Drawing pedigree diagrams with R and graphviz

by Jing Hua Zhao

Human genetic studies often involve data collected from families and graphical display of them are necessary. The wide interest it has drawn over years led to many software packages, both commercial and noncommercial. A recent account of these packages is available ((Dudbridge et al., 2004)), and a very flexible package Madeline (`http://eyegene.ophthy.med.umich.edu/madeline/index.html`) is now released under the GNU General Public License. A comprehensive list of many packages, including the package LINKAGE for human parametric linkage analysis and GAS for some other analyses, can be seen at the linkage server at Rockefeller University (`http://linkage.rockefeller.edu`).

Here I describe two functions in R that are able to draw pedigree diagrams; the first being `plot.pedigree` in `kinship` developed S-PLUS by Terry Therneau and Beth Atkinson and ported to R by the author, and the second `pedtodot` in `gap` based on David Duffy's `gawk` script (`http://www2.qimr.edu.au/davidD/Course/pedtodot`) that requires `graphviz` (`http://www.graphviz.org`). Both are easy to use and can draw many pedigree diagrams quickly to a single file, therefore can serve as alternatives to some programs that only offer interactive use.

Representation of pedigrees

The key elements to store pedigrees using a database is via the so-called family trios each containing individual's, father's and mother's IDs. Founders, namely individuals whose parents are not in the pedigree, are sent to be zero or missing. Individual's gender (e.g. 1=male, 2=female) is included as auxiliary information, together with pedigree ID in order to maintain multiple pedigrees in a single database, each record of which indicating a node in the pedigree graph.

For instance, information for pedigree numbered 10081 in genetic analysis workshop 14 (GAW14, `http://www.gaworkshop.org`) is shown as follows.

```
pid id father mother sex affected
10081  1  2  3 2  1
10081  2  0  0 1  2
10081  3  0  0 2  1
10081  4  2  3 2  1
10081  5  2  3 2  2
10081  6  2  3 1  2
10081  7  2  3 2  2
10081  8  0  0 1  2
10081  9  8  4 1  2
10081 10  0  0 2  2
10081 11  2 10 2  2
10081 12  2 10 2  1
10081 13  0  0 1  2
10081 14 13 11 1  2
10081 15  0  0 1  2
10081 16 15 12 2  2
```

Here all IDs are integers with obvious meanings just described, and the variable `affected` indicates if an individual is alcoholic (1=nonalcoholic, 2=alcoholic) according to DSMIIIR and Feighner definition ALDX1 in the dataset.

In human genetic linkage studies, this is also called pre-makeped format since these IDs can also be string variables, e.g. individuals' names, and a utility program makeped in LINKAGE can be used to generate the serial integer IDs and perform simple checks on errors in family structure(s).

Suppose this is kept in a text file called 10081.pre, we use

```
pre <- read.table("10081.pre",header=T)
```

to read it into object *ped*.

The pedigree-drawing algorithm

Typically, in a pedigree diagram males and females are shown in squares and circles, respectively. Spouses can form marriage nodes from which nodes for children are derived. It is also customary to draw pedigree diagrams top down, so that children at a given generation could have children of their own in the next generation. This implies that the conceptually simple algorithm for pedigree drawing would involve sorting members of a pedigree by generation and align members of the same generation horizontally and those at different generations vertically. In other words, the family is drawn as a directed graph with members as nodes and ordered by their generation numbers. The algorithm could be more involved if there are marriage loops in the family, i.e. overlapping generations, or if the pedigree is too large to fit in a single page. More details on the algorithmic aspects of pedigree-drawing (Tores and Barillot (2001)) can be found for interested readers.

Fortunately, there are software publicly available that implements this algorithm. Among these the most notable is graphviz (`http://www.graphviz.org`). In the following section two approaches which implements the algorithm and which directly uses that in graphviz will be described.

Drawing pedigree diagrams with R and graphviz

via `plot.pedigree` in `kinship`

Package `kinship` was developed for linear mixed and mixed-effects Cox models of family data, but it has function `plot.pedigree` function for drawing pedigree diagrams. As an S3 function for class 'pedigree', it can be used as `plot` if supplied with argument with class 'pedigree'.

```
> library(kinship)
> attach(pre)
> ped <- pedigree(id,father,mother,sex,
+         affected)
> plot(ped)
```

This gives Figure 1. We can use R devices such as postscript to keep the diagram as an outside file, e.g., `postscript("10081.ps"); plot(ped); postscript()`. The postscript format is preferable since it can hold more than one pedigree diagrams.

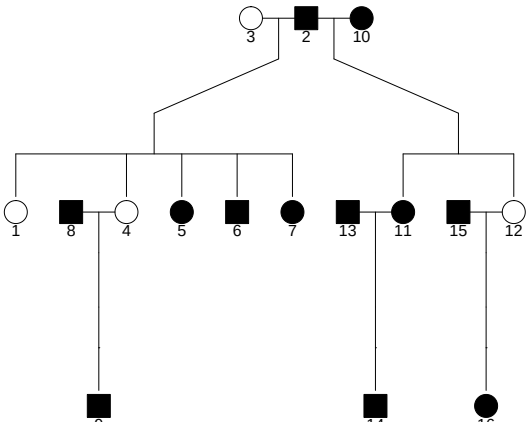

Figure 1: Pedigree 10081 by kinship

via `pedtodot` in `gap`

The diagram produced by `plot.pedigree` in `kinship` is a still image so that nodes in the pedigree graph can not be pulled and dragged. This might not be trivial to implement. However, we can use graphviz to achieve this. A nice feature of the package its ability to interpret dot language, which is in ASCII format that allows for text-editing. Keeping this in mind, we can simply generate a dot file for given pedigree(s) using dot syntax.

```
> library(gap)
> pedtodot(ped)
> dir()
```

By default, this generates `10081.dot` which can be used by dot (Figure 2).

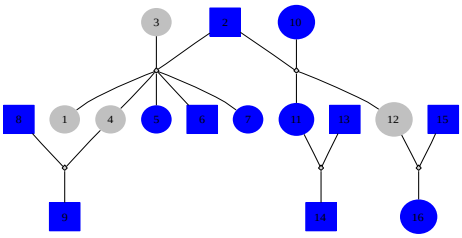

Figure 2: Pedigree 10081 by dot

When dir=forward is specified in the dot file, besides `dot` it also makes sense to use to neato leading to a more liberal graph (Figure 3). Its unusual looking makes it appealing for exposing the peeling algorithm in the likelihood calculation of pedigrees.

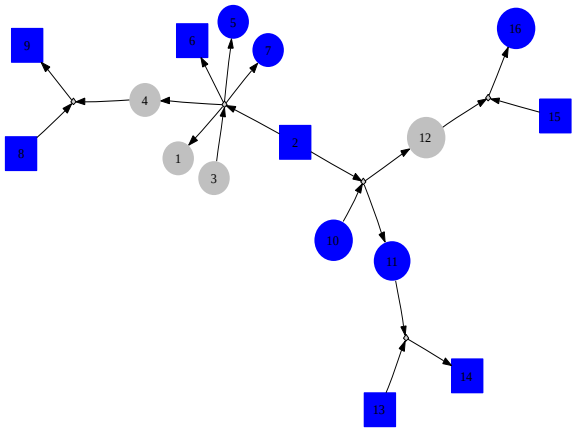

Figure 3: Pedigree 10081 by neato

A single file containing all pedigree diagrams can be generated through the use of R `sink` command and screen output using `sink=F` option in `pedtodot`. For example, a complete list of pedigree diagrams using pedigrees from GAW14 are available from the web sites `http://www.ucl.ac.uk/~rmjdjhz/r-progs.htm` and `http://www.hgmp.mrc.ac.uk/~jzhao/r-progs.htm`.

Summary and further remarks

Further information about the two functions is available from the packages themselves. Note that `plot.pedigree` in `kinship` does not require pedigree ID be specified, while `pedtodot` in `gap` does. Unlike `plot.pedigree`, `pedtodot` requires `graphviz` to visualize graphics, but can be edited with `dotty`, and be printed out in multiple pages when a pedigree diagram is too big to fit in a single page. Both can produce a set of pedigree diagrams in a single file.

Although `makeped` is a utility program in LINKAGE, this function is also available in package `gap`. If the pedigree file is in `post-makeped` format, then the option `makeped=T` can be used. However, `pedtodot` can also use string IDs, or file in the so-called GAS format in which gender can take values 'm', 'f', etc.

Most pedigrees in current studies have moderate sizes, therefore simple two-dimensional arrays will be sufficient to keep track of marriages and children from them, efficiency can be gained through use of abstract data structures such as linked-list or trees and it will not be difficult to achieve. At this point, `gawk` script may be more efficient since it can use strings to index arrays. In addition, other features will be benefited from further experiences of its use.

Finally, it is notable that `path.diagram` in the R package `sem` is designed to generate `dot` file to be used by `graphviz`, while a more elaborate implementation can be found in the Bioconductor (`http://www.bioconductor.org`) package `Rgraphviz`. As `graphviz` is open source, it should be possible to introduce more native `dotty`-like features in R. Although as yet human genetic linkage and genome-wide association analysis is not widely conducted with R, this might change in the near future, as has been demonstrated by the great success of the Bioconductor project. I believe the two R functions described in this note will be very useful to researchers in their genetic data analysis.

Jing Hua Zhao
University College London
`j.zhao@ucl.ac.uk`

Bibliography

F. Dudbridge, T. Carver, and G. Williams. Pelican: pedigree editor for linkage computer analysis. *Bioinformatics*, 20(14):2327–2328, 2004.

F. Tores and E. Barillot. The art of pedigree drawing: algorithmic aspects. *Bioinformatics*, 17(2):174–179, 2001.