# Package: FUNTA (via r-universe)

October 31, 2024

**Type** Package

**Title** Functional Tangential Angle Pseudo-Depth

**Version** 0.1.0

**Author** Andre Rehage

**Maintainer** Andre Rehage <rehage@statistik.tu-dortmund.de>

**Description** Computes the functional tangential angle pseudo-depth and its robustified version from the paper by Kuhnt and Rehage (2016). See Kuhnt, S.; Rehage, A. (2016): An angle-based multivariate functional pseudo-depth for shape outlier detection, JMVA 146, 325-340, <doi:10.1016/j.jmva.2015.10.016> for details.

**License** GPL-3

**LazyData** TRUE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-04-01 14:33:13

## Contents

## FUNTA-package *Obtain FUNTA and rFUNTA pseudo-depth for a given functional dataset*

### Description

Given the discretized functional observations, the functions `FUNTA` and `rFUNTA` are able to compute the functional tangential angle pseudo-depths as introduced in Kuhnt and Rehage (2016).

### Details

For FUNTA and rFUNTA, insert the data in matrix form (n by T, where n is the number of observations and T is the number of time points).

### Author(s)

A. Rehage.

### References

Kuhnt, S.; Rehage, A. (2016) An angle-based multivariate functional pseudo-depth for shape outlier detection. *Journal of Multivariate Analysis* 146, 325-340.

### Examples

```
x <- seq(0, 2*pi, by = 0.01)
y1 <- sin(x)
y2 <- sin(1.02*x)
y3 <- cos(x)
y <- rbind(y1, y2, y3)
FUNTA(y, tick.dist = 0.01)
rFUNTA(y, tick.dist = 0.01)
```

## angle *Auxiliary function to compute intersection angles*

### Description

Internal function that is used by `FUNTA` and `rFUNTA`.

### Usage

```
angle(cut.slope, ref.slope, tick.dist)
```

### Arguments

`cut.slope` The slope of function x1.

`ref.slope` The slope of function x2.

`tick.dist` The length of the slopes.

### Value

The intersection angle of x1 and x2.

### Author(s)

A. Rehage

### Examples

```
angle(1,2,0.1)
```

## `FUNTA` *Obtain FUNTA pseudo-depth values*

### Description

For a given dataset, FUNTA pseudo-depth values can be obtained. FUNTA is a functional data depth that is based on the intersection angles that the centered functions form with each other.

### Usage

```
FUNTA(Data, centered = FALSE, give.angles = FALSE, tick.dist = 1)
```

### Arguments

`Data` a matrix. Enter the discretized values of a functional data set in a n times T matrix, where n is the number of functional observations and T is the number of time points.

`centered` boolean. If the data are already centered, that means, the mean of each row of `Data` is 0, this can be set to `TRUE` to save computation time. Default value is `FALSE`.

`give.angles` boolean. If the intersection angles of each function with the other functions are to be displayed, set to `TRUE`. Default value is `FALSE`.

`tick.dist` atomic vector. The distance between two neighbored time points can be set here. Default value is `1`.

### Details

The larger the value of FUNTA is, the less it can be regarded as a shape outlier, and vice versa. The values are bounded by 0 and 1.

## Value

If `give.angles` = `TRUE`, a list of two elements FUNTA and Angles. Otherwise only the first element of that list is returned.

`FUNTA` Vector of FUNTA values. First row of `Data` corresponds to first element of `FUNTA`.

`Angles` List of intersection angles. First element of list corresponds to the intersection angles that the first row of `Data` has with every other row of `Data`, ordered by time point of intersection.

## Author(s)

A. Rehage

## References

Kuhnt, S.; Rehage, A. (2016) An angle-based multivariate functional pseudo-depth for shape outlier detection. *Journal of Multivariate Analysis* 146, 325-340.

## Examples

```
x <- seq(0, 2*pi, by = 0.01)
y1 <- sin(x)
y2 <- sin(1.02*x)
y3 <- cos(x)
y <- rbind(y1, y2, y3)
FUNTA(y, tick.dist = 0.01)
```

---

# `rFUNTA` *Obtain rFUNTA pseudo-depth values*

---

## Description

For a given dataset, rFUNTA pseudo-depth values can be obtained. rFUNTA is a robustified functional data depth that is based on the intersection angles that the centered functions form with each other.

## Usage

```
rFUNTA(Data, centered = FALSE, type.inner = "max", type.outer = "median", tick.dist = 1,
nObs = nrow(Data))
```

### Arguments

`Data` a matrix. Enter the discretized values of a functional data set in a n times T matrix, where n is the number of functional observations and T is the number of time points.

`centered` boolean. If the data are already centered, that means, the mean of each row of `Data` is 0, this can be set to `TRUE` to save computation time. Default value is `FALSE`.

`type.inner` One of `"max"` (default), `"median"`, `"mean"`. Note that only the default setting produces rFUNTA values as introduced in Kuhnt and Rehage (2016). The other options can be used if not the maximum intersection angle of each pair of functions is of interest, but the median or mean intersection angle.

`type.outer` One of `"max"`, `"median"` (default), `"mean"`. Note that only the default setting produces rFUNTA values as introduced in Kuhnt and Rehage (2016). The other options can be used if not the *median* value of the n-1 weighted intersection angles of each function is of interest, but the maximum or mean of it.

`tick.dist` atomic vector. The distance between two neighbored time points can be set here. Default value is `1`.

`nObs` atomic vector. If the dataset has more than one dimension, specify `nObs` with the number of observations per dimension. `Data` then has to be in the style of `rbind(Dim1, Dim2, ...)`. Note that `tick.dist` has to be equal for all the dimensions.

### Details

The larger the value of FUNTA is, the less it can be regarded as a shape outlier, and vice versa. The values are bounded by 0 and 1.

### Value

Vector of rFUNTA values. First observation in `Data` corresponds to first element of `FUNTA`.

### Author(s)

A. Rehage

### References

Kuhnt, S.; Rehage, A. (2016) An angle-based multivariate functional pseudo-depth for shape outlier detection. *Journal of Multivariate Analysis* 146, 325-340.

### Examples

```
x <- seq(0, 2*pi, by = 0.01)
y1 <- sin(x)
y2 <- sin(1.02*x)
y3 <- cos(x)
y <- rbind(y1, y2, y3)
rFUNTA(y, tick.dist = 0.01)
```

# Index