Package ‘incidentally’

July 23, 2025

Title Generates Incidence Matrices and Bipartite Graphs

Version 1.0.2

Description Functions to generate incidence matrices and bipartite graphs that have (1) a fixed fill rate, (2) given marginal sums, (3) marginal sums that follow given distributions, or (4) represent bill sponsorships in the US Congress <doi:10.31219/osf.io/ectms>. It can also generate an incidence matrix from an adjacency matrix, or bipartite graph from a unipartite graph, via a social process mirroring team, group, or organization formation <doi:10.48550/arXiv.2204.13670>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Depends R (>= 2.10)

Imports igraph, Matrix, methods, stats, xml2

Suggests backbone, rmarkdown, knitr

VignetteBuilder knitr

URL `https://www.zacharyneal.com/backbone`,
`https://github.com/zpneal/incidentally`

BugReports `https://github.com/zpneal/incidentally/issues`

NeedsCompilation no

Author Zachary Neal [aut, cre] (ORCID: <`https://orcid.org/0000-0003-3076-4995`>)

Maintainer Zachary Neal <`zpneal@msu.edu`>

Repository CRAN

Date/Publication 2023-02-15 21:00:02 UTC

Contents

`add.blocks` *Adds a block structure to an incidence matrix*

Description

`add.blocks` shuffles an incidence matrix to have a block structure or planted partition while preserving the row and column sums

Usage

```
add.blocks(
  I,
  rowblock = sample(1:2, replace = T, nrow(I)),
  colblock = sample(1:2, replace = T, ncol(I)),
  density = 0.5,
  sorted = FALSE
)
```

Arguments

`I`	An incidence matrix or igraph bipartite graph
`rowblock`	numeric: vector indicating each row node's block membership
`colblock`	numeric: vector indicating each column node's block membership
`density`	numeric: desired within-block density
`sorted`	boolean: if TRUE, return incidence matrix permuted by block

Details

Stochastic block and planted partition models generate graphs in which the probability that two nodes are connected depends on whether they are members of the same or different blocks/partitions. Functions such as sample_sbm can randomly sample from stochastic block models with given probabilities. In contrast `add.blocks` adds a block structure to an existing incidence matrix while preserving the row and column sums. Row nodes' and column nodes' block memberships are supplied in separate vectors. If block membership vectors are not provided, then nodes are randomly assigned to two groups.

Value

An incidence matrix or igraph bipartite graph with a block structure

References

Neal, Z. P., Domagalski, R., and Sagan, B. 2021. Comparing alternatives to the fixed degree sequence model for extracting the backbone of bipartite projections. *Scientific Reports, 11*, 23929. doi: 10.1038/s41598021032383

Neal, Z. P. 2022. incidentally: An R package to generate incidence matrices and bipartite graphs. *OSF Preprints* doi: 10.31219/osf.io/ectms

Examples

```
I <- incidence.from.probability(R = 100, C = 100, P = .1)
blocked <- add.blocks(I, density = .7)
all(rowSums(I)==rowSums(blocked))
all(colSums(I)==colSums(blocked))

B <- igraph::sample_bipartite(100, 100, p=.1)
blocked <- add.blocks(B, density = .7)
all(igraph::degree(B)==igraph::degree(blocked))
```

`curveball` *Randomize an incidence matrix or bipartite graph using the curveball algorithm*

Description

`curveball` randomizes an incidence matrix or bipartite graph, preserving the row and column sums

Usage

```
curveball(M, trades = 5 * nrow(M), class = NULL)
```

Arguments

`M`	a binary matrix of class `matrix` or `Matrix`, or a bipartite graph of class igraph.
`trades`	integer: number of trades; the default is 5 * nrow(M) (approx. mixing time)
`class`	string: Return object as `matrix`, `Matrix`, `igraph`. If `NULL`, object is returned in the same class as `M`.

Details

Strona et al. (2014) provided an initial implementation of the Curveball algorithm in R. `curveball()` is a modified R implementation that is slightly more efficient. For an even more efficient algorithm, see `backbone::fastball()`.

Value

An incidence matrix of class `matrix` or `Matrix`, or a bipartite graph of class igraph.

References

Strona, Giovanni, Domenico Nappo, Francesco Boccacci, Simone Fattorini, and Jesus San-Miguel-Ayanz. 2014. A Fast and Unbiased Procedure to Randomize Ecological Binary Matrices with Fixed Row and Column Totals. *Nature Communications, 5*, 4114. doi: 10.1038/ncomms5114

Godard, Karl and Neal, Zachary P. 2022. fastball: A fast algorithm to sample bipartite graphs with fixed degree sequences. *arXiv:2112.04017*

Neal, Z. P. 2022. incidentally: An R package to generate incidence matrices and bipartite graphs. *OSF Preprints* doi: 10.31219/osf.io/ectms

Examples

```
M <- incidence.from.probability(5,5,.5)  #A matrix
Mrand <- curveball(M)  #Random matrix with same row/col sums
all.equal(rowSums(M), rowSums(curveball(M)))
all.equal(colSums(M), colSums(curveball(M)))
```

`incidence.from.adjacency` *Generates an incidence matrix from an adjacency matrix*

Description

`incidence.from.adjacency` generates an incidence matrix from an adjacency matrix or network using a given generative model

Usage

```
incidence.from.adjacency(
  G,
  k = 1,
  p = 1,
  blau.param = c(2, 1, 10),
  maximal = TRUE,
  model = "team",
  class = NULL,
  narrative = TRUE
)
```

Arguments

`G`	A symmetric, binary adjacency matrix of class `matrix` or `Matrix`, or an undirected, unweighted unipartite graph of class igraph.
`k`	integer: Number of artifacts to generate
`p`	numeric: Tuning parameter for artifacts, 0 <= p <= 1
`blau.param`	vector: Vector of parameters that control blau space in the organizations model (see details)

`maximal`	boolean: Should teams/clubs models be seeded with *maximal* cliques?
`model`	string: Generative model, one of c("team", "club", "org") (see details)
`class`	string: Return object as `matrix`, `Matrix`, or `igraph`. If `NULL`, object is returned in the same class as `G`.
`narrative`	boolean: TRUE if suggested text & citations should be displayed.

Details

Given a unipartite network composed of *i agents* (i.e. nodes) that can be represented by an *i x i* adjacency matrix, `incidence.from.adjacency` generates a random *i x k* incidence matrix that indicates whether agent *i* is associated with *artifact k*. Generative models differ in how they conceptualize artifacts and how they associate agents with these artifacts.

The **Team Model** (`model == "team"`) mirrors a team formation process, where each artifact represents a new team formed from the incumbants of a prior team (with probability `p`) and newcomers (with probability 1-`p`).

The **Club Model** (`model == "club"`) mirrors a social club formation process, where each artifact represents a social club. Club members attempt to recruit non-member friends, who join the club if it would have a density of at least `p`.

The **Organizations Model** (`model == "org"`) mirrors an organization (the artifact) recruiting members from social space, where those within the organization's niche join with probability `p`, and those outside the niche join with probability 1-`p`. `blau.param` is a vector containing three values that control the characteristics of the blau space. The first value is the space's dimensionality. The second two values are shape parameters of a Beta distribution that describes niche sizes. The default is a two-dimensional blau space, with organization niche sizes that are strongly positively skewed (i.e., many specialist organizations, few generalists).

Value

An incidence matrix of class `matrix` or `Matrix`, or a bipartite graph of class igraph.

References

Neal, Z. P. 2023. The duality of networks and groups: Models to generate two-mode networks from one-mode networks. *Network Science*.

Examples

```
G <- igraph::erdos.renyi.game(10, .4)
I <- incidence.from.adjacency(G, k = 1000, p = .95,
                              model = "team", narrative = TRUE)
```

incidence.from.congress

Generate bill sponsorship incidence matrices and bipartite graphs

Description

`incidence.from.congress()` uses data from https://www.congress.gov/ to construct an incidence matrix or bipartite graph recording legislators' bill (co-)sponsorships.

Usage

```
incidence.from.congress(
  session = NULL,
  types = NULL,
  areas = "all",
  nonvoting = FALSE,
  weighted = FALSE,
  format = "data",
  narrative = FALSE
)
```

Arguments

`session`	numeric: the session of congress
`types`	vector: types of bills to include. May be any combination of c("s", "sres", "sjres", "sconres") OR any combination of c("hr", "hres", "hjres", "hconres").
`areas`	string: policy areas of bills to include (see details)
`nonvoting`	boolean: should non-voting members be included
`weighted`	boolean: should sponsor-bill edges have a weight of 2, but cosponsor-bill edges have a weight of 1
`format`	string: format of output, one of c("data", "igraph")
`narrative`	boolean: TRUE if suggested text & citations should be displayed.

Details

The `incidence.from.congress()` function uses data from https://www.congress.gov/ to construct an incidence matrix or bipartite graph recording legislators' bill (co-)sponsorships. In an incidence matrix **I**, entry *Iik = 1* if legislator *i* sponsored or co-sponsored bill *k*, and otherwise is 0. In a bipartite graph **G**, a legislator *i* is connected to a bill *k* if *i* sponsored or co-sponsored *k*.

In the US Congress, the law making process begins when a *sponsor* legislator introduces a bill in their chamber (House of Representatives or Senate). Additional legislators in the same chamber can support the bill by joining as a *co-sponsor*. The bill is discussed, revised, and possibly voted on in the chamber. If it passes in one chamber, it is sent to the other chamber for further discussion, revision, and possibly a vote. If it passed both chambers, it is sent to the President. If the President signs the bill, it becomes law.

In the House of Representatives, legislators can introduce four types of bills: a House Bill (hr), a House Joint Resolution (hjres), a House Concurrent Resolution (hconres), and a House Simple Resolution (hres). Similarly, in the Senate, legislators can introduce four types of bills: a Senate Bill (s), a Senate Joint Resolution (sjres), a Senate Concurrent Resolution (sconres), and a Senate Simple Resolution (sres). In both chambers, concurrent and simple resolutions are used for minor procedural matters and do not have the force of law. Only bills and joint resolutions require the President's signature and have the force of law if signed.

Each bill is assigned a policy area by the Congressional Research Service. By default, bills from all policy areas are included, however the `areas` parameter can be used to include only bills addressing certain policy areas. The `areas` takes a vector of strings listing the desired policy areas (e.g., `areas = c("Congress", "Animals")`). A complete list of policy areas and brief descriptions is available at https://www.congress.gov/help/field-values/policy-area.

Value

If `format = "data"`, a list containing an incidence matrix, a dataframe of legislator characteristics, and a dataframe of bill characteristics.

If `format = "igraph"`, a bipartite igraph object composed of legislator vertices and bill vertices, each with vertex attributes.

For both formats, legislator characteristics include: BioGuide ID, full name, last name, party affiliation, and state. Bill characteristics include: bill ID, introduction date, title, policy area, status, sponsor's party, and number of co-sponsors from each party.

References

Tutorial: Neal, Z. P. 2022. Constructing legislative networks in R using incidentally and backbone. *Connections, 42*. doi: 10.2478/connections2019.026

Package: Neal, Z. P. 2022. incidentally: An R package to generate incidence matrices and bipartite graphs. *OSF Preprints* doi: 10.31219/osf.io/ectms

Examples

```
## Not run:
D <- incidence.from.congress(session = 116, types = "s", format = "data")
D <- incidence.from.congress(session = 116, types = "s", format = "data", areas = "animals")
G <- incidence.from.congress(session = 115, types = c("hr", "hres"), format = "igraph")

## End(Not run)
```

`incidence.from.distribution` *Generates an incidence matrix with row and column sums that follow given distributions*

Description

`incidence.from.distribution` generates a random incidence matrix with row and column sums that approximately follow beta distributions with given parameters.

Usage

```
incidence.from.distribution(
  R,
  C,
  P,
  rowdist = c(1, 1),
  coldist = c(1, 1),
  class = "matrix",
  narrative = TRUE
)
```

Arguments

`R`	integer: number of rows
`C`	integer: number of columns
`P`	numeric: probability that a cell contains a 1
`rowdist`	vector length 2: Row marginals will approximately follow a Beta(a,b) distribution
`coldist`	vector length 2: Column marginals will approximately follow a Beta(a,b) distribution
`class`	string: the class of the returned backbone graph, one of c("matrix", "Matrix", "igraph").
`narrative`	boolean: TRUE if suggested text & citations should be displayed.

Value

An incidence matrix of class `matrix` or `Matrix`, or a bipartite graph of class igraph.

References

Neal, Z. P., Domagalski, R., and Sagan, B. 2021. Comparing alternatives to the fixed degree sequence model for extracting the backbone of bipartite projections. *Scientific Reports, 11*, 23929. doi: 10.1038/s41598021032383

Neal, Z. P. 2022. incidentally: An R package to generate incidence matrices and bipartite graphs. *OSF Preprints* doi: 10.31219/osf.io/ectms

Examples

```
I <- incidence.from.distribution(R = 100, C = 100, P = 0.1,
     rowdist = c(10000,10000), coldist = c(10000,10000))  #Constant
I <- incidence.from.distribution(R = 100, C = 100, P = 0.1,
     rowdist = c(1,1), coldist = c(1,1))  #Uniform
I <- incidence.from.distribution(R = 100, C = 100, P = 0.1,
     rowdist = c(1,10), coldist = c(1,10))  #Right-tailed
I <- incidence.from.distribution(R = 100, C = 100, P = 0.1,
     rowdist = c(10,1), coldist = c(10,1))  #Left-tailed
I <- incidence.from.distribution(R = 100, C = 100, P = 0.1,
     rowdist = c(10,10), coldist = c(10,10),
     narrative = TRUE)  #Normal
```

incidence.from.probability

Generates an incidence matrix with a given cell-filling probability

Description

`incidence.from.probability` generates a random incidence matrix in which each cell is filled with a 1 with a given probability.

Usage

```
incidence.from.probability(
  R,
  C,
  P = 0,
  constrain = TRUE,
  class = "matrix",
  narrative = FALSE
)
```

Arguments

`R`	integer: number of rows
`C`	integer: number of columns
`P`	numeric: probability that a cell contains a 1; if P = 0 a probability will be chosen randomly
`constrain`	boolean: ensure that no rows or columns sum to 0 (i.e., contain all 0s) or to 1 (i.e., contain all 1s)
`class`	string: the class of the returned backbone graph, one of c("matrix", "Matrix", "igraph").
`narrative`	boolean: TRUE if suggested text & citations should be displayed.

Value

An incidence matrix of class `matrix` or `Matrix`, or a bipartite graph of class igraph.

References

Neal, Z. P., Domagalski, R., and Sagan, B. 2021. Comparing alternatives to the fixed degree sequence model for extracting the backbone of bipartite projections. *Scientific Reports, 11*, 23929. doi: 10.1038/s41598021032383

Neal, Z. P. 2022. incidentally: An R package to generate incidence matrices and bipartite graphs. *OSF Preprints* doi: 10.31219/osf.io/ectms

Examples

```
I <- incidence.from.probability(R = 10, C = 10)
I <- incidence.from.probability(R = 10, C = 10, P = .5)
I <- incidence.from.probability(R = 10, C = 10, P = .5,
     class = "igraph", narrative = TRUE)
```

`incidence.from.vector` *Generates an incidence matrix with given row and column marginal sums*

Description

`incidence.from.vector` generates a random incidence matrix with given row and column sums

Usage

```
incidence.from.vector(R, C, class = "matrix", narrative = FALSE)
```

Arguments

`R`	numeric vector: row marginal sums
`C`	numeric vector: column marginal sums
`class`	string: the class of the returned backbone graph, one of c("matrix", "Matrix", "igraph").
`narrative`	boolean: TRUE if suggested text & citations should be displayed.

Value

An incidence matrix of class `matrix` or `Matrix`, or a bipartite graph of class igraph.

References

Neal, Z. P., Domagalski, R., and Sagan, B. 2021. Comparing alternatives to the fixed degree sequence model for extracting the backbone of bipartite projections. *Scientific Reports, 11*, 23929. doi: 10.1038/s41598021032383

Neal, Z. P. 2022. incidentally: An R package to generate incidence matrices and bipartite graphs. *OSF Preprints* doi: 10.31219/osf.io/ectms

Examples

```
I <- incidence.from.vector(R = c(1,1,2), C = c(1,1,2))
I <- incidence.from.vector(R = c(1,1,2), C = c(1,1,2),
     class = "igraph", narrative = TRUE)
```

`incidentally` *incidentally: Generates incidence matrices and bipartite graphs*

Description

Functions to generate incidence matrices and bipartite graphs that have (1) a fixed fill rate, (2) given marginal sums, (3) marginal sums that follow given distributions, or (4) represent bill sponsorships in the US Congress. It can also generate an incidence matrix from an adjacency matrix, or bipartite graph from a unipartite graph, via a social process mirroring team, group, or organization formation.

Incidence matrices can be generated:

- ...with a fixed fill rate: `incidence.from.probability()`.
- ...with given marginals: `incidence.from.vector()`.
- ...with marginals that follow given distributions: `incidence.from.distribution()`.
- ...from a network, by a social process mirroring team, group, or organization formation `incidence.from.adjacency()`
- ...with a block structure or planted partition: `add.blocks()`.
- ...from US Congress bill sponsorships: `incidence.from.congress()`.

Index