

# Package ‘hypergraph’

December 17, 2024

**Title** A package providing hypergraph data structures

**Version** 1.78.0

**Author** Seth Falcon, Robert Gentleman

**Description** A package that implements some simple capabilities for representing and manipulating hypergraphs.

**Maintainer** Bioconductor Package Maintainer

<maintainer@bioconductor.org>

**License** Artistic-2.0

**Depends** R (>= 2.1.0), methods, utils, graph

**Suggests** BiocGenerics, RUnit

**LazyLoad** yes

**Collate** AllClasses.R AllGenerics.R kCores.R methods-Hyperedge.R  
methods-Hypergraph.R

**biocViews** GraphAndNetwork

**git\_url** <https://git.bioconductor.org/packages/hypergraph>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** b4abc3c

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-12-16

## Contents

DirectedHyperedge . . . . .	2
DirectedHyperedge-class . . . . .	2
Hyperedge . . . . .	3
Hyperedge-class . . . . .	4
Hypergraph . . . . .	5
Hypergraph-class . . . . .	5
kCoresHypergraph . . . . .	6
l2hel . . . . .	7
vCoverHypergraph . . . . .	8
<b>Index</b>	<b>10</b>

---

DirectedHyperedge      *Constructor for DirectedHyperedge objects*

---

### Description

A convenience constructor for [DirectedHyperedge-class](#) objects

### Usage

```
DirectedHyperedge(head, tail, label = "")
```

### Arguments

head	Character vector of nodes that are part of the head of the hyperedge
tail	Character vector of nodes that part of the tail of the hyperedge
label	A character string describing the directed hyperedge

### Value

An object of class [DirectedHyperedge-class](#)

### Author(s)

Seth Falcon

### See Also

[DirectedHyperedge-class](#) [Hyperedge-class](#) [Hypergraph-class](#)

---

DirectedHyperedge-class  
*Class DirectedHyperedge*

---

### Description

This class represents directed hyperedges in a [Hypergraph-class](#). A directed hyperedge consists of two disjoint sets of nodes, those in the tail and those in the head of the hyperedge. Directed hyperedges are sometimes called hyperarcs.

### Objects from the Class

Objects can be created by calls of the form `new("DirectedHyperedge", head, tail, label)`. You can also use the convenience function [DirectedHyperedge](#).

### Slots

tail: Character vector of nodes in the tail of the hyperedge  
 head: Character vector of nodes in the head of the hyperedge  
 label: Character string describing the directed hyperedge

**Extends**

Class "Hyperedge", directly.

**Methods**

**head** signature( $x = \text{"DirectedHyperedge"}$ ): Return a vector containing the nodes in the head of the hyperedge

**tail** signature( $x = \text{"DirectedHyperedge"}$ ): Return a vector containing the nodes in the tail of the hyperedge

**initialize** signature( $\text{.Object} = \text{"DirectedHyperedge"}$ ): Create a new instance.

**nodes** signature( $\text{object} = \text{"DirectedHyperedge"}$ ): Return a vector containing all nodes present in the hyperedge.

**show** signature( $\text{object} = \text{"DirectedHyperedge"}$ ): Print me

**toUndirected** signature( $\text{.Object} = \text{"DirectedHyperedge"}$ ): Return a [Hyperedge-class](#) object that results from coercing to an undirected hyperedge.

**Author(s)**

Seth Falcon

**See Also**

[DirectedHyperedge](#) [Hyperedge](#) [Hyperedge-class](#) [Hypergraph-class](#)

**Examples**

```
head <- LETTERS[1:4]
tail <- LETTERS[19:21]
label <- "Directed hyperedge"
dhe <- new("DirectedHyperedge", head=head, tail=tail, label=label)
```

---

Hyperedge

*Constructor for Hyeredge objects*

---

**Description**

A convenience constructor for [Hyperedge-class](#) objects

**Usage**

```
Hyperedge(nodes, label = "")
```

**Arguments**

**nodes** Character vector of nodes that are part of the hyperedge

**label** A character string describing the hyperedge

**Value**

An object of class [Hyperedge-class](#)

**Author(s)**

Seth Falcon

**See Also**[Hyperedge-class](#) [Hypergraph-class](#)

---

`Hyperedge-class`*Class Hyperedge*

---

**Description**

A Hyperedge object represents a hyperedge in a hypergraph, that is, a subset of the nodes of a hypergraph.

**Objects from the Class**

Objects can be created by calls of the form `new("Hyperedge", nodes, label)`. You can also use the convenience function `Hyperedge` to create instances. This is especially useful for creating a list of Hyperedge instances using [lapply](#).

**Slots**

**head:** A vector of mode "character" containing the node labels that are a part of the hyperedge

**label:** An arbitrary "character" string describing this hyperedge

**Methods**

**initialize** signature(.Object = "Hyperedge"): Create an instance

**label** signature(object = "Hyperedge"): Return the value of the label slot

**label<-** signature(object = "Hyperedge", value = "character"): Set the label slot.

**nodes** signature(object = "Hyperedge"): Return a vector containing the nodes in the hyperedge

**show** signature(object = "Hyperedge"): Print a textual summary of the hyperedge

**Author(s)**

Seth Falcon

**See Also**[Hyperedge](#) [Hypergraph-class](#) [DirectedHyperedge-class](#)**Examples**

```
nodes <- LETTERS[1:4]
label <- "Simple hyperedge"
## Use the convenience constructor
he <- Hyperedge(nodes, label)
```

---

Hypergraph

*Constructor for Hypergraph objects*

---

**Description**

A convenience constructor for `link{Hypergraph-class}` objects

**Usage**

`Hypergraph(nodes, hyperedges)`

**Arguments**

`nodes`            A vector of nodes (character)  
`hyperedges`       A list of [Hyperedge-class](#) objects

**Value**

An object of class [Hypergraph-class](#)

**Author(s)**

Seth Falcon

**See Also**

[Hypergraph-class](#) [Hyperedge-class](#) [DirectedHyperedge-class](#)

---

`Hypergraph-class`

*Class Hypergraph*

---

**Description**

A hypergraph consists of a set of nodes and a set of hyperedges. Each hyperedge is a subset of the node set. This class provides a representation of a hypergraph that is (hopefully) useful for computing.

**Objects from the Class**

Objects can be created by calls of the form `new("Hypergraph", nodes, hyperedges)`. You can also use the convenience function `Hypergraph`. The `nodes` argument should be a character vector of distinct labels representing the nodes of the hypergraph. The `hyperedges` argument must be a list of [Hyperedge-class](#) objects.

**Slots**

`nodes`: A "character" vector specifying the nodes  
`hyperedges`: A "list" of [Hyperedge-class](#) objects

**Methods**

- hyperedges** signature(.Object = "Hypergraph"): Return the list of Hyperedge objects
- hyperedgeLabels** signature(.Object = "Hypergraph"): Return a character vector of labels for the Hyperedge objects in the hypergraph.
- inciMat** signature(.Object = "Hypergraph"): Return the incidence matrix representation of this hypergraph
- inciMat2HG** signature(.Object = "matrix"): Return the hypergraph representation of this incidence matrix
- initialize** signature(.Object = "Hypergraph"): Create an instance
- nodes** signature(object = "Hypergraph"): Return the vector of nodes (character vector)
- numNodes** signature(object = "Hypergraph"): Return the number of nodes in the hypergraph
- toGraphNEL** signature(.Object = "Hypergraph"): Return the graphNEL representation of the hypergraph (a bipartite graph)

**Author(s)**

Seth Falcon

**See Also**

[Hyperedge-class DirectedHyperedge-class graphNEL-class](#)

**Examples**

```
nodes <- LETTERS[1:4]
hEdges <- lapply(list("A", LETTERS[1:2], LETTERS[3:4]), "Hyperedge")
hg <- new("Hypergraph", nodes=nodes, hyperedges=hEdges)
```

---

kCoresHypergraph

*Find all the k-cores in a hypergraph*

---

**Description**

Find all the k-cores in a hypergraph

**Usage**

```
kCoresHypergraph(hg)
```

**Arguments**

hg                    an instance of the Hypergraph class

**Details**

A k-core in a hypergraph is a maximal subhypergraph where (a) no hyperedge is contained in another, and (b) each node is adjacent to at least k hyperedges in the subgraph.

The implementation is based on the algorithm by E. Ramadan, A. Tarafdar, A. Pothen, 2004.

**Value**

A vector of the core numbers for all the nodes in *g*.

**Author(s)**

Li Long <li.long@isb-sib.ch>

**References**

A hypergraph model for the yeast protein complex network, Ramadan, E. Tarafdar, A. Pothen, A., Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International.

**Examples**

```
# to turn the snacoreex.gxl (from RBGL package) graph to a hypergraph
# this is a rough example
kc_hg_n <- c("A", "C", "B", "E", "F", "D", "G", "H", "J", "K", "I", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U")
kc_hg_e <- list(c("A", "C"), c("B", "C"), c("C", "E"), c("C", "F"), c("E", "D"), c("E", "F"), c("D", "G"), c("D", "H"))
kc_hg_he <- lapply(kc_hg_e, "Hyperedge")
kc_hg <- new("Hypergraph", nodes=kc_hg_n, hyperedges=kc_hg_he)

kCoresHypergraph(kc_hg)
```

---

l2hel

---

*Create lists of Hyperedge objects*


---

**Description**

Conveniently create lists of [Hyperedge-class](#) instances.

**Usage**

```
l2hel(e)
```

**Arguments**

*e* A list of character vectors. Each element of the list represents a hyperedge and the character vector value specifies the nodes of the hypergraph that are part of the hyperedge. The names of the list elements, if found, will be used as the label for the corresponding Hyperedge object.

**Value**

A list of Hyperedge-class objects. If the list *e* did not have names, the labels of the Hyperedges will be set to its index in the list coerced to character.

**Author(s)**

Seth Falcon

**See Also**

[Hyperedge-class](#) [Hypergraph-class](#)

**Examples**

```
edges <- list("e1"="A", "e2"=c("A", "B"), "e3"=c("C", "D"))
hEdgeList <- l2he1(edges)
```

---

vCoverHypergraph

*Approximate minimum weight vertex cover in a hypergraph*

---

**Description**

Approximate minimum weight vertex cover in a hypergraph

**Usage**

```
vCoverHypergraph(hg, vW=rep(1, numNodes(hg)))
```

**Arguments**

hg	an instance of the Hypergraph class
vW	vertex weights

**Details**

Hypergraph  $g$  has non-negative weights on its vertices. The minimum weight vertex cover problem is to find a subset of vertices  $C$  such that  $C$  includes at least one vertex from each hyperedge and the sum of the weights of the vertices in  $C$  is minimum. This problem is NP-hard.

We implement the greedy algorithm to approximate near-optimal solution, proposed by E. Ramadan, A. Tarafdar, A. Pothen, 2004.

**Value**

A list of vertices from hypergraph  $g$ .

**Author(s)**

Li Long <li.long@isb-sib.ch>

**References**

A hypergraph model for the yeast protein complex network, Ramadan, E. Tarafdar, A. Pothen, A., Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International.



**Examples**

```
# to turn the snacoreex.gxl graph (from RBGL package) to a hypergraph
# this is a rough example
kc_hg_n <- c("A", "C", "B", "E", "F", "D", "G", "H", "J", "K", "I", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U")
kc_hg_e <- list(c("A", "C"), c("B", "C"), c("C", "E"), c("C", "F"), c("E", "D"), c("E", "F"), c("D", "G"), c("D", "H"))
kc_hg_he <- lapply(kc_hg_e, "Hyperedge")
kc_hg <- new("Hypergraph", nodes=kc_hg_n, hyperedges=kc_hg_he)

vCoverHypergraph(kc_hg)
```

# Index

## \* classes

- DirectedHyperedge, 2
- DirectedHyperedge-class, 2
- Hyperedge, 3
- Hyperedge-class, 4
- Hypergraph, 5
- Hypergraph-class, 5
- l2hel, 7

## \* models

- kCoresHypergraph, 6
- vCoverHypergraph, 8

DirectedHyperedge, 2, 2, 3

DirectedHyperedge-class, 2

head (DirectedHyperedge-class), 2

head, DirectedHyperedge-method  
(DirectedHyperedge-class), 2

Hyperedge, 3, 3, 4

Hyperedge-class, 4

hyperedgeLabels (Hypergraph-class), 5

hyperedgeLabels, Hypergraph-method  
(Hypergraph-class), 5

hyperedges (Hypergraph-class), 5

hyperedges, Hypergraph-method  
(Hypergraph-class), 5

Hypergraph, 5

Hypergraph-class, 5

inciMat (Hypergraph-class), 5

inciMat, Hypergraph-method  
(Hypergraph-class), 5

inciMat2HG (Hypergraph-class), 5

inciMat2HG, matrix-method  
(Hypergraph-class), 5

initialize, DirectedHyperedge-method  
(DirectedHyperedge-class), 2

initialize, Hyperedge-method  
(Hyperedge-class), 4

initialize, Hypergraph-method  
(Hypergraph-class), 5

kCoresHypergraph, 6

l2hel, 7

label (Hyperedge-class), 4

label, Hyperedge-method  
(Hyperedge-class), 4

label<- (Hyperedge-class), 4

label<- , Hyperedge, character-method  
(Hyperedge-class), 4

lapply, 4

nodes, DirectedHyperedge-method  
(DirectedHyperedge-class), 2

nodes, Hyperedge-method  
(Hyperedge-class), 4

nodes, Hypergraph-method  
(Hypergraph-class), 5

numNodes, Hypergraph-method  
(Hypergraph-class), 5

show, DirectedHyperedge-method  
(DirectedHyperedge-class), 2

show, Hyperedge-method  
(Hyperedge-class), 4

tail (DirectedHyperedge-class), 2

tail, DirectedHyperedge-method  
(DirectedHyperedge-class), 2

toGraphNEL (Hypergraph-class), 5

toGraphNEL, Hypergraph-method  
(Hypergraph-class), 5

toUndirected (DirectedHyperedge-class),  
2

toUndirected, DirectedHyperedge-method  
(DirectedHyperedge-class), 2

vCoverHypergraph, 8