

# Package ‘similaRpeak’

May 20, 2024

**Type** Package

**Title** Metrics to estimate a level of similarity between two ChIP-Seq profiles

**Version** 1.36.0

**Date** 2014-08-08

**Description** This package calculates metrics which quantify the level of similarity between ChIP-Seq profiles. More specifically, the package implements six pseudometrics specialized in pattern similarity detection in ChIP-Seq profiles.

**biocViews** BiologicalQuestion, ChIPSeq, Genetics, MultipleComparison, DifferentialExpression

**Depends** R6 (>= 2.0)

**Imports** stats

**Suggests** RUnit, BiocGenerics, knitr, Rsamtools, GenomicAlignments, rtracklayer, rmarkdown, BiocStyle

**License** Artistic-2.0

**URL** <https://github.com/adeschen/similaRpeak>

**BugReports** <https://github.com/adeschen/similaRpeak/issues>

**VignetteBuilder** knitr

**Encoding** UTF-8

**NeedsCompilation** no

**RoxygenNote** 6.0.1

**git\_url** <https://git.bioconductor.org/packages/similaRpeak>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 39cd226

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-19

**Author** Astrid Deschênes [cre, aut],  
 Elsa Bernatchez [aut],  
 Charles Joly Beuparlant [aut],  
 Fabien Claude Lamaze [aut],  
 Rawane Samb [aut],  
 Pascal Belleau [aut],  
 Arnaud Droit [aut]

**Maintainer** Astrid Deschênes <adeschen@hotmail.com>

## Contents

|                                    |           |
|------------------------------------|-----------|
| similaRpeak-package . . . . .      | 2         |
| chr7Profiles . . . . .             | 3         |
| demoProfiles . . . . .             | 5         |
| DiffPosMax . . . . .               | 6         |
| diffPosMaxMethod . . . . .         | 8         |
| Metric . . . . .                   | 9         |
| MetricFactory . . . . .            | 10        |
| RatioArea . . . . .                | 12        |
| ratioAreaMethod . . . . .          | 13        |
| RatioIntersect . . . . .           | 14        |
| ratioIntersectMethod . . . . .     | 15        |
| RatioMaxMax . . . . .              | 15        |
| ratioMaxMaxMethod . . . . .        | 16        |
| RatioNormalizedIntersect . . . . . | 17        |
| similarity . . . . .               | 18        |
| SpearmanCorrelation . . . . .      | 21        |
| spearmanCorrMethod . . . . .       | 22        |
| <b>Index</b>                       | <b>24</b> |

---

|                     |   |
|---------------------|---|
| similaRpeak-package | <i>similaRpeak: Metrics to estimate a level of similarity between two ChIP-Seq profiles</i> |
|---------------------|---|

---

## Description

This package is calculating six different metrics to estimate a level of similarity between two ChIP-Seq profiles.

## Details

The `similarity` function calculates six different metrics:

- `RATIO_AREA`: The ratio between the areas. The larger value is always divided by the smaller value.

- **DIFF\_POS\_MAX**: The difference between the maximal peaks positions. The difference is always a positive value.
- **RATIO\_MAX\_MAX**: The ratio between the maximal peaks values. The larger value is always divided by the smaller value.
- **RATIO\_INTERSECT**: The ratio between the intersection area and the total area.
- **RATIO\_NORMALIZED\_INTERSECT**: The ratio between the intersection area and the total area of two normalized profiles. The profiles are normalized by dividing them by their average value.
- **SPEARMAN\_CORRELATION**: The Spearman's rho statistic between profiles.

The function `similarity` also reports basic information about each ChIP profile such as the number of positions, the area, the maximum value and the position of the maximum value.

To learn more about **similaRpeak** package see: <https://github.com/adeschen/similaRpeak/wiki>

#### Author(s)

Astrid Deschenes, Elsa Bernatchez, Charles Joly Beuparlant, Fabien Claude Lamaze, Rawane Samb, Pascal Belleau and Arnaud Droit

Maintainer: Astrid Deschenes <adeschen@hotmail.com>

#### See Also

- [MetricFactory](#) for using a interface to calculate all available metrics separately.
- [similarity](#) for calculating all available metrics between two ChIP-Seq profiles.

---

chr7Profiles

*ChIP-Seq profiles of region chr7:61968807-61969730 related to enhancers H3K27ac and H3K4me1 (for demonstration purpose)*

---

#### Description

ChIP-Seq profiles of region chr7:61968807-61969730 of two histone post-transcriptional modifications linked to highly active enhancers H3K27ac (DCC accession: ENCF000ASG) and H3K4me1 (DCC accession: ENCF000ARY) from the Encyclopedia of DNA Elements (ENCODE) data (Dunham I et al. 2012).

#### Usage

```
data(chr7Profiles)
```

## Format

A list with 1 entry. The entry is a list of 2 ChIP-Seq profiles, one per active enhancer (H3K27ac and H3K4me1). The 2 ChIP-Seq profiles are of identical length and specific to a genomic region. Each ChIP-Seq profile is a numerical vector containing the profiles values at each position, as reported in reads per million (RPM).

- `chr7Profiles` a list containing all demo ChIP-Seq profiles
- `chr7Profiles$chr7.61968807.61969730` a list containing 2 ChIP-Seq profiles for the genomic region `chr7:6196880-61969730`
- `demoProfiles$chr7.61968807.61969730$H3K27ac` a numeric vector containing the profiles values related to the enhancer H3K27ac, as reported in reads per million (RPM). The first entry of the vector is for position `chr7:61968807` while the last entry is for position `chr7:61969730`
- `demoProfiles$chr7.61968807.61969730$H3K4me1` a numeric vector containing the profiles values related to the enhancer H3K4me1, as reported in reads per million (RPM). The first entry of the vector is for position `chr7:61968807` while the last entry is for position `chr7:61969730`

## Source

The Encyclopedia of DNA Elements (ENCODE) (DCC accession: ENCFF000MZT)

## References

- Dunham I, Kundaje A, Aldred SF, et al. An integrated encyclopedia of DNA elements in the human genome. *Nature*. 2012 Sep 6;489(7414):57-74.

## See Also

- [demoProfiles](#) ChIP-seq profiles related to enhancers H3K27ac and H3K4me1 (for demonstration purpose)
- [MetricFactory](#) for using a interface to calculate all available metrics separately.
- [similarity](#) for calculating all available metrics between two ChIP-Seq profiles.

## Examples

```
data(chr7Profiles)

## Calculating all metrics for the "chr7.61968807.61969730" region
metrics <- similarity(chr7Profiles$chr7.61968807.61969730$H3K4me1,
chr7Profiles$chr7.61968807.61969730$H3K27ac,
  ratioAreaThreshold=10,
  ratioMaxMaxThreshold=4,
  ratioIntersectThreshold=5,
  ratioNormalizedIntersectThreshold=2,
  diffPosMaxThresholdMinValue=10,
  diffPosMaxThresholdMaxDiff=100,
  diffPosMaxTolerance=0.10)
metrics
```

```
## You can refer to the vignette to see more examples using ChIP-Seq profiles
## extracted from the Encyclopedia of DNA Elements (ENCODE) data.
```

---

|              |  |
|--------------|--|
| demoProfiles | <i>ChIP-Seq profiles of region chr7:61968807-61969730 related to enhancers H3K27ac and H3K4me1 (for demonstration purpose)</i> |
|--------------|--|

---

## Description

ChIP-Seq profiles of region chr7:61968807-61969730 of two histone post-transcriptional modifications linked to highly active enhancers H3K27ac (DCC accession: ENCF000ASG) and H3K4me1 (DCC accession: ENCF000ARY) from the Encyclopedia of DNA Elements (ENCODE) data (Dunham I et al. 2012).

## Usage

```
data(demoProfiles)
```

## Format

A list with 1 entry. The entry is a list of 2 ChIP-Seq profiles, one per active enhancer (H3K27ac and H3K4me1). The 2 ChIP-Seq profiles are of identical length and specific to a genomic region. Each ChIP-Seq profile is a numerical vector containing the profiles values at each position, as reported in reads per million (RPM).

- `demoProfiles` a list containing all demo ChIP-Seq profiles
- `demoProfiles$chr2.70360770.70361098` a list containing 2 ChIP-Seq profiles for the genomic region chr2:70360770-70361098
- `demoProfiles$chr2.70360770.70361098$H3K27ac` a numeric vector containing the profiles values related to the enhancer H3K27ac, as reported in reads per million (RPM). The first entry of the vector is for position chr1:70360770 while the last entry is for position chr2:70361098
- `demoProfiles$chr2.70360770.70361098$H3K4me1` a numeric vector containing the profiles values related to the enhancer H3K4me1, as reported in reads per million (RPM). The first entry of the vector is for position chr1:70360770 while the last entry is for position chr2:70361098
- `demoProfiles$chr3.73159773.73160145$H3K4me1` a list containing 2 ChIP-Seq profiles for the genomic region chr3:73159773-73160145
- `demoProfiles$chr3.73159773.73160145$H3K27ac` a numeric vector containing the profiles values related to the enhancer H3K27ac, as reported in reads per million (RPM). The first entry of the vector is for position chr2:73159773 while the last entry is for position chr3:73160145
- `demoProfiles$chr3.73159773.73160145$H3K4me1` a numeric vector containing the profiles values related to the enhancer H3K4me1, as reported in reads per million (RPM). The first entry of the vector is for position chr3:73159773 while the last entry is for position chr3:73160145

**Source**

The Encyclopedia of DNA Elements (ENCODE) (DCC accession: ENCF000MZT)

**References**

- Dunham I, Kundaje A, Aldred SF, et al. An integrated encyclopedia of DNA elements in the human genome. *Nature*. 2012 Sep 6;489(7414):57-74.

**See Also**

- [chr7Profiles](#) ChIP-Seq profiles of region chr7:61968807-61969730 related to enhancers H3K27ac and H3K4me1 (for demonstration purpose)
- [MetricFactory](#) for using a interface to calculate all available metrics separately.
- [similarity](#) for calculating all available metrics between two ChIP-Seq profiles.

**Examples**

```
data(demoProfiles)

# Calculate metrics for the "chr3:73159773-73160145" region
metrics <- similarity(demoProfiles$chr3.73159773.73160145$H3K27ac,
  demoProfiles$chr3.73159773.73160145$H3K4me1)
metrics

## You can refer to the vignette to see more examples using ChIP-Seq profiles
## extracted from the Encyclopedia of DNA Elements (ENCODE) data.
```

---

DiffPosMax

*DiffPosMax class*

---

**Description**

An object which is a interface to calculate the difference of profiles maximal peaks positions.

The DiffPosMax object is needed to calculate the difference of profiles maximal peaks positions. A threshold and the two profiles are set during the DiffPosMax object creation. If different thresholds or profiles are needed, the calculateMetric function should be used, with the new profiles and thresholds passed as arguments to update those values inside the DiffPosMax object.

The threshold is the minimum peak value accepted to calculate the ratio.

The thresholdDiff is the maximum distance accepted between two maximum peaks positions in the same profile. When the thresholdDiff is not respected, the profile is considered having more than one peak.

The tolerance is the maximum variation accepted on the maximum peak value to consider a position as a peak position. The tolerance must be between 0 and 1. All peaks within the tolerated range will be considered in the calculation of the metric.

**Usage**

DiffPosMax

**Format**

An object of class R6ClassGenerator of length 24.

**Value**

The DiffPosMax\$new function returns a DiffPosMax object which contains the information about the two profiles and the thresholds used to calculate the metric. It can be used, as many times needed, to calculate the specified metric.

**Constructor**

Create a DiffPosMax object.

```
DiffPosMax$new(profile1, profile2, threshold = 1, thresholdDiff = 100, tolerance = 0.01)
```

The threshold is the minimum peak value accepted to calculate the ratio.

The thresholdDiff is the maximum distance accepted between two maximum peaks positions in the same profile. When the thresholdDiff is not respected, the profile is considered having more than one peak.

The tolerance is the maximum variation accepted on the maximum peak value to consider a position as a peak position. All peaks within the tolerated range will be considered in the calculation of the metric. The tolerance must

The DiffPosMax object inherits those functions:

- `getMetric` A function that returns the value of the calculated metric
- `getInfo` A function that returns a description of the metric with the metric value.
- `getType` A function that returns the unique name associated to this metric
- `calculateMetric` A function that modifies the values of the two profiles and the threshold. The new values (profile1, profile2, threshold, thresholdDiff, tolerance) are passed as arguments.

**Author(s)**

Astrid Deschenes

**See Also**

- [MetricFactory](#) for using a interface to calculate all available metrics separately or together.

---

diffPosMaxMethod      *Difference between two profiles maximal peaks positions*

---

### Description

Calculate and return the difference between two profiles maximal peaks positions. The difference is always the first profile value (profile1 parameter) minus the second profile value (profile2 parameter). When more than one maximal peak is present in one profile, the median of the position is calculated and used as the maximal peak position. If one threshold is not respected, the function returns NA.

### Usage

```
diffPosMaxMethod(profile1, profile2, threshold = 1, thresholdDist = 100,  
tolerance = 0.01)
```

### Arguments

|               |   |
|---------------|---|
| profile1      | a vector of numeric values, the first profile containing the alignment depth for each position. The profile1 and profile2 should have the same length.  |
| profile2      | a vector of numeric values, the second profile containing the alignment depth for each position. The profile1 and profile2 should have the same length. |
| threshold     | a numeric, the minimum peak value accepted to calculate the metric. Default = 1.  |
| thresholdDist | a numeric, the maximum distance accepted between two maximum peaks positions in the same profile. Default = 100.  |
| tolerance     | a numeric, the maximum variation accepted on the maximum value to consider a position as a peak position. The tolerance must                            |

### Value

The calculated ratio or NA if not all thresholds are respected.

### Author(s)

Astrid Deschenes, Elsa Bernatchez

### See Also

- [MetricFactory](#) for using the recommended interface to calculate all available metrics separately or together.



---

|        |                     |
|--------|---------------------|
| Metric | <i>Metric class</i> |
|--------|---------------------|

---

**Description**

A class which represents a abstract Metric object which is used to quantify the similarity between two profiles covering the same range.

The Metric class should not be directly instanciated. It should be used as the parent class of a specific metric class.

**Usage**

Metric

**Format**

An object of class R6ClassGenerator of length 24.

**Value**

The Metric\$new function returns a Metric object which contains the information about the two profiles and the threshold used to calculate the metric. It can be used, as many times needed, to calculate the specified metric.

**Constructor**

Create a Metric object.

```
Metric$new(profile1, profile2, threshold = NULL)
```

The Metric object inherits those functions:

- `getMetric` A function that returns the value of the calculated metric
- `getInfo` A function that returns a description of the metric with the metric value.
- `getType` A function that returns the unique name associated to this metric
- `calculateMetric` A function that modifies the values of the two profiles and the threshold. The new values (`profile1`, `profile2`, `threshold`) are passed as arguments.

**Author(s)**

Astrid Deschenes

**See Also**

- [MetricFactory](#) for using the recommended interface to calculate all available metrics separately or together.

---

|               |                             |
|---------------|-----------------------------|
| MetricFactory | <i>MetricFactory object</i> |
|---------------|-----------------------------|

---

### Description

An object which is a interface to calculate all available metrics separately.

The MetricFactory object is inspired from the factory design pattern. Only one instance of MetricFactory object is necessary to calculate all available metrics for different profiles, as long as the thresholds set in the MetricFactory instance are appropriate for the calculation. The thresholds are set during the MetricFactory object creation and cannot be changed afterwards. If different thresholds are needed, a new MetricFactory object, with the new thresholds, must be instantiated.

### Usage

MetricFactory

### Format

An object of class R6ClassGenerator of length 24.

### Value

The MetricFactory\$new function returns a MetricFactory object which contains the information about the thresholds used to calculate each metric. It can be used, as many times needed, to calculate the specified metrics.

### Constructor

Create a MetricFactory object.

```
MetricFactory$new(ratioAreaThreshold=1, ratioMaxMaxThreshold=1, ratioIntersectThreshold=1,
ratioNormalizedIntersectThreshold=1, diffPosMaxThresholdMinValue=1, diffPosMaxThresholdMaxDiff=100,
diffPosMaxTolerance=0.01, spearmanCorrSDThreshold=1e-8)
```

- `ratioAreaThreshold` The minimum denominator accepted to calculate the ratio of the area between both profiles. Default = 1.
- `ratioMaxMaxThreshold` The minimum denominator accepted to calculate the ratio of the maximum values between both profiles. Default = 1.
- `ratioIntersectThreshold` The minimum denominator accepted to calculate the ratio of the intersection area of both profiles and the total area. Default = 1.
- `ratioNormalizedIntersectThreshold` The minimum denominator accepted to calculate the ratio of the intersection area of both profiles and the total area for normalized profiles. Default = 1.
- `diffPosMaxThresholdMinValue` The minimum peak accepted to calculate the metric. Default = 1.
- `diffPosMaxThresholdMaxDiff` The maximum distance accepted between 2 peaks positions in one profile to calculate the metric. Default=100.

- `diffPosMaxTolerance` The maximum variation accepted on the maximum value to consider a position as a peak position. Default=0.01.
- `spearmanCorrSDThreshold` The minimum standard deviation accepted on both profiles to consider to calculate the metric. Default=1e-8.

### Author(s)

Astrid Deschenes

### See Also

- [similarity](#) for calculating all available metrics between two ChIP-Seq profiles.
- [demoProfiles](#) for more informations about ChIP-Seq profiles present in the demoProfiles data.

### Examples

```
## Initialized the factory object
factory = MetricFactory$new(ratioAreaThreshold=100,
  ratioIntersectThreshold=20,
  diffPosMaxTolerance=0.04)

## Define 2 ChIP-Seq profiles
profile1 <- c(1,59,6,24,65,34,15,4,53,22)
profile2 <- c(15,9,46,44,9,39,27,34,34,4)

## Use the factory object to calculate each metric separatly
ratio_max_max <- factory$createMetric(metricType="RATIO_MAX_MAX",
  profile1, profile2)
ratio_max_max

diff_pos_max <- factory$createMetric(metric="DIFF_POS_MAX", profile1,
  profile2)
diff_pos_max

## Example using ChIP-Seq profiles of H3K27ac (DCC accession: ENCF000ASG)
## and H3K4me1 (DCC accession: ENCF000ARY) from the Encyclopedia of DNA
## Elements (ENCODE) for the region
data(demoProfiles)

## Visualize ChIP-Seq profiles
plot(demoProfiles$chr3.73159773.73160145$H3K27ac, type="l", col="blue",
  xlab="", ylab="", ylim=c(0, 125), main="chr3:73159773-73160145")
par(new=TRUE)
plot(demoProfiles$chr3.73159773.73160145$H3K4me1, type="l", col="darkgreen",
  xlab="Position", ylab="Coverage in reads per million (RPM)",
  ylim=c(0, 125))
legend("topright", c("H3K27ac","H3K4me1"), cex=1.2,
  col=c("blue","darkgreen"), lty=1)

## Calculate metrics using factory object
```

```
ratio_norm_intersect <- factory$createMetric(metricType =
  "RATIO_NORMALIZED_INTERSECT",
  profile1=demoProfiles$chr3.73159773.73160145$H3K4me1,
  profile2=demoProfiles$chr3.73159773.73160145$H3K27ac)
ratio_norm_intersect

ratio_area <- factory$createMetric(metricType="RATIO_AREA",
  profile1=demoProfiles$chr3.73159773.73160145$H3K4me1,
  profile2=demoProfiles$chr3.73159773.73160145$H3K27ac)
ratio_area
```

---

RatioArea

*RatioArea class*


---

### Description

An object which is a interface to calculate ratio between the profile area of two profiles.

The RatioArea object is needed to calculate the ratio between the profile area of two profile. A threshold and the two profiles are set during the RatioArea object creation. If different threshold or profiles are needed, the calculateMetric function should be used, with the new profiles and threshold passed as arguments to update those values inside the RatioArea object.

The threshold is the minimum profile area value accepted to calculate the ratio.

### Usage

```
RatioArea
```

### Format

An object of class R6ClassGenerator of length 24.

### Value

The RatioArea\$new function returns a RatioArea object which contains the information about the two profiles and the threshold used to calculate the metric. It can be used, as many times needed, to calculate the specified metric.

### Constructor

Create a RatioArea object.

```
RatioArea$new(profile1, profile2, threshold = 1)
```

The threshold is the minimum profile area value accepted to calculate the ratio.

The RatioArea object inherits those functions:

- getMetric A function that returns the value of the calculated metric
- getInfo A function that returns a description of the metric with the metric value.

- `getType` A function that returns the unique name associated to this metric
- `calculateMetric` A function that modifies the values of the two profiles and the threshold. The new values (`profile1`, `profile2`, `threshold`) are passed as arguments.

**Author(s)**

Astrid Deschenes

**See Also**

- [MetricFactory](#) for using a interface to calculate all available metrics separately or together.

---

|                 |  |
|-----------------|--|
| ratioAreaMethod | <i>Ratio of profiles area between two profiles</i> |
|-----------------|--|

---

**Description**

Calculate and return the ratio of profiles area between two profiles covering the same range. The area of the first profile is always divided by the area of the second profile. If one area value is inferior to the threshold, the function returns NA.

**Usage**

```
ratioAreaMethod(profile1, profile2, threshold = 1)
```

**Arguments**

|                        |   |
|------------------------|---|
| <code>profile1</code>  | a vector of numeric values, the first profile containing the alignment depth for each position. The <code>profile1</code> and <code>profile2</code> should have the same length.  |
| <code>profile2</code>  | a vector of numeric values, the second profile containing the alignment depth for each position. The <code>profile1</code> and <code>profile2</code> should have the same length. |
| <code>threshold</code> | a numeric, the minimum profile area value accepted to calculate a ratio.  |

**Value**

The calculated ratio or NA if one profile area value is inferior to the threshold.

**Author(s)**

Astrid Deschenes, Elsa Bernatchez

**See Also**

- [MetricFactory](#) for using the recommended interface to calculate all available metrics separately or together.

---

RatioIntersect

*RatioIntersect class*

---

## Description

An object which is a interface to calculate the ratio between the peaks values between two profiles. The RatioIntersect object is needed to calculate the ratio of profiles intersection area between two profiles and those profiles total areas. A threshold and the two profiles are set during the RatioIntersect object creation. If different threshold or profiles are needed, the calculateMetric function should be used, with the new profiles and threshold passed as arguments to update those values inside the RatioIntersect object.

The threshold is the minimum total area value accepted to calculate a ratio.

## Usage

RatioIntersect

## Format

An object of class R6ClassGenerator of length 24.

## Value

The RatioIntersect\$new function returns a RatioIntersect object which contains the information about the two profiles and the threshold used to calculate the metric. It can be used, as many times needed, to calculate the specified metric.

## Constructor

Create a RatioIntersect object.

```
RatioIntersect$new(profile1, profile2, threshold = 1)
```

The threshold is the minimum total area value accepted to calculate a ratio.

The RatioIntersect object inherits those functions:

- `getMetric` A function that returns the value of the calculated metric
- `getInfo` A function that returns a description of the metric with the metric value.
- `getType` A function that returns the unique name associated to this metric
- `calculateMetric` A function that modifies the values of the two profiles and the threshold. The new values (`profile1`, `profile2`, `threshold`) are passed as arguments.

## Author(s)

Astrid Deschenes

## See Also

- [MetricFactory](#) for using a interface to calculate all available metrics separately or together.

---

ratioIntersectMethod     *Ratio between the intersection area of two profiles and the total area covered by those profiles*

---

### Description

Calculate and return the ratio between the intersection area of two profiles and the total area covered by those profiles. If the total area is inferior to the threshold, the function returns NA.

### Usage

```
ratioIntersectMethod(profile1, profile2, threshold = 1)
```

### Arguments

|           |   |
|-----------|---|
| profile1  | a vector of numeric values, the first profile containing the alignment depth for each position. The profile1 and profile2 should have the same length.  |
| profile2  | a vector of numeric values, the second profile containing the alignment depth for each position. The profile1 and profile2 should have the same length. |
| threshold | a numeric, the minimum total area value accepted to calculate a ratio.  |

### Value

The calculated ratio or NA if the total area is inferior to the threshold.

### Author(s)

Astrid Deschenes, Elsa Bernatchez

### See Also

- [MetricFactory](#) for using the recommended interface to calculate all available metrics separately or together.

---

RatioMaxMax     *RatioMaxMax class*

---

### Description

An object which is a interface to calculate the ratio between the peaks values between two profiles. The RatioMaxMax object is needed to calculate the ratio between the peaks values between two profiles. A threshold and the two profiles are set during the RatioMaxMax object creation. If different threshold or profiles are needed, the calculateMetric function should be used, with the new profiles and threshold passed as arguments to update those values inside the RatioMaxMax object. The threshold is the minimum peak value accepted to calculate a ratio.

**Usage**

RatioMaxMax

**Format**

An object of class R6ClassGenerator of length 24.

**Value**

The RatioMaxMax\$new function returns a RatioMaxMax object which contains the information about the two profiles and the threshold used to calculate the metric. It can be used, as many times needed, to calculate the specified metric.

**Constructor**

Create a RatioMaxMax object.

```
RatioMaxMax$new(profile1, profile2, threshold = 1)
```

The threshold is the minimum peak value accepted to calculate a ratio.

The RatioMaxMax object inherits those functions:

- `getMetric` A function that returns the value of the calculated metric
- `getInfo` A function that returns a description of the metric with the metric value.
- `getType` A function that returns the unique name associated to this metric
- `calculateMetric` A function that modifies the values of the two profiles and the threshold. The new values (`profile1`, `profile2`, `threshold`) are passed as arguments.

**Author(s)**

Astrid Deschenes

**See Also**

- [MetricFactory](#) for using a interface to calculate all available metrics separately or together.

---

ratioMaxMaxMethod

*Ratio of profiles maximal peaks between two profiles*

---

**Description**

Calculate and return the ratio of profiles maximal peaks between two profiles covering the same range. The maximal peak of the first profile is always divided by the maximal peak of the second profile. If one peak value is inferior to the threshold, the function returns NA.

**Usage**

```
ratioMaxMaxMethod(profile1, profile2, threshold = 1)
```



**Arguments**

|           |   |
|-----------|---|
| profile1  | a vector of numeric values, the first profile containing the alignment depth for each position. The profile1 and profile2 should have the same length.  |
| profile2  | a vector of numeric values, the second profile containing the alignment depth for each position. The profile1 and profile2 should have the same length. |
| threshold | a numeric, the minimum peak value accepted to calculate a ratio.  |

**Value**

The calculated ratio or NA if one peak value is inferior to the threshold.

**Author(s)**

Astrid Deschenes, Elsa Bernatchez

**See Also**

- [MetricFactory](#) for using the recommended interface to calculate all available metrics separately or together.

---

RatioNormalizedIntersect

*RatioNormalizedIntersect class*

---

**Description**

An object which is a interface to calculate the ratio of profiles intersection area between two normalized profiles. The profiles are

The RatioNormalizedIntersect object is needed to calculate the ratio of profiles intersection area between two normalized profiles. A threshold and the two profiles are set during the RatioNormalizedIntersect object creation. If different profiles are needed, the calculateMetric function should be used, with the new profiles passed as arguments to update those values inside the RatioNormalizedIntersect object.

The threshold is the minimum total normalized area value accepted to calculate a ratio.

**Usage**

```
RatioNormalizedIntersect
```

**Format**

An object of class R6ClassGenerator of length 24.

**Value**

The `RatioNormalizedIntersect$new` function returns a `RatioNormalizedIntersect` object which contains the information about the two profiles. It can be used, as many times needed, to calculate the specified metric.

**Constructor**

Create a `RatioNormalizedIntersect` object.

```
RatioNormalizedIntersect$new(profile1, profile2, threshold = 1)
```

The threshold is the minimum total normalized area value accepted to calculate a ratio. Default = 1

The `RatioNormalizedIntersect` object inherits those functions:

- `getMetric` A function that returns the value of the calculated metric
- `getInfo` A function that returns a description of the metric with the metric value.
- `getType` A function that returns the unique name associated to this metric
- `calculateMetric` A function that modifies the values of the two profiles and the threshold. The new values (`profile1`, `profile2`, `threshold`) are passed as arguments.

**Author(s)**

Astrid Deschenes

**See Also**

- [MetricFactory](#) for using a interface to calculate all available metrics separately or together.

---

similarity

*Calculate metrics which estimate the level of similarity between two ChIP-Seq profiles*

---

**Description**

It returns a list containing information about both ChIP-Seq profiles and a list of all similarity metrics: the ratio of the maximum values, the ratio of the areas, the ratio between the intersection area and the total area (for normalized and non-normalized profiles), the difference between two profiles maximal peaks positions and the Spearman's rho statistic.

**Usage**

```
similarity(profile1, profile2, ratioAreaThreshold = 1,
  ratioMaxMaxThreshold = 1, ratioIntersectThreshold = 1,
  ratioNormalizedIntersectThreshold = 1, diffPosMaxThresholdMinValue = 1,
  diffPosMaxThresholdMaxDiff = 100, diffPosMaxTolerance = 0.01,
  spearmanCorrSDThreshold = 1e-08)
```

**Arguments**

|                                   |  |
|-----------------------------------|--|
| profile1                          | Vector containing the RPM values of the first ChIP-Seq profile for each position of the selected region.   |
| profile2                          | Vector containing the RPM values of the second ChIP-Seq profile for each position of the selected region.  |
| ratioAreaThreshold                | The minimum denominator accepted to calculate the ratio of the area between both profiles. The value has to be positive. Default = 1.  |
| ratioMaxMaxThreshold              | The minimum denominator accepted to calculate the ratio of the maximal peaks values between both profiles. The value has to be positive. Default = 1.                        |
| ratioIntersectThreshold           | The minimum denominator accepted to calculate the ratio of the intersection area of both profiles over the total area. The value has to be positive. Default = 1.            |
| ratioNormalizedIntersectThreshold | The minimum denominator accepted to calculate the ratio of the intersection area of both normalized profiles over the total area. The value has to be positive. Default = 1. |
| diffPosMaxThresholdMinValue       | The minimum peak accepted to calculate the metric. The value has to be positive. Default = 1.  |
| diffPosMaxThresholdMaxDiff        | The maximum distance accepted between 2 peaks positions in one profile to calculate the metric. The value has to be positive. Default=100.                                   |
| diffPosMaxTolerance               | The maximum of variation accepted on the maximum value to consider a position as a peak position. The value can be between 0 and 1. Default=0.01.                            |
| spearmanCorrSDThreshold           | The minimum standard deviation accepted on both profiles to calculate the metric. Default=1e-8.  |

**Details**

similarity uses the two vectors passed as arguments to calculate the metrics. When the metric is a ratio, it always verify that the threshold for the denominator is respected. If the threshold is not respected, the metric is assigned the NA value.

**Value**

A list containing :

- nbrPosition The number of positions included in each profile.
- areaProfile1 The area of the first profile.
- areaProfile2 The area of the second profile.
- maxProfile1 The maximum value in the first profile.

- `maxProfile2` The maximum value in the second profile.
- `maxPositionProfile1` The list of positions of the maximum value in the first profile.
- `maxPositionProfile2` The list of positions of the maximum value in the second profile.
- `metrics` A list with the following items:
  - `RATIO_AREA` The ratio between the areas. The larger value is always divided by the smaller value. NA if minimal threshold is not respected.
  - `DIFF_POS_MAX` The difference between the maximal peaks positions. The difference is always the first profile value minus the second profile value. NA is returned if minimal peak value is not respected. A profile can have more than one position with the maximum value. In that case, the median position is used. A threshold argument can be set to consider all positions within a certain range of the maximum value. A threshold argument can also be set to ensure that the distance between two maximum values is not too wide. When this distance is not respected, it is assumed that more than one peak is present in the profile and NA is returned.
  - `RATIO_MAX_MAX` The ratio between the maximal peaks values. The first profile is always divided by the second profile. NA if minimal threshold is not respected.
  - `RATIO_INTERSECT` The ratio between the intersection area and the total area. NA if minimal threshold is not respected.
  - `RATIO_NORMALIZED_INTERSECT` The ratio between the intersection area and the total area of normalized profiles. NA if minimal threshold is not respected.
  - `SPEARMAN_CORRELATION` The Spearman's rho statistic between profiles. NA if minimal threshold is not respected or when no complete element pair is present between both profiles.

### Author(s)

Astrid Deschenes, Elsa Bernatchez

### See Also

- [MetricFactory](#) for using a interface to calculate all available metrics separately or together.
- [demoProfiles](#) for more informations about ChIP-Seq profiles present in the `demoProfiles` data.

### Examples

```
## Defining two ChIP-Seq profiles
profile1<-c(3,59,6,24,65,34,15,4,53,22,21,12,11)
profile2<-c(15,9,46,44,9,39,27,34,34,4,3,4,2)

## Example using default thresholds
similarity(profile1, profile2)

## Example using customised thresholds
similarity(profile1, profile2,
  ratioAreaThreshold=5,
  ratioMaxMaxThreshold=5,
  ratioIntersectThreshold=12,
```

```

    ratioNormalizedIntersectThreshold=2.2,
    diffPosMaxThresholdMinValue=2,
    diffPosMaxThresholdMaxDiff=130,
    diffPosMaxTolerance=0.03,
    spearmanCorrSDThreshold=1e-3)

## Example using ChIP-Seq profiles of H3K27ac (DCC accession: ENCF000ASG)
## and H3K4me1 (DCC accession: ENCF000ARY) from the Encyclopedia of DNA
## Elements (ENCODE) for the region
data(demoProfiles)

## Visualize ChIP-Seq profiles
plot(demoProfiles$chr2.70360770.70361098$H3K27ac,
     type="l", col="blue", xlab="", ylab="", ylim=c(0, 25),
     main="chr2:70360770-70361098")
par(new=TRUE)
plot(demoProfiles$chr2.70360770.70361098$H3K4me1,
     type="l", col="darkgreen", xlab="Position",
     ylab="Coverage in reads per million (RPM)", ylim=c(0, 25))
legend("topright", c("H3K27ac","H3K4me1"), cex=1.2,
     col=c("blue","darkgreen"), lty=1)

# Calculate metrics
similarity(demoProfiles$chr2.70360770.70361098$H3K4me1,
           demoProfiles$chr2.70360770.70361098$H3K27ac,
           ratioAreaThreshold=15,
           ratioMaxMaxThreshold=5,
           ratioIntersectThreshold=12,
           ratioNormalizedIntersectThreshold=2.2,
           diffPosMaxThresholdMinValue=2,
           diffPosMaxThresholdMaxDiff=130,
           diffPosMaxTolerance=0.03,
           spearmanCorrSDThreshold=0.1)

## You can refer to the vignette to see more examples using ChIP-Seq profiles
## extracted from the Encyclopedia of DNA Elements (ENCODE) data.

```

---

SpearmanCorrelation    *SpearmanCorrelation class*

---

## Description

An object which is a interface to calculate the Spearman's

The SpearmanCorrelation object is needed to calculate the ratio between the peaks values between two profiles. A threshold and the two profiles are set during the SpearmanCorrelation object creation. If different threshold or profiles are needed, the calculateMetric function should be used, with the new profiles and threshold passed as arguments to update those values inside the SpearmanCorrelation object.

The threshold is the minimum standard deviation of the profile accepted to calculate a ratio.

**Usage**

SpearmanCorrelation

**Format**

An object of class R6ClassGenerator of length 24.

**Value**

The `SpearmanCorrelation$new` function returns a `SpearmanCorrelation` object which contains the information about the two profiles. It can be used, as many times needed, to calculate the specified metric.

**Constructor**

Create a `SpearmanCorrelation` object.

```
SpearmanCorrelation$new(profile1, profile2, threshold = 1e-8)
```

The threshold is the minimum standard deviation of the profile accepted to calculate a ratio. Default = 1e-8

The `SpearmanCorrelation` object inherits those functions:

- `getMetric` A function that returns the value of the calculated metric
- `getInfo` A function that returns a description of the metric with the metric value.
- `getType` A function that returns the unique name associated to this metric
- `calculateMetric` A function that modifies the values of the two profiles and the threshold. The new values (`profile1`, `profile2`) are passed as arguments.

**Author(s)**

Astrid Deschenes

**See Also**

- [MetricFactory](#) for using a interface to calculate all available metrics separately or together.

---

spearmanCorrMethod      *Spearman's rho statistic of two profiles*

---

**Description**

Calculate and return the Spearman's rho statistic of two profiles. If one profile has a standard deviation inferior to the threshold, the function returns NA. When no complete element pair are present, NA is returned.

**Usage**

```
spearmanCorrMethod(profile1, profile2, threshold = 1e-08)
```

**Arguments**

|           |   |
|-----------|---|
| profile1  | a vector of numeric values, the first profile containing the alignment depth for each position. The profile1 and profile2 should have the same length.  |
| profile2  | a vector of numeric values, the second profile containing the alignment depth for each position. The profile1 and profile2 should have the same length. |
| threshold | a numeric, the minimum standard deviation accepted to calculate a ratio. Default = 1e-8   |

**Value**

The calculated ratio or NA if one profile has a standard deviation inferior to the threshold. If profiles have no complete element pair, NA is returned.

**Author(s)**

Astrid Deschenes, Elsa Bernatchez

**See Also**

- [MetricFactory](#) for using the recommended interface to calculate all available metrics separately or together.

# Index

## \* datasets

- chr7Profiles, [3](#)
- demoProfiles, [5](#)
- DiffPosMax, [6](#)
- Metric, [9](#)
- MetricFactory, [10](#)
- RatioArea, [12](#)
- RatioIntersect, [14](#)
- RatioMaxMax, [15](#)
- RatioNormalizedIntersect, [17](#)
- SpearmanCorrelation, [21](#)

## \* package

- similaRpeak-package, [2](#)

chr7Profiles, [3](#), [6](#)

demoProfiles, [4](#), [5](#), [11](#), [20](#)

DiffPosMax, [6](#)

diffPosMaxMethod, [8](#)

Metric, [9](#)

MetricFactory, [3](#), [4](#), [6–9](#), [10](#), [13–18](#), [20](#), [22](#),  
[23](#)

RatioArea, [12](#)

ratioAreaMethod, [13](#)

RatioIntersect, [14](#)

ratioIntersectMethod, [15](#)

RatioMaxMax, [15](#)

ratioMaxMaxMethod, [16](#)

RatioNormalizedIntersect, [17](#)

similarity, [2–4](#), [6](#), [11](#), [18](#)

similaRpeak (similaRpeak-package), [2](#)

similaRpeak-package, [2](#)

SpearmanCorrelation, [21](#)

spearmanCorrMethod, [22](#)