Package: densitr (via r-universe)

October 14, 2026

Title Analysing Density Profiles from Resistance Drilling of Trees

Version 0.2

Description Provides various tools for analysing density profiles obtained by resistance drilling. It can load individual or multiple files and trim the starting and ending part of each density profile. Tools are also provided to trim profiles manually, to remove the trend from measurements using several methods, to plot the profiles and to detect tree rings automatically. Written with a focus on forestry use of resistance drilling in standing trees.

License GPL-3

Encoding UTF-8

URL https://github.com/krajnc/densitr

BugReports https://github.com/krajnc/densitr/issues

Imports stats, utils, changepoint (>= 2.2.2)

Suggests pbapply, mgcv, knitr, rmarkdown, testthat

RoxygenNote 7.1.2

VignetteBuilder knitr

Repository https://krajnc.r-universe.dev

Date/Publication 2022-03-22 09:51:25 UTC

RemoteUrl https://github.com/krajnc/densitr

RemoteRef HEAD

RemoteSha 2614b02621a9decaf1278dbf77a4d96bd22fc4c3

Contents

combine_data 2
combine_footers 3
correct_failures 4
dpdetect_e 5

dpdetect_s 6
dpdetrend 7
dpload 8
dprings 9
dptrim 11
dptrim_s 12
dptriml 13
dptriml_s 14
extract_dpa_name 15
get_RW 15
manual_trim_detect 16
plot_all 17
plot_end_detection 18
plot_failures 18
plot_start_detection 19
plot_trimming 20
read_dpa 20
remove_trim_failures 21
separate_trim_failures 22
trim_manually 23

Index **24**

`combine_data` *Combines density measurement from a dp object list into a single data frame*

Description

Given a dp object list, this function will extract all density measurement data from all dp objects in a given list and combine them in a single data frame.

Usage

```
combine_data(dp.list)
```

Arguments

`dp.list`	A list of dp objects, either from loading several files using dpload or combined manually. Note: the list should include only dp objects!

Value

A data frame, combining all density data from dp.list

See Also

dpload, combine_footer.

Examples

```
## load all files in directory
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
combine_data(dp.list)
```

`combine_footers` *Combines footer data from a dp object list into a single data frame*

Description

Given a dp object list, this function will extract all footers (the additional measurement data) from all dp objects in a given list and combine them in a single data frame. Will not work if trying to combine footer from newer and older format of data.

Usage

```
combine_footers(dp.list)
```

Arguments

`dp.list`	A list of dp objects, either from loading several files using dpload or combined manually. Note: the list should include only dp objects!

Value

A data frame, combining all footer data from dp.list

See Also

dpload, combine_data.

Examples

```
## load all files in directory
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
combine_footers(dp.list)
```

correct_failures *Manually correct failures after automatic trim detection*

Description

This function will take a list of trimmed dp objects (a result of dptriml or dptriml_s function) and interactively ask the user to assign starting/ending points manually. Follow-up to automatic trim functions or to be used manually, will display a plot with the density profiles for each failure in trim detection sequentially. The plot title will display whether you are selection start or end positions. Use your mouse to select starting/ending point on the plot, your selection will then be displayed on the plot. Will return a complete list, both with the non-failed automatically trimmed dp objects and those corrected manually. The automatic trim functions should be called with the option "rreport = TRUE", which embeds a trimming report when returning the list of trimmed dp objects.

Usage

```
correct_failures(dp.trimmed)
```

Arguments

dp.trimmed	A list of dp objects, trimmed, with the report embedded ("rreport = TRUE").

Value

A list of trimmed profiles, including both automatic and manual trimming.

See Also

dptrim, dptriml, manual_trim_detect

Examples

```
## load all dp files

dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the list
dp.trimmed <- dptriml(dp.list, rreport = TRUE)
## manually correct the failures
dp.corrected <- correct_failures(dp.trimmed)
```

dpdetect_e *Detect measurement ending point automatically using changepoint segmentation*

Description

The opposite of the dpdetect_s, it will check the mean values of the last four segments and compare them to the cutoff limit. Will give a warning if end not detected, which is expected on measurements where the needle did not exit the tree on the opposite side of the tree. See `return.plot = TRUE` to display the actual process. The function is called on a dp object and returns either a row number of the measurement ending or a plot displaying the segmentation and detection. The sensitivity can be adjusted using the cutoff.sd parameter, which is an indicator on how many standard deviations the segment mean value can be before cutting it off.

Usage

```
dpdetect_e(
  dp,
  cutoff.sd = 1,
  return.plot = FALSE,
  minseglen = 250,
  span = 0.1,
  nroll = 100
)
```

Arguments

dp	A dp object, see dpload.
cutoff.sd	How many standard deviations for the cutoff limit?
return.plot	If true, will return a plot displaying segment detection for the current dp file.
minseglen	Minimum segment length for segment detection, default setting of 250 points is for data resolution of 1/100 mm, test a few options with return.plot = TRUE to find the right value
span	Span for loess regression, use to adjust sensitivity of detection detection for the current dp file.
nroll	Number of points for rolling mean, use to adjust sensitivity of detection for the current dp file.

Value

Either a row number where the actual measurement ends or a plot, displaying changepoint segmentation and set limits.

See Also

dpdetect_s, dptrim, dptriml, dptrim_s, dptriml_s

Examples

```
## load a single file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## get ending point
start <- dpdetect_e(dp)
## plot the end detection

dpdetect_e(dp, return.plot = TRUE)
```

`dpdetect_s` *Detect measurement starting point automatically using changepoint segmentation*

Description

A typical resistance drilling measurement starts with an increase in resistance values in between the measurement start and the immersion of the needle in the wood. These values are not useful when estimating density and should be removed before further analysis. This function will detect the starting point automatically using binary segmentation from the package `changepoint`, which separates the measurement in segments based on their mean and variance. Start is detected, when the segment mean is outside of the cutoff limit, see `return.plot = TRUE` to display the diagnostic plot. This function will only check the mean values of the first four (4) segments and compare them to the cutoff value. The function is called on a dp object and returns either a row number of the starting point or a plot displaying the segmentation and detection. The sensitivity can be adjusted using the cutoff.sd parameter, which is an indicator on how many standard deviations the segment mean value can be before cutting it off. Will return a warning if start not detected.

Usage

```
dpdetect_s(
  dp,
  cutoff.sd = 1,
  return.plot = FALSE,
  minseglen = 250,
  span = 0.1,
  nroll = 100
)
```

Arguments

`dp`	A dp object, see dpload.
`cutoff.sd`	How many standard deviations for the cutoff limit?
`return.plot`	If true, will return a plot displaying segment detection for the current dp file.
`minseglen`	Minimum segment length for segment detection, default setting of 250 points is for data resolution of 1/100 mm, test a few options with return.plot = TRUE to find the right value

span	Span for loess regression, use to adjust sensitivity of detection detection for the current dp file.
nroll	Number of points for rolling mean, use to adjust sensitivity of detection for the current dp file.

Value

Either a row number where the actual measurement starts or a plot, displaying changepoint segmentation and set limits.

See Also

dpdetect_e, dptrim, dptriml, dptrim_s, dptriml_s

Examples

```
## load a single file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## get starting point
start <- dpdetect_s(dp)
## plot the start detection

dpdetect_s(dp, return.plot = TRUE)
```

---

dpdetrend *Detrend (remove a trend) a density profile either using linear or GAM regression*

Description

This function will take a dp object and remove the trend from the measurement either by fitting a linear regression or by fitting a GAM regression using REML. The trend is then subtracted from the actual data and a detrended dp object is returned. Be advised detrending should be done on measurements without the starting or ending point, e.g. they should be trimmed. GAM is more useful in tree ring detection, while linear regression is more commonly used for further analysis of the density data. GAM requires `mcgv` package to run.

Usage

```
dpdetrend(dp, type = "")
```

Arguments

dp	A dp object, see dpload.
type	Either "linear" for a fitting linear regression or "gam" for a GAM fit using REML.

Value

A dp object without the trend.

See Also

dptrim, dptriml, dptrim_s, dptriml_s

Examples

```
## load a single file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## load several dp objects
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurement
dp.trimmed <- dptrim(dp)
## detrend the measurement
dp.detrended <- dpdetrend(dp, type = "linear")
## detrend a list without displaying progress
dp.list.detrended <- lapply(dp.list, dpdetrend, type = "linear")
## detrend a list with displaying progress and run in parallel to
## speed things up - requires pbapply library, adjust the cl argument to
## desired number of cores

dp.list.detrended <- pbapply::pblapply(dp.list, dpdetrend, type = "linear", cl = 1)
```

`dpload` *Load a single density profile measurement file (*.dpa) or a directory of *.dpa files.*

Description

Loads either a single .dpa file or a list of .dpa files. If dpa.file is specified, it will load a single file. If dp.directory is specified, it will search for all dpa files in that directory (recursively in all subfolders, can be turned off) and return a list of dp files. It will use pbapply to display progress, if loading a directory.

Usage

```
dpload(dp.file = NULL, dp.directory = "", recursive = TRUE, name = "file")
```

Arguments

`dp.file`	A path to a single file, including file name.
`dp.directory`	A directory with .dpa files.
`recursive`	Also look for density profiles files in subfolders?

name — Either c("file", "folder"), used for naming of list items. If "file", only file name without the complete path will be used for naming ("00050060"). If "folder", the complete path along with file name will be used to name the dpa objects ("data/0005/00/00050060"). *.dpa ending is removed from the name in both cases.

Details

NOTE: for now this function only supports loading density profiles created by the Rinntech Resistograph® resistance drilling device (*.dpa). It was tested to work on files produced by R650-RC drill.

Value

A dp object or a list of dp objects.

Examples

```
## load a single file
dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## load all files in directory
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
```

---

dprings *Automatically identify tree rings in a density profile*

Description

Called on a density profile it will return tree rings, which were automatically detected in the density profile. For best results, run on a trimmed and detrended density profile (use GAM for best results, see dpdetrend). The function will then search for local peaks and valleys within the profile. Normally works well in softwood species, where density increases in late wood and decreases in nearly wood. It will return a data frame containing peaks and valleys, along with their horizontal position. A diagnostic plot will be returned instead when return.plot = TRUE. Green points are valleys, blue points are peaks and red points were automatically excluded. The algorithm will search for peaks and valleys, after which it will automatically exclude all repeated points. Each peak should be followed by a valley and vice versa, when peak-peak situation is found, it will always take the higher peak and the opposite in valleys (keeps the lowest values). Adjust sensitivity by either adjusting pps, which dictates how many points on each side of the identified peak are the minimum. Essentially this dictates the minimum width of detected rings, try adjusting it and display the plot. Minimum peak value can also be adjusted with the parameter threshold, which dictates how many stand deviations from the mean amplitude of the profile is the lowest minimum peak value. Before ring detection the profile can also be denoised by setting smooth = TRUE, which applies a loess regression to smooth the data using the span parameter.

Usage

```
dprings(
  dp,
  pps = 200,
  threshold.sd = 0,
  return.plot = FALSE,
  smooth = FALSE,
  span = 0.01
)
```

Arguments

dp	An dp object, see dpload
pps	Points per peak, the minimum width of a peak, half on each side. A local peak is identified when half of those points are lower on each side of the potential peak. The inverse is true in valleys.
threshold.sd	Minimum peak value in standard deviations away from the overall mean of the signal. By default no peaks are allowed to be beneath the overall mean, can be adjusted to negative to lower the minimum peak allowed.
return.plot	If TRUE, the function will return a diagnostic plot. Green points are valleys, blue points are peaks and red points were automatically excluded.
smooth	Set to TRUE, the profile will be denoised using a LOESS regression.
span	Span of the LOESS regression.

Value

A data frame including the values and positions for all peaks and values. Usually piped into get_RW to get ring widths.

See Also

get_RW

Examples

```
## load a single file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## trim and detrend the measurement
dp.trimmed <- dptrim(dp)
dp.detrended <- dpdetrend(dp.trimmed, type = "gam")
## identify rings
rings <- dprings(dp.detrended)
## plot a diagnostic
dprings(dp.detrended, return.plot = TRUE)
## get tree ring widths:
get_RW(rings)
```

dptrim *Automatically trim an individual density profile on both sides*

Description

Calls dpdetect_s and dpdetect_e on a given dp object and returns a trimmed dp object with the the row before the starting point and after the ending removed. If `return.plot = TRUE`, it will return a plot displaying the dp object with detected starting and ending point. If called with the option `return.fail = FALSE` and `return.plot = FALSE`, the returned object will also include information on whether both cutoff points were detected. If starting/ending point not detected, dp object is returned with no changes. When running on a list of dp objects, use `dptriml`.

Usage

```
dptrim(dp, return.plot = FALSE, return.fail = FALSE, silent = FALSE, ...)
```

Arguments

`dp`	An dp object, see `dpload`
`return.plot`	Return a plot instead of dp object? If TRUE, returns a plot instead of a dp object. When return.fail = TRUE, it returns a list of three: dp object, start trimming success and end trimming success.
`return.fail`	Should information on the success of trimming be included when returning a dp object?
`silent`	Mute detection warnings, used when calling on list. A list of trimmed dp objects, a result of calling dtriml or dptriml_s on a dp list with return.fail = FALSE.
`...`	Parameters minseglen, span and nroll, will get passed through to dpdetect_s, adjust when profile resolution is not 1/100 of a millimeter.

Value

A trimmed dp object, with the beginning and ending removed, if they were detected. When return.plot = TRUE, it returns a plot displaying the process.

See Also

dptriml, dptrim_s, dptriml_s

Examples

```
## load a single dp file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## trim the measurements
dp.trimmed <- dptrim(dp)
## plot trimming
dptrim(dp, return.plot = TRUE)
```

dptrim_s *Automatically trim an individual density profile on the starting side*

Description

Calls dpdetect_s on a given dpa object and returns a trimmed dpa object with the the rows before the starting point removed. If return.plot = TRUE, it will return a plot displaying the dp object with detected starting point. If called with the option return.fail = FALSE and return.plot = FALSE, the returned object will also include information on whether starting cutoff point was detected. If starting point not detected, dp object is returned with no changes. When running on a list of dp objects, use `dtriml_s`.

Usage

```
dptrim_s(dp, return.plot = FALSE, return.fail = FALSE, silent = FALSE, ...)
```

Arguments

`dp`	An dp object, see `dpload`
`return.plot`	Return a plot instead of dp object? If TRUE, returns a plot instead of a dp object. When return.fail = TRUE, it returns a list of three: dp object, start trimming success and end trimming success.
`return.fail`	Should information on the success of trimming be included when returning a dp object?
`silent`	Mute detection warnings, used when calling on list. A list of trimmed dp objects, a result of calling dptriml or dptriml_s on a dp list with rreport = FALSE.
`...`	Parameters minseglen, span and nroll, will get passed through to dpdetect_s, adjust when profile resolution is not 1/100 of a millimeter.

Value

A trimmed dp object, with the beginning and ending removed, if they were detected. When return.plot = TRUE, it returns a plot displaying the process.

See Also

dptrim, dptriml, dptriml_s

Examples

```
## load a single file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## trim the measurement at start
dp.trimmed <- dptrim_s(dp)
## plot trimming
dptrim_s(dp, return.plot = TRUE)
```

`dptriml` *Automatically trim a list of density profiles on both sides*

Description

Calls `dptrim` on a list of dp objects and return a list of trimmed objects. If automatic detection fails, the dp objects are not trimmed. Can be run in parallel on multiple cores, this speeds up the trimming process significantly.

Usage

```
dptriml(dp.list, rreport = FALSE, cl = 1, ...)
```

Arguments

`dp.list`	A list of dp objects, see `dpload`
`rreport`	Return an embedded report on automatic trim success, mandatory when using `correct_failures` to manually pick starting/ending.
`cl`	Number of cores to run the trimming in parallel, passed through to `pbapply`.
`...`	Parameters minseglen, span and nroll, will get passed through to dpdetect_s, adjust when profile resolution is not 1/100 of a millimeter.

Value

A list of trimmed dp objects. When rreport = TRUE, it return a two-item list of (i) trimmed dp objects and (ii) trimming report data frame.

See Also

dptrim, dptrim_s, dptriml_s,

Examples

```
## load several dpa files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements
dp.trimmed <- dptriml(dp.list)
```

dptriml_s *Automatically trim a list of density profiles on the starting side*

Description

Calls dptrim on a list of dp objects and returns a list of trimmed objects. If automatic detection fails, the dp objects are not trimmed. Can be run in parallel on multiple cores, this speeds up the trimming process significantly. Only trims the starting side, see dptriml for trimming both side simultaneously.

Usage

```
dptriml_s(dp.list, rreport = FALSE, cl = 1, ...)
```

Arguments

dp.list	A list of dp objects, see dpload
rreport	Return an embedded report on automatic trim success, mandatory when using correct_failures to manually pick starting/ending.
cl	Number of cores to run the trimming in parallel, passed through to pbapply.
...	Parameters minseglen, span and nroll, will get passed through to dpdetect_s, adjust when profile resolution is not 1/100 of a millimeter.

Value

A list of trimmed dp objects. When rreport = TRUE, it return a two-item list of (i) trimmed dp objects and (ii) trimming report data frame.

See Also

dptrim, dptrim_s, dptriml_s,

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements
dp.trimmed <- dptriml_s(dp.list)
```

`extract_dpa_name` *Extract a file name from a full path*

Description

A wrapper function for regex extraction of filename. Given a character string ("data/0005/00/00050060.dpa"), it will return only the file name without the extension ("00050060").

Usage

```
extract_dpa_name(string)
```

Arguments

`string`	A path to file, including file name. Can be nested in many directories or in none.

Value

An extracted filename, a character string.

References

https://stackoverflow.com/questions/47678725/how-to-do-str-extract-with-base-r

`get_RW` *Get ring widths from identified tree rings*

Description

Called on an object returned by `dprings`, it will return ring widths for all detected rings. The units are determined by the xUnit from the footer of density profile.

Usage

```
get_RW(rings)
```

Arguments

`rings`	A data frame with the identified rings, a result of the dprings() call on an individual profile

Value

A vector of ring widths, which are peak-to-peak differences.

See Also

dprings

Examples

```
## load a single file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## trim and detrend the measurement
dp.trimmed <- dptrim(dp)
dp.detrended <- dpdetrend(dp.trimmed, type = "gam")
## identify rings
rings <- dprings(dp.detrended)
## get tree ring widths:
get_RW(rings)
```

`manual_trim_detect` *Manually select a starting or ending location of a density profile*

Description

Follow-up to automatic trim functions or to be used manually, will display a plot with the density profiles. Most commonly used in automatic failure corrections by the function correct_failures. Use your mouse to select starting/ending point on the plot, your selection will then be displayed on the plot. Either returns a numeric value or NA in case of errors. There are two special cases: when encountering an error with a label = " - PICK START" it will return the starting position, and with label " - PICK STOP" it will return the ending position. These labels are used when correcting several density profiles at once using `correct_failures`. This function uses `graphics::locator`, which only works on screen devices X11, windows and quartz. It will not work on other devices, returning NA.

Usage

```
manual_trim_detect(failure, label = "")
```

Arguments

`failure`	A dp object, usually see dpload.
`label`	Optional label to be displayed on the plot after the file

Value

The x position selected on the graph, row number in the dp$data data frame.

See Also

dptrim, dptriml

Examples

```
## load a single file
dp <- dpload(system.file("extdata", "00010001.dpa", package = "densitr"))
## get a starting point on the plot

manual_trim_detect(dp)
```

`plot_all` *Plot a list of dp objects, one by one*

Description

Plot a list of dp objects, one by one. Press any key to move to the next dp object. Returns nothing.

Usage

```
plot_all(dp.list)
```

Arguments

`dp.list`	A list of dp objects, see `dpload`

See Also

dptrim, dptrim_s, dptriml_s,

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements

if(interactive()){
plot_all(dp.list)
}
```

plot_end_detection *Display end detection on a list of dp objects*

Description

Display an automatic end detection of dp list, each dp object individually. Press any key to move to the next dp object. Returns nothing.

Usage

```
plot_end_detection(dp.list)
```

Arguments

dp.list	A list of dp objects, see dpload

See Also

dptrim, dptrim_s, dptriml_s,

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements

if(interactive()){
plot_end_detection(dp.list)
}
```

plot_failures *Plot trimming failures one by one*

Description

Plot each failed trimming detection, one by one. Press any key to move to the next dp object. Returns nothing. The entry list of dp trimmed objects must include the trimming report (rreport = TRUE).

Usage

```
plot_failures(dp.trimmed)
```

Arguments

dp.trimmed	A list of trimmed dp objects, see dpload

See Also

dptrim, dptrim_s, dptriml_s,

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements
## Not run:
plot_failures(dp.list)

## End(Not run)
```

`plot_start_detection` *Display start detection on a list of dp objects*

Description

Display an automatic start detection of dp list, each dp object individually. Press any key to move to the next dp object. Returns nothing.

Usage

```
plot_start_detection(dp.list)
```

Arguments

`dp.list`	A list of dp objects, see `dpload`

See Also

dptrim, dptrim_s, dptriml_s,

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements

if(interactive()){
plot_start_detection(dp.list)
}
```

plot_trimming *Display automatic trimming on a list of dp objects*

Description

Display an automatic trimming of dp list, each dp object individually. Press any key to move to the next dp object. Returns nothing.

Usage

```
plot_trimming(dp.list)
```

Arguments

dp.list	A list of dp objects, see dpload

See Also

dptrim, dptrim_s, dptriml_s,

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## plot trimming the measurements

if(interactive()){
plot_trimming(dp.list)
}
```

read_dpa *Read a single resistance-drilling density profile measurement file (*.dpa)*

Description

Reads a single *.dpa file and returns a dp object, constructed from two lists: data and footer. The former one contains actual measurement values, the latter includes supplementary data recorded by the Resistograph® device, such as time, firmware number...

Usage

```
read_dpa(file)
```

Arguments

file	A path to file, including file name.

Value

A dp object.

See Also

dpload

`remove_trim_failures` *Remove automatic trim failures from a list of trimmed dp objects and return only non-failed trimmed objects*

Description

Returns a dp list of trimmed dp objects without the failed trim objects. Trimmed dp list should be a result of either calling dtriml on a list of dp objects or calling dtriml_s to remove the starting portions of the measurement. Both functions should be called with the option "return.fail = FALSE", which embeds a trimming report when returning the list of trimmed dp objects.

Usage

```
remove_trim_failures(dp.trimmed)
```

Arguments

`dp.trimmed`	A list of trimmed dp objects, a result of calling dtriml or dtriml_s on a dp list with "return.fail = FALSE".

Value

A dp list of trimmed objects with failures removed.

See Also

dptriml, dptriml_s

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements
dp.trimmed <- dptriml(dp.list, rreport = TRUE)
## remove trimming failures
dp.nofailures <- remove_trim_failures(dp.trimmed)
```

separate_trim_failures *Remove automatic trim failures from a list of trimmed dp objects and return ONLY failures*

Description

An inverse of remove_trim_failures, return a list of failed trimming objects from a trimmed dp list. Trimmed dp list should be a result of either calling dtriml on a list of dp objects or calling dtriml_s to remove the starting portions of the measurement. Both functions should be called with the option "rreport = FALSE", which embeds a trimming report when returning the list of trimmed dp objects. If no failures found, it will return a list of trimmed profiles without the report attached.

Usage

```
separate_trim_failures(dp.trimmed)
```

Arguments

dp.trimmed	A list of trimmed dp objects, a result of calling dptriml or dptriml_s on a dp list with "rreport = FALSE".

Value

Two lists, one with start failures and one with end failures.

See Also

dptriml, dptriml_s

Examples

```
## load several dp files
dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## trim the measurements
dp.trimmed <- dptriml(dp.list, rreport = TRUE)
## separate trimming failures
dp.nofailures <- separate_trim_failures(dp.trimmed)
```

`trim_manually` *Manually trim a list of density profiles*

Description

This function will take a list of dp objects and interactively ask the user to assign starting/ending points manually for all density profiles in sequentially. Used as alternative to automatic trim functions. The plot title will display whether you are selecting start or end position. Use your mouse to select starting/ending points on the plot.

Usage

```
trim_manually(dp.list)
```

Arguments

`dp.list`	A list of dp objects.

Value

A list of trimmed density profiles.

See Also

dptrim, dptriml, manual_trim_detect

Examples

```
## load all dp files

dp.list <- dpload(dp.directory = system.file("extdata", package = "densitr"))
## manually trim the list
dp.trimmed <- trim_manually(dp.list)
```

Index

combine_data, 2
combine_footers, 3
correct_failures, 4

dpdetect_e, 5
dpdetect_s, 6
dpdetrend, 7
dpload, 8
dprings, 9
dptrim, 11
dptrim_s, 12
dptriml, 13
dptriml_s, 14

extract_dpa_name, 15

get_RW, 15

manual_trim_detect, 16

plot_all, 17
plot_end_detection, 18
plot_failures, 18
plot_start_detection, 19
plot_trimming, 20

read_dpa, 20
remove_trim_failures, 21

separate_trim_failures, 22

trim_manually, 23