

Notes on radagio

Christof Neumann

2018-07-17

This package is an R wrapper for the ADAGIO software (<http://ngonga.github.io/>, <https://github.com/ngonga/adagio>) that is originally written in Java by Axel Ngonga.

To run it, you need Java SE Development Kit 8, which is available here: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

To check whether you have Java installed already you need to run in your terminal/command line:

```
java -version
```

Installation

In order to install the **radagio** package please use the following commands in R. Note that the package file (currently **radagio_0.3.0.tar.gz**, download here: <https://doi.org/10.5281/zenodo.1280950>) does *not* need to be unpacked. Leave this file as it is and save it somewhere locally on your computer. Also, the file name may change in the future if the package is updated.

```
# install other necessary packages:
# if you already have some of them you don't need to re-install
# requires working internet connection
install.packages("jpeg")
install.packages("EloRating")
install.packages("network")
install.packages("EloChoice")
install.packages("sna")
install.packages("statnet")
install.packages("devtools")

# install radagio
# if you are on Windows:
install.packages("C:/downloads/rstuff/radagio_0.3.0.tar.gz", repos = NULL, type = "source")
# if you are on a Mac:
install.packages("~/Desktop/rstuff/radagio_0.3.0.tar.gz", repos = NULL, type = "source")
```

Alternatively, you can also install directly from GitHub via the **devtools** package:

```
library(devtools)
install_github("gobbios/radagio")
```

To test whether this worked, try:

```
library(radagio)
adagio(archie, bottomup = TRUE, preproc = TRUE, plotfig = TRUE)
```

This should result in output that contains the results of the ADAGIO software. Also, a figure of the ADAGIO ranking should be produced.

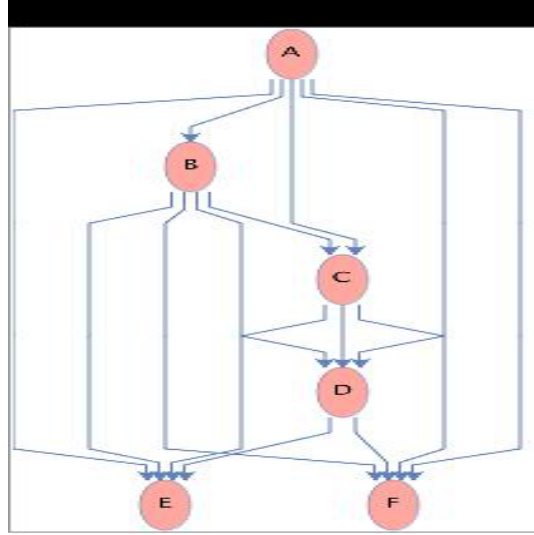
Examples

```
library(radagio)
library(EloRating)
```

Example 1 (Archie et al. 2006: family CB)

First, we reproduce the example DNH using elephant data (Archie, Morrison, Foley, Moss, & Alberts (2006); figure 7 in Douglas, Ngonga Ngomo, & Hohmann (2017)).

```
res <- adagio(archie, bottomup = TRUE, preproc = TRUE, plotfig = TRUE)
```



```
res$res
```

```
## ID adagio elo ds
## 1 A      1  1  1
## 2 B      2  3  2
## 3 C      3  2  3
## 4 D      4  4  4
## 5 E      5  5  5
## 6 F      5  6  6
```

The results of the ADAGIO algorithm (`adagio` column in `res$res`) match the results reported in the paper (p. 28, figure 7), i.e. E and F both have rank 5 (using the bottom-up ranking with preprocessing). The actual figure looks quite different from the graph in the paper, but also here it is apparent that E and F are at the bottom together and all other individuals occupy ‘individual’ ranks.

Now, let’s calculate Elo-ratings and David’s scores for that data set.

```
set.seed(123)
elores <- randomelo(archie, runs = 100)
```

```
DSD <- DS(archie, prop = "Dij")
DSD <- DSD[order(DSD$ID), ]
DSP <- DS(archie, prop = "Pij")
DSP <- DSP[order(DSP$ID), ]
```

And put everything together.

```
data.frame(res$res, meanelos = colMeans(elores[[1]]), DSD = round(DSD$normDS, 1), DSP = round(DSP$normDS, 1))
```

```
## ID adagio elo ds meanelos DSD DSP
## 1 A      1  1  1 1145.31 4.0 4.5
## 2 B      2  3  2  989.49 3.2 3.7
## 3 C      3  2  3 1060.67 2.8 2.8
## 4 D      4  4  4 1037.41 2.2 2.1
## 5 E      5  5  5  852.49 1.3 1.0
## 6 F      5  6  6  863.36 1.6 0.8
```

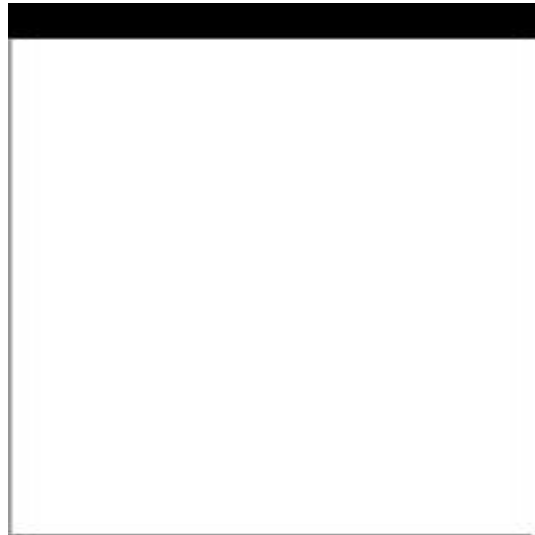
This table represents the results produced by the ADAGIO software (columns 2, 3 and 4) and the results of recalculations of Elo-ratings and David’s scores with our package (`EloRating`).

There are a number of inconsistencies here. The mean Elo-ratings of B and D indicate that they should be ranked 4 and 3 respectively, but this does not match with the results of ADAGIO’s calculations of Elo-ratings where the two

individuals hold ranks 3 and 4. Similarly, the ranks of E and F do not match for David's score (at least if David's score is based on winning proportions corrected for chance (D_{ij}), de Vries, Stevens, & Vervaecke (2006); column DSD).

Example 2 (Douglas et al. 2017, figure 1 and 3b)

```
data(dnh)
res <- adagio(dnh, bottomup = TRUE, preproc = TRUE, plotfig = TRUE)
```



```
res$res
```

```
## ID adagio elo ds
## 1 a      1  2  3
## 2 b      4  5  6
## 3 c      2  3  1
## 4 d      3  4  4
## 5 e      4  6  5
## 6 f      2  1  2
```

```
set.seed(123)
elores <- randomelo(dnh, runs = 10)
```

```
DSD <- DS(dnh, prop = "Dij")
DSD <- DSD[order(DSD$ID), ]
DSP <- DS(dnh, prop = "Pij")
DSP <- DSP[order(DSP$ID), ]
```

```
data.frame(res$res, meanelos = colMeans(elores[[1]]), DSD = round(DSD$normDS, 2), DSP = round(DSP$normDS, 2))
```

```
## ID adagio elo ds meanelos DSD DSP
## 1 a      1  2  3  1013.2 2.74 2.83
## 2 b      4  5  6  1077.2 1.92 1.33
## 3 c      2  3  1   934.3 2.76 3.00
## 4 d      3  4  4  1046.6 2.51 2.67
## 5 e      4  6  5   897.7 2.33 2.17
## 6 f      2  1  2   953.7 2.74 3.00
```

Here again we see some discrepancies. For example the three individuals (b, d, a) with the highest mean Elo-ratings don't have ranks 1 to 3, but 5, 4 and 2. The two individuals with the highest David's scores (c and f, each with DS=3.00 (column DSP)) have different ranks (column ds) according to the ADAGIO software although they should be assigned the same rank.

Preprocessing

The following simulation indicates that this behaviour is not due to the preprocessing although it only checks for results produced with the bottom-up option. Here we submit a number of random matrices to ADAGIO and then check whether the preprocessing option affects the ADAGIO ranks (it does, as expected) and the ranks from Elo-rating and David's scores (it appears not to). The resulting table (`res`) shows in the first column the summed absolute difference in ranks between the two preprocessing options, and the remaining two columns show the same for Elo and DS. It appears that the ADAGIO software produces consistent (albeit questionable, see above) rankings for Elo and DS regardless of preprocessing, i.e. the columns `ELOdiff` and `DSdiff` only contain zeros (i.e. rankings are identical with and without preprocessing), while rankings generated with the ADAGIO method differ depending on whether preprocessing was applied.

```
res <- matrix(ncol = 3, nrow = 100, NA)
colnames(res) <- c("ADAGIOdiff", "ELOdiff", "DSdiff")
for(i in 1:nrow(res)) {
  mat <- matrix(rpois(36, round(runif(1, 1, 10))), ncol = 6)
  diag(mat) <- 0
  colnames(mat) <- rownames(mat) <- LETTERS[1:6]
  # calculate ADAGIO ranks with and without preprocessing
  res1 <- adagio(mat, bottomup = TRUE, preproc = TRUE, plotfig = FALSE)
  res2 <- adagio(mat, bottomup = TRUE, preproc = FALSE, plotfig = FALSE)
  # absolute differences
  r <- abs(as.matrix(res1$res[, 2:4]) - as.matrix(res2$res[, 2:4]))
  # store results
  res[i, ] <- colSums(r)
  # clean up
  rm(mat, res1, res2, r)
  # a crude progress bar
  cat(i, "\n")
}

# summarize results
apply(res, MARGIN = 2, FUN = table)
```

Pyramid example

The figure (figure 1) produced by the following code illustrates some severe problems with the ADAGIO software with regard to how the software calculates ranks for Elo-rating and David's scores.

```
# generate interaction matrix
mat <- matrix(ncol = 6, nrow = 6, 2)
colnames(mat) <- rownames(mat) <- LETTERS[1:6]
diag(mat) <- 0
mat[1, 2:6] <- 4
mat[, 1] <- 0

# generate Elo-ratings from 500 randomized sequences
set.seed(123)
res <- randomelo(mat, 500)
x <- data.frame(ID = colnames(res[[1]]), avg = round(colMeans(res[[1]]), 1),
               lower = apply(res[[1]], 2, min), upper = apply(res[[1]], 2, max))

# set up figure
par(mfrow = c(2, 3), mar = c(3, 4, 2, 1))

# data matrix
plot(1:6, 1:6, "n", ylim = c(6.5, 0.5), xlim = c(0.5, 6.5), axes = FALSE, xlab = "", ylab = "winner",
     main = "interaction matrix", cex.main = 0.8)
title(xlab = "loser", line = 1)
for(i in 1:6) text(1:6, rep(i, 6), labels = as.character(mat[i, ]), cex = 1)
axis(side = 3, line = -1, tcl = 0, lwd = NA, at = 1:6, labels = LETTERS[1:6], cex.axis = 0.8)
axis(side = 2, line = -0.5, tcl = 0, lwd = NA, at = 1:6, labels = LETTERS[1:6], las = 1, cex.axis = 0.8)
box()
addfiglabel("(a)")

# mean Elo-ratings from Elo-rating package
```

```

plot(1:6, x$avg, ylim = range(x[, 2:4]), xlab = "", ylab = "Elo-rating (mean and range)",
     axes = FALSE, pch = 16, cex = 1.5, cex.lab = 0.8, main = "randomized Elo-rating", cex.main = 0.8)
title(xlab = "ID", line = 1)
segments(1:6, x$lower, 1:6, x$upper)
axis(side = 1, line = -1, tcl = 0, lwd = NA, at = 1:6, labels = LETTERS[1:6], cex.axis = 0.8)
axis(side = 2, las = 1, cex.axis = 0.8)
box()
addfiglabel("(b)")

# David's score from Elo-rating package
d <- DS(mat)
plot(1:6, d$normDS, ylim = c(-0.5, 5.5), xlab = "", ylab = "normalized David's score",
     axes = FALSE, pch = 16, cex = 1.5, cex.lab = 0.8, main = "David's score", cex.main = 0.8)
title(xlab = "ID", line = 1)
axis(side = 1, line = -1, tcl = 0, lwd = NA, at = 1:6, labels = LETTERS[1:6], cex.axis = 0.8)
axis(side = 2, las = 1, cex.axis = 0.8)
box()
addfiglabel("(c)")

# ADAGIO ranks from ADAGIO software
adagiores <- adagio(mat, bottomup = TRUE)
aranks <- adagiores$res$adagio
plot(1:6, aranks, ylim = c(6.5, 0.5), xlab = "", ylab = "ADAGIO rank (bottom-up approach)",
     axes = F, pch = 16, cex = 1.5, cex.lab = 0.8, main = "ADAGIO", cex.main = 0.8)
title(xlab = "ID", line = 1)
axis(side = 1, line = -1, tcl = 0, lwd = NA, at = 1:6, labels = LETTERS[1:6], cex.axis = 0.8)
axis(2, las = 1, cex.axis = 0.8)
box()
addfiglabel("(d)")

# Elo-rating ranks from ADAGIO software
eranks <- adagiores$res$elo
plot(1:6, eranks, ylim = c(6.5, 0.5), xlab = "", ylab = "ADAGIO-generated ranks", axes = F,
     pch = 16, cex = 1.5, cex.lab = 0.8, main = "ADAGIO-generated Elo-rating", cex.main = 0.8)
title(xlab = "ID", line = 1)
axis(side = 1, line = -1, tcl = 0, lwd = NA, at = 1:6, labels = LETTERS[1:6], cex.axis = 0.8)
axis(2, las = 1, at = 1:6, labels = 1:6, cex.axis = 0.8)
box()
addfiglabel("(e)")

# David's score ranks from ADAGIO software
dranks <- adagiores$res$ds
plot(1:6, dranks, ylim = c(6.5, 0.5), xlab = "", ylab = "ADAGIO-generated ranks", axes = F,
     pch = 16, cex = 1.5, cex.lab = 0.8, main = "ADAGIO-generated David's score", cex.main = 0.8)
title(xlab = "ID", line = 1)
axis(side = 1, line = -1, tcl = 0, lwd = NA, at = 1:6, labels = LETTERS[1:6], cex.axis = 0.8)
axis(2, las = 1, at = 1:6, labels = 1:6, cex.axis = 0.8)
box()
addfiglabel("(e)")

```

References

- Archie, E. A., Morrison, T. A., Foley, C. A. H., Moss, C. J., & Alberts, S. C. (2006). Dominance rank relationships among wild female African elephants, *Loxodonta africana*. *Animal Behaviour*, 71, 117–127. <https://doi.org/10.1016/j.anbehav.2005.03.023>
- de Vries, H., Stevens, J. M. G., & Vervaecke, H. (2006). Measuring and testing the steepness of dominance hierarchies. *Animal Behaviour*, 71, 585–592. <https://doi.org/10.1016/j.anbehav.2005.05.015>
- Douglas, P. H., Ngonga Ngomo, A.-C., & Hohmann, G. (2017). A novel approach for dominance assessment in gregarious species: ADAGIO. *Animal Behaviour*, 123, 21–32. <https://doi.org/10.1016/j.anbehav.2016.10.014>

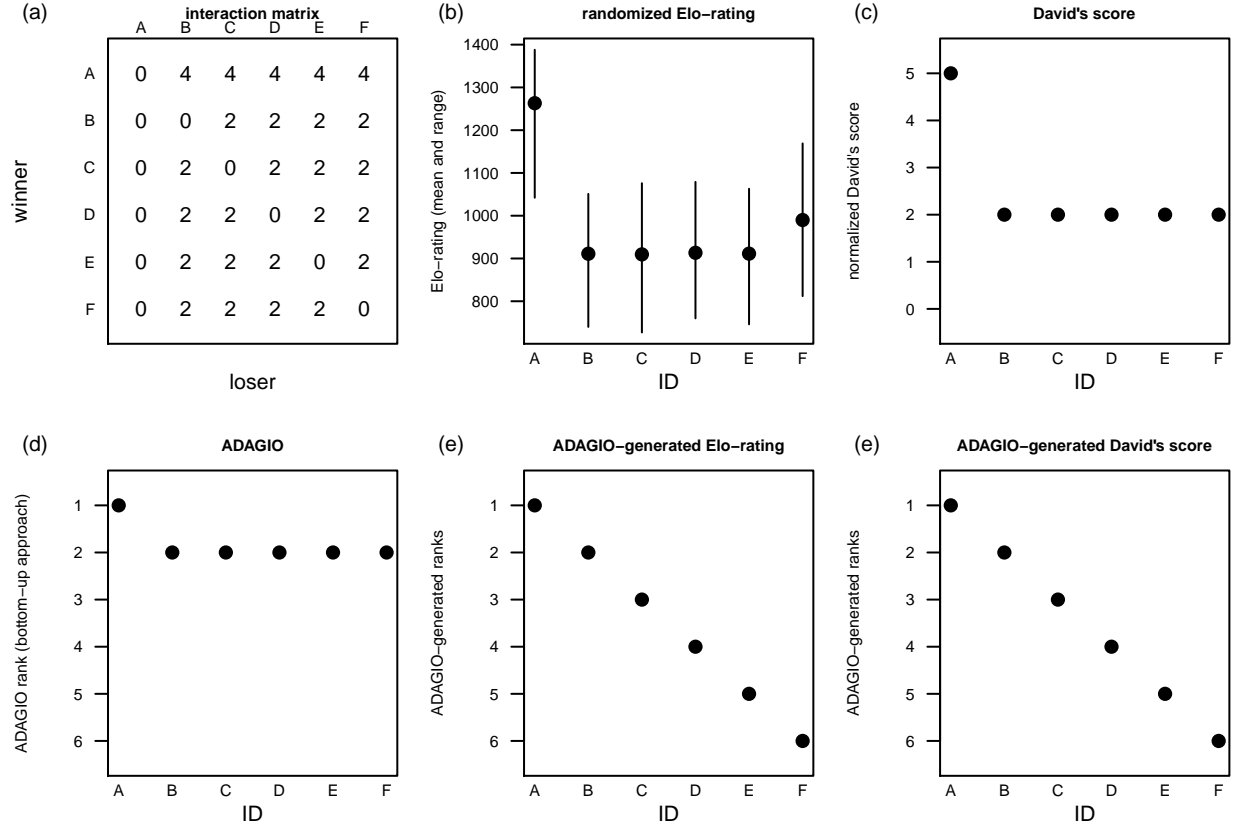


Figure 1: Results of ADAGIO software compared to results from **EloRating**. This example comprises six individuals, in which one individual (A) is dominant over all other individuals, while all remaining dyads have tied relationships (a). ADAGIO returns the intended pyramidal structure (d). Average Elo-ratings from 500 random sequences (with lines indicating range), in which the interactions in the matrix may have occurred are presented in (b), and David's scores calculated directly from the matrix are in (c). When subjected to the ADAGIO software (v. 1.1) and inspecting the returned ranks of individuals with Elo-rating and David's score, individuals are ranked in a linear order, i.e without tied/shared ranks (e) and (f).