

Package ‘colocalization’

May 17, 2018

Type Package

Title Normalized Spatial Intensity Correlation

Version 1.0.0

Author Jiahui Xu, Xueyan Liu, Cheng Cheng, Hui Zhang

Maintainer Hui Zhang <hui.zhang@stjude.org>

Depends R (>= 2.10)

Description Calculate the colocalization index, NSInC, in two different ways as described in the paper (Liu et al., 2018. Manuscript submitted for publication.) for multiple-species spatial data which contain the precise locations and membership of each spatial point. The two main functions are `nsinc.d()` and `nsinc.z()`. They provide the Pearson’s correlation coefficients of signal proportions in different memberships within a concerned proximity of every signal (or every base signal if single direction colocalization is considered) across all (base) signals using two different ways of normalization. The proximity sizes could be an individual value or a range of values, where the default ranges of values are different for the two functions.

License GPL-3

Encoding UTF-8

Suggests testthat, rgl

Imports utils, stats, ggplot2

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-17 08:11:19 UTC

R topics documented:

<code>nsinc.d</code>	2
<code>nsinc.z</code>	5
<code>plot.colocal</code>	9
<code>summary.colocal</code>	10
<code>twolines</code>	12

Index	13
--------------	-----------

 nsinc.d

Colocalization index of d-type

Description

nsinc.d is used to calculate the Pearson's correlation coefficient of the average proportion densities with complete spatial randomness (CSR) as reference of two types of signals in a specified proximity of all signals or all signals of interested type (or base signals) as the colocalization index for a whole image. If a range of proximity sizes are concerned, the nsinc.d will take the average of the index values over the range. In the case of multiple-species data, the average of index values of all pairs at each proximity size is taken as the index for the image at that size of neighborhood.

Usage

```
nsinc.d(data, membership, dim = 2, r.min = NULL,
        r.max = NULL, r.count = NULL, r.adjust = NULL,
        box = NULL, edge.effect = TRUE, strata = FALSE,
        base.member = NULL, r.model = "full", ...)
```

Arguments

data	a data frame (or object coercible by <code>as.data.frame</code> to a data frame) containing at least the columns <code>membership</code> and <code>x</code> (<code>xc</code> , <code>X</code> or <code>Xc</code>), <code>y</code> (<code>yc</code> , <code>Y</code> or <code>Yc</code>) if <code>dim = 2</code> and <code>x</code> (<code>xc</code> , <code>X</code> or <code>Xc</code>), <code>y</code> (<code>yc</code> , <code>Y</code> or <code>Yc</code>), <code>z</code> (<code>zc</code> , <code>Z</code> or <code>Zc</code>) if <code>dim = 3</code> .
membership	a string describing the column name in the data representing the membership of data points. There should be no less than 2 levels in the membership.
dim	an integer either = 2 or = 3. If <code>dim = 2</code> , the data are treated as two-dimensional; if <code>dim = 3</code> , the data are treated as three-dimensional.
r.min	the minimum proximity size that the user identifies as colocalization of signals. It should be numeric. If <code>r.model = "full"</code> , the function will automatically choose the smallest inter-point distance as the <code>r.min</code> ; if <code>r.model = "r.med"</code> , the function will use the median inter-point distance for both <code>r.min</code> and <code>r.max</code> ; if <code>r.model = "other"</code> , the user must specify <code>r.min</code> , which should be no larger than <code>r.max</code> .
r.max	the maximum proximity size that the user identifies as colocalization of signals. It should be numeric. If <code>r.model = "full"</code> , the function will automatically choose the largest inter-point distance as the <code>r.max</code> ; if <code>r.model = "r.med"</code> , the function will use the median inter-point distance for both <code>r.min</code> and <code>r.max</code> ; if <code>r.model = "other"</code> , the user must specify <code>r.max</code> , which should be between the smallest and the largest inter-point distances and no smaller than <code>r.min</code> .
r.count	the total count of the series of proximity sizes between <code>r.min</code> and <code>r.max</code> . If <code>r.max = r.min</code> or <code>r.adjust = (r.max - r.min)/2</code> , then <code>r.count = 1</code> , otherwise <code>r.count = 30</code> by default or is specified by the user.

<code>r.adjust</code>	a small adjustment for <code>r.min</code> and <code>r.max</code> to get the series of proximity sizes between <code>r.min + r.adjust</code> and <code>r.max - r.adjust</code> to avoid zero standard deviation of average proportion densities at extremely small and large <code>r</code> 's. The values of <code>r.adjust</code> depends on the choice of <code>r.model</code> and values of <code>r.min</code> and <code>r.max</code> . For most scenarios, it is suggested to use <code>r.adjust = NULL</code> and let the function choose the default value for <code>r.adjust</code> . In general, by default either <code>r.adjust = 0</code> or <code>r.adjust = (r.max - r.min)/(r.count + 1)</code> ; otherwise it is a positive number specified by the user satisfying <code>r.adjust ≤ (r.max - r.min)/2</code> .
<code>box</code>	a one-row data frame describing the study region which must contain columns <code>xmin</code> , <code>xmax</code> , <code>ymin</code> , <code>ymax</code> if <code>dim = 2</code> and additionally <code>zmin</code> , <code>zmax</code> if <code>dim = 3</code> . If <code>box = NULL</code> , the function will detect the smallest box containing all data points and add a buffer edge in each dimension which is equal to the median of nearest neighbor distances in that dimension. If <code>box</code> is specified by the user, only the data enclosed in the specified box will be considered in the analysis and signals outside the box will be ignored.
<code>edge.effect</code>	a logical value showing whether the edge effect should be corrected. By default it should be corrected otherwise the results are not accurate.
<code>strata</code>	a logical value showing whether the user wants to consider single-direction or bi-direction colocalization. By default <code>strata = FALSE</code> is for bi-direction colocalization. In this case, all proximity regions around all signals are considered. If <code>strata = TRUE</code> , then <code>base.member</code> must be specified or the first membership that R detects in the membership column will be used by default and only the circular regions around signals in the base membership are considered. Then, colocalization will be single-direction in this case.
<code>base.member</code>	one level of the memberships that is designated as the base. It works only when <code>strata = TRUE</code> . If <code>strata = TRUE</code> and no <code>base.member</code> is specified by the user, the first membership that R detects in the membership column will be used by default for <code>base.member</code> .
<code>r.model</code>	equals either "full", "r.med" or "other". The <code>r.model</code> will be used to choose the proximity size ranges that the user defines for colocalization. "full" or "r.med" can be used if the user has no specific sense of proximity sizes for colocalization. In "full" model, the colocalization proximity sizes will range from the smallest inter-point distance to the largest inter-point distances; in "r.med" model, the fixed proximity size is the median of inter-point distances; in "other" model, the user can define their research driven proximity sizes by specifying <code>r.min</code> and <code>r.max</code> .
<code>...</code>	Parameters passed to <code>cor</code> . The user could choose methods other than Pearson for calculating correlation.

Details

The function calculates the average proportion density with CSR as reference of two types of signals in a specified `r` neighborhood with edge effect corrected of all signals or all base signals if `strata = TRUE` is specified, then obtains the Pearson correlation coefficients of each pair of channels and average them among all pairs at each `r` in the `r` series from `r.min` to `r.max`. In the case of multiple-species data, the average of index values of all pairs at each proximity size is taken as

the index for the image at that size of neighborhood. The index for the whole image is named as $NSInC^d$ or NSInC of type d. The index will be close to 1 if signals are colocalized, 0 if random and -1 if dispersed. The function can deal with 2D or 3D data.

If the users have their specific proximity size, then they are encouraged to specify `r.model = "other"`, and same values for `r.min` and `r.max`.

Value

nsinc.d returns all colocalization index at each separate proximity size r, and the average colocalization index across all r's, the data that the colocalization index is calculated from, the study region, i.e., the carrying box, the original and normalized proportions of each type of signals in an r neighborhood of all (base) signals, the r series, and some summary information:

method	"nsinc.d"
input.data.summary	a list containing the number of membership levels and the signal counts in each channel or membership of the input data.
post.data.summary	a list showing the number of membership levels and signal counts in each channel of the data after removal of signals located outside the specified box by the user. If there is no signals excluded, then post.data.summary presents the same results as input.data.summary.
r.summary	a data frame listing the r.min, r.max, r.count, r.adjust used in the calculation and the r.model specified by the user or the default. r.summary also gives the r range for the default "full" model, i.e., the minimum and maximum of the inter-point distance of all signals, and the median value in addition.
strata	a list showing the default setting of strata or the specified strata by the user. It also presents the base membership used in the function if strata is TRUE.
edge.effect	a data frame containing a logical value indicating whether edge effect is corrected or not.
index.all	a data frame showing the colocalization index of d-type at each r.
index	the averaged colocalization index of d-type across all r's.
post.data	a data frame representing the data after removal of signals located outside the specified box by the user. If there is no signal excluded, then post.data presents the same observations as data.
study.region	the carrying box with the size of buffer width in each dimension.
P.all	the data frame showing all original and normalized proportions of each type of signals in an r-neighborhood around every (base) signal. Rows are (base) signals and columns are all memberships and r's.
r	the r series for which the colocalization indices are calculated.

Author(s)

Xueyan Liu, Jiahui Xu, Cheng Cheng, Hui Zhang.

References

Liu, X., Xu, J., Guy C., Romero E., Green D., Cheng, C., Zhang, H. (2018). *A novel statistical tool for analyzing colocalization in super-resolution images*. Manuscript submitted for publication.

Examples

```
## A simulated 2D example data.
set.seed(1234)
x <- runif(300, min = -1, max = 1)
y <- runif(300, min = -1, max = 1)
red <- data.frame(x,y, color = "red")
x <- runif(50, min = -1, max = 1)
y <- runif(50, min = -1, max = 1)
green <- data.frame(x,y, color = "green")

mydata <- rbind(red,green)
plot(mydata$x,mydata$y,col = mydata$color)

mydata.results <- nsinc.d(data = mydata, membership = "color", dim = 2)

mydata.results$index.all
mydata.results$index

## A simulated 3D example data.
data("twolines")

library("rgl")
plot3d(twolines[,c("x","y","z")], type='s', size=0.7, col = twolines$membership)
aspect3d("iso")

twolines.results <- nsinc.d(data = twolines, membership = "membership",
                           dim = 3, r.model = "r.med")

twolines.results$index
```

 nsinc.z

Colocalization index of z-type

Description

nsinc.z is used to calculate the Pearson's correlation coefficient of the signal proportions of two channels with a z-score normalization based on complete spatial randomness (CSR) in a specified proximity of all signals or all signals of interested type as the colocalization index for a whole image. If a range of proximity sizes are concerned, the nsinc.z will take the average of the index values over the range. In the case of multiple-species data, the average of index values of all pairs at each proximity size is taken as the index for the image at that size of neighborhood.

Usage

```
nsinc.z(data, membership, dim = 2, r.min = NULL,
        r.max = NULL, r.count = NULL, r.adjust = NULL,
        box = NULL, edge.effect = TRUE, strata = FALSE,
        base.member = NULL, r.model = "full", ...)
```

Arguments

<code>data</code>	a data frame (or object coercible by <code>as.data.frame</code> to a data frame) containing at least the columns <code>membership</code> and <code>x</code> (<code>xc</code> , <code>X</code> or <code>Xc</code>), <code>y</code> (<code>yc</code> , <code>Y</code> or <code>Yc</code>) if <code>dim = 2</code> and <code>x</code> (<code>xc</code> , <code>X</code> or <code>Xc</code>), <code>y</code> (<code>yc</code> , <code>Y</code> or <code>Yc</code>), <code>z</code> (<code>zc</code> , <code>Z</code> or <code>Zc</code>) if <code>dim = 3</code> .
<code>membership</code>	a string describing the column name in the data representing the membership of data points. There should be no less than 2 levels in the membership.
<code>dim</code>	an integer either = 2 or = 3. If <code>dim = 2</code> , the data are treated as two-dimensional; if <code>dim = 3</code> , the data are treated as three-dimensional.
<code>r.min</code>	the minimum proximity size that the user identifies as colocalization of signals. It should be numeric. If <code>r.model = "full"</code> , the function will automatically choose the smallest inter-point distance as the <code>r.min</code> ; if <code>r.model = "r.med"</code> , the function will use the median inter-point distance for both <code>r.min</code> and <code>r.max</code> ; if <code>r.model = "other"</code> , the user must specify <code>r.min</code> , which should be no larger than <code>r.max</code> .
<code>r.max</code>	the maximum proximity size that the user identifies as colocalization of signals. It should be numeric. If <code>r.model = "full"</code> , the function will automatically choose half of the largest inter-point distance as the <code>r.max</code> ; if <code>r.model = "r.med"</code> , the function will use the median inter-point distance for both <code>r.min</code> and <code>r.max</code> ; if <code>r.model = "other"</code> , the user must specify <code>r.max</code> , which should be between the smallest and the largest inter-point distances and no smaller than <code>r.min</code> .
<code>r.count</code>	the total count of the series of proximity sizes between <code>r.min</code> and <code>r.max</code> . If <code>r.max = r.min</code> or <code>r.adjust = (r.max - r.min)/2</code> , then <code>r.count = 1</code> , otherwise <code>r.count = 30</code> by default or is specified by the user.
<code>r.adjust</code>	a very small adjustment for <code>r.min</code> and <code>r.max</code> to get the series of proximity sizes between <code>r.min + r.adjust</code> and <code>r.max - r.adjust</code> to avoid zero standard deviation of normalized proportions of signals at extremely small and large <code>r</code> 's. The values of <code>r.adjust</code> depends on the choice of <code>r.model</code> and values of <code>r.min</code> and <code>r.max</code> . For most scenarios, it is suggested to use <code>r.adjust = NULL</code> and let the function choose the default value for <code>r.adjust</code> . In general, by default either <code>r.adjust = 0</code> or <code>r.adjust = (r.max - r.min)/(r.count + 1)</code> ; otherwise it is a positive number specified by the user satisfying <code>r.adjust ≤ (r.max - r.min)/2</code> .
<code>box</code>	a one-row data frame describing the study region which must contain columns <code>xmin</code> , <code>xmax</code> , <code>ymin</code> , <code>ymax</code> if <code>dim = 2</code> and additionally <code>zmin</code> , <code>zmax</code> if <code>dim = 3</code> . If <code>box = NULL</code> , the function will detect the smallest box containing all data points and add a buffer edge in each dimension which is equal to the median of nearest neighbor distances in that dimension. If <code>box</code> is specified by the user, only the data enclosed in the specified box will be considered in the analysis and signals outside the box will be ignored.

<code>edge.effect</code>	a logical value showing whether the edge effect should be corrected. By default it should be corrected otherwise the results are not accurate.
<code>strata</code>	a logical value showing whether the user wants to consider single-direction or bi-direction colocalization. By default <code>strata = FALSE</code> is for bi-direction colocalization. In this case, all proximity regions around all signals are considered. If <code>strata = TRUE</code> , then <code>base.member</code> must be specified or the first membership that R detects in the membership column will be used by default and only the circular regions around signals in the base membership are considered. Then, colocalization will be single-direction in this case.
<code>base.member</code>	one level of the memberships that is designated as the base. It works only when <code>strata = TRUE</code> . If <code>strata = TRUE</code> and no <code>base.member</code> is specified by the user, the first membership that R detects in the membership column will be used by default for <code>base.member</code> .
<code>r.model</code>	equals either "full", "r.med" or "other". The <code>r.model</code> will be used to choose the proximity size ranges that the user defines for colocalization. "full" or "r.med" can be used if the user has no specific sense of proximity sizes for colocalization. In "full" model, the colocalization proximity sizes will range from the smallest inter-point distance to half of the largest inter-point distances; in "r.med" model, the fixed proximity size is the median of inter-point distances; in "other" model, the user can define their research driven proximity sizes by specifying <code>r.min</code> and <code>r.max</code> .
...	Parameters passed to <code>cor</code> . The user could choose methods other than Pearson for calculating correlation.

Details

The function calculates the proportion of two types of signals normalized by a z-score under CSR in a specified `r` neighborhood with edge effect corrected of all signals or all base signals if `strata = TRUE` is specified, then obtains the Pearson correlation coefficients of each pair of channels and average them among all pairs at each `r` in the `r` series between `r.min` to `r.max`. In the case of multiple-species data, the average of index values of all pairs at each proximity size is taken as the index for the image at that size of neighborhood. The index for the whole image is named as $NSInC^z$ or $NSInC$ of type `z`. The index will be close to 1 if signals are colocalized, 0 if random and -1 if dispersed. The function can deal with 2D or 3D data.

If the users have their specific proximity size, then they are encouraged to specify `r.model = "other"`, and values of `r.min` and `r.max`.

The difference from `nsinc.d` is the normalization of the signal proportions. The z-type normalization has no heterogeneity under CSR caused by the edge effects related to the locations of signals. In many cases, `nsinc.d` and `nsinc.z` can give similar results. However, if the user's proximity of interest is larger than half of the largest inter-point distances, then `nsinc.d` is suggested.

Value

`nsinc.z` returns colocalization index values at each separate proximity size `r`, and the average colocalization index across all `r`'s, the data that the colocalization index is calculated from, the study region, i.e., the carrying box, the original and normalized proportions of each type of signals in an `r` neighborhood of all (base) signals, the `r` series, and some summary information:

method	"nsinc.z"
input.data.summary	a list containing the number of membership levels and the signal counts in each channel or membership of the input data.
post.data.summary	a list showing the number of membership levels and signal counts in each channel of the data after removal of signals located outside the specified box by the user. If there is no signals excluded, then post.data.summary presents the same results as input.data.summary.
r.summary	a data frame listing the r.min, r.max, r.count, r.adjust used in the calculation and the r.model specified by the user or the default. r.summary also gives the r range for the default full model, i.e., the minimum and half of the maximum of the inter-point distance of all signals, and the median value in addition.
strata	a list showing the default setting of strata or the specified strata by the user. It also presents the base membership used in the function if strata is TRUE.
edge.effect	a data frame containing a logical value indicating whether edge effect is corrected or not.
index.all	a data frame showing the colocalization index of z-type at each r.
index	the averaged colocalization index of z-type across all r's.
post.data	a data frame representing the data after removal of signals located outside the specified box by the user. If there is no signal excluded, then post.data presents the same observations as data.
study.region	the carrying box with the size of buffer width in each dimension.
P.all	the data frame showing all original and normalized proportions of each type of signals in an r-neighborhood around every (base) signal. Rows are (base) signals and columns are all memberships and r's.
r	the r series for which the colocalization indices are calculated.

Author(s)

Xueyan Liu, Jiahui Xu, Cheng Cheng, Hui Zhang.

References

Liu, X., Xu, J., Guy C., Romero E., Green D., Cheng, C., Zhang, H. (2018). *A novel statistical tool for analyzing colocalization in super-resolution images*. Manuscript submitted for publication.

Examples

```
## a simulated 2D example data.
set.seed(1234)
x <- runif(300, min = -1, max = 1)
y <- runif(300, min = -1, max = 1)
red <- data.frame(x,y, color = "red")
x <- runif(50, min = -1, max = 1)
y <- runif(50, min = -1, max = 1)
green <- data.frame(x,y, color = "green")
```



```

mydata <- rbind(red,green)
plot(mydata$x,mydata$y,col = mydata$color)

mydata.results <- nsinc.z(data = mydata, membership = "color", dim = 2,
                          r.model = "other", r.min = 0.01, r.max = 0.5, r.count = 5, r.adjust = 0)

mydata.results$index.all
mydata.results$index

## a simulated 3D example data.
data("twolines")

library("rgl")
plot3d(twolines[,c("x","y","z")], type='s', size=0.7, col = twolines$membership)
aspect3d("iso")

twolines.results <- nsinc.z(data = twolines, membership = "membership",
                            dim = 3, r.model = "full")

twolines.results$index

```

plot.colocal	<i>Making scatter plots for signal proportions before and after d-type or z-type normalization</i>
--------------	--

Description

This function is used to make scatter plots for signal proportions based on the results returned from the `nsinc.d` or `nsinc.z` function.

Usage

```

## S3 method for class 'colocal'
plot(x, ...)

```

Arguments

x	an object of class "colocal", containing the results returned from <code>nsinc.d</code> or <code>nsinc.z</code> .
...	further arguments to be passed from or to other methods.

Details

The function currently works for results from `nsinc.d` or `nsinc.z` with bi-direction colocalization for dual-color images. At each proximity size `r`, the function makes two panels of scatter plots for signal proportions at all signals before and after d-type or z-type normalizations. Each signal in the original image contributes a point in the scatter plots whose x coordinate is the proportion of signals in one channel and y coordinate is the other channel. The scatter plots have the same color codes as signals in the original image. If the returned results contain colocalization results at multiple `r`'s, then the scatter plots at each `r` are generated.

Value

`plot.colocal` returns a list of plots which summarize the results returned from the `nsinc.d` or `nsinc.z` function.

Author(s)

Xueyan Liu, Jiahui Xu, Cheng Cheng, Hui Zhang.

References

Liu, X., Xu, J., Guy C., Romero E., Green D., Cheng, C., Zhang, H. (2018). *A novel statistical tool for analyzing colocalization in super-resolution images*. Manuscript submitted for publication.

Examples

```
## a simulated 3D example data.
data("twolines")

twolines.results <- nsinc.d(data = twolines, membership = "membership",
                           dim = 3, r.model = "r.med")

plot(twolines.results)
```

summary.colocal

Summarizing the colocalization results

Description

This function is used to summarize the results returned from the `nsinc.d` or `nsinc.z` function.

Usage

```
## S3 method for class 'colocal'
summary(object, ...)
```

Arguments

object an object of class "colocal", containing the results returned from [nsinc.d](#) or [nsinc.z](#).
... further arguments to be passed from or to other methods.

Details

The results successfully returned from [nsinc.d](#) or [nsinc.z](#) functions give a list of length 12 encompassing the summarized information of the calculation of colocalization index and the detailed quantities used to calculate the index.

First the summary prints the strategic parameters for calculation of colocalization index, such as, the method, the strata, the edge effect, dimension, study region, membership levels and number of the observed signals in each channel.

The summary also prints the summarized information of proximity sizes, i.e., r , including the r model, the r range, the length of r series, etc.

Then the separate index results are listed at each r . The average colocalization index for the whole image is given at last.

Value

`summary.colocal` does not return values. It only prints summarized results returned from [nsinc.d](#) or [nsinc.z](#) functions.

Author(s)

Xueyan Liu, Jiahui Xu, Cheng Cheng, Hui Zhang.

References

Liu, X., Xu, J., Guy C., Romero E., Green D., Cheng, C., Zhang, H. (2018). *A novel statistical tool for analyzing colocalization in super-resolution images*. Manuscript submitted for publication.

See Also

[nsinc.d](#), [nsinc.z](#), [summary](#)

Examples

```
## a simulated 3D example data.  
data("twolines")  
  
twolines.results <- nsinc.d(data = twolines, membership = "membership",  
                          dim = 3, r.model = "r.med")  
  
summary(twolines.results)
```

`twolines`*A simulated 2-lines test data in 3D*

Description

The test data is a simulated 3D dataset of 2-colored lines, i.e., red, green, whose pivots intersect at the origin in the unit box $[-1, 1] \times [-1, 1] \times [-1, 1]$.

Usage

```
data("twolines")
```

Format

A data frame with the precise x,y,z coordinates of 426 signal points with marked colors for their memberships. The x,y,z coordinates are within the interval $[-1,1]$.

x the x coordinate

y the y coordinate

z the z coordinate

membership a factor with levels red and green

Details

The pivots of the red and green lines are respectively, $(t,0,0)$, $(t*\cos(\text{atan}(5)), t*\sin(\text{atan}(5)), 0)$. The number of points along each pivot within the unit box is $\text{Poisson}(200)$. The t values are generated by the uniform distribution. The perturbations for locations of all signals are generated independently in the orthogonal plane away from the pivot with zero mean and $\text{SD} = 0.1$ for the displacements together with the uniformly distributed random rotations.

Source

From Xueyan Liu, Jiahui Xu, Cheng Cheng, Hui Zhang.

References

N/A

Examples

```
data("twolines")
```

```
library(rgl)
plot3d(twolines[,c("x","y","z")], type = 's', size = 0.8, col = twolines$membership)
aspect3d("iso")
```

Index

`nsinc.d`, [2](#), [7](#), [9–11](#)

`nsinc.z`, [5](#), [9–11](#)

`plot.colocal`, [9](#)

`summary`, [11](#)

`summary.colocal`, [10](#)

`twolines`, [12](#)