Package: flows (via r-universe)

November 14, 2026

Type Package

Title Flow Matrix Selection with Statistics and Visualisation

Version 2.0.0

Description The analysis and representation of flows often presuppose a selection to facilitate interpretation. Various methods have been proposed for selecting flows, one of the most widely used being based on major flows: it selects only the most important flows, absolute or relative, on a local or global scale. These methods often highlight hierarchies between locations, but the loss of information caused by selection is rarely taken into account. We propose statistical indicators to assess the loss of information and the characteristics of selected flows. We provide functions that select flows (main, dominant or major flows), provide statistics on selections and offer visualizations in the form of maps and graphs. See Beauguitte et al. (2015) <doi:10.4000/netcom.2134> for more details.

License GPL-3

Depends R (>= 3.6.0)

Imports igraph, graphics, mapsf, maplegend, utils, stats

Suggests sf, knitr, rmarkdown

URL https://codeberg.org/riatelab/flows

BugReports https://codeberg.org/riatelab/flows/issues

VignetteBuilder knitr

Encoding UTF-8

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Timothée Giraud [cre, aut] (ORCID: <https://orcid.org/0000-0002-1932-3323>), Laurent Beauguitte [aut] (ORCID: <https://orcid.org/0000-0001-8803-6056>), Marianne Guérois [aut] (ORCID: <https://orcid.org/0000-0001-8285-1877>)

Maintainer Timothée Giraud <timothee.giraud@cnrs.fr>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-13 18:00:08 UTC

RemoteUrl https://github.com/cran/flows

RemoteRef HEAD

RemoteSha 1bc131520124a1e375904c3bdecf80910c4a0424

Contents

commuters_datasets 2
compare_mat 3
flows 4
map_nodal_flows 5
nodal_flows 6
plot_nodal_flows 7
prepare_mat 8
select_flows 9
stat_mat 10

Index **13**

`commuters_datasets` *Commuters datasets*

Description

Data on commuters between Urban Areas of the French Grand Est region in 2011. Fields:

- i: Code of the urban area of residence
- namei: Name of the urban area of residence
- wi: Total number of active occupied persons in the urban area of residence
- j: Code of the urban area of work
- namej: Name of the urban area of work
- wj: Total number of active occupied persons in the urban area of work
- fij: Number of commuters between i and j

Geopackage of the Grand Est region in France and its urban areas (2010 delineation).

References

Commuters dataset: https://www.insee.fr/fr/statistiques/2022113
Spatial dataset: https://www.data.gouv.fr/en/datasets/geofla-r

Examples

```
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
library(sf)
UA <- st_read(system.file("gpkg/GE.gpkg", package = "flows"), layer = "urban_area")
GE <- st_read(system.file("gpkg/GE.gpkg", package = "flows"), layer = "region")
```

`compare_mat` *Comparison of two matrices*

Description

Compares two matrices of same dimension, with same column and row names.

Usage

```
compare_mat(mat1, mat2, digits = 0)
```

Arguments

`mat1`	A square matrix of flows.
`mat2`	A square matrix of flows.
`digits`	An integer indicating the number of decimal places to be used when printing the data.frame in the console (see round).

Value

A data.frame that provides statistics on differences between mat1 and mat2: absdiff are the absolute differences and reldiff are the relative differences (in percent).

Examples

```
# # Import data
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
mat <- prepare_mat(x = nav, i = "i", j = "j", fij = "fij")
# Remove the matrix diagonal
diag(mat) <- 0

# Select the first flows
flowSel1 <- select_flows(mat = mat, method = "nfirst", k = 1)

# Select flows greater than 2000
flowSel2 <- select_flows(mat = mat, method = "xfirst", k = 2000)

# Combine selections
flowSel <- mat * flowSel1 * flowSel2

# Compare flow matrices
compare_mat(mat1 = mat, mat2 = flowSel, digits = 1)
```

flows *Flow Matrix Selection with Statistics and Visualisation*

Description

Selections on flow matrices, statistics on selected flows, map and graph visualisations. The analysis and representation of flows often presuppose a selection to facilitate interpretation. Various methods have been proposed for selecting flows, one of the most widely used being based on major flows: it selects only the most important flows, absolute or relative, on a local or global scale. These methods often highlight hierarchies between locations, but the loss of information caused by selection is rarely taken into account. We propose statistical indicators to assess the loss of information and the characteristics of selected flows. We provide functions that select flows (main, dominant or major flows), provide statistics on selections and offer visualizations in the form of maps and graphs.

An introduction to the package conceptual background and usage is proposed in a vignette (see `vignette(topic = "flows")`) and a paper (Beauguitte, Giraud & Guérois 2015).

Author(s)

Maintainer: Timothée Giraud <timothee.giraud@cnrs.fr> (ORCID)

Authors:

- Timothée Giraud <timothee.giraud@cnrs.fr> (ORCID)
- Laurent Beauguitte (ORCID)
- Marianne Guérois (ORCID)

References

L. Beauguitte, T. Giraud & M. Guérois, 2015. "Un outil pour la sélection et la visualisation de flux : le package flows", *Netcom*, 29-3/4:399-408. doi:10.4000/netcom.2134.

See Also

Useful links:

- https://codeberg.org/riatelab/flows
- Report bugs at https://codeberg.org/riatelab/flows/issues

`map_nodal_flows` *Nodal flows map*

Description

Perform a Nystuen & Dacey's dominants, or nodal, flows analysis and plot a dominant flows map.

Usage

```
map_nodal_flows(
  mat,
  x,
  inches = 0.15,
  col_node = c("red", "orange", "yellow"),
  breaks = "equal",
  nbreaks = 4,
  lwd = c(1, 5, 10, 20),
  col_flow = "grey20",
  leg_node = c("Dominant", "Intermediate", "Dominated",
    "Size proportional\nto sum of inflows"),
  leg_flow = "Flow intensity",
  leg_pos = "topleft",
  add = FALSE
)
```

Arguments

`mat`	A square matrix of flows.
`x`	An sf object, the first column contains a unique identifier matching mat column and row names.
`inches`	Size of the largest circle.
`col_node`	Node colors, a vector of 3 colors.
`breaks`	How to classify flows, either a numeric vector with the actual breaks, or a classification method name (see mf_get_breaks())
`nbreaks`	Number of classes.
`lwd`	Flows widths
`col_flow`	Flows color
`leg_node`	Labels for the nodes legend
`leg_flow`	Label for the flows legend
`leg_pos`	Position of the legend
`add`	A boolean, if TRUE, add the layer to an existing plot.

Value

A list of sf objects is returned. The first element contains the nodes with their weight and classification (dominant, intermediary, dominated). The second element contains the flows (i, j, fij)

Examples

```
library(sf)
library(mapsf)
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
mat <- prepare_mat(x = nav, i = "i", j = "j", fij = "fij")
UA <- st_read(system.file("gpkg/GE.gpkg", package = "flows"), layer = "urban_area")
GE <- st_read(system.file("gpkg/GE.gpkg", package = "flows"), layer = "region")
mf_map(GE)
map_nodal_flows(
  mat = mat, x = UA,
  col_node = c("red", "orange", "yellow"),
  col_flow = "grey30",
  breaks = c(4, 100, 1000, 2500, 8655),
  lwd = c(1, 4, 8, 16), add = TRUE
)
mf_title("Dominant flows")
```

---

nodal_flows *Nodal flows selection*

Description

Perform a Nystuen & Dacey's dominants flows analysis.

Usage

```
nodal_flows(mat)
```

Arguments

mat	A square matrix of flows.

Value

The matrix of the selected flows is returned.

Examples

```
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
mat <- prepare_mat(x = nav, i = "i", j = "j", fij = "fij")
res <- nodal_flows(mat)
res[1:5, 1:5]
```

plot_nodal_flows *Nodal flows graph*

Description

This function plots a dominant flows graph.

Usage

```
plot_nodal_flows(
  mat,
  leg_pos = "topright",
  leg_flow = "Flow Intensity",
  leg_node = "Node type",
  leg_node_modalities = c("Dominant", "Intermediary", "Dominated"),
  labels = FALSE
)
```

Arguments

mat	A square matrix of dominant flows (see nodal_flows).
leg_pos	Position of the legend, one of "topleft", "top", "topright", "left", "right", "bottomleft", "bottom", "bottomright".
leg_flow	Title of the flows legend.
leg_node	Title of the nodes legend.
leg_node_modalities	Text of node modalities.
labels	A boolean, if TRUE, labels of dominant and intermediary nodes are plotted.

Details

This function uses the Davidson Harel algorithm from igraph.

Value

The graph is returned, a graph is displayed.

Note

As square matrices can easily be plotted with plot.igraph or gplot functions from igraph and sna packages respectively, we do not propose visualisation for other output.

Examples

```
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
mat <- prepare_mat(x = nav, i = "i", j = "j", fij = "fij")
res <- nodal_flows(mat)

# Plot dominant flows graph
plot_nodal_flows(mat = res)
```

`prepare_mat` *Flow matrix preparation*

Description

From a long format matrix to a a wide format matrix.

Usage

```
prepare_mat(x, i, j, fij)
```

Arguments

`x`	A data.frame of flows between origins and destinations: long format matrix (origins, destinations, flows intensity).
`i`	A character giving the origin field name in mat.
`j`	A character giving the destination field name in mat.
`fij`	A character giving the flow field name in mat.

Value

A square matrix of flows. Diagonal can be filled or empty depending on data used.

Examples

```
# Import data
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
# Prepare data
myflows <- prepare_mat(x = nav, i = "i", j = "j", fij = "fij")
myflows[1:5, 1:5]
```

`select_flows` *Flow selection*

Description

Flow selection from origins.

Usage

```
select_flows(mat, method = "nfirst", ties = "first", global = FALSE, k, w)
```

Arguments

`mat`	A square matrix of flows.
`method`	A method of flow selection, one of "dominant", "nfirst", "xfirst" or "xsumfirst": • dominant selects the dominant flows (see Details) • nfirst selects the k first flows from origins, • xfirst selects flows greater than k, • xsumfirst selects as many flows as necessary for each origin so that their sum is at least equal to k. If k is not reached for one origin, all its flows are selected.
`ties`	In case of equality with "nfirst" method, use "random" or "first" (see rank).
`global`	If TRUE flows selections is done at the matrix scale.
`k`	Selection threshold for nfirst, xfirst and xsumfirst methods, ratio for dominant method.
`w`	A vector of units weigths (sum of incoming flows, sum of outgoing flows...).

Details

If method = "dominant", select which flow (fij or fji) must be kept. If the ratio weight of destination (wj) / weight of origin (wi) is greater than k, then fij is selected and fji is not. This function can perform the second criterion of the Nystuen & Dacey's dominants flows analysis.

Value

A boolean matrix of selected flows. Use element-wise multiplication to get flows intensity.

Examples

```
# Import data
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
# Prepare data
mat <- prepare_mat(x = nav, i = "i", j = "j", fij = "fij")
# remove diagonal
diag(mat) <- 0
```

```
# Select the first flow from each origin
res <- select_flows(mat = mat, method = "nfirst", global = FALSE, k = 1)
rowSums(res)

# Select the 5 first flows of the matrix
res <- select_flows(mat = mat, method = "nfirst", global = TRUE, k = 5)
sum(res)

# Select the flows greater than 5000
res <- select_flows(mat = mat, method = "xfirst", k = 5000)
r <- mat * res
r[r > 0]

# Select as many flows as necessary for each origin so that their sum is at least equal to 500.
res <- select_flows(mat = mat, method = "xsumfirst", global = FALSE, k = 500)
r <- mat * res
rowSums(r)

# Select as many flows in the matrix so that their sum is at least equal to 50000.
res <- select_flows(mat = mat, method = "xsumfirst", global = TRUE, k = 50000)
r <- mat * res
sum(rowSums(r))

# Select dominant flows
m <- mat[1:5, 1:5]
ws <- colSums(m)
res <- select_flows(mat = m, method = "dominant", k = 1, w = ws)
# 2nd element has a lower weight than 3rd element (ratio > 1)
ws[3] / ws[2]
# The flow from 2nd element to 3rd element is kept
res[2, 3]
# The flow from 3rd element to 2nd element is removed
res[3, 2]
```

`stat_mat` *Descriptive statistics on flow matrix*

Description

This function provides various indicators and graphical outputs on a flow matrix.

Usage

```
stat_mat(mat, output = "all", verbose = TRUE)
```

Arguments

`mat`	A square matrix of flows.

output	Graphical output. Choices are "all" for all graphics, "none" to avoid any graphical output, "degree" for degree distribution, "wdegree" for weighted degree distribution, "lorenz" for Lorenz curve of link weights and "boxplot" for boxplot of link weights (see 'Details').
verbose	A boolean, if TRUE, returns statistics in the console.

Details

Graphical ouputs concern outdegrees by default. If the matrix is transposed, outputs concern indegrees.

Value

The function returns a list of statistics and may plot graphics.

- nblinks: number of cells with values > 0
- density: number of links divided by number of possible links (also called gamma index by geographers), loops excluded
- connectcomp: number of connected components (isolates included, weakly connected: use of igraph::clusters where mode = "weak")
- connectcompx: number of connected components (isolates deleted, weakly connected: use of igraph::clusters where mode = "weak")
- sizecomp: a data.frame of connected components: size and sum of flows per component (isolates included)
- compocomp: a data.frame of connected components giving membership of units (isolates included)
- degrees: a data.frame of nodes degrees and weighted degrees
- sumflows: sum of flows
- min: minimum flow
- Q1: first quartile of flows
- median: median flow
- Q3: third quartile of flows
- max: maximum flow
- mean: mean flow
- sd: standart deviation of flows

Examples

```
# Import data
nav <- read.csv(system.file("csv/nav.csv", package = "flows"))
myflows <- prepare_mat(x = nav, i = "i", j = "j", fij = "fij")

# Get statistics and graphs about the matrix
mystats <- stat_mat(mat = myflows, output = "all", verbose = TRUE)
```

```
# Size of connected components
mystats$sizecomp

# Sum of flows
mystats$sumflows

# Plot Lorenz curve only
stat_mat(mat = myflows, output = "lorenz", verbose = FALSE)

# Statistics only
mystats <- stat_mat(mat = myflows, output = "none", verbose = FALSE)
str(mystats)
```

Index

commuters_datasets, 2
compare_mat, 3

flows, 4
flows-package (flows), 4

map_nodal_flows, 5

nodal_flows, 6, *7*

plot_nodal_flows, 7
prepare_mat, 8

rank, *9*
round, *3*

select_flows, 9
stat_mat, 10