Package 'epror'

July 25, 2025

Encoding UTF-8

Type Package

Title Transcript quantification by express profiling

Version 1.1

Date 2025-07-25

Author Markus Haas

Maintainer <markus.haas@ovgu.de>

Description
This package is intended to calculate transcript amounts with the express profiling method.

License GPL-3

RoxygenNote 7.3.2

Contents

epr.calcall 2
epr.calcamount 3
epr.checkfrag 3
epr.checksizestd 4
epr.makestd 5
epr.plotallcv 6
epr.plotallfrag 6
epr.plotallicv 7
epr.plotallraw 7
epr.plotcv 8
epr.plotfrag 8
epr.ploticv 9
epr.plotraw 10
epr.plotstdcurve 10
epr.readfragfile 11
epr.readinjection 11
epr.readrawdata 12
epr.readsettings 12
epr.readtranscripts 13

Index 14

`epr.calcall` *This function calculates transcript amounts of all files in folder*

Description

This function calculates the amount of a given set of transcripts based on so called standard values (see epr.makestd for details). All files which match the pattern given with fragfileext in the working directory will be analyzed. This function is a wrapper for the function epr.calcamount() and causes that epr.calcamount() cycles through all files in the working directory which match the file extension pattern fragfileext.

Usage

```
epr.calcall(
  mrna,
  fragfileext = "ceqfrag",
  wtf = FALSE,
  resultfileext = "frag",
  standardvaluefile = "standardvalues",
  transcripts = "transcripts.txt",
  outputdir = "./"
)
```

Arguments

`mrna`	Numeric. The amount of RNA in ng which was added to the RT reaction for the express profiling method.
`fragfileext`	The extension of the files which should be read by the function. Defaults to "ceqfrag".
`wtf`	Logical. When set to TRUE the detected fragments of the input file will be written to a file with the extension "resultfileext" (see below). Defaults to FALSE.
`resultfileext`	If wtf is set to TRUE, resultfileext will be chosen as extension for the files containing all detected fragments of each input file. Defaults to "frag".
`standardvaluefile`	Name of the file with the standard values from the function epr.makestd() from the epror package.
`transcripts`	Name of the file with a table of all fragments which should be detected. This file will be read by the function epr.readtranscripts(). Defaults to "transcripts.txt".

`epr.calcamount` *This function calculates transcript amounts*

Description

This function calculates the amount of a given set of transcripts based on so called standard values (see epr.makestd for details).

Usage

```
epr.calcamount(
  mrna,
  file,
  wtf = FALSE,
  resultfileext = "frag",
  standardvaluefile = "standardvalues",
  transcripts = "transcripts.txt",
  outputdir = "./"
)
```

Arguments

`mrna`	Numeric. The amount of RNA in ng which was added to the RT reaction for the express profiling method.
`file`	The name of a fragment file from the AB Sciex CEQ800 software.
`wtf`	Logical. When set to TRUE the detected fragments of the input file will be written to a file with the extension "resultfileext" (see below). Defaults to FALSE.
`resultfileext`	If wtf is set to TRUE, resultfileext will be chosen as extension for the files containing all detected fragments of each input file. Defaults to "frag".
`standardvaluefile`	Name of the file with the standard values from the function epr.makestd() from the epror package.
`transcripts`	Name of the file with a table of all fragments which should be detected. This file will be read by the function epr.readtranscripts(). Defaults to "transcripts.txt".

`epr.checkfrag` *This function checks the presence of a given set of fragment s in fragment files generated by the AB Sciex CEQ8000 software*

Description

This function checks the presence of a set of fragments in a set of files generated by the AB Sciex CEQ8800 software. All files in the working directory which match the file extension pattern will checked. This function returns a data frame. The set of fragments to be analyzed. The set of fragments to be looked for will be read from a file. The function also checks whether the signals of the fragments in the electropherogram are saturated. Therefor also the corresponding raw data files to each fragment file should be in the working directory.

Usage

```
epr.checkfrag(
  rawfileext = "ceqraw",
  fragfileext = "ceqfrag",
  transcripts = "transcripts.txt"
)
```

Arguments

rawfileext	Name of the extension of your raw data files from an AB Sciex CEQ800 instrument. Defaults to "ceqraw".
fragfileext	Name of the extension of your fragment result files from an AB Sciex CEQ800 instrument. Defaults to "ceqfrag".
transcripts	Name of the file wiht a table of all fragments which should be detected. This file will be read by the function epr.readtranscripts(). Defaults to "transcripts.txt".

---

epr.checksizestd *This function checks the presence of a set of size standard fragment in fragment files generated by the AB Sciex CEQ8000 software*

Description

This function checks the presence of fragments of a size standard in a set of fragment files generated by an AB Sciex CEQ8800 software. All files in the working directory which match the file extension pattern will checked. This function returns a data frame.

Usage

```
epr.checksizestd(fragfileext = "ceqfrag")
```

Arguments

fragfileext	Name of the extension of your fragment result files from an AB Sciex CEQ800 instrument.

epr.makestd *This function calculates the standard curves for the express profiling method*

Description

This function calculates the standard curves for the express profiling method to quantify transcripts. The function needs a file with a list of all fragments run data from an AB Sciex CEQ8800 instrument with the amount of RNA in a second column which was used for each run. Additionally you need a table with all transcripts which should be quantified. See epr.readtranscripts() for details. The function returns a list. The first item called standardvalues is a data frame with the standardvalues which are necessary to calculate the transcript amounts of sample data. The second item called data points contains a data frame with all values which were used to calculate the standard curves. The third item contains a data frame called alldata which contains all fragments which could be detected by the function.

Usage

```
epr.makestd(
  file = "amount.txt",
  wtf = FALSE,
  transcripts = "transcripts.txt",
  resultfileext = "frag",
  outputdir = "./"
)
```

Arguments

file	Name of the file which contains a <TAB> separated table with the file names of the fragment files from the AB Sciex CEQ8800 software in the first column and the RNA amount which was used for samples in the second column. Defaults to "amount.txt".
wtf	Logical. When set to TRUE the standardvalues will be written in a file named "standardvalues". Additionally the detected fragments of each input file will be written to a file with the extension "resultfileext" (see below). Defaults to FALSE.
transcripts	Name of the file with a table of all fragments which should be detected. This file will be read by the function epr.readtranscripts(). Defaults to "transcripts.txt".
resultfileext	If wtf is set to TRUE, resultfileext will be choosen as extension for the files containing all detected fragments of each input file. Defaults to "frag".

`epr.plotallcv` *This function plots current and voltage from electropherograms of all AB Sciex CEQ8800 instrument files in the current directory*

Description

This function plots current and voltage from electropherogram from AB Sciex CEQ8800 instrument files. All files which match the pattern given with rawfileext parameter in the current working directory will parsed. This function is a wrapper for the function epr.plotcv() and allows you to cycles through all files in the working directory which match the file extension pattern given with the parameter rawfileext.

Usage

```
epr.plotallcv(rawfileext = "ceqraw", xlim = c(NULL, NULL))
```

Arguments

`rawfileext`	The extension of the files which should be read by the function. Defaults to "ceqraw".
`xlim`	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).

`epr.plotallfrag` *This function graphically displays the fragment list of all AB Sciex CEQ8800 instrument fragment files in the current directory*

Description

This function displays the fragment list of all AB Sciex CEQ8800 instrument fragment files in the current directory which match the file extension pattern given with the parameter fragfileext. All files which match the pattern given of the parameter fragfileext in the current working directory will be parsed. This function is a wrapper for the function epr.plotfrag() and allows you to cycles through all files in the working directory which match the file extension pattern given with the parameter fragfileext.

Usage

```
epr.plotallfrag(
  fragfileext = "ceqfrag",
  transcripts = "transcripts.txt",
  xlim = c(NULL, NULL),
  ylim = c(NULL, NULL)
)
```

Arguments

fragfileext	The extension of the files which should be read by the function. Defaults to "ceqfrag".
transcripts	Name of the file wiht a table of all fragments which should be detected. This file will be read by the function epr.readtranscripts(). Defaults to "transcripts.txt".
xlim	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).
ylim	Vector containing the limits of the y-axis. Defaults to c(NULL,NULL).

---

epr.plotallicv *This function plots current and voltage from electropherograms during the injection phase of all AB Sciex CEQ8800 instrument files in the current directory*

Description

This function current and voltage from electropherograms from AB Sciex CEQ8800 instrument files during the injection phase. All files which match the pattern given with the parameter rawfileext in the current working directory will parsed. This function is a wrapper for the function epr.ploticv() and allows you to cycles through all files in the working directory which match the file extension pattern given with the parameter rawfileext.

Usage

```
epr.plotallicv(rawfileext = "ceqraw", xlim = c(NULL, NULL))
```

Arguments

rawfileext	The extension of the files which should be read by the function. Defaults to "ceqraw".
xlim	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).

---

epr.plotallraw *This function plots an electropherograms of all AB Sciex CEQ8800 instrument files in the current directory*

Description

This function plots electropherograms from AB Sciex CEQ8800 instrument files. All files which match the pattern given of the parameter rawfileext in the current working directory will be parsed. This function is a wrapper for the function epr.plotraw() and allows you to cycles through all files in the working directory which match the file extension pattern given with the parameter rawfileext.

Usage

```
epr.plotallraw(
  rawfileext = "ceqraw",
  xlim = c(NULL, NULL),
  ylim = c(NULL, NULL)
)
```

Arguments

rawfileext	The extension of the files which should be read by the function. Defaults to "ceqraw".
xlim	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).
ylim	Vector containing the limits of the y-axis. Defaults to c(NULL,NULL).

---

epr.plotcv — *This function plots current and voltage from electropherograms of AB Sciex CEQ8800 instrument files*

Description

This function plots current and voltage from electropherogram from AB Sciex CEQ8800 instrument files.

Usage

```
epr.plotcv(file, xlim = c(NULL, NULL))
```

Arguments

file	The name of the AB Sciex CEQ8800 instrument file with the raw data of the electropheorgram. The file must be exported from the database module of the instrument software in the txt format.
xlim	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).

---

epr.plotfrag — *This function graphically displays the fragment list of an AB Sciex CEQ8800 instrument fragment file*

Description

This function displays the fragment list of an AB Sciex CEQ8800 instrument fragment file.

Usage

```
epr.plotfrag(
  file,
  transcripts = "transcripts.txt",
  xlim = c(NULL, NULL),
  ylim = c(NULL, NULL)
)
```

Arguments

file	The name of the AB Sciex CEQ8800 instrument file with the fragment list. The file must be exported from the database module of the instrument software in the txt format.
transcripts	Name of the file wiht a table of all fragments which should be detected. This file will be read by the function epr.readtranscripts(). Defaults to "transcripts.txt".
xlim	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).
ylim	Vector containing the limits of the y-axis. Defaults to c(NULL,NULL).

---

epr.ploticv *This function plots current and voltage from electropherograms of AB Sciex CEQ8800 instrument files during the injection phase*

Description

This function current and voltage from electropherogram from AB Sciex CEQ8800 instrument files during the injection phase.

Usage

```
epr.ploticv(file, xlim = c(NULL, NULL))
```

Arguments

file	The name of the AB Sciex CEQ8800 instrument file with the raw data of the electropheorgramm. The file must be exported from the database module of the instrument software in the txt format.
xlim	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).

`epr.plotraw` *This function plots an electropherograms from an AB Sciex CEQ8800 instrument*

Description

This function plots an electropherogram from an AB Sciex CEQ8800 instrument files.

Usage

```
epr.plotraw(file, xlim = c(NULL, NULL), ylim = c(NULL, NULL))
```

Arguments

`file`	The name of the AB Sciex CEQ8800 instrument file with the raw data of the electropheorgramm. The file must be exported from the database module of the instrument software in the txt format.
`xlim`	Vector containing the limits of the x-axis. Defaults to c(NULL,NULL).
`ylim`	Vector containing the limits of the y-axis. Defaults to c(NULL,NULL).

`epr.plotstdcurve` *This function plots standard curves for the express profiling method*

Description

This function plots the standard curves for the express profiling method. Use the output of the function epr.makestd() as input for this function. As a result, this function plots graphs for each standard curves.

Usage

```
epr.plotstdcurve(
  standardvalues = output[["standardvalues"]],
  datapoints = output[["datapoints"]]
)
```

Arguments

`standardvalues`	Dataframe which was generated by the function epr.makestad() from the package epror. It contains the standardvalues for each fragment. Defaults to "output[["standardvalues"]]".
`datapoints`	Dataframe which was generated by the function epr.makestad() from the package epror. It contains the data points which were used to calculate the standardvalues. Defaults to "output[["datapoints"]]".

`epr.readfragfile` *This function reads a file with the fragment data of an AB Sciex CEQ800 instrument*

Description

This function reads a table with fragment data of an AB Sciex CEQ800 instrument. The file must be exported in the csv format from the database module of the CEQ800 software. This function returns a data frame with the fragment data.

Usage

```
epr.readfragfile(file)
```

Arguments

`file`	File name with fragment data from an AB Sciex CEQ800 instrument.

Examples

```
file <- system.file("extdata", "FA034580.ceqfrag", package="epror")
epr.readfragfile(file)
```

`epr.readinjection` *This function reads the injection part of a file with the raw data of an electropherogram of AB Sciex CEQ800 instrument*

Description

This function reads a table with the injection data from raw data of an electropherogram of an AB Sciex CEQ800 instrument. The file must be exported in the csv format from the database modul of the CEQ800 software. This function returns a data frame with the injection data.

Usage

```
epr.readinjection(file)
```

Arguments

`file`	File name with raw data from an AB Sciex CEQ800 instrument.

Examples

```
file <- system.file("extdata", "FA034580.ceqraw", package="epror")
epr.readinjection(file)
```

epr.readrawdata *This function reads a file with the raw data of an electropherogram of AB Sciex CEQ800 instrument*

Description

This function reads a table from raw data of an electropherogram of an AB Sciex CEQ800 instrument. The file must be exported in the csv format from the database module of the CEQ800 software. This function returns a data frame with the run data.

Usage

```
epr.readrawdata(file)
```

Arguments

file	File name with raw data from an AB Sciex CEQ800 instrument.

Examples

```
file <- system.file("extdata", "FA034580.ceqraw", package="epror")
epr.readrawdata(file)
```

epr.readsettings *Read Settings for Package epror*

Description

This function reads a file with important settings.

Usage

```
epr.readsettings(file = "epr.settings")
```

Arguments

file	File name with your settings. Defaults to "epr.settings".

Examples

```
file <- system.file("extdata", "epr.settings", package="epror")
epr.readsettings(file)
```

`epr.readtranscripts` *This function reads a table with all transcripts that should be detected with the epror package.*

Description

This function reads a table with all transcripts that should be detected with the package. The file must have two <TAB> separated columns. In the first column the file should have a numerical id. The id 0 is reserved for the transcript which serves as internal reference for the rt-pcr. The values in the second column have also to be numerical and stand for the run length of the transcripts. The function returns a data frame with the transcriptid in the first column and the runlength in the second column.

Usage

```
epr.readtranscripts(file = "transcripts.txt")
```

Arguments

`file`	File name with your table of transcripts. Defaults to "transcripts.txt".

Examples

```
file <- system.file("extdata", "transcripts.txt", package="epror")
epr.readtranscripts(file)
```

Index

epr.calcall, 2
epr.calcamount, 3
epr.checkfrag, 3
epr.checksizestd, 4
epr.makestd, 5
epr.plotallcv, 6
epr.plotallfrag, 6
epr.plotallicv, 7
epr.plotallraw, 7
epr.plotcv, 8
epr.plotfrag, 8
epr.ploticv, 9
epr.plotraw, 10
epr.plotstdcurve, 10
epr.readfragfile, 11
epr.readinjection, 11
epr.readrawdata, 12
epr.readsettings, 12
epr.readtranscripts, 13