Package 'spatialTIME'

November 22, 2022

Title Spatial Analysis of Vectra Immunoflourescent Data

Version 1.2.2

Description Visualization and analysis of Vectra Immunoflourescent data. Options for calculating both the univariate and bivariate Ripley's K are included. Calculations are performed using a permutation-based approach presented by Wilson et al. <doi:10.1101/2021.04.27.21256104>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.2

Imports magrittr, dplyr, tidyr, ggplot2, scales, grDevices, purrr, rlang, plyr, spatstat.geom, spatstat.explore, RColorBrewer, furrr, gridExtra, future, tidyselect, crayon, pheatmap

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL https://github.com/FridleyLab/spatialTIME

BugReports https://github.com/FridleyLab/spatialTIME/issues

NeedsCompilation no

Author Jordan Creed [aut],
Ram Thapa [aut],
Christopher Wilson [aut],
Alex Soupir [aut],
Oscar Ospina [aut],
Brooke Fridley [cph],
Fridley Lab [cre]

Maintainer Fridley Lab <fridley.lab@moffitt.org>

Repository CRAN

Date/Publication 2022-11-22 17:30:02 UTC

R topics documented:

`bi_NN_G` *Bivariate Nearest Neighbor Based Measures of Spatial Clustering for IF data*

Description

This function computes the nearest neighbor distribution for a particular marker relative to another marker for the observed and permuted point processes.

Usage

```
bi_NN_G(
  mif,
  mnames,
  r_range = seq(0, 100, 50),
  num_permutations = 50,
  edge_correction = "rs",
  keep_perm_dis = FALSE,
  exhaustive = TRUE,
  workers = 1,
  overwrite = FALSE,
  xloc = NULL,
  yloc = NULL
)
```

Arguments

`mif`	An MIF object
`mnames`	Character vector of marker names to estimate degree of nearest neighbor distribution
`r_range`	Numeric vector of potential r values this range must include 0. Note that the range selected is very different than count based measures. See details.

num_permutations
: Numeric value indicating the number of permutations used. Default is 50.

edge_correction
: Character value indicating the type of edge correction to use. Options include "rs" or "hans".

keep_perm_dis
: Logical value determining whether or not to keep the full distribution of permuted G values

exhaustive
: Logical. If TRUE then markers must be a vector and spatial measures will be computed all pairs of unique markers. If FALSE then markers must be a data.frame with the desired combinations.

workers
: Integer value for the number of workers to spawn

overwrite
: Logical value determining if you want the results to replace the current output (TRUE) or be to be appended (FALSE).

xloc
: a string corresponding to the x coordinates. If null the average of XMin and XMax will be used

yloc
: a string corresponding to the y coordinates. If null the average of YMin and YMax will be used

Value

Returns a data frame

anchor
: Marker for which the distances are measured from

counted
: Marker for which the distances are measured to

Theoretical CSR
: Expected value assuming complete spatial randomness

Permuted CSR
: Average observed G for the permuted point process

Observed
: Observed value for the observed point process

Degree of Clustering Permuted
: Degree of spatial clustering where the reference is the permuted estimate of CSR

Degree of Clustering Theoretical
: Degree of spatial clustering where the reference is the theoretical estimate of CSR

Examples

```
#' #Create mif object
library(dplyr)
x <- create_mif(clinical_data = example_clinical %>%
mutate(deidentified_id = as.character(deidentified_id)),
sample_data = example_summary %>%
mutate(deidentified_id = as.character(deidentified_id)),
spatial_list = example_spatial,
patient_id = "deidentified_id",
sample_id = "deidentified_sample")

#Nearest Neighbor distribution for the colocalization of CD3+CD8+ positive
```

```
#cells and CD3+FOXP3+ positive cells where CD3+FOXP3+ is the reference cell
#type at neighborhood size of 10,20,...,100 (zero must be included in the
#input).

x <- bi_NN_G(mif = x, mnames = c("CD3..CD8.", "CD3..FOXP3."),
num_permutations = 1, edge_correction = 'rs', r = seq(0,100,10),
keep_perm_dis = FALSE, workers = 1, exhaustive = TRUE)
```

`bi_ripleys_k` *Bivariate Count Based Measures of Spatial Clustering function for IF data*

Description

This function calculates count based Measures (Ripley's K, Besag L, and Marcon's M) of IF data to characterize correlation of the observed and permyted spatial point processes for two markers.

Usage

```
bi_ripleys_k(
  mif,
  mnames,
  r_range = seq(0, 100, 50),
  num_permutations = 50,
  edge_correction = "translation",
  method = "K",
  keep_perm_dis = FALSE,
  exhaustive = TRUE,
  workers = 1,
  overwrite = FALSE,
  xloc = NULL,
  yloc = NULL
)
```

Arguments

`mif`	An MIF object
`mnames`	Character vector of marker names to estimate degree of spatial clustering. Spatial clustering will be computed between each combination of markers in this list.
`r_range`	Numeric vector of potential r values this range must include 0
`num_permutations`	Numeric value indicating the number of permutations used. Default is 50.
`edge_correction`	Character value indicating the type of edge correction to use. Options include "theoretical", "translation", "isotropic" or "border". Various edges corrections are most appropriate in different settings. Default is "translation".

method	Character value indicating which measure (K, L, M) used to estimate the degree of spatial clustering. Description of the methods can be found in Details section.
keep_perm_dis	Logical value determining whether or not to keep the full distribution of permuted K values
exhaustive	Logical. If TRUE then markers must be a vector and spatial measures will be computed all pairs of unique markers. If FALSE then markers must be a data.frame with the desired combinations.
workers	Integer value for the number of workers to spawn
overwrite	Logical value determining if you want the results to replace the current output (TRUE) or be to be appended (FALSE).
xloc	a string corresponding to the x coordinates. If null the average of XMin and XMax will be used
yloc	a string corresponding to the y coordinates. If null the average of YMin and YMax will be used

Value

Returns a data frame

anchor	Marker for which the distances are measured from
counted	Marker for which the distances are measured to
Theoretical CSR	Expected value assuming complete spatial randomness
Permuted CSR	Average observed K, L, or M for the permuted point process
Observed	Observed value for the observed point process
Degree of Clustering Permuted	Degree of spatial clustering where the reference is the permuted estimate of CSR
Degree of Clustering Theoretical	Degree of spatial clustering where the reference is the theoretical estimate of CSR

Examples

```
#' #Create mif object
library(dplyr)
x <- create_mif(clinical_data = example_clinical %>%
mutate(deidentified_id = as.character(deidentified_id)),
sample_data = example_summary %>%
mutate(deidentified_id = as.character(deidentified_id)),
spatial_list = example_spatial,
patient_id = "deidentified_id",
sample_id = "deidentified_sample")

#Ripley's K for the colocalization of CD3+CD8+ positive cells and
#CD3+FOXP3+ positive cells where CD3+FOXP3+ is the reference cell type at
#neighborhood size of 10,20,...,100 (zero must be included in the input).
```

```
x <- bi_ripleys_k(mif = x, mnames = c("CD3..CD8.", "CD3..FOXP3."),
num_permutations = 1, edge_correction = 'translation', r = seq(0,100,10),
keep_perm_dis = FALSE, workers = 1, exhaustive = TRUE)
```

`create_mif` *Create Multiplex Immunoflourescent object*

Description

Creates an MIF object for use in spatialIF functions

Usage

```
create_mif(
  clinical_data,
  sample_data,
  spatial_list = NULL,
  patient_id = "patient_id",
  sample_id = "image_tag"
)
```

Arguments

`clinical_data`	A data frame containing patient level data with one row per participant.
`sample_data`	A data frame containing sample level data with one row per sample. Should at a minimum contain a 2 columns: one for sample names and one for the corresponding patient name.
`spatial_list`	A named list of data frames with the spatial data from each sample making up each individual data frame
`patient_id`	A character string indicating the column name for patient id in sample and clinical data frames.
`sample_id`	A character string indicating the column name for sample id in the sample data frame

Value

Returns a custom MIF

`clinical`	Data frame of clinical data
`sample`	Data frame of sample data
`spatial`	Named list of spatial data
`derived`	List of data derived using the MIF object
`patient_id`	The column name for sample id in the sample data frame with the clinical data
`sample_id`	The column name for sample id in the sample data frame to merge with the spatial data

Examples

```
#Create mif object
library(dplyr)
x <- create_mif(clinical_data = example_clinical %>%
mutate(deidentified_id = as.character(deidentified_id)),
sample_data = example_summary %>%
mutate(deidentified_id = as.character(deidentified_id)),
spatial_list = example_spatial,
patient_id = "deidentified_id",
sample_id = "deidentified_sample")
```

---

example_clinical *Clinical variables of 229 patients*

Description

A tibble wuith clinical characteristics for 229 patients

Usage

```
example_clinical
```

Format

A tibble with 229 rows and 6 variables

age age at diagnosis

race self-idenitifed race

sex patient biological sex

status disease status

deidenitifed_sample sample identifier

deidentified_id patient identifier

---

example_spatial *Example list of 5 spatial TMA data*

Description

A list containing 5 spatial data frames

Usage

```
example_spatial
```

Format

A list of 5 data frames:

- TMA_[3,B].tiff
- TMA_[6,F].tiff
- TMA_[7,B].tiff
- TMA_[9,K].tiff
- TMA_[8,U].tiff

---

example_summary *Marker summaries of 229 samples*

Description

A dataset containing summaries of 25 markers and 229 samples

Usage

```
example_summary
```

Format

A tibble with 229 rows and 29 variables:

deidentified_id patient-level id

deidentified_sample sample-level id ...

NN_G *Nearest Neighbor Based Measures of Spatial Clustering for IF data*

Description

For a given cell type, this function computes proportion of cells that have nearest neighbor less than r for the observed and permuted point processes.

Usage

```
NN_G(
  mif,
  mnames,
  r_range = seq(0, 100, 50),
  num_permutations = 50,
  edge_correction = "rs",
  keep_perm_dis = FALSE,
  workers = 1,
  overwrite = FALSE,
  xloc = NULL,
  yloc = NULL
)
```

Arguments

mif	An MIF object
mnames	Character vector of marker names to estimate degree of nearest neighbor distribution
r_range	Numeric vector of potential r values this range must include 0.
num_permutations	Numeric value indicating the number of permutations used. Default is 50.
edge_correction	Character value indicating the type of edge correction to use. Options include "rs" or "hans".
keep_perm_dis	Logical value determining whether or not to keep the full distribution of permuted G values
workers	Integer value for the number of workers to spawn
overwrite	Logical value determining if you want the results to replace the current output (TRUE) or be to be appended (FALSE).
xloc	a string corresponding to the x coordinates. If null the average of XMin and XMax will be used
yloc	a string corresponding to the y coordinates. If null the average of YMin and YMax will be used

Value

Returns a data.frame

Theoretical CSR	Expected value assuming complete spatial randomnessn
Permuted CSR	Average observed G for the permuted point process
Observed	Observed valuefor the observed point process
Degree of Clustering Permuted	Degree of spatial clustering where the reference is the permuted estimate of CSR

Degree of Clustering Theoretical
: Degree of spatial clustering where the reference is the theoretical estimate of CSR

Examples

```
#Create mif object
library(dplyr)
x <- create_mif(clinical_data = example_clinical %>%
mutate(deidentified_id = as.character(deidentified_id)),
sample_data = example_summary %>%
mutate(deidentified_id = as.character(deidentified_id)),
spatial_list = example_spatial,
patient_id = "deidentified_id",
sample_id = "deidentified_sample")

# Define the set of markers to study
markers <- c("CD3..Opal.570..Positive","CD8..Opal.520..Positive",
"FOXP3..Opal.620..Positive","CD3..CD8.","CD3..FOXP3.")

# Nearest Neighbor distribution for all markers with a neighborhood size
# of  10,20,...,100 (zero must be included in the input).


x <- NN_G(mif = x, mnames = markers, num_permutations = 1,
edge_correction = 'rs', r = seq(0,100,10),
keep_perm_dis = FALSE, workers = 1)
```

---

plot_immunoflo *Generate plot of TMA point process*

Description

This function generates plot of point process in rectangular or circular window.

Usage

```
plot_immunoflo(
  mif,
  plot_title,
  mnames,
  mcolors = NULL,
  cell_type = NULL,
  filename = NULL,
  path = NULL
)
```

Arguments

mif	MIF object created using create_MIF().
plot_title	Character string or vector of character strings of variable name(s) to serve as plot title(s).
mnames	Character vector containing marker names.
mcolors	Character vector of color names to display markers in the plot.
cell_type	Character vector of cell type
filename	Character string of file name to store plots. Plots are generated as single .pdf file.
path	Different path than file name or to use in conjunction with filename ???

Value

mif object and the ggplot objects can be viewed form the derived slot of the mif object

Examples

```
#Create mif object
library(dplyr)
x <- create_mif(clinical_data = example_clinical %>%
mutate(deidentified_id = as.character(deidentified_id)),
sample_data = example_summary %>%
mutate(deidentified_id = as.character(deidentified_id)),
spatial_list = example_spatial,
patient_id = "deidentified_id",
sample_id = "deidentified_sample")

mnames_good <- c("CD3..Opal.570..Positive","CD8..Opal.520..Positive",
"FOXP3..Opal.620..Positive","PDL1..Opal.540..Positive",
"PD1..Opal.650..Positive","CD3..CD8.","CD3..FOXP3.")

x <- plot_immunoflo(x, plot_title = "deidentified_sample", mnames = mnames_good,
cell_type = "Classifier.Label")

x[["derived"]][["spatial_plots"]][[4]]
```

---

ripleys_k *Calculate Count Based Measures of Spatial Clustering for IF data*

Description

This function calculates count based Measures (Ripley's K, Besag L, and Marcon's M) of IF data to characterize correlation of spatial point process.

Usage

```
ripleys_k(
  mif,
  mnames,
  r_range = seq(0, 100, 50),
  num_permutations = 50,
  edge_correction = "translation",
  method = "K",
  keep_perm_dis = FALSE,
  workers = 1,
  overwrite = FALSE,
  xloc = NULL,
  yloc = NULL
)
```

Arguments

mif	An MIF object
mnames	Character vector of marker names to estimate degree of spatial clustering.
r_range	Numeric vector of potential r values this range must include 0.
num_permutations	Numeric value indicating the number of permutations used. Default is 50.
edge_correction	Character value indicating the type of edge correction to use. Options include "translation" or "isotropic".
method	Character value indicating which measure (K, L, M) used to estimate the degree of spatial clustering. Description of the methods can be found in Details section.
keep_perm_dis	Logical value determining whether or not to keep the full distribution of permuted K values
workers	Integer value for the number of workers to spawn
overwrite	Logical value determining if you want the results to replace the current output (TRUE) or be to be appended (FALSE).
xloc	a string corresponding to the x coordinates. If null the average of XMin and XMax will be used
yloc	a string corresponding to the y coordinates. If null the average of YMin and YMax will be used

Value

Returns a data.frame

Theoretical CSR	Expected value assuming complete spatial randomnessn
Permuted CSR	Average observed K, L, or M for the permuted point process
Observed	Observed valuefor the observed point process

Degree of Clustering Permuted
: Degree of spatial clustering where the reference is the permutated estimate of CSR

Degree of Clustering Theoretical
: Degree of spatial clustering where the reference is the theoretical estimate of CSR

Examples

```
#Create mif object
library(dplyr)
x <- create_mif(clinical_data = example_clinical %>%
mutate(deidentified_id = as.character(deidentified_id)),
sample_data = example_summary %>%
mutate(deidentified_id = as.character(deidentified_id)),
spatial_list = example_spatial,
patient_id = "deidentified_id",
sample_id = "deidentified_sample")

# Define the set of markers to study
markers <- c("CD3..Opal.570..Positive","CD8..Opal.520..Positive",
"FOXP3..Opal.620..Positive","CD3..CD8.","CD3..FOXP3.")

# Ripley's K for all markers with a neighborhood size
# of  10,20,...,100 (zero must be included in the input).

x <- ripleys_k(mif = x, mnames = markers, num_permutations = 1,
edge_correction = 'translation', r = seq(0,100,10),
keep_perm_dis = FALSE, workers = 1)
```

---

subset_mif *Subset mif object on cellular level*

Description

This function allows to subset the mif object into compartments. For instance a mif object includes all cells and the desired analysis is based on only the tumor or stroma compartment then this function will subset the spatial list to just the cells in the desired compartment

Usage

```
subset_mif(mif, classifier, level, markers)
```

Arguments

mif	An MIF object
classifier	Column name for spatial dataframe to subset
level	Determines which level of the classifier to keep.
markers	vector of

Value

mif object where the spatial list only as the cell that are the specified level.

Examples

```
#' #Create mif object
library(dplyr)
x <- create_mif(clinical_data = example_clinical %>%
mutate(deidentified_id = as.character(deidentified_id)),
sample_data = example_summary %>%
mutate(deidentified_id = as.character(deidentified_id)),
spatial_list = example_spatial,
patient_id = "deidentified_id",
sample_id = "deidentified_sample")

markers = c("CD3..Opal.570..Positive","CD8..Opal.520..Positive",
"FOXP3..Opal.620..Positive","PDL1..Opal.540..Positive",
"PD1..Opal.650..Positive","CD3..CD8.","CD3..FOXP3.")

mif_tumor = subset_mif(mif = x, classifier = 'Classifier.Label',
level = 'Tumor', markers = markers)
```

Index