# Mutational constrains and adaptative therapy

This is an example inspired *by Gingi and Wood, 2021*. The current case shows how the fitness, in terms of cells grow, of a bacterial population depends on the probability of one strain mutate to another strain.

μ: mutation rate

mji: probability that strain j mutates to strain i, given that there is a mutation in strain j

The model consist of four genotypes and assumes a sequential pathway from ancestral to single mutants and then to double mutants:

WT: sensitive to both antibiotics

A: sensitive to one drug

B: sensitive to another drug

A,B: sensitive to both drugs

The genotypes were defined with an initial fitness for each (1.1(WT), 1.05 (A), 1.05 (B) and 1.01(A,B)) and the effect on that fitness of the probability (0.5) of mutating to other genotypes with a fixed mutation rate (0.01), in reference to the total size of the population (N):

|  |
| --- |
| dfat <- data.frame(Genotype = c("WT", "B", "A", "B, A"),  Fitness = c("1.1 - 0.01\*n\_/N + n\_A\*0.01\*0.5/N + n\_B\*0.01\*0.5/N",  "1.05 - 0.01\*n\_B/N + n\_\*0.01\*0.5/N + n\_A\_B\*0.01\*0.5/N",  "1.05 - 0.01\*n\_A/N + n\_\*0.01\*0.5/N + n\_A\_B\*0.01\*0.5/N",  "1.01 - 0.01\*n\_A\_B/N + n\_B\*0.01\*0.5/N + n\_A\*0.01\*0.5/N"))  afe2 <- allFitnessEffects(genotFitness = dfat,  frequencyDependentFitness = TRUE,  frequencyType = "abs")  evalAllGenotypes(afe2,spPopSizes = c(900,33,33,33)) |

The aim was to simulate how the addition of two different drugs to control the growth rate of a strain sensitive to both, could benefit the increase of resistant strains and how this increase could be also controlled.

αiβi: physiological effects of mutations on the growth rate

x,y: external drug/antibiotic dosage combination

The treatment was defined as a variable of the model:

|  |
| --- |
| variables\_modelo <- list(  list(Name = "treatment\_1",  Value = 0  )  )  v\_Model <- createUserVars(variables\_modelo) |

That follows the following rules:

* If the WT population size is half the initial (defined later in the code) there is not treatment (=0) (Treatment decisions based on the “50% rule” inspired by *Hansen and Read, 2020*)
* If the WT population size is higher or equal than the initial size the treatment is applied (=1)

|  |
| --- |
| rules <- list(  list(ID = "rule\_1",  Condition = "n\_ <= 500",  Action = "treatment\_1 = 0"  ),list(ID = "rule\_5",  Condition = "n\_ >= 1000",  Action = "treatment\_1 = 1"  )  )  rules <- createRules(rules, afe2) |

The treatment and its effect over each genotype population were defined as interventions:

|  |
| --- |
| intervenciones <- list(  list(ID="ANTIBIOTICO 1 y 2 SOBRE WT",  Trigger = "treatment\_1 == 1",  WhatHappens = "n\_ = n\_ -0.3 \* n\_ -0.3 \* n\_",  Periodicity = 0.07,  Repetitions = Inf),  list(ID="ANTIBIOTICO 1 y 2 SOBRE A",  Trigger = "treatment\_1 == 1",  WhatHappens = "n\_A = n\_A -0.3\*n\_A\*0.2-0.3\*n\_A",  Periodicity = 0.07,  Repetitions = Inf),  list(ID="ANTIBIOTICO 1 y 2 SOBRE B",  Trigger = "treatment\_1 == 1",  WhatHappens = "n\_B = n\_B - 0.3\*n\_B-0.3\*n\_B\*0.2",  Periodicity = 0.07,  Repetitions = Inf),  list(ID="ANTIBIOTICO 1 y 2 SOBRE A\_B",  Trigger = "treatment\_1 == 1",  WhatHappens = "n\_A\_B = n\_A\_B - 0.3\*n\_A\_B\*0.2 - 0.3\*n\_A\_B\*0.2",  Periodicity = 0.07,  Repetitions = Inf)    )  inter2 <- createInterventions(intervenciones,afe2) |

Therefore, the simulation includes all the previous considerations:

|  |
| --- |
| initSize = c(1000,30,30,30)  simu2 <- oncoSimulIndiv(afe2,  initMutant = c("WT", "B", "A", "B, A"),  initSize = initSize,  finalTime = 1000,  mu=0.00000000001,  userVars = v\_Model,  interventions = inter2,  rules=rules,  keepEvery = 1)  plot(simu2,show="genotypes") |

Finally some tests were incorporated in order to avoid:

* The exponential growth of the total population: setting the limit as the product of the sum of all individual populations and the total number of populations
* That the WT population reaches the limit of 0

|  |
| --- |
| pobs <- unlist(simu2$pops.by.time)[,2:5]  pobs  totalpob <- rowSums(unlist(simu2$pops.by.time)[,2:5])  totalpob  stopifnot(totalpob<(length(initSize)\*sum(initSize)))  stopifnot(pobs[, 1]>0) |

Some of the examples obtained:

Gráfico

Descripción generada automáticamente con confianza bajaGráfico

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza media