The giRaph package for graph representation in R
La Rocca, Luca; Badsberg, Jens Henrik; Dethlefsen, Claus

Publication date:
2006

Document Version
Publisher’s PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
The giRaph package for graph representation in R
Luca La Rocca* Jens Henrik Badsberg** Claus Dethlefsen***
* University of Modena and Reggio Emilia, Italy ** Statens Serum Institut, Denmark *** Aalborg Hospital, Denmark

<table>
<thead>
<tr>
<th>A broad notion of graph</th>
<th>Multi-graphs</th>
<th>Simple-graphs</th>
</tr>
</thead>
</table>

General graphs may have
- **hyper-edges**, such as
  - b--d--e
  - f->e->b--d->a--c
- **loops**, such as
  - i<>i
- **multiple edges**, such as
  - b->d, d->b
  - e->h, e->h, e->h
  - f--i, f->i

Represented via **incidence list**
```r
G <- new("incidenceList", V = letters[1:12], E = list(d(6, 5, c(2, 4), c(1, 3)), u(2, 4, 5), d(2, 4), d(4, 2), d(1, 7), d(3, 7), d(4, 7), d(5, 8), d(5, 8), u(6, 9), d(6, 9), u(9, 9), d(9, 8), d(9, 12), u(7, 8), u(8, 12), u(12, 11), u(11, 7), u(11, 8), d(11, 10)))
```
```
> G
An object of class "incidenceList"
V={a,b,c,d,e,f,g,h,i,j,k,l} E={f->e->b--d->a--c, b--d--e, b->d, d->b, a->g, c->g, d->g, e->h, e->h, e->h, f--i, f->i, i<>i, i->h, i->l, g--h, h--l, l--k, k--g, k--h, k->j}
```

or via **incidence matrix**
```r
I <- as(G, "incidenceMatrix")
```
```r
> I[1:6] # incidence matrix of induced subgraph
An object of class incidenceMatrix
a b c
d e f
1.1 4 3 4 3 2 1
2.2 1 0 2 0 0 0
3.3 0 1 2 0 0 0
4.4 0 2 0 1 0 0
```

Hyper-edges banned
```r
mg <- as(gg, "multiGraph")
```
```r
> adjacencyMatrix((mg[7:12])) # gets induced subgraph first
An object of class adjacencyMatrix
b c d e f
1 1 1 1 1
2 1 1 1 1 1
3 1 1 1 1 1
4 1 1 1 1 1
5 1 1 1 1 1
6 1 1 1 1 1
7 1 1 1 1 1
8 1 1 1 1 1
9 1 1 1 1 1
```

Simple-graphs
Loops and multiple edges banned
```r
sg <- as(mg, "simpleGraph")
```
```
> adjacencyMatrix((sg[7:12])) # gets induced subgraph first
An object of class adjacencyMatrix
b c d e f
1 1 1 1 1
2 1 1 1 1 1
3 1 1 1 1 1
4 1 1 1 1 1
5 1 1 1 1 1
6 1 1 1 1 1
7 1 1 1 1 1
8 1 1 1 1 1
9 1 1 1 1 1
```

Setting representations
Representation in use can be changed
```r
isしまう(gg@incidenceMatrix) <- f
```
```
> c(isEmpty(gg@incidenceList), isEmpty(gg@incidenceMatrix))
[1] TRUE FALSE
```
```
or a consistent representation can be added
```r
> adjacencyMatrix((mg[7:12]), force=F) <- adjacencyMatrix((mg)
```
```
> c(isEmpty(mg@adjacencyList), isEmpty(mg@incidenceList))
[1] FALSE FALSE
```
```
Working with vertices & edges
Possible via overloaded operators
```r
ng <- asGraph("simpleGraph")
```
```r
> ng <- addVertex("x", "y") # adds two isolated vertices
> ng <- addVertex("a") # removes a vertex (& an edge...)
> ng <- addEdge("b", "y") # gets an induced subgraph
```

Connection to other packages
Suggests (but does not depend on)

References