

# Pisaura simulations

06/07/2021

## Summary

Leading with the following fitness functions for females (virgins and mated) and males,

$$w_{\text{virgin}} = a_s Q_{av} + f_s Q_{fv} + g_s Q_{gv},$$

$$w_{\text{mated}} = a_s Q_{am} + f_s Q_{fm} + g_s Q_{gm},$$

$$w_{\text{male}} = n_s (p_s M_v + (1 - p_s) M_m).$$

In the above,  $M_v$  and  $M_m$  are the payoffs for males from mating with virgin and non-virgin females, respectively,

$$M_v = a_s P_{av} + f_s P_{fv} + g_s P_{gv},$$

$$M_m = a_s P_{am} + f_s P_{fm} + g_s P_{gm}.$$

The values  $n_s$  and  $p_s$  give the number of females a male encounters on average and the proportion of females that are unmated, respectively. Could this be subsumed somehow in a single parameter for density,  $N_s$ ?

## Variable definitions

Variable	Description
$a_s$	Probability of male playing no gift
$f_s$	Probability of male playing worthless gift
$g_s$	Probability of male playing genuine gift
$Q_{av}$	Payoff to virgin female given no-gift male
$Q_{fv}$	Payoff to virgin female given worthless gift male
$Q_{gv}$	Payoff to virgin female given genuine gift male
$Q_{am}$	Payoff to mated female given no-gift male
$Q_{fm}$	Payoff to mated female given worthless gift male
$Q_{gm}$	Payoff to mated female given genuine gift male
$P_{av}$	Payoff to male for no gift to virgin female
$P_{fv}$	Payoff to male for worthless gift to virgin female
$P_{gv}$	Payoff to male for genuine gift to virgin female
$P_{am}$	Payoff to male for no gift to mated female
$P_{fm}$	Payoff to male for worthless gift to mated female
$P_{gm}$	Payoff to male for genuine gift to mated female

Note that  $1 = a + f + g$ . We had  $b$  used to represent fitness benefits for females obtaining different gifts, but I think that these can just be encapsulated in  $Q$  values? The same with the  $c$  values, which were being used to represent costs on top of the payoffs; I think that the payoffs are probably enough?

I think one thing that we will want to do is allow population density to vary, thereby causing  $p_s$  and  $n_s$  to vary too. For fixed values of population density, it is difficult for me to see why  $a_s$ ,  $f_s$ , or  $g_s$  would be expected to change (assuming we want them to?), but the optimal values of each will surely depend on population density, which will affect encounter rate.

I'm wondering if we can define fitness (payoffs) as a function of male behaviours ( $a_s$ ,  $f_s$ , and  $g_s$ ) and density ( $N_s$ ),  $W_{female}(a, f, g, N)$  and  $W_{male}(a, f, g, N)$ , with the idea that the probability of a female being unmated changes with  $N$ . The idea could be that  $a$ ,  $f$ , and  $g$  will evolve due to  $W_{male}$ , while  $N$  will change with absolute female fitness  $W_{female}$ . Alternatively, maybe it would make sense to understand male strategy given a fixed  $N$ , assuming that  $N$  is fixed for ecological reasons unrelated to male strategy? This kind of gets us back to the original approach, which isn't necessarily a bad thing, though it's not really game-theoretic so much as the optimisation of a phenotype ( $a$ ,  $f$ , and  $g$ ). Hence, there is not really a need for complicated strategies of earlier modelling when we can look for an optimisation of  $a$ ,  $f$ , and  $g$  values.

## The individual-based model

Here is the code for an individual-based model of the system.

```
make_inds <- function(N = 100, Qav = 1, Qfv = 1, Qgv = 2, Qam = 1, Qfm = 1,
                     Qgm = 1, Pav = 2, Pfv = 1, Pgv = 1, Pam = 1, Pfm = 1,
                     Pgm = 1){

  inds      <- matrix(data = 0, nrow = N, ncol = 19);
  as_raw    <- runif(n = N, min = 0, max = 1);
  fs_raw    <- runif(n = N, min = 0, max = 1);
  gs_raw    <- runif(n = N, min = 0, max = 1);
  sums      <- as_raw + fs_raw + gs_raw;
  as        <- as_raw / sums;
  fs        <- fs_raw / sums;
  gs        <- gs_raw / sums;

  inds[, 1] <- c(rep(0, 0.5*N), rep(1, 0.5*N));
  inds[, 2] <- 0;
  inds[, 3] <- 0;
  inds[, 4] <- 0;
  inds[, 5] <- as;
  inds[, 6] <- fs;
  inds[, 7] <- gs;
  inds[, 8] <- Qav;
  inds[, 9] <- Qfv;
  inds[, 10] <- Qgv;
  inds[, 11] <- Qam;
  inds[, 12] <- Qfm;
  inds[, 13] <- Qgm;
  inds[, 14] <- Pav;
  inds[, 15] <- Pfv;
  inds[, 16] <- Pgv;
  inds[, 17] <- Pam;
  inds[, 18] <- Pfm;
  inds[, 19] <- Pgm;
```

```

    return(inds);
}

W_female <- function(a, f, g, Qav, Qfv, Qgv, Qam, Qfm, Qgm, mated){
  if(mated == 0){
    payoff <- (a * Qav) + (f * Qfv) + (g * Qgv);
  }else{
    payoff <- (a * Qam) + (f * Qfm) + (g * Qgm);
  }
  return(payoff);
}

W_male <- function(a, f, g, Pav, Pfv, Pgv, Pam, Pfm, Pgm, mated){
  if(mated == 0){
    payoff <- (a * Pav) + (f * Pfv) + (g * Pgv);
  }else{
    payoff <- (a * Pam) + (f * Pfm) + (g * Pgm);
  }
  return(payoff);
}

encounter <- function(inds, male_row, female_row){

  a      <- inds[male_row, 5];
  f      <- inds[male_row, 6];
  g      <- inds[male_row, 7];
  Qav    <- inds[male_row, 8];
  Qfv    <- inds[male_row, 9];
  Qgv    <- inds[male_row, 10];
  Qam    <- inds[male_row, 11];
  Qfm    <- inds[male_row, 12];
  Qgm    <- inds[male_row, 13];
  Pav    <- inds[male_row, 14];
  Pfv    <- inds[male_row, 15];
  Pgv    <- inds[male_row, 16];
  Pam    <- inds[male_row, 17];
  Pfm    <- inds[male_row, 18];
  Pgm    <- inds[male_row, 19];
  mated  <- inds[female_row, 2];

  fem_fitness <- W_female(a, f, g, Qav, Qfv, Qgv, Qam, Qfm, Qgm, mated);
  mal_fitness <- W_male(a, f, g, Qav, Qfv, Qgv, Qam, Qfm, Qgm, mated);

  # One more mating has occurred
  inds[female_row, 2] <- inds[female_row, 2] + 1;
  inds[male_row, 2]   <- inds[male_row, 2]   + 1;

  # Insert the fitness of each
  inds[female_row, 3] <- inds[female_row, 3] + fem_fitness;
  inds[male_row, 3]   <- inds[male_row, 3]   + mal_fitness;

  return(inds);
}

```

```

ind_encounters <- function(inds, encounter_rate){

  while(encounter_rate > 0){
    #sample male
    N <- dim(inds)[1];
    i <- sample(x = 1:N, size = 1);
    while(inds[i] == 0){
      i <- sample(x = 1:N, size = 1);
    }

    #sample female
    j <- sample(x = 1:N, size = 1);
    while(inds[j] == 1){
      j <- sample(x = 1:N, size = 1);
    }
    inds <- encounter(inds, i, j);

    encounter_rate <- encounter_rate - 1;
  }

  return(inds);
}

reproduce <- function(inds){

  females <- inds[inds[,1] == 0,];
  males <- inds[inds[,1] == 1,];

  female_w <- females[, 3] / sum(females[, 3]);
  male_w <- males[, 3] / sum(males[, 3]);

  cols <- dim(inds)[2]
  N_f <- dim(females)[1];
  N_m <- dim(males)[1];

  off_F <- sample(x = 1:N_f, size = N_f, replace = TRUE, prob = female_w);
  off_M <- sample(x = 1:N_m, size = N_m, replace = TRUE, prob = male_w);

  new_F <- females[off_F, ];
  new_M <- males[off_M, ];

  new_inds <- matrix(data = NA, nrow = N_f + N_m, ncol = cols);

  new_inds[1:N_f, ] <- new_F;
  new_inds[(N_f + 1):(N_f + N_m), ] <- new_M;

  new_inds[, 2:3] <- 0;

  return(new_inds);
}

mutation <- function(inds){

```

```

N      <- dim(inds)[1];
mu_dat <- rbinom(n = 3 * N, size = 1, prob = 0.01);
mu_mat <- matrix(data = mu_dat, nrow = N);
mu_vals <- rnorm(n = 3 * N, mean = 0, sd = 0.02);
mv_mat <- matrix(data = mu_vals, nrow = N);
change <- mv_mat * mu_mat;

inds[, 5:7] <- inds[, 5:7] + change;
inds[inds[, 5] < 0] <- 0;
inds[inds[, 6] < 0] <- 0;
inds[inds[, 7] < 0] <- 0;

col_sum <- apply(X = inds[, 5:7], MARGIN = 1, FUN = sum);

inds[, 5] <- inds[, 5] / col_sum;
inds[, 6] <- inds[, 6] / col_sum;
inds[, 7] <- inds[, 7] / col_sum;

return(inds);
}

run_sim <- function(N = 1000, gens = 2000, print_gen = TRUE, Qav = 1, Qfv = 1,
                    Qgv = 2, Qam = 1, Qfm = 1, Qgm = 1, Pav = 2, Pfv = 1,
                    Pgv = 1, Pam = 1, Pfm = 1, Pgm = 1, encounter_rate = 200){

  inds      <- make_inds(N = N, Qav, Qfv, Qgv, Qam, Qfm, Qgm, Pav, Pfv, Pgv,
                        Pam, Pfm, Pgm);
  ind_hist <- matrix(data = NA, nrow = gens, ncol = 4);
  for(i in 1:gens){
    inds      <- ind_encounters(inds, encounter_rate);
    inds      <- reproduce(inds);
    inds      <- mutation(inds);
    ind_hist[i, 1] <- i; # Get just male freqs
    ind_hist[i, 2] <- mean(inds[inds[,1] == 1, 5]);
    ind_hist[i, 3] <- mean(inds[inds[,1] == 1, 6]);
    ind_hist[i, 4] <- mean(inds[inds[,1] == 1, 7]);
    if(print_gen == TRUE){
      print(ind_hist[i, ]);
    }
  }
  return(ind_hist);
}

replicate_sims <- function(N = 1000, gens = 200, print_end = TRUE, reps = 10,
                           Qav = 1, Qfv = 1, Qgv = 2, Qam = 1, Qfm = 1,
                           Qgm = 1, Pav = 2, Pfv = 1, Pgv = 1, Pam = 1, Pfm = 1,
                           Pgm = 1, encounter_rate = 200){

  rep_results <- NULL;
  for(i in 1:reps){
    sim_res      <- run_sim(N = N, gens = gens, print_gen = FALSE, Qav,

```

```

                                Qfv, Qgv, Qam, Qfm, Qgm, Pav, Pfv, Pgv, Pam,
                                Pfm, Pgm , encounter_rate);
rep_results[[i]] <- sim_res;
if(print_end == TRUE){
  print(sim_res[gens, ]);
}
}
return(rep_results);
}

```

I have not started to play around with actual parameter values, so here is what I used (see the defaults above).

```

N      <- 1000; # Number of females and males;
Qav    <- 1; # Payoff to virigin female given no-gift male
Qfv    <- 1; # Payoff to virigin female given worthless gift male
Qgv    <- 2; # Payoff to virigin female given genuine gift male
Qam    <- 1; # Payoff to mated female given no-gift male
Qfm    <- 1; # Payoff to mated female given worthless gift male
Qgm    <- 1; # Payoff to mated female given genuine gift male
Pav    <- 2; # Payoff to male for no gift to virgin female
Pfv    <- 1; # Payoff to male for worthless gift to virgin female
Pgv    <- 1; # Payoff to male for genuine gift to virgin female
Pam    <- 1; # Payoff to male for no gift to mated female
Pfm    <- 1; # Payoff to male for worthless gift to mated female
Pgm    <- 1; # Payoff to male for genuine gift to mated female

```

With the above, we can run 10 replicate simulations under the same starting conditions as above, each 200 generations of evolution.

```

ind_hist <- replicate_sims(gens = 200);

## [1] 200.00000000  0.53306858  0.43143959  0.03549183
## [1] 200.00000000  0.5165299  0.3093179  0.1741522
## [1] 200.00000000  0.4223250  0.4769885  0.1006865
## [1] 200.00000000  0.28621709  0.69155434  0.02222857
## [1] 200.00000000  0.25395831  0.67718105  0.06886063
## [1] 200.00000000  0.18575896  0.79547511  0.01876593
## [1] 200.00000000  0.35708758  0.58251974  0.06039269
## [1] 200.00000000  0.50290783  0.43134256  0.06574961
## [1] 200.00000000  0.39004608  0.57667671  0.03327721
## [1] 200.00000000  0.50804211  0.48254512  0.00941277

```

Note that the second columns 2-4 give  $a$ ,  $f$ , and  $g$  for males at the end of the simulation (the evolution of which is given in `ind_hist`). Obviously the parameters can be changed, and the functions can be adjusted if we do not like how the mutations or interactions work.