

Package ‘EconGeo’

June 26, 2023

Title Computing Key Indicators of the Spatial Distribution of Economic Activities

Version 2.0

Date 2023-06-22

Description Computes a series of indices commonly used in the fields of economic geography, economic complexity, and evolutionary economics to describe the location, distribution, spatial organization, structure, and complexity of economic activities. Functions include basic spatial indicators such as the location quotient, the Krugman specialization index, the Herfindahl or the Shannon entropy indices but also more advanced functions to compute different forms of normalized relatedness between economic activities or network-based measures of economic complexity. Most of the functions use matrix calculus and are based on bipartite (incidence) matrices consisting of region - industry pairs. These are described in Balland (2017) <<http://econ.geo.uu.nl/peeg/peeg1709.pdf>>.

URL <https://github.com/PABalland/EconGeo>

Depends R (>= 3.3.1)

Imports Matrix, reshape

License GPL-2 | GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

BugReports <https://github.com/PABalland/EconGeo/issues>

NeedsCompilation no

Author Pierre-Alexandre Balland [aut, cre, cph]

Maintainer Pierre-Alexandre Balland <p.balland@uu.nl>

Repository CRAN

Date/Publication 2023-06-26 12:00:05 UTC

R topics documented:

co_occurrence	3
diversity	4

ease_recombination	5
entropy	6
entry_list	7
entry_mat	9
exit_list	10
exit_mat	12
expy	13
get_list	14
get_matrix	15
gini	16
growth_ind	17
growth_list	19
growth_list_ind	20
growth_list_reg	21
growth_mat	23
growth_reg	24
hachman	25
herfindahl	26
hoover_curve	27
hoover_gini	29
hoover_index	31
inv_norm_ubiquity	32
kci	33
krugman_index	35
locational_gini	36
locational_gini_curve	37
location_quotient	38
location_quotient_avg	40
lorenz_curve	41
match_mat	43
modular_complexity	44
modular_complexity_avg	45
more	47
mort	48
norm_ubiquity	50
prody	51
rca	52
relatedness	53
relatedness_density	55
relatedness_density_ext	56
relatedness_density_ext_avg	58
relatedness_density_int	59
relatedness_density_int_avg	60
spec_coeff	62
tci	63
ubiquity	64
weighted_avg	65
z_score	66

co_occurrence	<i>Compute the number of co-occurrences between industry pairs from an incidence (industry - event) matrix</i>
---------------	--

Description

This function computes the number of co-occurrences between industry pairs from an incidence (industry - event) matrix

Usage

```
co_occurrence(mat, diagonal = FALSE, list = FALSE)
```

Arguments

mat	An incidence matrix with industries in rows and events in columns
diagonal	Logical; shall the values in the diagonal of the co-occurrence matrix be included in the output? Defaults to FALSE (values in the diagonal are set to 0), but can be set to TRUE (values in the diagonal reflects in how many events a single industry can be found)
list	Logical; is the input a list? Defaults to FALSE (input = adjacency matrix), but can be set to TRUE if the input is an edge list

Value

The co-occurrence matrix as an R matrix object.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

See Also

[relatedness](#), [relatedness_density](#)

Examples

```
## generate a region - events matrix
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 5)
rownames(mat) <- c("I1", "I2", "I3", "I4")
colnames(mat) <- c("US1", "US2", "US3", "US4", "US5")

## run the function
co_occurrence(mat)
co_occurrence(mat, diagonal = TRUE)

## generate a regular data frame (list)
my_list <- get_list(mat)

## run the function
co_occurrence(my_list, list = TRUE)
co_occurrence(my_list, list = TRUE, diagonal = TRUE)
```

diversity

Compute a simple measure of diversity of regions

Description

This function computes a simple measure of diversity of regions by counting the number of industries in which a region has a relative comparative advantage (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
diversity(mat, rca = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
rca	Logical; should the index of relative comparative advantage (RCA - also referred to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Value

A numeric vector representing the share of a tech in a city's portfolio

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[ubiquity](#), [location_quotient](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
diversity(mat, rca = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
diversity(mat)
```

ease_recombination *Compute the ease of recombination of a given technological class*

Description

This function computes the ease of recombination of a given technological class from technological classes - patents (incidence) matrices

Usage

```
ease_recombination(mat, sparse = FALSE, list = FALSE)
```

Arguments

mat	A bipartite adjacency matrix (can be a sparse matrix)
sparse	Logical; is the input matrix a sparse matrix? Defaults to FALSE, but can be set to TRUE if the input matrix is a sparse matrix
list	Logical; is the input a list? Defaults to FALSE, but can be set to TRUE if the input matrix is a list

Value

A data frame with two columns: "tech" representing the technological class and "eor" representing the ease of recombination of the technological class

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Fleming, L. and Sorenson, O. (2001) Technology as a complex adaptive system: evidence from patent data, *Research Policy* **30**: 1019-1039

See Also

[modular_complexity](#), [tci](#), [mort](#)

Examples

```
## generate a technology - patent matrix
set.seed(31)
mat <- matrix(sample(0:1, 30, replace = TRUE), ncol = 5)
rownames(mat) <- c("T1", "T2", "T3", "T4", "T5", "T6")
colnames(mat) <- c("US1", "US2", "US3", "US4", "US5")

## generate a technology - patent sparse matrix
library(Matrix)
smat <- Matrix(mat, sparse = TRUE)

## run the function
ease_recombination(mat)
ease_recombination(smat, sparse = TRUE)

## generate a regular data frame (list)
my_list <- get_list(mat)

## run the function
ease_recombination(my_list, list = TRUE)
```

entropy

Compute the Shannon entropy index from regions - industries matrices

Description

This function computes the Shannon entropy index from regions - industries matrices from (incidence) regions - industries matrices

Usage

```
entropy(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A numeric vector representing the Shannon entropy index computed from the regions - industries matrix

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Shannon, C.E., Weaver, W. (1949) *The Mathematical Theory of Communication*. Univ of Illinois Press.

Frenken, K., Van Oort, F. and Verburg, T. (2007) Related variety, unrelated variety and regional economic growth, *Regional studies* **41** (5): 685-697.

See Also

[diversity](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
entropy(mat)
```

entry_list	<i>Generate a data frame of entry events from multiple regions - industries matrices (same matrix composition for the different periods)</i>
------------	--

Description

This function generates a data frame of entry events from multiple regions - industries matrices (different matrix compositions are allowed). In this function, the maximum number of periods is limited to 20.

Usage

```
entry_list(...)
```

Arguments

... Incidence matrices with regions in rows and industries in columns (period ... - optional)

Value

A data frame representing the entry events from multiple regions - industries matrices, with columns "region" (representing the region), "industry" (representing the industry), "entry" (representing the entry event), and "period" (representing the period)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
 Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[growth_list](#), [exit_list](#)

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
mat2[3, 1] <- 1

## run the function
entry_list(mat1, mat2)

## generate a third region - industry matrix in which cells represent the presence/absence
## of a RCA (period 3)
mat3 <- mat2
mat3[5, 2] <- 1

## run the function
```



```
entry_list(mat1, mat2, mat3)

## generate a fourth region - industry matrix in which cells represent the presence/absence
## of a RCA (period 4)
mat4 <- mat3
mat4[5, 4] <- 1

## run the function
entry_list(mat1, mat2, mat3, mat4)
```

entry_mat	<i>Generate a matrix of entry events from two regions - industries matrices (same matrix composition from two different periods)</i>
-----------	--

Description

This function generates a matrix of entry events from two regions - industries matrices (different matrix compositions are allowed)

Usage

```
entry_mat(mat1, mat2)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1)
mat2	An incidence matrix with regions in rows and industries in columns (period 2)

Value

A matrix representing the entry events from two regions - industries matrices, with rows representing regions and columns representing industries

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[growth_list](#), [entry_list](#), [exit_list](#)

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
mat2[3, 1] <- 1

## run the function
entry_mat(mat1, mat2)
```

exit_list	<i>Generate a data frame of exit events from multiple regions - industries matrices (same matrix composition for the different periods)</i>
-----------	---

Description

This function generates a data frame of exit events from multiple regions - industries matrices (different matrix compositions are allowed). In this function, the maximum number of periods is limited to 20.

Usage

```
exit_list(...)
```

Arguments

... Incidence matrices with regions in rows and industries in columns (period ... - optional)

Value

A data frame representing the exit events from multiple regions - industries matrices, with columns "region" (representing the region), "industry" (representing the industry), "exit" (representing the exit event), and "period" (representing the period)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[entry_list](#)

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
mat2[2, 1] <- 0

## run the function
exit_list(mat1, mat2)

## generate a third region - industry matrix in which cells represent the presence/absence
## of a RCA (period 3)
mat3 <- mat2
mat3[5, 1] <- 0

## run the function
exit_list(mat1, mat2, mat3)

## generate a fourth region - industry matrix in which cells represent the presence/absence
## of a RCA (period 4)
mat4 <- mat3
mat4[5, 3] <- 0

## run the function
exit_list(mat1, mat2, mat3, mat4)
```

exit_mat	<i>Generate a matrix of exit events from two regions - industries matrices (same matrix composition from two different periods)</i>
----------	---

Description

This function generates a matrix of exit events from two regions - industries matrices (different matrix compositions are allowed)

Usage

```
exit_mat(mat1, mat2)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1)
mat2	An incidence matrix with regions in rows and industries in columns (period 2)

Value

A matrix representing the exit events from two regions - industries matrices, with rows representing regions and columns representing industries

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[growth_list](#), [exit_list](#), [entry_list](#)

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
```

```
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
mat2[2, 1] <- 0

## run the function
exit_mat(mat1, mat2)
```

expy

Compute the expy index of regions from regions - industries matrices

Description

This function computes the expy index of regions from (incidence) regions - industries matrices, as proposed by Hausmann, Hwang & Rodrik (2007). The index is a measure of the productivity level associated with a region's specialization pattern.

Usage

```
expy(mat, vec)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
vec	A vector that gives GDP, R&D, education or any other relevant regional attribute that will be used to compute the weighted average for each industry

Value

A numeric vector representing the expy index of regions computed from the regions - industries matrix

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123

Hausmann, R., Hwang, J. & Rodrik, D. (2007) What you export matters, *Journal of economic growth* **12**: 1-25.

See Also[location_quotient](#)**Examples**

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## a vector of GDP of regions
vec <- c(5, 10, 15, 25, 50)
## run the function
expy(mat, vec)
```

`get_list`*Create regular data frames from regions - industries matrices*

Description

This function creates regular data frames with three columns (regions, industries, count) from (incidence) matrices (wide to long format) using the reshape2 package

Usage

```
get_list(mat)
```

Arguments

`mat` An incidence matrix with regions in rows and industries in columns (or the other way around)

Value

A data frame with three columns: "Region" (representing the region), "Industry" (representing the industry), and "Count" (representing the count of occurrences)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

See Also[get_matrix](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
get_list(mat)
```

get_matrix

Create regions - industries matrices from regular data frames

Description

This function creates regions - industries (incidence) matrices from regular data frames (long to wide format) using the reshape2 package or the Matrix package

Usage

```
get_matrix (my_data, sparse = FALSE)
```

Arguments

my_data	is a data frame with three columns (regions, industries, count)
sparse	Logical; shall the returned output be a sparse matrix? Defaults to FALSE, but can be set to TRUE if the dataset is very large

Value

A regions - industries matrix in either dense or sparse format, depending on the value of the "sparse" parameter

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

See Also

[get_list](#)

Examples

```
## generate a region - industry data frame
set.seed(31)
region <- c("R1", "R1", "R1", "R1", "R2", "R2", "R3", "R4", "R5", "R5")
industry <- c("I1", "I2", "I3", "I4", "I1", "I2", "I1", "I1", "I3", "I3")
my_data <- data.frame(region, industry)
my_data$count <- 1

## run the function
get_matrix(my_data)
get_matrix(my_data, sparse = TRUE)
```

gini

Compute the Gini coefficient

Description

This function computes the Gini coefficient. The Gini index measures spatial inequality. It ranges from 0 (perfect income equality) to 1 (perfect income inequality) and is derived from the Lorenz curve. The Gini coefficient is defined as a ratio of two surfaces derived from the Lorenz curve. The numerator is given by the area between the Lorenz curve of the distribution and the uniform distribution line (45 degrees line). The denominator is the area under the uniform distribution line (the lower triangle). This index gives an indication of the unequal distribution of an industry across n regions. Maximum inequality in the sample occurs when $n-1$ regions have a score of zero and one region has a positive score. The maximum value of the Gini coefficient is $(n-1)/n$ and approaches 1 (theoretical maximum limit) as the number of observations (regions) increases.

Usage

```
gini(mat)
```

Arguments

mat A region-industry count matrix

Value

The Gini coefficient or a data frame with the Gini coefficient for each industry (if the input is a matrix with multiple columns)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Gini, C. (1921) Measurement of Inequality of Incomes, *The Economic Journal* **31**: 124-126

See Also

[hoover_gini](#), [locational_gini](#), [locational_gini_curve](#), [lorenz_curve](#), [hoover_curve](#)

Examples

```
## generate vectors of industrial count
ind <- c(0, 10, 10, 30, 50)

## run the function
gini(ind)

## generate a region - industry matrix
mat <- matrix(
  c(
    0, 1, 0, 0,
    0, 1, 0, 0,
    0, 1, 0, 0,
    0, 1, 0, 1,
    0, 1, 1, 1
  ),
  ncol = 4, byrow = TRUE
)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
gini(mat)

## run the function by aggregating all industries
gini(rowSums(mat))

## run the function for industry #1 only (perfect equality)
gini(mat[, 1])

## run the function for industry #2 only (perfect equality)
gini(mat[, 2])

## run the function for industry #3 only (perfect inequality: max gini = (5-1)/5)
gini(mat[, 3])

## run the function for industry #4 only (top 40% produces 100% of the output)
gini(mat[, 4])
```

growth_ind

Generate a matrix of industrial growth by industries from two regions - industries matrices (same matrix composition from two different periods)

Description

This function generates a matrix of industrial growth by industries from two regions - industries matrices (same matrix composition from two different periods)

Usage

```
growth_ind(mat1, mat2)
```

Arguments

mat1 An incidence matrix with regions in rows and industries in columns (period 1)
mat2 An incidence matrix with regions in rows and industries in columns (period 2)

Value

A matrix of industrial growth by industries

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[entry_list](#), [exit_list](#)

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3, 1] <- 8

## run the function
growth_ind(mat1, mat2)
```

growth_list	<i>Generate a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods)</i>
-------------	--

Description

This function generates a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods). In this function, the maximum number of periods is limited to 20.

Usage

```
growth_list(...)
```

Arguments

... Incidence matrices with regions in rows and industries in columns (period ... - optional)

Value

A data frame of industrial growth in regions

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[exit_list](#)

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")
```

```
## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3, 1] <- 8

## run the function
growth_list(mat1, mat2)

## generate a third region - industry matrix with full count (period 3)
mat3 <- mat2
mat3[5, 2] <- 1

## run the function
growth_list(mat1, mat2, mat3)

## generate a fourth region - industry matrix with full count (period 4)
mat4 <- mat3
mat4[5, 4] <- 1

## run the function
growth_list(mat1, mat2, mat3, mat4)
```

growth_list_ind	<i>Generate a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods)</i>
-----------------	--

Description

This function generates a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods). In this function, the maximum number of periods is limited to 20.

Usage

```
growth_list_ind(...)
```

Arguments

... Incidence matrices with regions in rows and industries in columns (period ... - optional)

Value

A data frame of industrial growth in regions

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[growth_list](#), [entry_list](#), [exit_list](#)

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3, 1] <- 8

## run the function
growth_list_ind(mat1, mat2)

## generate a third region - industry matrix with full count (period 3)
mat3 <- mat2
mat3[5, 2] <- 1

## run the function
growth_list_ind(mat1, mat2, mat3)

## generate a fourth region - industry matrix with full count (period 4)
mat4 <- mat3
mat4[5, 4] <- 1

## run the function
growth_list_ind(mat1, mat2, mat3, mat4)
```

growth_list_reg	<i>Generate a data frame of region growth from multiple regions - industries matrices (same matrix composition for the different periods)</i>
-----------------	---

Description

This function generates a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods). In this function, the maximum number of periods is limited to 20.

Usage

```
growth_list_reg(...)
```

Arguments

... Incidence matrices with regions in rows and industries in columns (period ... - optional)

Value

A data frame of region growth from multiple regions - industries matrices

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[growth_list](#), [entry_list](#), [exit_list](#)

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3, 1] <- 8

## run the function
growth_list_reg(mat1, mat2)

## generate a third region - industry matrix with full count (period 3)
mat3 <- mat2
mat3[5, 2] <- 1

## run the function
growth_list_reg(mat1, mat2, mat3)
```

```
## generate a fourth region - industry matrix with full count (period 4)
mat4 <- mat3
mat4[5, 4] <- 1

## run the function
growth_list_reg(mat1, mat2, mat3, mat4)
```

growth_mat	<i>Generate a matrix of industrial growth in regions from two regions - industries matrices (same matrix composition from two different periods)</i>
------------	--

Description

This function generates a matrix of industrial growth in regions from two regions - industries matrices (same matrix composition from two different periods)

Usage

```
growth_mat(mat1, mat2)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1)
mat2	An incidence matrix with regions in rows and industries in columns (period 2)

Value

A matrix of industrial growth in regions from two regions - industries matrices

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[growth_list](#), [entry_list](#), [exit_list](#)

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3, 1] <- 8

## run the function
growth_mat(mat1, mat2)
```

growth_reg	<i>Generate a matrix of industrial growth by regions from two regions - industries matrices (same matrix composition from two different periods)</i>
------------	--

Description

This function generates a matrix of industrial growth by regions from two regions - industries matrices (same matrix composition from two different periods)

Usage

```
growth_reg(mat1, mat2)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1)
mat2	An incidence matrix with regions in rows and industries in columns (period 2)

Value

A vector of industrial growth by regions from two regions - industries matrices

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[growth_list](#), [entry_list](#), [exit_list](#)

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3, 1] <- 8

## run the function
growth_reg(mat1, mat2)
```

hachman

Compute the Hachman index from regions - industries matrices

Description

This function computes the Hachman index from regions - industries matrices. The Hachman index indicates how closely the industrial distribution of a region resembles the one of a more global economy (nation, world). The index varies between 0 (extreme dissimilarity between the region and the more global economy) and 1 (extreme similarity between the region and the more global economy)

Usage

```
hachman(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A vector of Hachman index values indicating the similarity between the industrial distribution of a region and a more global economy

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

See Also

[location_quotient_avg](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
hachman(mat)
```

herfindahl

Compute the Herfindahl index from regions - industries matrices

Description

This function computes the Herfindahl index from regions - industries matrices from (incidence) regions - industries matrices. This index is also known as the Herfindahl-Hirschman index.

Usage

```
herfindahl(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A vector of Herfindahl index values indicating the concentration of industries within regions

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Herfindahl, O.C. (1959) *Copper Costs and Prices: 1870-1957*. Baltimore: The Johns Hopkins Press.

Hirschman, A.O. (1945) *National Power and the Structure of Foreign Trade*, Berkeley and Los Angeles: University of California Press.

See Also

[krugman_index](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
herfindahl(mat)
```

hoover_curve

Plot a Hoover curve from regions - industries matrices

Description

This function plots a Hoover curve from regions - industries matrices.

Usage

```
hoover_curve(mat, pop, plot = TRUE, pdf = FALSE, pdf_location = NULL)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
pop	A vector of population regional count
plot	Logical; shall the curve be automatically plotted? Defaults to TRUE. If set to TRUE, the function will return x y coordinates that you can latter use to plot and customize the curve.
pdf	Logical; shall a pdf be saved? Defaults to FALSE. If set to TRUE, a pdf with all will be compiled and saved to R's temp dir if no 'pdf_location' is specified.
pdf_location	Output location of pdf file

Value

If 'plot = FALSE', a list containing the cumulative distribution of population shares ('cum.reg') and industry shares ('cum.out') is returned. If 'plot = TRUE', no return value is specified.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. (1936) The Measurement of Industrial Localization, *The Review of Economics and Statistics* **18** (1): 162-171

See Also

[hoover_gini](#), [locational_gini](#), [locational_gini_curve](#), [lorenz_curve](#), [gini](#)

Examples

```
## generate vectors of industrial and population count
ind <- c(0, 10, 10, 30, 50)
pop <- c(10, 15, 20, 25, 30)

## run the function (30% of the population produces 50% of the industrial output)
hoover_curve (ind, pop)
hoover_curve (ind, pop, pdf = FALSE)
hoover_curve (ind, pop, plot = FALSE)

## generate a region - industry matrix
mat = matrix (
  c (0, 10, 0, 0,
    0, 15, 0, 0,
    0, 20, 0, 0,
    0, 25, 0, 1,
    0, 30, 1, 1), ncol = 4, byrow = TRUE)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
hoover_curve (mat, pop)
hoover_curve (mat, pop, plot = FALSE)

## run the function by aggregating all industries
hoover_curve (rowSums(mat), pop)
hoover_curve (rowSums(mat), pop, plot = FALSE)

## run the function for industry #1 only
hoover_curve (mat[,1], pop)
hoover_curve (mat[,1], pop, plot = FALSE)

## run the function for industry #2 only (perfectly proportional to population)
hoover_curve (mat[,2], pop)
hoover_curve (mat[,2], pop, plot = FALSE)

## run the function for industry #3 only (30% of the pop. produces 100% of the output)
hoover_curve (mat[,3], pop)
hoover_curve (mat[,3], pop, plot = FALSE)

## run the function for industry #4 only (55% of the pop. produces 100% of the output)
hoover_curve (mat[,4], pop)
hoover_curve (mat[,4], pop, plot = FALSE)

## Compare the distribution of the #industries
oldpar <- par(mfrow = c(2, 2)) # Save the current graphical parameter settings
hoover_curve (mat[,1], pop)
hoover_curve (mat[,2], pop)
```

```
hoover_curve (mat[,3], pop)
hoover_curve (mat[,4], pop)
par(oldpar) # Reset the graphical parameters to their original values

## Save output as pdf
hoover_curve (mat, pop, pdf = TRUE)

## To specify an output directory for the pdf,
## specify 'pdf_location', for instance as '/Users/jones/hoover_curve.pdf'
## hoover_curve(mat, pop, pdf = TRUE, pdf_location = '/Users/jones/hoover_curve.pdf')
```

hoover_gini

Compute the Hoover Gini

Description

This function computes the Hoover Gini, named after Hedgar Hoover. The Hoover index is a measure of spatial inequality. It ranges from 0 (perfect equality) to 1 (perfect inequality) and is calculated from the Hoover curve associated with a given distribution of population, industries or technologies and a reference category. In this sense, it is closely related to the Gini coefficient and the Hoover index. The numerator is given by the area between the Hoover curve of the distribution and the uniform distribution line (45 degrees line). The denominator is the area under the uniform distribution line (the lower triangle).

Usage

```
hoover_gini(mat, pop)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
pop	A vector of population regional count

Value

The Hoover Gini value(s). If the input matrix has a single column, the function returns a numeric value representing the Hoover Gini index. If the input matrix has multiple columns, the function returns a data frame with two columns: "Industry" (names of the industries) and "hoover_gini" (corresponding Hoover Gini values).

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. (1936) The Measurement of Industrial Localization, *The Review of Economics and Statistics* **18** (1): 162-171

See Also

[hoover_curve](#), [locational_gini](#), [locational_gini_curve](#), [lorenz_curve](#), [gini](#)

Examples

```
## generate vectors of industrial and population count
ind <- c(0, 10, 10, 30, 50)
pop <- c(10, 15, 20, 25, 30)

## run the function (30% of the population produces 50% of the industrial output)
hoover_gini(ind, pop)

## generate a region - industry matrix
mat <- matrix(
  c(
    0, 10, 0, 0,
    0, 15, 0, 0,
    0, 20, 0, 0,
    0, 25, 0, 1,
    0, 30, 1, 1
  ),
  ncol = 4, byrow = TRUE
)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
hoover_gini(mat, pop)

## run the function by aggregating all industries
hoover_gini(rowSums(mat), pop)

## run the function for industry #1 only
hoover_gini(mat[, 1], pop)

## run the function for industry #2 only (perfectly proportional to population)
hoover_gini(mat[, 2], pop)

## run the function for industry #3 only (30% of the pop. produces 100% of the output)
hoover_gini(mat[, 3], pop)

## run the function for industry #4 only (55% of the pop. produces 100% of the output)
hoover_gini(mat[, 4], pop)
```

 hoover_index

 Compute the Hoover index

Description

This function computes the Hoover index, named after Hedgar Hoover. The Hoover index is a measure of spatial inequality. It ranges from 0 (perfect equality) to 100 (perfect inequality) and is calculated from the Lorenz curve associated with a given distribution of population, industries or technologies. In this sense, it is closely related to the Gini coefficient. The Hoover index represents the maximum vertical distance between the Lorenz curve and the 45 degree line of perfect spatial equality. It indicates the proportion of industries, jobs, or population needed to be transferred from the top to the bottom of the distribution to achieve perfect spatial equality. The Hoover index is also known as the Robin Hood index in studies of income inequality.

Computation of the Hoover index: $H = 1/2 \sum_{i=1}^N \left| \frac{E_i}{E_{total}} - \frac{A_i}{A_{total}} \right|$

Usage

```
hoover_index(mat, pop)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
pop	A vector of population regional count; if this argument is missing an equal distribution of the reference group will be assumed.

Value

The Hoover index value(s) as either a numeric value or a data frame with two columns: "Industry" (names of the industries) and "hoover_index" (corresponding Hoover index values).

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. (1936) The Measurement of Industrial Localization, *The Review of Economics and Statistics* **18** (1): 162-171

See Also

[hoover_curve](#), [hoover_gini](#), [locational_gini](#), [locational_gini_curve](#), [lorenz_curve](#), [gini](#)

Examples

```

## generate vectors of industrial and population count
ind <- c(0, 10, 10, 30, 50)
pop <- c(10, 15, 20, 25, 30)

## run the function (30% of the population produces 50% of the industrial output)
hoover_index(ind, pop)

## generate a region - industry matrix
mat <- matrix(
  c(
    0, 10, 0, 0,
    0, 15, 0, 0,
    0, 20, 0, 0,
    0, 25, 0, 1,
    0, 30, 1, 1
  ),
  ncol = 4, byrow = TRUE
)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
hoover_index(mat, pop)

## run the function by aggregating all industries
hoover_index(rowSums(mat), pop)

## run the function for industry #1 only
hoover_index(mat[, 1], pop)

## run the function for industry #2 only (perfectly proportional to population)
hoover_index(mat[, 2], pop)

## run the function for industry #3 only (30% of the pop. produces 100% of the output)
hoover_index(mat[, 3], pop)

## run the function for industry #4 only (55% of the pop. produces 100% of the output)
hoover_index(mat[, 4], pop)

```

inv_norm_ubiquity	<i>Compute a measure of complexity from the inverse of the normalized ubiquity of industries</i>
-------------------	--

Description

This function computes a measure of complexity from the inverse of the normalized ubiquity of industries. We divide the logarithm of the total count (employment, number of firms, number of patents, ...) in an industry by its ubiquity. Ubiquity is given by the number of regions in which an industry can be found (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
inv_norm_ubiquity(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A vector of complexity values computed from the inverse of the normalized ubiquity of industries.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[diversity](#), [location_quotient](#), [ubiquity](#), [tci](#), [mort](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
inv_norm_ubiquity(mat)
```

kci	<i>Compute an index of knowledge complexity of regions using the eigenvector method</i>
-----	---

Description

This function computes an index of knowledge complexity of regions using the eigenvector method from regions - industries (incidence) matrices. Technically, the function returns the eigenvector associated with the second largest eigenvalue of the projected region - region matrix.

Usage

```
kci(mat, rca = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
rca	Logical; should the index of relative comparative advantage (RCA - also referred to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Value

A vector representing the index of knowledge complexity of regions computed using the eigenvector method.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[location_quotient](#), [ubiquity](#), [diversity](#), [morc](#), [tci](#), [mort](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
kci(mat, rca = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
kci(mat)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
```

```
products <- c("P1", "P2", "P3", "P4", "P2", "P3", "P4", "P4")
my_data <- data.frame(countries, products)
my_data$freq <- 1
mat <- get_matrix(my_data)

## run the function
kci(mat)
```

krugman_index

Compute the Krugman index from regions - industries matrices

Description

This function computes the Krugman index from regions - industries matrices. The higher the coefficient, the greater the regional specialization. This index is often referred to as the Krugman specialisation index and measures the distance between the distributions of industry shares in a region and at a more aggregated level (country for instance).

Usage

```
krugman_index(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A vector representing the Krugman index of regional specialization computed from the regions - industries matrix.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Krugman P. (1991) *Geography and Trade*, MIT Press, Cambridge

See Also

[location_quotient_avg](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
krugman_index(mat)
```

locational_gini	<i>Compute the locational Gini coefficient from regions - industries matrices</i>
-----------------	---

Description

This function computes the locational Gini coefficient as proposed by Krugman from regions - industries matrices. The higher the coefficient (theoretical limit = 0.5), the greater the industrial concentration. The locational Gini of an industry that is not localized at all (perfectly spread out) in proportion to overall employment would be 0.

Usage

```
locational_gini(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A data frame with two columns: "Industry" and "Loc_gini". The "Industry" column contains the names of the industries, and the "Loc_gini" column contains the locational Gini coefficient computed for each industry from the regions - industries matrix.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Krugman P. (1991) *Geography and Trade*, MIT Press, Cambridge (chapter 2 - p.56)

See Also

[hoover_gini](#), [locational_gini_curve](#), [hoover_curve](#), [lorenz_curve](#), [gini](#)

Examples

```
## generate a region - industry matrix
mat <- matrix(
  c(
    100, 0, 0, 0, 0,
    0, 15, 5, 70, 10,
    0, 20, 10, 20, 50,
    0, 25, 30, 5, 40,
    0, 40, 55, 5, 0
  ),
  ncol = 5, byrow = TRUE
)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4", "I5")

## run the function
locational_gini(mat)
```

locational_gini_curve *Plot a locational Gini curve from regions - industries matrices*

Description

This function plots a locational Gini curve following Krugman from regions - industries matrices.

Usage

```
locational_gini_curve(mat, pdf = FALSE, pdf_location = NULL)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
pdf	Logical; shall a pdf be saved? Defaults to FALSE. If set to TRUE, a pdf with all will be compiled and saved to R's temp dir if no 'pdf_location' is specified.
pdf_location	Output location of pdf file

Value

No return value, produces a plot or pdf.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Krugman P. (1991) *Geography and Trade*, MIT Press, Cambridge (chapter 2 - p.56)

See Also

[hoover_gini](#), [locational_gini](#), [hoover_curve](#), [lorenz_curve](#), [gini](#)

Examples

```
## generate a region - industry matrix
mat <- matrix(
  c(
    100, 0, 0, 0, 0,
    0, 15, 5, 70, 10,
    0, 20, 10, 20, 50,
    0, 25, 30, 5, 40,
    0, 40, 55, 5, 0
  ),
  ncol = 5, byrow = TRUE
)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4", "I5")

## run the function (shows industry #5)
locational_gini_curve(mat, pdf = FALSE)
locational_gini_curve(mat, pdf = FALSE)

## Save output as pdf
locational_gini_curve(mat, pdf = TRUE)

## To specify an output directory for the pdf,
## specify 'pdf_location', for instance as '/Users/jones/locational_gini_curve.pdf'
## locational_gini_curve(mat, pdf = TRUE, pdf_location = '/Users/jones/locational_gini_curve.pdf')
```

location_quotient

Compute location quotients from regions - industries matrices

Description

This function computes location quotients from (incidence) regions - industries matrices. The numerator is the share of a given industry in a given region. The denominator is the share of a this industry in a larger economy (overall country for instance). This index is also referred to as the index of Revealed Comparative Advantage (RCA) following Ballasa (1965), or the Hoover-Balassa index.

Usage

```
location_quotient(mat, binary = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
binary	Logical; shall the returned output be a dichotomized version (0/1) of the location quotient? Defaults to FALSE (the full values of the location quotient will be returned), but can be set to TRUE (location quotient values above 1 will be set to 1 & location quotient values below 1 will be set to 0)

Value

A matrix of location quotients computed from the regions - industries matrix. If the 'binary' parameter is set to TRUE, the returned matrix will contain binary values (0/1) representing the location quotient. If 'binary' is set to FALSE, the full values of the location quotient will be returned.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123.

See Also

[rca](#)

Examples

```
## generate a region - industry matrix
mat <- matrix(
  c(
    100, 0, 0, 0, 0,
    0, 15, 5, 70, 10,
    0, 20, 10, 20, 50,
    0, 25, 30, 5, 40,
    0, 40, 55, 5, 0
  ),
  ncol = 5, byrow = TRUE
)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4", "I5")

## run the function
location_quotient(mat)
location_quotient(mat, binary = TRUE)
```

location_quotient_avg *Compute average location quotients of regions from regions - industries matrices*

Description

This function computes the average location quotients of regions from (incidence) regions - industries matrices. This index is also referred to as the *coefficient of specialization* (Hoover and Giarratani, 1985).

Usage

```
location_quotient_avg(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A vector of average location quotients computed for each region from the regions - industries matrix. The average location quotient represents the degree of specialization of each region in different industries.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. and Giarratani, F. (1985) *An Introduction to Regional Economics*. 3rd edition. New York: Alfred A. Knopf

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

See Also

[location_quotient](#), [hachman](#)

Examples

```
## generate a region - industry matrix
mat <- matrix(
  c(
    100, 0, 0, 0, 0,
    0, 15, 5, 70, 10,
    0, 20, 10, 20, 50,
```



```

      0, 25, 30, 5, 40,
      0, 40, 55, 5, 0
    ),
    ncol = 5, byrow = TRUE
  )
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4", "I5")

## run the function
location_quotient_avg(mat)

```

lorenz_curve

*Plot a Lorenz curve from regional industrial counts***Description**

This function plots a Lorenz curve from regional industrial counts. This curve gives an indication of the unequal distribution of an industry across regions.

Usage

```
lorenz_curve(mat, plot = TRUE, pdf = TRUE, pdf_location = NULL)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
plot	Logical; shall the curve be automatically plotted? Defaults to TRUE. If set to TRUE, the function will return x y coordinates that you can later use to plot and customize the curve.
pdf	Logical; shall a pdf be saved? Defaults to FALSE. If set to TRUE, a pdf with all will be compiled and saved to R's temp dir if no 'pdf_location' is specified.
pdf_location	Output location of pdf file

Value

If 'plot = FALSE', the function returns a list with two components: - 'cum.reg': A vector of cumulative proportions of regions. - 'cum.out': A vector of cumulative proportions of industrial output. If 'plot = TRUE', the function generates a plot of the Lorenz curve and does not return a value.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Lorenz, M. O. (1905) Methods of measuring the concentration of wealth, *Publications of the American Statistical Association* **9**: 209–219

See Also

[hoover_gini](#), [locational_gini](#), [locational_gini_curve](#), [hoover_curve](#), [gini](#)

Examples

```
## generate vectors of industrial count
ind <- c(0, 10, 10, 30, 50)

## run the function
lorenz_curve (ind)
lorenz_curve (ind, plot = FALSE)

## generate a region - industry matrix
mat = matrix (
  c (0, 1, 0, 0,
    0, 1, 0, 0,
    0, 1, 0, 0,
    0, 1, 0, 1,
    0, 1, 1, 1), ncol = 4, byrow = TRUE)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
lorenz_curve (mat)
lorenz_curve (mat, plot = FALSE)

## run the function by aggregating all industries
lorenz_curve (rowSums(mat))
lorenz_curve (rowSums(mat), plot = FALSE)

## run the function for industry #1 only (perfect equality)
lorenz_curve (mat[,1])
lorenz_curve (mat[,1], plot = FALSE)

## run the function for industry #2 only (perfect equality)
lorenz_curve (mat[,2])
lorenz_curve (mat[,2], plot = FALSE)

## run the function for industry #3 only (perfect inequality)
lorenz_curve (mat[,3])
lorenz_curve (mat[,3], plot = FALSE)

## run the function for industry #4 only (top 40% produces 100% of the output)
lorenz_curve (mat[,4])
lorenz_curve (mat[,4], plot = FALSE)

## Compare the distribution of the #industries
oldpar <- par(mfrow = c(2, 2)) # Save the current graphical parameter settings
lorenz_curve (mat[,1])
lorenz_curve (mat[,2])
lorenz_curve (mat[,3])
lorenz_curve (mat[,4])
```

```
par(oldpar) # Reset the graphical parameters to their original values

## Save output as pdf
lorenz_curve (mat, pdf = TRUE)

## To specify an output directory for the pdf,
## specify 'pdf_location', for instance as '/Users/jones/lorenz_curve.pdf'
## lorenz_curve(mat, pdf = TRUE, pdf_location = '/Users/jones/lorenz_curve.pdf')
```

match_mat	<i>Re-arrange the dimension of a matrix based on the dimension of another matrix</i>
-----------	--

Description

This function e-arranges the dimension of a matrix based on the dimension of another matrix

Usage

```
match_mat(fill, dim, missing = TRUE)
```

Arguments

fill	A matrix that will be used to populate the matrix output
dim	A matrix that will be used to determine the dimensions of the matrix output
missing	Logical; Shall the cells of the non matching rows/columns set to NA? Default to TRUE but can be set to FALSE to set the cells of the non matching rows/columns to 0 instead.

Value

The matrix output with the dimensions rearranged based on the input 'dim' matrix.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

See Also

[location_quotient](#)

Examples

```
## generate a first region - industry matrix
set.seed(31)
mat1 <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat1) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c("I1", "I2", "I3", "I4")

## generate a second region - industry matrix
set.seed(31)
mat2 <- matrix(sample(0:1, 16, replace = TRUE), ncol = 4)
rownames(mat2) <- c("R1", "R2", "R3", "R5")
colnames(mat2) <- c("I1", "I2", "I3", "I4")

## run the function
match_mat(fill = mat1, dim = mat2)
match_mat(fill = mat2, dim = mat1)
match_mat(fill = mat2, dim = mat1, missing = FALSE)
```

modular_complexity *Compute a measure of modular complexity of patent documents*

Description

This function computes a measure of modular complexity of patent documents from technological classes - patents (incidence) matrices

Usage

```
modular_complexity(mat, sparse = FALSE, list = FALSE)
```

Arguments

mat	A bipartite adjacency matrix (can be a sparse matrix)
sparse	Logical; is the input matrix a sparse matrix? Defaults to FALSE, but can be set to TRUE if the input matrix is a sparse matrix
list	Logical; is the input a list? Defaults to FALSE (input = adjacency matrix), but can be set to TRUE if the input is an edge list

Value

A data frame with columns "patent" and "mod.comp" representing the patents and their corresponding modular complexity values.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Fleming, L. and Sorenson, O. (2001) Technology as a complex adaptive system: evidence from patent data, *Research Policy* **30**: 1019-1039

See Also

[ease_recombination](#), [tci](#), [mort](#)

Examples

```
## generate a technology - patent matrix
set.seed(31)
mat <- matrix(sample(0:1, 30, replace = TRUE), ncol = 5)
rownames(mat) <- c("T1", "T2", "T3", "T4", "T5", "T6")
colnames(mat) <- c("US1", "US2", "US3", "US4", "US5")

## run the function
modular_complexity(mat)

## generate a technology - patent sparse matrix
library(Matrix)

## run the function
smat <- Matrix(mat, sparse = TRUE)

modular_complexity(smat, sparse = TRUE)
## generate a regular data frame (list)
my_list <- get_list(mat)

## run the function
modular_complexity(my_list, list = TRUE)
```

modular_complexity_avg

Compute a measure of average modular complexity of technologies

Description

This function computes a measure of average modular complexity of technologies (average complexity of patent documents in a given technological class) from technological classes - patents (incidence) matrices

Usage

```
modular_complexity_avg(mat, sparse = FALSE, list = FALSE)
```

Arguments

mat	A bipartite adjacency matrix (can be a sparse matrix)
sparse	Logical; is the input matrix a sparse matrix? Defaults to FALSE, but can be set to TRUE if the input matrix is a sparse matrix
list	Logical; is the input a list? Defaults to FALSE (input = adjacency matrix), but can be set to TRUE if the input is an edge list

Value

A data frame with columns "tech" and "avg.mod.comp" representing the technologies and their corresponding average modular complexity values.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Fleming, L. and Sorenson, O. (2001) Technology as a complex adaptive system: evidence from patent data, *Research Policy* **30**: 1019-1039

See Also

[ease_recombination](#), [tci](#), [mort](#)

Examples

```
## generate a technology - patent matrix
set.seed(31)
mat <- matrix(sample(0:1, 30, replace = TRUE), ncol = 5)
rownames(mat) <- c("T1", "T2", "T3", "T4", "T5", "T6")
colnames(mat) <- c("US1", "US2", "US3", "US4", "US5")

## run the function
modular_complexity_avg(mat)

## generate a technology - patent sparse matrix
library(Matrix)

## run the function
smat <- Matrix(mat, sparse = TRUE)

modular_complexity_avg(smat, sparse = TRUE)
## generate a regular data frame (list)
my_list <- get_list(mat)

## run the function
modular_complexity_avg(my_list, list = TRUE)
```

morc	<i>Compute an index of knowledge complexity of regions using the method of reflection</i>
------	---

Description

This function computes an index of knowledge complexity of regions using the method of reflection from regions - industries (incidence) matrices. The index has been developed by Hidalgo and Hausmann (2009) for country - product matrices and adapted by Balland and Rigby (2016) to city - technology matrices.

Usage

```
morc(mat, rca = FALSE, steps = 20)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
rca	Logical; should the index of relative comparative advantage (RCA - also referred to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed
steps	Number of iteration steps. Defaults to 20, but can be set to 0 to give diversity (number of industry in which a region has a RCA), to 1 to give the average ubiquity of the industries in which a region has a RCA, to 2 to give the average diversity of regions that have similar industrial structures, or to any other number of steps < or = to 22. Note that above steps = 2 the index will be rescaled from 0 (minimum relative complexity) to 100 (maximum relative complexity).

Value

If 'steps' is set to 0, the function returns a numeric vector representing the diversification of regions. Otherwise, it returns

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[location_quotient](#), [ubiquity](#), [diversity](#), [kci](#), [tci](#), [mort](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
morc(mat, rca = TRUE)
morc(mat, rca = TRUE, steps = 0)
morc(mat, rca = TRUE, steps = 1)
morc(mat, rca = TRUE, steps = 2)

## generate a region - industry matrix in which cells represent the presence/absence of an RCA
set.seed(32)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
morc(mat)
morc(mat, steps = 0)
morc(mat, steps = 1)
morc(mat, steps = 2)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
products <- c("P1", "P2", "P3", "P4", "P2", "P3", "P4", "P4")
my_data <- data.frame(countries, products)
my_data$freq <- 1
mat <- get_matrix(my_data)

## run the function
morc(mat)
morc(mat, steps = 0)
morc(mat, steps = 1)
morc(mat, steps = 2)
```

mort

Compute an index of knowledge complexity of industries using the method of reflection

Description

This function computes an index of knowledge complexity of industries using the method of reflection from regions - industries (incidence) matrices. The index has been developed by Hidalgo and

Hausmann (2009) for country - product matrices and adapted by Balland and Rigby (2016) to city - technology matrices.

Usage

```
mort(mat, rca = FALSE, steps = 19)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
rca	Logical; should the index of relative comparative advantage (RCA - also referred to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed
steps	Number of iteration steps. Defaults to 19, but can be set to 0 to give ubiquity (number of regions that have a RCA in an industry), to 1 to give the average diversity of the regions that have a RCA in this industry, to 2 to give the average ubiquity of technologies developed in the same regions, or to any other number of steps < or = to 21. Note that above steps = 2 the index will be rescaled from 0 (minimum relative complexity) to 100 (maximum relative complexity).

Value

If 'steps' is set to 0, the function returns a numeric vector representing the ubiquity (number of regions that have a relative comparative advantage) of industries. Otherwise, it returns a numeric vector representing the index of knowledge complexity of industries based on the specified number of iteration steps.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[location_quotient](#), [ubiquity](#), [diversity](#), [kci](#), [tci](#), [morc](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
```

```

colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
mort(mat, rca = TRUE)
mort(mat, rca = TRUE, steps = 0)
mort(mat, rca = TRUE, steps = 1)
mort(mat, rca = TRUE, steps = 2)

## generate a region - industry matrix in which cells represent the presence/absence of a rca
set.seed(32)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
mort(mat)
mort(mat, steps = 0)
mort(mat, steps = 1)
mort(mat, steps = 2)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
products <- c("P1", "P2", "P3", "P4", "P2", "P3", "P4", "P4")
my_data <- data.frame(countries, products)
my_data$freq <- 1
mat <- get_matrix(my_data)

## run the function
mort(mat)
mort(mat, steps = 0)
mort(mat, steps = 1)
mort(mat, steps = 2)

```

norm_ubiquity

Compute a measure of complexity by normalizing ubiquity of industries

Description

This function computes a measure of complexity by normalizing ubiquity of industries. We divide the share of the total count (employment, number of firms, number of patents, ...) in an industry by its share of ubiquity. Ubiquity is given by the number of regions in which an industry can be found (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
norm_ubiquity(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A numeric vector representing the measure of complexity obtained by normalizing the ubiquity of industries. Each value in the vector corresponds to the normalized complexity score of an industry.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[diversity](#), [location_quotient](#), [ubiquity](#), [tci](#), [mort](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
norm_ubiquity(mat)
```

prody

Compute the prody index of industries from regions - industries matrices

Description

This function computes the prody index of industries from (incidence) regions - industries matrices, as proposed by Hausmann, Hwang & Rodrik (2007). The index gives an associated income level for each industry. It represents a weighted average of per-capita GDPs (but GDP can be replaced by R&D, education...), where the weights correspond to the revealed comparative advantage of each region in a given industry (or sector, technology, ...).

Usage

```
prody(mat, vec)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
vec	A vector that gives GDP, R&D, education or any other relevant regional attribute that will be used to compute the weighted average for each industry

Value

A numeric vector representing the prody index of industries. Each value in the vector corresponds to the associated income level for an industry.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123

Hausmann, R., Hwang, J. & Rodrik, D. (2007) What you export matters, *Journal of economic growth* **12**: 1-25.

See Also

[location_quotient](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## a vector of GDP of regions
vec <- c(5, 10, 15, 25, 50)
## run the function
prody(mat, vec)
```

rca

Compute an index of revealed comparative advantage (RCA) from regions - industries matrices

Description

This function computes an index of revealed comparative advantage (RCA) from (incidence) regions - industries matrices. The numerator is the share of a given industry in a given region. The denominator is the share of a this industry in a larger economy (overall country for instance). This index is also referred to as a location quotient, or the Hoover-Balassa index.

Usage

```
rca(mat, binary = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
binary	Logical; shall the returned output be a dichotomized version (0/1) of the RCA? Defaults to FALSE (the full values of the RCA will be returned), but can be set to TRUE (RCA above 1 will be set to 1 & RCA values below 1 will be set to 0)

Value

A matrix representing the index of revealed comparative advantage (RCA) or location quotient. Each cell in the matrix corresponds to the RCA value for a specific region and industry. If the 'binary' parameter is set to TRUE, the returned matrix will be dichotomized, with values above 1 set to 1 and values below 1 set to 0.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123.

See Also

[location_quotient](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
rca(mat)
rca(mat, binary = TRUE)
```

relatedness	<i>Compute the relatedness between entities (industries, technologies, ...) from their co-occurrence matrix</i>
-------------	---

Description

This function computes the relatedness between entities (industries, technologies, ...) from their co-occurrence (adjacency) matrix. Different normalization procedures are proposed following van Eck and Waltman (2009): association strength, cosine, Jaccard, and an adapted version of the association strength that we refer to as probability index.

Usage

```
relatedness(mat, method = "prob")
```

Arguments

mat	An adjacency matrix of co-occurrences between entities (industries, technologies, cities...)
method	Which normalization method should be used to compute relatedness? Defaults to "prob", but it can be "association", "cosine" or "Jaccard"

Value

A matrix representing the relatedness between entities (industries, technologies, etc.) based on their co-occurrence matrix. The specific method of normalization used is determined by the 'method' parameter, which can be "prob" (probability index), "association" (association strength), "cosine" (cosine similarity), or "jaccard" (Jaccard index).

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Joan Crespo <J.Crespo@uu.nl>
Mathieu Steijn <M.P.A.Steijn@uu.nl>

References

- van Eck, N.J. and Waltman, L. (2009) How to normalize cooccurrence data? An analysis of some well-known similarity measures, *Journal of the American Society for Information Science and Technology* **60** (8): 1635-1651
- Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114
- Hidalgo, C.A., Klinger, B., Barabasi, A. and Hausmann, R. (2007) The product space conditions the development of nations, *Science* **317**: 482-487
- Balland, P.A. (2016) Relatedness and the Geography of Innovation, in: R. Shearmur, C. Carrincazeaux and D. Doloreux (eds) *Handbook on the Geographies of Innovation*. Northampton, MA: Edward Elgar
- Steijn, M.P.A. (2017) Improvement on the association strength: implementing probability measures based on combinations without repetition, *Working Paper, Utrecht University*

See Also

[relatedness_density](#), [co_occurrence](#)

Examples

```
## generate an industry - industry matrix in which cells give the number of co-occurrences
## between two industries
set.seed(31)
mat <- matrix(sample(0:10, 36, replace = TRUE), ncol = 6)
mat[lower.tri(mat, diag = TRUE)] <- t(mat)[lower.tri(t(mat), diag = TRUE)]
rownames(mat) <- c("I1", "I2", "I3", "I4", "I5", "I6")
colnames(mat) <- c("I1", "I2", "I3", "I4", "I5", "I6")

## run the function
relatedness(mat)
relatedness(mat, method = "association")
relatedness(mat, method = "cosine")
relatedness(mat, method = "jaccard")
```

relatedness_density	<i>Compute the relatedness density between regions and industries from regions - industries matrices and industries - industries matrices</i>
---------------------	---

Description

This function computes the relatedness density between regions and industries from regions - industries (incidence) matrices and industries - industries (adjacency) matrices

Usage

```
relatedness_density(mat, relatedness)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Value

A matrix representing the relatedness density between regions and industries. The values in the matrix indicate the share of industries related to each industry in each region, scaled from 0 to 100. Rows represent regions and columns represent industries.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[relatedness](#), [co_occurrence](#)

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1, 16, replace = TRUE), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
  diag = TRUE
)]
rownames(relatedness) <- c("I1", "I2", "I3", "I4")
colnames(relatedness) <- c("I1", "I2", "I3", "I4")

## run the function
relatedness_density(mat, relatedness)
```

relatedness_density_ext

Compute the relatedness density between regions and industries that are not part of the regional portfolio from regions - industries matrices and industries - industries matrices

Description

This function computes the relatedness density between regions and industries that are not part of the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices

Usage

```
relatedness_density_ext(mat, relatedness)
```


Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Value

A matrix representing the relatedness density between regions and industries that are not part of the regional portfolio. The values in the matrix indicate the share of industries related to each industry in each region, scaled from 0 to 100. Rows represent regions and columns represent industries. Industries that are part of the regional portfolio are assigned NA.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[relatedness_co_occurrence](#)

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1, 16, replace = TRUE), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
  diag = TRUE
)]
rownames(relatedness) <- c("I1", "I2", "I3", "I4")
colnames(relatedness) <- c("I1", "I2", "I3", "I4")

## run the function
relatedness_density_ext(mat, relatedness)
```

relatedness_density_ext_avg

Compute the average relatedness density of regions to industries that are not part of the regional portfolio from regions - industries matrices and industries - industries matrices

Description

This function computes the average relatedness density of regions to industries that are not part of the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices. This is the technological flexibility indicator proposed by Balland et al. (2015).

Usage

```
relatedness_density_ext_avg(mat, relatedness)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Value

A vector representing the average relatedness density of regions to industries that are not part of the regional portfolio. The values in the vector indicate the average relatedness density for each region, rounded to the nearest integer.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Balland P.A., Rigby, D., and Boschma, R. (2015) The Technological Resilience of U.S. Cities, *Cambridge Journal of Regions, Economy and Society*, **8** (2): 167-184

See Also

[relatedness](#), [relatedness_density](#), [relatedness_density_ext](#), [relatedness_density_int](#), [relatedness_density_int_avg](#), [relatedness_density_ext_avg](#)

Examples

```

## generate a region - industry matrix in which cells represent the presence/absence
## of a RCA
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1, 16, replace = TRUE), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
  diag = TRUE
)]
rownames(relatedness) <- c("I1", "I2", "I3", "I4")
colnames(relatedness) <- c("I1", "I2", "I3", "I4")

## run the function
relatedness_density_ext_avg(mat, relatedness)

```

relatedness_density_int

Compute the relatedness density between regions and industries that are part of the regional portfolio from regions - industries matrices and industries - industries matrices

Description

This function computes the relatedness density between regions and industries that are part of the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices

Usage

```
relatedness_density_int(mat, relatedness)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Value

A matrix representing the relatedness density between regions and industries that are part of the regional portfolio. The values in the matrix indicate the relatedness density for each region and industry, scaled from 0 to 100.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

[relatedness, co_occurrence](#)

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence
## of a RCA
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1, 16, replace = TRUE), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
  diag = TRUE
)]
rownames(relatedness) <- c("I1", "I2", "I3", "I4")
colnames(relatedness) <- c("I1", "I2", "I3", "I4")

## run the function
relatedness_density_int(mat, relatedness)
```

relatedness_density_int_avg

*Compute the average relatedness density within the regional portfolio
from regions - industries matrices and industries - industries matrices*

Description

This function computes the average relatedness density within the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices. This is a measure of the technological coherence of the regional industrial structure.

Usage

```
relatedness_density_int_avg(mat, relatedness)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Value

A vector representing the average relatedness density within the regional portfolio. The values in the vector indicate the average relatedness density for each region, scaled from 0 to 100.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Balland P.A., Rigby, D., and Boschma, R. (2015) The Technological Resilience of U.S. Cities, *Cambridge Journal of Regions, Economy and Society*, **8** (2): 167-184

See Also

[relatedness](#), [relatedness_density](#), [relatedness_density_ext](#), [relatedness_density_int](#), [relatedness_density_ext_avg](#),

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence
## of a RCA
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1, 16, replace = TRUE), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
  diag = TRUE
)]
rownames(relatedness) <- c("I1", "I2", "I3", "I4")
colnames(relatedness) <- c("I1", "I2", "I3", "I4")
```

```
## run the function
relatedness_density_int_avg(mat, relatedness)
```

spec_coeff	<i>Compute the Hoover coefficient of specialization from regions - industries matrices</i>
------------	--

Description

This function computes the Hoover coefficient of specialization from regions - industries matrices. The higher the coefficient, the greater the regional specialization. This index is closely related to the Krugman specialisation index.

Usage

```
spec_coeff(mat)
```

Arguments

mat An incidence matrix with regions in rows and industries in columns

Value

A vector representing the Hoover coefficient of specialization for each region. The values in the vector indicate the degree of regional specialization, with higher values indicating greater specialization.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. and Giarratani, F. (1985) *An Introduction to Regional Economics*. 3rd edition. New York: Alfred A. Knopf (see table 9-4 in particular)

See Also

[krugman_index](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
spec_coeff(mat)
```

tci	<i>Compute an index of knowledge complexity of industries using the eigenvector method</i>
-----	--

Description

This function computes an index of knowledge complexity of industries using the eigenvector method from regions - industries (incidence) matrices. Technically, the function returns the eigenvector associated with the second largest eigenvalue of the projected industry - industry matrix.

Usage

```
tci(mat, rca = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
rca	Logical; should the index of relative comparative advantage (RCA - also referred to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Value

A numeric vector representing the index of knowledge complexity of industries. The vector contains the values of the eigenvector associated with the second largest eigenvalue of the projected industry - industry matrix.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[location_quotient](#), [ubiquity](#), [diversity](#), [morc](#), [kci](#), [mort](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
tci(mat, rca = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a rca
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
tci(mat)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
products <- c("P1", "P2", "P3", "P4", "P2", "P3", "P4", "P4")
my_data <- data.frame(countries, products)
my_data$freq <- 1
mat <- get_matrix(my_data)

## run the function
tci(mat)
```

 ubiquity

Compute a simple measure of ubiquity of industries

Description

This function computes a simple measure of ubiquity of industries by counting the number of regions in which an industry can be found (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
ubiquity(mat, rca = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
rca	Logical; should the index of relative comparative advantage (RCA - also referred to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Value

A numeric vector representing the measure of ubiquity of industries. Each element of the vector corresponds to the number of regions in which an industry can be found (location quotient > 1).

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

[diversity](#) [location_quotient](#)

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
ubiquity(mat, rca = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a rca
set.seed(31)
mat <- matrix(sample(0:1, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## run the function
ubiquity(mat)
```

weighted_avg

Compute a weighted average of regions or industries from regions - industries matrices

Description

This function computes a weighted average of regions or industries from (incidence) regions - industries matrices.

Usage

```
weighted_avg(mat, vec, reg = TRUE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
vec	A vector that will be used to compute the weighted average for each industry/region
reg	Logical; Shall the weighted average for regions be returned? Default to TRUE (requires a vector of industry value) but can be set to FALSE (requires a vector of region value) if the weighted average for industries should be returned

Value

A numeric vector representing the weighted average of regions or industries, depending on the value of the 'reg' argument. If 'reg = TRUE', the weighted average for regions is returned; if 'reg = FALSE', the weighted average for industries is returned.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

See Also

[location_quotient](#)

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100, 20, replace = TRUE), ncol = 4)
rownames(mat) <- c("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c("I1", "I2", "I3", "I4")

## a vector for regions will be used to computed the weighted average of industries
vec <- c(5, 10, 15, 25, 50)
## run the function
weighted_avg(mat, vec, reg = FALSE)

## a vector for industries will be used to computed the weighted average of regions
vec <- c(5, 10, 15, 25)
## run the function
weighted_avg(mat, vec, reg = TRUE)
```

Description

This function computes the z-score between pairs of technologies from a patent-technology incidence matrix. The z-score is a measure to analyze the co-occurrence of technologies in patent documents (i.e. knowledge combination). It compares the observed number of co-occurrences to what would be expected under the hypothesis that combination is random. A positive z-score indicates a typical co-occurrence which has occurred multiple times before. In contrast, a negative z-score indicates an atypical co-occurrence. The z-score has been used to estimate the degree of novelty of patents (Kim 2016), scientific publications (Uzzi et al. 2013) or the relatedness between industries (Teece et al. 1994).

Usage

```
z_score(mat)
```

Arguments

mat	A patent-technology incidence matrix with patents in rows and technologies in columns
-----	---

Value

A matrix of z-scores representing the co-occurrence of technologies in the input incidence matrix. The z-score measures the deviation of the observed co-occurrence from the expected co-occurrence under the assumption of random combination. Positive z-scores indicate typical co-occurrences, while negative z-scores indicate atypical co-occurrences.

Author(s)

Lars Mewes <mewes@wigeo.uni-hannover.de>

References

- Kim, D., Cerigo, D. B., Jeong, H., and Youn, H. (2016). Technological novelty profile and invention's future impact. *EPJ Data Science*, **5** (1):1–15
- Teece, D. J., Rumelt, R., Dosi, G., and Winter, S. (1994). Understanding corporate coherence. Theory and evidence. *Journal of Economic Behavior and Organization*, **23** (1):1–30
- Uzzi, B., Mukherjee, S., Stringer, M., and Jones, B. (2013). Atypical Combinations and Scientific Impact. *Science*, **342** (6157):468–472

See Also

[relatedness_density](#), [co_occurrence](#)

Examples

```
## Generate a toy incidence matrix
set.seed(2210)
```

```
techs <- paste0("T", seq(1, 5))
techs <- sample(techs, 50, replace = TRUE)
patents <- paste0("P", seq(1, 20))
patents <- sort(sample(patents, 50, replace = TRUE))
my_data <- data.frame(patents, techs)
my_dat <- unique(my_data)
mat <- as.matrix(table(my_data$patents, my_data$techs))

## run the function
z_score(mat)
```

Index

- * **complexity**
 - ease_recombination, 5
 - kci, 33
 - modular_complexity, 44
 - modular_complexity_avg, 45
 - morc, 47
 - mort, 48
 - norm_ubiquity, 50
 - prody, 51
 - tci, 63
 - * **concentration**
 - gini, 16
 - herfindahl, 26
 - hoover_curve, 27
 - hoover_gini, 29
 - hoover_index, 31
 - location_quotient, 38
 - location_quotient_avg, 40
 - locational_gini, 36
 - locational_gini_curve, 37
 - lorenz_curve, 41
 - prody, 51
 - * **data.management**
 - get_list, 14
 - get_matrix, 15
 - match_mat, 43
 - * **diversification**
 - entry_list, 7
 - entry_mat, 9
 - exit_list, 10
 - exit_mat, 12
 - * **diversity**
 - diversity, 4
 - entropy, 6
 - expy, 13
 - relatedness_density_ext_avg, 58
 - weighted_avg, 65
 - * **growth**
 - growth_ind, 17
 - growth_list, 19
 - growth_list_ind, 20
 - growth_list_reg, 21
 - growth_mat, 23
 - growth_reg, 24
 - * **inequality**
 - gini, 16
 - hoover_curve, 27
 - hoover_gini, 29
 - hoover_index, 31
 - locational_gini, 36
 - locational_gini_curve, 37
 - lorenz_curve, 41
 - * **relatedness**
 - co_occurrence, 3
 - relatedness, 53
 - relatedness_density, 55
 - relatedness_density_ext, 56
 - relatedness_density_ext_avg, 58
 - relatedness_density_int, 59
 - relatedness_density_int_avg, 60
 - z_score, 66
 - * **specialization**
 - expy, 13
 - hachman, 25
 - herfindahl, 26
 - krugman_index, 35
 - location_quotient, 38
 - location_quotient_avg, 40
 - rca, 52
 - relatedness_density_int_avg, 60
 - spec_coeff, 62
 - weighted_avg, 65
 - * **ubiquity**
 - inv_norm_ubiquity, 32
 - norm_ubiquity, 50
 - ubiquity, 64
- co_occurrence, 3, 54, 56, 57, 60, 67

- diversity, [4](#), [7](#), [33](#), [34](#), [48](#), [49](#), [51](#), [63](#), [65](#)
- ease_recombination, [5](#), [45](#), [46](#)
- entropy, [6](#)
- entry_list, [7](#), [10–12](#), [18](#), [21–23](#), [25](#)
- entry_mat, [9](#)
- exit_list, [8](#), [10](#), [10](#), [12](#), [18](#), [19](#), [21–23](#), [25](#)
- exit_mat, [12](#)
- expy, [13](#)

- get_list, [14](#), [15](#)
- get_matrix, [14](#), [15](#)
- gini, [16](#), [28](#), [30](#), [31](#), [36](#), [38](#), [42](#)
- growth_ind, [17](#)
- growth_list, [8](#), [10](#), [12](#), [19](#), [21–23](#), [25](#)
- growth_list_ind, [20](#)
- growth_list_reg, [21](#)
- growth_mat, [23](#)
- growth_reg, [24](#)

- hachman, [25](#), [40](#)
- herfindahl, [26](#)
- hoover_curve, [17](#), [27](#), [30](#), [31](#), [36](#), [38](#), [42](#)
- hoover_gini, [17](#), [28](#), [29](#), [31](#), [36](#), [38](#), [42](#)
- hoover_index, [31](#)

- inv_norm_ubiquity, [32](#)

- kci, [33](#), [48](#), [49](#), [63](#)
- krugman_index, [26](#), [35](#), [62](#)

- location_quotient, [5](#), [14](#), [33](#), [34](#), [38](#), [40](#), [43](#),
[48](#), [49](#), [51–53](#), [63](#), [65](#), [66](#)
- location_quotient_avg, [26](#), [35](#), [40](#)
- locational_gini, [17](#), [28](#), [30](#), [31](#), [36](#), [38](#), [42](#)
- locational_gini_curve, [17](#), [28](#), [30](#), [31](#), [36](#),
[37](#), [42](#)
- lorenz_curve, [17](#), [28](#), [30](#), [31](#), [36](#), [38](#), [41](#)

- match_mat, [43](#)
- modular_complexity, [6](#), [44](#)
- modular_complexity_avg, [45](#)
- morc, [34](#), [47](#), [49](#), [63](#)
- mort, [6](#), [33](#), [34](#), [45](#), [46](#), [48](#), [48](#), [51](#), [63](#)

- norm_ubiquity, [50](#)

- prody, [51](#)

- rca, [39](#), [52](#)

- relatedness, [3](#), [53](#), [56–58](#), [60](#), [61](#)
- relatedness_density, [3](#), [54](#), [55](#), [58](#), [61](#), [67](#)
- relatedness_density_ext, [56](#), [58](#), [61](#)
- relatedness_density_ext_avg, [58](#), [58](#), [61](#)
- relatedness_density_int, [58](#), [59](#), [61](#)
- relatedness_density_int_avg, [58](#), [60](#)

- spec_coeff, [62](#)

- tci, [6](#), [33](#), [34](#), [45](#), [46](#), [48](#), [49](#), [51](#), [63](#)

- ubiquity, [5](#), [33](#), [34](#), [48](#), [49](#), [51](#), [63](#), [64](#)

- weighted_avg, [65](#)

- z_score, [66](#)