

Evolution of demographic parameters and population stability

Code ▼

Hide

```

# initial values
K <- 950 # equilibrium density
lambda <- 1.5 # growth rate
N_init <- 1000 # initial total density
c <- -2 # complexity
mu <- 0.01 # mutation rate
# later, we define r as being the relative density à savoir N/mean(K)

# population initiale
population <- data.frame(K=rep(K, N_init), lambda=rep(lambda, N_init), c=rep(c, N_init))

# calculate fitnesses
calc_fitness <- function(N, K, lambda, c) {
  # make sure we're not accidentally dividing by 0
  if (any(lambda == 1)) {
    lambda <- 1.0000000001
  }
  res <- lambda / (1 + (lambda - 1)*((N/K)**((1-c)*lambda/(lambda-1))))
  return(res)
}
# test
# with(population[1,], calc_fitness(N_init, K, lambda, c))

# run
fitness <- with(population[1,], calc_fitness(N_init, K, lambda, c))
population$fitness <- rep(fitness, N_init)

# offspring per individual without mutation
offspring <- function(individual, N) {
  n_offspring <- rpois(1, individual$fitness)
  if (is.na(n_offspring)) {
    return(vector())
  } else {
    offspring <- data.frame(K=rep(individual$K, n_offspring), lambda=rep(individual$lambda, n_offspring), c=rep(individual$c, n_offspring), fitness=rep(individual$fitness, n_offspring))
    return(offspring)
  }
}
# test
# offspring(population[1,], N_init)

# next generation
next_generation <- function(population){
  next_gen <- data.frame(K=vector(), lambda=vector(), c=vector(), fitness=vector())
  for (i in 1:nrow(population)){
    o <- offspring(population[i,], nrow(population))
    next_gen <- rbind(next_gen, o)
  }
  return(next_gen)
}
# test
# next_generation(population)

# add mutation

```

```

offspring_with_mutation <- function(individual, N) {
  offspring <- offspring(individual, N)
  n_offspring_with_mutation <- rpois(1, nrow(offspring)*mu)
  for (n in n_offspring_with_mutation) {
    for(i in 1:(ncol(individual)-1)) {
      offspring[n,i] <- offspring[n, i] + rnorm(1, 0, i*0.04)
    }
  }
  return(offspring)
}

mutate <- function(individual){
  for(i in 1:(ncol(individual)-1)) {
    individual[,i] <- individual[n, i] + rnorm(1, 0, i*0.04)
  }
}

next_generation_with_mutation <- function(population){
  next_gen <- data.frame(K=vector(), lambda=vector(), c=vector(), fitness=vector())
  for (i in 1:nrow(population)){
    o <- offspring(population[i,], nrow(population))
    next_gen <- rbind(next_gen, o)
  }
  n_mutations <- rpois(1, nrow(next_gen)*mu)
  mutants <- sample(nrow(next_gen), n_mutations, replace=FALSE)
  if(length(mutants) > 0) {
    for(i in 1:(ncol(next_gen)-1)) {
      next_gen[mutants,i] <- next_gen[mutants, i] + rnorm(1, 0, i*0.04)
    }
    next_gen[mutants,4] <- with(next_gen[mutants,], calc_fitness(N, K, lambda, c))
  }
  return(next_gen)
}

# run over x generations and plot evolution of population size etc
run <- function(x, population) {
  generations <- population[1,]
  generations$N <- nrow(population)
  # without mutation
  for (i in 1:(x-1)) {
    population <- next_generation(population)
    population$N[1] <- nrow(population)
    generations <- rbind(generations, population[1,])
  }
  return(generations)
}

run_with_mutation <- function(x, population) {
  population_with_mutation <- population
  # with mutation
  generations_with_mutation <- data.frame(N=nrow(population_with_mutation), K_means=mean(population_with_mutation$K), K_sd=sd(population_with_mutation$K), lambda_means=mean(population_with_mutation$lambda), lambda_sd=sd(population_with_mutation$lambda), c_means=mean(population_with_mutation$c), c_sd=sd(population_with_mutation$c), fitness_means=mean(population_with_mutation$fitness), fitness_sd=sd(population_with_mutation$fitness))
}

```

```

tion$fitness), fitness_sd=sd(population_with_mutation$fitness))
  for (i in 1:(x-1)) {
    population_with_mutation <- next_generation_with_mutation(population_with_mutation)
    generations_with_mutation <- rbind(generations_with_mutation, c(nrow(population_with_mutation), mean(population_with_mutation$K), sd(population_with_mutation$K), mean(population_with_mutation$lambda), sd(population_with_mutation$lambda), mean(population_with_mutation$c), sd(population_with_mutation$c), mean(population_with_mutation$fitness), sd(population_with_mutation$fitness)))
  }
  generations_with_mutation$r <- generations_with_mutation$N / generations_with_mutation$K_means
  return(generations_with_mutation)
}

x <- 20
generations <- run(x, population)
generations_with_mutation <- run_with_mutation(x, population)

plot(1:x, generations$N, type="b", pch=19, col="blue", xlim=c(1, x), ylim=c(0, max(c(generations$N, generations_with_mutation$N))), xlab="Time in generations", ylab="Population size", main="Evolution of population sizes over generations with and without mutations")
par(new = T)

```

Hide

```

plot(1:x, generations_with_mutation$N, type="b", pch=18, col="red", xlim=c(1, x), ylim=c(0, max(c(generations$N, generations_with_mutation$N))), xlab="Time in generations", ylab="Population size")

legend(1, max(c(generations$N, generations_with_mutation$N))-200, legend=c("Without mutation", "With mutation"), col=c("blue", "red"), lty=1:2, cex=0.8)

```

Evolution of population sizes over generations with and without mutation

