Chimpanzee females queue but males strive for social status

Steffen Foerster, Mathias Franz, Carson M. Murray, Ian C. Gilby, Joseph T. Feldblum, Kara Schroepfer-Walker, Anne E. Pusey

```
################################################################################

# Model 1 -- constant initial socres + fitting k

################################################################################


elo.model1 <- function(par, burn_in=100, init_elo = 1000, IA_data, all_ids, return_likelihood = T)

{

  k <- par


  # Initialize output columns

  if (!return_likelihood) IA_data$elo_l_before <- IA_data$elo_w_before <- IA_data$elo_l_after <-
IA_data$elo_w_after <- NA


  # Set intitial elo scores

  currentELO <- rep(init_elo,length(all_ids))

  names(currentELO) <- all_ids


  # Initialize the log likelihood

  L <- 0


  # Start loop

  for(i in 1:nrow(IA_data))

  {

    ind1 <- which(names(currentELO)==IA_data$Winner[i])

    ind2 <- which(names(currentELO)==IA_data$Loser[i])


    if (!return_likelihood)

    {

      IA_data$elo_w_before[i] <- currentELO[ind1]

      IA_data$elo_l_before[i] <- currentELO[ind2]

    }


    # calculate predited winning probablity of the winner
```

```
  p_win <- 1/(1+exp(-.01*(currentELO[ind1] - currentELO[ind2])))

  # Calculation of new ELO scores
  if (i <= burn_in)   # during burn-in period all k values are fixed to 100
  {
    currentELO[ind1] <- currentELO[ind1] + 100 * (1 - p_win)  # new Elo score of the Winner
    currentELO[ind2] <- currentELO[ind2] - 100 * (1 - p_win)  # new Elo score of the Loser
  }
  else  # after the burn-in period fitted k values are used
  {
    currentELO[ind1] <- currentELO[ind1] + exp(k) * (1 - p_win)  # new Elo score of the Winner
    currentELO[ind2] <- currentELO[ind2] - exp(k) * (1 - p_win)  # new Elo score of the Loser
  }

  # write calculated elo scores to output columns
  if (!return_likelihood)
  {
    IA_data$elo_w_after[i] <- currentELO[ind1]
    IA_data$elo_l_after[i] <- currentELO[ind2]
  }

  # Update log likelihood
  if (i > burn_in) L <- L + log(p_win)
 }

 if (return_likelihood) return(-1*L)
 else return(IA_data)
}
```

```
##############################################################################
# Model 2 -- initial scores at hierarchy bottom + fitting k
##############################################################################
elo.model2 <- function(par, burn_in=100, init_elo = 0, IA_data, pres_data, all_ids, return_likelihood = T)
{
  k <- par

  # Initialize output columns
  if (!return_likelihood) IA_data$elo_l_before <- IA_data$elo_w_before <- IA_data$elo_l_after <-
IA_data$elo_w_after <- NA

  # Set intitial elo scores
  currentELO <- rep(init_elo,length(all_ids))
  names(currentELO) <- all_ids

  # Initialize the log likelihood
  L <- 0

  # Start loop
  for(i in 1:nrow(IA_data))
  {
    ind1 <- which(names(currentELO)==IA_data$Winner[i])
    ind2 <- which(names(currentELO)==IA_data$Loser[i])

    if (!return_likelihood)
    {
      IA_data$elo_w_before[i] <- currentELO[ind1]
      IA_data$elo_l_before[i] <- currentELO[ind2]
    }

    # calculate predited winning probablity of the winner
    p_win <- 1/(1+exp(-.01*(currentELO[ind1] - currentELO[ind2])))
```

```
  # Calculation of new ELO scores
  if (i <= burn_in)   # during burn-in period all k values are fixed to 100
  {
    currentELO[ind1] <- currentELO[ind1] + 100 * (1 - p_win)  # new Elo score of the Winner
    currentELO[ind2] <- currentELO[ind2] - 100 * (1 - p_win)  # new Elo score of the Loser
  }
  else  # after the burn-in period fitted k values are used
  {
    currentELO[ind1] <- currentELO[ind1] + exp(k) * (1 - p_win)  # new Elo score of the Winner
    currentELO[ind2] <- currentELO[ind2] - exp(k) * (1 - p_win)  # new Elo score of the Loser
  }

  #rescale Elo scores of present individuals so that the smallest Elo score is 0
  presence <- pres_data[pres_data$Date == IA_data$Date[i], 2:ncol(pres_data)]==1
  currentELO[presence] <- currentELO[presence] - min(currentELO[presence])

  # write calculated elo scores to output columns
  if (!return_likelihood)
  {
    IA_data$elo_w_after[i] <- currentELO[ind1]
    IA_data$elo_l_after[i] <- currentELO[ind2]
  }

  # Update log likelihood
  if (i > burn_in) L <- L + log(p_win)
 }

 if (return_likelihood) return(-1*L)
 else return(IA_data)
}
```

```
##############################################################################

# Model 3 -- fitting of initial scores  and  k

##############################################################################

elo.model3 <- function(par, IA_data, all_ids, return_likelihood = T)

{

  k <- par[1]


  init_elo <- par[2:length(par)]


  # Initialize output columns

  if (!return_likelihood) IA_data$elo_l_before <- IA_data$elo_w_before <- IA_data$elo_l_after <-
IA_data$elo_w_after <- NA


  # Set intitial elo scores

  currentELO <- c(init_elo)


  names(currentELO) <- all_ids


  # Initialize the log likelihood

  L <- 0


  # Start loop

  for(i in 1:nrow(IA_data))

  {

    ind1 <- which(names(currentELO)==IA_data$Winner[i])

    ind2 <- which(names(currentELO)==IA_data$Loser[i])


    if (!return_likelihood)

    {

      IA_data$elo_w_before[i] <- currentELO[ind1]

      IA_data$elo_l_before[i] <- currentELO[ind2]

    }
```

```
   # calculate predited winning probablity of the winner
   p_win <- 1/(1+exp(-.01*(currentELO[ind1] - currentELO[ind2])))

   # Calculation of new ELO scores
   currentELO[ind1] <- currentELO[ind1] + exp(k) * (1 - p_win)  # new Elo score of the Winner
   currentELO[ind2] <- currentELO[ind2] - exp(k) * (1 - p_win)  # new Elo score of the Loser


   # write calculated elo scores to output columns
   if (!return_likelihood)
   {
     IA_data$elo_w_after[i] <- currentELO[ind1]
     IA_data$elo_l_after[i] <- currentELO[ind2]
   }

   # Update log likelihood
   L <- L + log(p_win)
 }

 if (return_likelihood) return(-1*L)
 else return(IA_data)
}



################################################################################
# Read data
################################################################################

# females
female_ago <- read.csv(file.choose(), header=T, stringsAsFactors=F, sep=",")
```

```
female_ago$Winner <- as.character(female_ago$Winner)

female_ago$Loser <- as.character(female_ago$Loser)

female_ago$Date <- as.character(female_ago$Date)


female_presence <- read.csv(file.choose(), header=T, check.names=F, stringsAsFactors=F, sep=",")

female_presence$Date <- as.character(female_presence$Date)


## vector with female IDs

all_females <- c(1:44)



# males

male_ago <- read.csv(file.choose(), header=T, stringsAsFactors=F, sep=",")


male_ago$Winner <- as.character(male_ago$Winner)

male_ago$Loser <- as.character(male_ago$Loser)

male_ago$Date <- as.character(male_ago$Date)


male_presence <- read.csv(file.choose(), header=T, check.names=F, stringsAsFactors=F, sep=",")

male_presence$Date <- as.character(male_presence$Date)


## vector with male IDs

all_males <- c(1:22)



# table with results of model fitting

results_f <- data.frame('model' = 1:3, 'convergence' = NA, 'AIC' = NA, 'delta_AIC' = NA, 'k' = NA,
'pred_accuracy'=NA)

results_m <- data.frame('model' = 1:3, 'convergence' = NA, 'AIC' = NA, 'delta_AIC' = NA, 'k' = NA,
'pred_accuracy'=NA)
```

```
################################################################################

# Model fitting

################################################################################

## females ##

# Fitting model 1

res_fem_model1 <- optim(par=5, burn_in=100, elo.model1, all_ids = all_females, IA_data = female_ago, return_likelihood=T, method='Brent', lower=-10, upper=10)


results_f$convergence[1] <- res_fem_model1$convergence

results_f$AIC[1] <- res_fem_model1$value * 2 + 2

results_f$k[1] <- exp(res_fem_model1$par)


# Fitting model 2

res_fem_model2 <- optim(par=5, burn_in=100, elo.model2, pres_data = female_presence, all_ids = all_females, IA_data = female_ago, return_likelihood=T, method='Brent', lower=-10, upper=10)


results_f$convergence[2] <- res_fem_model2$convergence

results_f$AIC[2] <- res_fem_model2$value * 2 + 2

results_f$k[2] <- exp(res_fem_model2$par)


# Fitting model 3

res_fem_model3 <- optim(par=c(5, rep(0, length(all_females))), elo.model3, all_ids = all_females, IA_data = female_ago[101:nrow(female_ago),], return_likelihood=T, method='BFGS', control = list(maxit = 10000, reltol=1e-10))


results_f$convergence[3] <- res_fem_model3$convergence

results_f$AIC[3] <- res_fem_model3$value * 2 + 2 * (length(all_females) + 1)

results_f$k[3] <- exp(res_fem_model3$par[1])


results_f$delta_AIC <- results_f$AIC - min(results_f$AIC)

results_f
```

```
## males ##

# Fitting model 1

res_m_model1 <- optim(par=5, burn_in=100, elo.model1, all_ids = all_males, IA_data = male_ago,
return_likelihood=T, method='Brent', lower=-10, upper=10)


results_m$convergence[1] <- res_m_model1$convergence

results_m$AIC[1] <- res_m_model1$value * 2 + 2

results_m$k[1] <- exp(res_m_model1$par)


# Fitting model 2

res_m_model2 <- optim(par=5, burn_in=100, elo.model2, pres_data = male_presence, all_ids =
as.character(all_males), IA_data = male_ago2, return_likelihood=T, method='Brent', lower=-10, upper=10)


results_m$convergence[2] <- res_m_model2$convergence

results_m$AIC[2] <- res_m_model2$value * 2 + 2

results_m$k[2] <- exp(res_m_model2$par)


# Fitting model 3

res_m_model3 <- optim(par=c(5, rep(0, length(all_males))), elo.model3, all_ids = all_males, IA_data =
male_ago[101:nrow(male_ago),], return_likelihood=T, method='BFGS', control = list(maxit = 10000, reltol=1e-10))


results_m$convergence[3] <- res_m_model3$convergence

results_m$AIC[3] <- res_m_model3$value * 2 + 2 * (length(all_males) + 1)

results_m$k[3] <- exp(res_m_model3$par[1])


results_m$delta_AIC <- results_m$AIC - min(results_m$AIC)
```