Package ‘tidyDenovix’

May 8, 2026

Title Cleans Spectrophotometry Data Obtained from the Denovix DS-11 Instrument

Version 2.1.0

Description
Cleans spectrophotometry data obtained from the Denovix instrument. The package also provides an option to normalize the data in order to compare the quality of the samples obtained.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, learnr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports data.table, janitor, readxl, tibble, dplyr, tidyr, stats, utils

Depends R (>= 2.10)

LazyData true

URL `https://github.com/AlphaPrime7/tidyDenovix`

BugReports `https://github.com/AlphaPrime7/tidyDenovix/issues`

NeedsCompilation no

Author Tingwei Adeck [aut, cre] (ORCID: <`https://orcid.org/0009-0009-7450-8863`>)

Maintainer Tingwei Adeck <`awesome.tingwei@outlook.com`>

Repository CRAN

Date/Publication 2024-01-18 10:00:02 UTC

Contents

extract_col_names 2
extract_sample_names 3
extract_wavelength 3

file_ext 4
lambda_check 5
lambda_check_source 6
make_wavelength 7
min_max_norm 7
qc_attributes 8
read_denovix 9
read_denovix_data 10
REnum 10
rna1 11
rna2 11
rna3 12
spec 12
tidyDenovix 12

Index **14**

`extract_col_names` *Title: Extract key colnames from the Denovix data frame*

Description

Title: Extract key colnames from the Denovix data frame

Usage

```
extract_col_names(xdf)
```

Arguments

`xdf`	The data frame for colname(s) extraction.

Value

A vector of key column names.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = read_denovix_data(fpath, file_type = 'csv')
col_names = extract_col_names(rna_data)
```

`extract_sample_names` *Title: Extract sample names from the Denovix data frame*

Description

Title: Extract sample names from the Denovix data frame

Usage

```
extract_sample_names(dfile, file_type = NULL)
```

Arguments

`dfile`	The denovix raw file for sample name(s) extraction.
`file_type`	The type of file.

Value

A vector of sample names.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
esn = extract_sample_names(fpath, file_type = 'csv')
```

`extract_wavelength` *Title: Extract wavelength*

Description

Title: Extract wavelength

Usage

```
extract_wavelength(xdf)
```

Arguments

`xdf`	The original data frame derived from importing Denovix data.

Value

A numeric data frame for the wavelength attribute.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = read_denovix_data(fpath, file_type = 'csv')
wl = extract_wavelength(rna_data)
```

`file_ext` *Title: File Extension Finder*

Description

Title: File Extension Finder

Usage

```
file_ext(epath)
```

Arguments

`epath`	File path.

Value

A string representing the file extension.

Author(s)

Unknown (Adapted by Tingwei Adeck)

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
ext = file_ext(fpath)
```

`lambda_check` *Title: Wavelength quality control*

Description

Title: Wavelength quality control

Usage

```
lambda_check(
  qdf,
  sample_type = c("RNA", "DNA"),
  check_level = c("strict", "lax")
)
```

Arguments

`qdf`	A data frame with quality attributes.
`sample_type`	The type of sample under investigation.
`check_level`	The level of strictness based on sample type.

Value

A data frame that meets the quality check criteria.

Note

Some key assumptions are made about quality for RNA or DNA. At the moment column names is the main issue found with using this approach.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = read_denovix_data(fpath, file_type = 'csv')
qc_check = lambda_check(rna_data,sample_type='RNA',check_level='lax')
```

`lambda_check_source` *Title: Wavelength quality control*

Description

Title: Wavelength quality control

Usage

```
lambda_check_source(
  odf,
  sample_type = c("RNA", "DNA"),
  check_level = c("strict", "lax")
)
```

Arguments

`odf`	A data frame with quality attributes.
`sample_type`	The type of sample under investigation.
`check_level`	The level of strictness based on sample type.

Value

A vector of sample names for the different QC criteria.

Note

Some key assumptions are made about quality for RNA or DNA. At the moment column names is the main issue found with using this approach.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = read_denovix_data(fpath, file_type = 'csv')
qc_check = lambda_check_source(rna_data,sample_type='RNA',check_level='lax')
```

`make_wavelength` *Title: Make wavelength*

Description

Title: Make wavelength

Usage

```
make_wavelength()
```

Value

A numeric data frame for the wavelength attribute.

Author(s)

Tingwei Adeck

Examples

```
wl = make_wavelength()
```

`min_max_norm` *Title: Min-Max normalization of attributes that require normalization*

Description

Title: Min-Max normalization of attributes that require normalization

Usage

```
min_max_norm(x)
```

Arguments

`x`	A single value from an attribute passed in the function for normalization.

Value

A normalized value (value between 1 and 0)

Note

lapply is needed to apply the function across several columns in a data set.

Author(s)

Tingwei Adeck (Adapted from Statology)

References

https://www.statology.org/how-to-normalize-data-in-r/

Examples

```
test_df <- as.data.frame(c(seq(40)))
colnames(test_df) <- "test"
test_df_norm <- lapply(test_df[1:ncol(test_df)], min_max_norm)
```

`qc_attributes` *Title: Quality Control data frame*

Description

Title: Quality Control data frame

Usage

```
qc_attributes(dfile, file_type = NULL, xdf)
```

Arguments

`dfile`	The Denovix file path.
`file_type`	The type of file.
`xdf`	The Denovix data frame.

Value

A quality control data frame.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = read_denovix_data(fpath, file_type = 'csv')
qc_attributes = qc_attributes(fpath, file_type = 'csv', rna_data)
```

`read_denovix` *Title: Read Denovix files*

Description

A function read Denovix data files.

Usage

```
read_denovix(dfile)
```

Arguments

`dfile`	A Denovix file or path to the Denovix file.

Value

A data frame.

Note

Denovix files can be saved as csv, txt or even excel files. This function accounts for these file types.

Author(s)

Tingwei Adeck

See Also

`read_denovix_data()`

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = read_denovix(fpath)
```

`read_denovix_data` *Title: Read Denovix files*

Description

A function read Denovix data files.

Usage

```
read_denovix_data(dfile, file_type = c("csv", "txt", "excel"))
```

Arguments

`dfile`	A Denovix file or path to the Denovix file.
`file_type`	The file type being imported.

Value

A data frame.

Note

Denovix files can be saved as csv, txt or even excel files. This function accounts for these file types.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = read_denovix_data(fpath, file_type = 'csv')
```

`REnum` *Title: Enum function*

Description

Title: Enum function

Usage

```
REnum(elist)
```

Arguments

`elist`	A list with variable binding.

Value

The list supplied by the user.

Author(s)

Tingwei Adeck

Examples

```
cols = list('x260' = 1, 'x280' = 2, 'x230' = 3)
qc = REnum(cols)
a = (qc$x260)
```

`rna1` *rna1.*

Description

An excel file version of the Denovix Spectrophotometry data.

Usage

```
rna1
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 14 rows and 151 columns.

`rna2` *rna2.*

Description

A csv file version of the Denovix Spectrophotometry data.

Usage

```
rna2
```

Format

An object of class `data.frame` with 15 rows and 151 columns.

`rna3` *rna3.*

Description

A csv file version of the Denovix Spectrophotometry data.

Usage

```
rna3
```

Format

An object of class `data.frame` with 15 rows and 152 columns.

`spec` *spec.*

Description

A txt file version of the Denovix Spectrophotometry data.

Usage

```
spec
```

Format

An object of class `data.frame` with 15 rows and 151 columns.

`tidyDenovix` *Title: Clean data from the Denovix DS-11 instrument*

Description

Title: Clean data from the Denovix DS-11 instrument

Usage

```
tidyDenovix(
  dfile,
  file_type = NULL,
  sample_type = c("RNA", "DNA"),
  check_level = c("strict", "lax"),
  qc_omit = NULL,
  normalized = c("yes", "no"),
  fun = NA
)
```

Arguments

`dfile`	The raw file obtained from the machine.
`file_type`	The file type specification.
`sample_type`	The sample type specification used in quality control.
`check_level`	The level of quality control performed.
`qc_omit`	Takes 'yes' or 'no' and determines if the qc data would be provided.
`normalized`	Takes 'yes' or 'no'.
`fun`	A parameter used for boolean expressions.

Value

A cleaned data frame with attribute names in some instances.

Note

The strict level of QC yields a data frame with no attribute names. This will be worked on so that users get the sample names for their data.

Author(s)

Tingwei Adeck

Examples

```
fpath <- system.file("extdata", "rnaspec2018.csv", package = "tidyDenovix", mustWork = TRUE)
rna_data = tidyDenovix(fpath, file_type = 'csv', sample_type = 'RNA', check_level = 'lax')
```

Index

∗ **datasets**
    rna1, 11
    rna2, 11
    rna3, 12
    spec, 12

extract_col_names, 2
extract_sample_names, 3
extract_wavelength, 3

file_ext, 4

lambda_check, 5
lambda_check_source, 6

make_wavelength, 7
min_max_norm, 7

qc_attributes, 8

read_denovix, 9
read_denovix_data, 10
read_denovix_data(), *9*
REnum, 10
rna1, 11
rna2, 11
rna3, 12

spec, 12

tidyDenovix, 12