

# Package ‘hdxmsqc’

December 20, 2024

**Type** Package

**Title** An R package for quality Control for hydrogen deuterium exchange mass spectrometry experiments

**Version** 1.3.0

**Description** The hdxmsqc package enables us to analyse and visualise the quality of HDX-MS experiments. Either as a final quality check before downstream analysis and publication or as part of a interactive procedure to determine the quality of the data. The package builds on the QFeatures and Spectra packages to integrate with other mass-spectrometry data.

**License** file LICENSE

**Encoding** UTF-8

**LazyData** false

**Depends** R(>= 4.3), QFeatures, S4Vectors, Spectra

**Imports** dplyr, tidyr, ggplot2, BiocStyle, knitr, methods, grDevices, stats, MsCoreUtils

**Suggests** RColorBrewer, pheatmap, MASS, patchwork, testthat

**VignetteBuilder** knitr

**Roxygen** list(markdown=TRUE)

**RoxygenNote** 7.2.3

**biocViews** QualityControl,DataImport, Proteomics, MassSpectrometry, Metabolomics

**BugReports** <https://github.com/ococrook/hdxmsqc/issues>

**URL** <http://github.com/ococrook/hdxmsqc>

**Language** en-US

**git\_url** <https://git.bioconductor.org/packages/hdxmsqc>

**git\_branch** devel

**git\_last\_commit** 75e2815

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-12-20

**Author** Oliver M. Crook [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-5669-8506>>)

**Maintainer** Oliver M. Crook <oliver.crook@stats.ox.ac.uk>

## Contents

BRD4df . . . . .	2
BRD4df_full . . . . .	3
chargeCorrelationHdx . . . . .	3
compatibleUptake . . . . .	4
computeMassError . . . . .	4
computeMonotoneStats . . . . .	5
exchangeableAmides . . . . .	6
fourierIsotope . . . . .	6
generateSpectra . . . . .	7
hdmsqc . . . . .	8
imTimeOutlier . . . . .	8
intensityOutliers . . . . .	9
isMissingAtRandom . . . . .	10
isotopicDistributionHDXfourier . . . . .	10
plotImTimeOutlier . . . . .	11
plotIntensityOutliers . . . . .	12
plotMassError . . . . .	13
plotMissing . . . . .	13
plotMonotoneStat . . . . .	14
plotrTimeOutliers . . . . .	15
processHDE . . . . .	16
qualityControl . . . . .	16
replicateCorrelation . . . . .	18
replicateOutlier . . . . .	19
rTimeOutliers . . . . .	20
spectraSimilarity . . . . .	21
<b>Index</b>	<b>23</b>

---

BRD4df

*This is data to be included in my package*

---

## Description

A small HDX-MS dataset for BRD4 in apo state and in complex with IBET151

## Author(s)

My Name <ococrook@gmail.com>

---

`BRD4df_full`*This is data to be included in my package*

---

**Description**

A complete HDX-MS dataset for BRD4 in apo state and in complex with IBET151

**Author(s)**

My Name <ococrook@gmail.com>

---

`chargeCorrelationHdx`*Charge states should have correlated incorporation but they need not be exactly the same*

---

**Description**

Charge states should have correlated incorporation but they need not be exactly the same

**Usage**

```
chargeCorrelationHdx(object, experiment = NULL, timepoints = NULL)
```

**Arguments**

<code>object</code>	An object of class <code>QFeatures</code>
<code>experiment</code>	A character vector indicating the experimental conditions
<code>timepoints</code>	A numeric vector indicating the experimental timepoints

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df_full")
BRD4df_filtered <- isMissingAtRandom(object = BRD4df_full)
BRD4df_full_imputed <- impute(BRD4df_filtered, method = "zero", i = 1)
experiment <- c("wt", "iBET")
timepoints <- rep(c(0, 15, 60, 600, 3600, 14000), each = 3)
monoStat <- chargeCorrelationHdx(object = BRD4df_full_imputed,
  experiment = experiment,
  timepoints = timepoints)
```

---

compatibleUptake	<i>Check whether deuterium uptakes are compatible with difference overlapping sequences.</i>
------------------	--

---

**Description**

Check whether deuterium uptakes are compatible with difference overlapping sequences.

**Usage**

```
compatibleUptake(object, overlap = 5, experiment = NULL, timepoints = NULL)
```

**Arguments**

object	An object of class QFeatures
overlap	How much overlap is required to check consistency. Default is sequences within 5 residues
experiment	A character vector indicating the experimental conditions
timepoints	A numeric vector indicating the experimental timepoints

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df")
result <- compatibleUptake(BRD4df, experiment = 1, timepoints = 1)
```

---

computeMassError	<i>Empirical versus theoretical mass errors</i>
------------------	---

---

**Description**

Empirical versus theoretical mass errors

**Usage**

```
computeMassError(object, eCentroid = "Exp.Cent", tCentroid = "Theor.Cent")
```

**Arguments**

object	An object of class QFeatures
eCentroid	character string indicating column identifier for experimental centroid
tCentroid	character string indicating column identifier for theoretical centroid

**Value**

The error difference between the empirical and theoretical centroid

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df")
result <- computeMassError(BRD4df, "Exp.Cent", "Theor.Cent")
head(result)
```

---

`computeMonotoneStats` *Monotonicity based outlier detection.*

---

**Description**

Monotonicity based outlier detection.

**Usage**

```
computeMonotoneStats(object, experiment = NULL, timepoints = NULL)
```

**Arguments**

<code>object</code>	An object of class <code>QFeatures</code>
<code>experiment</code>	A character vector indicating the experimental conditions
<code>timepoints</code>	A numeric vector indicating the experimental timepoints

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df")
result <- computeMonotoneStats(BRD4df, experiment = 1, timepoint = 1)
```

---

exchangeableAmides      *Compute exchangeable amides.*

---

**Description**

Computes the number of exchangeable amides based on the sequence

**Usage**

```
exchangeableAmides(sequence)
```

**Arguments**

sequence      The sequence of the peptide

**Value**

Returns a numeric indicating the number of exchangeable amides

**Examples**

```
exchangeableAmides(sequence = "HDAEHAHEAPRKL")
```

---

fourierIsotope      *fourier transform approach to computing isotopic distribution*

---

**Description**

fourier transform approach to computing isotopic distribution

**Usage**

```
fourierIsotope(
  elements,
  incorp = 0,
  num_exch_sites = 0,
  charge = 1,
  isotopes = NULL
)
```

**Arguments**

elements      A list of elements  
 incorp      The deuterium incorporation  
 num\_exch\_sites      The number of exchangeable amides. Default is 0.  
 charge      The charge state of the peptide  
 isotopes      The number of isotopes to compute. The default is NULL, in which a default heuristic is used to make a good guess that covers the expected peaks.

**Value**

A list of mass and intensity value corresponding to the isotope distribution

**Author(s)**

Oliver Crook

**Examples**

```
fourierIsotope(c(C = 0, H = 2, N = 0, O = 1, S = 0, P = 0))
```

---

`generateSpectra`      *generate Spectra using a fourier transform*

---

**Description**

generate Spectra using a fourier transform

**Usage**

```
generateSpectra(  
  sequences,  
  incorps,  
  charges,  
  customs = list(code = NULL, elements = NULL)  
)
```

**Arguments**

sequences	A vector of peptide sequences
incorps	A vector of deuterium incorporation
charges	A vector of charge states of the peptide
customs	Custom elements supplied as a list

**Value**

A Spectra object corresponding to the isotope distributions

**Author(s)**

Oliver Crook

**Examples**

```
generateSpectra(sequence = "HDAEHAHEAPRKL", incorps = c(0.5), charges = 2)
```

---

hdxmsqc	<i>A package to perform quality control for mass-spectrometry based hydrogen deuterium exchange experiment.</i>
---------	---

---

**Description**

'hdxmsqc' provides the functionality to assess the quality and perform quality control of HDX-MS experiments. Raw and processed data can be visualized and analyzed to identify potential issues with the data. The package is designed to work with data from any HDX-MS platform. Typically, users will have exported results from either HDExaminer or DynamX software. There is not need to filter the data in either of those software systems.

**Author(s)**

Oliver Crook

---

imTimeOutlier	<i>Ion Mobility time based outlier analysis</i>
---------------	---

---

**Description**

Ion Mobility time based outlier analysis

**Usage**

```
imTimeOutlier(
  object,
  rightIMS = "rightIMS",
  leftIMS = "leftIMS",
  searchIMS = "Search.IMS"
)
```

**Arguments**

object	An object of class QFeatures
rightIMS	A string indicating the right boundary of the ion mobility separation time. Defaults is "rightIMS".
leftIMS	A string indicating the left boundary of the ion mobility separation time. Default is "leftIMS".
searchIMS	A string indicating the actual ion mobility search time. The default is "Search.IMS"

**Author(s)**

Oliver Crook



**Examples**

```
data("BRD4df_full")
BRD4df_filtered <- isMissingAtRandom(object = BRD4df_full)
BRD4df_full_imputed <- impute(BRD4df_filtered, method = "zero", i = 1)
imTimeOutlier(object = BRD4df_full_imputed)
```

---

intensityOutliers      *Intensity based deviations*

---

**Description**

Intensity based deviations

**Usage**

```
intensityOutliers(object, fcolIntensity = "Max.Inty")
```

**Arguments**

object	An object of class QFeatures
fcolIntensity	character to intensity intensity columns. Default is "Max.Inty" and uses regular expressions to find relevant columns

**Value**

The Cook's distance to characterise outliers

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df_full")

intensityOutliers(BRD4df_full)
```

---

isMissingAtRandom      *Missing at random versus missing not at random*

---

**Description**

Missing at random versus missing not at random

**Usage**

```
isMissingAtRandom(object, threshold = NULL, filter = TRUE)
```

**Arguments**

object	An object of class QFeatures
threshold	A threshold indicated how many missing values indicate whether missingness is not at random. Default is NULL, which means leads to a threshold which is half the number of columns.
filter	A logical indicating whether to filter out data that is deemed missing not at random <pre>data("BRD4df_full") isMissingAtRandom(BRD4df_full)</pre>

**Value**

Adds a missing not at random indicator column

**Author(s)**

Oliver Crook

---

isotopicDistributionHDXfourier  
*fourier transform approach to computing isotopic distribution*

---

**Description**

fourier transform approach to computing isotopic distribution

**Usage**

```
isotopicDistributionHDXfourier(
  sequence,
  incorp = 0,
  charge = 1,
  custom = list(code = NULL, elements = NULL)
)
```

**Arguments**

sequence	A peptide
incorp	The deuterium incorporation
charge	The charge state of the peptide
custom	custom amino acids can be provided here provide a list of the elements.

**Value**

A list of mass and intensity value corresponding to the isotope distribution

**Author(s)**

Oliver Crook

**Examples**

```
isotopicDistributionHDXfourier(sequence = "HDAEHAHEAPRKL")
```

---

plotImTimeOutlier      *Ion Mobility time based outlier analysis*

---

**Description**

Ion Mobility time based outlier analysis

**Usage**

```
plotImTimeOutlier(
  object,
  rightIMS = "rightIMS",
  leftIMS = "leftIMS",
  searchIMS = "Search.IMS"
)
```

**Arguments**

object	An object of class QFeatures
rightIMS	A string indicating the right boundary of the ion mobility separation time. Defaults is "rightIMS".
leftIMS	A string indicating the left boundary of the ion mobility separation time. Default is "leftIMS".
searchIMS	A string indicating the actual ion mobility search time. The default is "Search.IMS"

**Author(s)**

Oliver Crook

## Examples

```
library(RColorBrewer)
data("BRD4df_full")
BRD4df_filtered <- isMissingAtRandom(object = BRD4df_full)
BRD4df_full_imputed <- impute(BRD4df_filtered, method = "zero", i = 1)
plotImTimeOutlier(object = BRD4df_full_imputed)
```

---

plotIntensityOutliers *Intensity based deviation plot*

---

## Description

Intensity based deviation plot

## Usage

```
plotIntensityOutliers(object, fcolIntensity = "Max.Inty")
```

## Arguments

object	An object of class QFeatures
fcolIntensity	character to intensity intensity columns. Default is "Max.Inty" and uses regular expressions to find relevant columns

## Value

A ggplot2 object showing intensity based outliers

## Author(s)

Oliver Crook

## Examples

```
data("BRD4df_full")
library(RColorBrewer)

plotIntensityOutliers(BRD4df_full)
```

---

plotMassError	<i>Mass error plot</i>
---------------	------------------------

---

**Description**

Mass error plot

**Usage**

```
plotMassError(object, eCentroid = "Exp.Cent", tCentroid = "Theor.Cent")
```

**Arguments**

object	An object of class QFeatures
eCentroid	character string indicating column identifier for experimental centroid
tCentroid	character string indicating column identifier for theoretical centroid

**Value**

a ggplot2 object which can be used to visualise the

**Author(s)**

Oliver Crook

**Examples**

```
library(RColorBrewer)
data("BRD4df")
result <- plotMassError(BRD4df, "Exp.Cent", "Theor.Cent")
```

---

plotMissing	<i>missing value plot</i>
-------------	---------------------------

---

**Description**

missing value plot

**Usage**

```
plotMissing(object, ...)
```

**Arguments**

object	An object of class QFeatures
...	Additional arguemnts to pheatmap

**Value**

a pheatmap showing missing values

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df_full")
library(pheatmap)
library(RColorBrewer)

plotMissing(BRD4df_full)
```

---

plotMonotoneStat      *Monotonicity based outlier detection, plot.*

---

**Description**

Monotonicity based outlier detection, plot.

**Usage**

```
plotMonotoneStat(object, experiment = NULL, timepoints = NULL)
```

**Arguments**

object	An object of class QFeatures
experiment	A character vector indicating the experimental conditions
timepoints	A numeric vector indicating the experimental timepoints

**Author(s)**

Oliver Crook

**Examples**

```
library("RColorBrewer")
data("BRD4df_full")
experiment <- c("wt", "iBET")
timepoints <- rep(c(0, 15, 60, 600, 3600, 14000), each = 3)
monoStat <- computeMonotoneStats(object = BRD4df_full,
  experiment = experiment,
  timepoints = timepoints)
```

---

plotrTimeOutliers      *Retention time based analysis*

---

## Description

Retention time based analysis

## Usage

```
plotrTimeOutliers(  
  object,  
  leftRT = "leftRT",  
  rightRT = "rightRT",  
  searchRT = "Search.RT"  
)
```

## Arguments

object	An object of class QFeatures
leftRT	A character indicated pattern associated with left boundary of retention time search. Default is "leftRT".
rightRT	A character indicated pattern associated with right boundary of retention time search. Default is "rightRT".
searchRT	The actual search retention time pattern. Default is "Search.RT"

## Value

a ggplot2 object showing distribution of retention time windows.

## Author(s)

Oliver Crook

## Examples

```
data("BRD4df_full")  
library(RColorBrewer)  
  
plotrTimeOutliers(BRD4df_full)
```

---

processHDE	<i>Function to curate and HDExaminer file so that in contains all the information in a sensible format. This object can then be straightforwardly passed to a object of class QFeatures</i>
------------	---

---

**Description**

Function to curate and HDExaminer file so that in contains all the information in a sensible format. This object can then be straightforwardly passed to a object of class QFeatures

**Usage**

```
processHDE(HDExaminerFile, proteinStates = NULL)
```

**Arguments**

HDExaminerFile an object of class data.frame containing an HDExaminer data  
proteinStates a character vector indicating the protein states

**Value**

A wide format data frame with HDExaminer data

**Author(s)**

Oliver Crook

**Examples**

```
sample_data <- data.frame(read.csv(system.file("extdata", "ELN55049_AllResultsTables_Uncurated.csv", package = "
processHDE(sample_data)
```

---

qualityControl	<i>Quality Control table function. Generate a table that collates quality control metrics</i>
----------------	---

---

**Description**

Quality Control table function. Generate a table that collates quality control metrics



**Usage**

```
qualityControl(
  object,
  massError = NULL,
  intensityOutlier = NULL,
  retentionOutlier = NULL,
  monotonicityStat = NULL,
  mobilityOutlier = NULL,
  chargeCorrelation = NULL,
  replicateCorrelation = NULL,
  replicateOutlier = NULL,
  sequenceCheck = NULL,
  spectraCheck = NULL,
  experiment = NULL,
  timepoints = NULL,
  undeuterated = FALSE
)
```

**Arguments**

object	An object of class Qfeatures, with the data used for the analysis
massError	The output of the computeMassError function
intensityOutlier	The output of the intensityOutliers function
retentionOutlier	The output of the rTimeOutliers function
monotonicityStat	The output of the computeMonotoneStats function
mobilityOutlier	The output of the imTimeOutliers function
chargeCorrelation	The output of the chargeCorrelationsHdx function
replicateCorrelation	The output of the replicateCorrelation function
replicateOutlier	The output of the replicateOutlier function
sequenceCheck	The output of the compatibleUptake function
spectraCheck	The output of the spectraSimilarity function
experiment	The experimental conditions.
timepoints	The timepoints used in the analysis, must include repeat for replicates
undeuterated	A logical indicating whether only the undeuterated data should be exported

**Value**

An object of class DataFrame containing a summary of the quality control results.

**Author(s)**

Oliver Crook

---

replicateCorrelation *Correlation based checks*

---

**Description**

Correlation based checks

**Usage**

```
replicateCorrelation(object, experiment, timepoints)
```

**Arguments**

object	An object of class QFeatures.
experiment	A character vector indicating the experimental conditions
timepoints	A numeric vector indicating the experimental timepoints

**Value**

Returns A list of the same length as the number of experiments indicating outlier from correlation analysis. Outliers are flagged if their deuterium uptake is highly variable.

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df_full")
experiment <- c("wt", "iBET")
timepoints <- rep(c(0, 15, 60, 600, 3600, 14000), each = 3)
monoStat <- replicateCorrelation(object = BRD4df_full,
  experiment = experiment,
  timepoints = timepoints)
```

---

replicateOutlier	<i>Correlation based checks</i>
------------------	---------------------------------

---

**Description**

Correlation based checks

**Usage**

```
replicateOutlier(object, experiment, timepoints)
```

**Arguments**

object	An object of class QFeatures.
experiment	A character vector indicating the experimental conditions
timepoints	A numeric vector indicating the experimental timepoints

**Value**

Returns A list of the same length as the number of experiments indicating outlier from correlation analysis. Outliers are flagged if their deuterium uptake is highly variable.

**Author(s)**

Oliver Crook

**Examples**

```
data("BRD4df_full")
BRD4df_filtered <- isMissingAtRandom(object = BRD4df_full)
BRD4df_full_imputed <- impute(BRD4df_filtered, method = "zero", i = 1)
experiment <- c("wt", "iBET")
timepoints <- rep(c(0, 15, 60, 600, 3600, 14000), each = 3)
monoStat <- replicateOutlier(object = BRD4df_full_imputed,
  experiment = experiment,
  timepoints = timepoints)
```

---

rTimeOutliers      *Retention time based analysis*

---

### Description

Retention time based analysis

### Usage

```
rTimeOutliers(  
  object,  
  leftRT = "leftRT",  
  rightRT = "rightRT",  
  searchRT = "Search.RT"  
)
```

### Arguments

object	An object of class QFeatures
leftRT	A character indicated pattern associated with left boundary of retention time search. Default is "leftRT".
rightRT	A character indicated pattern associated with right boundary of retention time search. Default is "rightRT".
searchRT	The actual search retention time pattern. Default is "Search.RT"

### Value

A list indicating the retention time based outliers.

### Author(s)

Oliver Crook

### Examples

```
data("BRD4df_full")  
  
rTimeOutliers(BRD4df_full)
```

---

spectraSimilarity      *Spectral checking using data from HDsite*

---

### Description

Spectral checking using data from HDsite

### Usage

```
spectraSimilarity(
  peaks,
  object,
  experiment = NULL,
  mzCol = 14,
  startRT = "Start.RT",
  endRT = "End.RT",
  charge = "z",
  incorpD = "X.D.left",
  maxD = "maxD",
  numSpectra = NULL,
  ppm = 300,
  BPPARAM = bpparam()
)
```

### Arguments

peaks	a data.frame containing data exported from hdsite
object	a data.frame obtained from HDexaminer data
experiment	A character vector indicating the experimental conditions
mzCol	The column in the peak information indicating the base mz value
startRT	The column indicatng the start of the retention time. Default is "Start.RT"
endRT	The column indicating the end of the retention time. Default is "End.RT"
charge	The column indicating the charge information. Default is "z".
incorpD	The deuterium uptake value column. Default is "X.D.left".
maxD	The maximum allowed deuterium incorporation column. Default is "maxD".
numSpectra	The number of spectra to analyse. Default is NULL in which all Spectra are analysed.
ppm	The ppm error
BPPARAM	Bioconductor parallel options.

### Value

Two list of spectra observed and matching theoretical Spectra

**Author(s)**

Oliver Crook

# Index

## \* data

- BRD4df, [2](#)
- BRD4df\_full, [3](#)

BRD4df, [2](#)  
BRD4df\_full, [3](#)

chargeCorrelationHdx, [3](#)  
compatibleUptake, [4](#)  
computeMassError, [4](#)  
computeMonotoneStats, [5](#)

exchangeableAmides, [6](#)

fourierIsotope, [6](#)

generateSpectra, [7](#)

hdxmsqc, [8](#)

imTimeOutlier, [8](#)  
intensityOutliers, [9](#)  
isMissingAtRandom, [10](#)  
isotopicDistributionHDXfourier, [10](#)

plotImTimeOutlier, [11](#)  
plotIntensityOutliers, [12](#)  
plotMassError, [13](#)  
plotMissing, [13](#)  
plotMonotoneStat, [14](#)  
plotrTimeOutliers, [15](#)  
processHDE, [16](#)

qualityControl, [16](#)

replicateCorrelation, [18](#)  
replicateOutlier, [19](#)  
rTimeOutliers, [20](#)

spectraSimilarity, [21](#)