method` specifying the plot method 'trellis' | 'violin' | 'overlay'. The default is 'trellis'.

Value

a `ggplot` object.

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() |> rawDiag::plotCycleTime()
```

plotInjectionTime *Plot Injection Time*

Description

shows the injection time density of each mass spectrometry file as a violin plot. The higher the maximum number of MS2 scans is in the method, the more the density is shifted towards the maximum injection time value.

Usage

```
plotInjectionTime(x, method = "trellis")
```

Arguments

`x` a `data.frame` object adhering to the specified criteria for the `is.rawDiag` function.

`method` specifying the plot method 'trellis' | 'violin' | 'overlay'. The default is 'trellis'.

Value

a `ggplot` object.

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() |> rawDiag::plotInjectionTime()
```

plotLockMassCorrection

Lock Mass Correction Plot

Description

Lock Mass Correction Plot

Usage

```
plotLockMassCorrection(x, method = "trellis")
```

Arguments

- | | |
|--------|--|
| x | a data.frame object adhering to the specified criteria for the is.rawDiag function. |
| method | specifying the plot method 'trellis' 'violin' 'overlay'. The default is 'trellis'. |

Value

a `ggplot` object.

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |>  
rawDiag::readRaw() |>  
rawDiag::plotLockMassCorrection()
```

plotMassDistribution *Mass Distribution Plot*

Description

plots the mass frequency in dependency to the charge state

Usage

```
plotMassDistribution(x, method = "trellis")
```

Arguments

x	a data.frame object adhering to the specified criteria for the <code>is.rawDiag</code> function.
method	specifying the plot method 'trellis' 'violin' 'overlay'. The default is 'trellis'.

Details

displays charge state resolved frequency of precursor masses.

Value

a `ggplot` object.

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() |> rawDiag::plotMassDistribution('overlay')
```

plotMzDistribution *mZ Distribution Plot of Ms2 Scans*

Description

draws precursor mass vs retention time for each MS2 scan in the raw file.

Usage

```
plotMzDistribution(x, method = "trellis")
```

Arguments

- x a data.frame object adhering to the specified criteria for the `is.rawDiag` function.
- method specifying the plot method 'trellis' | 'violin' | 'overlay'. The default is 'trellis'.

Value

a `ggplot` object.

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() -> S
rawDiag::plotMzDistribution(S)
```

plotPrecursorHeatmap *Precursor Mass versus StartTime MS2 based hexagons*

Description

Precursor Mass versus StartTime MS2 based hexagons

Usage

```
plotPrecursorHeatmap(x, method = "overlay", bins = 80)
```

Arguments

- x a data.frame object adhering to the specified criteria for the `is.rawDiag` function.
- method specifying the plot method 'trellis' | 'violin' | 'overlay'. The default is 'trellis'.
- bins number of bins in both vertical and horizontal directions. default is 80.

Value

a `ggplot` object.

Note

TODO: define bin with dynamically as $h = 2 \times \text{IQR} \times n^{-1/3}$ or number of bins $(\text{max-min})/h$

Author(s)

Christian Trachsel (2017)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() |> rawDiag::plotPrecursorHeatmap()
```

plotScanTime

Scan Event Plot

Description

Plotting the elapsed scan time for each individual scan event.

Usage

```
plotScanTime(x, method = "trellis")
```

Arguments

- | | |
|--------|--|
| x | a data.frame object adhering to the specified criteria for the is.rawDiag function. |
| method | specifying the plot method 'trellis' 'violin' 'overlay'. The default is 'trellis'. |

Value

a `ggplot` object.

Author(s)

Christian Trachsel (2017), Christian Panse (2023)

References

- rawDiag: [doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173),
- rawrr: [doi:10.1021/acs.jproteome.0c00866](https://doi.org/10.1021/acs.jproteome.0c00866)

Examples

```
## for debugging bioconductor check
if (Sys.info()['sysname'] %in% c("Darwin", "Linux")) {Sys.which('mono')}
rawrr::sampleFilePath() |> rawDiag::readRaw() -> S

rawDiag::checkRawrr()

S |> rawDiag::plotScanTime()
```

plotTicBasepeak	<i>Total Ion Count and Base Peak Plot</i>
-----------------	---

Description

displays the Total Ion Count (TIC) and the Base Peak Chromatogram of a mass spectrometry measurement. Multiple files are handled by faceting based on rawfile name.

Usage

```
plotTicBasepeak(x, method = "trellis")
```

Arguments

x	a data.frame object adhering to the specified criteria for the <code>is.rawDiag</code> function.
method	specifying the plot method 'trellis' 'violin' 'overlay'. The default is 'trellis'.

Value

a ggplot2 object for graphing the TIC and the Base Peak chromatogram.

Author(s)

Christian Trachsel (2017), Christian Panse (20231130) refactored

Examples

```
rawrr::sampleFilePath() |> rawDiag::readRaw() |> rawDiag::plotTicBasepeak()
```

rawDiagServer	<i>rawDiag shiny module</i>
---------------	-----------------------------

Description

rawDiag shiny module

Usage

```
rawDiagServer(id, vals)
```

Arguments

id	An ID string that corresponds with the ID used to call the module's UI function.
vals	containing rawfile

Value

rawDiag shiny module server

Examples

```
shiny::shiny(rawDir = (rawrr::sampleFilePath() |> dirname()))
```

rawDiagUI

rawDiag shiny module UI

Description

rawDiag shiny module UI

Usage

```
rawDiagUI(id)
```

Arguments

`id` An ID string that corresponds with the ID used to call the module's UI function.

Value

a shiny UI module

rawDiag shiny module UI

Examples

```
rawDiag::shiny(rawDir = (rawrr::sampleFilePath() |> dirname()))
```

readRaw

Reads selected raw file trailer information for rawDiag plot functions

Description

implements a wrapper function using the rawrr methods [readIndex](#), [readTrailer](#), and [readChromatogram](#) to read proprietary mass spectrometer generated data using third-party libraries.

Usage

```
readRaw(
  rawfile,
  msgFUN = function(x) {
    message(x)
  }
)
```

Arguments

rawfile	the name of the raw file containing the mass spectrometry data from the Thermo Fisher Scientific instrument.
msgFUN	this function is used for logging information while composing the resulting data.frame. It can also be used for shiny progress bar. The default is using the message.

Value

a data.frame containing the selected trailer information.

Note

The set up procedure for the rawrr package needs to be run in order to use this package.

Author(s)

Christian Panse (2016-2023)

References

[doi:10.1021/acs.jproteome.8b00173](https://doi.org/10.1021/acs.jproteome.8b00173)

Examples

```
rawDiag::checkRawrr()  
rawrr::sampleFilePath() |>  
  rawDiag::readRaw()
```

Index

- * **internal**
 - rawDiag-package, [2](#)
 - .calculatioMasterScan, [3](#)
 - .cycleTime, [3](#)
 - .fillNAGaps, [4](#)
- buildRawDiagShinyApp, [4](#)
- checkRawrr, [5](#)
- ggplot, [6–12](#)
- is.rawDiag, [6](#)
- PlotChargeState (plotChargeState), [6](#)
- plotChargeState, [6](#)
- PlotCycleLoad (plotCycleLoad), [7](#)
- plotCycleLoad, [7](#)
- PlotCycleTime (plotCycleTime), [8](#)
- plotCycleTime, [8](#)
- PlotInjectionTime (plotInjectionTime), [8](#)
- plotInjectionTime, [8](#)
- PlotLockMassCorrection
 - (plotLockMassCorrection), [9](#)
- plotLockMassCorrection, [9](#)
- PlotMassDistribution
 - (plotMassDistribution), [10](#)
- plotMassDistribution, [10](#)
- PlotMzDistribution
 - (plotMzDistribution), [10](#)
- plotMzDistribution, [10](#)
- PlotPrecursorHeatmap
 - (plotPrecursorHeatmap), [11](#)
- plotPrecursorHeatmap, [11](#)
- PlotScanTime (plotScanTime), [12](#)
- plotScanTime, [12](#)
- PlotTicBasepeak (plotTicBasepeak), [13](#)
- plotTicBasepeak, [13](#)
- rawDiag (readRaw), [14](#)
- rawDiag-package, [2](#)
- rawDiagServer, [13](#)
- rawDiagUI, [14](#)
- readChromatogram, [14](#)
- readIndex, [14](#)
- readRaw, [14](#)
- readTrailer, [14](#)