Homework 5: Manduca genetic algorithms

Eric R. Scott

I had trouble getting `matplotlib` to work in Python, so I instead edited `manducaEv()` to add an option to export the data on best fitness for each generation to a .csv file. Then, I read those .csv files into R and did my plotting there.

```r
library(here)
library(tidyverse)
```

Read in results of three 70 generation trials

```r
seed33 <- read_csv(here("Homeworks", "Manduca33.csv"), col_names = "Fitness") %>%
  rownames_to_column(var = "Generation")
seed121 <- read_csv(here("Homeworks", "Manduca121.csv"), col_names = "Fitness") %>%
  rownames_to_column(var = "Generation")
seed9000 <- read_csv(here("Homeworks", "Manduca9000.csv"), col_names = "Fitness") %>%
  rownames_to_column(var = "Generation")
seed9000_long <- read_csv(here("Homeworks", "Manduca9000_210gen.csv"), col_names = "Fitness") %>%
  rownames_to_column(var = "Generation")
```

Winners:

Seed 33 winner with fitness 531.6907559

```
   legs    |     muscles
1  1  0  1  1 |   0  100  100  100
1  1  0  1  1 | 100  100    0  100
0  0  0  0  1 | 100  100  100  100
1  1  1  1  1 | 100  100  100  100
1  1  0  1  1 | 100  100  100  100
1  1  0  0  0 | 100  100    0    0
1  0  0  1  1 |   0  100  100  100
1  1  1  0  1 | 100  100    0  100
1  1  0  1  0 | 100  100  100    0
0  0  1  0  1 | 100  100    0  100
```

Seed 121 winner with fitness 1101.4863654

```
   legs    |     muscles
1  1  1  1  1 | 100  100  100  100
0  0  0  0  1 | 100  100  100  100
1  0  0  1  0 | 100  100    0  100
1  0  0  0  0 | 100    0    0  100
1  1  0  1  1 | 100    0  100  100
0  0  1  0  1 | 100  100  100  100
1  1  1  0  1 | 100  100  100  100
1  0  0  0  0 |   0    0    0    0
0  0  0  0  1 | 100  100  100  100
1  1  1  0  1 | 100    0    0  100
```

Seed 9000 winner with fitness 561.3557324

```
   legs    |     muscles
0 0 1 1 1 |   0   0 100 100
0 0 0 0 1 | 100 100 100 100
1 1 1 1 0 | 100   0 100   0
1 1 0 1 1 | 100 100 100   0
1 0 0 0 1 |   0 100 100 100
1 1 0 0 0 | 100   0 100 100
1 0 1 1 1 |   0 100 100   0
1 0 1 1 1 |   0 100 100   0
0 1 1 0 0 | 100 100   0   0
1 1 0 1 1 |   0   0 100 100
```

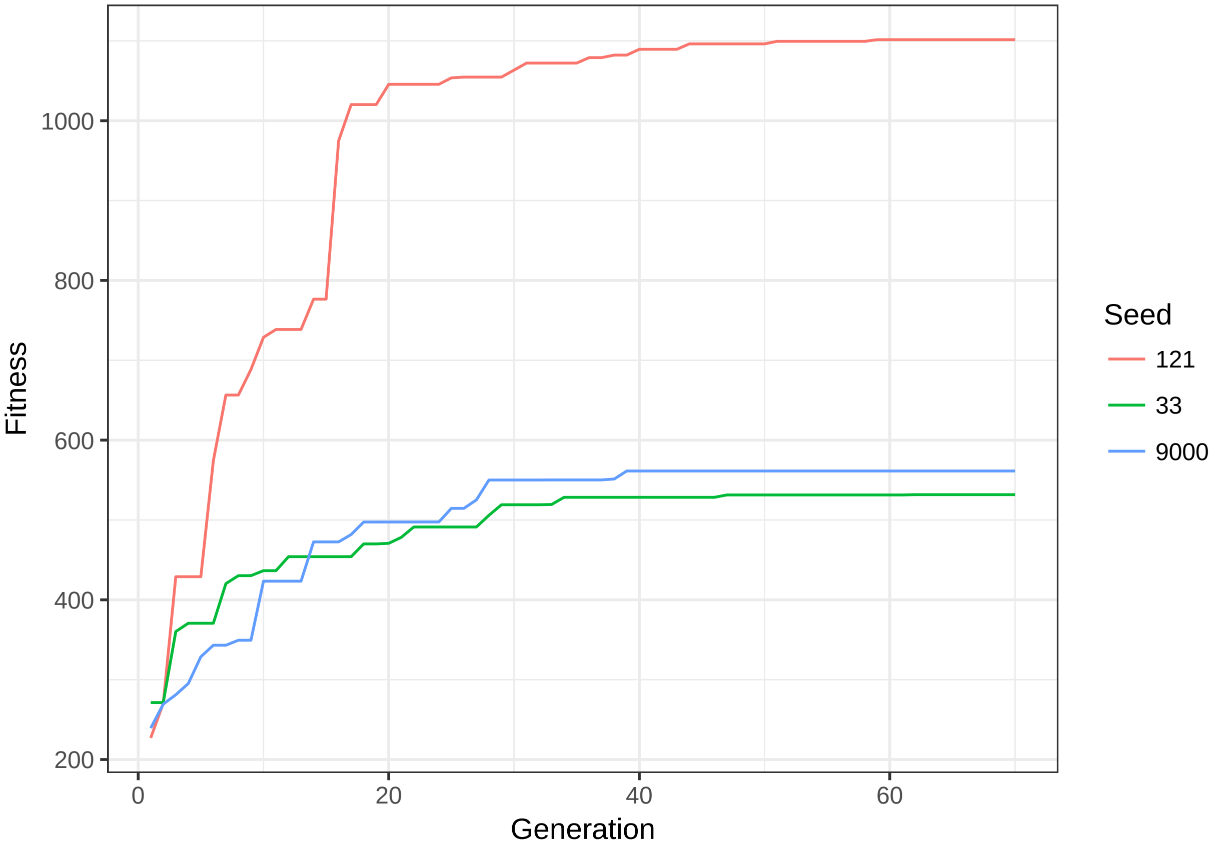
Seed 9000 winner with 210 generations. Fitness: 566.6956821

```
   legs    |     muscles
1 0 1 1 1 | 100   0 100 100
0 0 0 0 1 | 100 100 100 100
1 1 1 1 0 | 100 100 100   0
1 1 0 1 1 | 100 100 100   0
1 0 0 0 1 |   0 100 100 100
1 1 0 0 0 | 100   0 100 100
1 0 1 1 1 |   0 100 100   0
1 0 1 1 1 |   0 100 100 100
0 1 1 0 0 | 100 100   0   0
1 1 0 1 1 |   0   0 100 100
```

1. Plot Best Distance Vs. Generation

```r
all <- list(seed33, seed121, seed9000)
names(all) <- c("33", "121", "9000")
plot_data <- bind_rows(all, .id = "Seed")
plot_data <- plot_data %>%
  mutate(Generation = as.integer(Generation))

ggplot(plot_data, aes(x = Generation, y = Fitness, color = Seed)) +
  geom_line() +
  theme_bw()
```



2. Why are they different?

Each run uses a different random seed. That seed is used to choose how many mutations are made each generation, which rows to choose from which parents, and which individuals to keep each generation. The differences in each run are purely stochastic, but lead to big differences in final fitness. These differences demonstrate the effects of local maxima. For seed 33 and 9000, fitness levels off after generation 40 such that no better solution is reached through mutations and mating, even though a better solution clearly exists, as is evident from seed 121.

3. What's a good initial solution?

I'm not sure what a good initial population or individual really means. In this example, the three seeds started with roughly the same initial max fitness for their respective populations. Despite that, clear differences in the outcome of seed 121 vs. the other two are obvious after only 5 generations. It looks like max fitness levels off for a couple of generations, but then evolution results in a steep increase in max fitness in the next few generations. Because of mating and random mutations, I think even if the starting population all moved backwards or couldn't move at all, you'd still get evolution of a forward moving *Manduca*.

4. Several short runs, or one long run?

For the homework, I ran `manducaEv()` using seed 9000 for both 70 and 210 generations (not shown in plot). In this example, the long run didn't improve on the shorter run by very much (short run max fitness =

561.3557324, long run max fitness = 566.6956821). I had much better luck from running the simulation for 70 generations with three different random seeds. It's possible that if it was run for enough generations that a mutation would get evolution "unstuck" from this local maximum. I'm not sure one approach would *always* work better than the other, but I think it would generally be a bad idea to only run the simulation with one random seed since it seems to have a big impact on the outcome. I also know from previous work with Bayesian models and Markov Chain Monte Carlo simulations, that you generally seed many random "populations" in these types of random simulations.