

# Package ‘landscapemetrics’

September 10, 2019

**Type** Package

**Title** Landscape Metrics for Categorical Map Patterns

**Version** 1.2.2

**Maintainer** Maximillian H.K. Hesselbarth <maximilian.hesselbarth@uni-goettingen.de>

**Description** Calculates landscape metrics for categorical landscape patterns in a tidy workflow. 'landscapemetrics' reimplements the most common metrics from 'FRAGSTATS' (<<https://www.umass.edu/landeco/research/fragstats/fragstats.html>>) and new ones from the current literature on landscape metrics.

This package supports 'raster' spatial objects and takes RasterLayer, RasterStacks, RasterBricks or lists of RasterLayer from the 'raster' package as input arguments. It further provides utility functions to visualize patches, select metrics and building blocks to develop new metrics.

**License** GPL-3

**URL** <https://r-spatialecology.github.io/landscapemetrics/>

**BugReports** <https://github.com/r-spatialecology/landscapemetrics/issues>

**Depends** R (>= 3.1)

**Imports** cli, ggplot2, methods, raster, Rcpp, sp, stats, tibble

**Suggests** covr, dplyr, knitr, rgeos, rmarkdown, testthat

**Enhances** stars, sf

**LinkingTo** Rcpp, RcppArmadillo

**ByteCompile** true

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**SystemRequirements** C++11

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Maximillian H.K. Hesselbarth [aut, cre]  
 (<<https://orcid.org/0000-0003-1125-9918>>),  
 Marco Sciaini [aut] (<<https://orcid.org/0000-0002-3042-5435>>),  
 Jakub Nowosad [aut] (<<https://orcid.org/0000-0002-1057-3721>>),  
 Sebastian Hanss [aut] (<<https://orcid.org/0000-0002-3990-4897>>),  
 Laura J. Graham [ctb] (Input on package structure),  
 Jeffrey Hollister [ctb] (Input on package structure),  
 Kimberly A. With [ctb] (Input on package structure),  
 Florian Privé [ctb] (Original author of underlying C++ code for  
 get\_nearestneighbour() function),  
 Jeremy VanDerWal [ctb] (Original author of underlying C code for  
 get\_patches() function),  
 Matt Strimas-Mackey [ctb] (Bugfix in sample\_metrics())

**Repository** CRAN

**Date/Publication** 2019-09-10 10:40:02 UTC

## R topics documented:

augusta_nlcd . . . . .	6
calculate_correlation . . . . .	6
calculate_lsm . . . . .	7
check_landscape . . . . .	9
extract_lsm . . . . .	10
fragstats_class_augusta_nlcd . . . . .	12
fragstats_class_landscape . . . . .	13
fragstats_class_podlasie . . . . .	13
fragstats_landscape_augusta_nlcd . . . . .	14
fragstats_landscape_landscape . . . . .	14
fragstats_landscape_podlasie . . . . .	15
fragstats_patch_augusta_nlcd . . . . .	15
fragstats_patch_landscape . . . . .	16
fragstats_patch_podlasie . . . . .	16
get_adjacencies . . . . .	17
get_boundaries . . . . .	18
get_circumscribingcircle . . . . .	20
get_nearestneighbour . . . . .	21
get_patches . . . . .	22
get_unique_values . . . . .	24
landscape . . . . .	25
landscapemetrics . . . . .	26
list_lsm . . . . .	27
lsm_abbreviations_names . . . . .	28
lsm_c_ai . . . . .	29
lsm_c_area_cv . . . . .	30
lsm_c_area_mn . . . . .	32
lsm_c_area_sd . . . . .	33
lsm_c_ca . . . . .	35

lsm_c_cai_cv . . . . .	36
lsm_c_cai_mn . . . . .	38
lsm_c_cai_sd . . . . .	39
lsm_c_circle_cv . . . . .	41
lsm_c_circle_mn . . . . .	43
lsm_c_circle_sd . . . . .	44
lsm_c_clumpy . . . . .	46
lsm_c_cohesion . . . . .	47
lsm_c_contig_cv . . . . .	49
lsm_c_contig_mn . . . . .	50
lsm_c_contig_sd . . . . .	52
lsm_c_core_cv . . . . .	54
lsm_c_core_mn . . . . .	55
lsm_c_core_sd . . . . .	57
lsm_c_cpland . . . . .	59
lsm_c_dcad . . . . .	60
lsm_c_dcore_cv . . . . .	62
lsm_c_dcore_mn . . . . .	64
lsm_c_dcore_sd . . . . .	66
lsm_c_division . . . . .	67
lsm_c_ed . . . . .	69
lsm_c_enn_cv . . . . .	71
lsm_c_enn_mn . . . . .	72
lsm_c_enn_sd . . . . .	74
lsm_c_frac_cv . . . . .	76
lsm_c_frac_mn . . . . .	77
lsm_c_frac_sd . . . . .	79
lsm_c_gyrate_cv . . . . .	80
lsm_c_gyrate_mn . . . . .	82
lsm_c_gyrate_sd . . . . .	83
lsm_c_iji . . . . .	85
lsm_c_lpi . . . . .	86
lsm_c_lsi . . . . .	88
lsm_c_mesh . . . . .	89
lsm_c_ndca . . . . .	91
lsm_c_nlsi . . . . .	92
lsm_c_np . . . . .	94
lsm_c_pafrac . . . . .	95
lsm_c_para_cv . . . . .	97
lsm_c_para_mn . . . . .	99
lsm_c_para_sd . . . . .	100
lsm_c_pd . . . . .	102
lsm_c_pladj . . . . .	103
lsm_c_pland . . . . .	104
lsm_c_shape_cv . . . . .	106
lsm_c_shape_mn . . . . .	108
lsm_c_shape_sd . . . . .	109
lsm_c_split . . . . .	111

lsm_c_tca . . . . .	112
lsm_c_te . . . . .	114
lsm_l_ai . . . . .	115
lsm_l_area_cv . . . . .	117
lsm_l_area_mn . . . . .	118
lsm_l_area_sd . . . . .	120
lsm_l_cai_cv . . . . .	121
lsm_l_cai_mn . . . . .	123
lsm_l_cai_sd . . . . .	125
lsm_l_circle_cv . . . . .	127
lsm_l_circle_mn . . . . .	128
lsm_l_circle_sd . . . . .	130
lsm_l_cohesion . . . . .	131
lsm_l_condent . . . . .	133
lsm_l_contag . . . . .	134
lsm_l_contig_cv . . . . .	136
lsm_l_contig_mn . . . . .	137
lsm_l_contig_sd . . . . .	139
lsm_l_core_cv . . . . .	141
lsm_l_core_mn . . . . .	142
lsm_l_core_sd . . . . .	144
lsm_l_dcad . . . . .	146
lsm_l_dcore_cv . . . . .	147
lsm_l_dcore_mn . . . . .	149
lsm_l_dcore_sd . . . . .	151
lsm_l_division . . . . .	153
lsm_l_ed . . . . .	154
lsm_l_enn_cv . . . . .	156
lsm_l_enn_mn . . . . .	157
lsm_l_enn_sd . . . . .	159
lsm_l_ent . . . . .	161
lsm_l_frac_cv . . . . .	162
lsm_l_frac_mn . . . . .	164
lsm_l_frac_sd . . . . .	165
lsm_l_gyrate_cv . . . . .	167
lsm_l_gyrate_mn . . . . .	168
lsm_l_gyrate_sd . . . . .	170
lsm_l_iji . . . . .	171
lsm_l_joinent . . . . .	173
lsm_l_lpi . . . . .	174
lsm_l_lsi . . . . .	175
lsm_l_mesh . . . . .	177
lsm_l_msidi . . . . .	179
lsm_l_msiei . . . . .	180
lsm_l_mutinf . . . . .	182
lsm_l_ndca . . . . .	183
lsm_l_np . . . . .	185
lsm_l_pafrac . . . . .	186

lsm_l_para_cv . . . . .	188
lsm_l_para_mn . . . . .	189
lsm_l_para_sd . . . . .	191
lsm_l_pd . . . . .	192
lsm_l_pladj . . . . .	194
lsm_l_pr . . . . .	195
lsm_l_prd . . . . .	196
lsm_l_rpr . . . . .	198
lsm_l_shape_cv . . . . .	199
lsm_l_shape_mn . . . . .	201
lsm_l_shape_sd . . . . .	202
lsm_l_shdi . . . . .	204
lsm_l_shei . . . . .	205
lsm_l_sidi . . . . .	207
lsm_l_siei . . . . .	208
lsm_l_split . . . . .	210
lsm_l_ta . . . . .	211
lsm_l_tca . . . . .	213
lsm_l_te . . . . .	214
lsm_p_area . . . . .	216
lsm_p_cai . . . . .	217
lsm_p_circle . . . . .	219
lsm_p_contig . . . . .	221
lsm_p_core . . . . .	222
lsm_p_enn . . . . .	224
lsm_p_frac . . . . .	226
lsm_p_gyrate . . . . .	227
lsm_p_ncore . . . . .	229
lsm_p_para . . . . .	230
lsm_p_perim . . . . .	232
lsm_p_shape . . . . .	233
options_landscapemetrics . . . . .	235
podlasie_ccile . . . . .	235
sample_lsm . . . . .	236
scale_sample . . . . .	238
scale_window . . . . .	240
show_cores . . . . .	242
show_correlation . . . . .	243
show_lsm . . . . .	244
show_patches . . . . .	246
spatialize_lsm . . . . .	247
window_lsm . . . . .	248

---

 augusta\_nlcd

*Augusta NLCD 2011*


---

### Description

A real landscape of area near Augusta, Georgia obtained from the National Land Cover Database (NLCD)

### Usage

```
augusta_nlcd
```

### Format

A raster layer object.

### Source

<https://www.mrlc.gov/nlcd2011.php>

### References

Homer, C.G., Dewitz, J.A., Yang, L., Jin, S., Danielson, P., Xian, G., Coulston, J., Herold, N.D., Wickham, J.D., and Megown, K., 2015, Completion of the 2011 National Land Cover Database for the conterminous United States-Representing a decade of land cover change information. Photogrammetric Engineering and Remote Sensing, v. 81, no. 5, p. 345-354

---

 calculate\_correlation *Calculate correlation*


---

### Description

Calculate correlation

### Usage

```
calculate_correlation(metrics, method = "pearson", diag = TRUE,
  simplify = FALSE)
```

### Arguments

metrics	Tibble with results of as returned by the landscapemetrics package.
method	Type of correlation. See <code>link{cor}</code> for details.
diag	If FALSE, values on the diagonal will be NA.
simplify	If TRUE and only one level is present, only a tibble is returned.

## Details

The functions calculates the correlation between all metrics. In order to calculate correlations, for the landscape level more than one landscape needs to be present. All input must be structured as returned by the **landscapemetrics** package.

## Value

list

## Examples

```
metrics <- calculate_lsm(landscape, what = c("patch", "class"))
calculate_correlation(metrics, method = "pearson")
```

---

calculate_lsm	<i>calculate_lsm</i>
---------------	----------------------

---

## Description

Calculate a selected group of metrics

## Usage

```
calculate_lsm(landscape, level, metric, name, type, what, directions,
  count_boundary, consider_boundary, edge_depth, classes_max,
  neighbourhood, ordered, base, full_name, verbose, progress)
```

```
## S3 method for class 'RasterLayer'
calculate_lsm(landscape, level = NULL,
  metric = NULL, name = NULL, type = NULL, what = NULL,
  directions = 8, count_boundary = FALSE, consider_boundary = FALSE,
  edge_depth = 1, classes_max = NULL, neighbourhood = 4,
  ordered = TRUE, base = "log2", full_name = FALSE, verbose = TRUE,
  progress = FALSE)
```

```
## S3 method for class 'RasterStack'
calculate_lsm(landscape, level = NULL,
  metric = NULL, name = NULL, type = NULL, what = NULL,
  directions = 8, count_boundary = FALSE, consider_boundary = FALSE,
  edge_depth = 1, classes_max = NULL, neighbourhood = 4,
  ordered = TRUE, base = "log2", full_name = FALSE, verbose = TRUE,
  progress = FALSE)
```

```
## S3 method for class 'RasterBrick'
calculate_lsm(landscape, level = NULL,
  metric = NULL, name = NULL, type = NULL, what = NULL,
```

```

directions = 8, count_boundary = FALSE, consider_boundary = FALSE,
edge_depth = 1, classes_max = NULL, neighbourhood = 4,
ordered = TRUE, base = "log2", full_name = FALSE, verbose = TRUE,
progress = FALSE)

## S3 method for class 'stars'
calculate_lsm(landscape, level = NULL, metric = NULL,
  name = NULL, type = NULL, what = NULL, directions = 8,
  count_boundary = FALSE, consider_boundary = FALSE, edge_depth = 1,
  classes_max = NULL, neighbourhood = 4, ordered = TRUE,
  base = "log2", full_name = FALSE, verbose = TRUE,
  progress = FALSE)

## S3 method for class 'list'
calculate_lsm(landscape, level = NULL, metric = NULL,
  name = NULL, type = NULL, what = NULL, directions = 8,
  count_boundary = FALSE, consider_boundary = FALSE, edge_depth = 1,
  classes_max = NULL, neighbourhood = 4, ordered = TRUE,
  base = "log2", full_name = FALSE, verbose = TRUE,
  progress = FALSE)

```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
level	Level of metrics. Either 'patch', 'class' or 'landscape' (or vector with combination).
metric	Abbreviation of metrics (e.g. 'area').
name	Full name of metrics (e.g. 'core area')
type	Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').
what	Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. what = c("lsm_c_ca", "lsm_l_ta").
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
count_boundary	Include landscape boundary in edge length
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell
classes_max	Potential maximum number of present classes
neighbourhood	The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4.
ordered	The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE.



base	The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used.
full_name	Should the full names of all functions be included in the tibble.
verbose	Print warning messages.
progress	Print progress report.

## Details

Wrapper to calculate several landscape metrics. The metrics can be specified by the arguments `what`, `level`, `metric`, `name` and/or `type` (combinations of different arguments are possible (e.g. `level = "class"`, `type = "aggregation metric"`). If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, don't specify any of the above arguments.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[list\\_lsm](#)

## Examples

```
## Not run:
calculate_lsm(landscape, progress = TRUE)
calculate_lsm(landscape, what = c("patch", "lsm_c_te", "lsm_l_pr"))
calculate_lsm(landscape, level = c("class", "landscape"), type = "aggregation metric")

## End(Not run)
```

---

check_landscape	<i>Check input landscape</i>
-----------------	------------------------------

---

## Description

Check input landscape

**Usage**

```

check_landscape(landscape, verbose)

## S3 method for class 'RasterLayer'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'stars'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'list'
check_landscape(landscape, verbose = TRUE)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick, Stars or a list of rasterLayers
verbose	Print warning messages.

**Details**

This function extracts basic information about the input landscape. It includes a type of coordinate reference system (crs) - either "geographic", "projected", or NA, units of the coordinate reference system, a class of the input landscape's values and the number of classes found in the landscape.

**Value**

tibble

**Examples**

```

check_landscape(augusta_nlcd)
check_landscape(podlasie_ccilc)
check_landscape(raster::stack(landscape, landscape))

```

---

extract\_lsm

*extract\_lsm*

---

**Description**

Extract metrics

**Usage**

```
extract_lsm(landscape, y, extract_id, metric, name, type, what, directions,
            progress, verbose, ...)
```

```
## S3 method for class 'RasterLayer'
extract_lsm(landscape, y, extract_id = NULL,
            metric = NULL, name = NULL, type = NULL, what = NULL,
            directions = 8, progress = FALSE, verbose = TRUE, ...)
```

```
## S3 method for class 'RasterStack'
extract_lsm(landscape, y, extract_id = NULL,
            metric = NULL, name = NULL, type = NULL, what = NULL,
            directions = 8, progress = FALSE, verbose = TRUE, ...)
```

```
## S3 method for class 'RasterBrick'
extract_lsm(landscape, y, extract_id = NULL,
            metric = NULL, name = NULL, type = NULL, what = NULL,
            directions = 8, progress = FALSE, verbose = TRUE, ...)
```

```
## S3 method for class 'stars'
extract_lsm(landscape, y, extract_id = NULL,
            metric = NULL, name = NULL, type = NULL, what = NULL,
            directions = 8, progress = FALSE, verbose = TRUE, ...)
```

```
## S3 method for class 'list'
extract_lsm(landscape, y, extract_id = NULL,
            metric = NULL, name = NULL, type = NULL, what = NULL,
            directions = 8, progress = FALSE, verbose = TRUE, ...)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
y	2-column matrix with coordinates, SpatialPoints, SpatialLines or sf point geometries.
extract_id	Vector with id of sample points. If not provided, sample points will be labelled 1...n.
metric	Abbreviation of metrics (e.g. 'area').
name	Full name of metrics (e.g. 'core area')
type	Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').
what	Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. what = c("lsm_c_ca", "lsm_l_ta").
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
progress	Print progress report.
verbose	Print warning messages.
...	Arguments passed to calculate_lsm().

**Details**

This functions extracts the metrics of all patches the spatial object(s) *y* (e.g. spatial points) are located within. Only patch level metrics are possible to extract. Please be aware that the output is slightly different to all other *lsm*-function of *landscapemetrics*. Returns a tibble with chosen metrics and the ID of the spatial objects.

**Value**

tibble

**See Also**

[calculate\\_lsm](#)

**Examples**

```
points <- matrix(c(10, 5, 25, 15, 5, 25), ncol = 2, byrow = TRUE)
extract_lsm(landscape, y = points)
extract_lsm(landscape, y = points, type = "aggregation metric")

points_sp <- sp::SpatialPoints(points)
extract_lsm(landscape, y = points_sp, what = "lsm_p_area")

## Not run:
# use lines (works only if rgeos is installed)
x1 <- c(1, 5, 15, 10)
y1 <- c(1, 5, 15, 25)

x2 <- c(10, 25)
y2 <- c(5, 5)

sample_lines <- sp::SpatialLines(list(sp::Lines(list(sp::Line(cbind(x1, y1))),
sp::Line(cbind(x2, y2))), ID = "a")))
extract_lsm(landscape, y = sample_lines, what = "lsm_p_area")

## End(Not run)
```

---

fragstats\_class\_augusta\_nlcd

*Fragstats results for landscapemetrics::augusta\_nlcd (class level)*

---

**Description**

A single tibble for every spatial dataset included in *landscapemetrics* that contains the FRAGSTAT results for every implemented metric on class level.

**Usage**

```
fragstats_class_augusta_nlcd
```

**Format**

A tibble object.

---

```
fragstats_class_landscape
```

*Fragstats results for landscapemetrics::landscape (class level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on class level.

**Usage**

```
fragstats_class_landscape
```

**Format**

A tibble object.

---

```
fragstats_class_podlasie
```

*Fragstats results for landscapemetrics::podlasie (class level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on class level.

**Usage**

```
fragstats_class_podlasie
```

**Format**

A tibble object.

---

`fragstats_landscape_augusta_nlcd`*Fragstats results for landscapemetrics::augusta\_nlcd (landscape level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on landscape level.

**Usage**`fragstats_landscape_augusta_nlcd`**Format**

A tibble object.

---

`fragstats_landscape_landscape`*Fragstats results for landscapemetrics::landscape (landscape level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on landscape level.

**Usage**`fragstats_landscape_landscape`**Format**

A tibble object.

---

`fragstats_landscape_podlasie`*Fragstats results for landscapemetrics::podlasie\_ccilc (landscape level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on landscape level.

**Usage**`fragstats_landscape_podlasie`**Format**

A tibble object.

---

`fragstats_patch_augusta_nlcd`*Fragstats results for landscapemetrics::augusta\_nlcd (patch level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on patch level.

**Usage**`fragstats_patch_augusta_nlcd`**Format**

A tibble object.

---

`fragstats_patch_landscape`*Fragstats results for landscapemetrics::landscape (patch level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on patch level.

**Usage**`fragstats_patch_landscape`**Format**

A tibble object.

---

`fragstats_patch_podlasie`*Fragstats results for landscapemetrics::podlasie (patch level)*

---

**Description**

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on patch level.

**Usage**`fragstats_patch_podlasie`**Format**

A tibble object.



---

get_adjacencies	<i>get_adjacencies</i>
-----------------	------------------------

---

## Description

Fast calculation of adjacencies between classes in a raster

## Usage

```
get_adjacencies(landscape, neighbourhood, what, upper)
```

```
## S3 method for class 'RasterLayer'
get_adjacencies(landscape, neighbourhood = 4,
  what = "full", upper = FALSE)
```

```
## S3 method for class 'RasterStack'
get_adjacencies(landscape, neighbourhood = 4,
  what = "full", upper = FALSE)
```

```
## S3 method for class 'RasterBrick'
get_adjacencies(landscape, neighbourhood = 4,
  what = "full", upper = FALSE)
```

```
## S3 method for class 'stars'
get_adjacencies(landscape, neighbourhood = 4,
  what = "full", upper = FALSE)
```

```
## S3 method for class 'list'
get_adjacencies(landscape, neighbourhood = 4,
  what = "full", upper = FALSE)
```

```
## S3 method for class 'matrix'
get_adjacencies(landscape, neighbourhood = 4,
  what = "full", upper = FALSE)
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
neighbourhood	The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case), 8 (queen's case) or a binary matrix where the ones define the neighbourhood. The default is 4.
what	Which adjacencies to calculate: "full" for a full adjacency matrix, "like" for the diagonal, "unlike" for the off diagonal part of the matrix and "triangle" for a triangular matrix counting adjacencies only once.
upper	Logical value indicating whether the upper triangle of the adjacency matrix should be returned (default FALSE).

## Details

A fast implementation with Rcpp to calculate the adjacency matrix for raster. The adjacency matrix is most often used in landscape metrics to describe the configuration of landscapes, as it is a cellwise count of edges between classes.

The "full" adjacency matrix is double-count method, as it contains the pairwise counts of cells between all classes. The diagonal of this matrix contains the like adjacencies, a count for how many edges a shared in each class with the same class.

The "unlike" adjacencies are counting the cellwise edges between different classes.

## Value

matrix with adjacencies between classes in a raster and between cells from the same class.

## Examples

```
# calculate full adjacency matrix
get_adjacencies(landscape, 4)

# count diagonal neighbour adjacencies
diagonal_matrix <- matrix(c(1, NA, 1,
                           NA, 0, NA,
                           1, NA, 1), 3, 3, byrow = TRUE)
get_adjacencies(landscape, diagonal_matrix)

# equivalent with the raster package:
adjacencies <- raster::adjacent(landscape, 1:raster::ncell(landscape), 4, pairs=TRUE)
table(landscape[adjacencies[,1]], landscape[adjacencies[,2]])
```

---

get_boundaries	<i>get_boundaries</i>
----------------	-----------------------

---

## Description

Get boundary cells of patches

## Usage

```
get_boundaries(landscape, directions, as_NA, consider_boundary,
               return_raster)

## S3 method for class 'RasterLayer'
get_boundaries(landscape, directions = 4,
               as_NA = FALSE, consider_boundary = FALSE, return_raster = TRUE)

## S3 method for class 'RasterStack'
get_boundaries(landscape, directions = 4,
```

```

as_NA = FALSE, consider_boundary = FALSE, return_raster = TRUE)

## S3 method for class 'RasterBrick'
get_boundaries(landscape, directions = 4,
  as_NA = FALSE, consider_boundary = FALSE, return_raster = TRUE)

## S3 method for class 'stars'
get_boundaries(landscape, directions = 4,
  as_NA = FALSE, consider_boundary = FALSE, return_raster = TRUE)

## S3 method for class 'list'
get_boundaries(landscape, directions = 4, as_NA = FALSE,
  consider_boundary = FALSE, return_raster = TRUE)

## S3 method for class 'matrix'
get_boundaries(landscape, directions = 4,
  as_NA = FALSE, consider_boundary = FALSE, return_raster = FALSE)

```

### Arguments

landscape	RasterLayer or matrix.
directions	Rook's case (4 neighbours) or queen's case (8 neighbours) should be used as neighbourhood rule
as_NA	If true, non-boundary cells are labeled NA
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as edge
return_raster	If false, matrix is returned

### Details

All boundary/edge cells are labeled 1, all non-boundary cells 0. NA values are not changed. Boundary cells are defined as cells that neighbour either a NA cell or a cell with a different value than itself. Non-boundary cells only neighbour cells with the same value than themselves.

### Value

List with RasterLayer or matrix

### Examples

```

class_1 <- get_patches(landscape, class = 1)[[1]]

get_boundaries(class_1)
get_boundaries(class_1, return_raster = FALSE)

class_1_matrix <- raster::as.matrix(class_1)
get_boundaries(class_1_matrix, return_raster = FALSE)

```

---

```
get_circumscribingcircle
    get_circumscribingcircle
```

---

## Description

Diameter of the circumscribing circle around patches

## Usage

```
get_circumscribingcircle(landscape, resolution_x, resolution_y)

## S3 method for class 'RasterLayer'
get_circumscribingcircle(landscape,
    resolution_x = NULL, resolution_y = NULL)

## S3 method for class 'RasterStack'
get_circumscribingcircle(landscape,
    resolution_x = NULL, resolution_y = NULL)

## S3 method for class 'RasterBrick'
get_circumscribingcircle(landscape,
    resolution_x = NULL, resolution_y = NULL)

## S3 method for class 'stars'
get_circumscribingcircle(landscape, resolution_x = NULL,
    resolution_y = NULL)

## S3 method for class 'list'
get_circumscribingcircle(landscape, resolution_x = NULL,
    resolution_y = NULL)

## S3 method for class 'matrix'
get_circumscribingcircle(landscape, resolution_x = NULL,
    resolution_y = NULL)
```

## Arguments

landscape	RasterLayer or matrix (with x, y, id columns)
resolution_x	Resolution of the landscape (only needed if matrix as input is used)
resolution_y	Resolution of the landscape (only needed if matrix as input is used)

## Details

The diameter of the smallest circumscribing circle around a patch in the landscape is based on the maximum distance between the corners of each cell. This ensures that all cells of the patch are

included in the patch. All patches need an unique ID (see [get\\_patches](#)). If one uses this functions with a matrix the resolution of the underlying data must be provided.

### Examples

```
# get patches for class 1 from testdata as raster
class_1 <- get_patches(landscape, class = 1)[[1]]

# calculate the minimum circumscribing circle of each patch in class 1
get_circumscribingcircle(class_1)

# do the same with a 3 column matrix (x, y, id)
class_1_matrix <- raster::rasterToPoints(class_1)
get_circumscribingcircle(class_1_matrix, resolution_x = 1, resolution_y = 1)
```

---

get\_nearestneighbour    *get\_nearestneighbour*

---

### Description

Euclidean distance to nearest neighbour

### Usage

```
get_nearestneighbour(landscape)

## S3 method for class 'RasterLayer'
get_nearestneighbour(landscape)

## S3 method for class 'RasterStack'
get_nearestneighbour(landscape)

## S3 method for class 'RasterBrick'
get_nearestneighbour(landscape)

## S3 method for class 'stars'
get_nearestneighbour(landscape)

## S3 method for class 'list'
get_nearestneighbour(landscape)

## S3 method for class 'matrix'
get_nearestneighbour(landscape)
```

### Arguments

landscape            RasterLayer or matrix (with x,y,id columns)

## Details

Fast and memory safe Rcpp implementation for calculating the minimum Euclidean distances to the nearest patch of the same class in a raster or matrix. All patches need an unique ID (see [get\\_patches](#)).

## References

Based on RCpp code of Florian Privé <florian.prive.21@gmail.com>

## Examples

```
# get patches for class 1 from testdata as raster
class_1 <- get_patches(landscape,1)[[1]]

# calculate the distance between patches
get_nearestneighbour(class_1)

# do the same with a 3 column matrix (x, y, id)
class_1_matrix <- raster::rasterToPoints(class_1)
get_nearestneighbour(class_1_matrix)
```

---

get\_patches

get\_patches

---

## Description

Connected components labeling to derive patches in a landscape.

## Usage

```
get_patches(landscape, class, directions, to_disk, return_raster)

## S3 method for class 'RasterLayer'
get_patches(landscape, class = "all",
  directions = 8, to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE)

## S3 method for class 'RasterStack'
get_patches(landscape, class = "all",
  directions = 8, to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE)

## S3 method for class 'RasterBrick'
get_patches(landscape, class = "all",
  directions = 8, to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE)
```

```

## S3 method for class 'stars'
get_patches(landscape, class = "all", directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE)

## S3 method for class 'list'
get_patches(landscape, class = "all", directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE)

## S3 method for class 'matrix'
get_patches(landscape, class = "all", directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE)

```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
class	Either "all" (default) for every class in the raster, or specify class value. See Details.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
to_disk	Logical argument, if FALSE results of get_patches are hold in memory. If true, get_patches writes temporary files and hence, does not hold everything in memory. Can be set with a global option, e.g. option(to_disk = TRUE). See Details.
return_raster	If false, matrix is returned

### Details

Searches for connected patches (neighbouring cells of the same class *i*). The 8-neighbours rule ('queen's case) or 4-neighbours rule (rook's case) is used. Returns a list with raster. For each class the connected patches have the value 1 - *n*. All cells not belonging to the class are NA. The underlying C code comes from the SDMTools package (VanDerWal *et al.* 2014) and we appreciate their effort for implementing this efficient connected labeling algorithm.

Landscape metrics rely on the delineation of patches. Hence, get\_patches is heavily used in **landscapemetrics**. As raster can be quite big, the fact that get\_patches creates a copy of the raster for each class in a landscape becomes a burden for computer memory. Hence, the argument *to\_disk* allows to store the results of the connected labeling algorithm on disk. Furthermore, this option can be set globally, so that every function that internally uses get\_patches can make use of that.

### Value

List

## References

- VanDerWal, J., Falconi, L., Januchowski, S., Shoo, L., and Storlie, C. 2014. SDMTTools: Species Distribution Modelling Tools: Tools for processing data associated with species distribution modelling exercises. R package version 1.1-221. <https://CRAN.R-project.org/package=SDMTTools>
- Chang, F., C.-J. Chen, and C.-J. Lu. 2004. A linear-time component-labeling algorithm using contour tracing technique. Comput. Vis. Image Underst. 93:206-220.

## Examples

```
# check for patches of class 1
patched_raster <- get_patches(landscape, class = 1)

# count patches
length(raster::unique(patch_raster[[1]]))

# check for patches of every class
patched_raster <- get_patches(landscape)
```

---

<code>get_unique_values</code>	<i>get_unique_values</i>
--------------------------------	--------------------------

---

## Description

This function returns the unique values of an object.

## Usage

```
get_unique_values(x, simplify, verbose)

## S3 method for class 'numeric'
get_unique_values(x, simplify = FALSE,
  verbose = TRUE)

## S3 method for class 'matrix'
get_unique_values(x, simplify = FALSE, verbose = TRUE)

## S3 method for class 'RasterLayer'
get_unique_values(x, simplify = FALSE,
  verbose = TRUE)

## S3 method for class 'list'
get_unique_values(x, simplify = FALSE, verbose = TRUE)

## S3 method for class 'RasterStack'
get_unique_values(x, simplify = FALSE,
  verbose = TRUE)
```



```
## S3 method for class 'RasterBrick'
get_unique_values(x, simplify = FALSE,
  verbose = TRUE)

## S3 method for class 'stars'
get_unique_values(x, simplify = FALSE, verbose = TRUE)
```

### Arguments

<code>x</code>	vector, matrix or Raster* object
<code>simplify</code>	If true, a vector will be returned instead of a list for 1-dimensional input
<code>verbose</code>	If true, warning messages are printend

### Details

Fast and memory friendly Rcpp implementation to find the unique values of an object.

### Examples

```
get_unique_values(landscape)

landscape_stack <- raster::stack(landscape, landscape, landscape)
get_unique_values(landscape_stack)

landscape_matrix <- raster::as.matrix(landscape)
get_unique_values(landscape_matrix)

x_vec <- c(1, 2, 1, 1, 2, 2)
get_unique_values(x_vec)

landscape_list <- list(landscape, landscape_matrix, x_vec)
get_unique_values(landscape_list)
```

---

landscape

*Example map (random cluster neutral landscape model).*

---

### Description

An example map to show landscapetools functionality generated with the `nlm_randomcluster()` algorithm.

### Usage

```
landscape
```

**Format**

A raster layer object.

**Source**

Simulated neutral landscape model with R. <https://github.com/ropensci/NLMR/>

---

landscapemetrics

*landscapemetrics*

---

**Description**

Calculates landscape metrics for categorical landscape patterns in a tidy workflow. 'landscapemetrics' reimplements the most common metrics from FRAGSTATS and new ones from the current literature on landscape metrics. This package supports raster spatial objects and takes RasterLayer, RasterStacks, RasterBricks or lists of RasterLayer from the 'raster' package as input arguments. It further provides utility functions to visualize patches, select metrics and building blocks to develop new metrics.

**Author(s)**

**Maintainer:** Maximillian H.K. Hesselbarth <maximilian.hesselbarth@uni-goettingen.de> (0000-0003-1125-9918)

Authors:

- Marco Sciaini <sciaini.marco@gmail.com> (0000-0002-3042-5435)
- Jakub Nowosad <nowosad.jakub@gmail.com> (0000-0002-1057-3721)
- Sebastian Hanss (0000-0002-3990-4897)

Other contributors:

- Laura J. Graham (Input on package structure) [contributor]
- Jeffrey Hollister (Input on package structure) [contributor]
- Kimberly A. With (Input on package structure) [contributor]
- Florian Privé (Original author of underlying C++ code for get\_nearestneighbour() function) [contributor]
- Jeremy VanDerWal (Original author of underlying C code for get\_patches() function) [contributor]
- Matt Strimas-Mackey (Bugfix in sample\_metrics()) [contributor]

**See Also**

Useful links:

- <https://r-spatialecology.github.io/landscapemetrics/>
- Report bugs at <https://github.com/r-spatialecology/landscapemetrics/issues>

---

list_lsm	<i>List landscape metrics</i>
----------	-------------------------------

---

## Description

List landscape metrics

## Usage

```
list_lsm(level = NULL, metric = NULL, name = NULL, type = NULL,
         what = NULL, simplify = FALSE, verbose = TRUE)
```

## Arguments

level	Level of metrics. Either 'patch', 'class' or 'landscape' (or vector with combination).
metric	Abbreviation of metrics (e.g. 'area').
name	Full name of metrics (e.g. 'core area')
type	Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').
what	Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. <code>what = c("lsm_c_ca", "lsm_l_ta")</code> .
simplify	If true, function names are returned as vector.
verbose	Print warning messages

## Details

List all available landscape metrics depending on the provided filter arguments. If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, use simply `list_lsm()`. For all arguments with exception of the `what` argument, it is also possible to use a negative subset, i.e. all metrics **but** the selected ones. Therefore, simply use e.g. `level = "-patch"`. Furthermore, it is possible to only get a vector with all function names instead of the full tibble.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**Examples**

```
list_lsm(level = c("patch", "landscape"), type = "aggregation metric")
list_lsm(level = "-patch", type = "area and edge metric")
list_lsm(metric = "area", simplify = TRUE)

list_lsm(metric = "area", what = "lsm_p_shape")
list_lsm(metric = "area", what = c("patch", "lsm_l_ta"))
list_lsm(what = c("lsm_c_tca", "lsm_l_ta"))
```

---

```
lsm_abbreviations_names
```

*Tibble of abbreviations coming from FRAGSTATS*

---

**Description**

A single tibble for every abbreviation of every metric that is reimplemented in landscapemetrics and its corresponding full name in the literature.

**Usage**

```
lsm_abbreviations_names
```

**Format**

A tibble object.

**Details**

Can be used after calculating the metric(s) with a join to have a more readable results tibble or for visualizing your results.

**Examples**

```
patch_area <- lsm_p_area(landscape)
patch_area <- merge(x = patch_area, y = lsm_abbreviations_names, by = c("level", "metric"))
```

---

lsm_c_ai	<i>AI (class level)</i>
----------	-------------------------

---

### Description

Aggregation index (Aggregation metric)

### Usage

```
lsm_c_ai(landscape)

## S3 method for class 'RasterLayer'
lsm_c_ai(landscape)

## S3 method for class 'RasterStack'
lsm_c_ai(landscape)

## S3 method for class 'RasterBrick'
lsm_c_ai(landscape)

## S3 method for class 'stars'
lsm_c_ai(landscape)

## S3 method for class 'list'
lsm_c_ai(landscape)
```

### Arguments

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers

### Details

$$AI = \left[ \frac{g_{ii}}{max - g_{ii}} \right] (100)$$

where  $g_{ii}$  is the number of like adjacencies based on the single-count method and  $max - g_{ii}$  is the classwise maximum number of like adjacencies of class i.

AI is an 'Aggregation metric'. It equals the number of like adjacencies divided by the theoretical maximum possible number of like adjacencies for that class. The metric is based on the adjacency matrix and the single-count method.

**Units:** Percent

**Range:**  $0 \leq AI \leq 100$

**Behaviour:** Equals 0 for maximally disaggregated and 100 for maximally aggregated classes.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

He, H. S., DeZonia, B. E., & Mladenoff, D. J. 2000. An aggregation index (AI) to quantify spatial patterns of landscapes. *Landscape ecology*, 15(7), 591-601.

**See Also**

[lsm\\_l\\_ai](#)

**Examples**

```
lsm_c_ai(landscape)
```

---

lsm\_c\_area\_cv

*AREA\_CV (class level)*


---

**Description**

Coefficient of variation of patch area (Area and edge metric)

**Usage**

```
lsm_c_area_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_area_cv(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$AREA_{CV} = cv(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares.

AREA\_CV is an 'Area and Edge metric'. The metric summarises each class as the Coefficient of variation of all patch areas belonging to class i. The metric describes the differences among patches of the same class i in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** AREA\_CV >= 0

**Behaviour:** Equals AREA\_CV = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_area, cv,](#)  
[lsm\\_c\\_area\\_mn, lsm\\_c\\_area\\_sd,](#)  
[lsm\\_l\\_area\\_mn, lsm\\_l\\_area\\_sd, lsm\\_l\\_area\\_cv](#)

**Examples**

```
lsm_c_area_cv(landscape)
```

---

lsm_c_area_mn	<i>AREA_MN (class level)</i>
---------------	------------------------------

---

### Description

Mean of patch area (Area and edge metric)

### Usage

```
lsm_c_area_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_area_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$AREA_{MN} = mean(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares

AREA\_MN is an 'Area and Edge metric'. The metric summarises each class as the mean of all patch areas belonging to class i. The metric is a simple way to describe the composition of the landscape. Especially together with the total class area ([lsm\\_c\\_ca](#)), it can also give an idea of patch structure (e.g. many small patches vs. few large patches).

**Units:** Hectares

**Range:**  $AREA_{MN} > 0$

**Behaviour:** Approaches  $AREA_{MN} = 0$  if all patches are small. Increases, without limit, as the patch areas increase.



**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_area](#), [mean](#),  
[lsm\\_c\\_area\\_cv](#), [lsm\\_c\\_area\\_sd](#),  
[lsm\\_l\\_area\\_mn](#), [lsm\\_l\\_area\\_sd](#), [lsm\\_l\\_area\\_cv](#)

**Examples**

```
lsm_c_area_mn(landscape)
```

---

lsm_c_area_sd	<i>AREA_SD (class level)</i>
---------------	------------------------------

---

**Description**

Standard deviation of patch area (Area and edge metric)

**Usage**

```
lsm_c_area_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_division(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$AREA_{SD} = sd(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares.

AREA\_SD is an 'Area and Edge metric'. The metric summarises each class as the standard deviation of all patch areas belonging to class i. The metric describes the differences among patches of the same class i in the landscape.

**Units:** Hectares

**Range:** AREA\_SD >= 0

**Behaviour:** Equals AREA\_SD = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_area](#), [sd](#),  
[lsm\\_c\\_area\\_mn](#), [lsm\\_c\\_area\\_cv](#),  
[lsm\\_l\\_area\\_mn](#), [lsm\\_l\\_area\\_sd](#), [lsm\\_l\\_area\\_cv](#)

**Examples**

```
lsm_c_area_sd(landscape)
```

---

lsm_c_ca	CA (class level)
----------	------------------

---

### Description

Total (class) area (Area and edge metric)

### Usage

```
lsm_c_ca(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_ca(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CA = \text{sum}(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares.

CA is an 'Area and edge metric' and a measure of composition. The total (class) area sums the area of all patches belonging to class i. It shows if the landscape is e.g. dominated by one class or if all classes are equally present. CA is an absolute measure, making comparisons among landscapes with different total areas difficult.

**Units:** Hectares

**Range:**  $CA > 0$

**Behaviour:** Approaches  $CA > 0$  as the patch areas of class i become small. Increases, without limit, as the patch areas of class i become large.  $CA = TA$  if only one class is present.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_area](#), [sum](#),  
[lsm\\_l\\_ta](#)

**Examples**

```
lsm_c_ca(landscape)
```

---

lsm_c_cai_cv	<i>CAI_CV (class level)</i>
--------------	-----------------------------

---

**Description**

Coefficient of variation of core area index (Core area metric)

**Usage**

```
lsm_c_cai_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$CAI_{CV} = cv(CAI[patch_{ij}])$$

where  $CAI[patch_{ij}]$  is the core area index of each patch.

CAI\_CV is a 'Core area metric'. The metric summarises each class as the Coefficient of variation of the core area index of all patches belonging to class i. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among patches of the same class i in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** Percent

**Range:** CAI\_CV >= 0

**Behaviour:** Equals CAI\_CV = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of the core area indices increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_cai](#), [cv](#),  
[lsm\\_c\\_cai\\_mn](#), [lsm\\_c\\_cai\\_sd](#),  
[lsm\\_l\\_cai\\_mn](#), [lsm\\_l\\_cai\\_sd](#), [lsm\\_l\\_cai\\_cv](#)

**Examples**

```
lsm_c_cai_cv(landscape)
```

---

lsm_c_cai_mn	<i>CAI<sub>MN</sub></i> (class level)
--------------	---------------------------------------

---

## Description

Mean of core area index (Core area metric)

## Usage

```
lsm_c_cai_mn(landscape, directions, consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
lsm_c_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
lsm_c_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
lsm_c_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_c_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
lsm_c_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

## Details

$$CAI_{MN} = \text{mean}(CAI[\text{patch}_{ij}])$$

where  $CAI[patch_{ij}]$  is the core area index of each patch.

CAI\_MN is a 'Core area metric'. The metric summarises each class as the mean of the core area index of all patches belonging to class  $i$ . The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case).

**Units:** Percent

**Range:** CAI\_MN  $\geq 0$

**Behaviour:** Equals CAI\_MN = 0 if CAI = 0 for all patches. Increases, without limit, as the core area indices increase.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[lsm\\_p\\_cai](#), [mean](#),  
[lsm\\_c\\_cai\\_sd](#), [lsm\\_c\\_cai\\_cv](#),  
[lsm\\_l\\_cai\\_mn](#), [lsm\\_l\\_cai\\_sd](#), [lsm\\_l\\_cai\\_cv](#)

## Examples

```
lsm_c_cai_mn(landscape)
```

---

lsm_c_cai_sd	<i>CAI_SD (class level)</i>
--------------	-----------------------------

---

## Description

Standard deviation of core area index (Core area metric)

**Usage**

```

lsm_c_cai_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_cai_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_cai_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_cai_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_cai_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_cai_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$CAI_{SD} = sd(CAI[patch_{ij}])$$

where  $CAI[patch_{ij}]$  is the core area index of each patch.

CAI\_SD is a 'Core area metric'. The metric summarises each class as the standard deviation of the core area index of all patches belonging to class i. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among patches of the same class i in the landscape.

**Units:** Percent



**Range:** CAI\_SD  $\geq 0$

**Behaviour:** Equals CAI\_SD = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of core area indices increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_cai](#), [sd](#),  
[lsm\\_c\\_cai\\_mn](#), [lsm\\_c\\_cai\\_cv](#),  
[lsm\\_l\\_cai\\_mn](#), [lsm\\_l\\_cai\\_sd](#), [lsm\\_l\\_cai\\_cv](#)

### Examples

```
lsm_c_cai_sd(landscape)
```

---

lsm_c_circle_cv	<i>CIRCLE_CV (Class level)</i>
-----------------	--------------------------------

---

### Description

Coefficient of variation of related circumscribing circle (Shape metric)

### Usage

```
lsm_c_circle_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_circle_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_circle_cv(landscape, directions = 8)
```

```
## S3 method for class 'list'
lsm_c_circle_cv(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CIRCLE_{CV} = cv(CIRCLE[patch_{ij}])$$

where  $CIRCLE[patch_{ij}]$  is the related circumscribing circle of each patch.

$CIRCLE_{CV}$  is a 'Shape metric' and summarises each class as the Coefficient of variation of the related circumscribing circle of all patches belonging to class  $i$ .  $CIRCLE$  describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch.  $CIRCLE_{CV}$  describes the differences among patches of the same class  $i$  in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** None

**Range:**  $CIRCLE_{CV} \geq 0$

**Behaviour:** Equals  $CIRCLE_{CV}$  if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landscape Ecology* 7: 291-302.

### See Also

[lsm\\_p\\_circle](#), [mean](#),  
[lsm\\_c\\_circle\\_mn](#), [lsm\\_c\\_circle\\_sd](#),  
[lsm\\_l\\_circle\\_mn](#), [lsm\\_l\\_circle\\_sd](#), [lsm\\_l\\_circle\\_cv](#)

### Examples

```
lsm_c_circle_cv(landscape)
```

---

lsm_c_circle_mn	<i>CIRCLE_MN</i> (Class level)
-----------------	--------------------------------

---

### Description

Mean of related circumscribing circle (Shape metric)

### Usage

```
lsm_c_circle_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_circle_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CIRCLE_{MN} = mean(CIRCLE[patch_{ij}])$$

where  $CIRCLE[patch_{ij}]$  is the related circumscribing circle of each patch.

CIRCLE\_MN is a 'Shape metric' and summarises each class as the mean of the related circumscribing circle of all patches belonging to class i. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch.

**Units:** None

**Range:** CIRCLE\_MN > 0

**Behaviour:** Approaches CIRCLE\_MN = 0 if the related circumscribing circle of all patches is small. Increases, without limit, as the related circumscribing circles increase.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landscape Ecology* 7: 291-302.

**See Also**

[lsm\\_p\\_circle](#), [mean](#),  
[lsm\\_c\\_circle\\_sd](#), [lsm\\_c\\_circle\\_cv](#),  
[lsm\\_l\\_circle\\_mn](#), [lsm\\_l\\_circle\\_sd](#), [lsm\\_l\\_circle\\_cv](#)

**Examples**

```
lsm_c_circle_mn(landscape)
```

---

lsm_c_circle_sd	<i>CIRCLE_SD (Class level)</i>
-----------------	--------------------------------

---

**Description**

Standard deviation of related circumscribing circle (Shape metric)

**Usage**

```
lsm_c_circle_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_circle_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$CIRCLE_{SD} = sd(CIRCLE[patch_{ij}])$$

where  $CIRCLE[patch_{ij}]$  is the related circumscribing circle of each patch.

CIRCLE\_SD is a 'Shape metric' and summarises each class as the standard deviation of the related circumscribing circle of all patches belonging to class i. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch. The metric describes the differences among patches of the same class i in the landscape.

**Units:** None

**Range:** CIRCLE\_SD  $\geq 0$

**Behaviour:** Equals CIRCLE\_SD if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

**Value**

tibble

**References**

- McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>
- Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

**See Also**

[lsm\\_p\\_circle](#), [mean](#),  
[lsm\\_c\\_circle\\_mn](#), [lsm\\_c\\_circle\\_cv](#),  
[lsm\\_l\\_circle\\_mn](#), [lsm\\_l\\_circle\\_sd](#), [lsm\\_l\\_circle\\_cv](#)

**Examples**

```
lsm_c_circle_sd(landscape)
```

---

lsm_c_clumpy	<i>CLUMPY (class level)</i>
--------------	-----------------------------

---

## Description

Clumpiness index (Aggregation metric)

## Usage

```
lsm_c_clumpy(landscape)

## S3 method for class 'RasterLayer'
lsm_c_clumpy(landscape)

## S3 method for class 'RasterStack'
lsm_c_clumpy(landscape)

## S3 method for class 'RasterBrick'
lsm_c_clumpy(landscape)

## S3 method for class 'stars'
lsm_c_clumpy(landscape)

## S3 method for class 'list'
lsm_c_clumpy(landscape)
```

## Arguments

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers

## Details

$$Given G_i = \left( \frac{g_{ii}}{\left( \sum_{k=1}^m g_{ik} \right) - mine_i} \right)$$

$$CLUMPY = \left[ \frac{G_i - P_i}{P_i} \text{ for } G_i < P_i \& P_i < .5; \text{ else } \frac{G_i - P_i}{1 - P_i} \right]$$

where  $g_{ii}$  is the number of like adjacencies,  $g_{ik}$  is the classwise number of all adjacencies including the focal class,  $mine_i$  is the minimum perimeter of the total class in terms of cell surfaces assuming total clumping and  $P_i$  is the proportion of landscape occupied by each class.

CLUMPY is an 'Aggregation metric'. It equals the proportional deviation of the proportion of like adjacencies involving the corresponding class from that expected under a spatially random distribution. The metric is based on the adjacency matrix and the double-count method.

**Units:** None

, directions = directions

**Range:** -1 <= CLUMPY <= 1

**Behaviour:** Equals -1 for maximally disaggregated, 0 for randomly distributed and 1 for maximally aggregated classes.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## Examples

```
lsm_c_clumpy(landscape)
```

---

lsm_c_cohesion	<i>COHESION (class level)</i>
----------------	-------------------------------

---

## Description

Patch Cohesion Index (Aggregation metric)

## Usage

```
lsm_c_cohesion(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_cohesion(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$COHESION = 1 - \left( \frac{\sum_{j=1}^n p_{ij}}{\sum_{j=1}^n p_{ij} \sqrt{a_{ij}}} \right) * \left( 1 - \frac{1}{\sqrt{Z}} \right)^{-1} * 100$$

where  $p_{ij}$  is the perimeter in meters,  $a_{ij}$  is the area in square meters and  $Z$  is the number of cells.

COHESION is an 'Aggregation metric'. It characterises the connectedness of patches belonging to class  $i$ . It can be used to assess if patches of the same class are located aggregated or rather isolated and thereby COHESION gives information about the configuration of the landscape.

**Units:** Percent

**Ranges:**  $0 < COHESION < 100$

**Behaviour:** Approaches COHESION = 0 if patches of class  $i$  become more isolated. Increases if patches of class  $i$  become more aggregated.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Schumaker, N. H. 1996. Using landscape indices to predict habitat connectivity. *Ecology*, 77(4), 1210-1225.

**See Also**

[lsm\\_p\\_perim](#), [lsm\\_p\\_area](#),  
[lsm\\_l\\_cohesion](#)

**Examples**

```
lsm_c_cohesion(landscape)
```



---

lsm_c_contig_cv	<i>CONTIG_CV (class level)</i>
-----------------	--------------------------------

---

## Description

Coefficient of variation of Contiguity index (Shape metric)

## Usage

```
lsm_c_contig_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_contig_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_contig_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_contig_cv(landscape, directions = 8)
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$CONTIG_{CV} = cv(CONTIG[patch_{ij}])$$

where  $CONTIG[patch_{ij}]$  is the contiguity of each patch.

CONTIG\_CV is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. CONTIG\_CV asses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG\_CV >= 0

**Behaviour:** CONTIG\_CV = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

**See Also**

[lsm\\_p\\_contig](#), [lsm\\_c\\_contig\\_mn](#), [lsm\\_c\\_contig\\_cv](#),  
[lsm\\_l\\_contig\\_mn](#), [lsm\\_l\\_contig\\_sd](#), [lsm\\_l\\_contig\\_cv](#)

**Examples**

```
lsm_c_contig_cv(landscape)
```

---

lsm_c_contig_mn	<i>CONTIG_MN (class level)</i>
-----------------	--------------------------------

---

**Description**

Mean of Contiguity index (Shape metric)

**Usage**

```
lsm_c_contig_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
```

```
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_contig_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CONTIG_{MN} = mean(CONTIG[patch_{ij}])$$

where  $CONTIG[patch_{ij}]$  is the contiguity of each patch.

CONTIG\_MN is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. CONTIG\_MN assesses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rook case result in larger contiguity index values.

**Units:** None

**Range:**  $0 \leq CONTIG\_MN \leq 1$

**Behaviour:** CONTIG equals the mean of the contiguity index on class level for all patches.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

**See Also**

[lsm\\_p\\_contig](#), [lsm\\_c\\_contig\\_sd](#), [lsm\\_c\\_contig\\_cv](#),  
[lsm\\_l\\_contig\\_mn](#), [lsm\\_l\\_contig\\_sd](#), [lsm\\_l\\_contig\\_cv](#)

**Examples**

```
lsm_c_contig_mn(landscape)
```

---

lsm_c_contig_sd	<i>CONTIG_SD (class level)</i>
-----------------	--------------------------------

---

**Description**

Standard deviation of Contiguity index (Shape metric)

**Usage**

```
lsm_c_contig_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_contig_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$CONTIG_{SD} = sd(CONTIG[patch_{ij}])$$

where  $CONTIG[patch_{ij}]$  is the contiguity of each patch.

CONTIG\_SD is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. CONTIG\_SD assesses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rook case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG\_CV  $\geq 0$

**Behaviour:** CONTIG\_SD = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

## Value

tibble

## References

- McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>
- LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

## See Also

[lsm\\_p\\_contig](#), [lsm\\_c\\_contig\\_mn](#), [lsm\\_c\\_contig\\_cv](#),  
[lsm\\_l\\_contig\\_mn](#), [lsm\\_l\\_contig\\_sd](#), [lsm\\_l\\_contig\\_cv](#)

## Examples

```
lsm_c_contig_sd(landscape)
```

---

lsm_c_core_cv	<i>CORE_CV (class level)</i>
---------------	------------------------------

---

## Description

Coefficient of variation of core area (Core area metric)

## Usage

```
lsm_c_core_cv(landscape, directions, consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
lsm_c_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
lsm_c_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
lsm_c_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_c_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
lsm_c_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

## Details

$$CORE_{CV} = cv(CORE[patch_{ij}])$$

where  $CORE[patch_{ij}]$  is the core area in square meters of each patch.

CORE\_CV is a 'Core area metric'. It equals the Coefficient of variation of the core area of each patch belonging to class  $i$ . The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among patches of the same class  $i$  in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** CORE\_CV  $\geq 0$

**Behaviour:** Equals CORE\_CV = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[lsm\\_p\\_core\\_cv](#),  
[lsm\\_c\\_core\\_mn](#), [lsm\\_c\\_core\\_sd](#),  
[lsm\\_l\\_core\\_mn](#), [lsm\\_l\\_core\\_sd](#), [lsm\\_l\\_core\\_cv](#)

## Examples

```
lsm_c_core_cv(landscape)
```

---

lsm_c_core_mn	<i>CORE_MN (class level)</i>
---------------	------------------------------

---

## Description

Mean of core area (Core area metric)

**Usage**

```

lsm_c_core_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$CORE_{MN} = mean(CORE[patch_{ij}])$$

where  $CORE[patch_{ij}]$  is the core area in square meters of each patch.

CORE\_MN is a 'Core area metric' and equals the mean of core areas of all patches belonging to class i. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case).

**Units:** Hectares

**Range:** CORE\_MN  $\geq 0$

**Behaviour:** Equals CORE\_MN = 0 if CORE = 0 for all patches. Increases, without limit, as the core area indices increase.



**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_core](#), [mean](#),  
[lsm\\_c\\_core\\_sd](#), [lsm\\_c\\_core\\_cv](#),  
[lsm\\_l\\_core\\_mn](#), [lsm\\_l\\_core\\_sd](#), [lsm\\_l\\_core\\_cv](#)

**Examples**

```
lsm_c_core_mn(landscape)
```

---

lsm_c_core_sd	<i>CORE_SD (class level)</i>
---------------	------------------------------

---

**Description**

Standard deviation patch core area (class level)

**Usage**

```
lsm_c_core_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
lsm_c_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$CORE_{SD} = sd(CORE[patch_{ij}])$$

where  $CORE[patch_{ij}]$  is the core area in square meters of each patch.

CORE\_SD is a 'Core area metric'. It equals the standard deviation of the core area of each patch belonging to class i. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among patches of the same class i in the landscape.

**Units:** Hectares

**Range:** CORE\_SD >= 0

**Behaviour:** Equals CORE\_SD = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_core](#), [sd](#),  
[lsm\\_c\\_core\\_mn](#), [lsm\\_c\\_core\\_cv](#),  
[lsm\\_l\\_core\\_mn](#), [lsm\\_l\\_core\\_sd](#), [lsm\\_l\\_core\\_cv](#)

**Examples**

```
lsm_c_core_sd(landscape)
```

---

lsm_c_cpland	<i>CPLAND (class level)</i>
--------------	-----------------------------

---

**Description**

Core area percentage of landscape (Core area metric)

**Usage**

```
lsm_c_cpland(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_cpland(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_cpland(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_cpland(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_cpland(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_cpland(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

Details

$$CPLAND = (\frac{\sum_{j=1}^n a_{ij}^{core}}{A}) * 100$$

where  $a_{ij}^{core}$  is the core area in square meters and  $A$  is the total landscape area in square meters.

CPLAND is a 'Core area metric'. It is the percentage of core area of class i in relation to the total landscape area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). Because CPLAND is a relative measure, it is comparable among landscapes with different total areas.

**Units:** Percentage

**Range:**  $0 \leq CPLAND < 100$

**Behaviour:** Approaches  $CPLAND = 0$  if  $CORE = 0$  for all patches. Increases as the amount of core area increases, i.e. patches become larger while being rather simple in shape.

Value

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_p\\_core](#) and [lsm\\_l\\_ta](#)

Examples

```
lsm_c_cpland(landscape)
```

---

lsm_c_dcad	<i>DCAD (class level)</i>
------------	---------------------------

---

Description

Disjunct core area density (core area metric)

**Usage**

```

lsm_c_dcad(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$DCAD = \left( \frac{\sum_{j=1}^n n_{ij}^{core}}{A} \right) * 10000 * 100$$

where  $n_{ij}^{core}$  is the number of disjunct core areas and  $A$  is the total landscape area in square meters.

DCAD is a 'Core area metric'. It equals the number of disjunct core areas per 100 ha relative to the total area. A disjunct core area is a 'patch within the patch' containing only core cells. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric is relative and therefore comparable among landscapes with different total areas.

**Units:** Number per 100 hectares

**Range:** DCAD >= 0

**Behaviour:** Equals DCAD = 0 when DCORE = 0, i.e. no patch of class i contains a disjunct core area. Increases, without limit, as disjunct core areas become more present, i.e. patches becoming larger and less complex.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_ndca](#), [lsm\\_l\\_ta](#),  
[lsm\\_l\\_dcad](#)

**Examples**

```
lsm_c_dcad(landscape)
```

---

lsm_c_dcore_cv	<i>DCORE_CV (class level)</i>
----------------	-------------------------------

---

**Description**

Coefficient of variation number of disjunct core areas (Core area metric)

**Usage**

```
lsm_c_dcore_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_c_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$DCORE_{CV} = cv(NCORE[patch_{ij}])$$

where  $NCORE[patch_{ij}]$  is the number of core areas.

$DCORE_{CV}$  is an 'Core area metric'. It summarises each class as the Coefficient of variation of all patch areas belonging to class  $i$ . A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case).  $NCORE$  counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among patches of the same class  $i$  in the landscape and is easily comparable because it is scaled to the mean.

**Units:** None

**Range:**  $DCORE_{CV} \geq 0$

**Behaviour:** Equals  $DCORE_{CV} = 0$  if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_p\\_ncore](#), [cv](#),  
[lsm\\_c\\_dcore\\_mn](#), [lsm\\_c\\_dcore\\_sd](#),  
[lsm\\_l\\_dcore\\_mn](#), [lsm\\_l\\_dcore\\_sd](#), [lsm\\_l\\_dcore\\_cv](#)

Examples

```
lsm_c_dcore_cv(landscape)
```

---

lsm_c_dcore_mn	<i>DCORE_MN (class level)</i>
----------------	-------------------------------

---

Description

Mean number of disjunct core areas (Core area metric)

Usage

```
lsm_c_dcore_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).



consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

## Details

$$DCORE_{MN} = mean(NCORE[patch_{ij}])$$

where  $NCORE[patch_{ij}]$  is the number of core areas.

DCORE\_MN is an 'Core area metric'. It summarises each class as the mean of all patch areas belonging to class i. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells.

**Units:** None

**Range:**  $DCORE_{MN} > 0$

**Behaviour:** Equals  $DCORE_{MN} = 0$  if  $NCORE = 0$  for all patches. Increases, without limit, as the number of disjunct core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[lsm\\_p\\_ncore](#), [mean](#),  
[lsm\\_c\\_dcore\\_sd](#), [lsm\\_c\\_dcore\\_cv](#),  
[lsm\\_l\\_dcore\\_mn](#), [lsm\\_l\\_dcore\\_sd](#), [lsm\\_l\\_dcore\\_cv](#)

## Examples

```
lsm_c_dcore_mn(landscape)
```

---

lsm_c_dcore_sd	<i>DCORE_SD (class level)</i>
----------------	-------------------------------

---

### Description

Standard deviation number of disjunct core areas (Core area metric)

### Usage

```
lsm_c_dcore_sd(landscape, directions, consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
lsm_c_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
lsm_c_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
lsm_c_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_c_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
lsm_c_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$DCORE_{SD} = sd(NCORE[patch_{ij}])$$

where  $NCORE[patch_{ij}]$  is the number of core areas.

DCORE\_SD is an 'Core area metric'. It summarises each class as the standard deviation of all patch areas belonging to class  $i$ . A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among patches of the same class  $i$  in the landscape.

**Units:** None

**Range:** DCORE\_SD  $\geq 0$

**Behaviour:** Equals DCORE\_SD = 0 if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_ncore](#), [sd](#),  
[lsm\\_c\\_dcore\\_mn](#), [lsm\\_c\\_dcore\\_cv](#),  
[lsm\\_l\\_dcore\\_mn](#), [lsm\\_l\\_dcore\\_sd](#), [lsm\\_l\\_dcore\\_cv](#)

**Examples**

```
lsm_c_dcore_sd(landscape)
```

---

lsm_c_division	<i>DIVISION (class level)</i>
----------------	-------------------------------

---

**Description**

Landscape division index (Aggregation metric)

**Usage**

```
lsm_c_division(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_division(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_division(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_division(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_division(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$DIVISION = (1 - \sum_{j=1}^n (\frac{a_{ij}}{A})^2)$$

where  $a_{ij}$  is the area in square meters and  $A$  is the total landscape area in square meters.

DIVISION is an 'Aggregation metric'. It can be in as the probability that two randomly selected cells are not located in the same patch of class  $i$ . The landscape division index is negatively correlated with the effective mesh size ([lsm\\_c\\_mesh](#)).

**Units:** Proportion

**Ranges:**  $0 \leq \text{Division} < 1$

**Behaviour:** Equals  $DIVISION = 0$  if only one patch is present. Approaches  $DIVISION = 1$  if all patches of class  $i$  are single cells.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. *Landscape ecology*, 15(2), 115-130.

**See Also**

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_l\\_division](#)

**Examples**

```
lsm_c_division(landscape)
```

---

lsm\_c\_ed

*ED (class level)*


---

**Description**

Edge Density (Area and Edge metric)

**Usage**

```
lsm_c_ed(landscape, count_boundary, directions)
```

```
## S3 method for class 'RasterLayer'
lsm_c_ed(landscape, count_boundary = FALSE,
         directions = 8)
```

```
## S3 method for class 'RasterStack'
lsm_c_ed(landscape, count_boundary = FALSE,
         directions = 8)
```

```
## S3 method for class 'RasterBrick'
lsm_c_ed(landscape, count_boundary = FALSE,
         directions = 8)
```

```
## S3 method for class 'stars'
lsm_c_ed(landscape, count_boundary = FALSE,
         directions = 8)
```

```
## S3 method for class 'list'
lsm_c_ed(landscape, count_boundary = FALSE,
         directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
count_boundary	Count landscape boundary as edge.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$ED = \frac{\sum_{k=1}^m e_{ik}}{A} * 10000$$

where  $e_{ik}$  is the total edge length in meters and  $A$  is the total landscape area in square meters.

ED is an 'Area and Edge metric'. The edge density equals the sum of all edges of class  $i$  in relation to the landscape area. The boundary of the landscape is only included in the corresponding total class edge length if `count_boundary = TRUE`. The metric describes the configuration of the landscape, e.g. because an aggregation of the same class will result in a low edge density. The metric is standardized to the total landscape area, and therefore comparisons among landscapes with different total areas are possible.

**Units:** Meters per hectare

**Range:**  $ED \geq 0$

**Behaviour:** Equals  $ED = 0$  if only one patch is present (and the landscape boundary is not included) and increases, without limit, as the landscapes becomes more patchy

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[lsm\\_c\\_te](#), [lsm\\_l\\_ta](#),  
[lsm\\_l\\_ed](#)

## Examples

```
lsm_c_ed(landscape)
```

---

lsm_c_enn_cv	<i>ENN_CV (class level)</i>
--------------	-----------------------------

---

### Description

Coefficient of variation of euclidean nearest-neighbor distance (Aggregation metric)

### Usage

```
lsm_c_enn_cv(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_c_enn_cv(landscape, directions = 8,
             verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_enn_cv(landscape, directions = 8,
             verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_c_enn_cv(landscape, directions = 8,
             verbose = TRUE)

## S3 method for class 'stars'
lsm_c_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_c_enn_cv(landscape, directions = 8, verbose = TRUE)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
verbose	Print warning message if not sufficient patches are present

### Details

$$ENN_{CV} = cv(ENN[patch_{ij}])$$

where  $ENN[patch_{ij}]$  is the euclidean nearest-neighbor distance of each patch.

ENN\_CV is an 'Aggregation metric'. It summarises each class as the Coefficient of variation of each patch belonging to class i. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN\_CV >= 0

**Behaviour:** Equals ENN\_CV = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

**See Also**

[lsm\\_p\\_enn, cv,](#)  
[lsm\\_c\\_enn\\_mn, lsm\\_c\\_enn\\_sd,](#)  
[lsm\\_l\\_enn\\_mn, lsm\\_l\\_enn\\_sd, lsm\\_l\\_enn\\_cv](#)

**Examples**

```
lsm_c_enn_cv(landscape)
```

---

lsm_c_enn_mn	<i>ENN_MN (class level)</i>
--------------	-----------------------------

---

**Description**

Mean of euclidean nearest-neighbor distance (Aggregation metric)

**Usage**

```
lsm_c_enn_mn(landscape, directions, verbose)
```

```
## S3 method for class 'RasterLayer'
```

```
lsm_c_enn_mn(landscape, directions = 8,  
  verbose = TRUE)
```

```
## S3 method for class 'RasterStack'
```

```
lsm_c_enn_mn(landscape, directions = 8,  
  verbose = TRUE)
```



```
## S3 method for class 'RasterBrick'
lsm_c_enn_mn(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'stars'
lsm_c_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_c_enn_mn(landscape, directions = 8, verbose = TRUE)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
verbose	Print warning message if not sufficient patches are present

### Details

$$ENN_{MN} = mean(ENN[patch_{ij}])$$

where  $ENN[patch_{ij}]$  is the euclidean nearest-neighbor distance of each patch.

ENN\_MN is an 'Aggregation metric'. It summarises each class as the mean of each patch belonging to class i. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit.

**Units:** Meters

**Range:** ENN\_MN > 0

**Behaviour:** Approaches ENN\_MN = 0 as the distance to the nearest neighbour decreases, i.e. patches of the same class i are more aggregated. Increases, without limit, as the distance between neighbouring patches of the same class i increases, i.e. patches are more isolated.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. *Ecological monographs*, 65(3), 235-260.

See Also

```
lsm_p_enn, mean,  
lsm_c_enn_sd, lsm_c_enn_cv,  
lsm_l_enn_mn, lsm_l_enn_sd, lsm_l_enn_cv
```

Examples

```
lsm_c_enn_mn(landscape)
```

---

lsm_c_enn_sd	<i>ENN_SD (class level)</i>
--------------	-----------------------------

---

Description

Standard deviation of euclidean nearest-neighbor distance (Aggregation metric)

Usage

```
lsm_c_enn_sd(landscape, directions, verbose)  
  
## S3 method for class 'RasterLayer'  
lsm_c_enn_sd(landscape, directions = 8,  
  verbose = TRUE)  
  
## S3 method for class 'RasterStack'  
lsm_c_enn_sd(landscape, directions = 8,  
  verbose = TRUE)  
  
## S3 method for class 'RasterBrick'  
lsm_c_enn_sd(landscape, directions = 8,  
  verbose = TRUE)  
  
## S3 method for class 'stars'  
lsm_c_enn_sd(landscape, directions = 8, verbose = TRUE)  
  
## S3 method for class 'list'  
lsm_c_enn_sd(landscape, directions = 8, verbose = TRUE)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).
verbose	Print warning message if not sufficient patches are present

## Details

$$ENN_{SD} = sd(ENN[patch_{ij}])$$

where  $ENN[patch_{ij}]$  is the euclidean nearest-neighbor distance of each patch.

ENN\_CV is an 'Aggregation metric'. It summarises each class as the standard deviation of each patch belonging to class i. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN\_SD >= 0

**Behaviour:** Equals ENN\_SD = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

## See Also

[lsm\\_p\\_enn, sd,](#)  
[lsm\\_c\\_enn\\_mn, lsm\\_c\\_enn\\_cv,](#)  
[lsm\\_l\\_enn\\_mn, lsm\\_l\\_enn\\_sd, lsm\\_l\\_enn\\_cv](#)

## Examples

```
lsm_c_enn_sd(landscape)
```

---

lsm_c_frac_cv	<i>FRAC_CV (class level)</i>
---------------	------------------------------

---

### Description

Coefficient of variation fractal dimension index (Shape metric)

### Usage

```
lsm_c_frac_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_frac_cv(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$FRAC_{CV} = cv(FRAC[patch_{ij}])$$

where  $FRAC[patch_{ij}]$  equals the fractal dimension index of each patch.

FRAC\_CV is a 'Shape metric'. The metric summarises each class as the Coefficient of variation of the fractal dimension index of all patches belonging to class i. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:** FRAC\_CV  $\geq 0$

**Behaviour:** Equals FRAC\_CV = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

**See Also**

[lsm\\_p\\_frac](#), [cv](#),  
[lsm\\_c\\_frac\\_mn](#), [lsm\\_c\\_frac\\_sd](#),  
[lsm\\_l\\_frac\\_mn](#), [lsm\\_l\\_frac\\_sd](#), [lsm\\_l\\_frac\\_cv](#)

**Examples**

```
lsm_c_frac_cv(landscape)
```

---

lsm_c_frac_mn	<i>FRAC_MN (class level)</i>
---------------	------------------------------

---

**Description**

Mean fractal dimension index (Shape metric)

**Usage**

```
lsm_c_frac_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_frac_mn(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$FRAC_{MN} = mean(FRAC[patch_{ij}])$$

where  $FRAC[patch_{ij}]$  equals the fractal dimension index of each patch.

FRAC\_MN is a 'Shape metric'. The metric summarises each class as the mean of the fractal dimension index of all patches belonging to class i. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:** FRAC\_MN > 0

**Behaviour:** Approaches FRAC\_MN = 1 if all patches are squared and FRAC\_MN = 2 if all patches are irregular.

**References**

- McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>
- Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

**See Also**

[lsm\\_p\\_frac](#), [mean](#),  
[lsm\\_c\\_frac\\_sd](#), [lsm\\_c\\_frac\\_cv](#),  
[lsm\\_l\\_frac\\_mn](#), [lsm\\_l\\_frac\\_sd](#), [lsm\\_l\\_frac\\_cv](#)

**Examples**

```
lsm_c_frac_mn(landscape)
```

---

lsm_c_frac_sd	<i>FRAC_SD (class level)</i>
---------------	------------------------------

---

### Description

Standard deviation fractal dimension index (Shape metric)

### Usage

```
lsm_c_frac_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_frac_sd(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$FRAC_{SD} = sd(FRAC[patch_{ij}])$$

where  $FRAC[patch_{ij}]$  equals the fractal dimension index of each patch.

FRAC\_SD is a 'Shape metric'. The metric summarises each class as the standard deviation of the fractal dimension index of all patches belonging to class i. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity.

**Units:** None

**Range:** FRAC\_SD >= 0

**Behaviour:** Equals FRAC\_SD = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

**See Also**

[lsm\\_p\\_frac](#), [sd](#),  
[lsm\\_c\\_frac\\_mn](#), [lsm\\_c\\_frac\\_cv](#),  
[lsm\\_l\\_frac\\_mn](#), [lsm\\_l\\_frac\\_sd](#), [lsm\\_l\\_frac\\_cv](#)

**Examples**

```
lsm_c_frac_sd(landscape)
```

---

lsm_c_gyrate_cv	<i>GYRATE_CV (class level)</i>
-----------------	--------------------------------

---

**Description**

Coefficient of variation radius of gyration (Area and edge metric)

**Usage**

```
lsm_c_gyrate_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_gyrate_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_gyrate_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_gyrate_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_gyrate_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_gyrate_cv(landscape, directions = 8)
```



**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$GYRATE_{CV} = cv(GYRATE[patch_{ij}])$$

where  $GYRATE[patch_{ij}]$  equals the radius of gyration of each patch.

GYRATE\_CV is an 'Area and edge metric'. The metric summarises each class as the Coefficient of variation of the radius of gyration of all patches belonging to class i. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** Meters

**Range:** GYRATE\_CV >= 0

**Behaviour:** Equals GYRATE\_CV = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

**See Also**

[lsm\\_p\\_gyrate\\_cv](#),  
[lsm\\_c\\_gyrate\\_mn](#), [lsm\\_c\\_gyrate\\_sd](#),  
[lsm\\_l\\_gyrate\\_mn](#), [lsm\\_l\\_gyrate\\_sd](#), [lsm\\_l\\_gyrate\\_cv](#)

**Examples**

```
lsm_c_gyrate_cv(landscape)
```

---

lsm_c_gyrate_mn	<i>GYRATE_MN (class level)</i>
-----------------	--------------------------------

---

### Description

Mean radius of gyration (Area and edge metric)

### Usage

```
lsm_c_gyrate_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_gyrate_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_gyrate_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_gyrate_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_gyrate_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_gyrate_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$GYRATE_{MN} = mean(GYRATE[patch_{ij}])$$

where  $GYRATE[patch_{ij}]$  equals the radius of gyration of each patch.

GYRATE\_MN is an 'Area and edge metric'. The metric summarises each class as the mean of the radius of gyration of all patches belonging to class i. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** Meters

**Range:** GYRATE\_MN  $\geq 0$

**Behaviour:** Approaches GYRATE\_MN = 0 if every patch is a single cell. Increases, without limit, when only one patch is present.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. *Conservation ecology*, 1(1).

**See Also**

[lsm\\_p\\_gyrate](#), [mean](#),  
[lsm\\_c\\_gyrate\\_sd](#), [lsm\\_c\\_gyrate\\_cv](#),  
[lsm\\_l\\_gyrate\\_mn](#), [lsm\\_l\\_gyrate\\_sd](#), [lsm\\_l\\_gyrate\\_cv](#)

**Examples**

```
lsm_c_gyrate_mn(landscape)
```

---

lsm_c_gyrate_sd	<i>GYRATE_SD (class level)</i>
-----------------	--------------------------------

---

**Description**

Standard deviation radius of gyration (Area and edge metric)

**Usage**

```
lsm_c_gyrate_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_gyrate_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_gyrate_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_gyrate_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_gyrate_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_gyrate_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$GYRATE_{SD} = sd(GYRATE[patch_{ij}])$$

where  $GYRATE[patch_{ij}]$  equals the radius of gyration of each patch.

GYRATE\_SD is an 'Area and edge metric'. The metric summarises each class as the standard deviation of the radius of gyration of all patches belonging to class i. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness.

**Units:** Meters

**Range:** GYRATE\_SD >= 0

**Behaviour:** Equals GYRATE\_SD = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

**See Also**

[lsm\\_p\\_gyrate\\_cv](#),  
[lsm\\_c\\_gyrate\\_mn](#), [lsm\\_c\\_gyrate\\_cv](#),  
[lsm\\_l\\_gyrate\\_mn](#), [lsm\\_l\\_gyrate\\_sd](#), [lsm\\_l\\_gyrate\\_cv](#)

**Examples**

```
lsm_c_gyrate_sd(landscape)
```

lsm\_c\_iji

*Interspersion and Juxtaposition index (class level)***Description**

Interspersion and Juxtaposition index (Aggregation metric)

**Usage**

```
lsm_c_iji(landscape, verbose)

## S3 method for class 'RasterLayer'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'stars'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'list'
lsm_c_iji(landscape, verbose = TRUE)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
verbose	Print warning message if not sufficient patches are present

**Details**

$$IJI = \frac{- \sum_{k=1}^m \left[ \left( \frac{e_{ik}}{m} \right) \ln \left( \frac{e_{ik}}{m} \right) \right]}{\ln(m-1)} * 100$$

where  $e_{ik}$  are the unique adjacencies of all classes (lower/upper triangle of the adjacency table - without the diagonal) and  $m$  is the number of classes.

IJI is an 'Aggregation metric'. It is a so called "salt and pepper" metric and describes the intermixing of classes (i.e. without considering like adjacencies - the diagonal of the adjacency table). The number of classes to calculate IJI must be  $\geq$  than 3.

**Units:** Percent

**Range:**  $0 < IJI \leq 100$

**Behaviour:** Approaches 0 if a class is only adjacent to a single other class and equals 100 when a class is equally adjacent to all other classes.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., & Marks, B. J. 1995. FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. Gen. Tech. Rep. PNW-GTR-351. Portland, OR: US Department of Agriculture, Forest Service, Pacific Northwest Research Station. 122 p, 351.

**See Also**

[lsm\\_l\\_iji](#)

**Examples**

```
lsm_c_iji(landscape)
```

---

lsm_c_lpi	<i>LPI (class level)</i>
-----------	--------------------------

---

**Description**

Largest patch index (Area and Edge metric)

**Usage**

```
lsm_c_lpi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_lpi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_lpi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_lpi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_lpi(landscape, directions = 8)
```

```
## S3 method for class 'list'
lsm_c_lpi(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$LPI = \frac{\max_{j=1}^n(a_{ij})}{A} * 100$$

where  $\max(a_{ij})$  is the area of the patch in square meters and  $A$  is the total landscape area in square meters.

The largest patch index is an 'Area and edge metric'. It is the percentage of the landscape covered by the corresponding largest patch of each class  $i$ . It is a simple measure of dominance.

**Units:** Percentage

**Range:**  $0 < LPI \leq 100$

**Behaviour:** Approaches  $LPI = 0$  when the largest patch is becoming small and equals  $LPI = 100$  when only one patch is present

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_l\\_lpi](#)

### Examples

```
lsm_c_lpi(landscape)
```

lsm\_c\_lsi

*LSI (class level)***Description**

Landscape shape index (Aggregation metric)

**Usage**

```
lsm_c_lsi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_lsi(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$LSI = \frac{e_i}{\min e_i}$$

where  $e_i$  is the total edge length in cell surfaces and  $\min e_i$  is the minimum total edge length in cell surfaces

LSI is an 'Aggregation metric'. It is the ratio between the actual edge length of class i and the hypothetical minimum edge length of class i. The minimum edge length equals the edge length if class i would be maximally aggregated.

**Units:** None

**Ranges:** LSI >= 1



**Behaviour:** Equals  $LSI = 1$  when only one squared patch is present or all patches are maximally aggregated. Increases, without limit, as the length of the actual edges increases, i.e. the patches become less compact.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

### See Also

[lsm\\_p\\_shape](#),  
[lsm\\_l\\_lsi](#)

### Examples

```
lsm_c_lsi(landscape)
```

---

lsm_c_mesh	<i>MESH (class level)</i>
------------	---------------------------

---

### Description

Effective Mesh Size (Aggregation metric)

### Usage

```
lsm_c_mesh(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_mesh(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_mesh(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_mesh(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_mesh(landscape, directions = 8)
```

```
## S3 method for class 'list'
lsm_c_mesh(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$MESH = \frac{\sum_{j=1}^n a_{ij}^2}{A} * \frac{1}{10000}$$

where  $a_{ij}$  is the patch area in square meters and  $A$  is the total landscape area in square meters.

The effective mesh size is an 'Aggregation metric'. Because each patch is squared before the sums for each group  $i$  are calculated and the sum is standardized by the total landscape area, MESH is a relative measure of patch structure. MESH is perfectly, negatively correlated to [lsm\\_c\\_division](#).

**Units:** Hectares

**Range:** cell size / total area <= MESH <= total area

**Behaviour:** Equals cellsize/total area if class covers only one cell and equals total area if only one patch is present.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. *Landscape ecology*, 15(2), 115-130.

### See Also

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_l\\_mesh](#)

### Examples

```
lsm_c_mesh(landscape)
```

---

lsm_c_ndca	<i>NDCA (class level)</i>
------------	---------------------------

---

### Description

Number of disjunct core areas (Core area metric)

### Usage

```
lsm_c_ndca(landscape, directions, consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
lsm_c_ndca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
lsm_c_ndca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
lsm_c_ndca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_c_ndca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
lsm_c_ndca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$NDCA = \sum_{j=1}^n n_{ij}^{core}$$

where  $n_{ij}^{core}$  is the number of disjunct core areas.

NDCA is a 'Core area metric'. The metric summarises class i as the sum of all patches belonging to class i. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NDCA counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. It describes patch area and shape simultaneously (more core area when the patch is large, however, the shape must allow disjunct core areas). Thereby, a compact shape (e.g. a square) will contain less disjunct core areas than a more irregular patch.

**Units:** None

**Range:** NDCA >= 0

**Behaviour:** NDCA = 0 when TCA = 0, i.e. every cell in patches of class i is an edge. NDCA increases, with out limit, as core area increases and patch shapes allow disjunct core areas (i.e. patch shapes become rather complex).

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_tca](#),  
[lsm\\_p\\_ncore](#), [lsm\\_l\\_ndca](#)

**Examples**

```
lsm_c_ndca(landscape)
```

---

lsm_c_nlsi	<i>nLSI (class level)</i>
------------	---------------------------

---

**Description**

Normalized landscape shape index (Aggregation metric)

**Usage**

```
lsm_c_nlsi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_nlsi(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$nLSI = \frac{e_i}{\min e_i}$$

where  $e_i$  is the total edge length in cell surfaces and  $\min e_i$  is the minimum total edge length in cell surfaces

nLSI is an 'Aggregation metric'. It is the ratio between the actual edge length of class i and the hypothetical minimum edge length of class i. The minimum edge length equals the edge length if class i would be maximally aggregated.

**Units:** None

**Ranges:** nlsi >= 1

**Behaviour:** Equals nlsi = 1 when only one squared patch is present or all patches are maximally aggregated. Increases, without limit, as the length of the actual edges increases, i.e. the patches become less compact.

**Value**

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

See Also

[lsm\\_p\\_shape](#)

Examples

```
lsm_c_nlsi(landscape)
```

---

lsm_c_np	<i>NP (class level)</i>
----------	-------------------------

---

Description

Number of patches (Aggregation metric)

Usage

```
lsm_c_np(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_np(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_np(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_np(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_np(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_np(landscape, directions = 8)
```

Arguments

- landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.
- directions      The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

Details

$$NP = n_i$$

where  $n_i$  is the number of patches.  
NP is an 'Aggregation metric'. It describes the fragmentation of a class, however, does not necessarily contain information about the configuration or composition of the class.

**Units:** None

**Ranges:** NP >= 1

**Behaviour:** Equals NP = 1 when only one patch is present and increases, without limit, as the number of patches increases

Value

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_l\\_np](#)

Examples

`lsm_c_np(landscape)`

---

lsm_c_pafrac	<i>PAFRAC (class level)</i>
--------------	-----------------------------

---

Description

Perimeter-Area Fractal Dimension (Shape metric)

**Usage**

```
lsm_c_pafrac(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_c_pafrac(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_pafrac(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_c_pafrac(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'stars'
lsm_c_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_c_pafrac(landscape, directions = 8, verbose = TRUE)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
verbose	Print warning message if not sufficient patches are present

**Details**

$$PAFRAC = \frac{2}{\beta}$$

where  $\beta$  is the slope of the regression of the area against the perimeter (logarithm)  $n_i \sum_{j=1}^n \ln a_{ij} =$

$$a + \beta n_i \sum_{j=1}^n \ln p_{ij}$$

PAFRAC is a 'Shape metric'. It describes the patch complexity of class i while being scale independent. This means that increasing the patch size while not changing the patch form will not change the metric. However, it is only meaningful if the relationship between the area and perimeter is linear on a logarithmic scale. Furthermore, if there are less than 10 patches in class i, the metric returns NA because of the small-sample issue.

**Units:** None

**Range:**  $1 \leq PAFRAC \leq 2$

**Behaviour:** Approaches PAFRAC = 1 for patches with simple shapes and approaches PAFRAC = 2 for irregular shapes



**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Burrough, P. A. 1986. Principles of Geographical Information Systems for Land Resources Assessment. Monographs on Soil and Resources Survey No. 12. Clarendon Press, Oxford

**See Also**

[lsm\\_p\\_area](#), [lsm\\_p\\_perim](#),  
[lsm\\_l\\_pafrac](#)

**Examples**

```
lsm_c_pafrac(landscape)
```

---

lsm_c_para_cv	<i>PARAM_CV (class level)</i>
---------------	-------------------------------

---

**Description**

Coefficient of variation perimeter-area ratio (Shape metric)

**Usage**

```
lsm_c_para_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_para_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_para_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_para_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_para_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_para_cv(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$PARA_{CV} = cv(PARA[patch_{ij}])$$

where  $PARA[patch_{ij}]$  is the perimeter area ratio of each patch.

PARA\_CV is a 'Shape metric'. It summarises each class as the Coefficient of variation of each patch belonging to class i. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standardised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA\_CV  $\geq 0$

**Behaviour:** Equals PARA\_CV = 0 if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_para](#), [cv](#),  
[lsm\\_c\\_para\\_mn](#), [lsm\\_c\\_para\\_sd](#),  
[lsm\\_l\\_para\\_mn](#), [lsm\\_l\\_para\\_sd](#), [lsm\\_l\\_para\\_cv](#)

**Examples**

```
lsm_c_para_cv(landscape)
```

---

lsm_c_para_mn	<i>PARAMN (class level)</i>
---------------	-----------------------------

---

### Description

Mean perimeter-area ratio (Shape metric)

### Usage

```
lsm_c_para_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_para_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$PARAMN = \text{mean}(PARA[patch_{ij}])$$

where  $PARA[patch_{ij}]$  is the perimeter area ratio of each patch.

PARAMN is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standardised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:**  $PARAMN > 0$

**Behaviour:** Approaches  $PARAMN > 0$  if PARA for each patch approaches  $PARA > 0$ , i.e. the form approaches a rather small square. Increases, without limit, as PARA increases, i.e. patches become more complex.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_para](#), [mean](#),  
[lsm\\_c\\_para\\_sd](#), [lsm\\_c\\_para\\_cv](#),  
[lsm\\_l\\_para\\_mn](#), [lsm\\_l\\_para\\_sd](#), [lsm\\_l\\_para\\_cv](#)

**Examples**

```
lsm_c_para_mn(landscape)
```

---

lsm_c_para_sd	<i>PARAMETER SD (class level)</i>
---------------	-----------------------------------

---

**Description**

Standard deviation perimeter-area ratio (Shape metric)

**Usage**

```
lsm_c_para_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_para_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$PARA_{SD} = sd(PARA[patch_{ij}])$$

where  $PARA[patch_{ij}]$  is the perimeter area ratio of each patch.

PARA\_SD is a 'Shape metric'. It summarises each class as the standard deviation of each patch belonging to class i. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standardised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA\_SD >= 0

**Behaviour:** Equals PARA\_SD = 0 if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_para](#), [sd](#),  
[lsm\\_c\\_para\\_mn](#), [lsm\\_c\\_para\\_cv](#),  
[lsm\\_l\\_para\\_mn](#), [lsm\\_l\\_para\\_sd](#), [lsm\\_l\\_para\\_cv](#)

**Examples**

```
lsm_c_para_sd(landscape)
```

---

lsm_c_pd	<i>PD (class level)</i>
----------	-------------------------

---

### Description

Patch density (Aggregation metric)

### Usage

```
lsm_c_pd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_pd(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$PD = \frac{n_i}{A} * 10000 * 100$$

where  $n_i$  is the number of patches and  $A$  is the total landscape area in square meters.

PD is an 'Aggregation metric'. It describes the fragmentation of a class, however, does not necessarily contain information about the configuration or composition of the class. In contrast to [lsm\\_c\\_np](#) it is standardized to the area and comparisons among landscapes with different total area are possible.

**Units:** Number per 100 hectares

**Ranges:**  $0 < PD \leq 1e+06$

**Behaviour:** Increases as the landscape gets more patchy. Reaches its maximum if every cell is a different patch.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_np](#), [lsm\\_l\\_ta](#),  
[lsm\\_l\\_pd](#)

**Examples**

```
lsm_c_pd(landscape)
```

---

lsm\_c\_pladj

---

*PLADJ (class level)*


---

**Description**

Percentage of Like Adjacencies (Aggregation metric)

**Usage**

```
lsm_c_pladj(landscape)

## S3 method for class 'RasterLayer'
lsm_c_pladj(landscape)

## S3 method for class 'RasterStack'
lsm_c_pladj(landscape)

## S3 method for class 'RasterBrick'
lsm_c_pladj(landscape)

## S3 method for class 'stars'
lsm_c_pladj(landscape)

## S3 method for class 'list'
lsm_c_pladj(landscape)
```

**Arguments**

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.

Details

$$PLADJ = (\frac{g_{ii}}{\sum_{k=1}^m g_{ik}}) * 100$$

where  $g_{ii}$  is the number of adjacencies between cells of class  $i$  and  $g_{ik}$  is the number of adjacencies between cells of class  $i$  and  $k$ .

PLADJ is an 'Aggregation metric'. It calculates the frequency how often patches of different classes  $i$  (focal class) and  $k$  are next to each other, and following is a measure of class aggregation. The adjacencies are counted using the double-count method.

**Units:** Percent

**Ranges:**  $0 \leq PLADJ \leq 100$

**Behaviour:** Equals  $PLADJ = 0$  if class  $i$  is maximal disaggregated, i.e. every cell is a different patch. Equals  $PLADJ = 100$  when the only one patch is present.

Value

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Examples

```
lsm_c_pladj(landscape)
```

---

lsm_c_pland	<i>PLAND (class level)</i>
-------------	----------------------------

---

Description

Percentage of landscape of class (Area and Edge metric)



**Usage**

```
lsm_c_pland(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_pland(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$PLAND = \frac{\sum_{j=1}^n a_{ij}}{A} * 100$$

where  $a_{ij}$  is the area of each patch and  $A$  is the total landscape area.

PLAND is an 'Area and edge metric'. It is the percentage of the landscape belonging to class  $i$ . It is a measure of composition and because of the relative character directly comparable among landscapes with different total areas.

**Units:** Percentage

**Range:**  $0 < PLAND \leq 100$

**Behaviour:** Approaches  $PLAND = 0$  when the proportional class area is decreasing. Equals  $PLAND = 100$  when only one patch is present.

**Value**

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_c\\_ca](#), [lsm\\_l\\_ta](#)

Examples

```
lsm_c_pland(landscape)
```

---

lsm_c_shape_cv	<i>SHAPE_CV (class level)</i>
----------------	-------------------------------

---

Description

Covariance of variation shape index (Shape metric)

Usage

```
lsm_c_shape_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_shape_cv(landscape, directions = 8)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

**Details**

$$SHAPE_{CV} = cv(SHAPE[patch_{ij}])$$

where  $SHAPE[patch_{ij}]$  is the shape index of each patch.

SHAPE\_CV is a 'Shape metric'. Each class is summarised as the Coefficient of variation of each patch belonging to class i. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE\_CV >= 0

**Behaviour:** Equals SHAPE\_CV = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

**See Also**

[lsm\\_p\\_shape\\_cv](#),  
[lsm\\_c\\_shape\\_mn](#), [lsm\\_c\\_shape\\_sd](#),  
[lsm\\_l\\_shape\\_mn](#), [lsm\\_l\\_shape\\_sd](#), [lsm\\_l\\_shape\\_cv](#)

**Examples**

```
lsm_c_shape_cv(landscape)
```

---

lsm_c_shape_mn	<i>SHAPE_MN (class level)</i>
----------------	-------------------------------

---

### Description

Mean shape index (Shape metric)

### Usage

```
lsm_c_shape_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_shape_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$SHAPE_{MN} = mean(SHAPE[patch_{ij}])$$

where  $SHAPE[patch_{ij}]$  is the shape index of each patch.

SHAPE\_MN is a 'Shape metric'. Each class is summarised as the mean of each patch belonging to class i. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE\_SD >= 1

**Behaviour:** Equals SHAPE\_MN = 0 if all patches are squares. Increases, without limit, as the shapes of patches become more complex.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

**See Also**

[lsm\\_p\\_shape](#), [mean](#),  
[lsm\\_c\\_shape\\_sd](#), [lsm\\_c\\_shape\\_cv](#),  
[lsm\\_l\\_shape\\_mn](#), [lsm\\_l\\_shape\\_sd](#), [lsm\\_l\\_shape\\_cv](#)

**Examples**

```
lsm_c_shape_mn(landscape)
```

---

lsm_c_shape_sd	<i>SHAPE_SD (class level)</i>
----------------	-------------------------------

---

**Description**

Standard deviation shape index (Shape metric)

**Usage**

```
lsm_c_shape_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_shape_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$SHAPE_{SD} = sd(SHAPE[patch_{ij}])$$

where  $SHAPE[patch_{ij}]$  is the shape index of each patch.

SHAPE\_SD is a 'Shape metric'. Each class is summarised as the standard deviation of each patch belonging to class i. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE\_SD >= 0

**Behaviour:** Equals SHAPE\_SD = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

**See Also**

[lsm\\_p\\_shape](#), [sd](#),  
[lsm\\_c\\_shape\\_mn](#), [lsm\\_c\\_shape\\_cv](#),  
[lsm\\_l\\_shape\\_mn](#), [lsm\\_l\\_shape\\_sd](#), [lsm\\_l\\_shape\\_cv](#)

**Examples**

```
lsm_c_shape_sd(landscape)
```

---

lsm_c_split	<i>SPLIT (class level)</i>
-------------	----------------------------

---

### Description

Splitting index (Aggregation metric)

### Usage

```
lsm_c_split(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_split(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$SPLIT = \frac{A^2}{\sum_{j=1}^n a_{ij}^2}$$

where  $a_{ij}$  is the patch area in square meters and  $A$  is the total landscape area.

SPLIT is an 'Aggregation metric'. It describes the number of patches if all patches of class  $i$  would be divided into equally sized patches.

**Units:** None

**Range:**  $1 \leq SPLIT \leq \text{Number of cells squared}$

**Behaviour:** Equals  $SPLIT = 1$  if only one patch is present. Increases as the number of patches of class  $i$  increases and is limited if all cells are a patch

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. *Landscape ecology*, 15(2), 115-130.

**See Also**

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_l\\_split](#)

**Examples**

```
lsm_c_split(landscape)
```

---

lsm_c_tca	<i>TCA (class level)</i>
-----------	--------------------------

---

**Description**

Total core area (Core area metric)

**Usage**

```
lsm_c_tca(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```



```
## S3 method for class 'list'
lsm_c_tca(landscape, directions = 8,
          consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$TCA = \sum_{j=1}^n a_{ij}^{core} * (\frac{1}{10000})$$

where here  $a_{ij}^{core}$  is the core area in square meters.

TCA is a 'Core area metric' and equals the sum of core areas of all patches belonging to class i. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). In other words, the core area of a patch is all area that is not an edge. It characterises patch areas and shapes of patches belonging to class i simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square). Additionally, TCA is a measure for the configuration of the landscape, because the sum of edges increase as patches are less aggregated.

**Units:** Hectares

**Range:** TCA >= 0

**Behaviour:** Increases, without limit, as patch areas increase and patch shapes simplify. TCA = 0 when every cell in every patch of class i is an edge.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_core](#), [lsm\\_l\\_tca](#)

Examples

```
lsm_c_tca(landscape)
```

---

lsm_c_te	<i>TE (class level)</i>
----------	-------------------------

---

Description

Total (class) edge (Area and Edge metric)

Usage

```
lsm_c_te(landscape, count_boundary, directions)

## S3 method for class 'RasterLayer'
lsm_c_te(landscape, count_boundary = FALSE,
         directions = 8)

## S3 method for class 'RasterStack'
lsm_c_te(landscape, count_boundary = FALSE,
         directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_te(landscape, count_boundary = FALSE,
         directions = 8)

## S3 method for class 'stars'
lsm_c_te(landscape, count_boundary = FALSE,
         directions = 8)

## S3 method for class 'list'
lsm_c_te(landscape, count_boundary = FALSE,
         directions = 8)
```

Arguments

- landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.
- count\_boundary    Include landscape boundary in edge length
- directions      The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

Details

$$TE = \sum_{k=1}^m e_{ik}$$

where  $e_{ik}$  is the edge lengths in meters. TE is an 'Area and edge metric'. Total (class) edge includes all edges between class i and all other classes k. It measures the configuration of the landscape because a highly fragmented landscape will have many edges. However, total edge is an absolute measure, making comparisons among landscapes with different total areas difficult. If `cound_boundary = TRUE` also edges to the landscape boundary are included.

**Units:** Meters

**Range:** TE >= 0

**Behaviour:** Equals TE = 0 if all cells are edge cells. Increases, without limit, as landscape becomes more fragmented

Value

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_p\\_perim](#) [lsm\\_l\\_te](#)

Examples

```
lsm_c_te(landscape)
```

---

lsm_l_ai	AI (landscape level)
----------	----------------------

---

Description

Aggregation index (Aggregation metric)

**Usage**

```
lsm_l_ai(landscape)

## S3 method for class 'RasterLayer'
lsm_l_ai(landscape)

## S3 method for class 'RasterStack'
lsm_l_ai(landscape)

## S3 method for class 'RasterBrick'
lsm_l_ai(landscape)

## S3 method for class 'stars'
lsm_l_ai(landscape)

## S3 method for class 'list'
lsm_l_ai(landscape)
```

**Arguments**

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers

**Details**

$$AI = \left[ \sum_{i=1}^m \left( \frac{g_{ii}}{max - g_{ii}} \right) P_i \right] (100)$$

where  $g_{ii}$  is the number of like adjacencies based on the single-count method and  $max - g_{ii}$  is the classwise maximum number of like adjacencies of class  $i$  and  $P_i$  the proportion of landscape comprised of class  $i$ .

AI is an 'Aggregation metric'. It equals the number of like adjacencies divided by the theoretical maximum possible number of like adjacencies for that class summed over each class for the entire landscape. The metric is based on the adjacency matrix and the single-count method.

**Units:** Percent

**Range:**  $0 \leq AI \leq 100$

**Behaviour:** Equals 0 for maximally disaggregated and 100 for maximally aggregated classes.

**Value**

tibble

## References

- McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>
- He, H. S., DeZonia, B. E., & Mladenoff, D. J. 2000. An aggregation index (AI) to quantify spatial patterns of landscapes. *Landscape ecology*, 15(7), 591-601.

## See Also

[lsm\\_c\\_ai](#)

## Examples

```
lsm_l_ai(landscape)
```

---

lsm_l_area_cv	<i>AREA_CV (landscape level)</i>
---------------	----------------------------------

---

## Description

Coefficient of variation of patch area (Area and edge metric)

## Usage

```
lsm_l_area_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_area_cv(landscape, directions = 8)
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

Details

$$AREA_{CV} = cv(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares.

AREA\_CV is an 'Area and Edge metric'. The metric summarises the landscape as the Coefficient of variation of all patches in the landscape. The metric describes the differences among patches in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** AREA\_CV >= 0

**Behaviour:** Equals AREA\_CV = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

Value

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_p\\_area](#), [cv](#),  
[lsm\\_c\\_area\\_mn](#), [lsm\\_c\\_area\\_sd](#), [lsm\\_c\\_area\\_cv](#),  
[lsm\\_l\\_area\\_mn](#), [lsm\\_l\\_area\\_sd](#)

Examples

```
lsm_l_area_cv(landscape)
```

---

lsm_l_area_mn	<i>AREA_MN (landscape level)</i>
---------------	----------------------------------

---

Description

Mean of patch area (Area and edge metric)

**Usage**

```
lsm_l_area_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_area_mn(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$AREA_{MN} = mean(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares

AREA\_MN is an 'Area and Edge metric'. The metric summarises the landscape as the mean of all patch in the landscape. The metric is a simple way to describe the composition of the landscape. Especially together with the total landscape area ([lsm\\_l\\_ta](#)), it can also give an idea of patch structure (e.g. many small patches vs. few large patches).

**Units:** Hectares

**Range:**  $AREA_{MN} > 0$

**Behaviour:** Approaches  $AREA_{MN} = 0$  if all patches are small. Increases, without limit, as the patch areas increase.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

```
lsm_p_area, mean,  
lsm_c_area_mn, lsm_c_area_sd, lsm_c_area_cv  
lsm_l_area_sd, lsm_l_area_cv
```

Examples

```
lsm_l_area_mn(landscape)
```

---

lsm_l_area_sd	<i>AREA_SD (landscape level)</i>
---------------	----------------------------------

---

Description

Standard deviation of patch area (Area and edge metric)

Usage

```
lsm_l_area_sd(landscape, directions)  
  
## S3 method for class 'RasterLayer'  
lsm_l_area_sd(landscape, directions = 8)  
  
## S3 method for class 'RasterStack'  
lsm_l_area_sd(landscape, directions = 8)  
  
## S3 method for class 'RasterBrick'  
lsm_l_area_sd(landscape, directions = 8)  
  
## S3 method for class 'stars'  
lsm_l_area_sd(landscape, directions = 8)  
  
## S3 method for class 'list'  
lsm_l_area_sd(landscape, directions = 8)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).



Details

$$AREA_{SD} = sd(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares.

AREA\_SD is an 'Area and Edge metric'. The metric summarises the landscape as the standard deviation of all patch in the landscape. The metric describes the differences among all patches in the landscape.

**Units:** Hectares

**Range:** AREA\_SD >= 0

**Behaviour:** Equals AREA\_SD = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

Value

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_p\\_area](#), [sd](#),  
[lsm\\_c\\_area\\_mn](#), [lsm\\_c\\_area\\_sd](#), [lsm\\_c\\_area\\_cv](#)  
[lsm\\_l\\_area\\_mn](#), [lsm\\_l\\_area\\_cv](#)

Examples

```
lsm_l_area_sd(landscape)
```

---

lsm_l_cai_cv	<i>CAI_CV (landscape level)</i>
--------------	---------------------------------

---

Description

Coefficient of variation of core area index (Core area metric)

**Usage**

```

lsm_l_cai_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_cai_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$CAI_{CV} = cv(CAI[patch_{ij}])$$

where  $CAI[patch_{ij}]$  is the core area index of each patch.

CAI\_CV is a 'Core area metric'. The metric summarises the landscape as the Coefficient of variation of the core area index of all patches in the landscape. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among all patches in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** Percent

**Range:** CAI\_CV  $\geq 0$

**Behaviour:** Equals CAI\_CV = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of the core area indices increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_cai](#), [cv](#),  
[lsm\\_c\\_cai\\_mn](#), [lsm\\_c\\_cai\\_sd](#), [lsm\\_c\\_cai\\_cv](#),  
[lsm\\_l\\_cai\\_mn](#), [lsm\\_l\\_cai\\_sd](#)

### Examples

```
lsm_l_cai_cv(landscape)
```

---

lsm_l_cai_mn	<i>CAI_MN (landscape level)</i>
--------------	---------------------------------

---

### Description

Mean of core area index (Core area metric)

### Usage

```
lsm_l_cai_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_cai_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_l_cai_mn(landscape, directions = 8,
             consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_cai_mn(landscape, directions = 8,
             consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$CAI_{MN} = mean(CAI[patch_{ij}])$$

where  $CAI[patch_{ij}]$  is the core area index of each patch.

CAI\_MN is a 'Core area metric'. The metric summarises the landscape as the mean of the core area index of all patches in the landscape. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case).

**Units:** Percent

**Range:** CAI\_MN  $\geq 0$

**Behaviour:** Equals CAI\_MN = 0 if CAI = 0 for all patches. Increases, without limit, as the core area indices increase.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

```
lsm_p_cai, mean,  
lsm_c_cai_sd, lsm_c_cai_sd, lsm_c_cai_cv,  
lsm_l_cai_sd, lsm_l_cai_cv
```

Examples

```
lsm_l_cai_mn(landscape)
```

---

lsm_l_cai_sd	<i>CAI_SD (landscape level)</i>
--------------	---------------------------------

---

Description

Standard deviation of core area index (Core area metric)

Usage

```
lsm_l_cai_sd(landscape, directions, consider_boundary, edge_depth)  
  
## S3 method for class 'RasterLayer'  
lsm_l_cai_sd(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)  
  
## S3 method for class 'RasterStack'  
lsm_l_cai_sd(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)  
  
## S3 method for class 'RasterBrick'  
lsm_l_cai_sd(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)  
  
## S3 method for class 'stars'  
lsm_l_cai_sd(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)  
  
## S3 method for class 'list'  
lsm_l_cai_sd(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)
```

Arguments

- landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.
- directions    The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

## Details

$$CAI_{SD} = sd(CAI[patch_{ij}])$$

where  $CAI[patch_{ij}]$  is the core area index of each patch.

CAI\_SD is a 'Core area metric'. The metric summarises the landscape as the standard deviation of the core area index of all patches in the landscape. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among all patches in the landscape.

**Units:** Percent

**Range:** CAI\_SD >= 0

**Behaviour:** Equals CAI\_SD = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of core area indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[lsm\\_p\\_cai](#), [sd](#),  
[lsm\\_c\\_cai\\_mn](#), [lsm\\_c\\_cai\\_sd](#), [lsm\\_c\\_cai\\_cv](#),  
[lsm\\_l\\_cai\\_mn](#), [lsm\\_l\\_cai\\_cv](#)

## Examples

```
lsm_l_cai_sd(landscape)
```

---

lsm_l_circle_cv	<i>CIRCLE_CV (landscape level)</i>
-----------------	------------------------------------

---

### Description

Coefficient of variation of related circumscribing circle (Shape metric)

### Usage

```
lsm_l_circle_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_circle_cv(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CIRCLE_{CV} = cv(CIRCLE[patch_{ij}])$$

where  $CIRCLE[patch_{ij}]$  is the related circumscribing circle of each patch.

$CIRCLE_{CV}$  is a 'Shape metric' and summarises the landscape as the Coefficient of variation of the related circumscribing circle of all patches in the landscape.  $CIRCLE$  describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch.  $CIRCLE_{CV}$  describes the differences among all patches in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** None

**Range:**  $CIRCLE_{CV} \geq 0$

**Behaviour:** Equals  $CIRCLE_{CV}$  if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landscape Ecology* 7: 291-302.

**See Also**

[lsm\\_p\\_circle](#), [mean](#),  
[lsm\\_c\\_circle\\_mn](#), [lsm\\_c\\_circle\\_sd](#), [lsm\\_c\\_circle\\_cv](#),  
[lsm\\_l\\_circle\\_mn](#), [lsm\\_l\\_circle\\_sd](#)

**Examples**

```
lsm_l_circle_cv(landscape)
```

---

lsm_l_circle_mn	<i>CIRCLE_MN (landscape level)</i>
-----------------	------------------------------------

---

**Description**

Mean of related circumscribing circle (Shape metric)

**Usage**

```
lsm_l_circle_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_circle_mn(landscape, directions = 8)
```



**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$CIRCLE_{MN} = mean(CIRCLE[patch_{ij}])$$

where  $CIRCLE[patch_{ij}]$  is the related circumscribing circle of each patch.

CIRCLE\_MN is a 'Shape metric' and summarises the landscape as the mean of the related circumscribing circle of all patches in the landscape. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch.

**Units:** None

**Range:** CIRCLE\_MN > 0

**Behaviour:** Approaches CIRCLE\_MN = 0 if the related circumscribing circle of all patches is small. Increases, without limit, as the related circumscribing circles increase.

**Value**

tibble

**References**

- McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>
- Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landscape Ecology* 7: 291-302.

**See Also**

`lsm_p_circle`, `mean`,  
`lsm_c_circle_mn`, `lsm_c_circle_sd`, `lsm_c_circle_cv`,  
`lsm_l_circle_sd`, `lsm_l_circle_cv`

**Examples**

```
lsm_l_circle_mn(landscape)
```

---

lsm_l_circle_sd	<i>CIRCLE_SD (landscape level)</i>
-----------------	------------------------------------

---

### Description

Standard deviation of related circumscribing circle (Shape metric)

### Usage

```
lsm_l_circle_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_circle_sd(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CIRCLE_{SD} = sd(CIRCLE[patch_{ij}])$$

where  $CIRCLE[patch_{ij}]$  is the related circumscribing circle of each patch.

$CIRCLE_{SD}$  is a 'Shape metric' and summarises the landscape as the standard deviation of the related circumscribing circle of all patches in the landscape.  $CIRCLE$  describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch. The metric describes the differences among all patches of the landscape.

**Units:** None

**Range:**  $CIRCLE_{SD} \geq 0$

**Behaviour:** Equals  $CIRCLE_{SD}$  if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landscape Ecology* 7: 291-302.

**See Also**

[lsm\\_p\\_circle](#), [mean](#),  
[lsm\\_c\\_circle\\_mn](#), [lsm\\_c\\_circle\\_sd](#), [lsm\\_c\\_circle\\_cv](#),  
[lsm\\_l\\_circle\\_mn](#), [lsm\\_l\\_circle\\_cv](#)

**Examples**

```
lsm_l_circle_sd(landscape)
```

---

lsm_l_cohesion	<i>COHESION (landscape level)</i>
----------------	-----------------------------------

---

**Description**

Patch Cohesion Index (Aggregation metric)

**Usage**

```
lsm_l_cohesion(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_cohesion(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$COHESION = 1 - \left( \frac{\sum_{i=1}^m \sum_{j=1}^n p_{ij}}{\sum_{i=1}^m \sum_{j=1}^n p_{ij} \sqrt{a_{ij}}} \right) * \left( 1 - \frac{1}{\sqrt{Z}} \right)^{-1} * 100$$

where  $p_{ij}$  is the perimeter in meters,  $a_{ij}$  is the area in square meters and  $Z$  is the number of cells. COHESION is an 'Aggregation metric'.

**Units:** Percent

**Ranges:** Unknown

**Behaviour:** Unknown

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Schumaker, N. H. 1996. Using landscape indices to predict habitat connectivity. *Ecology*, 77(4), 1210-1225.

**See Also**

[lsm\\_p\\_perim](#), [lsm\\_p\\_area](#),  
[lsm\\_l\\_cohesion](#)

**Examples**

```
lsm_l_cohesion(landscape)
```

---

lsm_l_condent	<i>Conditional entropy (landscape level)</i>
---------------	--

---

## Description

Conditional entropy  $\backslash H(y|x) \backslash$

## Usage

```
lsm_l_condent(landscape, neighbourhood = 4, ordered = TRUE,
              base = "log2")
```

```
## S3 method for class 'RasterLayer'
lsm_l_condent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'RasterStack'
lsm_l_condent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'RasterBrick'
lsm_l_condent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'stars'
lsm_l_condent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'list'
lsm_l_condent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
neighbourhood	The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4.
ordered	The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE.
base	The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used.

## Details

Complexity of a landscape pattern configuration. It measures a only a geometric intricacy (configurational complexity) of a landscape pattern.

**Value**

tibble

**References**

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. <https://doi.org/10.1007/s10980-019-00830-x>

**See Also**

[lsm\\_l\\_ent](#), [lsm\\_l\\_mutinf](#), [lsm\\_l\\_joinent](#),

**Examples**

```
lsm_l_condent(landscape)
```

---

lsm_l_contag	<i>CONTAG (landscape level)</i>
--------------	---------------------------------

---

**Description**

Contagion (Aggregation metric)

**Usage**

```
lsm_l_contag(landscape, verbose)

## S3 method for class 'RasterLayer'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'list'
lsm_l_contag(landscape, verbose = TRUE)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
verbose	Print warning message if not sufficient patches are present

## Details

$$CONTAG = 1 + \frac{\sum_{q=1}^{n_a} p_q \ln(p_q)}{2 \ln(t)}$$

where  $p_q$  the adjacency table for all classes divided by the sum of that table and  $t$  the number of classes in the landscape.

CONTAG is an 'Aggregation metric'. It is based on cell adjacencies and describes the probability of two random cells belonging to the same class.  $p_q$  is the cell adjacency table, where the order is preserved and pairs of adjacent cells are counted twice. Contagion is affected by both the dispersion and interspersions of classes. E.g., low class dispersion (= high proportion of like adjacencies) and low interspersions (= uneven distribution of pairwise adjacencies) lead to a high contagion value.

The number of classes to calculate CONTAG must be  $\geq$  than 2.

**Units:** Percent

**Range:**  $0 < \text{Contag} \leq 100$

**Behaviour:** Approaches CONTAG = 0 if all cells are unevenly distributed and 100 indicates that all cells are equally adjacent to all other classes.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Riitters, K.H., O'Neill, R.V., Wickham, J.D. & Jones, K.B. (1996). A note on contagion indices for landscape analysis. *Landscape ecology*, 11, 197–202.

## Examples

```
lsm_l_contag(landscape)
```

---

lsm_l_contig_cv	<i>CONTIG_CV (landscape level)</i>
-----------------	------------------------------------

---

### Description

Coefficient of variation of Contiguity index (Shape metric)

### Usage

```
lsm_l_contig_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_contig_cv(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CONTIG_{CV} = cv(CONTIG[patch_{ij}])$$

where  $CONTIG[patch_{ij}]$  is the contiguity of each patch.

CONTIG\_CV is a 'Shape metric'. It summarises the landscape as the coefficient of variation of all patches in the landscape. CONTIG\_CV assesses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```



... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rook case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG\_CV  $\geq 0$

**Behaviour:** CONTIG\_CV = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

### See Also

[lsm\\_p\\_contig](#), [lsm\\_c\\_contig\\_sd](#), [lsm\\_c\\_contig\\_cv](#), [lsm\\_c\\_contig\\_mn](#),  
[lsm\\_l\\_contig\\_sd](#), [lsm\\_l\\_contig\\_mn](#)

### Examples

```
lsm_l_contig_cv(landscape)
```

---

lsm_l_contig_mn	<i>CONTIG_MN (landscape level)</i>
-----------------	------------------------------------

---

### Description

Mean of Contiguity index (Shape metric)

### Usage

```
lsm_l_contig_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
```

```
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_contig_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CONTIG_{MN} = mean(CONTIG[patch_i])$$

where  $CONTIG[patch_{ij}]$  is the contiguity of each patch.

CONTIG\_MN is a 'Shape metric'. It summarises the landscape as the mean of all patches in the landscape. CONTIG\_MN assesses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rook case result in larger contiguity index values.

**Units:** None

**Range:**  $0 \leq CONTIG\_MN \leq 1$

**Behaviour:** CONTIG equals the mean of the contiguity index on landscape level for all patches.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

**See Also**

[lsm\\_p\\_contig](#), [lsm\\_c\\_contig\\_sd](#), [lsm\\_c\\_contig\\_cv](#), [lsm\\_c\\_contig\\_mn](#),  
[lsm\\_l\\_contig\\_sd](#), [lsm\\_l\\_contig\\_cv](#)

**Examples**

```
lsm_l_contig_mn(landscape)
```

---

lsm_l_contig_sd	<i>CONTIG_SD (landscape level)</i>
-----------------	------------------------------------

---

**Description**

Standard deviation of Contiguity index (Shape metric)

**Usage**

```
lsm_l_contig_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_contig_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$CONTIG_{SD} = sd(CONTIG[patch_{ij}])$$

where  $CONTIG[patch_{ij}]$  is the contiguity of each patch.

CONTIG\_SD is a 'Shape metric'. It summarises the landscape as the standard deviation of all patches in the landscape. CONTIG\_SD assesses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rook case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG\_SD >= 0

**Behaviour:** CONTIG\_SD = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

## Value

tibble

## References

- McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>
- LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

## See Also

[lsm\\_p\\_contig](#), [lsm\\_c\\_contig\\_sd](#), [lsm\\_c\\_contig\\_cv](#), [lsm\\_c\\_contig\\_mn](#),  
[lsm\\_l\\_contig\\_cv](#), [lsm\\_l\\_contig\\_mn](#)

## Examples

```
lsm_l_contig_sd(landscape)
```

---

lsm_l_core_cv	<i>CORE_CV (landscape level)</i>
---------------	----------------------------------

---

### Description

Coefficient of variation of core area (Core area metric)

### Usage

```
lsm_l_core_cv(landscape, directions, consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
lsm_l_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
lsm_l_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
lsm_l_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_l_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
lsm_l_core_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$CORE_{CV} = cv(CORE[patch_{ij}])$$

where  $CORE[patch_{ij}]$  is the core area in square meters of each patch.

CORE\_CV is a 'Core area metric'. It equals the Coefficient of variation of the core area of each patch in the landscape. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among all patches in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** CORE\_CV  $\geq 0$

**Behaviour:** Equals CORE\_CV = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_core](#), [cv](#),  
[lsm\\_c\\_core\\_mn](#), [lsm\\_c\\_core\\_sd](#), [lsm\\_c\\_core\\_cv](#),  
[lsm\\_l\\_core\\_mn](#), [lsm\\_l\\_core\\_sd](#)

**Examples**

```
lsm_l_core_cv(landscape)
```

---

lsm_l_core_mn	<i>CORE_MN (landscape level)</i>
---------------	----------------------------------

---

**Description**

Mean of core area (Core area metric)

**Usage**

```

lsm_l_core_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_core_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$CORE_{MN} = mean(CORE[patch_{ij}])$$

where  $CORE[patch_{ij}]$  is the core area in square meters of each patch.

CORE\_MN is a 'Core area metric' and equals the mean of core areas of all patches in the landscape. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case).

**Units:** Hectares

**Range:** CORE\_MN  $\geq 0$

**Behaviour:** Equals CORE\_MN = 0 if CORE = 0 for all patches. Increases, without limit, as the core area indices increase.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_core](#), [mean](#),  
[lsm\\_c\\_core\\_mn](#), [lsm\\_c\\_core\\_sd](#), [lsm\\_c\\_core\\_cv](#),  
[lsm\\_l\\_core\\_sd](#), [lsm\\_l\\_core\\_cv](#)

**Examples**

```
lsm_l_core_mn(landscape)
```

---

lsm_l_core_sd	<i>CORE_SD (landscape level)</i>
---------------	----------------------------------

---

**Description**

Standard deviation of patch core area (class level)

**Usage**

```
lsm_l_core_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```



```
## S3 method for class 'list'
lsm_l_core_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$CORE_{SD} = sd(CORE[patch_{ij}])$$

where  $CORE[patch_{ij}]$  is the core area in square meters of each patch.

CORE\_SD is a 'Core area metric'. It equals the standard deviation of the core area of all patches in the landscape. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among all patches in the landscape.

**Units:** Hectares

**Range:** CORE\_SD >= 0

**Behaviour:** Equals CORE\_SD = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_core](#), [sd](#),  
[lsm\\_c\\_core\\_mn](#), [lsm\\_c\\_core\\_sd](#), [lsm\\_c\\_core\\_cv](#),  
[lsm\\_l\\_core\\_mn](#), [lsm\\_l\\_core\\_cv](#)

### Examples

```
lsm_l_core_sd(landscape)
```

---

lsm_l_dcad	<i>DCAD (landscape level)</i>
------------	-------------------------------

---

### Description

Disjunct core area density (core area metric)

### Usage

```
lsm_l_dcad(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_dcad(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$DCAD = \left( \frac{\sum_{i=1}^m \sum_{j=1}^n n_{ij}^{core}}{A} \right) * 10000 * 100$$

where  $n_{ij}^{core}$  is the number of disjunct core areas and  $A$  is the total landscape area in square meters. DCAD is a 'Core area metric'. It equals the number of disjunct core areas per 100 ha relative to the total area. A disjunct core area is a 'patch within the patch' containing only core cells. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric is relative and therefore comparable among landscapes with different total areas.

**Units:** Number per 100 hectares

**Range:** DCAD  $\geq 0$

**Behaviour:** Equals DCAD = 0 when DCORE = 0, i.e. no patch contains a disjunct core area. Increases, without limit, as disjunct core areas become more present, i.e. patches becoming larger and less complex.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_ndca](#), [lsm\\_l\\_ta](#),  
[lsm\\_c\\_dcad](#)

**Examples**

`lsm_l_dcad(landscape)`

---

lsm_l_dcore_cv	<i>DCORE_CV (landscape level)</i>
----------------	-----------------------------------

---

**Description**

Coefficient of variation number of disjunct core areas (Core area metric)

**Usage**

```

lsm_l_dcore_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_dcore_cv(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$DCORE_{CV} = cv(NCORE[patch_{ij}])$$

where  $NCORE[patch_{ij}]$  is the number of core areas.

$DCORE_{CV}$  is an 'Core area metric'. It summarises the landscape as the Coefficient of variation of all patches belonging to the landscape. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case).  $NCORE$  counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among all patches in the landscape and is easily comparable because it is scaled to the mean.

**Units:** None

**Range:** DCORE\_CV  $\geq 0$

**Behaviour:** Equals DCORE\_CV = 0 if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_ncore](#), [cv](#),  
[lsm\\_c\\_dcore\\_mn](#), [lsm\\_c\\_dcore\\_sd](#), [lsm\\_c\\_dcore\\_cv](#),  
[lsm\\_l\\_dcore\\_mn](#), [lsm\\_l\\_dcore\\_sd](#)

### Examples

```
lsm_l_dcore_cv(landscape)
```

---

lsm_l_dcore_mn	<i>DCORE_MN (landscape level)</i>
----------------	-----------------------------------

---

### Description

Mean number of disjunct core areas (Core area metric)

### Usage

```
lsm_l_dcore_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_l_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_dcore_mn(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$DCORE_{MN} = mean(NCORE[patch_{ij}])$$

where  $NCORE[patch_{ij}]$  is the number of core areas.

DCORE\_MN is an 'Core area metric'. It summarises the landscape as the mean of all patches in the landscape. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells.

**Units:** None

**Range:**  $DCORE_{MN} > 0$

**Behaviour:** Equals  $DCORE_{MN} = 0$  if  $NCORE = 0$  for all patches. Increases, without limit, as the number of disjunct core areas increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

lsm\_p\_ncore, mean,  
 lsm\_c\_dcore\_mn, lsm\_c\_dcore\_sd, lsm\_c\_dcore\_cv,  
 lsm\_l\_dcore\_sd, lsm\_l\_dcore\_cv

**Examples**

```
lsm_l_dcore_mn(landscape)
```

---

lsm_l_dcore_sd	<i>DCORE_SD (landscape level)</i>
----------------	-----------------------------------

---

**Description**

Standard deviation number of disjunct core areas (Core area metric)

**Usage**

```
lsm_l_dcore_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_dcore_sd(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

## Details

$$DCORE_{SD} = sd(NCORE[patch_{ij}])$$

where  $NCORE[patch_{ij}]$  is the number of core areas.

DCORE\_SD is an 'Core area metric'. It summarises the landscape as the standard deviation of all patches. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among all patches in the landscape.

**Units:** None

**Range:** DCORE\_SD  $\geq 0$

**Behaviour:** Equals DCORE\_SD = 0 if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[lsm\\_p\\_ncore](#), [sd](#),  
[lsm\\_c\\_dcore\\_mn](#), [lsm\\_c\\_dcore\\_sd](#),  
[lsm\\_c\\_dcore\\_cv](#), [lsm\\_l\\_dcore\\_mn](#), [lsm\\_l\\_dcore\\_cv](#)

## Examples

```
lsm_l_dcore_sd(landscape)
```



---

lsm_l_division	<i>DIVISION (landscape level)</i>
----------------	-----------------------------------

---

### Description

Landscape division index (Aggregation metric)

### Usage

```
lsm_l_division(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_division(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_division(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_division(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_division(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_division(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$DIVISION = (1 - \sum_{i=1}^m \sum_{j=1}^n (\frac{a_{ij}}{A})^2)$$

where  $a_{ij}$  is the area in square meters and  $A$  is the total landscape area in square meters.

DIVISION is an 'Aggregation metric'. It can be in as the probability that two randomly selected cells are not located in the same patch. The landscape division index is negatively correlated with the effective mesh size ([lsm\\_c\\_mesh](#)).

**Units:** Proportion

**Ranges:**  $0 \leq \text{Division} < 1$

**Behaviour:** Equals  $DIVISION = 0$  if only one patch is present. Approaches  $DIVISION = 1$  if all patches of class  $i$  are single cells.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. *Landscape ecology*, 15(2), 115-130.

**See Also**

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_c\\_division](#)

**Examples**

```
lsm_l_division(landscape)
```

---

lsm_l_ed	<i>ED (landscape level)</i>
----------	-----------------------------

---

**Description**

Edge Density (Area and Edge metric)

**Usage**

```
lsm_l_ed(landscape, count_boundary, directions)

## S3 method for class 'RasterLayer'
lsm_l_ed(landscape, count_boundary = FALSE,
         directions = 8)

## S3 method for class 'RasterStack'
lsm_l_ed(landscape, count_boundary = FALSE,
         directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_ed(landscape, count_boundary = FALSE,
         directions = 8)

## S3 method for class 'stars'
lsm_l_ed(landscape, count_boundary = FALSE,
         directions = 8)
```

```
## S3 method for class 'list'
lsm_l_ed(landscape, count_boundary = FALSE,
         directions = 8)
```

### Arguments

**landscape** Raster\* Layer, Stack, Brick or a list of rasterLayers.  
**count\_boundary** Count landscape boundary as edge  
**directions** The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$ED = \frac{E}{A} * 10000$$

where  $E$  is the total landscape edge in meters and  $A$  is the total landscape area in square meters.

ED is an 'Area and Edge metric'. The edge density equals all edges in the landscape in relation to the landscape area. The boundary of the landscape is only included in the corresponding total class edge length if `count_boundary = TRUE`. The metric describes the configuration of the landscape, e.g. because an overall aggregation of classes will result in a low edge density. The metric is standardized to the total landscape area, and therefore comparisons among landscapes with different total areas are possible.

**Units:** Meters per hectare

**Range:**  $ED \geq 0$

**Behaviour:** Equals  $ED = 0$  if only one patch is present (and the landscape boundary is not included) and increases, without limit, as the landscapes becomes more patchy

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_l\\_te](#), [lsm\\_l\\_ta](#),  
[lsm\\_c\\_ed](#)

### Examples

```
lsm_l_ed(landscape)
```

---

lsm_l_enn_cv	<i>ENN_CV (landscape level)</i>
--------------	---------------------------------

---

### Description

Coefficient of variation of euclidean nearest-neighbor distance (Aggregation metric)

### Usage

```
lsm_l_enn_cv(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_l_enn_cv(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_enn_cv(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_enn_cv(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'stars'
lsm_l_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_l_enn_cv(landscape, directions = 8, verbose = TRUE)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
verbose	Print warning message if not sufficient patches are present

### Details

$$ENN_{CV} = cv(ENN[patch_{ij}])$$

where  $ENN[patch_{ij}]$  is the euclidean nearest-neighbor distance of each patch.

ENN\_CV is an 'Aggregation metric'. It summarises the landscape as the Coefficient of variation of all patches in the landscape. ENN measures the distance to the nearest neighbouring patch of the same class  $i$ . The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN\_CV  $\geq 0$

**Behaviour:** Equals ENN\_CV = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. *Ecological monographs*, 65(3), 235-260.

## See Also

[lsm\\_p\\_enn, cv,](#)  
[lsm\\_c\\_enn\\_mn, lsm\\_c\\_enn\\_sd, lsm\\_c\\_enn\\_cv,](#)  
[lsm\\_l\\_enn\\_mn, lsm\\_l\\_enn\\_sd,](#)

## Examples

```
lsm_l_enn_cv(landscape)
```

---

lsm_l_enn_mn	<i>ENN_MN (landscape level)</i>
--------------	---------------------------------

---

## Description

Mean of euclidean nearest-neighbor distance (Aggregation metric)

## Usage

```
lsm_l_enn_mn(landscape, directions, verbose)
```

```
## S3 method for class 'RasterLayer'
lsm_l_enn_mn(landscape, directions = 8,
  verbose = TRUE)
```

```
## S3 method for class 'RasterStack'
lsm_l_enn_mn(landscape, directions = 8,
  verbose = TRUE)
```

```
## S3 method for class 'RasterBrick'
lsm_l_enn_mn(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'stars'
lsm_l_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_l_enn_mn(landscape, directions = 8, verbose = TRUE)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
verbose	Print warning message if not sufficient patches are present

### Details

$$ENN_{MN} = cv(mean[patch_{ij}])$$

where  $ENN[patch_{ij}]$  is the euclidean nearest-neighbor distance of each patch.

ENN\_CV is an 'Aggregation metric'. It summarises the landscape as the mean of all patches in the landscape. ENN measures the distance to the nearest neighbouring patch of the same class  $i$ . The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit.

**Units:** Meters

**Range:**  $ENN_{MN} > 0$

**Behaviour:** Approaches  $ENN_{MN} = 0$  as the distance to the nearest neighbour decreases, i.e. patches of the same class  $i$  are more aggregated. Increases, without limit, as the distance between neighbouring patches of the same class  $i$  increases, i.e. patches are more isolated.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. *Ecological monographs*, 65(3), 235-260.

See Also

```
lsm_p_enn, mean,  
lsm_c_enn_mn, lsm_c_enn_sd, lsm_c_enn_cv,  
lsm_l_enn_sd, lsm_l_enn_cv
```

Examples

```
lsm_l_enn_mn(landscape)
```

---

lsm_l_enn_sd	<i>ENN_SD (landscape level)</i>
--------------	---------------------------------

---

Description

Standard deviation of euclidean nearest-neighbor distance (Aggregation metric)

Usage

```
lsm_l_enn_sd(landscape, directions, verbose)  
  
## S3 method for class 'RasterLayer'  
lsm_l_enn_sd(landscape, directions = 8,  
             verbose = TRUE)  
  
## S3 method for class 'RasterStack'  
lsm_l_enn_sd(landscape, directions = 8,  
             verbose = TRUE)  
  
## S3 method for class 'RasterBrick'  
lsm_l_enn_sd(landscape, directions = 8,  
             verbose = TRUE)  
  
## S3 method for class 'stars'  
lsm_l_enn_sd(landscape, directions = 8, verbose = TRUE)  
  
## S3 method for class 'list'  
lsm_l_enn_sd(landscape, directions = 8, verbose = TRUE)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).
verbose	Print warning message if not sufficient patches are present

## Details

$$ENN_{SD} = sd(ENN[patch_{ij}])$$

where  $ENN[patch_{ij}]$  is the euclidean nearest-neighbor distance of each patch.

ENN\_CV is an 'Aggregation metric'. It summarises in the landscape as the standard deviation of all patches in the landscape. ENN measures the distance to the nearest neighbouring patch of the same class *i*. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN\_SD >= 0

**Behaviour:** Equals ENN\_SD = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. *Ecological monographs*, 65(3), 235-260.

## See Also

[lsm\\_p\\_enn](#), [sd](#),  
[lsm\\_c\\_enn\\_mn](#), [lsm\\_c\\_enn\\_sd](#), [lsm\\_c\\_enn\\_cv](#),  
[lsm\\_l\\_enn\\_mn](#), [lsm\\_l\\_enn\\_cv](#)

## Examples

```
lsm_l_enn_sd(landscape)
```



---

lsm_l_ent	<i>ENT (landscape level)</i>
-----------	------------------------------

---

## Description

Marginal entropy  $\sqrt{H(x)}$

## Usage

```
lsm_l_ent(landscape, neighbourhood, base)

## S3 method for class 'RasterLayer'
lsm_l_ent(landscape, neighbourhood = 4,
  base = "log2")

## S3 method for class 'RasterStack'
lsm_l_ent(landscape, neighbourhood = 4,
  base = "log2")

## S3 method for class 'RasterBrick'
lsm_l_ent(landscape, neighbourhood = 4,
  base = "log2")

## S3 method for class 'stars'
lsm_l_ent(landscape, neighbourhood = 4, base = "log2")

## S3 method for class 'list'
lsm_l_ent(landscape, neighbourhood = 4, base = "log2")
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
neighbourhood	The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4.
base	The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used.

## Details

It measures a diversity (thematic complexity) of landscape classes.

## Value

tibble

References

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. <https://doi.org/10.1007/s10980-019-00830-x>

See Also

[lsm\\_l\\_condent](#), [lsm\\_l\\_mutinf](#), [lsm\\_l\\_joinent](#),

Examples

```
lsm_l_ent(landscape)
```

---

lsm_l_frac_cv	<i>FRAC_CV (landscape level)</i>
---------------	----------------------------------

---

Description

Coefficient of variation fractal dimension index (Shape metric)

Usage

```
lsm_l_frac_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_frac_cv(landscape, directions = 8)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

**Details**

$$FRAC_{CV} = cv(FRAC[patch_{ij}])$$

where  $FRAC[patch_{ij}]$  equals the fractal dimension index of each patch.

FRAC\_CV is a 'Shape metric'. The metric summarises the landscape as the Coefficient of variation of the fractal dimension index of all patches in the landscape. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:** FRAC\_CV  $\geq 0$

**Behaviour:** Equals FRAC\_CV = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

**See Also**

[lsm\\_p\\_frac](#), [cv](#),  
[lsm\\_c\\_frac\\_mn](#), [lsm\\_c\\_frac\\_sd](#), [lsm\\_c\\_frac\\_cv](#),  
[lsm\\_l\\_frac\\_mn](#), [lsm\\_l\\_frac\\_sd](#),

**Examples**

```
lsm_l_frac_cv(landscape)
```

---

lsm_l_frac_mn	<i>FRAC_MN (landscape level)</i>
---------------	----------------------------------

---

### Description

Mean fractal dimension index (Shape metric)

### Usage

```
lsm_l_frac_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_frac_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$FRAC_{MN} = mean(FRAC[patch_{ij}])$$

where  $FRAC[patch_{ij}]$  equals the fractal dimension index of each patch.

FRAC\_MN is a 'Shape metric'. The metric summarises the landscape as the mean of the fractal dimension index of all patches in the landscape. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:**  $FRAC_{MN} > 0$

**Behaviour:** Approaches  $FRAC_{MN} = 1$  if all patches are squared and  $FRAC_{MN} = 2$  if all patches are irregular.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

**See Also**

[lsm\\_p\\_frac](#), [mean](#),  
[lsm\\_c\\_frac\\_mn](#), [lsm\\_c\\_frac\\_sd](#), [lsm\\_c\\_frac\\_cv](#),  
[lsm\\_l\\_frac\\_sd](#), [lsm\\_l\\_frac\\_cv](#)

**Examples**

```
lsm_l_frac_mn(landscape)
```

---

lsm_l_frac_sd	<i>FRAC_SD (landscape level)</i>
---------------	----------------------------------

---

**Description**

Standard deviation fractal dimension index (Shape metric)

**Usage**

```
lsm_l_frac_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_frac_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$FRAC_{SD} = sd(FRAC[patch_{ij}])$$

where  $FRAC[patch_{ij}]$  equals the fractal dimension index of each patch.

FRAC\_SD is a 'Shape metric'. The metric summarises the landscape as the standard deviation of the fractal dimension index of all patches in the landscape. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity.

**Units:** None

**Range:** FRAC\_SD  $\geq 0$

**Behaviour:** Equals FRAC\_SD = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

**See Also**

[lsm\\_p\\_frac](#), [sd](#),  
[lsm\\_c\\_frac\\_mn](#), [lsm\\_c\\_frac\\_sd](#), [lsm\\_c\\_frac\\_cv](#),  
[lsm\\_l\\_frac\\_mn](#), [lsm\\_l\\_frac\\_cv](#)

**Examples**

```
lsm_l_frac_sd(landscape)
```

---

lsm_l_gyrate_cv	<i>GYRATE_CV (landscape level)</i>
-----------------	------------------------------------

---

### Description

Coefficient of variation radius of gyration (Area and edge metric)

### Usage

```
lsm_l_gyrate_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_gyrate_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_gyrate_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_gyrate_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_gyrate_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_gyrate_cv(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$GYRATE_{CV} = cv(GYRATE[patch_{ij}])$$

where  $GYRATE[patch_{ij}]$  equals the radius of gyration of each patch.

GYRATE\_CV is an 'Area and edge metric'. The metric summarises the landscape as the Coefficient of variation of the radius of gyration of all patches in the landscape. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** Meters

**Range:** GYRATE\_CV >= 0

**Behaviour:** Equals GYRATE\_CV = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

**See Also**

[lsm\\_p\\_gyrate\\_cv](#),  
[lsm\\_c\\_gyrate\\_mn](#), [lsm\\_c\\_gyrate\\_sd](#), [lsm\\_c\\_gyrate\\_cv](#),  
[lsm\\_l\\_gyrate\\_mn](#), [lsm\\_l\\_gyrate\\_sd](#)

**Examples**

```
lsm_l_gyrate_cv(landscape)
```

---

lsm_l_gyrate_mn	<i>GYRATE_MN (landscape level)</i>
-----------------	------------------------------------

---

**Description**

Mean radius of gyration (Area and edge metric)

**Usage**

```
lsm_l_gyrate_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_gyrate_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_gyrate_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_gyrate_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_gyrate_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_gyrate_mn(landscape, directions = 8)
```



**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$GYRATE_{MN} = mean(GYRATE[patch_{ij}])$$

where  $GYRATE[patch_{ij}]$  equals the radius of gyration of each patch.

GYRATE\_MN is an 'Area and edge metric'. The metric summarises the landscape as the mean of the radius of gyration of all patches in the landscape. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** Meters

**Range:** GYRATE\_MN  $\geq 0$

**Behaviour:** Approaches GYRATE\_MN = 0 if every patch is a single cell. Increases, without limit, when only one patch is present.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

**See Also**

[lsm\\_p\\_gyrate](#), [mean](#),  
[lsm\\_c\\_gyrate\\_mn](#), [lsm\\_c\\_gyrate\\_sd](#), [lsm\\_c\\_gyrate\\_cv](#),  
[lsm\\_l\\_gyrate\\_sd](#), [lsm\\_l\\_gyrate\\_cv](#)

**Examples**

```
lsm_l_gyrate_mn(landscape)
```

---

lsm_l_gyrate_sd	<i>GYRATE_SD (landscape level)</i>
-----------------	------------------------------------

---

### Description

Standard deviation radius of gyration (Area and edge metric)

### Usage

```
lsm_l_gyrate_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_gyrate_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_gyrate_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_gyrate_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_gyrate_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_gyrate_sd(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$GYRATE_{SD} = sd(GYRATE[patch_{ij}])$$

where  $GYRATE[patch_{ij}]$  equals the radius of gyration of each patch.

GYRATE\_SD is an 'Area and edge metric'. The metric summarises the landscape as the standard deviation of the radius of gyration of all patches in the landscape. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness.

**Units:** Meters

**Range:** GYRATE\_SD >= 0

**Behaviour:** Equals GYRATE\_SD = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. *Conservation ecology*, 1(1).

**See Also**

[lsm\\_p\\_gyrate\\_cv](#),  
[lsm\\_c\\_gyrate\\_mn](#), [lsm\\_c\\_gyrate\\_sd](#), [lsm\\_c\\_gyrate\\_cv](#),  
[lsm\\_l\\_gyrate\\_mn](#), [lsm\\_l\\_gyrate\\_cv](#)

**Examples**

```
lsm_l_gyrate_sd(landscape)
```

---

lsm_l_iji	<i>Interspersion and Juxtaposition index (landscape level)</i>
-----------	--

---

**Description**

Interspersion and Juxtaposition index (Aggregation metric)

**Usage**

```
lsm_l_iji(landscape, verbose)

## S3 method for class 'RasterLayer'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'list'
lsm_l_iji(landscape, verbose = TRUE)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
verbose	Print warning message if not sufficient patches are present

**Details**

$$IJI = \frac{-\sum_{i=1}^m \sum_{k=i+1}^m \left[ \left( \frac{e_{ik}}{E} \right) \ln \left( \frac{e_{ik}}{E} \right) \right]}{\ln(0.5[m(m-1)])} * 100$$

where  $e_{ik}$  are the unique adjacencies of all classes (lower/upper triangle of the adjacency table - without the diagonal),  $E$  is the total length of edges in the landscape and  $m$  is the number of classes.

IJI is an 'Aggregation metric'. It is a so called "salt and pepper" metric and describes the intermixing of classes (i.e. without considering like adjacencies - the diagonal of the adjacency table). The number of classes to calculate IJI must be  $\geq$  than 3.

**Units:** Percent

**Range:**  $0 < IJI \leq 100$

**Behaviour:** Approaches 0 if a class is only adjacent to a single other class and equals 100 when a class is equally adjacent to all other classes.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., & Marks, B. J. 1995. FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. Gen. Tech. Rep. PNW-GTR-351. Portland, OR: US Department of Agriculture, Forest Service, Pacific Northwest Research Station. 122 p, 351.

**See Also**

[lsm\\_c\\_iji](#)

**Examples**

```
lsm_l_iji(landscape)
```

---

lsm_l_joinent	<i>JOINENT (landscape level)</i>
---------------	----------------------------------

---

### Description

Joint entropy  $\sqrt{H(x, y)}$

### Usage

```
lsm_l_joinent(landscape, neighbourhood, ordered, base)
```

```
## S3 method for class 'RasterLayer'
lsm_l_joinent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'RasterStack'
lsm_l_joinent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'RasterBrick'
lsm_l_joinent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'stars'
lsm_l_joinent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

```
## S3 method for class 'list'
lsm_l_joinent(landscape, neighbourhood = 4,
              ordered = TRUE, base = "log2")
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
neighbourhood	The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4.
ordered	The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE.
base	The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used.

### Details

Complexity of a landscape pattern. An overall spatio-thematic complexity metric.

**Value**

tibble

**References**

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. <https://doi.org/10.1007/s10980-019-00830-x>

**See Also**

[lsm\\_l\\_ent](#), [lsm\\_l\\_condent](#), [lsm\\_l\\_mutinf](#)

**Examples**

```
lsm_l_joinent(landscape)
```

---

lsm_l_lpi	<i>LPI (landscape level)</i>
-----------	------------------------------

---

**Description**

Largest patch index (Area and Edge metric)

**Usage**

```
lsm_l_lpi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_lpi(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

**Details**

$$LPI = \frac{\max(a_{ij})}{A} * 100$$

where  $\max(a_{ij})$  is the area of the patch in square meters and  $A$  is the total landscape area in square meters.

The largest patch index is an 'Area and edge metric'. It is the percentage of the landscape covered by the largest patch in the landscape. It is a simple measure of dominance.

**Units:** Percentage

**Range:**  $0 < LPI \leq 100$

**Behaviour:** Approaches  $LPI = 0$  when the largest patch is becoming small and equals  $LPI = 100$  when only one patch is present

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_c\\_lpi](#)

**Examples**

```
lsm_l_lpi(landscape)
```

---

lsm\_l\_lsi

*LSI (landscape level)*


---

**Description**

Landscape shape index (Aggregation metric)

**Usage**

```

lsm_l_lsi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_lsi(landscape, directions = 8)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$LSI = \frac{E}{\min E}$$

where  $E$  is the total edge length in cell surfaces and  $\min E$  is the minimum total edge length in cell surfaces

LSI is an 'Aggregation metric'. It is the ratio between the actual landscape edge length and the hypothetical minimum edge length. The minimum edge length equals the edge length if only one patch would be present.

**Units:** None

**Ranges:**  $LSI \geq 1$

**Behaviour:** Equals  $LSI = 1$  when only one squared patch is present. Increases, without limit, as the length of the actual edges increases, i.e. the patches become less compact.

**Value**

tibble



References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

See Also

[lsm\\_p\\_shape](#),  
[lsm\\_c\\_lsi](#)

Examples

```
lsm_l_lsi(landscape)
```

---

lsm_l_mesh	<i>MESH (landscape level)</i>
------------	-------------------------------

---

Description

Effective Mesh Size (Aggregation metric)

Usage

```
lsm_l_mesh(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_mesh(landscape, directions = 8)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$MESH = \frac{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}{A} * \frac{1}{10000}$$

where  $a_{ij}$  is the patch area in square meters and  $A$  is the total landscape area in square meters.

The effective mesh size is an 'Aggregation metric'. Because each patch is squared before the sum is calculated and the sum is standardized by the total landscape area, MESH is a relative measure of patch structure. MESH is perfectly, negatively correlated to [lsm\\_c\\_division](#).

**Units:** Hectares

**Range:** cell size / total area <= MESH <= total area

**Behaviour:** Equals cellsize/total area if class covers only one cell and equals total area if only one patch is present.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. *Landscape ecology*, 15(2), 115-130.

## See Also

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_c\\_mesh](#)

## Examples

```
lsm_l_mesh(landscape)
```

---

lsm_l_msidi	<i>MSIDI (landscape level)</i>
-------------	--------------------------------

---

### Description

Modified Simpson's diversity index (Diversity metric)

### Usage

```
lsm_l_msidi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_msidi(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$MSIDI = -\ln \sum_{i=1}^m P_i^2$$

where  $P_i$  is the landscape area proportion of class i.

MSIDI is a 'Diversity metric'.

**Units:** None

**Range:** MSIDI  $\geq 0$

**Behaviour:** MSIDI = 0 if only one patch is present and increases, without limit, as the amount of patches with equally distributed landscape proportions increases

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Simpson, E. H. 1949. Measurement of diversity. Nature 163:688

Pielou, E. C. 1975. Ecological Diversity. Wiley-Interscience, New York.

Romme, W. H. 1982. Fire and landscapediversity in subalpine forests of Yellowstone National Park.Ecol.Monogr. 52:199-221

**See Also**

[lsm\\_l\\_sidi](#)

**Examples**

```
lsm_l_msidi(landscape)
```

---

lsm_l_msiei	<i>MSIEI (landscape level)</i>
-------------	--------------------------------

---

**Description**

Modified Simpson’s evenness index (Diversity metric)

**Usage**

```
lsm_l_msiei(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_msiei(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$MSIEi = \frac{-\ln \sum_{i=1}^m P_i^2}{\ln m}$$

where  $P_i$  is the landscape area proportion of class i. MSIEI is a 'Diversity metric'.

**Units:** None

**Range:**  $0 \leq MSIEI < 1$

**Behaviour:** MSIEI = 0 when only one patch is present and approaches MSIEI = 1 as the proportional distribution of patches becomes more even

**Value**

tibble

**References**

- McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>
- Simpson, E. H. 1949. Measurement of diversity. Nature 163:688
- Pielou, E. C. 1975. Ecological Diversity. Wiley-Interscience, New York.

**See Also**

[lsm\\_l\\_siei](#)

**Examples**

```
lsm_l_msiei(landscape)
```

---

lsm_l_mutinf	<i>MUTINF (landscape level)</i>
--------------	---------------------------------

---

**Description**

Mutual information  $\backslash I(y,x)\backslash$

**Usage**

```
lsm_l_mutinf(landscape, neighbourhood, ordered, base)

## S3 method for class 'RasterLayer'
lsm_l_mutinf(landscape, neighbourhood = 4,
  ordered = TRUE, base = "log2")

## S3 method for class 'RasterStack'
lsm_l_mutinf(landscape, neighbourhood = 4,
  ordered = TRUE, base = "log2")

## S3 method for class 'RasterBrick'
lsm_l_mutinf(landscape, neighbourhood = 4,
  ordered = TRUE, base = "log2")

## S3 method for class 'stars'
lsm_l_mutinf(landscape, neighbourhood = 4,
  ordered = TRUE, base = "log2")

## S3 method for class 'list'
lsm_l_mutinf(landscape, neighbourhood = 4,
  ordered = TRUE, base = "log2")
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
neighbourhood	The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4.
ordered	The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE.
base	The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used.

**Details**

It disambiguates landscape pattern types characterize by the same value of an overall complexity ([lsm\\_l\\_joinent](#)).

**Value**

tibble

**References**

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. <https://doi.org/10.1007/s10980-019-00830-x>

**See Also**

[lsm\\_l\\_ent](#), [lsm\\_l\\_condent](#), [lsm\\_l\\_joinent](#),

**Examples**

```
lsm_l_mutinf(landscape)
```

---

lsm_l_ndca	<i>NDCA (landscape level)</i>
------------	-------------------------------

---

**Description**

Number of disjunct core areas (Core area metric)

**Usage**

```
lsm_l_ndca(landscape, directions, consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
```

```
lsm_l_ndca(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
```

```
lsm_l_ndca(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
```

```
lsm_l_ndca(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
```

```
lsm_l_ndca(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
```

```
lsm_l_ndca(landscape, directions = 8,  
  consider_boundary = FALSE, edge_depth = 1)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$NDCA = \sum_{i=1}^m \sum_{j=1}^n n_{ij}^{core}$$

where  $n_{ij}^{core}$  is the number of disjunct core areas.

NDCA is a 'Core area metric'. The metric summarises the landscape as the sum of all patches in the landscape. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NDCA counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. It describes patch area and shape simultaneously (more core area when the patch is large, however, the shape must allow disjunct core areas). Thereby, a compact shape (e.g. a square) will contain less disjunct core areas than a more irregular patch.

**Units:** None

**Range:** NDCA  $\geq 0$

**Behaviour:** NDCA = 0 when TCA = 0, i.e. every cell in the landscape is an edge cell. NDCA increases, without limit, as core area increases and patch shapes allow disjunct core areas (i.e. patch shapes become rather complex).

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_tca](#),  
[lsm\\_p\\_ncore](#), [lsm\\_c\\_ndca](#)



Examples

```
lsm_l_ndca(landscape)
```

---

lsm_l_np	<i>NP (landscape level)</i>
----------	-----------------------------

---

Description

Number of patches (Aggregation metric)

Usage

```
lsm_l_np(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_np(landscape, directions = 8)
```

Arguments

- landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.
- directions      The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

Details

$$NP = N$$

where  $N$  is the number of patches.

NP is an 'Aggregation metric'. It describes the fragmentation of the landscape, however, does not necessarily contain information about the configuration or composition of the landscape.

**Units:** None

**Ranges:** NP >= 1

**Behaviour:** Equals NP = 1 when only one patch is present and increases, without limit, as the number of patches increases

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_np](#)

**Examples**

```
lsm_l_np(landscape)
```

---

lsm_l_pafrac	<i>PAFRAC (landscape level)</i>
--------------	---------------------------------

---

**Description**

Perimeter-Area Fractal Dimension (Shape metric)

**Usage**

```
lsm_l_pafrac(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_l_pafrac(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_pafrac(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_pafrac(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'stars'
```

```
lsm_l_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_l_pafrac(landscape, directions = 8, verbose = TRUE)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
verbose	Print warning message if not sufficient patches are present

### Details

$$PAFRAC = \frac{2}{\beta}$$

where  $\beta$  is the slope of the regression of the area against the perimeter (logarithm)  $N \sum_{i=1}^m \sum_{j=1}^n \ln a_{ij} =$

$$a + \beta N \sum_{i=1}^m \sum_{j=1}^n \ln p_{ij}$$

PAFRAC is a 'Shape metric'. It describes the patch complexity of the landscape while being scale independent. This means that increasing the patch size while not changing the patch form will not change the metric. However, it is only meaningful if the relationship between the area and perimeter is linear on a logarithmic scale. Furthermore, if there are less than 10 patches in the landscape, the metric returns NA because of the small-sample issue.

**Units:** None

**Range:**  $1 \leq PAFRAC \leq 2$

**Behaviour:** Approaches PAFRAC = 1 for patches with simple shapes and approaches PAFRAC = 2 for irregular shapes

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Burrough, P. A. 1986. Principles of Geographical Information Systems for Land Resources Assessment. Monographs on Soil and Resources Survey No. 12. Clarendon Press, Oxford

### See Also

[lsm\\_p\\_area](#), [lsm\\_p\\_perim](#),  
[lsm\\_c\\_pafrac](#)

Examples

```
lsm_l_pafrac(landscape)
```

---

lsm_l_para_cv	<i>PARA_CV (landscape level)</i>
---------------	----------------------------------

---

Description

Coefficient of variation perimeter-area ratio (Shape metric)

Usage

```
lsm_l_para_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_para_cv(landscape, directions = 8)
```

Arguments

- landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.
- directions      The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

Details

$$PARA_{CV} = cv(PARA[patch_{ij}])$$

where  $PARA[patch_{ij}]$  is the perimeter area ratio of each patch.

PARA\_CV is a ‘Shape metric’. It summarises the landscape as the Coefficient of variation of each patch belonging in the landscape The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA\_CV  $\geq 0$

**Behaviour:** Equals PARA\_CV = 0 if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_para\\_cv](#),  
[lsm\\_c\\_para\\_mn](#), [lsm\\_c\\_para\\_sd](#), [lsm\\_c\\_para\\_cv](#),  
[lsm\\_l\\_para\\_mn](#), [lsm\\_l\\_para\\_sd](#)

### Examples

```
lsm_l_para_cv(landscape)
```

---

lsm_l_para_mn	<i>PARA_MN (landscape level)</i>
---------------	----------------------------------

---

### Description

Mean perimeter-area ratio (Shape metric)

### Usage

```
lsm_l_para_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_para_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_para_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_para_mn(landscape, directions = 8)
```

```
## S3 method for class 'stars'
lsm_l_para_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_para_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$PARA_{MN} = mean(PARA[patch_{ij}])$$

where  $PARA[patch_{ij}]$  is the perimeter area ratio of each patch.

PARA\_MN is a 'Shape metric'. It summarises the landscape as the mean of each patch in the landscape. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standardised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:**  $PARA_{MN} > 0$

**Behaviour:** Approaches  $PARA_{MN} > 0$  if PARA for each patch approaches  $PARA > 0$ , i.e. the form approaches a rather small square. Increases, without limit, as PARA increases, i.e. patches become more complex.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_para](#), [mean](#),  
[lsm\\_c\\_para\\_mn](#), [lsm\\_c\\_para\\_sd](#), [lsm\\_c\\_para\\_cv](#),  
[lsm\\_l\\_para\\_sd](#), [lsm\\_l\\_para\\_cv](#)

### Examples

```
lsm_l_para_mn(landscape)
```

---

lsm_l_para_sd	<i>PARA_SD (landscape level)</i>
---------------	----------------------------------

---

### Description

Standard deviation perimeter-area ratio (Shape metric)

### Usage

```
lsm_l_para_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_para_sd(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$PARA_{SD} = sd(PARA[patch_{ij}])$$

where  $PARA[patch_{ij}]$  is the perimeter area ratio of each patch.

$PARA_{SD}$  is a 'Shape metric'. It summarises the landscape as the standard deviation of each patch belonging in the landscape. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standardised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:**  $PARA_{SD} \geq 0$

**Behaviour:** Equals  $PARA_{SD} = 0$  if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_para](#), [sd](#),  
[lsm\\_c\\_para\\_mn](#), [lsm\\_c\\_para\\_sd](#), [lsm\\_c\\_para\\_cv](#),  
[lsm\\_l\\_para\\_mn](#), [lsm\\_l\\_para\\_cv](#)

**Examples**

```
lsm_l_para_sd(landscape)
```

---

lsm_l_pd	<i>PD (landscape level)</i>
----------	-----------------------------

---

**Description**

Patch density (Aggregation metric)

**Usage**

```
lsm_l_pd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_pd(landscape, directions = 8)
```



**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$PD = \frac{N}{A} * 10000 * 100$$

where  $N$  is the number of patches and  $A$  is the total landscape area in square meters.

PD is an 'Aggregation metric'. It describes the fragmentation the landscape, however, does not necessarily contain information about the configuration or composition of the landscape. In contrast to [lsm\\_l\\_np](#) it is standardized to the area and comparisons among landscapes with different total area are possible.

**Units:** Number per 100 hectares

**Ranges:**  $0 < PD \leq 1e+06$

**Behaviour:** Increases as the landscape gets more patchy. Reaches its maximum if every cell is a different patch.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_np](#), [lsm\\_l\\_ta](#),  
[lsm\\_c\\_pd](#)

**Examples**

```
lsm_l_pd(landscape)
```

---

lsm_l_pladj	<i>PLADJ (landscape level)</i>
-------------	--------------------------------

---

### Description

Percentage of Like Adjacencies (Aggregation metric)

### Usage

```
lsm_l_pladj(landscape)

## S3 method for class 'RasterLayer'
lsm_l_pladj(landscape)

## S3 method for class 'RasterStack'
lsm_l_pladj(landscape)

## S3 method for class 'RasterBrick'
lsm_l_pladj(landscape)

## S3 method for class 'stars'
lsm_l_pladj(landscape)

## S3 method for class 'list'
lsm_l_pladj(landscape)
```

### Arguments

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.

### Details

$$PLADJ = \left( \frac{g_{ii}}{\sum_{k=1}^m g_{ik}} \right) * 100$$

where  $g_{ii}$  is the number of adjacencies between cells of class i and  $g_{ik}$  is the number of adjacencies between cells of class i and k.

PLADJ is an 'Aggregation metric'. It calculates the frequency how often patches of different classes i (focal class) and k are next to each other, and following is a measure of class aggregation. The adjacencies are counted using the double-count method.

**Units:** Percent

**Ranges:** 0 <= PLADJ <= 100

**Behaviour:** Equals PLADJ = 0 if class i is maximal disaggregated, i.e. every cell is a different patch. Equals PLADJ = 100 when the only one patch is present.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**Examples**

```
lsm_l_pladj(landscape)
```

---

lsm_l_pr	<i>PR (landscape level)</i>
----------	-----------------------------

---

**Description**

Patch richness (Diversity metric)

**Usage**

```
lsm_l_pr(landscape)

## S3 method for class 'RasterLayer'
lsm_l_pr(landscape)

## S3 method for class 'RasterStack'
lsm_l_pr(landscape)

## S3 method for class 'RasterBrick'
lsm_l_pr(landscape)

## S3 method for class 'stars'
lsm_l_pr(landscape)

## S3 method for class 'list'
lsm_l_pr(landscape)
```

**Arguments**

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.

Details

$$PR = m$$

where  $m$  is the number of classes

PR is a 'Diversity metric'. It is one of the simplest diversity and composition measures. However, because of its absolute nature, it is not comparable among landscapes with different total areas.

- Units:** None
- Range:**  $PR \geq 1$
- Behaviour:** Equals  $PR = 1$  when only one patch is present and increases, without limit, as the number of classes increases

Value

tibble

References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Examples

```
lsm_l_pr(landscape)
```

---

lsm_l_prd	<i>PRD (landscape level)</i>
-----------	------------------------------

---

Description

Patch richness density (Diversity metric)

Usage

```
lsm_l_prd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
```

```
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_prd(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$PRD = \frac{m}{A} * 10000 * 100$$

where  $m$  is the number of classes and  $A$  is the total landscape area in square meters.

PRD is a 'Diversity metric'. It is one of the simplest diversity and composition measures. In contrast to `lsm_l_pr`, it is a relative measure and following, comparable among landscapes with different total landscape areas.

**Units:** Number per 100 hectares

**Range:**  $PR > 0$

**Behaviour:** Approaches  $PRD > 1$  when only one patch is present and the landscape is rather large. Increases, without limit, as the number of classes increases and the total landscape area decreases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### Examples

```
lsm_l_prd(landscape)
```

---

lsm_l_rpr	<i>RPD (landscape level)</i>
-----------	------------------------------

---

### Description

Relative patch richness (Diversity metric)

### Usage

```
lsm_l_rpr(landscape, classes_max, verbose)

## S3 method for class 'RasterLayer'
lsm_l_rpr(landscape, classes_max = NULL,
  verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_rpr(landscape, classes_max = NULL,
  verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_rpr(landscape, classes_max = NULL,
  verbose = TRUE)

## S3 method for class 'stars'
lsm_l_rpr(landscape, classes_max = NULL,
  verbose = TRUE)

## S3 method for class 'list'
lsm_l_rpr(landscape, classes_max = NULL, verbose = TRUE)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
classes_max	Potential maximum number of present classes
verbose	Print warning message if not sufficient patches are present

### Details

$$RPR = \frac{m}{m_{max}} * 100$$

where  $m$  is the number of classes and  $m_{max}$  is the (theoretical) maximum number of classes.

RPR is an 'Diversity metric'. The metric calculates the percentage of present classes in the landscape in relation to a (theoretical) number of maximum classes. The user has to specify the maximum number of classes. Note, that if `classes_max` is not provided, the functions returns NA.

**Units:** Percentage

**Ranges:**  $0 < \text{RPR} \leq 100$

**Behaviour:** Approaches  $\text{RPR} > 0$  when only one class type is present, but the maximum number of classes is large. Equals  $\text{RPR} = 100$  when  $m = m_{\text{max}}$

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Romme, W. H. 1982. Fire and landscapediversity in subalpine forests of Yellowstone National Park. *Ecol.Monogr.* 52:199-221

### Examples

```
lsm_l_rpr(landscape, classes_max = 5)
```

---

lsm_l_shape_cv	<i>SHAPE_CV (landscape level)</i>
----------------	-----------------------------------

---

### Description

Covariance of variation shape index (Shape metric)

### Usage

```
lsm_l_shape_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_shape_cv(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$SHAPE_{CV} = cv(SHAPE[patch_{ij}])$$

where  $SHAPE[patch_{ij}]$  is the shape index of each patch.

SHAPE\_CV is a 'Shape metric'. The landscape is summarised as the Coefficient of variation of all patches in the landscape. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE\_CV  $\geq 0$

**Behaviour:** Equals SHAPE\_CV = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

**See Also**

[lsm\\_p\\_shape](#), [cv](#),  
[lsm\\_c\\_shape\\_mn](#), [lsm\\_c\\_shape\\_sd](#),  
[lsm\\_c\\_shape\\_cv](#), [lsm\\_l\\_shape\\_mn](#), [lsm\\_l\\_shape\\_sd](#)

**Examples**

```
lsm_l_shape_cv(landscape)
```



---

lsm_l_shape_mn	<i>SHAPE_MN (landscape level)</i>
----------------	-----------------------------------

---

### Description

Mean shape index (Shape metric)

### Usage

```
lsm_l_shape_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_shape_mn(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$SHAPE_{MN} = mean(SHAPE[patch_{ij}])$$

where  $SHAPE[patch_{ij}]$  is the shape index of each patch.

SHAPE\_MN is a 'Shape metric'. The landscape is summarised as the mean of all patches in the landscape. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE\_SD >= 1

**Behaviour:** Equals SHAPE\_MN = 0 if all patches are squares. Increases, without limit, as the shapes of patches become more complex.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

**See Also**

[lsm\\_p\\_shape](#), [mean](#),  
[lsm\\_c\\_shape\\_mn](#), [lsm\\_c\\_shape\\_sd](#), [lsm\\_c\\_shape\\_cv](#),  
[lsm\\_l\\_shape\\_sd](#), [lsm\\_l\\_shape\\_cv](#)

**Examples**

```
lsm_l_shape_mn(landscape)
```

---

lsm_l_shape_sd	<i>SHAPE_SD (landscape level)</i>
----------------	-----------------------------------

---

**Description**

Standard deviation shape index (Shape metric)

**Usage**

```
lsm_l_shape_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_shape_sd(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$SHAPE_{SD} = sd(SHAPE[patch_{ij}])$$

where  $SHAPE[patch_{ij}]$  is the shape index of each patch.

SHAPE\_SD is a 'Shape metric'. The landscape summarised as the standard deviation of all patches in the landscape. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE\_SD >= 0

**Behaviour:** Equals SHAPE\_SD = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

**See Also**

[lsm\\_p\\_shape](#), [sd](#),  
[lsm\\_c\\_shape\\_mn](#), [lsm\\_c\\_shape\\_sd](#), [lsm\\_c\\_shape\\_cv](#),  
[lsm\\_l\\_shape\\_mn](#), [lsm\\_l\\_shape\\_cv](#)

**Examples**

```
lsm_l_shape_sd(landscape)
```

---

lsm_l_shdi	<i>SHDI (landscape level)</i>
------------	-------------------------------

---

### Description

Shannon's diversity index (Diversity metric)

### Usage

```
lsm_l_shdi(landscape)

## S3 method for class 'RasterLayer'
lsm_l_shdi(landscape)

## S3 method for class 'RasterStack'
lsm_l_shdi(landscape)

## S3 method for class 'RasterBrick'
lsm_l_shdi(landscape)

## S3 method for class 'stars'
lsm_l_shdi(landscape)

## S3 method for class 'list'
lsm_l_shdi(landscape)
```

### Arguments

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.

### Details

$$SHDI = - \sum_{i=1}^m (P_i * \ln P_i)$$

where  $P_i$  is the proportion of class i.

SHDI is a 'Diversity metric'. It is a widely used metric in biodiversity and ecology and takes both the number of classes and the abundance of each class into account.

**Units:** None

**Range:** SHDI >= 0

**Behaviour:** Equals SHDI = 0 when only one patch is present and increases, without limit, as the number of classes increases while the proportions are equally distributed

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Shannon, C., and W. Weaver. 1949. The mathematical theory of communication. Univ. Illinois-Press, Urbana

**See Also**

[lsm\\_c\\_pland](#)

**Examples**

```
lsm_l_shdi(landscape)
```

---

lsm_l_shei	<i>SHEI (landscape level)</i>
------------	-------------------------------

---

**Description**

Shannons's evenness index (Diversity metric)

**Usage**

```
lsm_l_shei(landscape)

## S3 method for class 'RasterLayer'
lsm_l_shei(landscape)

## S3 method for class 'RasterStack'
lsm_l_shei(landscape)

## S3 method for class 'RasterBrick'
lsm_l_shei(landscape)

## S3 method for class 'stars'
lsm_l_shei(landscape)

## S3 method for class 'list'
lsm_l_shei(landscape)
```

**Arguments**

landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.

**Details**

$$SHEI = \frac{- \sum_{i=1}^m (P_i * \ln P_i)}{\ln m}$$

where  $P_i$  is the proportion of class  $i$  and  $m$  is the number of classes.

SHEI is a 'Diversity metric'. It is the ratio between the actual Shannon's diversity index and the theoretical maximum of the Shannon diversity index. It can be understood as a measure of dominance.

**Units:** None

**Range:**  $0 \leq SHEI < 1$

**Behaviour:** Equals  $SHEI = 0$  when only one patch present and equals  $SHEI = 1$  when the proportion of classes is completely equally distributed

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Shannon, C., and W. Weaver. 1949. The mathematical theory of communication. Univ. Illinois-Press, Urbana

**See Also**

[lsm\\_c\\_pland](#), [lsm\\_l\\_pr](#)

**Examples**

```
lsm_l_shei(landscape)
```

---

lsm_l_sidi	<i>SIDI (landscape level)</i>
------------	-------------------------------

---

### Description

Simpson's diversity index (Diversity metric)

### Usage

```
lsm_l_sidi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_sidi(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$SIDI = 1 - \sum_{i=1}^m P_i^2$$

where  $P_i$  is the proportion of class  $i$  and  $m$  is the number of classes.

SIDI is a 'Diversity metric'. It is widely used in biodiversity and ecology. It is less sensitive to rare class types than [lsm\\_l\\_shdi](#). It can be interpreted as the probability that two randomly selected cells belong to the same class.

**Units:** None

**Range:**  $0 \leq SIDI < 1$

**Behaviour:** Equals  $SIDI = 0$  when only one patch is present and approaches  $SIDI < 1$  when the number of class types increases while the proportions are equally distributed

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Simpson, E. H. 1949. Measurement of diversity. Nature 163:688

**See Also**

[lsm\\_c\\_pland](#), [lsm\\_l\\_pr](#)

**Examples**

```
lsm_l_sidi(landscape)
```

---

lsm_l_siei	<i>SIEI (landscape level)</i>
------------	-------------------------------

---

**Description**

Simpson’s evenness index (Diversity metric)

**Usage**

```
lsm_l_siei(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_siei(landscape, directions = 8)
```



**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$SIEI = \frac{1 - \sum_{i=1}^m P_i^2}{1 - \frac{1}{m}}$$

where  $P_i$  is the proportion of class  $i$  and  $m$  is the number of classes.

SIEI is a 'Diversity metric'. The metric is widely used in biodiversity and ecology. It is the ratio between the actual Simpson's diversity index and the theoretical maximum Simpson's diversity index.

**Units:** None

**Range:**  $0 < SIEI \leq 1$

**Behaviour:** Equals  $SIEI = 0$  when only one patch is present and approaches  $SIEI = 1$  when the number of class types increases while the proportions are equally distributed

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Simpson, E. H. 1949. Measurement of diversity. *Nature* 163:688

**See Also**

[lsm\\_c\\_pland](#), [lsm\\_l\\_pr](#)

**Examples**

```
lsm_l_siei(landscape)
```

---

lsm_l_split	<i>SPLIT (landscape level)</i>
-------------	--------------------------------

---

### Description

Splitting index (Aggregation metric)

### Usage

```
lsm_l_split(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_split(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$SSPLIT = \frac{A^2}{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}$$

where  $a_{ij}$  is the patch area in square meters and  $A$  is the total landscape area.

SPLIT is an 'Aggregation metric'. It describes the number of patches if all patches the landscape would be divided into equally sized patches.

**Units:** None

**Range:**  $1 \leq SPLIT \leq \text{Number of cells squared}$

**Behaviour:** Equals  $SPLIT = 1$  if only one patch is present. Increases as the number of patches increases and is limited if all cells are a patch

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. *Landscape ecology*, 15(2), 115-130.

**See Also**

[lsm\\_p\\_area](#), [lsm\\_l\\_ta](#),  
[lsm\\_c\\_split](#)

**Examples**

```
lsm_l_split(landscape)
```

---

lsm_l_ta	<i>TA (landscape level)</i>
----------	-----------------------------

---

**Description**

Total area (Area and edge metric)

**Usage**

```
lsm_l_ta(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_ta(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$CA = \text{sum}(AREA[patch_{ij}])$$

where  $AREA[patch_{ij}]$  is the area of each patch in hectares.

TA is an 'Area and edge metric'. The total (class) area sums the area of all patches in the landscape. It is the area of the observation area.

**Units:** Hectares

**Range:**  $TA > 0$

**Behaviour:** Approaches  $TA > 0$  if the landscape is small and increases, without limit, as the size of the landscape increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_area](#), [sum](#),  
[lsm\\_c\\_ca](#)

**Examples**

```
lsm_l_ta(landscape)
```

---

lsm_l_tca	<i>TCA (landscape level)</i>
-----------	------------------------------

---

### Description

Total core area (Core area metric)

### Usage

```
lsm_l_tca(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_tca(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$TCA = \sum_{j=1}^n a_{ij}^{core} * (\frac{1}{10000})$$

where here  $a_{ij}^{core}$  is the core area in square meters.

TCA is a 'Core area metric' and equals the sum of core areas of all patches in the landscape. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). In other words, the core area of a patch is all area that is not an edge. It characterises patch areas and shapes of all patches in the landscape simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square). Additionally, TCA is a measure for the configuration of the landscape, because the sum of edges increase as patches are less aggregated.

**Units:** Hectares

**Range:** TCA >= 0

**Behaviour:** Increases, without limit, as patch areas increase and patch shapes simplify. TCA = 0 when every cell in every patch is an edge.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_p\\_core](#), [lsm\\_c\\_tca](#)

**Examples**

```
lsm_l_tca(landscape)
```

---

lsm_l_te	<i>TE (landscape level)</i>
----------	-----------------------------

---

**Description**

Total edge (Area and Edge metric)

**Usage**

```
lsm_l_te(landscape, count_boundary)

## S3 method for class 'RasterLayer'
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'RasterStack'
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'RasterBrick'
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'stars'
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'list'
lsm_l_te(landscape, count_boundary = FALSE)
```

**Arguments**

**landscape** Raster\* Layer, Stack, Brick or a list of rasterLayers.  
**count\_boundary** Include landscape boundary in edge length

**Details**

$$TE = \sum_{k=1}^m e_{ik}$$

where  $e_{ik}$  is the edge lengths in meters. TE is an 'Area and edge metric'. Total edge includes all edges. It measures the configuration of the landscape because a highly fragmented landscape will have many edges. However, total edge is an absolute measure, making comparisons among landscapes with different total areas difficult. If `count_boundary = TRUE` also edges to the landscape boundary are included.

**Units:** Meters

**Range:**  $TE \geq 0$

**Behaviour:** Equals  $TE = 0$  if all cells are edge cells. Increases, without limit, as landscape becomes more fragmented

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

See Also

[lsm\\_p\\_perim](#) [lsm\\_l\\_te](#)

Examples

```
lsm_l_te(landscape)
```

---

lsm_p_area	<i>AREA (patch level)</i>
------------	---------------------------

---

Description

Patch area (Area and edge metric)

Usage

```
lsm_p_area(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_area(landscape, directions = 8)
```

Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

Details

$$AREA = a_{ij} * (\frac{1}{10000})$$

where  $a_{ij}$  is the area in square meters.



AREA is an 'Area and edge metric' and equals the area of each patch in hectares. The lower limit of AREA is limited by the resolution of the input raster, i.e. AREA can't be smaller than the resolution squared (in hectares). It is one of the most basic, but also most important metrics, to characterise a landscape. The metric is the simplest measure of composition.

- Units:** Hectares
- Range:** AREA > 0
- Behaviour:** Increases, without limit, as the patch size increases.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_area\\_mn](#), [lsm\\_c\\_area\\_sd](#), [lsm\\_c\\_area\\_cv](#), [lsm\\_c\\_ca](#),  
[lsm\\_l\\_area\\_mn](#), [lsm\\_l\\_area\\_sd](#), [lsm\\_l\\_area\\_cv](#), [lsm\\_l\\_ta](#)

**Examples**

```
lsm_p_area(landscape)
```

---

lsm_p_cai	<i>CAI (patch level)</i>
-----------	--------------------------

---

**Description**

Core area index (Core area metric)

**Usage**

```
lsm_p_cai(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_p_cai(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_p_cai(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
lsm_p_cai(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_p_cai(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_p_cai(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

### Details

$$CAI = \left( \frac{a_{ij}^{core}}{a_{ij}} \right) * 100$$

where  $a_{ij}^{core}$  is the core area in square meters and  $a_{ij}$  is the area in square meters.

CAI is a 'Core area metric'. It equals the percentage of a patch that is core area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). It describes patch area and shape simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square). Because the index is relative, it is comparable among patches with different area.

**Units:** Percent

**Range:**  $0 \leq CAI \leq 100$

**Behaviour:** CAI = 0 when the patch has no core area and approaches CAI = 100 with increasing percentage of core area within a patch.

### Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## See Also

[lsm\\_p\\_core](#), [lsm\\_p\\_area](#),  
[lsm\\_c\\_cai\\_mn](#), [lsm\\_c\\_cai\\_sd](#), [lsm\\_c\\_cai\\_cv](#), [lsm\\_c\\_cpland](#),  
[lsm\\_l\\_cai\\_mn](#), [lsm\\_l\\_cai\\_sd](#), [lsm\\_l\\_cai\\_cv](#)

## Examples

```
lsm_p_cai(landscape)
```

---

lsm_p_circle	<i>CIRCLE (patch level)</i>
--------------	-----------------------------

---

## Description

Related Circumscribing Circle (Shape metric)

## Usage

```
lsm_p_circle(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_circle(landscape, directions = 8)
```

## Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$CIRCLE = 1 - \left( \frac{a_{ij}}{a_{ij}^{circle}} \right)$$

where  $a_{ij}$  is the area in square meters and  $a_{ij}^{circle}$  the area of the smallest circumscribing circle.

CIRCLE is a 'Shape metric'. The metric is the ratio between the patch area and the smallest circumscribing circle of the patch. The diameter of the smallest circumscribing circle is the 'diameter' of the patch connecting the opposing corner points of the two cells that are the furthest away from each other. The metric characterises the compactness of the patch and is comparable among patches with different area.

**Units:** None

**Range:**  $0 \leq CIRCLE < 1$

**Behaviour:** CIRCLE = 0 for a circular patch and approaches CIRCLE = 1 for a linear patch.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landscape Ecology* 7: 291-302.

## See Also

[lsm\\_p\\_area](#),  
[lsm\\_c\\_circle\\_mn](#), [lsm\\_c\\_circle\\_sd](#), [lsm\\_c\\_circle\\_cv](#),  
[lsm\\_l\\_circle\\_mn](#), [lsm\\_l\\_circle\\_sd](#), [lsm\\_l\\_circle\\_cv](#)

## Examples

```
lsm_p_circle(landscape)
```

---

lsm_p_contig	<i>CONTIG (patch level)</i>
--------------	-----------------------------

---

### Description

Contiguity index (Shape metric)

### Usage

```
lsm_p_contig(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_contig(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$CONTIG = \frac{\left[ \frac{\sum_{r=1}^z c_{ijr}}{a_{ij}} \right] - 1}{v - 1}$$

where  $c_{ijr}$  is the contiguity value for pixel  $r$  in patch  $ij$ ,  $a_{ij}$  the area of the respective patch (number of cells) and  $v$  is the size of the filter matrix (13 in this case).

CONTIG is a 'Shape metric'. It assesses the spatial connectedness (contiguity) of cells in patches. CONTIG coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** 0 >= CONTIG <= 1

**Behaviour:** Equals 0 for one-pixel patches and increases to a limit of 1 (fully connected patch).

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

**See Also**

[lsm\\_c\\_contig\\_mn](#), [lsm\\_c\\_contig\\_sd](#), [lsm\\_c\\_contig\\_cv](#),  
[lsm\\_l\\_contig\\_mn](#), [lsm\\_l\\_contig\\_sd](#), [lsm\\_l\\_contig\\_cv](#)

**Examples**

```
lsm_p_contig(landscape)
```

---

lsm_p_core	<i>CORE (patch level)</i>
------------	---------------------------

---

**Description**

Core area (Core area metric)

**Usage**

```

lsm_p_core(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_p_core(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_p_core(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_p_core(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_p_core(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_p_core(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

$$CORE = a_{ij}^{core}$$

where  $a_{ij}^{core}$  is the core area in square meters

CORE is a 'Core area metric' and equals the area within a patch that is not on the edge of it. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). It describes patch area and shape simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square).

**Units:** Hectares

**Range:** CORE  $\geq 0$

**Behaviour:** Increases, without limit, as the patch area increases and the patch shape simplifies (more core area). CORE = 0 when every cell in the patch is an edge.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_core\\_mn](#), [lsm\\_c\\_core\\_sd](#), [lsm\\_c\\_core\\_cv](#), [lsm\\_c\\_tca](#),  
[lsm\\_l\\_core\\_mn](#), [lsm\\_l\\_core\\_sd](#), [lsm\\_l\\_core\\_cv](#), [lsm\\_l\\_tca](#)

**Examples**

```
lsm_p_core(landscape)
```

---

lsm_p_enn	<i>ENN (patch level)</i>
-----------	--------------------------

---

**Description**

Euclidean Nearest-Neighbor Distance (Aggregation metric)

**Usage**

```
lsm_p_enn(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_p_enn(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_p_enn(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_p_enn(landscape, directions = 8,
  verbose = TRUE)

## S3 method for class 'stars'
lsm_p_enn(landscape, directions = 8, verbose = TRUE)
```



```
## S3 method for class 'list'
lsm_p_enn(landscape, directions = 8, verbose = TRUE)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
verbose	Print warning message if not sufficient patches are present

### Details

$$ENN = h_{ij}$$

where  $h_{ij}$  is the distance to the nearest neighbouring patch of the same class  $i$  in meters

ENN is an 'Aggregation metric'. The distance to the nearest neighbouring patch of the same class  $i$ . The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation.

**Units:** Meters

**Range:**  $ENN > 0$

**Behaviour:** Approaches  $ENN = 0$  as the distance to the nearest neighbour decreases, i.e. patches of the same class  $i$  are more aggregated. Increases, without limit, as the distance between neighbouring patches of the same class  $i$  increases, i.e. patches are more isolated.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

### See Also

[lsm\\_c\\_enn\\_mn](#), [lsm\\_c\\_enn\\_sd](#), [lsm\\_c\\_enn\\_cv](#),  
[lsm\\_l\\_enn\\_mn](#), [lsm\\_l\\_enn\\_sd](#), [lsm\\_l\\_enn\\_cv](#)

### Examples

```
lsm_p_enn(landscape)
```

---

lsm_p_frac	<i>FRAC (patch level)</i>
------------	---------------------------

---

### Description

Fractal dimension index (Shape metric)

### Usage

```
lsm_p_frac(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_frac(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$FRAC = \frac{2 * \ln *(0.25 * p_{ij})}{\ln a_{ij}}$$

where  $p_{ij}$  is the perimeter in meters and  $a_{ij}$  is the area in square meters

FRAC is a 'Shape metric'. The index is based on the patch perimeter and the patch area and describes the patch complexity. Because it is standardized, it is scale independent, meaning that increasing the patch size while not changing the patch form will not change the ratio.

**Units:** None

**Range:**  $1 \leq FRAC \leq 2$

**Behaviour:** Approaches  $FRAC = 1$  for a squared patch shape form and  $FRAC = 2$  for a irregular patch shape.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

**See Also**

[lsm\\_p\\_area](#), [lsm\\_p\\_perim](#),  
[lsm\\_c\\_frac\\_mn](#), [lsm\\_c\\_frac\\_sd](#), [lsm\\_c\\_frac\\_cv](#),  
[lsm\\_l\\_frac\\_mn](#), [lsm\\_l\\_frac\\_sd](#), [lsm\\_l\\_frac\\_cv](#)

**Examples**

```
lsm_p_frac(landscape)
```

---

lsm_p_gyrate	<i>GYRATE (patch level)</i>
--------------	-----------------------------

---

**Description**

Radius of Gyration (Area and edge metric)

**Usage**

```
lsm_p_gyrate(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_gyrate(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_gyrate(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_gyrate(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_gyrate(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_gyrate(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$GYRATE = \sum_{r=1}^z \frac{h_{ijr}}{z}$$

where  $h_{ijr}$  is the distance from each cell to the centroid of the patch and  $z$  is the number of cells.

GYRATE is an 'Area and edge metric'. The distance from each cell to the patch centroid is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness.

**Units:** Meters

**Range:** GYRATE  $\geq 0$

**Behaviour:** Approaches GYRATE = 0 if patch is a single cell. Increases, without limit, when only one patch is present.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

**See Also**

[lsm\\_c\\_gyrate\\_mn](#), [lsm\\_c\\_gyrate\\_sd](#), [lsm\\_c\\_gyrate\\_cv](#),  
[lsm\\_l\\_gyrate\\_mn](#), [lsm\\_l\\_gyrate\\_sd](#), [lsm\\_l\\_gyrate\\_cv](#)

**Examples**

```
lsm_p_gyrate(landscape)
```

---

lsm_p_ncore	<i>NCORE (patch level)</i>
-------------	----------------------------

---

### Description

Number of core areas (Core area metric)

### Usage

```
lsm_p_ncore(landscape, directions, consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
lsm_p_ncore(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
lsm_p_ncore(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
lsm_p_ncore(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
lsm_p_ncore(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
lsm_p_ncore(landscape, directions = 8,
  consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell
	#' @details

$$NCORE = n_{ij}^{core}$$

where  $n_{ij}^{core}$  is the number of disjunct core areas.

NCORE is a 'Core area metric'. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). The metric counts the

disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. It describes patch area and shape simultaneously (more core area when the patch is large, however, the shape must allow disjunct core areas). Thereby, a compact shape (e.g. a square) will contain less disjunct core areas than a more irregular patch.

**Units:** None

**Range:** NCORE >= 0

**Behaviour:** NCORE = 0 when CORE = 0, i.e. every cell in patch is edge. Increases, without limit, as core area increases and patch shape allows disjunct core areas (i.e. patch shape becomes rather complex).

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

**See Also**

[lsm\\_c\\_dcore\\_mn](#), [lsm\\_c\\_dcore\\_sd](#), [lsm\\_c\\_dcore\\_cv](#), [lsm\\_c\\_ndca](#),  
[lsm\\_l\\_dcore\\_mn](#), [lsm\\_l\\_dcore\\_sd](#), [lsm\\_l\\_dcore\\_cv](#), [lsm\\_l\\_ndca](#)

**Examples**

```
lsm_p_ncore(landscape)
```

---

lsm_p_para	<i>PARAM (patch level)</i>
------------	----------------------------

---

**Description**

Perimeter-Area ratio (Shape metric)

**Usage**

```
lsm_p_para(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_para(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_para(landscape, directions = 8)
```

```
## S3 method for class 'RasterBrick'
lsm_p_para(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_para(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_para(landscape, directions = 8)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$PARA = \frac{p_{ij}}{a_{ij}}$$

where  $p_{ij}$  is the perimeter in meters and  $a_{ij}$  is the area in square meters.

PARA is a 'Shape metric'. It describes the patch complexity in a straightforward way. However, because it is not standardised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:**  $PARA > 0$

**Behaviour:** Increases, without limit, as the shape complexity increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

### See Also

[lsm\\_p\\_area](#), [lsm\\_p\\_perim](#),  
[lsm\\_c\\_para\\_mn](#), [lsm\\_c\\_para\\_sd](#), [lsm\\_c\\_para\\_cv](#),  
[lsm\\_l\\_para\\_mn](#), [lsm\\_l\\_para\\_sd](#), [lsm\\_l\\_para\\_cv](#)

Examples

```
lsm_p_para(landscape)
```

---

lsm_p_perim	<i>PERIM (patch level)</i>
-------------	----------------------------

---

Description

Perimeter (Area and edge metric))

Usage

```
lsm_p_perim(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_perim(landscape, directions = 8)
```

Arguments

- landscape      Raster\* Layer, Stack, Brick or a list of rasterLayers.
- directions      The number of directions in which patches should be connected: 4 (rook’s case) or 8 (queen’s case).

Details

$$PERIM = p_{ij}$$

where  $p_{ij}$  is the perimeter in meters.

PERIM is an 'Area and edge metric'. It equals the perimeter of the patch including also the edge to the landscape boundary. The metric describes patch area (larger perimeter for larger patches), but also patch shape (large perimeter for irregular shapes).

**Units:** Meters



**Range:** PERIM > 0

**Behaviour:** Increases, without limit, as patch size and complexity increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

## Examples

```
lsm_p_perim(landscape)
```

---

lsm_p_shape	<i>SHAPE (patch level)</i>
-------------	----------------------------

---

## Description

Shape index (Shape metric)

## Usage

```
lsm_p_shape(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_shape(landscape, directions = 8)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

**Details**

$$SHAPE = \frac{p_{ij}}{\min p_{ij}}$$

where  $p_{ij}$  is the perimeter in terms of cell surfaces and  $\min p_{ij}$  is the minimum perimeter of the patch in terms of cell surfaces.

SHAPE is a 'Shape metric'. It describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE  $\geq 1$

**Behaviour:** Equals SHAPE = 1 for a squared patch and increases, without limit, as the patch shape becomes more complex.

**Value**

tibble

**References**

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: <http://www.umass.edu/landeco/research/fragstats/>

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

**See Also**

[lsm\\_p\\_perim](#), [lsm\\_p\\_area](#),  
[lsm\\_c\\_shape\\_mn](#), [lsm\\_c\\_shape\\_sd](#), [lsm\\_c\\_shape\\_cv](#),  
[lsm\\_l\\_shape\\_mn](#), [lsm\\_l\\_shape\\_sd](#), [lsm\\_l\\_shape\\_cv](#)

**Examples**

```
lsm_p_shape(landscape)
```

---

options_landscapemetrics	
	<i>options_landscapemetrics</i>

---

**Description**

Sets global options for landscapemetrics

**Usage**

options\_landscapemetrics(to\_disk = NULL)

**Arguments**

to_disk	Logical argument, if FALSE results of get_patches are hold in memory. If true, get_patches writes temporary files and hence, does not hold everything in memory. Can be set with a global option, e.g. options(to_disk = TRUE). See Details.
---------	--

**Details**

Landscape metrics rely on the delineation of patches. Hence, get\_patches is heavily used in **landscapemetrics**. As raster can be quite big, the fact that get\_patches creates a copy of the raster for each class in a landscape becomes a burden for computer memory. Hence, the argument *to\_disk* allows to store the results of the connected labeling algorithm on disk. Furthermore, this option can be set globally, so that every function that internally uses get\_patches can make use of that.

**Value**

Global option to be used internally in the package

---

podlasie_ccilc	<i>Podlasie ESA CCI LC</i>
----------------	----------------------------

---

**Description**

A real landscape of the Podlasie region in Poland from the ESA CCI Land Cover

**Usage**

podlasie\_ccilc

**Format**

A raster layer object.

**Source**

<http://maps.elie.ucl.ac.be/CCI/viewer/>

---

sample\_lsm

*sample\_lsm*

---

**Description**

Sample metrics

**Usage**

```
sample_lsm(landscape, y, plot_id, shape, size, return_raster, verbose,
           progress, ...)
```

```
## S3 method for class 'RasterLayer'
sample_lsm(landscape, y, plot_id = NULL,
           shape = "square", size, return_raster = FALSE, verbose = TRUE,
           progress = FALSE, ...)
```

```
## S3 method for class 'RasterStack'
sample_lsm(landscape, y, plot_id = NULL,
           shape = "square", size, return_raster = FALSE, verbose = TRUE,
           progress = FALSE, ...)
```

```
## S3 method for class 'RasterBrick'
sample_lsm(landscape, y, plot_id = NULL,
           shape = "square", size, return_raster = FALSE, verbose = TRUE,
           progress = FALSE, ...)
```

```
## S3 method for class 'stars'
sample_lsm(landscape, y, plot_id = NULL,
           shape = "square", size, return_raster = FALSE, verbose = TRUE,
           progress = FALSE, ...)
```

```
## S3 method for class 'list'
sample_lsm(landscape, y, plot_id = NULL,
           shape = "square", size, return_raster = FALSE, verbose = TRUE,
           progress = FALSE, ...)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
y	2-column matrix with coordinates, SpatialPoints, SpatialLines, SpatialPolygons of sf points.

plot_id	Vector with id of sample points. If not provided, sample points will be labelled 1...n.
shape	String specifying plot shape. Either "circle" or "square"
size	Approximated size of sample plot. Equals the radius for circles or half of the side-length for squares in mapunits. For lines size equals the width of the buffer.
return_raster	Logical if the clipped raster of the sample plot should be returned
verbose	Print warning messages.
progress	Print progress report.
...	Arguments passed on to calculate_lsm().

## Details

This function samples the selected metrics in a buffer area (sample plot) around sample points, sample lines or within provided SpatialPolygons. The size of the actual sampled landscape can be different to the provided size due to two reasons. Firstly, because clipping raster cells using a circle or a sample plot not directly at a cell center lead to inaccuracies. Secondly, sample plots can exceed the landscape boundary. Therefore, we report the actual clipped sample plot area relative in relation to the theoretical, maximum sample plot area e.g. a sample plot only half within the landscape will have a `percentage_inside = 50`. Please be aware that the output is slightly different to all other lsm-function of landscapemetrics.

The metrics can be specified by the arguments `what`, `level`, `metric`, `name` and/or `type` (combinations of different arguments are possible (e.g. `level = "class"`, `type = "aggregation metric"`). If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, don't specify any of the above arguments.

## Value

tibble

## See Also

[list\\_lsm](#)  
[calculate\\_lsm](#)

## Examples

```
# use a matrix
sample_points <- matrix(c(10, 5, 25, 15, 5, 25), ncol = 2, byrow = TRUE)
sample_lsm(landscape, y = sample_points, size = 15, what = "lsm_l_np")

# use sp points
points_sp <- sp::SpatialPoints(sample_points)
sample_lsm(landscape, y = points_sp, size = 15, what = "lsm_l_np", return_raster = TRUE)

## Not run:
# use lines (works only if rgeos is installed)
x1 <- c(1, 5, 15, 10)
y1 <- c(1, 5, 15, 25)
```

```

x2 <- c(10, 25)
y2 <- c(5, 5)

sample_lines <- sp::SpatialLines(list(sp::Lines(list(sp::Line(cbind(x1, y1)),
sp::Line(cbind(x2, y2))), ID = "a")))
sample_lsm(landscape, y = sample_lines, size = 10, what = "lsm_l_np")

# use polygons
poly_1 <- sp::Polygon(cbind(c(2.5, 2.5, 17.5, 17.5),
                             c(-2.5, 12.5, 12.5, -2.5)))
poly_2 <- sp::Polygon(cbind(c(7.5, 7.5, 23.5, 23.5),
                             c(-7.5, 23.5, 23.5, -7.5)))
poly_1 <- sp::Polygons(list(poly_1), "p1")
poly_2 <- sp::Polygons(list(poly_2), "p2")
sample_plots <- sp::SpatialPolygons(list(poly_1, poly_2))

sample_lsm(landscape, y = sample_plots, what = "lsm_l_np", size = 5)

## End(Not run)

```

---

scale\_sample

*scale\_sample*


---

## Description

Metrics on changing sample scale

## Usage

```
scale_sample(landscape, y, shape, size, max_size, verbose, progress, ...)
```

```
## S3 method for class 'RasterLayer'
```

```
scale_sample(landscape, y, shape = "square", size,
             max_size, verbose = TRUE, progress = FALSE, ...)
```

```
## S3 method for class 'RasterStack'
```

```
scale_sample(landscape, y, shape = "square", size,
             max_size, verbose = TRUE, progress = FALSE, ...)
```

```
## S3 method for class 'RasterBrick'
```

```
scale_sample(landscape, y, shape = "square", size,
             max_size, verbose = TRUE, progress = FALSE, ...)
```

```
## S3 method for class 'stars'
```

```
scale_sample(landscape, y, shape = "square", size,
             max_size, verbose = TRUE, progress = FALSE, ...)
```

```
## S3 method for class 'list'
scale_sample(landscape, y, shape = "square", size,
             max_size, verbose = TRUE, progress = FALSE, ...)
```

### Arguments

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
y	2-column matrix with coordinates or SpatialPoints.
shape	String specifying plot shape. Either "circle" or "square"
size	Approximated size of sample plot. Equals the radius for circles or half of the side-length for squares in mapunits. For lines size equals the width of the buffer.
max_size	Maximum size to which sample plot size is summed up.
verbose	Print warning messages.
progress	Print progress report.
...	Arguments passed on to calculate_lsm().

### Details

This function calculates the selected metrics in subsequential buffers around a/multiple point(s) of interest.

The size of the actual sampled landscape can be different to the provided size due to two reasons. Firstly, because clipping raster cells using a circle or a sample plot not directly at a cell center lead to inaccuracies. Secondly, sample plots can exceed the landscape boundary. Therefore, we report the actual clipped sample plot area relative in relation to the theoretical, maximum sample plot area e.g. a sample plot only half within the landscape will have a `percentage_inside = 50`. Please be aware that the output is slightly different to all other `lsm`-function of `landscapemetrics`.

The metrics can be specified by the arguments `what`, `level`, `metric`, `name` and/or `type` (combinations of different arguments are possible (e.g. `level = "class"`, `type = "aggregation metric"`). If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, don't specify any of the above arguments.

### Value

tibble

### See Also

[list\\_lsm](#)  
[calculate\\_lsm](#)  
[sample\\_lsm](#)  
[construct\\_buffer](#)

**Examples**

```
my_points = matrix(c(1265000, 1250000, 1255000, 1257000),
                   ncol = 2, byrow = TRUE)

scale_sample(landscape = augusta_nlcd, y = my_points,
             size = 500, max_size = 5000, what = c("lsm_l_ent", "lsm_l_mutinf"))
```

---

scale_window	<i>scale_window</i>
--------------	---------------------

---

**Description**

Metrics on changing sample scale

**Usage**

```
scale_window(landscape, percentages_col, percentages_row, what, stat,
             verbose, progress, ...)

## S3 method for class 'RasterLayer'
scale_window(landscape, percentages_col = c(2, 4,
      8, 16, 32, 64, 100), percentages_row = NULL, what, stat,
      verbose = TRUE, progress = FALSE, ...)

## S3 method for class 'RasterStack'
scale_window(landscape, percentages_col = c(2, 4,
      8, 16, 32, 64, 100), percentages_row = NULL, what, stat,
      verbose = TRUE, progress = FALSE, ...)

## S3 method for class 'RasterBrick'
scale_window(landscape, percentages_col = NULL,
      percentages_row = NULL, what, stat, verbose = TRUE,
      progress = FALSE, ...)

## S3 method for class 'stars'
scale_window(landscape, percentages_col = NULL,
      percentages_row = NULL, what, stat, verbose = TRUE,
      progress = FALSE, ...)

## S3 method for class 'list'
scale_window(landscape, percentages_col = NULL,
      percentages_row = NULL, what, stat, verbose = TRUE,
      progress = FALSE, ...)
```



**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
percentages_col	2-column matrix with coordinates or SpatialPoints.
percentages_row	String specifying plot shape. Either "circle" or "square"
what	Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. <code>what = c("lsm_l_mutinf", "lsm_l_ta")</code> .
stat	The function to be applied. See Details
verbose	If TRUE, warnings are printed.
progress	Print progress report.
...	Arguments passed on to <code>calculate_lsm()</code> .

**Details**

This function calculates the selected metrics in moving windows over the provided landscape.

Please be aware that the output is slightly different to all other `lsm`-function of `landscapemetrics`.

The metrics can be specified by the arguments `what`, `level`, `metric`, `name` and/or `type` (combinations of different arguments are possible (e.g. `level = "class"`, `type = "aggregation metric"`). If an argument is not provided, automatically all possibilities are selected. Only metrics on landscape level are supported for this function.

**Value**

tibble

**See Also**

[list\\_lsm](#)  
[window\\_lsm](#)  
[scale\\_sample](#)

**Examples**

```
## Not run:
percentages_col <- c(2, 4, 8, 16, 32, 64, 100)
percentages_row <- c(2, 4, 8, 16, 32, 64, 100)

what = c("lsm_l_pr", "lsm_l_joinent")

stat <- "mean"

scale_window(landscape, percentages_col, percentages_row, what, stat)

## End(Not run)
```

show\_cores

*Show core area***Description**

Show core area

**Usage**

```
show_cores(landscape, directions, class, labels, nrow, ncol,
           consider_boundary, edge_depth)
```

```
## S3 method for class 'RasterLayer'
show_cores(landscape, directions = 8,
           class = "all", labels = TRUE, nrow = NULL, ncol = NULL,
           consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterStack'
show_cores(landscape, directions = 8,
           class = "all", labels = TRUE, nrow = NULL, ncol = NULL,
           consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'RasterBrick'
show_cores(landscape, directions = 8,
           class = "all", labels = TRUE, nrow = NULL, ncol = NULL,
           consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'stars'
show_cores(landscape, directions = 8, class = "all",
           labels = TRUE, nrow = NULL, ncol = NULL,
           consider_boundary = FALSE, edge_depth = 1)
```

```
## S3 method for class 'list'
show_cores(landscape, directions = 8, class = "all",
           labels = TRUE, nrow = NULL, ncol = NULL,
           consider_boundary = FALSE, edge_depth = 1)
```

**Arguments**

landscape	Raster object
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
class	How to show the core area: "global" (single map), "all" (every class as facet), or a vector with the specific classes one wants to show (every selected class as facet).
labels	Logical flag indicating whether to print or not to print core labels. boundary should be considered as core

<code>nrow, ncol</code>	Number of rows and columns for the facet.
<code>consider_boundary</code>	Logical if cells that only neighbour the landscape boundary should be considered as core.
<code>edge_depth</code>	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell

**Details**

The functions plots the core area of patches labeled with the corresponding patch id. The edges are the grey cells surrounding the patches and are always shown.

**Value**

ggplot

**Examples**

```
# show "global" core area
show_cores(landscape, class = "global", labels = FALSE)

# show the core area of every class as facet
show_cores(landscape, class = "all", labels = FALSE)

# show only the core area of class 1 and 3
show_cores(landscape, class = c(1, 3), labels = TRUE)
```

---

<code>show_correlation</code>	<i>Show correlation</i>
-------------------------------	-------------------------

---

**Description**

Show correlation

**Usage**

```
show_correlation(data, method = "pearson", diag = TRUE,
  labels = FALSE, vjust = 0, text_size = 15)
```

**Arguments**

<code>data</code>	Tibble with results of as returned by the landscapemetrics package.
<code>method</code>	Type of correlation. See <code>link{cor}</code> for details.
<code>diag</code>	If FALSE, values on the diagonal will be NA and not plotted.
<code>labels</code>	If TRUE, the correlation value will be added as text.
<code>vjust</code>	Will be passed on to ggplot2 as vertical justification of x-axis text.
<code>text_size</code>	Text size of the plot.

## Details

The functions calculates the correlation between all metrics. In order to calculate correlations, for the landscape level more than one landscape needs to be present. All input must be structured as returned by the **landscapemetrics** package.

## Value

ggplot

## Examples

```
metrics <- calculate_lsm(landscape, what = c("patch", "class"))
show_correlation(data = metrics, method = "pearson")

## Not run:
metrics <- calculate_lsm(landscape, what = c("patch", "class"))#'
correlations <- calculate_correlation(metrics)
show_correlation(data = correlations, method = "pearson")

## End(Not run)
```

---

show_lsm	<i>Show landscape metrics</i>
----------	-------------------------------

---

## Description

Show landscape metrics on patch level printed in their corresponding patch.

## Usage

```
show_lsm(landscape, what, class, directions, consider_boundary, edge_depth,
  labels, label_lsm, nrow, ncol)

## S3 method for class 'RasterLayer'
show_lsm(landscape, what, class = "global",
  directions = 8, consider_boundary = FALSE, edge_depth = 1,
  labels = TRUE, label_lsm = FALSE, nrow = NULL, ncol = NULL)

## S3 method for class 'RasterStack'
show_lsm(landscape, what, class = "global",
  directions = 8, consider_boundary = FALSE, edge_depth = 1,
  labels = TRUE, label_lsm = FALSE, nrow = NULL, ncol = NULL)

## S3 method for class 'RasterBrick'
show_lsm(landscape, what, class = "global",
  directions = 8, consider_boundary = FALSE, edge_depth = 1,
```

```

labels = TRUE, label_lsm = FALSE, nrow = NULL, ncol = NULL)

## S3 method for class 'stars'
show_lsm(landscape, what, class = "global",
         directions = 8, consider_boundary = FALSE, edge_depth = 1,
         labels = TRUE, label_lsm = FALSE, nrow = NULL, ncol = NULL)

## S3 method for class 'list'
show_lsm(landscape, what, class = "global",
         directions = 8, consider_boundary = FALSE, edge_depth = 1,
         labels = TRUE, label_lsm = FALSE, nrow = NULL, ncol = NULL)

```

### Arguments

landscape	*Raster object
what	Patch level what to plot
class	How to show the labeled patches: "global" (single map), "all" (every class as facet), or a vector with the specific classes one wants to show (every selected class as facet).
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
consider_boundary	Logical if cells that only neighbour the landscape boundary should be considered as core
edge_depth	Distance (in cells) a cell has to be away from the patch edge to be considered as core cell
labels	Logical flag indicating whether to print or not to print patch labels.
label_lsm	If true, the value of the landscape metric is used as label
nrow, ncol	Number of rows and columns for the facet.

### Details

The function plots all patches with a fill corresponding to the value of the chosen landscape metric on patch level.

### Value

ggplot

### Examples

```

show_lsm(landscape, what = "lsm_p_area", directions = 4)
show_lsm(landscape, what = "lsm_p_shape", class = c(1, 2), label_lsm = TRUE)
show_lsm(landscape, what = "lsm_p_circle", class = 3, labels = TRUE)

```

show\_patches

*Show patches***Description**

Show patches

**Usage**

```
show_patches(landscape, class, directions, labels, nrow, ncol)
```

```
## S3 method for class 'RasterLayer'
show_patches(landscape, class = "global",
  directions = 8, labels = TRUE, nrow = NULL, ncol = NULL)
```

```
## S3 method for class 'RasterStack'
show_patches(landscape, class = "global",
  directions = 8, labels = TRUE, nrow = NULL, ncol = NULL)
```

```
## S3 method for class 'RasterBrick'
show_patches(landscape, class = "global",
  directions = 8, labels = TRUE, nrow = NULL, ncol = NULL)
```

```
## S3 method for class 'stars'
show_patches(landscape, class = "global",
  directions = 8, labels = TRUE, nrow = NULL, ncol = NULL)
```

```
## S3 method for class 'list'
show_patches(landscape, class = "global",
  directions = 8, labels = TRUE, nrow = NULL, ncol = NULL)
```

**Arguments**

landscape	*Raster object
class	How to show the labeled patches: "global" (single map), "all" (every class as facet), or a vector with the specific classes one wants to show (every selected class as facet).
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
labels	Logical flag indicating whether to print or not to print patch labels.
nrow, ncol	Number of rows and columns for the facet.

**Details**

The functions plots the landscape with the patches labeled with the corresponding patch id.

**Value**

ggplot

**Examples**

```
show_patches(landscape)
show_patches(landscape, class = c(1, 2))
show_patches(landscape, class = 3, labels = FALSE)
```

---

spatialize_lsm	<i>spatialize_lsm</i>
----------------	-----------------------

---

**Description**

Spatialize landscape metric values

**Usage**

```
spatialize_lsm(landscape, metric, name, type, what, directions, progress,
  ...)

## S3 method for class 'RasterLayer'
spatialize_lsm(landscape, metric = NULL,
  name = NULL, type = NULL, what = NULL, directions = 8,
  progress = FALSE, ...)

## S3 method for class 'RasterStack'
spatialize_lsm(landscape, metric = NULL,
  name = NULL, type = NULL, what = NULL, directions = 8,
  progress = FALSE, ...)

## S3 method for class 'RasterBrick'
spatialize_lsm(landscape, metric = NULL,
  name = NULL, type = NULL, what = NULL, directions = 8,
  progress = FALSE, ...)

## S3 method for class 'stars'
spatialize_lsm(landscape, metric = NULL, name = NULL,
  type = NULL, what = NULL, directions = 8, progress = FALSE, ...)

## S3 method for class 'list'
spatialize_lsm(landscape, metric = NULL, name = NULL,
  type = NULL, what = NULL, directions = 8, progress = FALSE, ...)
```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
metric	Abbreviation of metrics (e.g. 'area').
name	Full name of metrics (e.g. 'core area')
type	Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').
what	Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. <code>what = c("lsm_c_ca", "lsm_l_ta")</code> .
directions	The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).
progress	Print progress report.
...	Arguments passed on to <code>calculate_lsm()</code> .

**Details**

The functions returns a nested list with RasterLayers. The first level contains each input layer (only one element if RasterLayer was provided). The second level contains a RasterLayer for each selected metric (see `list_lsm` for details) where each cell has the landscape metric value of the patch it belongs to. Only patch level metrics are allowed.

**Value**

list

**See Also**

[list\\_lsm](#)  
[show\\_lsm](#)

**Examples**

```
spatialize_lsm(landscape, what = "lsm_p_area")
```

---

<code>window_lsm</code>	<i>window_lsm</i>
-------------------------	-------------------

---

**Description**

Moving window



**Usage**

```

window_lsm(landscape, window, metric, name, type, what, progress, ...)

## S3 method for class 'RasterLayer'
window_lsm(landscape, window, metric = NULL,
  name = NULL, type = NULL, what = NULL, progress = FALSE, ...)

## S3 method for class 'RasterStack'
window_lsm(landscape, window, metric = NULL,
  name = NULL, type = NULL, what = NULL, progress = FALSE, ...)

## S3 method for class 'RasterBrick'
window_lsm(landscape, window, metric = NULL,
  name = NULL, type = NULL, what = NULL, progress = FALSE, ...)

## S3 method for class 'stars'
window_lsm(landscape, window, metric = NULL,
  name = NULL, type = NULL, what = NULL, progress = FALSE, ...)

## S3 method for class 'list'
window_lsm(landscape, window, metric = NULL,
  name = NULL, type = NULL, what = NULL, progress = FALSE, ...)

```

**Arguments**

landscape	Raster* Layer, Stack, Brick or a list of rasterLayers.
window	matrix
metric	Abbreviation of metrics (e.g. 'area').
name	Full name of metrics (e.g. 'core area')
type	Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').
what	Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. <code>what = c("lsm_c_ca", "lsm_l_ta")</code> .
progress	Print progress report.
...	Arguments passed on to <code>calculate_lsm()</code> .

**Details**

The function calculates for each focal cell the selected landscape metrics (currently only landscape level metrics are allowed) for a local neighbourhood. The neighbourhood can be specified using a matrix. For more details, see `?raster::focal()`. The result will be a `RasterLayer` in which each focal cell includes the value of its neighbourhood and thereby allows to show gradients and variability in the landscape (Hagen-Zanker 2016). To be type stable, the actually result is always a nested list (first level for `RasterStack` layers, second level for selected landscape metrics).

**Value**

list

## References

Fletcher, R., Fortin, M.-J. 2018. Spatial Ecology and Conservation Modeling: Applications with R. Springer International Publishing. 523 pages

Hagen-Zanker, A. (2016). A computational framework for generalized moving windows and its application to landscape pattern analysis. International journal of applied earth observation and geoinformation, 44, 205-216.

McGarigal, K., Cushman, S.A., and Ene E. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following website: <http://www.umass.edu/landeco/research/fragstats/fragstats.html>

## See Also

[list\\_lsm](#)  
[calculate\\_lsm](#)  
[focal](#)

## Examples

```
## Not run:  
window <- matrix(1, nrow = 5, ncol = 5)  
window_lsm(landscape, window = window, what = c("lsm_l_pr", "lsm_l_joinent"))  
window_lsm(landscape_stack, window = window, what = c("lsm_l_pr", "lsm_l_joinent"))  
  
## End(Not run)
```

# Index

## \*Topic **datasets**

augusta\_nlcd, [6](#)  
fragstats\_class\_augusta\_nlcd, [12](#)  
fragstats\_class\_landscape, [13](#)  
fragstats\_class\_podlasie, [13](#)  
fragstats\_landscape\_augusta\_nlcd, [14](#)  
fragstats\_landscape\_landscape, [14](#)  
fragstats\_landscape\_podlasie, [15](#)  
fragstats\_patch\_augusta\_nlcd, [15](#)  
fragstats\_patch\_landscape, [16](#)  
fragstats\_patch\_podlasie, [16](#)  
landscape, [25](#)  
lsm\_abbreviations\_names, [28](#)  
podlasie\_ccilc, [235](#)

augusta\_nlcd, [6](#)

calculate\_correlation, [6](#)  
calculate\_lsm, [7](#), [12](#), [237](#), [239](#), [250](#)  
check\_landscape, [9](#)  
construct\_buffer, [239](#)  
cv, [31](#), [37](#), [55](#), [64](#), [72](#), [77](#), [81](#), [84](#), [98](#), [107](#), [118](#),  
[123](#), [142](#), [149](#), [157](#), [163](#), [168](#), [171](#),  
[189](#), [200](#)

extract\_lsm, [10](#)

focal, [250](#)  
fragstats\_class\_augusta\_nlcd, [12](#)  
fragstats\_class\_landscape, [13](#)  
fragstats\_class\_podlasie, [13](#)  
fragstats\_landscape\_augusta\_nlcd, [14](#)  
fragstats\_landscape\_landscape, [14](#)  
fragstats\_landscape\_podlasie, [15](#)  
fragstats\_patch\_augusta\_nlcd, [15](#)  
fragstats\_patch\_landscape, [16](#)  
fragstats\_patch\_podlasie, [16](#)

get\_adjacencies, [17](#)  
get\_boundaries, [18](#)

get\_circumscribingcircle, [20](#)  
get\_nearestneighbour, [21](#)  
get\_patches, [21](#), [22](#), [22](#)  
get\_unique\_values, [24](#)

landscape, [25](#)  
landscapemetrics, [26](#)  
landscapemetrics-package  
(landscapemetrics), [26](#)  
list\_lsm, [9](#), [27](#), [237](#), [239](#), [241](#), [248](#), [250](#)  
lsm\_abbreviations\_names, [28](#)  
lsm\_c\_ai, [29](#), [117](#)  
lsm\_c\_area\_cv, [30](#), [33](#), [34](#), [118](#), [120](#), [121](#), [217](#)  
lsm\_c\_area\_mn, [31](#), [32](#), [34](#), [118](#), [120](#), [121](#), [217](#)  
lsm\_c\_area\_sd, [31](#), [33](#), [33](#), [118](#), [120](#), [121](#), [217](#)  
lsm\_c\_ca, [32](#), [35](#), [106](#), [212](#), [217](#)  
lsm\_c\_cai\_cv, [36](#), [39](#), [41](#), [123](#), [125](#), [126](#), [219](#)  
lsm\_c\_cai\_mn, [37](#), [38](#), [41](#), [123](#), [126](#), [219](#)  
lsm\_c\_cai\_sd, [37](#), [39](#), [39](#), [123](#), [125](#), [126](#), [219](#)  
lsm\_c\_circle\_cv, [41](#), [44](#), [45](#), [128](#), [129](#), [131](#),  
[220](#)  
lsm\_c\_circle\_mn, [42](#), [43](#), [45](#), [128](#), [129](#), [131](#),  
[220](#)  
lsm\_c\_circle\_sd, [42](#), [44](#), [44](#), [128](#), [129](#), [131](#),  
[220](#)  
lsm\_c\_clumpy, [46](#)  
lsm\_c\_cohesion, [47](#)  
lsm\_c\_contig\_cv, [49](#), [50](#), [52](#), [53](#), [137](#), [139](#),  
[140](#), [222](#)  
lsm\_c\_contig\_mn, [50](#), [50](#), [53](#), [137](#), [139](#), [140](#),  
[222](#)  
lsm\_c\_contig\_sd, [52](#), [52](#), [137](#), [139](#), [140](#), [222](#)  
lsm\_c\_core\_cv, [54](#), [57](#), [58](#), [142](#), [144](#), [145](#), [224](#)  
lsm\_c\_core\_mn, [55](#), [55](#), [58](#), [142](#), [144](#), [145](#), [224](#)  
lsm\_c\_core\_sd, [55](#), [57](#), [57](#), [142](#), [144](#), [145](#), [224](#)  
lsm\_c\_cpiland, [59](#), [219](#)  
lsm\_c\_dcad, [60](#), [147](#)  
lsm\_c\_dcore\_cv, [62](#), [65](#), [67](#), [149](#), [151](#), [152](#),  
[230](#)

- lsm\_c\_dcore\_mn, [64](#), [64](#), [67](#), [149](#), [151](#), [152](#), [230](#)
- lsm\_c\_dcore\_sd, [64](#), [65](#), [66](#), [149](#), [151](#), [152](#), [230](#)
- lsm\_c\_division, [67](#), [90](#), [154](#), [178](#)
- lsm\_c\_division.stars(lsm\_c\_area\_sd), [33](#)
- lsm\_c\_ed, [69](#), [155](#)
- lsm\_c\_enn\_cv, [71](#), [74](#), [75](#), [157](#), [159](#), [160](#), [225](#)
- lsm\_c\_enn\_mn, [72](#), [72](#), [75](#), [157](#), [159](#), [160](#), [225](#)
- lsm\_c\_enn\_sd, [72](#), [74](#), [74](#), [157](#), [159](#), [160](#), [225](#)
- lsm\_c\_frac\_cv, [76](#), [78](#), [80](#), [163](#), [165](#), [166](#), [227](#)
- lsm\_c\_frac\_mn, [77](#), [77](#), [80](#), [163](#), [165](#), [166](#), [227](#)
- lsm\_c\_frac\_sd, [77](#), [78](#), [79](#), [163](#), [165](#), [166](#), [227](#)
- lsm\_c\_gyrate\_cv, [80](#), [83](#), [84](#), [168](#), [169](#), [171](#), [228](#)
- lsm\_c\_gyrate\_mn, [81](#), [82](#), [84](#), [168](#), [169](#), [171](#), [228](#)
- lsm\_c\_gyrate\_sd, [81](#), [83](#), [83](#), [168](#), [169](#), [171](#), [228](#)
- lsm\_c\_iji, [85](#), [172](#)
- lsm\_c\_lpi, [86](#), [175](#)
- lsm\_c\_lsi, [88](#), [177](#)
- lsm\_c\_mesh, [68](#), [89](#), [153](#), [178](#)
- lsm\_c\_ndca, [62](#), [91](#), [147](#), [184](#), [230](#)
- lsm\_c\_nlsi, [92](#)
- lsm\_c\_np, [94](#), [102](#), [103](#), [186](#), [193](#)
- lsm\_c\_pafrac, [95](#), [187](#)
- lsm\_c\_para\_cv, [97](#), [100](#), [101](#), [189](#), [190](#), [192](#), [231](#)
- lsm\_c\_para\_mn, [98](#), [99](#), [101](#), [189](#), [190](#), [192](#), [231](#)
- lsm\_c\_para\_sd, [98](#), [100](#), [100](#), [189](#), [190](#), [192](#), [231](#)
- lsm\_c\_pd, [102](#), [193](#)
- lsm\_c\_pladj, [103](#)
- lsm\_c\_pland, [104](#), [205](#), [206](#), [208](#), [209](#)
- lsm\_c\_shape\_cv, [106](#), [109](#), [110](#), [200](#), [202](#), [203](#), [234](#)
- lsm\_c\_shape\_mn, [107](#), [108](#), [110](#), [200](#), [202](#), [203](#), [234](#)
- lsm\_c\_shape\_sd, [107](#), [109](#), [109](#), [200](#), [202](#), [203](#), [234](#)
- lsm\_c\_split, [111](#), [211](#)
- lsm\_c\_tca, [92](#), [112](#), [184](#), [214](#), [224](#)
- lsm\_c\_te, [70](#), [114](#)
- lsm\_l\_ai, [30](#), [115](#)
- lsm\_l\_area\_cv, [31](#), [33](#), [34](#), [117](#), [120](#), [121](#), [217](#)
- lsm\_l\_area\_mn, [31](#), [33](#), [34](#), [118](#), [118](#), [121](#), [217](#)
- lsm\_l\_area\_sd, [31](#), [33](#), [34](#), [118](#), [120](#), [120](#), [217](#)
- lsm\_l\_cai\_cv, [37](#), [39](#), [41](#), [121](#), [125](#), [126](#), [219](#)
- lsm\_l\_cai\_mn, [37](#), [39](#), [41](#), [123](#), [123](#), [126](#), [219](#)
- lsm\_l\_cai\_sd, [37](#), [39](#), [41](#), [123](#), [125](#), [125](#), [219](#)
- lsm\_l\_circle\_cv, [42](#), [44](#), [45](#), [127](#), [129](#), [131](#), [220](#)
- lsm\_l\_circle\_mn, [42](#), [44](#), [45](#), [128](#), [128](#), [131](#), [220](#)
- lsm\_l\_circle\_sd, [42](#), [44](#), [45](#), [128](#), [129](#), [130](#), [220](#)
- lsm\_l\_cohesion, [48](#), [131](#), [132](#)
- lsm\_l\_condent, [133](#), [162](#), [174](#), [183](#)
- lsm\_l\_contag, [134](#)
- lsm\_l\_contig\_cv, [50](#), [52](#), [53](#), [136](#), [139](#), [140](#), [222](#)
- lsm\_l\_contig\_mn, [50](#), [52](#), [53](#), [137](#), [137](#), [140](#), [222](#)
- lsm\_l\_contig\_sd, [50](#), [52](#), [53](#), [137](#), [139](#), [139](#), [222](#)
- lsm\_l\_core\_cv, [55](#), [57](#), [58](#), [141](#), [144](#), [145](#), [224](#)
- lsm\_l\_core\_mn, [55](#), [57](#), [58](#), [142](#), [142](#), [145](#), [224](#)
- lsm\_l\_core\_sd, [55](#), [57](#), [58](#), [142](#), [144](#), [144](#), [224](#)
- lsm\_l\_dcad, [62](#), [146](#)
- lsm\_l\_dcore\_cv, [64](#), [65](#), [67](#), [147](#), [151](#), [152](#), [230](#)
- lsm\_l\_dcore\_mn, [64](#), [65](#), [67](#), [149](#), [149](#), [152](#), [230](#)
- lsm\_l\_dcore\_sd, [64](#), [65](#), [67](#), [149](#), [151](#), [151](#), [230](#)
- lsm\_l\_division, [69](#), [153](#)
- lsm\_l\_ed, [70](#), [154](#)
- lsm\_l\_enn\_cv, [72](#), [74](#), [75](#), [156](#), [159](#), [160](#), [225](#)
- lsm\_l\_enn\_mn, [72](#), [74](#), [75](#), [157](#), [157](#), [160](#), [225](#)
- lsm\_l\_enn\_sd, [72](#), [74](#), [75](#), [157](#), [159](#), [159](#), [225](#)
- lsm\_l\_ent, [134](#), [161](#), [174](#), [183](#)
- lsm\_l\_frac\_cv, [77](#), [78](#), [80](#), [162](#), [165](#), [166](#), [227](#)
- lsm\_l\_frac\_mn, [77](#), [78](#), [80](#), [163](#), [164](#), [166](#), [227](#)
- lsm\_l\_frac\_sd, [77](#), [78](#), [80](#), [163](#), [165](#), [165](#), [227](#)
- lsm\_l\_gyrate\_cv, [81](#), [83](#), [84](#), [167](#), [169](#), [171](#), [228](#)
- lsm\_l\_gyrate\_mn, [81](#), [83](#), [84](#), [168](#), [168](#), [171](#), [228](#)
- lsm\_l\_gyrate\_sd, [81](#), [83](#), [84](#), [168](#), [169](#), [170](#), [228](#)
- lsm\_l\_iji, [86](#), [171](#)
- lsm\_l\_joinent, [134](#), [162](#), [173](#), [182](#), [183](#)
- lsm\_l\_lpi, [87](#), [174](#)
- lsm\_l\_lsi, [89](#), [175](#)

- lsm\_l\_mesh, [90](#), [177](#)
- lsm\_l\_msidi, [179](#)
- lsm\_l\_msiei, [180](#)
- lsm\_l\_mutinf, [134](#), [162](#), [174](#), [182](#)
- lsm\_l\_ndca, [92](#), [183](#), [230](#)
- lsm\_l\_np, [95](#), [185](#), [193](#)
- lsm\_l\_pafrac, [97](#), [186](#)
- lsm\_l\_para\_cv, [98](#), [100](#), [101](#), [188](#), [190](#), [192](#), [231](#)
- lsm\_l\_para\_mn, [98](#), [100](#), [101](#), [189](#), [189](#), [192](#), [231](#)
- lsm\_l\_para\_sd, [98](#), [100](#), [101](#), [189](#), [190](#), [191](#), [231](#)
- lsm\_l\_pd, [103](#), [192](#)
- lsm\_l\_pladj, [194](#)
- lsm\_l\_pr, [195](#), [197](#), [206](#), [208](#), [209](#)
- lsm\_l\_prd, [196](#)
- lsm\_l\_rpr, [198](#)
- lsm\_l\_shape\_cv, [107](#), [109](#), [110](#), [199](#), [202](#), [203](#), [234](#)
- lsm\_l\_shape\_mn, [107](#), [109](#), [110](#), [200](#), [201](#), [203](#), [234](#)
- lsm\_l\_shape\_sd, [107](#), [109](#), [110](#), [200](#), [202](#), [202](#), [234](#)
- lsm\_l\_shdi, [204](#), [207](#)
- lsm\_l\_shei, [205](#)
- lsm\_l\_sidi, [180](#), [207](#)
- lsm\_l\_siei, [181](#), [208](#)
- lsm\_l\_split, [112](#), [210](#)
- lsm\_l\_ta, [36](#), [60](#), [62](#), [69](#), [70](#), [87](#), [90](#), [103](#), [106](#), [112](#), [119](#), [147](#), [154](#), [155](#), [175](#), [178](#), [193](#), [211](#), [211](#), [217](#)
- lsm\_l\_tca, [113](#), [213](#), [224](#)
- lsm\_l\_te, [115](#), [155](#), [214](#), [216](#)
- lsm\_p\_area, [31](#), [33](#), [34](#), [36](#), [48](#), [69](#), [87](#), [90](#), [97](#), [112](#), [118](#), [120](#), [121](#), [132](#), [154](#), [175](#), [178](#), [187](#), [211](#), [212](#), [216](#), [219](#), [220](#), [227](#), [231](#), [234](#)
- lsm\_p\_cai, [37](#), [39](#), [41](#), [123](#), [125](#), [126](#), [217](#)
- lsm\_p\_circle, [42](#), [44](#), [45](#), [128](#), [129](#), [131](#), [219](#)
- lsm\_p\_contig, [50](#), [52](#), [53](#), [137](#), [139](#), [140](#), [221](#)
- lsm\_p\_core, [55](#), [57](#), [58](#), [60](#), [113](#), [142](#), [144](#), [145](#), [214](#), [219](#), [222](#)
- lsm\_p\_enn, [72](#), [74](#), [75](#), [157](#), [159](#), [160](#), [224](#)
- lsm\_p\_frac, [77](#), [78](#), [80](#), [163](#), [165](#), [166](#), [226](#)
- lsm\_p\_gyrate, [81](#), [83](#), [84](#), [168](#), [169](#), [171](#), [227](#)
- lsm\_p\_ncore, [64](#), [65](#), [67](#), [92](#), [149](#), [151](#), [152](#), [184](#), [229](#)
- lsm\_p\_para, [98](#), [100](#), [101](#), [189](#), [190](#), [192](#), [230](#)
- lsm\_p\_perim, [48](#), [97](#), [115](#), [132](#), [187](#), [216](#), [227](#), [231](#), [232](#), [234](#)
- lsm\_p\_shape, [89](#), [94](#), [107](#), [109](#), [110](#), [177](#), [200](#), [202](#), [203](#), [233](#)
- mean, [33](#), [39](#), [42](#), [44](#), [45](#), [57](#), [65](#), [74](#), [78](#), [83](#), [100](#), [109](#), [120](#), [125](#), [128](#), [129](#), [131](#), [144](#), [151](#), [159](#), [165](#), [169](#), [190](#), [202](#)
- options\_landscapemetrics, [235](#)
- podlasie\_ccilc, [235](#)
- sample\_lsm, [236](#), [239](#)
- scale\_sample, [238](#), [241](#)
- scale\_window, [240](#)
- sd, [34](#), [41](#), [58](#), [67](#), [75](#), [80](#), [101](#), [110](#), [121](#), [126](#), [145](#), [152](#), [160](#), [166](#), [192](#), [203](#)
- show\_cores, [242](#)
- show\_correlation, [243](#)
- show\_lsm, [244](#), [248](#)
- show\_patches, [246](#)
- spatialize\_lsm, [247](#)
- sum, [36](#), [212](#)
- window\_lsm, [241](#), [248](#)