

# Package: SSIMmap (via r-universe)

October 18, 2024

**Title** The structural similarity index measure (SSIM) for maps

**Version** 0.1.0

**Description** The SSIMmap package extend 'the classical SSIM method <<https://doi.org/10.1109/TIP.2003.819861>> for irregular lattice-based maps and raster images. The SSIMmap package applies this method to two types of maps (polygon and raster). The geographical SSIM method incorporates well-developed 'geographically weighted summary statistics' <[https://doi.org/10.1016/S0198-9715\(01\)00009-6](https://doi.org/10.1016/S0198-9715(01)00009-6)> with an adaptive bandwidth kernel function for irregular lattice-based maps.

**Imports** stats, scales, dplyr, terra, ggplot2, sf, knitr

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**URL** <https://github.com/Hailyee-Ha/SSIMmap>

**BugReports** <https://github.com/Hailyee-Ha/SSIMmap/issues>

**Depends** R (>= 3.5.0)

**LazyData** true

**Suggests** RColorBrewer, testthat (>= 3.0.0), rmarkdown

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Repository** <https://hailyee-ha.r-universe.dev>

**RemoteUrl** <https://github.com/hailyee-ha/ssimmap>

**RemoteRef** HEAD

**RemoteSha** 817fb7c597e44a5f7e2914b704864f6080273f5a

## Contents

ssim_bandwidth . . . . .	2
ssim_constant . . . . .	3
ssim_polygon . . . . .	4
ssim_raster . . . . .	5
Toronto . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

ssim_bandwidth	<i>Finding bandwidth ranges for the SSIM Index for polygon maps</i>
----------------	---

---

## Description

This function calculates the range of the bandwidth for the SSIM index using the square root of N and optimal trade-off between bias and variance

## Usage

```
ssim_bandwidth(
  shape,
  map1,
  map2,
  max_bandwidth = max_bandwidth,
  standardize = TRUE,
  option = "midpoint"
)
```

## Arguments

shape	a sf polygon containing the polygon data with attributes that can create polygon-based maps
map1	the name of the first map to compare as a column in the shape
map2	the name of the second map to compare as a column in the shape
max_bandwidth	maximum size of the bandwidth, and the maximum size needs to be larger than 12
standardize	if TRUE, standardize the variables before computing the SSIM. Default is TRUE
option	the option for selecting the range of the bandwidth derived from the optimal trade-off between bias and variance. Default is "midpoint"

## Details

This function calculates the bandwidth range for the SSIM index for polygon maps using Gaussian kernel weighting. The bandwidth is calculated by two methods: 1) the square root of N and 2) the optimal trade-off between bias and variance. Users can select the bandwidth values from the range of the optimal trade-off, which minimize the trade-off between bias and variance, generated by two maps.

**Value**

a plot showing the bias/variance trade off and the range of the optimal trade-off as vertical lines including the square root of N results as well. In addition, the console shows the results.

**Examples**

```
# Load example sf class object Toronto Area with attributes for maps:
# Pampalon Index, CIMD Index,
# and percentage of household commuting within the same Census Sub Division of residence
shape<-SSIMmap::Toronto

# Mapping two attributes
plot(shape$CIMD_SDD)
plot(shape$PP_SDD)

# Execution of bandwidth with maps above

ssim_bandwidth(shape, "CIMD_SDD", "PP_SDD", max_bandwidth=100)
```

---

ssim\_constant

*Calculate constants for SSIM Index*


---

**Description**

This function rescales the constants (k1 and k2) for the SSIM index based on the global maximum value of the maps

**Usage**

```
ssim_constant(shape, map1, map2, standardize = TRUE)
```

**Arguments**

shape	a sf polygon containing the polygon data with attributes that can create polygon-based maps
map1	the name of the first map to compare
map2	the name of the second map to compare
standardize	If TRUE, standardize the variables before computing the SSIM. Default is TRUE.

**Details**

This function calculates the rescaled constants (k1 and k2) for the SSIM index. k1 and k2 in the original SSIM method, which are for the 8-bit grey scale images, are 0.01 and 0.03 respectively. The SSIM for maps can use the rescaled k1 and k2 based on the global maximum value of two maps.

**Value**

The rescaled constants (k1 and k2)

**Examples**

```
# Load example sf class Toronto Area with attributes for maps:
# Toronto Areas with attributes for maps:Pampalon Index,CIMD Index,
# and percentage of household commuting within the same Census Sub Division of residence)
shape<-SSIMmap::Toronto
ssim_constant(shape,"PP_SDD","CIMD_SDD")
```

---

ssim\_polygon

---

*Calculate the structural similarity index measure for polygon maps.*


---

**Description**

This function computes the SSIM, a measure of similarity between two polygon maps. The `ssim_polygon` function computes the SSIM for each polygon and can be returned as a global average or for each polygon as a `sf` object

**Usage**

```
ssim_polygon(
  shape,
  map1,
  map2,
  standardize = TRUE,
  bandwidth = NULL,
  k1 = NULL,
  k2 = NULL,
  global = TRUE
)
```

**Arguments**

shape	a <code>sf</code> polygon containing the polygon data with attributes that can create polygon-based maps
map1	the name of the first map to compare as a column in the shape
map2	the name of the second map to compare as a column in the shape
standardize	If <code>TRUE</code> , standardize the variables before computing the SSIM. Default is <code>TRUE</code> .
bandwidth	bandwidth for the Gaussian kernel weighting used in the SSIM calculation. Default is the square root of <code>N</code>
k1	the constant used in the SSIM calculation. Default is <code>NULL</code> , in which case it is computed from the maximum value of variables.

k2	the constant used in the SSIM calculation. Default is NULL, in which case it is computed from the maximum value of variables.
global	If global is True, returning the global average of SSIM, SIM, SIV, and SIP. If the option is FALSE, a a sf SpatialPolygonsDataFrame containing the SSIM, SIM, SIV, and SIP for each polygon is returned Default is TRUE

### Details

This function computes the SSIM index for two polygon maps.

### Value

If global is TRUE, a string containing the global average SSIM, SIM, SIV, and SIP. If global is FALSE, a sf polygon containing the SSIM, SIM, SIV, and SIP for each polygon.

### Examples

```
# Load example sf object Toronto Area with attributes for maps:
# Pampalon Index,CIMD Index,
# and percentage of household commuting within the same Census Sub Division of residence
shape<-SSIMmap::Toronto

# Mapping two attributes
plot(shape$CIMD_SDD)
plot(shape$PP_SDD)
# Finding global ssim

ssim_polygon(shape,"CIMD_SDD","PP_SDD")
# Finding local ssim
df<-ssim_polygon(shape,"CIMD_SDD","PP_SDD",global=FALSE)
```

---

ssim\_raster

*Calculate SSIM index for raster images*


---

### Description

This function calculates the SSIM, a measure of similarity between two raster images

### Usage

```
ssim_raster(img1, img2, global = TRUE, w = 3, k1 = NULL, k2 = NULL)
```

### Arguments

img1	Raster object representing the first image.
img2	Raster object representing the second image.
global	If global is True, returning the global average of SSIM, SIM, SIV, and SIP. If the option is FALSE, a a raster raster brick containing the SSIM, SIM, SIV, and SIP for each cell is returned Default is TRUE

w	Integer specifying the window size for the local neighborhood. Default is 3.
k1	the constant used in the SSIM calculation. Default is NULL, in which case it is computed from the maximum value of variables.
k2	the constant used in the SSIM calculation. Default is NULL, in which case it is computed from the maximum value of variables.

### Details

This function computes the SSIM index for two raster images.

### Value

If global is True, returning the global average of SSIM, SIM, SIV, and SIP. If the option is FALSE, a raster raster brick containing the SSIM, SIM, SIV, and SIP for each cell is returned

### Examples

```
image_path1 <- system.file("ex", "groups2nm.tif", package = "SSIMmap", mustWork = TRUE)
image_path2 <- system.file("ex", "single2nm.tif", package = "SSIMmap", mustWork = TRUE)
img1<-terra::rast(image_path1)
img2<-terra::rast(image_path2)
ssim_raster(img1,img2)
result_raster<-ssim_raster(img1,img2,global=FALSE)
```

---

Toronto

*Sample polygon data of Toronto*

---

### Description

A sf(simple feature) containing geometric boundaries of Toronto DAs(Dissemination Area) with their codes.

### Usage

Toronto

### Format

A sf object with 58 rows and 2 variables

**DAUID** Dissemination Area ID

**CIMD\_SDD** Factor score of CIMD(The Canadian Index of Multiple Deprivation) social deprivation dimension

**PP\_SDD** Principal score of Pampalon social deprivation dimension

**P\_commute** Percentage of households who commute within census subdivision (CSD) of residence

**geometry** the geometry column for counties(CRS: NAD83)

# Index

## \* datasets

Toronto, [6](#)

ssim\_bandwidth, [2](#)

ssim\_constant, [3](#)

ssim\_polygon, [4](#)

ssim\_raster, [5](#)

Toronto, [6](#)