Exercise 2: Frequency dependent fitness

Álvaro Huertas García
Miguel Hernández del Valle
Diego Mañanes Cayero
Alejandro Martín Muñoz
Sara Dorado Alfaro

January 7, 2020

Table of contents

1. Introduction
2. Functionality
   - Additional functions
   - Modification of the code
3. Experiments
   - List of experiments
   - An experiment with bacteria
   - An experiment with cells

Table of contents

1. Introduction
2. Functionality
   - Additional functions
   - Modification of the code
3. Experiments
   - List of experiments
   - An experiment with bacteria
   - An experiment with cells

Introduction → 2 Frequency dependent fitness

- Game theory.
- *OncoSimulR* model → the fitness of a subpopulation will depend on the relative abundance of the different subpopulations.
- The fitness of each subpopulation is defined as an arbitrary function of the genetic interactions between multiple genes.

- *allFitnessEffects* function:
  - *genoFitness* = dataframe
    - First column: genotypes.
    - Second column: expressions for the functions that relate fitness to frequencies of other genotypes.
  - *frequencyDependentFitness* = TRUE
  - *frequencyType* = "rel" or *frequencyType* = "abs"
  - *spPopSizes*

Introduction → Assess fitness

- *evalGenotype* function:
  - *fitnessEffects* = *allFitnessEffects* object
  - *genotype*
- *evalAllGenotypes* function:
  - *fitnessEffects* = *allFitnessEffects* object

Introduction → Perform simulations

- *oncoSimulIndiv* and *oncoSimulPop* functions.

```
simulScen1 <- oncoSimulIndiv(scen1,
                             model = "McFL",
                             onlyCancer = FALSE,
                             finalTime = 150,
                             mu = 1e-4,
                             initSize = 40000,
                             keepPhylog = TRUE,
                             seed = NULL,
                             errorHitMaxTries = FALSE,
                             errorHitWallTime = FALSE)
```

```
simulation <- oncoSimulPop(20,
                           mc.cores = 6,
                           afavc,
                           model = "McFL",
                           onlyCancer = FALSE,
                           finalTime = 25,
                           mu = 1e-2,
                           initSize = 4000,
                           keepPhylog = TRUE,
                           seed = NULL,
                           errorHitMaxTries = FALSE,
                           errorHitWallTime = FALSE)
```

Table of contents

1. Introduction

2. Functionality
   - Additional functions
   - Modification of the code

3. Experiments
   - List of experiments
   - An experiment with bacteria
   - An experiment with cells

Additional functions: graphical summary

- **Box-plot:** graphical summary of the distribution of simulations results

compositionPop2()

```
## Extract and create a data frame with results from several simulations
compositionPop2 <- function(objPop, ...) {
  ## Create genotype names
  clon_labels <- c("WT", objPop[[1]]$geneNames)

  ## Extract the information to create a data frame
  listPop <- vapply(objPop, function(x) tail(x[[1]], 1)[1, -1],
                    as.double(1:length(clon_labels)))

  dfPop <- data.frame("Genotype" = rep(clon_labels,
                                       length(listPop)/length(clon_labels)),
                      "N" = c(listPop))
  simul_boxplot2(dfPop, ...)
}
```

Figure: Code for compositionPop2() function

Additional functions: graphical summary

simul_boxplot2()

```r
## Plot box plot (by default same colors as plot.oncosimul type stream)
simul_boxplot2 <- function(df, main = FALSE, xlab = "Genotype", ylab = "N",
                           colors) {
  ## Create box plot, title and axis parameters
  e <- ggplot(df, aes(x = Genotype, y = N)) +
    theme(plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
          axis.title.x = element_text(size = 12, face = "bold"),
          axis.title.y = element_text(size = 12, face = "bold"),
          axis.text.x = element_text(size = 11),
          axis.text.y = element_text(size = 11))

  ## No title
  if (main == FALSE) {
    e + geom_boxplot(aes(fill = Genotype)) +
      stat_summary(fun.y = mean, geom = "point",
                   shape = 18, size = 2.5, color = "#FC4E07") +
      xlab(xlab) + ylab(ylab) +
      scale_fill_manual(values = colors)
      stat_summary(fun.y = mean, geom = "point",
                   shape = 18, size = 2.5, color = "#FC4E07") +
      xlab(xlab) + ylab(ylab) + scale_fill_manual(values = colors)}
  ## Title
  else {
    e + geom_boxplot(aes(fill = Genotype)) +
      stat_summary(fun.y = mean, geom = "point",
                   shape = 18, size = 2.5, color = "#FC4E07") +
      labs(title = main) +
      xlab(xlab) + ylab(ylab) + scale_fill_manual(values = colors)}
}
```

Figure: Code for simul_boxplot2() function

Additional functions: graphical summary

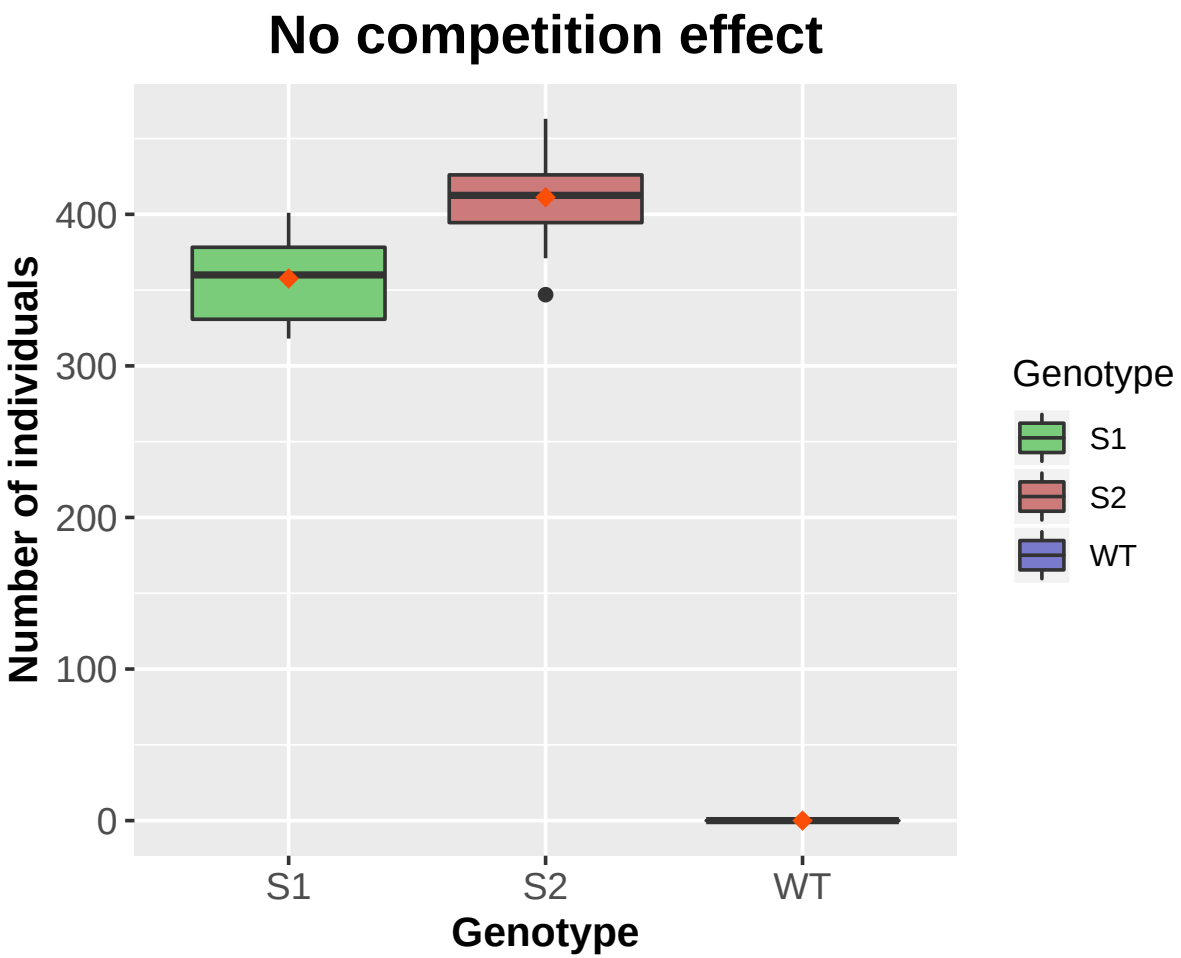


Figure: Box-plot from one of the Lotka-Volterra's example. 20 simulations were made

Additional functions: graphical summary

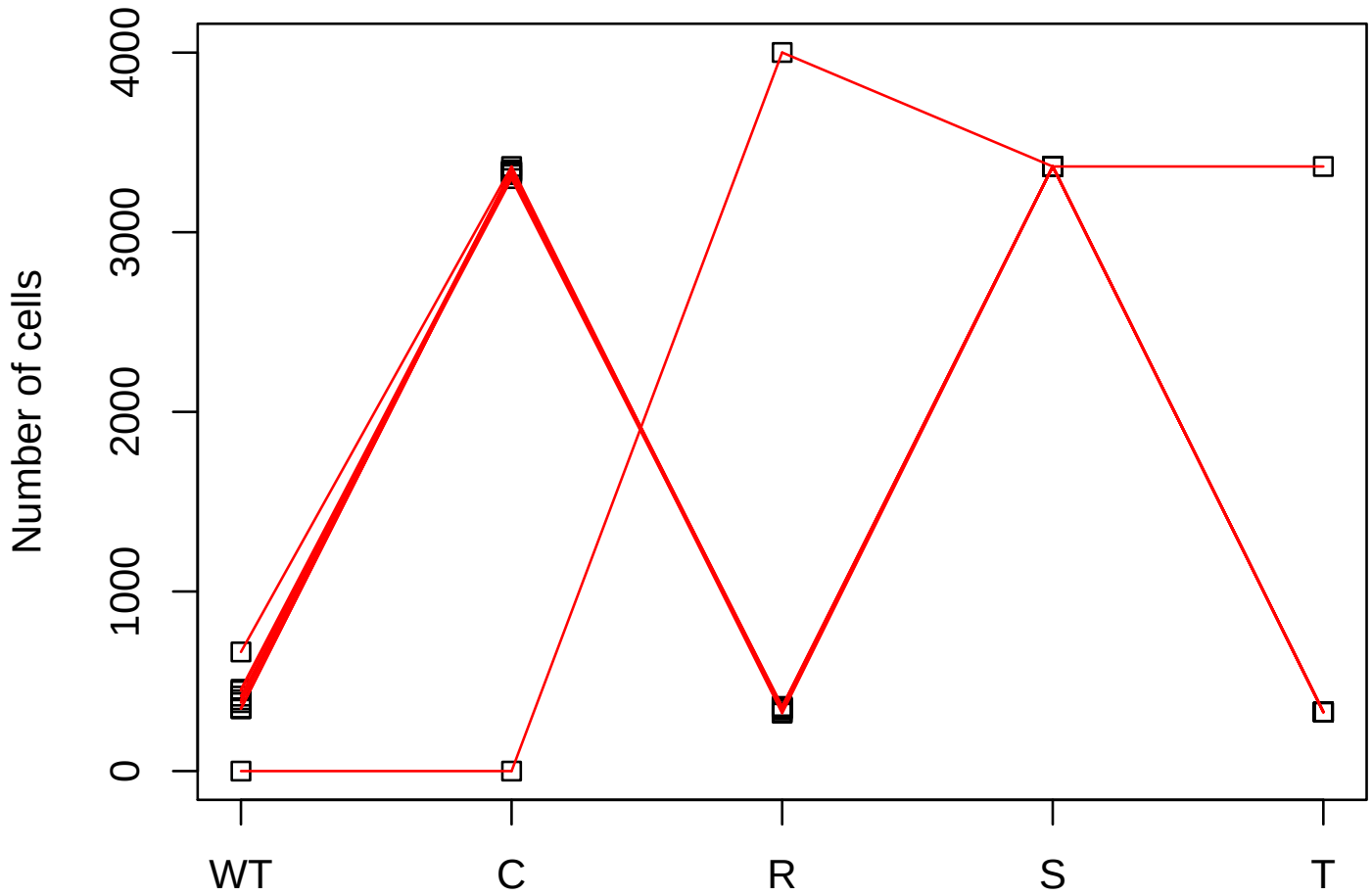
- **Stripchart:** summary of simulations with oscillating trajectories

```r
## Stripchart
stripChartPop <- function(dfPop, ylab = "N", ...) {
  stripchart(dfPop, vertical = TRUE, ylab = ylab, ...)
  f1 <- function(x, num_genotypes) {
    ## Draw a segment between genotype 1 and genotype 2
    num_genotypes <- length(x)
    i <- 1
    while (num_genotypes > i) {
      segments(x0 = i, x1 = i+1,
               y0 = x[i],
               y1 = x[i+1],
               col = rainbow(5))
      i <- i+1
    }
  }
  ## Read data frame by rows (simulation by simulation)
  apply(dfPop, 1, f1)
}

##  Plot the data as points and join with lines the ones that come from the same
##  simulation.
meanCompositionPop <- function(objPop, ...) {
  condi <- c("WT", objPop[[1]]$geneNames)
  ## Extract the information.
  ## $pops.by.time contains all the results
  ## $pops.by.time contains the times at wich results are taken
  ## see length of times and select results from the half time to the end
  ## Calculate the mean for each Genotype in each simulation
  ## Use lapply because results can be not rectangular
  listPop <- lapply(objPop, function(x)
    (colMeans(tail(x$pops.by.time, length(x$pops.by.time[,1])/2)[,-1])))
  ## Create data frame with the means Genotype from a list
  dfPop <- data.frame(matrix(unlist(listPop),
                             ncol = length(condi), byrow = TRUE))
  colnames(dfPop) <- condi
  stripChartPop(dfPop, ...)
  dfPop
}
```

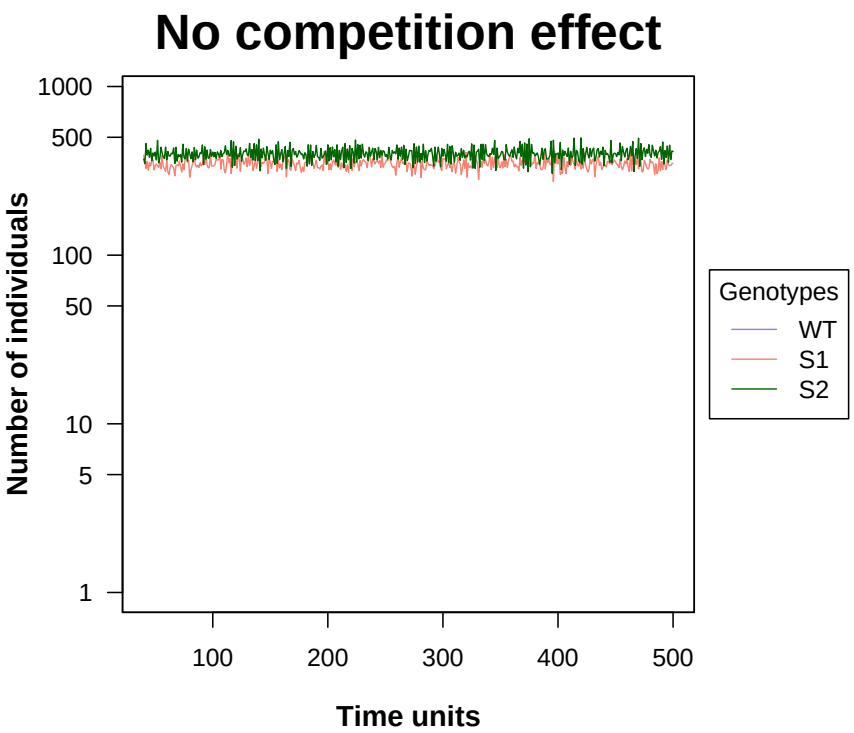
Figure: stripChartPop() and meanCompositionPop() code

Additional functions: graphical summary



Modification of the code

- **Legend location**



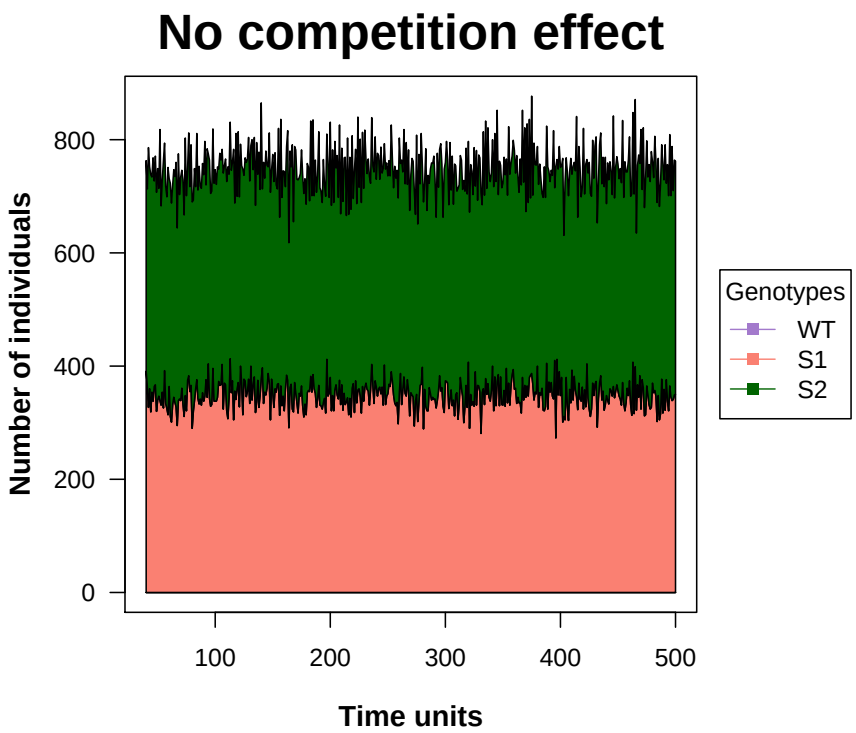


Table of contents

1. Introduction
2. Functionality
   - Additional functions
   - Modification of the code
3. Experiments
   - List of experiments
   - An experiment with bacteria
   - An experiment with cells

List of experiments

1. Rock-paper-scissors model in bacterial community
2. Evolutionary games theory: Hawk and Dove example
3. The Lotka-Volterra model of competition between two competing species
4. Game Theory with social dilemmas of tumour acidity and vasculature
5. Prostate cancer tumour–stroma interactions
6. Evolutionary Dynamics of Tumor-Stroma Interactions in Multiple Myeloma

List of experiments

1. Rock-paper-scissors model in bacterial community
2. Evolutionary games theory: Hawk and Dove example
3. The Lotka-Volterra model of competition between two competing species
4. Game Theory with social dilemmas of tumour acidity and vasculature
5. Prostate cancer tumour–stroma interactions
6. Evolutionary Dynamics of Tumor-Stroma Interactions in Multiple Myeloma

Rock-paper-scissors game in bacterial community

- **Title:** Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors.
- **Authors:** Benjamin Kerr, Margaret A. Riley, Marcus W. Feldman, Brendan J. M. Bohannan.
- Three competing species of bacterias with relationships similar to rock-paper-scissors game.

Local dispersal promotes biodiversity in a real game of rock–paper–scissors

Article *in* Nature · August 2002

DOI: 10.1038/nature00823 · Source: PubMed

CITATIONS
894

READS
1,040

4 authors, including:

Benjamin Kerr
University of Washington Seattle
55 PUBLICATIONS **3,174** CITATIONS
SEE PROFILE

Marcus W Feldman
Stanford University
771 PUBLICATIONS **41,882** CITATIONS
SEE PROFILE

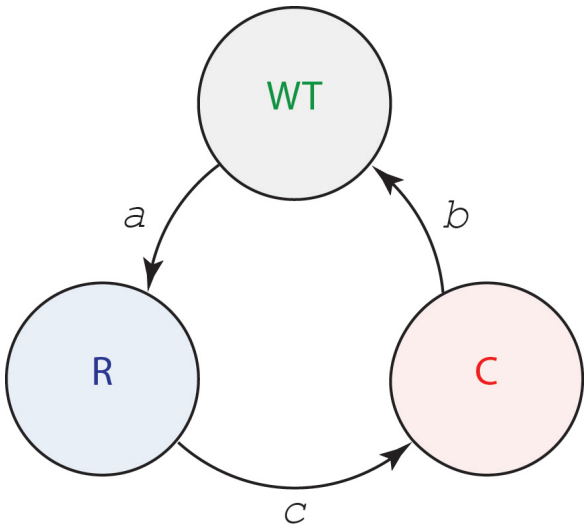
Brendan J M Bohannan
University of Oregon
268 PUBLICATIONS **14,204** CITATIONS
SEE PROFILE

Rock-paper-scissors game in bacterial community

- Three types of populations of *E. coli*:
  - Wild-type bacterias (WT): colicin-sensitive bacterias, hence, they are killed by colicin.
  - Colicinogenic bacterias (C): produce colicin toxin, a colicin-specific inmunity protein and a lysis protein which causes partial cell lysis and the release of the colicin.
  - Colicin-resistant bacterias (R): WT bacterias who have mutated getting membrane translocators of colicin.
- Parameters that describe the relationships of WT-C-R community:
  - a: advantage of WT over R $\Rightarrow$ R have costs in their fitness because the resistance consumes a lot of energy and WT have not this problem.
  - b: advantage of C over WT $\Rightarrow$ C are able to kill WT.
  - c: advantage of R over C $\Rightarrow$ R are resistant to colicin produced by C.

Rock-paper-scissors game in bacterial community

In summary, we have the relationships of rock-paper-scissors game:



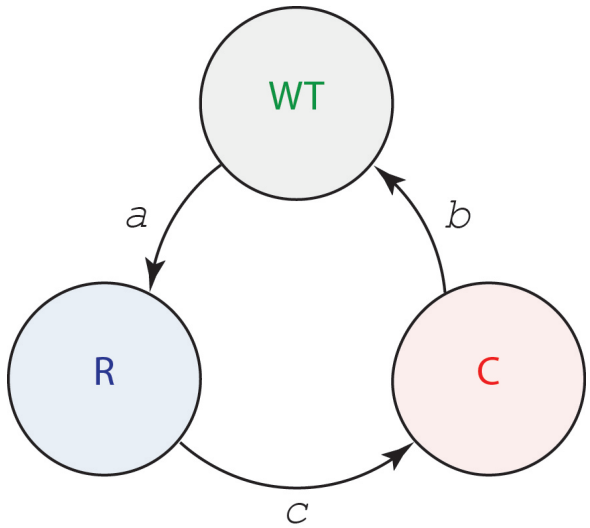
And the resulting equations:

$$W(WT) = 1 + af_3 - bf_2$$
$$W(C) = 1 + bf_1 - cf_3$$
$$W(R) = 1 + cf_2 - af_1$$

where f_1, f_2 and f_3 are the frequencies of WT, C and R, respectively.

Rock-paper-scissors game in bacterial community

In summary, we have the relationships of rock-paper-scissors game:



And the resulting equations:

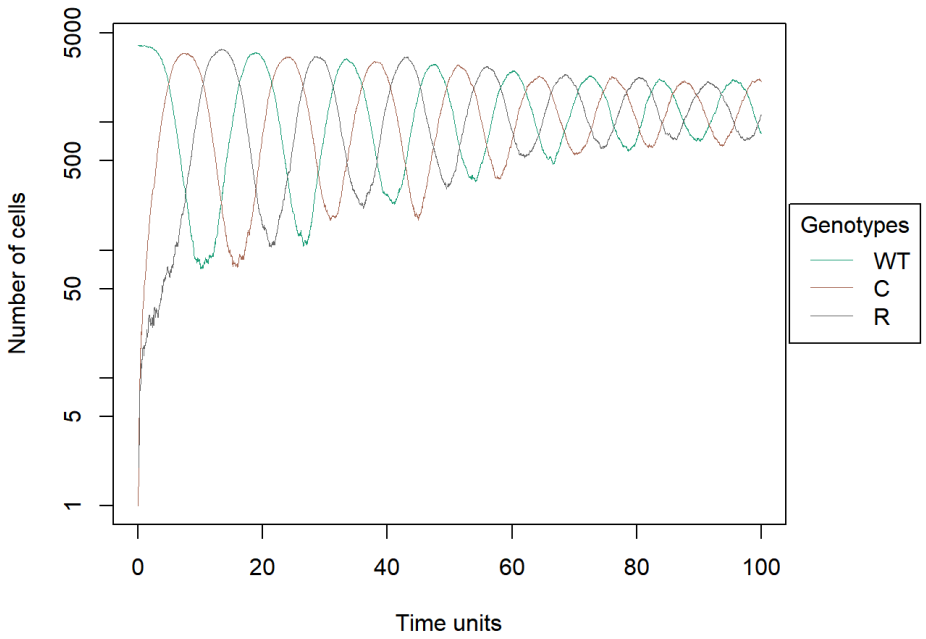
$$W(WT) = 1 + af_3 - bf_2$$
$$W(C) = 1 + bf_1 - cf_3$$
$$W(R) = 1 + cf_2 - af_1$$

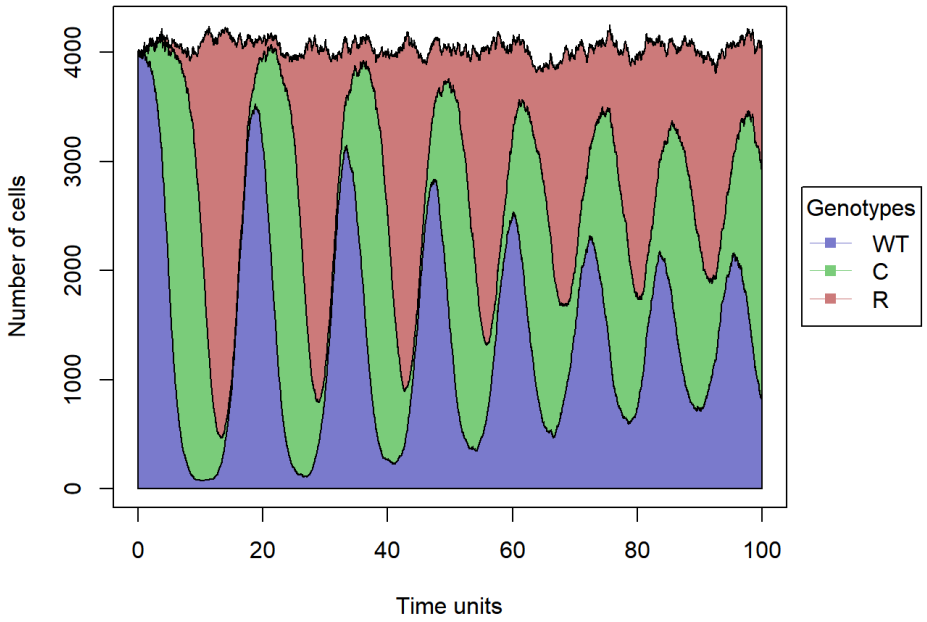
where f_1, f_2 and f_3 are the frequencies of WT, C and R, respectively.

allFitnessEffect and *oncoSimulIndiv* functions

Rock-paper-scissors game in bacterial community

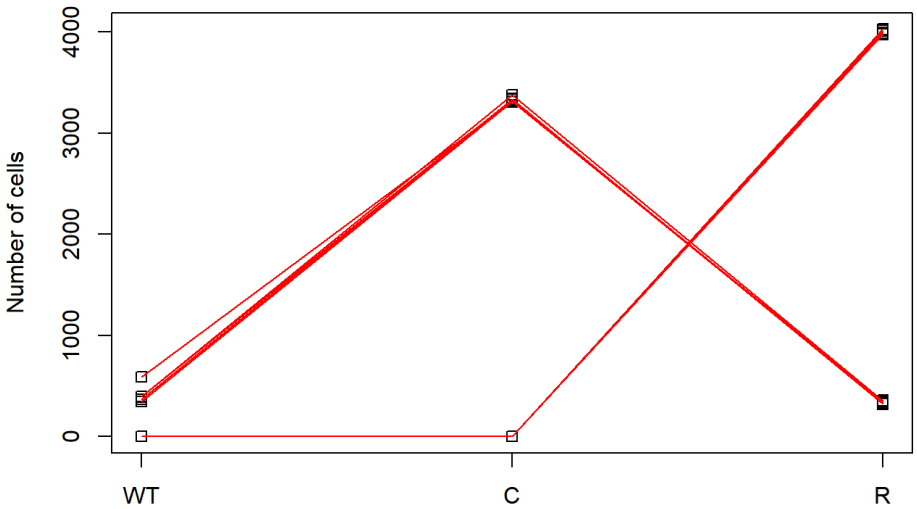
Case 1: $a = b = c = 1$

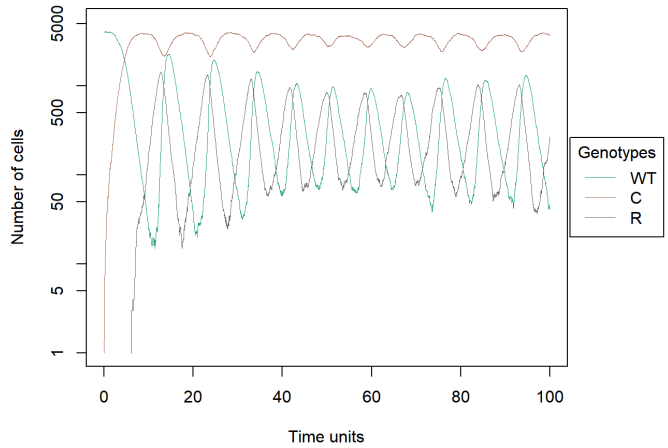


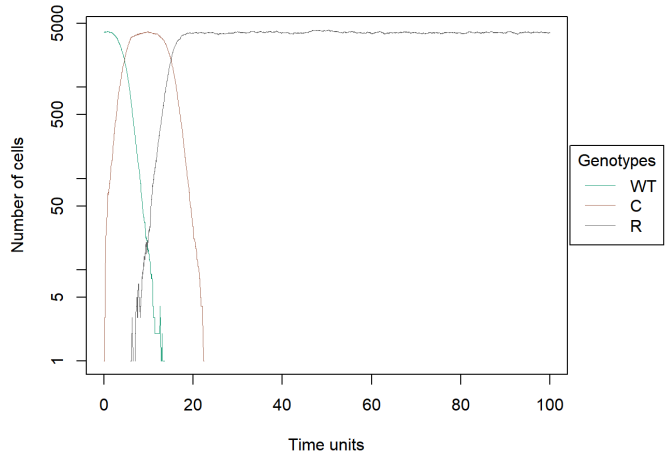


Rock-paper-scissors game in bacterial community

Case 2: a = 10, b = c = 1

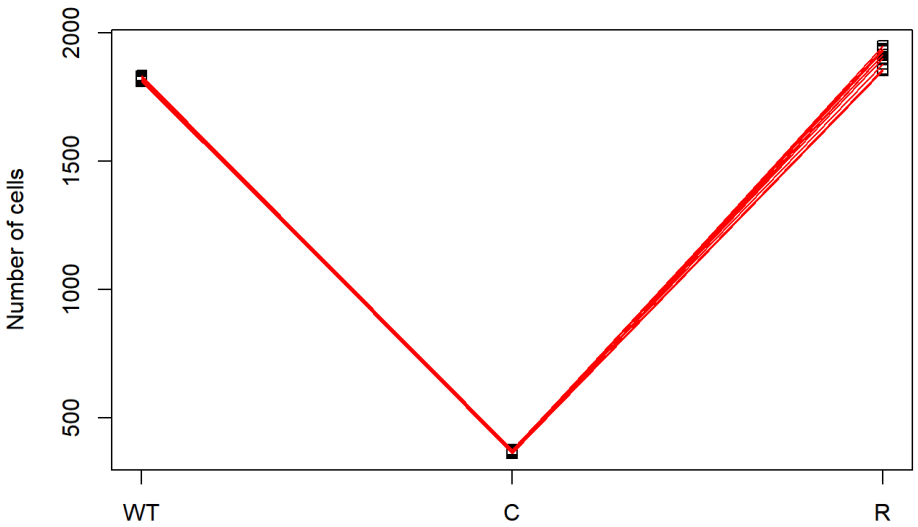


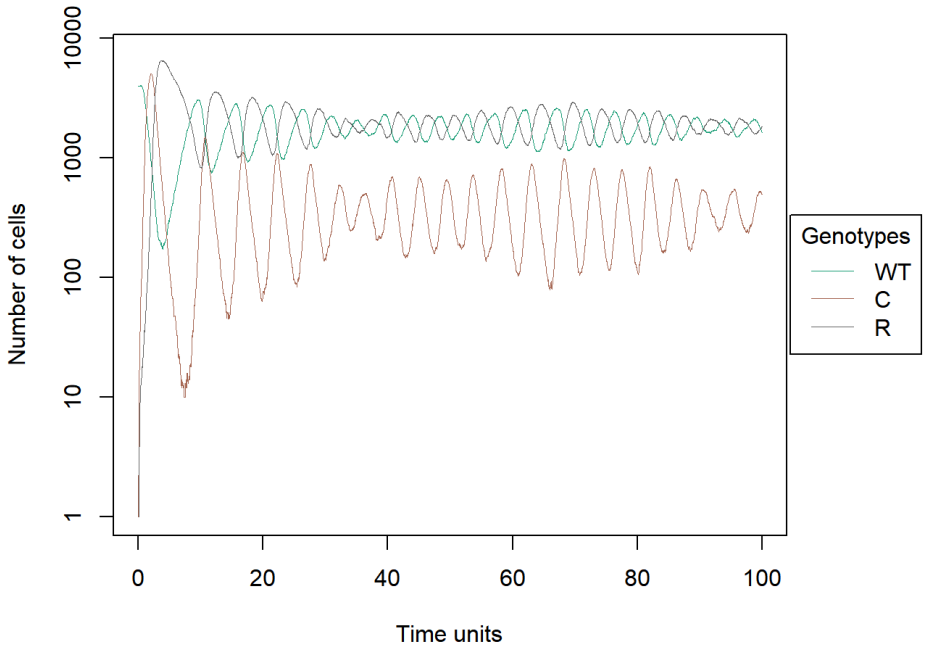




Rock-paper-scissors game in bacterial community

Case 3: $a = 1,\ b = c = 5$





Tumor-Stroma Interactions

- **Title:** Evolutionary Dynamics of Tumor-Stroma Interactions in Multiple Myeloma.
- **Authors:** Javad Salimi Sartakhti, Mohammad Hossein Manshaei, Soroosh Bateni, Marco Archetti.
- Cancer cells and stromal cells cooperate by exchanging diffusible factors.
    - Frequency-dependent selection that can be studied in the framework of evolutionary game theory.

Tumour-Stroma Interactions: payoff functions

- There are n phenotypes in a population denoted by $\{P_1, \ldots, P_n\}$.
- Each phenotype can produce one diffusible factor $\{G_1, \ldots, G_n\}$.
- Each diffusible factor j has a different effect $r_{i,j}$ on the other phenotypes i.
- The cost for P_i for growth factor G_i is denoted as c_i.
- M is the number of cells within the diffusion range.
  - There are M_j individuals of type P_j among the other group members.
- The payoff for strategy P_j is:

$$\pi_{P_j}(M_1, \ldots, M_n) = \frac{(M_j + 1) \times c_j}{M} r_{j,j} + \sum_{i=1, i \neq j}^{n} \frac{M_i \times c_i}{M} r_{j,i} - c_j \ .$$

Tumour-Stroma Interactions: dynamics

- Malignant plasma cells.
- Osteoblasts.
- Osteoclasts.
- Growth factors:
  - Autocrine effects.
  - Paracrine effects.

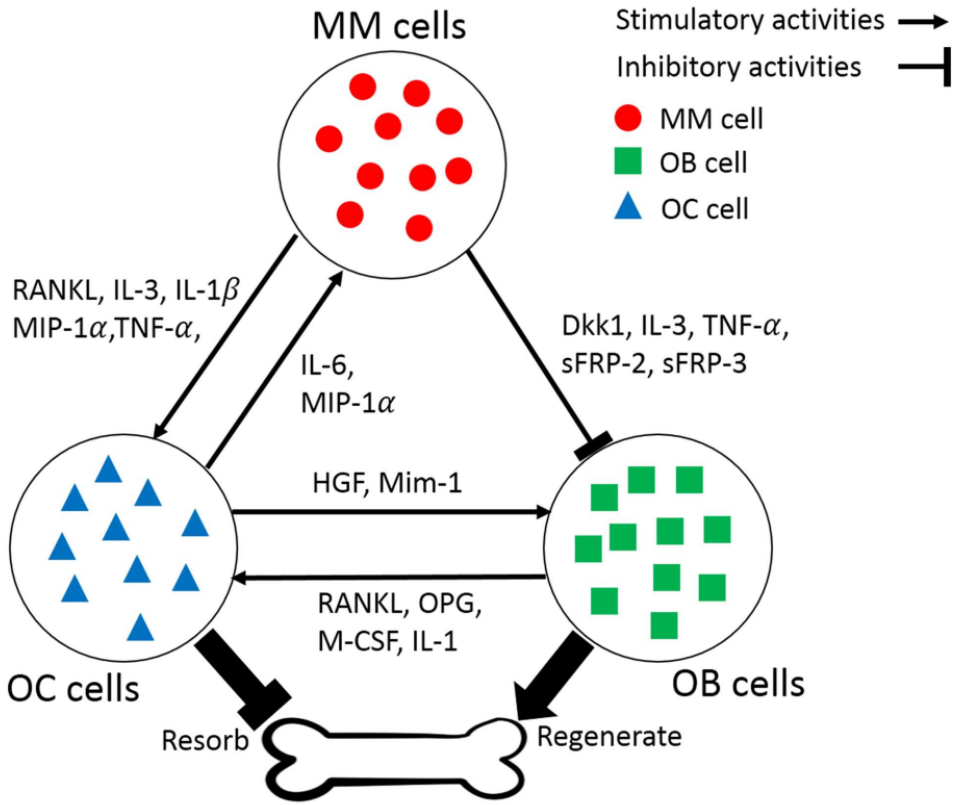


Fig 1. Bone remodeling in multiple myeloma. Multiple myeloma cells (MM) produce growth factors that activate osteoclasts (OC), which increase bone resorption, or that inhibit osteoblast (OB) differentiation. OC and OB secrete growth factors that affect each other and MM cells.

doi:10.1371/journal.pone.0168856.g001

Tumour-Stroma Interactions: Scenario 1

- $c_1 < c_2 < c_3$ (a common occurrence in multiple myeloma).
- In the presence of a small number of MM cells, the stable point on the OB-OC border becomes a saddle point and clonal selection leads to a stable coexistence of OC and MM cells.

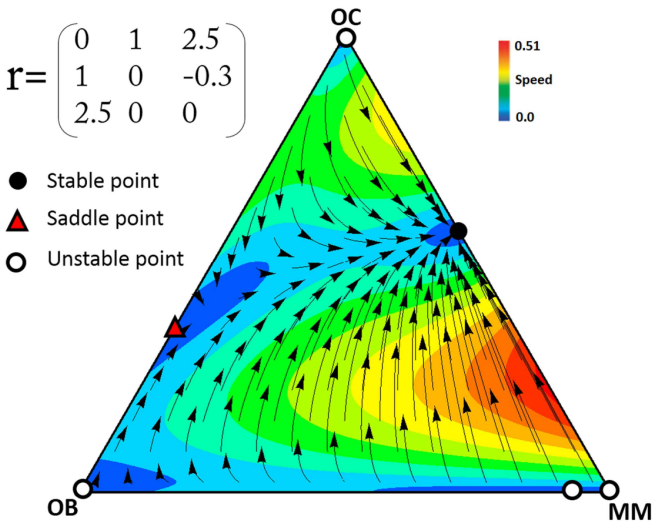
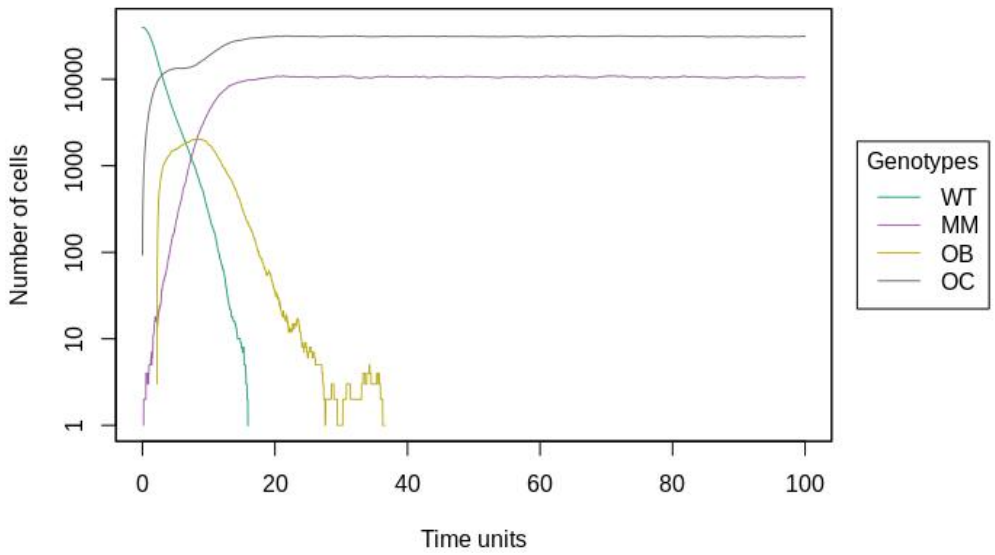


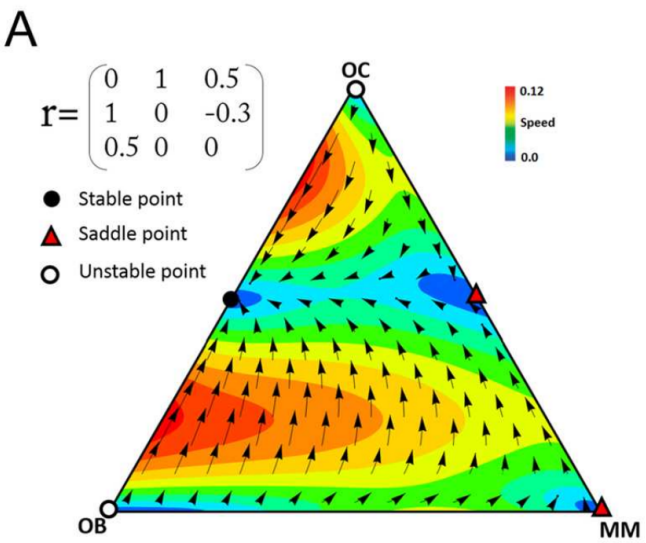
Fig 2. Example of the dynamics for scenario 1. In the presence of a small number of MM cells, the stable point on the OB-OC border becomes a saddle point and clonal selection leads to a stable coexistence of OC and MM cells. ($N = 10$, $c_3 = 1.4$, $c_2 = 1.2$, $c_1 = 1$). The arrows show the direction of the dynamics, and the colors show its speed (the euclidean distance between the frequencies at time t and t+1).

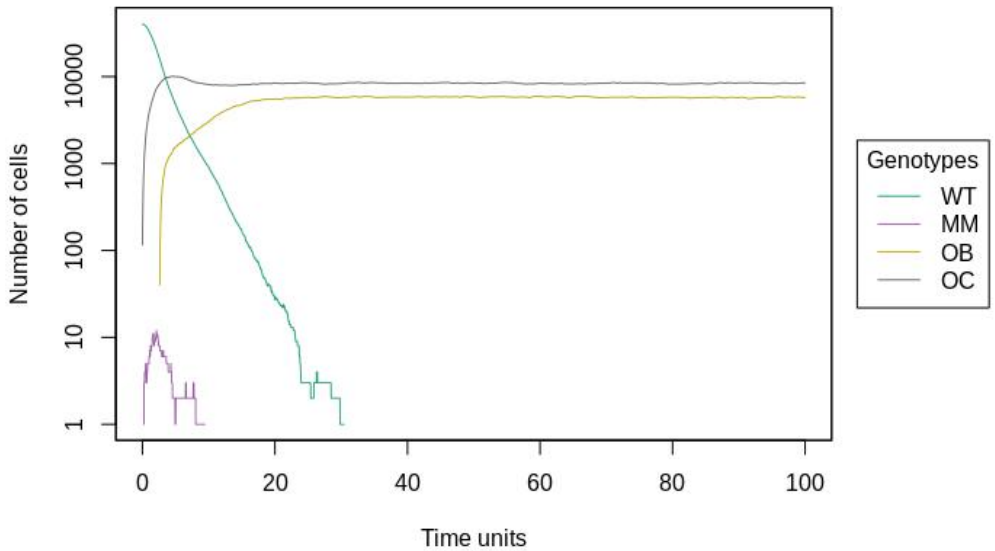
doi:10.1371/journal.pone.0168856.g002



Tumour-Stroma Interactions: Scenario 2

- $c_1 = c_2 = c_3$.
- The game has one polymorphic stable point between OB and OC. In this case, clonal selection leads to the regular OC-OB balance and prevents invasion of MM cells.





Exercise 2: Frequency dependent fitness

Álvaro Huertas García
Miguel Hernández del Valle
Diego Mañanes Cayero
Alejandro Martín Muñoz
Sara Dorado Alfaro

January 7, 2020