

Package ‘GCSM’

July 21, 2025

Title Implements Generic Composite Similarity Measure

Version 0.1.1

Description Provides implementation of the generic composite similarity measure (GCSM) described in Liu et al. (2020) <[doi:10.1016/j.ecoinf.2020.101169](https://doi.org/10.1016/j.ecoinf.2020.101169)>. The implementation is in C++ and uses 'RcppArmadillo'. Additionally, implementations of the structural similarity (SSIM) and the composite similarity measure based on means, standard deviations, and correlation coefficient (CMSC), are included.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

URL <https://github.com/liuyadong/GCSM>

BugReports <https://github.com/liuyadong/GCSM/issues>

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp

Suggests testthat

NeedsCompilation yes

Author Yadong Liu [aut, cre] (ORCID: <<https://orcid.org/0000-0002-1283-3451>>)

Maintainer Yadong Liu <liuyadong828@gmail.com>

Repository CRAN

Date/Publication 2021-03-27 14:40:08 UTC

Contents

cmsc	2
cmsc_sw	4
cmsc_tw	6
Index	9

`cmsc`*Composite similarity between vectors*

Description

Compute composite measures, GCSM or CMSC, between two vectors.

Usage

```
cmsc(  
  x,  
  y,  
  rescale = FALSE,  
  xmin = NA_real_,  
  xmax = NA_real_,  
  ymin = NA_real_,  
  ymax = NA_real_,  
  comp = "si"  
)
```

```
cmsc_e1(  
  x,  
  y,  
  rescale = FALSE,  
  xmin = NA_real_,  
  xmax = NA_real_,  
  ymin = NA_real_,  
  ymax = NA_real_,  
  comp = "si"  
)
```

```
cmsc_e2(  
  x,  
  y,  
  rescale = FALSE,  
  xmin = NA_real_,  
  xmax = NA_real_,  
  ymin = NA_real_,  
  ymax = NA_real_,  
  comp = "si"  
)
```

```
gcsn(  
  x,  
  y,  
  rescale = FALSE,  
  xmin = NA_real_,
```

```

    xmax = NA_real_,
    ymin = NA_real_,
    ymax = NA_real_,
    comp = "si"
  )

```

Arguments

<code>x</code>	A vector.
<code>y</code>	The other vector.
<code>rescale</code>	Rescale or not before computation.
<code>xmin, xmax, ymin, ymax</code>	Normalization parameters. If NA, are calculated from the ranges of <code>x</code> and <code>y</code> , respectively. See Details.
<code>comp</code>	Variable to return. If "si", the composite measure, if "s1", "s2" or "s3", the corresponding component.

Details

These functions compute composite measures between vectors. Missing values are omitted. Normalization parameters are used to rescale `x` and `y`, and determine the global minimum (`min`) and maximum (`max`). If `rescale` is `TRUE`, `x` and `y` are rescaled to $(x-xmin)/(xmax-xmin)$ and $(y-ymin)/(ymax-ymin)$; and set `min=0`, `max=1`. If `FALSE`, `min=min(xmin,ymin)`, `max=max(xmax,ymax)`.

Value

A number.

Examples

```

x = runif(9)
gscm(x, x)
cmisc(x, x)
# mean shift
gscm(x, x - 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
cmisc(x, x - 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
gscm(x, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
cmisc(x, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
## dissimilarity
y = 1 - x # y is the perfect antianalog of x
gscm(y, x)
gscm(y, x - 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
gscm(y, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)

# random noise
noise = rnorm(9, mean = 0, sd = 0.2)
gscm(x, x + noise, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
cmisc(x, x + noise, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
## dissimilarity
gscm(y, x + noise, xmin = 0, xmax = 1, ymin = 0, ymax = 1)

```

`cmc_sw`*Composite similarity on spatial windows*

Description

Compute composite measures, GCSM, CMC or SSIM, on spatial windows.

Usage

```
cmc_sw(  
  x,  
  y,  
  rescale = FALSE,  
  xmin = NA_real_,  
  xmax = NA_real_,  
  ymin = NA_real_,  
  ymax = NA_real_,  
  ksize = 9,  
  globe = FALSE,  
  comp = "si"  
)
```

```
cmc_e1_sw(  
  x,  
  y,  
  rescale = FALSE,  
  xmin = NA_real_,  
  xmax = NA_real_,  
  ymin = NA_real_,  
  ymax = NA_real_,  
  ksize = 9,  
  globe = FALSE,  
  comp = "si"  
)
```

```
cmc_e2_sw(  
  x,  
  y,  
  rescale = FALSE,  
  xmin = NA_real_,  
  xmax = NA_real_,  
  ymin = NA_real_,  
  ymax = NA_real_,  
  ksize = 9,  
  globe = FALSE,  
  comp = "si"  
)
```

```

gCsm_sw(
  x,
  y,
  rescale = FALSE,
  xmin = NA_real_,
  xmax = NA_real_,
  ymin = NA_real_,
  ymax = NA_real_,
  ksize = 9,
  globe = FALSE,
  comp = "si"
)

ssim_sw(
  x,
  y,
  rescale = FALSE,
  xmin = NA_real_,
  xmax = NA_real_,
  ymin = NA_real_,
  ymax = NA_real_,
  ksize = 11,
  sigma = 1.5,
  globe = FALSE,
  comp = "si"
)

```

Arguments

x	A matrix.
y	The other matrix.
rescale	Rescale or not before computation.
xmin, xmax, ymin, ymax	Normalization parameters. If NA, are calculated from the ranges of x and y, respectively. See Details.
ksize	Side length of spatial windows.
globe	Are data at the global scale? If TRUE, two vertical borders will be padded before computation.
comp	Variable to return. If "si", the composite measure, if "s1", "s2" or "s3", the corresponding component.
sigma	Standard deviation of Gaussian weighting function depending on the distance between the cell and kernel center.

Details

These functions slide the spatial window over space. Missing values are omitted. Normalization parameters are used to rescale x and y, and determine the global minimum (min) and maximum (max).

If rescale is TRUE, x and y are rescaled to $(x-xmin)/(xmax-xmin)$ and $(y-ymin)/(ymax-ymin)$; and set min=0, max=1. If FALSE, min=min(xmin,ymin), max=max(xmax,ymax). OpenMP is used for parallel computing.

Value

A matrix.

Examples

```
x = matrix(runif(36), nrow = 6, ncol = 6)

gscm_sw(x, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1, ksize = 3)
cmisc_sw(x, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1, ksize = 3)
ssim_sw(x, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1, ksize = 3)
```

cmisc_tw

Composite similarity on temporal windows

Description

Compute composite measures, GCSM or CMSC, on temporal windows.

Usage

```
cmisc_tw(
  xxx,
  yyy,
  rescale = FALSE,
  xmin = NA_real_,
  xmax = NA_real_,
  ymin = NA_real_,
  ymax = NA_real_,
  comp = "si"
)
```

```
cmisc_e1_tw(
  xxx,
  yyy,
  rescale = FALSE,
  xmin = NA_real_,
  xmax = NA_real_,
  ymin = NA_real_,
  ymax = NA_real_,
  comp = "si"
)
```

```
cmisc_e2_tw(
```

```

    xxx,
    yyy,
    rescale = FALSE,
    xmin = NA_real_,
    xmax = NA_real_,
    ymin = NA_real_,
    ymax = NA_real_,
    comp = "si"
)

gcm_tw(
  xxx,
  yyy,
  rescale = FALSE,
  xmin = NA_real_,
  xmax = NA_real_,
  ymin = NA_real_,
  ymax = NA_real_,
  comp = "si"
)

```

Arguments

xxx	A 3-d array with the 3rd dimension representing time.
yyy	The other 3-d array.
rescale	Rescale or not before computation.
xmin, xmax, ymin, ymax	Normalization parameters. If NA, are calculated from the ranges of xxx and yyy, respectively. See Details.
comp	Variable to return. If "si", the composite measure, if "s1", "s2" or "s3", the corresponding component.

Details

These functions slide the temporal window over space. Missing values are omitted. Normalization parameters are used to rescale xxx and yyy, and determine the global minimum (min) and maximum (max). If rescale is TRUE, xxx and yyy are rescaled to $(xxx - xmin) / (xmax - xmin)$ and $(yyy - ymin) / (ymax - ymin)$; and set min=0, max=1. If FALSE, min=min(xmin, ymin), max=max(xmax, ymax). OpenMP is used for parallel computing.

Value

A matrix.

Examples

```
x = array(runif(81), dim = c(3, 3, 9))
```

```
gcsn_tw(x, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)  
cmisc_tw(x, x + 0.2, xmin = 0, xmax = 1, ymin = 0, ymax = 1)
```

Index

`cmsc`, [2](#)
`cmsc_e1 (cmsc)`, [2](#)
`cmsc_e1_sw (cmsc_sw)`, [4](#)
`cmsc_e1_tw (cmsc_tw)`, [6](#)
`cmsc_e2 (cmsc)`, [2](#)
`cmsc_e2_sw (cmsc_sw)`, [4](#)
`cmsc_e2_tw (cmsc_tw)`, [6](#)
`cmsc_sw`, [4](#)
`cmsc_tw`, [6](#)

`gcsn (cmsc)`, [2](#)
`gcsn_sw (cmsc_sw)`, [4](#)
`gcsn_tw (cmsc_tw)`, [6](#)

`ssim_sw (cmsc_sw)`, [4](#)