

Package ‘PatternClass’

March 14, 2020

Type Package

Title Class-Focused Pattern Metric Comparisons using Simulation

Version 2.0.1

Date 2020-03-13

Author Tarmo K. Remmel, (Marie-Josée Fortin, Ferenc Csillag, Sandor Kabos)

Maintainer Tarmo K. Remmel <remmelt@yorku.ca>

Description

NOTE: This content is being migrated to the package ShapePattern to provide a comprehensive set of tools for shape and pattern analysis. All future maintenance will be in that package -- please update your links. This current package provides tools for estimating composition and configuration parameters from a categorical (binary) landscape map (grid) and then simulates a selected number of statistically similar landscapes. Class-focused pattern metrics are computed for each simulated map to produce empirical distributions against which statistical comparisons can be made. The code permits the analysis of single maps or pairs of maps. Current limitation is for binary (classes 1, 2) maps that are 64x64 cells in extent.

License GPL (>= 3)

LazyData yes

LazyDataCompression xz

Depends R (>= 3.5.0), landscapemetrics, raster

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-13 23:00:13 UTC

R topics documented:

PatternClass-package	2
build.lut	3
buildsurfs	5
CARsimu	6
data	8
doubleplotter	9

findcol	10
findrow	11
imaks	13
singlemap	14
singleplotter	16
surfplot	18
WhittleData	19
wi	20
wtest.run	21

Index	23
--------------	-----------

PatternClass-package *Class-focused pattern metric comparisons*

Description

Permits the estimation of composition and configuration parameters from a binary landscape map (provided as a numeric matrix). These values are then used to simulate a series of statistically similar maps from which a suite of class-focused pattern metrics are computed and stored to produce empirical (null) distributions for each pattern metric. The package also produces probabilities and allows the testing of whether metric values measured for a map could have been produced by random chance. Numerous graphic outputs are also produced to facilitate the analysis and comparison of spatial patterns at the class-focus.

Details

Package: PatternClass
 Type: Package
 Version: 2.0.0
 Date: 2020-01-21
 License: WGPS >= 3.0

singlemap, singleplotter, and doubleplotter are the workhorses of this package.

Author(s)

Tarmo K. Remmel Maintainer: Tarmo K. Remmel <remmelt@yorku.ca>

References

- Remmel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s/10980-013-9905-x.
- Remmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and tech-*

niques for a sustainable world (2nd ed.). New York: Springer. <http://sarahgergel.net/lel/learning-landscape-ecology/>

See Also

It would be prudent to also observe the landscapemetrics package.

Examples

```
# Specific examples are provided on the individual function manual pages.
# The order of function calls should be: singlemap(), then singleplotter()
# or
# singlemap(), singlemap(), doubleplotter()
```

build.lut

Bias correction lookup table (LUT) builder based on simulations.

Description

This function needs to be run only once, at the beginning of analysis, to produce a lookup table of spatial autocorrelation (ρ) bias correction factors. Since this process takes a long time, due to the need for simulating numerous replicates (generally 50 or more) landscapes at each pair-wise combination of composition and configuration parameterization, we provide a lookup table with the package that is used by the `singlemap` function internally. Currently, the implementation is limited to 64x64 pixel maps; however, future developments will expand on this limitation and thus the `build.lut` function is provided for those that may wish to fast-track the construction of scaled lookup tables for analyses of landscapes of different size. Bias corrections are achieved by comparing Whittle estimates of spatial autocorrelation on binary versus continuous surfaces with identical parameterizations.

Usage

```
build.lut(LEVEL = 6, REPSIM = 5, RAJZ = FALSE, CIM = "", ENV="data")
```

Arguments

LEVEL	A power (n) of base 2 that controls the size of the simulated maps used to process the bias correction surface: $2^n \times 2^n$ cells.
REPSIM	The number of simulation replicates at each combination of composition and configuration.
RAJZ	A Boolean flag that when TRUE will draw the result.
CIM	An option that permits the naming of image objects. This is provided for potential future expansion.
ENV	The name of the environment where temporary objects are stored. The default is data.

Details

This does not need to be run if you have downloaded and installed the `PatternClass` package and are using it to analyze 64x64 pixel image subsets as proposed. The `singlemap` function will call on the provided internal lookup table and thus save on substantial processing time. The current implementation provides a lookup table with bias correction values based on the mean of 50 replicate simulations for each pair-wise combination of composition and configuration parameterization. Thus, the 9 levels of composition, 11 levels of spatial autocorrelation, and 50 replicates, means that $9*11*50 = 4950$ landscapes are simulated and used to produce the bias correction lookup table that is provided with the package.

Value

The result is a lookuptable (LUT) that is a 9x11 matrix of values used by `singlemap()`.

Note

The current implementation is for 64x64 image subsets. Future developments will expand this extent limitation. However, our work has shown that this extent provides a reasonable compromise between statistical power and computing efficiency.

Author(s)

Tarmo K. Rimmel

References

Rimmel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s10980-013-9905-x.

Rimmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and techniques for a sustainable world* (2nd ed.). New York: Springer.

See Also

See Also [singlemap](#)

Examples

```
build.lut(LEVEL = 6, REPSIM = 5, RAJZ = FALSE, CIM = "", ENV="data")
```

buildsurfs	<i>Build an array of surfaces across compositional and configuration analysis space.</i>
------------	--

Description

This function iteratively cycles through combinations of 9 binary proportion intervals and 11 spatial autocorrelation intervals and at each simulates a specified number of statistically similar binary landscape patterns. Each landscape is processed by landscapemetrics to produce empirical (null) distributions for class-focused pattern metrics. These surfaces can then be plotted or assessed for their mean values at any point (along with their variability).

Usage

```
buildsurfs(reps = 1000, verbose = TRUE)
```

Arguments

reps	An argument that defines the number of simulated maps that are produced for each combination of composition and configuration across the parameter space. The default value, 1000, is appropriate, but will take a long time to compute. The example below uses 2 replicates to demonstrate the process; however, the result should be used only for demonstrating that the code is working. Using 1000 simulations ensures that a valid empirical (null) distribution is constructed at each point on this surface.
verbose	When this argument is set to TRUE, additional information is provided during processing on screen. When FALSE, the code runs in relatively quiet mode, with little information provided on screen. When set to TRUE, it is easier to assess the time required to complete the processing.

Details

This is a simulation-intensive procedure and takes a long time. It is only necessary for advanced users who wish to assess the influence of composition and configuration (jointly) on specific pattern metrics. Once this function is run and the surfaces object is created, all further analysis can be performed quickly. This lookup reference set requires a lot of up-front processing, but simplifies all future analyses. The produced reference set is also necessary for asking specific questions regarding the expectation and variability of pattern metrics within the context of composition and configuration values (i.e., distributions at all pair-wise combinations of composition and configuration parameterizations).

Value

The returned value is an array called surfaces having dimensions [metrics, proportions, configurations, reps].

Note

The enclosed code is currently limited to 64x64 pixel subsets; however, future developments will expand the applicability to incorporate other dimensions. Since this code requires the simulation and storage of numerous landscape (raster) maps, the processing time (up-front) can be quite time consuming; however, once complete, plotting and analysis of mean and variance for any combination of composition and configuration and class-focused pattern metric are nearly instantaneous.

Author(s)

Tarmo K. Rimmel

References

Rimmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. Learning landscape ecology: concepts and techniques for a sustainable world (2nd ed.). New York: Springer.

See Also

See Also [singlemap](#), [singleplotter](#), and [doubleplotter](#)

Examples

```
surfaces <- data$surfaces
# COMPUTE SURFACES
surfaceexample <- buildsurfs(reps = 2, verbose = TRUE)
# COMPUTE MEDIAN SURFACE FOR A SINGLE METRIC
tempmed <- apply(surfaces[9,,], MARGIN=c(1,2), median)
# PRODUCE A PERSPECTIVE PLOT OF MEDIAN VALUES RESULTS FOR THE SELECTED METRIC
persp(tempmed, ticktype="detailed", cex.axis=0.7, zlab="Metric",
ylab="Proportion", xlab="Rho", theta=-45)
# COMPUTE VARIANCE SURFACE FOR A SINGLE METRIC
tempvar <- apply(surfaces[9,,], MARGIN=c(1,2), var)
# PRODUCE A PERSPECTIVE PLOT OF VARIANCE VALUES RESULTS FOR THE SELECTED METRIC
persp(tempvar, ticktype="detailed", cex.axis=0.7, zlab="Metric",
ylab="Proportion", xlab="Rho", theta=-45)
```

CARsimu

This function simulates an image using a FFT implementation of a Conditional Autoregressive (CAR) model.

Description

A CAR landscape simulator for isotropic conditions.

Usage

```
CARsimu(LEVEL = 6, rho = 0.2499, row2 = 0, col2 = 0, rc1 = 0, cr1 = 0,
maindi = 1, rajz = TRUE)
```

Arguments

LEVEL	This is the power (n) with base 2 that defines the image size: $2^n \times 2^n$.
rho	This is the spatial autocorrelation parameter.
row2	For later implementation - 2nd order neighbour spatial autocorrelation parameter (rows).
col2	For later implementation - 2nd order neighbour spatial autocorrelation parameter (columns).
rc1	For later implementation - diagonal spatial autocorrelation parameter.
cr1	For later implementation - diagonal spatial autocorrelation parameter.
maindi	For later implementation.
rajz	When rajz = TRUE, the simulated map will be drawn.

Details

This function can actually be parameterized in many ways, however, for isotropic landscapes, only the rho parameter is to be used. Additional parameterizations, although possible, are not permitted with the use of the other functions provided with the PatternClass package

Value

The output is a simulated map on a grid (provided as a matrix).

Note

All simulated landscapes are stationary and isotropic, thus extent cannot be larger than what would be expected to generally adhere to these constraints. The parameterization requires that the sum of all spatial autocorrelation parameters sum to < 0.5 . This is because the simulator is isotropic. A perfect 1.0 spatial autocorrelation parameter can be approximated by 0.499999 (infinitely repeating), due to computational limits. Outputs of this function can be saved to objects and used as continuous simulated landscape maps. To obtain categorical maps, these continuous maps can be sliced based on their distributions, as necessary.

Author(s)

Ferenc (Ferko) Csillag and Sandor Kabos; modified by Tarmo K. Rimmel

References

- Rimmel, T.K. and F. Csillag. 2003. When are two landscape pattern indices significantly different? *Journal of Geographical Systems* 5(4):331-351.
- Rimmel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s10980-013-9905-x.
- Rimmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R) (Pending 2014). In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and techniques for a sustainable world* (2nd ed.). New York: Springer.

See Also

See Also as [singlemap](#)

Examples

```
CARsimu(LEVEL = 6, rho = 0.2499, row2 = 0, col2 = 0, rc1 = 0, cr1 = 0,  
maindi = 1, rajz = TRUE)
```

data

A data environment for this package.

Description

Specific data objects and temporary objects are written to this environment.

Usage

```
data(data)
```

Format

The format is: <environment: 0x0000000005686de0>

Details

`demoimage1` This is a simulated binary landscape (64x64 pixels), having 40

`demoimage2` This is a simulated binary landscape (64x64 pixels), having 40

`DIFF50` This is a lookup table that has been generated in advance and provided with the package. It provides the bias correction values for estimating spatial autocorrelation from a categorical map using the Whittle method. The lookup table is a matrix of 10 rows (configuration) and 9 columns (composition). Each bias correction value is based on the mean of 50 replicates. This matrix can be rebuilt using more replicates and for data sets with extents different from 64x64 cells by calling the `build.lut` function. Warning: rebuilding this lookup tables takes a lot of time.

`surfaces` This is a pre-constructed array containing the 38 class-focused pattern metrics for one colour computed for 1000 replicates of each combination of 9 composition and 11 configuration parameterizations of simulated 64x64 cell binary maps. This array is used by the `surfp1ot` function to produce numerous permutations of summary plots that facilitate understanding and comparison of metrics. However, the production of this object takes considerable time using `buildsurfs`; thus it is provided here. Note: if plots, summaries, and comparisons are desired for maps having a different extent, then this array needs to be rebuilt. However, our work shows that this size of landscape is reasonable for this type of analysis.

`result1` This is the result of running `singlemap` on `demoimage1` and is used to demonstrate plotting output. Users will produce their own versions of this object when processing their own images.

`result2` This is the result of running `singlemap` on `demoimage2` and is used to demonstrate plotting output. Users will produce their own versions of this object when processing their own images.

WIBI This is a temporary object shared among functions and does not need to be accessed by users.

WITRUNC This is a temporary object shared among functions and does not need to be accessed by users.

Source

The data are all simulated using functions available in the PatternClass package.

References

Rommel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s/10980-013-9905-x.

Rommel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and techniques for a sustainable world* (2nd ed.). New York: Springer.

Examples

```
data(data)
str(data)
```

doubleplotter	<i>A comparison of two maps based on their empirical (null) distributions for a class-focused pattern metric</i>
---------------	--

Description

This function produces boxplots comparing the expected values (empirical, null distributions) for a given class-focused pattern metric between two maps.

Usage

```
doubleplotter(data1 = data$result1,
              data2 = data$result2,
              metric = 5)
```

Arguments

data1	This is the result object for the first map to be compared, where the output is returned by the function <code>singlemap</code> .
data2	This is the result object for the second map to be compared, where the output is returned by the function <code>singlemap</code> .
metric	This is an integer (1-110) indicating which metric the comparison boxplots will be drawn for. A list of these metrics is given in the header of the function's source code.

Details

No additional details at this time.

Value

The output is a double boxplot drawn on the graphics device.

Note

No additional notes at this time.

Author(s)

Tarmo K. Rimmel

References

No references at this time.

See Also

See Also [singlemap](#), [singleplotter](#), and [imaks](#).

Examples

```
# EXAMPLE USES PREVIOUSLY PRODUCED RESULTS TO SPEED-UP THE EXAMPLE,  
# BUT THE EXAMPLE FROM singlemap() SHOULD BE CALLED FIRST, ONCE FOR EACH  
# MAP TO BE COMPARED (FOR EXAMPLE)  
#result1 <- singlemap(IMG = data$demoimage1, VERBOSE = TRUE, reps = 5, LEVEL=6)  
#result2 <- singlemap(IMG = data$demoimage2, VERBOSE = TRUE, reps = 5, LEVEL=6)  
doubleplotter(data1 = data$result1, data2 = data$result2, metric = 5)
```

findcol

An internal utility that identifies that identifies the appropriate column of the Whittle correction matrix.

Description

This is an internal utility function for identifying a column index value in the Whittle correction matrix and returns it to the calling function for use in further processing. This function is not intended for use directly by users.

Usage

```
findcol(prop = 0.32, DIFFMAT=data$DIFF50, VERBOSE=FALSE)
```

Arguments

prop	This is a numeric argument providing the proportion of white pixels in the input binary map.
DIFFMAT	This is a matrix object containing the Whittle estimations (corrections) for spatial autocorrelation and generally constructed by <code>build.lut</code> .
VERBOSE	A Boolean argument that indicates whether the function should run in verbose mode or not.

Details

This tool is only used internally.

Value

The result is an index to be used as a pointer.

Note

This function is not to be used directly by users of this package.

Author(s)

Tarmo K. Remmel

References

No references currently.

See Also

Currently none.

Examples

```
# No example.
```

findrow

An internal utility that identifies that identifies the appropriate row of the Whittle correction matrix.

Description

This is an internal utility function for identifying a row index value in the Whittle correction matrix and returns it to the calling function for use in further processing. This function is not intended for use directly by users.

Usage

```
findrow(autocorr = 0.2, DIFFMAT=data$DIFF50, VERBOSE=FALSE)
```

Arguments

autocorr	This is a numeric argument providing the level of spatial autocorrelation.
DIFFMAT	This is a matrix object containing the Whittle estimations (corrections) for spatial autocorrelation and generally constructed by <code>build.lut</code> .
VERBOSE	A Boolean argument that indicates whether the function should run in verbose mode or not.

Details

This tool is only used internally.

Value

The result is a an index to be used as a pointer.

Note

This function is not to be used directly by users of this package.

Author(s)

Tarmo K. Remmel

References

No references currently.

See Also

Currently none.

Examples

```
# No example.
```

imaks	<i>Draws a matrix in the proper orientation, as if it were a raster landscape.</i>
-------	--

Description

A simple drawing function for matrices that are actually representing raster landscape maps. This function also controls the use of colour for nominal maps. This is a generic function for drawing any raster image that should not be drawn as a matrix (where the origin is not at the lower-left corner, but rather the upper-left corner. This function draws a matrix as a map using the proper positioning of the origin and gridded values.

Usage

```
imaks(BE = data$demoimage1, numcol = NULL, LENG = 4, colour = FALSE)
```

Arguments

BE	The input landscape map as a matrix object.
numcol	A numeric value indicating the number of total colours on the map.
LENG	An argument that is depreciating. Do not adjust this value.
colour	Should a special colour scheme be applied (TRUE) or not (FALSE)

Details

In a future release, this function may become obsolete as we migrate to using the raster library and raster objects.

Value

The result is a graphic plot of a raster landscape as read from a matrix representation.

Note

This function was originally released in the package: hdeco

Author(s)

Sandor Kabos (modified by Tarmo K. Remmel)

References

No references currently.

See Also

Currently none.

Examples

```
data(demoimage1)
imaks(BE = data$demoimage1, numcol = NULL, LENG = 4, colour = FALSE)
```

singlemap

Process a single binary map

Description

This function estimates the composition and configuration parameters for a binary map having 64x64 pixel extent. The 64x64 extent is a current limitation and will be expanded upon in subsequent updates. The parameter estimates are corrected for bias introduced by the Whittle estimation and then these parameters are used to simulate a series of statistically similar landscapes. Each of these simulated landscapes are processed with SDMTTools to compute class-focused pattern metrics that are saved to produce empirical (null) distributions for each metric, given the specified level of spatial autocorrelation (configuration) and class proportion (composition). The results can be saved to a new object that is later used for producing graphical and statistical outputs.

Usage

```
singlemap(IMG = data$demoimage1,
          CORRECTIONMAT = data$DIFF50,
          VERBOSE = TRUE,
          reps = 1,
          LEVEL=6)
```

Arguments

- | | |
|---------------|---|
| IMG | This is a numeric, binary matrix that represents a raster landscape. Internally, for computing landscape metrics, this is handled as a raster class object, but input must be a two-valued matrix. There is no implementation for the spatial resolution of each cell as the computed metrics do not require this value. It assumes that the spatial resolution is consistent in both dimensions and across the entire scene. The image is also assumed to be a graphic representation resulting from a stationary spatial process that is 64x64 pixels in extent. The extent limitation will be expanded in a future release but for stationary and isotropic landscape patterns, this is deemed sufficient for now and could be used in a landscape sampling context. |
| CORRECTIONMAT | This is the name of a numeric matrix that provides the correction factors based on Whittle's estimation. The default matrix DIFF50 is provided in the Whittle-Data environment that is supplied with this package. However, if image sizes differ (i.e., not the 64x64 demos that are provided), this matrix needs to be recreated using <code>build.lut</code> and the result provided here. The code, by default, will search for this name in the data environment unless you specify a custom matrix that you have created. |

VERBOSE	This argument is set to TRUE if you desire extensive on-screen feedback as to what the program is doing. If set to FALSE, much less information is written to screen. It is useful to set to TRUE if reps is large, since processing times can be long; this gives an idea as to how much time is remaining to complete processing the data.
reps	This argument controls the number of replicate landscapes that will be simulated and used to produce the empirical (null) distributions for the class-focused pattern metrics. The default value is set at 1 to permit a quick check that the code is loaded and working correctly, but should be run with a value of 100-1000 or larger.
LEVEL	This is a measure of the image size being processed. It is necessary for preparing the necessary arrays used for estimating rho. The value must be an integer conforming to the image size, such that LEVEL = N, where N is $2^N \times 2^N$ for the image dimensions. Thus, for a 64x64 pixel map, LEVEL = 6.

Details

When analyzing multiple maps, this function should be called independently for each map, with the results stored to unique objects that become input to either `singleplotter` or `doubleplotter`.

Value

This returns an object containing the empirical (null) distributions for a suite of class-focused pattern metrics. The rows are observations for each replicate and the columns are the class metrics. Column names identify the actual metrics that are defined in the `landscapemetrics` package.

Note

Implementation is currently for 64x64 binary landscapes, provided as arguments in integer matrix format. The size limitation will be relaxed in a future release when some of the internal coding structure is streamlined. The result of this function call should be saved to an object that can subsequently be used in calls by either `singleplotter` or `doubleplotter`.

Author(s)

Tarmo K. Rimmel

References

- Rimmel, T.K. and F. Csillag. 2003. When are two landscape pattern indices significantly different? *Journal of Geographical Systems* 5(4):331-351
- Rimmel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s/10980-013-9905-x.
- Rimmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and techniques for a sustainable world* (2nd ed.). New York: Springer.

See Also

See Also [singleplotter](#), and [doubleplotter](#).

Examples

```
result1 <- singlemap(IMG = data$demoimage1, CORRECTIONMAT = data$DIFF50,
  VERBOSE = TRUE, reps = 1, LEVEL=6)
```

singleplotter	<i>Depict the empirical (null) distribution for class-focused pattern metrics.</i>
---------------	--

Description

Produce a series of one or more boxplots depicting the empirical (null) distributions of class-focused pattern metrics, as computed for a single landscape map. The code has been updated as of October 24 2016 to permit boxplots to be drawn even if landscape metrics have NA in them (those values are simply ignored). As of 21 January 2020, the code now relies on landscapemetrics and raster packages rather than the diminished SDMTools.

Usage

```
singleplotter(dat = data$result1,
  img = data$demoimage1,
  metrics = c(1, 5, 10),
  rows = 1,
  cols = 3,
  addactual = TRUE,
  colour = TRUE)
```

Arguments

dat	This is the output object produced by <code>singlemap</code> . It contains the empirical (null) distributions for class-focused pattern metrics computed for a single binary landscape map.
img	This is the binary integer matrix landscape map that corresponds to the data argument.
metrics	A vector listing the integers, corresponding to specific metrics, that you want to plot. There are 55 unique class-level metrics computed for each of 2 classes; thus, the specific metric values in the specified vector can range from 1-110.
rows	This argument controls the number of rows (of plots) on the output graphics device. The total number of graphs that can be displayed simultaneously will be (rows * cols). These values should be assigned to correspond with the number of metrics that you want to view. Trying to display too many will make them very small and not very informative. It may be more effective to simply call this function multiple times with fewer metrics.

cols	This argument controls the number of columns (of plots) on the output graphics device. The total number of graphs that can be displayed simultaneously will be (rows * cols). These values should be assigned to correspond with the number of metrics that you want to view. Trying to display too many will make them very small and not very informative. It may be more effective to simply call this function multiple times with fewer metrics.
addactual	If TRUE, then the landscape's actual class-focused pattern metric will be added to the plot to depict its relative position within the empirical (null) distribution.
colour	If TRUE, the actual landscape's class-focused pattern metric, if added, will be drawn in red. If FALSE, the plot will be without colour (hollow circle).

Details

This function requires as input, both the original image object (`img`) and output object from `singlemap` (`data`) such that the proper metric values can be plotted along with their expectations (distributions). This is the work-horse plotting function for metric values and distributions for any specific landscape object. Currently, implementation is for 64x64 pixel subset landscape maps. Future development will expand on this limitation; however, our work has shown that this extent is sufficient for adequate pattern characterization and comparison purposes.

Value

The output is graphical.

Note

No further notes at this time.

Author(s)

Tarmo K. Rimmel

References

Rimmel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s10980-013-9905-x.

Rimmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and techniques for a sustainable world* (2nd ed.). New York: Springer.

See Also

See Also [singlemap](#), and [doubleplotter](#).

Examples

```
# EXAMPLE USES PREVIOUSLY PRODUCED RESULTS TO SPEED-UP THE EXAMPLE,  
# BUT THE EXAMPLE FROM singlemap() SHOULD BE CALLED FIRST  
singleplotter(dat = data$result1, img = data$demoimage1, metrics = c(1, 5, 10))
```

surfplot	<i>Produce graphic plots providing class-focused pattern metric context for a landscape map</i>
----------	---

Description

This function produces three separate plots on a 1 row by 3 column output view. The first plot is the expected mean surface for a selected class-focused pattern metric across the possible range of composition and configuration. A dropped marker can be added to the plot that depicts the position of any combination of composition or configuration parameterization. The second and third plots provide series of boxplots representing orthogonal cross-sections of the surface in the first plot, intersecting at the specified location. This provides a sense of how the selected metric's expected value and variability will change as the composition and/or configuration parameter are altered.

Usage

```
surfplot(metric = 9, prop = 0.7, rho = 0.2, colour = TRUE, drop = TRUE,
cross = TRUE, dat=data$surfaces)
```

Arguments

metric	An integer identifying the class-focused pattern metric that the plots will represent. These integers are listed in the source code for <code>singleplotter</code> .
prop	A real value between zero (0) and one (1) that defines the proportion of the focal land cover category. This value along with <code>rho</code> define where the marker will be drawn on the first plot and through which the orthogonal cross-sections will be drawn.
rho	A real value between zero (0) and one (1) that defines the level of spatial auto-correlation for the focal land cover category. This value along with <code>prop</code> define where the marker will be drawn on the first plot and through which the orthogonal cross-sections will be drawn.
colour	A Boolean indication as to whether the plot should contain colour (TRUE) or not (FALSE).
drop	A Boolean indication as to whether the specified point based on composition (<code>prop</code>) and configuration (<code>rho</code>) should be added to the plot (TRUE) or not (FALSE).
cross	A Boolean indication as to whether the cross-sectional boxplots should be drawn (TRUE) or not (FALSE).
dat	A numeric array object created by <code>buildsurfs</code> that has dimensions [38,9,11,replicates]. The version provided is produced with only 5 replicates to save space; it is highly recommended to produce a <code>dat</code> object, such as <code>surfaces</code> that is produced with 1000 replicates.

Details

This function requires a valid result from the function `buildsurfs` to exist. That result is an array of simulated maps for a series of replicates produced for all pair-wise parameterizations of composition and configuration. This object can be quite large; however, needs to only be produced once. All plots thereafter can be produced (for any class-focused metric) from this stored object. To save processing time, the `PatternClass` package is provided with these reference (lookup) surfaces already produced for 64x64 images. Future developments will permit these layers to be produced at a wider range of extents.

Value

This function returns a graphic plot.

Note

No further notes at this time.

Author(s)

Tarmo K. Rimmel

References

Rimmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. Learning landscape ecology: concepts and techniques for a sustainable world (2nd ed.). New York: Springer.

See Also

See Also [singlemap](#), [singleplotter](#), [buildsurfs](#), and [doubleplotter](#).

Examples

```
surfplot(metric = 9, prop = 0.7, rho = 0.2, colour = TRUE, drop = TRUE,  
cross = TRUE, dat=data$surfaces)
```

WhittleData

Whittle correction matrix used by this package.

Description

Specific location for storing the Whittle correction matrix to protect it from being overwritten accidentally.

Usage

```
data(WhittleData)
```

Format

The format is: <environment: 0x0000000005686de0>

Details

DIFF50 This is a lookup table that has been generated in advance and provided with the package. It provides the bias correction values for estimating spatial autocorrelation from a categorical map using the Whittle method. The lookup table is a matrix of 10 rows (configuration) and 9 columns (composition). Each bias correction value is based on the mean of 50 replicates. This matrix can be rebuilt using more replicates and for data sets with extents different from 64x64 cells by calling the `build.lut` function. Warning: rebuilding this lookup tables takes a lot of time.

Source

The data are all simulated using functions available in the `PatternClass` package.

References

Rommel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s/10980-013-9905-x.

Rommel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and techniques for a sustainable world* (2nd ed.). New York: Springer.

Examples

```
data(WhittleData)
str(WhittleData)
```

 wi

An internal utility that helps with the Whittle estimation.

Description

This is an internal utility function for helping with the Whittle estimation. This function is not intended for use directly by users.

Usage

```
wi(BE=data$demoimage1, CONTROL=FALSE, SIZE=6)
```

Arguments

BE	This is a binary input image.
CONTROL	This is a Boolean flag.
SIZE	Controls the output matrix size.

Details

This tool is only used internally.

Value

The result is a value.

Note

This function is not to be used directly by users of this package.

Author(s)

Ferko Csillag, Sandor Kabos, Tarmo K. Remmel

References

No references currently.

See Also

Currently none.

Examples

```
# No example.
```

wtest.run

Whittle estimation for binary map

Description

This function is the workhorse for estimating the Whittle correction for a binary map. This function is generally called by a wrapping function to facilitate its use (e.g., `singlemap`).

Usage

```
wtest.run(LEVEL=6, REPSIM=20, RHO=0.2499999, CPROP=0.5, RAJZ=T, CIM="CIM", ENV="data")
```

Arguments

LEVEL This is a numeric, binary matrix that represents a raster landscape. In future updates, this argument may migrate to be a raster object, but for now, it must be a two-valued matrix. There is no implementation for the spatial resolution of each cell as the computed metrics do not require this value. It assumes that the spatial resolution is consistent in both dimensions and across the entire scene. The image is also assumed to be a graphic representation resulting from a stationary spatial process.

REPSIM	This is a numeric matrix that provides the correction factors based on Whittle's estimation. The default matrix DIFF50 is provided in the data environment that is supplied with this package. However, if image sizes differ (i.e., not the 64x64 demos that are provided), this matrix needs to be recreated using <code>build.lut</code> and the result provided here.
RHO	This is a numeric argument recording the spatial autocorrelation parameter. Note that this is divided into 4 and thus 0.2499999 approaches the theoretical maximum value of 1.
CPROP	This argument controls the proportion of white to black pixels.
RAJZ	This Boolean argument controls whether outputs of the simulation are drawn on the screen.
CIM	A parameter that can control the naming of outputs.
ENV	A parameter that can control the name of the environment where temporary objects are stored.

Details

This code is generally not called directly by users.

Value

Not for users.

Note

This is for internal use. Use `singlemap` instead.

Author(s)

Tarmo K. Rimmel

References

- Rimmel, T.K. and F. Csillag. 2003. When are two landscape pattern indices significantly different? *Journal of Geographical Systems* 5(4):331-351
- Rimmel, T.K. and M.-J. Fortin. 2013. Categorical class map patterns: characterization and comparison. *Landscape Ecology*. DOI: 10.1007/s/10980-013-9905-x.
- Rimmel, T.K. and M.-J. Fortin. What constitutes a significant difference in landscape pattern? (using R). 2016. In Gergel, S.E. and M.G. Turner. *Learning landscape ecology: concepts and techniques for a sustainable world* (2nd ed.). New York: Springer.

See Also

See Also [singlemap](#).

Examples

```
# No example.
```

Index

- *Topic **classes**
 - singlemap, 14
 - *Topic **datasets**
 - data, 8
 - WhittleData, 19
 - *Topic **distributions**
 - singleplotter, 16
 - surfplot, 18
 - *Topic **distribution**
 - build.lut, 3
 - buildsurfs, 5
 - doubleplotter, 9
 - singlemap, 14
 - *Topic **hplot**
 - CARsimu, 6
 - imaks, 13
 - *Topic **misc**
 - findcol, 10
 - findrow, 11
 - wi, 20
 - wtest.run, 21
 - *Topic **package**
 - PatternClass-package, 2
- build.lut, 3
- buildsurfs, 5, 19
- CARsimu, 6
- data, 8
- doubleplotter, 6, 9, 16, 17, 19
- findcol, 10
- findrow, 11
- imaks, 10, 13
- PatternClass (PatternClass-package), 2
- PatternClass-package, 2
- singlemap, 4, 6, 8, 10, 14, 17, 19, 22
- singleplotter, 6, 10, 16, 16, 19
- surfplot, 18
- WhittleData, 19
- wi, 20
- wtest.run, 21