# Graphs in the gRbase package

Søren Højsgaard

January 30, 2009

## Contents



## 1 Introduction

For the R community, the three packages graph, RBGL and Rgraphviz are extremely useful tools for graph operations, manipulation and layout. The gRbase package adds some additional tools to these fine packages. The most important ones are:

1. Undirected and direacted acyclic graphs can be specified using formulae.
2. Graphs can be represented as either graphNEL objects (the format used in the graph package) or as adjacency matrices.
3. Some additional graph algorithms are implemented. The most important ones are moralization, maximum cardinality search, triangulation and rip ordering.

## 2 Graphs

An undirected graph is created by the ug() function. The graph can be specified by a formula, a list of formulas or a list of vectors. Thus the following two forms are equivalent:

```
> ug11 <- ug(~a * b * c, ~c * d, ~d * e, ~e * a, ~f * g)

A graphNEL graph with undirected edges
Number of Nodes = 7
Number of Edges = 7

> ug12 <- ug(~a * b * c + c * d + d * e + a * e + f * g)

A graphNEL graph with undirected edges
Number of Nodes = 7
Number of Edges = 7

> ug13 <- ug(c("a", "b", "c"), c("c", "d"), c("d", "e"), c("a", "e"),
+     c("f", "g"))

A graphNEL graph with undirected edges
Number of Nodes = 7
Number of Edges = 7
```

Instead of "`*`", a "`:`" can be used in the specification. These graphs are of type `graphNEL` where "NEL" stands for "node–edge–list".

A representation as an adjacency matrix can be obtained with

```
> ug11m <- ugMAT(~a * b * c, ~c * d, ~d * e, ~e * a, ~f * g)

  a b c d e f g
a 0 1 1 0 1 0 0
b 1 0 1 0 0 0 0
c 1 1 0 1 0 0 0
d 0 0 1 0 1 0 0
e 1 0 0 1 0 0 0
f 0 0 0 0 0 0 1
g 0 0 0 0 0 1 0

> ug12m <- ugMAT(~a * b * c + c * d + d * e + a * e + f * g)

  a b c d e f g
a 0 1 1 0 1 0 0
b 1 0 1 0 0 0 0
c 1 1 0 1 0 0 0
d 0 0 1 0 1 0 0
e 1 0 0 1 0 0 0
f 0 0 0 0 0 0 1
g 0 0 0 0 0 1 0

> ug13m <- ugMAT(c("a", "b", "c"), c("c", "d"), c("d", "e"), c("a",
+     "e"), c("f", "g"))

  a b c d e f g
a 0 1 1 0 1 0 0
b 1 0 1 0 0 0 0
c 1 1 0 1 0 0 0
d 0 0 1 0 1 0 0
e 1 0 0 1 0 0 0
f 0 0 0 0 0 0 1
g 0 0 0 0 0 1 0
```

A directed acyclic graph can be specified as a collection of formulas or as a list of vectors:

```
> dag11 <- dag(~a, ~b * a, ~c * a * b, ~d * c * e, ~e * a, ~g * f)

A graphNEL graph with directed edges
Number of Nodes = 7
Number of Edges = 7

> dag12 <- dag("a", c("b", "a"), c("c", "a", "b"), c("d", "c", "e"),
+     c("e", "a"), c("g", "f"))

A graphNEL graph with directed edges
Number of Nodes = 7
Number of Edges = 7
```

Here `~a` means that "a" has no parents while `~d*b*c` means that "d" has parents "b" and "c". Instead of "`*`", a "`:`" can be used in the specification.

A representation as an adjacency matrix can be obtained with

```
> dag11m <- dagMAT(~a, ~b * a, ~c * a * b, ~d * c * e, ~e * a, ~g *
+     f)

  a b c d e g f
a 0 1 1 0 1 0 0
b 0 0 1 0 0 0 0
c 0 0 0 1 0 0 0
d 0 0 0 0 0 0 0
e 0 0 0 1 0 0 0
g 0 0 0 0 0 0 0
f 0 0 0 0 0 1 0

> dag12m <- dagMAT("a", c("b", "a"), c("c", "a", "b"), c("d", "c",
+     "e"), c("e", "a"), c("g", "f"))

  a b c d e g f
a 0 1 1 0 1 0 0
b 0 0 1 0 0 0 0
c 0 0 0 1 0 0 0
d 0 0 0 0 0 0 0
e 0 0 0 1 0 0 0
g 0 0 0 0 0 0 0
f 0 0 0 0 0 1 0
```

Graphs (represented as `graphNEL` objects) are displayed with `plot()`, but this requires that the `Rgraphviz` package is installed.

*Notice: At some point, ugMAT() and dagMAT() might be made redundant by providing an argument to ug() and dag() specifying which representation to use.*

## 2.1 Graph queries

The `graph` and `RBGL` packages implement various graph operations for `graphNEL` objects.

The `gRbase` implements a few additional functions.

*Notice: Need an overview over these.*

## 2.2 More advanced graph operations

A moralized directed acyclic graph is obtained with

```
> moralize(dag11)

A graphNEL graph with undirected edges
Number of Nodes = 7
Number of Edges = 8
```

Testing for whether a graph is triangulated is based on Maximum Cardinality Search. If `character(0)` is returned the graph is not triangulated. Otherwise a linear ordering of the nodes is returned.

```
> mcs(ug11)


```

Triangulate an undirected graph by adding extra edges to the graph:

```
> tug1 <- triangulate(ug11)

A graphNEL graph with undirected edges
Number of Nodes = 7
Number of Edges = 8
```

A RIP ordering of the cliques of a triangulated graph can be obtained as:

```
> r <- rip(tug1)
> r

cliques
  1 : c a b
  2 : e a c
  3 : d c e
  4 : g f
separators
  1 :
  2 : a c
  3 : c e
  4 :
parents
  1 : 0
  2 : 1
  3 : 2
  4 : 0
```

There exist corresponding functions `moralizeMAT`, `mcsMAT`, `triangulateMAT` and `ripMAT` for graphs represented as adjacency matrices.

# 3 Coercion

Coercion between representations as a `graphNEL` object and an adjacency matrix can be done with the `as()` method from the `graph` package:

```
> as(ug11, "matrix")

  a b c d e f g
a 0 1 1 0 1 0 0
b 1 0 1 0 0 0 0
c 1 1 0 1 0 0 0
d 0 0 1 0 1 0 0
e 1 0 0 1 0 0 0
f 0 0 0 0 0 0 1
g 0 0 0 0 0 1 0

> as(ug11m, "graphNEL")

A graphNEL graph with undirected edges
Number of Nodes = 7
Number of Edges = 7
```

```
> as(dag11, "matrix")

  a b c d e g f
a 0 1 1 0 1 0 0
b 0 0 1 0 0 0 0
c 0 0 0 1 0 0 0
d 0 0 0 0 0 0 0
e 0 0 0 1 0 0 0
g 0 0 0 0 0 0 0
f 0 0 0 0 0 1 0

> as(dag11m, "graphNEL")

A graphNEL graph with directed edges
Number of Nodes = 7
Number of Edges = 7
```

# 4 Speed considerations

It is worth noticing that working with graphs representated as `graphNEL` objects is somewhat slower working with graphs represented as adjacency matrices. Consider for example coerction from a `graphNEL` object. This can be obtained with `as()` as shown above or by using `as.adjMAT()` from `gRbase`. A simple timing speaks for itself:

```
> system.time({
+     for (ii in 1:100) as(ug11, "matrix")
+ })

   user  system elapsed
   0.38    0.00    0.38

> system.time({
+     for (ii in 1:100) as.adjMAT(ug11)
+ })

   user  system elapsed
   0.03    0.00    0.03
```