

# Package ‘HRM.curve’

December 18, 2020

**Type** Package

**Title** High-Resolution Melting (HRM) Curve Analysis

**Version** 0.0.1

**Description** The analysis of high-resolution melting (HRM) curves provides cost-efficient post-PCR distinction among amplicons differing by as little as a single base pair, based on their thermal denaturation (melting) properties. HRM curve analysis has been used mainly for genotyping and detection of mutations. Beside genotyping, the method has been used to differentiate viral, bacterial and fungal pathogens, and it has a potential for plant genotyping in breeding and as an alternative to DNA barcoding for taxonomic assignment of species. This package offers functions that facilitate drawing denaturation, melting, and difference curves in HRM curve analysis in R. Lukas Beule was supported by the German Federal Ministry of Education and Research (BMBF) in the framework of the Bonares-SIGNAL project (funding codes: 031A562A and 031B0510A). Please cite Schiwek et al. (2020) <doi:10.3390/pathogens9040270> if you use this package in your work.

**Depends** R (>= 3.6.0)

**License** CC BY-SA 4.0

**Encoding** UTF-8

**Author** Lukas Beule [aut, cre],  
Simon Schiwek [aut],  
Charlotte Rodemann [aut],  
Petr Karlovsky [aut]

**Maintainer** Lukas Beule <lukas.beule@agr.uni-goettingen.de>

**Repository** CRAN

**NeedsCompilation** no

**Date/Publication** 2020-12-18 09:20:08 UTC

## R topics documented:

Denaturation curves . . . . .	2
Difference curves . . . . .	4
Melting curves . . . . .	7

<b>Index</b>	<b>11</b>
--------------	-----------

## Description

Function draws denaturation curves for selected standards and samples. The number of standards is specified by `number_of_standards`. If all standards have the same number of replicates (balanced design), the mean for each standard can be drawn by specifying `sample_size_standards`; however, **replicates should always be drawn individually first**. The sample numbers of samples for which difference curves will be drawn can be specified as a vector of integers by specifying `sample_number`. The function allows to normalize relative fluorescence values (RFUs) by scaling between 0 and 1 (see [normalization](#)) within the selected temperature range (`temp_range_min`, and `temp_range_max`). Please cite Schiwiek et. al (2020) <<https://doi.org/10.3390/pathogens9040270>> if you use this function in your work.

## Usage

```
denaturation.curves(RFU_data,  
                    normalization = FALSE,  
                    number_of_standards = 1,  
                    sample_size_standards = 1,  
                    sample_number = 1,  
                    temp_range_min = 65,  
                    temp_range_max = 95,  
                    xlab = "temperature",  
                    ylab = "RFU",  
                    col_standards = "black",  
                    col_samples = "forestgreen",  
                    lwd_standards = 1.5,  
                    lwd_samples = 0.75,  
                    lty = 1,  
                    xlim,  
                    ...)
```

## Arguments

- |                                  |   |
|----------------------------------|---|
| <code>RFU_data</code>            | Data frame containing RFU data. Data must be numeric and arranged column-wise. The <b>first column</b> must contain the <b>temperature steps</b> . <b>Standards</b> must start in the <b>second column</b> . <b>Samples</b> must start <b>following the standards</b> . An example of input data can be found here: < <a href="https://github.com/LukasBeule/HRM/blob/main/HRM_example">https://github.com/LukasBeule/HRM/blob/main/HRM_example</a> > |
| <code>normalization</code>       | Logical, RFUs within the selected temperature range ( <code>temp_range_min</code> , and <code>temp_range_max</code> ) of each sample are scaled (0, 1). Default is FALSE.   |
| <code>number_of_standards</code> | Integer, total number of standards. Default is 1.   |

`sample_size_standards`  
 Integer, if all standards have the same number of replicates (balanced design) and `sample_size_standards` is > 1, the mean of all standards will be plotted instead of each individual standard. Default is 1.

`sample_number` Vector of integer(s), specifying sample number(s) of samples that will be analyzed. Default is 1.

`temp_range_min`, `temp_range_max`  
 Numeric, minimum (`temp_range_min`) and maximum (`temp_range_max`) temperature for analysis. Default `temp_range_min` is 65 and `temp_range_max` is 95.

`xlab`, `ylab`, `col_samples`, `col_standards`, `lwd_standards`, `lwd_samples`, `lty`, `xlim`, ...  
 Graphical parameters.

## Details

`denaturation.curves` draws denaturation curves for the selected standards and samples.

## Value

`denaturation.curves` returns a plot of denaturation curves.

## Author(s)

Lukas Beule, Simon Schiwiek, Charlotte Rodemann, Petr Karlovsky

## References

Schiwek, S, Beule, L, Vinas, M, Pfordt, A, von Tiedemann, A, & Karlovsky, P (2020). High-Resolution Melting (HRM) Curve Assay for the Identification of Eight *Fusarium* Species Causing Ear Rot in Maize. *Pathogens*, 9(4), 270.

<<https://doi.org/10.3390/pathogens9040270>>

## See Also

See also `melting.curves` and `difference.curves` for further HRM curve analysis.

## Examples

```
##Input data must be numeric and arranged columnwise.
##Input data should not contain any categorical data.
##An example dataset containing input data with 8 standards
##with 3 technical replicates each and 10 samples can be found here:
##<https://github.com/LukasBeule/HRM/blob/main/HRM\_example\_data.txt>

##load example RFU data:
RFU_data <- read.table("https://raw.githubusercontent.com/LukasBeule/HRM/main/HRM_example_data.txt",
                      sep="\t", dec=".", header=TRUE)

##generate non-normalized denaturation curves
denaturation.curves(RFU_data,
```

```

normalization = FALSE,
number_of_standards = 24,
sample_size_standards = 1,
sample_number = c(1:10),
col_samples = "#000000",
col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
"#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
)

##generate normalized denaturation curves from 81 to 92 degrees Celcius
denaturation.curves(RFU_data,
normalization = TRUE,
number_of_standards = 24,
sample_size_standards = 1,
sample_number = c(1:10),
temp_range_min = 81,
temp_range_max = 92,
col_samples = "#000000",
col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
"#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
)

```

---

Difference curves      *Difference curves for HRM curve analysis*

---

## Description

Function draws difference curves for selected standards and samples. The number of standards is specified by `number_of_standards`. If all standards have the same number of replicates (balanced design), the mean for each standard can be drawn by specifying `sample_size_standards`; however, **replicates should always be drawn individually first**. The sample numbers of samples for which difference curves will be drawn can be specified as a vector of integers by specifying `sample_number`. The function allows to normalize relative fluorescence values (RFUs) (see `normalization`) within the selected temperature range (`temp_range_min`, and `temp_range_max`). Please cite Schiwek et. al (2020) <<https://doi.org/10.3390/pathogens9040270>> if you use this function in your work.

## Usage

```

difference.curves(RFU_data,
derivative = "minus_1st_derivative",
normalization = FALSE,
standard_diff_curve = 1,
number_of_standards = 1,
sample_size_standards = 1,
sample_number = 1,
temp_range_min = 65,
temp_range_max = 95,
draw_peaks = FALSE,

```

```

xlab = "temperature",
ylab = "difference RFU",
col_standards = "black",
col_samples = "forestgreen",
lwd_standards = 1.5,
lwd_samples = 0.75,
lty = 1,
lty_peaks_standards = 1,
lty_peaks_samples = 3,
xlim = c(temp_range_min, temp_range_max),
...)

```

### Arguments

RFU_data	Data frame containing RFU data. Data must be numeric and arranged column-wise. The <b>first column</b> must contain the <b>temperature steps</b> . <b>Standards</b> must start in the <b>second column</b> . <b>Samples</b> must start <b>following the standards</b> . An example of input data can be found here: < <a href="https://github.com/LukasBeule/HRM/blob/main/HRM_example">https://github.com/LukasBeule/HRM/blob/main/HRM_example</a> >
derivative	Character, selection of the derivative of the RFU data. For the negative first derivative, enter "minus_1st_derivative", for the second derivative, enter "2nd_derivative". Default is "minus_1st_derivative".
normalization	Logical, RFUs within the selected temperature range ( <a href="#">temp_range_min</a> , and <a href="#">temp_range_max</a> ) of each sample are scaled (0, 1). Default is FALSE.
standard_diff_curve	Integer, standard used for the calculation of difference curves (zero line in the plot). Default is 1.
number_of_standards	Integer, total number of standards. Default is 1.
sample_size_standards	Integer, if all standards have the same number of replicates (balanced design) and <a href="#">sample_size_standards</a> is > 1, the mean of all standards will be plotted instead of each individual standard. Default is 1.
sample_number	Vector of integer(s), specifying sample number(s) of samples that will be analyzed. Default is 1.
temp_range_min, temp_range_max	Numeric, minimum ( <a href="#">temp_range_min</a> ) and maximum ( <a href="#">temp_range_min</a> ) temperature for analysis. Default <a href="#">temp_range_min</a> is 65 and <a href="#">temp_range_max</a> is 95.
draw_peaks	Logical, draws vertical lines from the maximum (and minimum if <a href="#">derivative</a> is "2nd_derivative") of the melting curves to the x-axis. Default is FALSE.
xlab, ylab, col_standards, col_samples, lwd_standards, lwd_samples, lty, lty_peaks_standards, lty_peaks	Graphical parameters.

### Details

[difference.curves](#) draws difference curves for selected standards and samples.

**Value**

[difference.curves](#) returns a plot of difference curves.

**Author(s)**

Lukas Beule, Simon Schiwiek, Charlotte Rodemann, Petr Karlovsky

**References**

Schiwek, S, Beule, L, Vinas, M, Pfordt, A, von Tiedemann, A, & Karlovsky, P (2020). High-Resolution Melting (HRM) Curve Assay for the Identification of Eight Fusarium Species Causing Ear Rot in Maize. *Pathogens*, 9(4), 270.

<<https://doi.org/10.3390/pathogens9040270>>

**See Also**

See also [denaturation.curves](#) and [melting.curves](#) for further HRM curve analysis.

**Examples**

```
##Input data must be numeric and arranged columnwise.
##Input data should not contain any categorial data.
##An example dataset containing input data with 8 standards
##with 3 technical replicates each and 10 samples can be found here:
##<https://github.com/LukasBeule/HRM/blob/main/HRM\_example\_data.txt>

##load example RFU data:
RFU_data <- read.table("https://raw.githubusercontent.com/LukasBeule/HRM/main/HRM_example_data.txt",
                      sep="\t", dec=".", header=TRUE)

##generate non-normalized difference curves for the first standard
##from 81 to 92 degrees Celcius using negative first derivative
difference.curves(RFU_data,
                 derivative = "minus_1st_derivative",
                 standard_diff_curve = 1,
                 normalization = FALSE,
                 number_of_standards = 24,
                 sample_size_standards = 1,
                 sample_number = c(1:10),
                 temp_range_min = 81,
                 temp_range_max = 92,
                 col_samples = "#000000",
                 col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
                                         "#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
                 )

##generate normalized difference curves for the first standard
##from 81 to 92 degrees Celcius using negative first derivative
##and draw_peak = TRUE.
difference.curves(RFU_data,
                 derivative = "minus_1st_derivative",
```

```

standard_diff_curve = 1,
normalization = TRUE,
number_of_standards = 24,
sample_size_standards = 1,
sample_number = c(1:10),
temp_range_min = 81,
temp_range_max = 92,
draw_peaks = TRUE,
col_samples = "#000000",
col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
"#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
)

##generate non-normalized difference curves for the first standard
##from 81 to 92 degrees Celcius using second derivative
##and draw_peak = TRUE.
difference.curves(RFU_data,
  derivative = "2nd_derivative",
  standard_diff_curve = 1,
  normalization = FALSE,
  number_of_standards = 24,
  sample_size_standards = 1,
  sample_number = c(1:10),
  temp_range_min = 81,
  temp_range_max = 92,
  draw_peaks = TRUE,
  col_samples = "#000000",
  col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
"#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
)

```

---

Melting curves

*Melting curves for HRM curve analysis*


---

## Description

Function draws melting curves for selected standards and samples. The number of standards is specified by `number_of_standards`. If all standards have the same number of replicates (balanced design), the mean for each standard can be drawn by specifying `sample_size_standards`; however, **replicates should always be drawn individually first**. The sample numbers of samples for which difference curves will be drawn can be specified as a vector of integers by specifying `sample_number`. The function allows to normalize relative fluorescence values (RFUs) by scaling between 0 to 1 (see `normalization`) within the selected temperature range (`temp_range_min`, and `temp_range_max`). Please cite Schiwiek et. al (2020) <<https://doi.org/10.3390/pathogens9040270>> if you use this function in your work.

## Usage

```
melting.curves(RFU_data,
```

```

derivative = "minus_1st_derivative",
normalization = FALSE,
number_of_standards = 1,
sample_size_standards = 1,
sample_number = 1,
temp_range_min = 65,
temp_range_max = 95,
draw_peaks = FALSE,
xlab = "temperature",
ylab = "-d(RFU)",
col_samples = "forestgreen",
col_standards = "black",
lwd_standards = 1.5,
lwd_samples = 0.75,
lty = 1,
lty_peaks_standards = 1,
lty_peaks_samples = 3,
xlim = c(temp_range_min, temp_range_max),
...)

```

### Arguments

RFU_data	Data frame containing RFU data. Data must be numeric and arranged column-wise. The <b>first column</b> must contain the <b>temperature steps</b> . <b>Standards</b> must start in the <b>second column</b> . <b>Samples</b> must start <b>following the standards</b> . An example of input data can be found here: < <a href="https://github.com/LukasBeule/HRM/blob/main/HRM_example">https://github.com/LukasBeule/HRM/blob/main/HRM_example</a> >
derivative	Character, selection of the derivative of the RFU data. For the negative first derivative, enter "minus_1st_derivative", for the second derivative, enter "2nd_derivative". Default is "minus_1st_derivative".
normalization	Logical, RFUs within the selected temperature range ( <a href="#">temp_range_min</a> , and <a href="#">temp_range_max</a> ) of each sample are scaled (0, 1). Default is FALSE.
number_of_standards	Integer, total number of standards. Default is 1.
sample_size_standards	Integer, if all standards have the same number of replicates (balanced design) and <a href="#">sample_size_standards</a> is > 1, the mean of all standards will be plotted instead of each individual standard. Default is 1.
sample_number	Vector of integer(s), specifying sample number(s) of samples that will be analyzed. Default is 1.
temp_range_min, temp_range_max	Numeric, minimum ( <a href="#">temp_range_min</a> ) and maximum ( <a href="#">temp_range_min</a> ) temperature for analysis. Default <a href="#">temp_range_min</a> is 65 and <a href="#">temp_range_max</a> is 95.
draw_peaks	Logical, draws vertical lines from the maximum (and minimum if <a href="#">derivative</a> is "2nd_derivative") of the melting curves to the x-axis. Default is FALSE.
xlab, ylab, col_standards, col_samples, lwd_standards, lwd_samples, lty, lty_peaks_standards, lty_peaks	Graphical parameters.

**Details**

[melting.curves](#) draws melting curves for selected standards and samples.

**Value**

[melting.curves](#) returns a plot of difference curves.

**Author(s)**

Lukas Beule, Simon Schiwiek, Charlotte Rodemann, Petr Karlovsky

**References**

Schiwek, S, Beule, L, Vinas, M, Pfordt, A, von Tiedemann, A, & Karlovsky, P (2020). High-Resolution Melting (HRM) Curve Assay for the Identification of Eight Fusarium Species Causing Ear Rot in Maize. *Pathogens*, 9(4), 270.

<<https://doi.org/10.3390/pathogens9040270>>

**See Also**

See also [denaturation.curves](#) and [difference.curves](#) for further HRM curve analysis.

**Examples**

```
##Input data must be numeric and arranged columnwise.
##Input data should not contain any categorical data.
##An example dataset containing input data with 8 standards
##with 3 technical replicates each and 10 samples can be found here:
##<https://github.com/LukasBeule/HRM/blob/main/HRM\_example\_data.txt>

##load example RFU data:
RFU_data <- read.table("https://raw.githubusercontent.com/LukasBeule/HRM/main/HRM_example_data.txt",
                      sep="\t", dec=".", header=TRUE)

##generate non-normalized melting curves from 81 to 92 degrees Celcius
##using negative first derivative
melting.curves(RFU_data,
               derivative = "minus_1st_derivative",
               normalization = FALSE,
               number_of_standards = 24,
               sample_size_standards = 1,
               sample_number = c(1:10),
               temp_range_min = 81,
               temp_range_max = 92,
               col_samples = "#000000",
               col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
                                     "#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
               )

##generate normalized melting curves from 81 to 92 degrees Celcius
##using negative first derivative and draw_peak = TRUE
```

```
melting.curves(RFU_data,
               derivative = "minus_1st_derivative",
               normalization = TRUE,
               number_of_standards = 24,
               sample_size_standards = 1,
               sample_number = c(1:10),
               temp_range_min = 81,
               temp_range_max = 92,
               draw_peaks = TRUE,
               col_samples = "#000000",
               col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
                                       "#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
)

##generate non-normalized melting curves from 81 to 92 degrees Celcius
##using second derivative and draw_peak = TRUE
melting.curves(RFU_data,
               derivative = "2nd_derivative",
               normalization = FALSE,
               number_of_standards = 24,
               sample_size_standards = 1,
               sample_number = c(1:10),
               temp_range_min = 81,
               temp_range_max = 92,
               draw_peaks = TRUE,
               ylab = "d2(RFU)",
               col_samples = "#000000",
               col_standards = c(rep(c("#999999", "#E69F00", "#56B4E9", "#009E73",
                                       "#F0E442", "#0072B2", "#D55E00", "#CC79A7"), each = 3))
)
```

# Index

Denaturation curves, [2](#)  
denaturation.curves, [3](#), [6](#), [9](#)  
denaturation.curves (Denaturation curves), [2](#)  
derivative, [5](#), [8](#)  
Difference curves, [4](#)  
difference.curves, [3](#), [5](#), [6](#), [9](#)  
difference.curves (Difference curves), [4](#)

Melting curves, [7](#)  
melting.curves, [3](#), [6](#), [9](#)  
melting.curves (Melting curves), [7](#)

normalization, [2](#), [4](#), [7](#)  
number\_of\_standards, [2](#), [4](#), [7](#)

sample\_number, [2](#), [4](#), [7](#)  
sample\_size\_standards, [2-5](#), [7](#), [8](#)

temp\_range\_max, [2-5](#), [7](#), [8](#)  
temp\_range\_min, [2-5](#), [7](#), [8](#)