RBGL: R interface to boost graph library

VJ Carey stvjc@channing.harvard.edu

March 31, 2003

Summary. A very preliminary implementation of an interface from R to the Boost Graph Library (BGL, an alternative to STL programming for mathematical graph objects) is presented. *This 2003 update employs the* graph *class of Bioconductor.*

1 Working with the Bioconductor graph class

An example object representing file dependencies is included, as shown in Figure 1.

```
> library(RBGL)

Loading required package: graph
Loading required package: Biobase
Welcome to Bioconductor
         Vignettes contain introductory material.  To view,
         simply type: openVignette()
         For details on reading vignettes, see
         the openVignette help page.
Creating a new generic function for "summary" in package
reposTools
Creating a new generic function for "print" in package
Ruuid

> data(FileDep)
> print(FileDep)

A graph with  directed  edges
Number of Nodes = 15
Number of Edges = 19
```

The tsort function will return the indices of vertices in topological sort order:

```
> plot(FileDep)
```

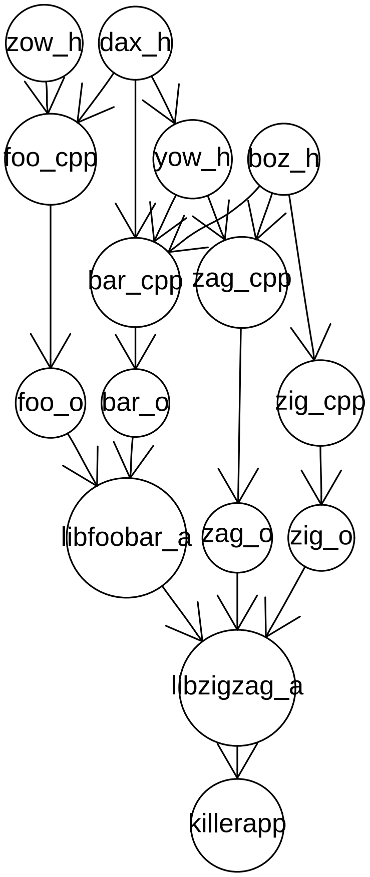

Figure 1: File dependency digraph example from Boost library.

```
> ts <- tsort(FileDep)
> print(nodes(FileDep)[ts + 1])

 [1] "zow_h"       "boz_h"       "zig_cpp"     "zig_o"       "dax_h"
 [6] "yow_h"       "zag_cpp"     "zag_o"       "bar_cpp"     "bar_o"
[11] "foo_cpp"     "foo_o"       "libfoobar_a" "libzigzag_a" "killerapp"
```

Note that if the input graph is not a DAG, BGL `topological_sort` will check this and throw 'not a dag'. This is crudely captured in the interface (a message is written to the console and zeroes are returned).

```
#FD2 <- FileDep
# now introduce a cycle
#FD2@edgeL[["bar_cpp"]]$edges <- c(8,1)
#tsort(FD2)
```