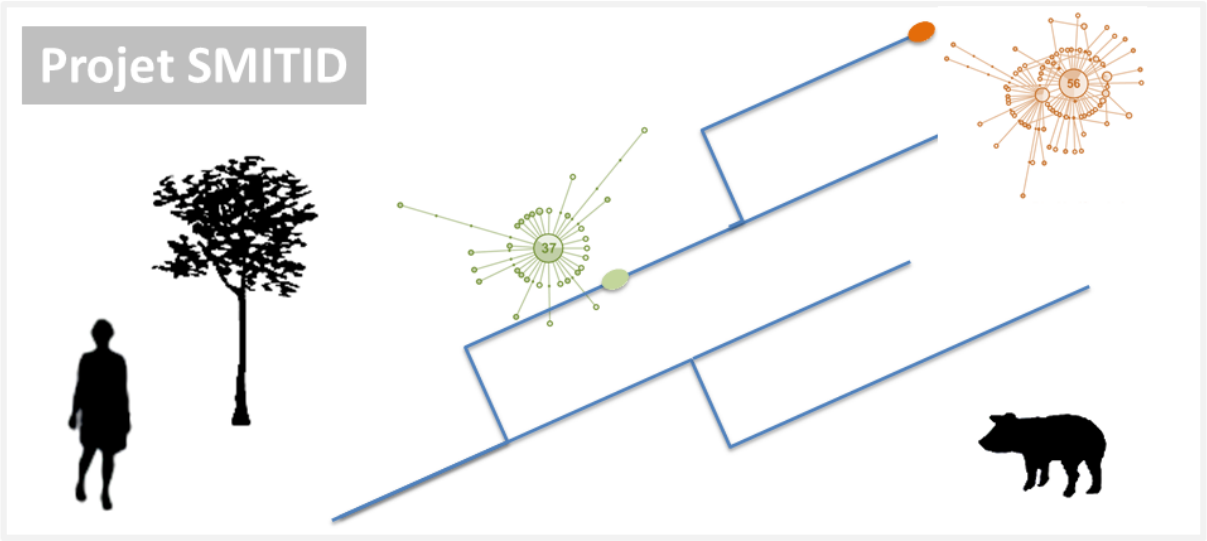
Meije Gawinowski
Ingineering student, INSA LYON

August 25, 2017

A simulation model for the kinetics, evolution and transmission of viral populations

Supervisor : Samuel Soubeyrand, INRA, Biostatistics and Spatial Processes

Funding : SMITID, ANR project, ANR-16-CE35006



Contents

Introduction

In order to predict and control the spread of infectious diseases, we have to understand how pathogens spread within a host and between hosts, but also what is the role of the environment.

The SMITID project (Statistical Methods to Infer Transmissions of Infectious Diseases from Deep Sequencing Data) aims at exploiting genomic sequencing data to explain the transmission links for pathogens that evolve quickly but also to explain the links between transmission and environment.

To do so, a simulation model that generates data under different conditions was created. This model consists of several steps :

- Modelling the viral kinetics ;
- Modelling the evolution of within host viral composition (in terms of different variants) ;
- Modelling the transmission dynamics (transmission times, quantity of transmitted virions, which variants in which proportions) ;
- Modelling the observation process of the epidemic and evolutionary dynamics (i.e modelling how the viral population is sampled from all or a fraction of the infected host units).

With this model we show that how kinetics and evolution parameters can impact intra-host viral genetic diversity.

1 Outbreak modelling

We wanted to model an outbreak in terms of viral genetic diversity in order to have access to the different sequences and their proportions at each time, so that we can assess diversity. In order to do so, we used the set of functions whose working is described below.

1.1 Model

1.1.1 Viral Kinetics

The first thing we want to do is assessing the viral kinetics within one host. We used the model described by Perelson and Smith (2012) [8], in which there are three compartments : the target cells T, the infected cells I and the infectious viral particles called virions V. The temporal evolution of those compartments is described by the following differential equations :

$$\begin{cases} \frac{\partial T}{\partial t} = (s - d) \cdot T - \beta \cdot T \cdot V \\ \frac{\partial I}{\partial t} = \beta \cdot T \cdot V - \delta \cdot I \\ \frac{\partial V}{\partial t} = p \cdot I - c \cdot V \end{cases}$$

This system of equations uses the following parameters (TCID_{50} is the 50% tissue culture infective dose) :

Parameter	Signification	Unit
V_0	Initial virion population	TCID_{50}/ml
s	Supply rate of target cells	day^{-1}
d	Death rate of target cells	day^{-1}
β	Infection rate	$(\text{TCID}_{50}/\text{ml})^{-1}\text{day}^{-1}$
δ	Death rate of infected cells	day^{-1}
p	Production rate of virions	$\text{TCID}_{50}\text{cell}^{-1}\text{day}^{-1}$
c	Clearance rate of virions	day^{-1}

Table 1 – Differents parameters of the viral kinetics model

We can represent this model with the diagram below :

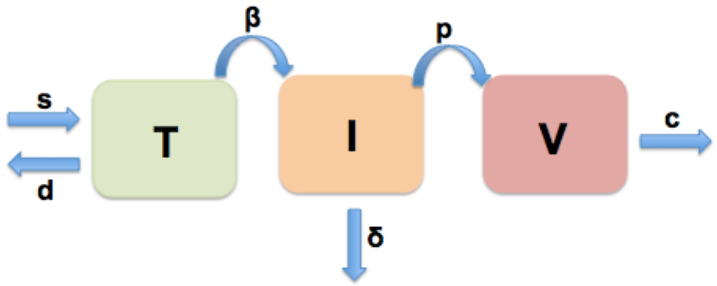


Figure 1 – Schematic diagram of the viral kinetics model

This model doesn't take into account the target cell limitation (bottleneck effect), the effect of immune responses or of antivirals. This might be a limit to our model.

We used an implicit method (more stable than an explicit one) to numerically solve the differential equations and create a dataframe with the number of individuals of the three compartments at each time.

1.1.2 Viral Composition

Once we have quantified the number of virions within a host during an outbreak, we simulate the viral composition, i.e the proportion of different variants (in terms of genetic sequences) of a virus in the host (so a variant matches a sequence). In this aim, the numerical model takes as arguments an initial set of N_0 sequences $\boldsymbol{S_0} = \{\boldsymbol{s_1}, ..., \boldsymbol{s_{N_0}}\}$ and their proportions $\boldsymbol{P}$ in the virion population. We consider N_i the number of sequences at the i^{th} time step with a set of sequences $\boldsymbol{S_i} = \{s_1, ..., s_{N_i}\}$. The pool of sequences and proportions is updated at each time. This function also takes as input the viral kinetics generated by the previous function, μ the mutation rate per nucleotide per virion per unit of time, a tolerance proportion threshold below which variants are eliminated and a fitness rule f that determines how fit a variant is. We call M the length of the nucleotide sequences.

An update consists in the mutation of the sequences (one mutation is one substitution of nucleotides on one sequence) and the viral multiplication to go to the next step. For mutations we don't distinguish transitions from transversions. At each time, the pool of sequences and proportions is updated according to the following steps :

1. At each time step i mutations are applied to the virions population at the previous time step $i-1$. We determine $N_{mut}(i)$ the number of mutations between time steps $i-1$ and i as a variable drawn from the following Poisson distribution :

$$N_{mut}(i) \sim \mathscr{P}(\mu \cdot M \cdot h(i-1) \cdot V(i-1)),$$

   where $h(i-1)$ is the time lapse between $i-1$ and i.

2. Then we draw the $N_{mut}(i)$ sequences that are going to mutate. As new sequences are going to be created by mutations, the proportions of sequences in the virion population are re-calculated.

3. Then we make the mutations actually happen : for each affected sequence of length M, the locus of the mutation is randomly uniformly drawn and the replacing nucleotide is also randomly uniformly drawn (although it has to be different from the previous one). These new sequences are added to the pool of sequences and their fitnesses are computed.

4. Once mutations are done we take into consideration the multiplication of the virions, that is the transition to the next generation (i.e from $i-1$ to i). We define a probability of multiplication $\boldsymbol{P_m}$ for each sequence s of the pool that depends on its fitness so that the fittest variants will have a more efficient multiplication :

$$P_m(s,i) = P'(s,i) \cdot f(s),$$

with f the fitness rule function and $P'(s,i)$ the proportion of variants after step 2 described above.

Then we add a truncated inflated Gaussian noise to this probability that depends on two transformation parameters γ_1 and γ_2. These parameters describe the effect of the environment on the viral multiplication and diversification as they enable us to compel the drawing of variants with low proportions. Indeed, in reality variants with low proportions can multiply despite the competition against variants with higher proportions if they have different localisations, and this what we want these parameter to express. To do so, we define $\boldsymbol{P^*_m}$ as follows :

$$\boldsymbol{P'_m} \sim \mathcal{N}(\boldsymbol{P_m}, \gamma_1 \cdot \boldsymbol{P_m} \cdot (1 - \boldsymbol{P_m})^{\gamma_2})$$
$$\boldsymbol{P^*_m} = min\,(1, max\,(0, \boldsymbol{P'_m}))$$

Then we draw $\boldsymbol{n(i)}$ the number of variants for a set of sequences at time i as :

$$\boldsymbol{n(i)} \sim \mathscr{M}(V(i), \boldsymbol{P^*_m(i)})$$

Then the proportions are updated :

$$\boldsymbol{P'(i)} = \frac{\boldsymbol{n(i)}}{V(i)}$$

5. Then we put the N_i sequences of length M in a matrix, we apply the tolerance threshold on the proportions (i.e we eliminate the variants whose proportions are too low) and as it might change the number of sequences and so the proportions, they are recalculated. Finally we check if in our pool of sequences some of them are identicals : if it is the case they are brought together and their proportions are added into one while the other is equal to 0 (and then we only keep the data with a proportion superior to 0).

1.1.3 Transmission function

The transmission function determines the transmission times and their respective target hosts for one infectious host. We determine transmission contacts and hosts for an host h when it is infected at $T^{inf}(h)$. We take into consideration $D(h)$ the duration of infection for host h, $n_s(h)$ the number of hosts susceptibles to be infected (i.e the ones that haven't been infected yet) at $T^{inf}(h)$ and the parameter λ. We draw the number of contact times $c(h)$ for host h with the following Poisson process :

$$c(h) \sim \mathscr{P}(\lambda \cdot D(h) \cdot n_s(h))$$

Then we sample $c(h)$ contact times $\boldsymbol{T(h)}$ and their $c(h)$ respective target hosts $\boldsymbol{H(h)}$. For each time of contact $T^c(h)$, $V\,(T^c(h))$ is the number of virions within host h and V_{max} the number of maximum virions within the host. For each contact time, the probability of transmission $P_{Tr}(T^c(h))$ is defined as :

$$P_{Tr}(T^c(h)) = \frac{V(T^c(h)}{V_{max}}$$

So the more virions there are within a host, the more likely it is to transmit its virions. Then we determine the successfull contacts (the ones leading to a transmission) with a Bernoulli distribution. For each contact time T^c we determine the variable c_{tr} as $c_{tr}(T^c(h)) \sim \mathscr{B}\,(P_{tr}(T^c(h)))$. If $c_{tr}(T^c(h)) = 0$ then $T^c(h)$ is not a transmission but if $c_{tr}(T^c(h)) = 1$ then $T^c(h)$ is a transmission at time. Then for each success time we retrieve the transmission time and the target host that will be the output of the function.

1.1.4 Within-Host viral composition and sampling for observation and/or transmission

We want to describe the whithin-host viral composition at specific times being observation times (chosen by the user) and transmission times (determined with the transmission function). A state is the pool of sequences and their respective proportions. From an initial state, at each time of interest (observation or transmission)

we update the previous state with the viral composition to update the sequences and their proportions for the new state.

If it is an observation time, we draw the variants that are going to be observed in the total population at this time. The sample size is defined by the user and probabilities are the sequence proportions in the population. An index j is given to each sampled variant, their sequences are turned into ACGT and written in a file (along with the host ID, the variant index and the observation time).

If i is a transmission time, we define the number of transmitted variants V_{Tr} as follows :

$$V_{Tr}(i) = V(i) \cdot \tau,$$

where τ is the transmission rate. The transmitted variants are drawn into the total population with a sample size of the number of transmitted variants and the probability is their proportion, then the transmitted proportions are determined as follows :

$$\boldsymbol{n_{Tr}(i)} \sim \mathscr{M}(V_{Tr}(i), \boldsymbol{P'(i)})$$

At each transmission time we add the pool of transmitted sequence whose proportion is superior to 0 and then we delete the proportions inferior or equal to 0. We build a list that contains the ID of the infectious host, the ID of the infected host, the time of transmission, the number of transmitted variants, their proportions and their sequences for each transmission time. The output of the function is this transmission list.

1.1.5 Generation of an outbreak, simulation of the multi-host compositions of sequences and simulation of the observed sequences

We generate an outbreak in a population of n_{TOT} hosts and we want to observe the viral dynamics for a chosen number of hosts. The aim is to observe genetic data from different hosts at different times of the outbreak, so we want to model a transmission chain. We choose an initial host (patient 0) who is infected at $t = 0$. We determine its viral kinetics, the evolution of its viral composition and its transmission dynamics (i.e its transmission times, infected hosts, transmitted sequences and their proportions). Then the next host of the transmission host is selected as the one with the minimum transmission time. This new host is only added to the transmission chain if it hasn't been infected yet. Each time a new host is added to the transmission chain, we determine its viral kinetics (with time initialized at transmission time and initial virion population initialized at transmitted virion composition), the evolution of its viral composition and its transmission dynamics. The transmission data is always stored so when a new host must be added, the one with the minimum transmission time is selected. For each host of the transmission chain we have retrieved data and we built a file with this transmission chain (who infected who and when).

1.1.6 Visualisation with a diversity graph

Then we can represent the within-host viral genetic diversity with a diversity graph. For each studied host, the observation data is retrieved and the pairwised distance matrix (cf the diversity assessment section) is calculated among the different sampled variant sequences. The minimum spanning tree (MST) is determined. Then a graph is drawn according to Fruchterman-Reingold vertex layout. The variants are the different vertices and the edges are the genetic distances between them : an edge matches a difference of one nucleotide. This graph is force-directed, i.e it is simulated as a physical system : forces are applied to the nodes, pulling them closer together or pushing them further apart.

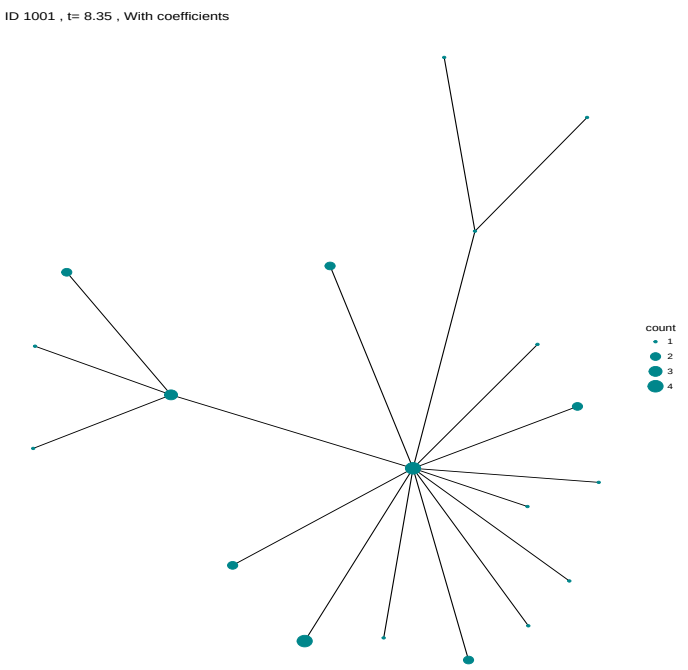


Figure 2 – Exemple of diversity graph

In the figure above, the turquoise spots are the vertices (the variants). The bigger the spot is, the higher the variant proportion is within the host. The more edges there are between two vertices, the more different the two matching variants are.

1.2 Results of the numerical modelling

Parameter	Signification	Value
V_0	Initial number of virions	10^3
S_0	Initial number of target cells	10^{12}
I_0	Initial number of infected cells	0
s	Target cell supply rate	0
d	Target cell death rate	0
β	Infection rate	$5 \cdot 10^{-5}$
δ	Infected cell death rate	2
p	Virus production rate	10^{-6}
c	Virus clearance rate	0.5
μ	Mutation Rate	$5 \cdot 10^{-6}$
γ_1	Transformation parameter	0.001
γ_2	Transformation parameter	1000
$tolerance$	Proportion tolerance threshold	10^{-5}
λ	Parameter for transmission	0.002
V_{max}	Maximal number of virions	10^6
τ	Transmission rate	10^{-2}
N_{S_0}	Initial number of susceptibles	1000

Table 2 – Simulation parameters

The initial viral composition in the initial host is a pool of 53 sequences of 1000 nucleotides each, mostly with A nucleotides. We used a time step of 0.001 days.

1.3 Without fitness rule

First of all we consider a situation without fitness rule (the fitness of all the variants is set to one), but with an influence of the transformation parameters $\gamma_1 = 0.001$ and $\gamma_2 = 1000$. We obtained this kind of figures for the kinetics of target cells, infected cells and virions :

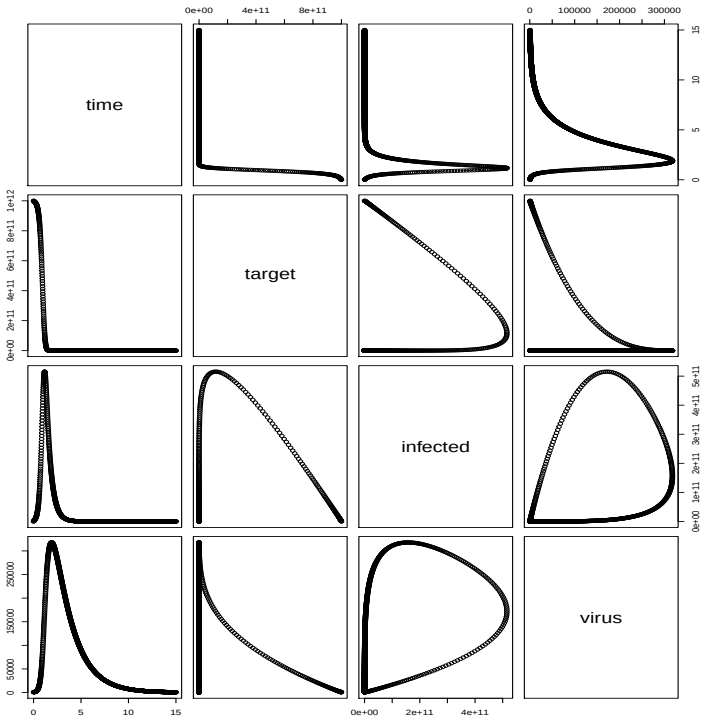


Figure 3 – Evolution of the three compartments

Then we can plot the graphs of the same host at different times of the infection. Here we chose the initial host :

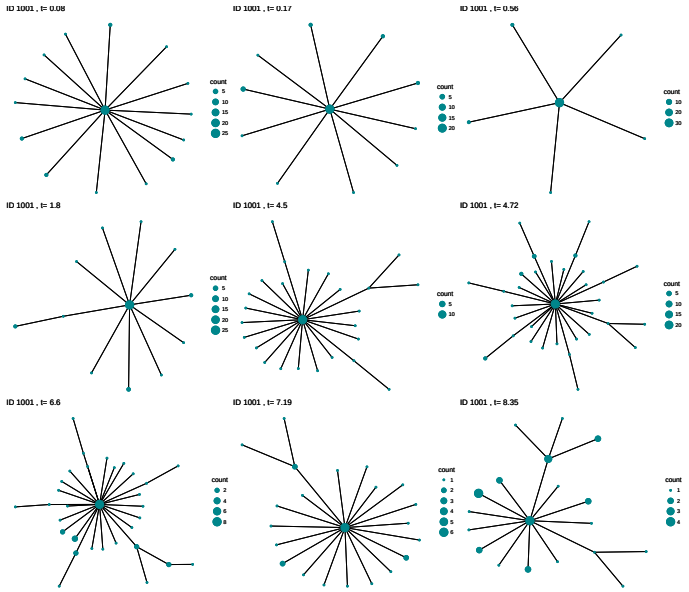


Figure 4 – Diversity graphs at different times

We can see that between the beginning of the of the infection and day 8, the viral genetic diversity became more important as there are more different variants and their proportions are higher. Then we can plot the graphs of the different hosts at one observation time :

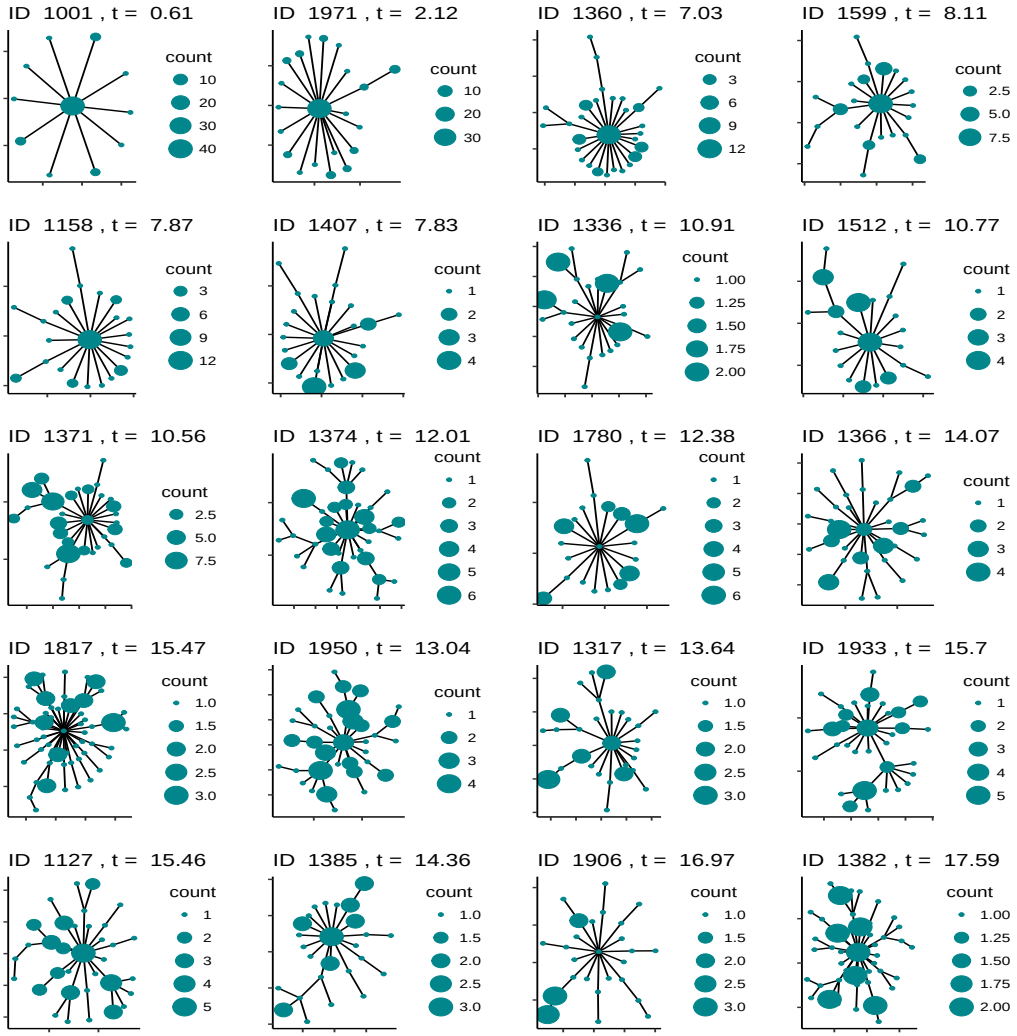


Figure 5 – Chain of transmission without any fitness rule

1.4 Without fitness rule and transformation parameters

The fitness of all the variants is still set to one but we also chose $\gamma_1 = 0$ so that there is no gaussian noise added to the probabilities of viral multiplication. For a chain of 20 hosts we have :

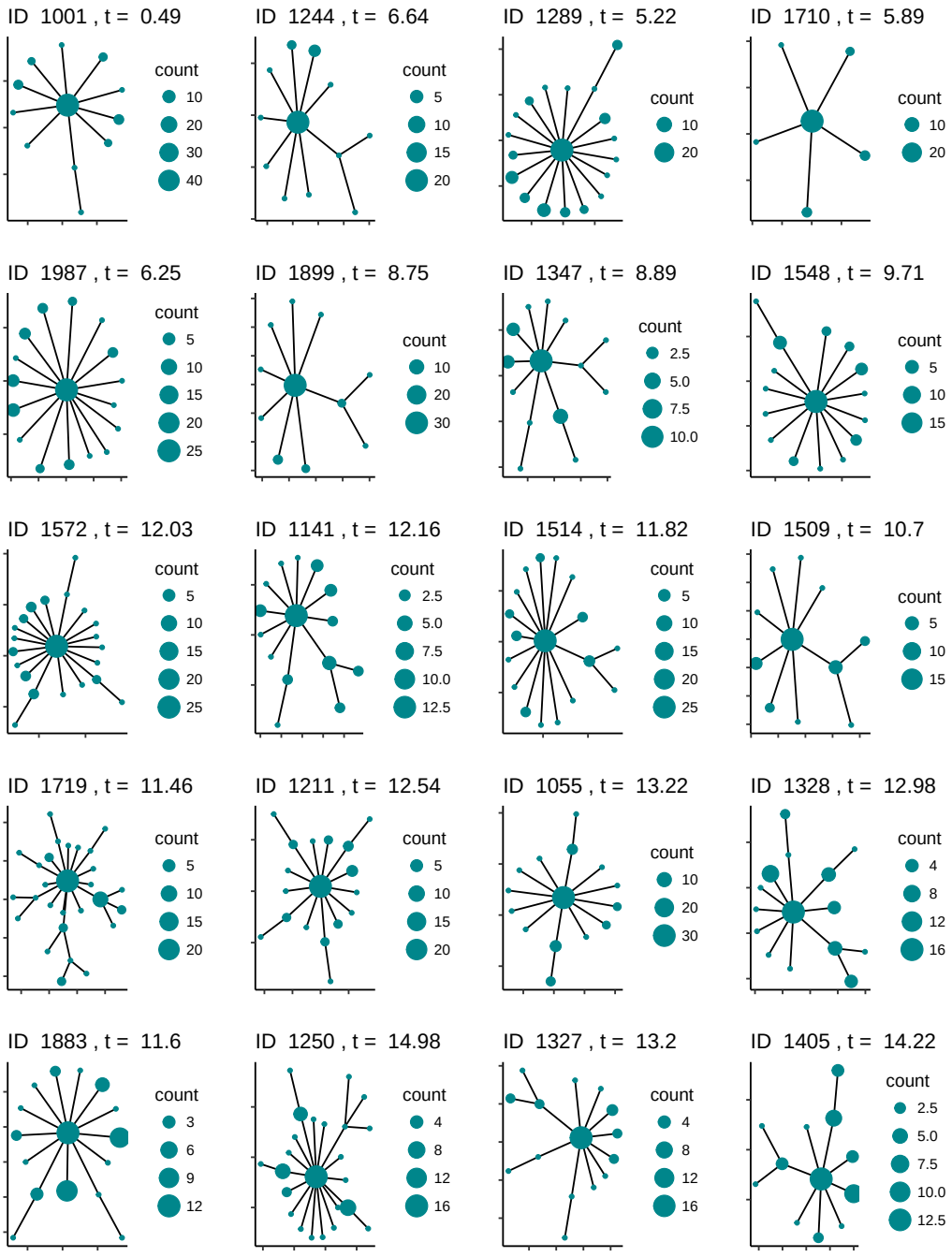


Figure 6 – Intra-host Viral Diversity with $\gamma_1 = 0$

We can also compares the graphs of an individual at different times of the outbreak. Here we chose the initial host :

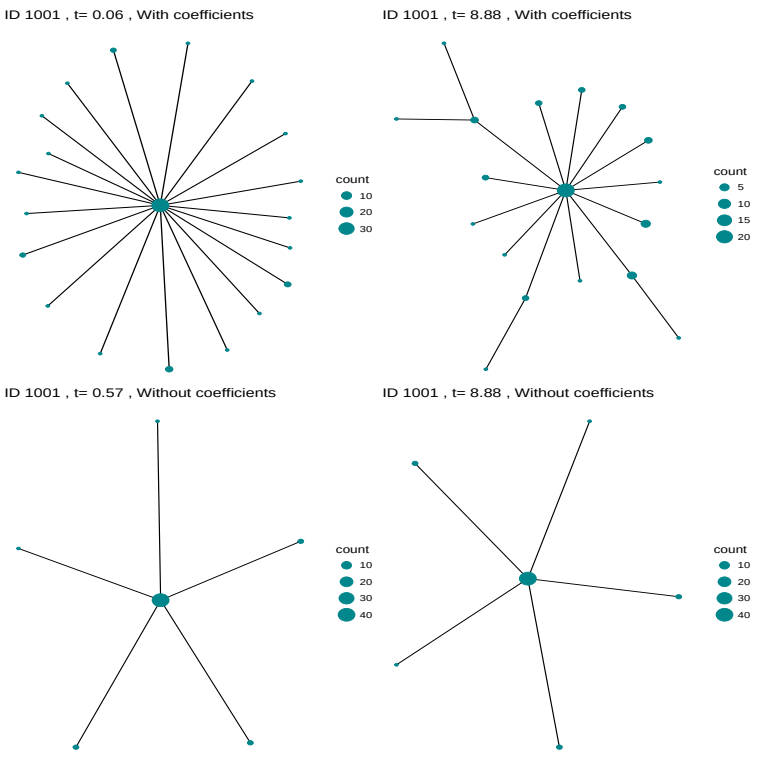


Figure 7 – Intra-host Viral Diversity without any transformation parameter

We can see that for similar times (at the beginning of the outbreak at day 8) diversity seems to be less important with $\gamma_1 = 0$. Indeed, at the beginning there are less variants and after 8 days, even if there are new variants, they are in low prorportions.

1.5 With fitness rule

A function was arbitrarily chosen to favour some variants over others. The fitness rule function is given a sequence S of M nucleotides, it takes $\bar{s}_C$ and $\bar{s}_G$ the average numbers of respectively C and G in the first $0.1M$ nucleotides of the input sequence. This function favours the sequences with many C and discriminates the ones with many G according to the following formula :

$$fitness(S) = \frac{1 + 9\ \bar{s}_C}{1 + 9\ \bar{s}_G}$$

We used the same other parameters as in the previous simulation without fitness rule. We obtained the following results :

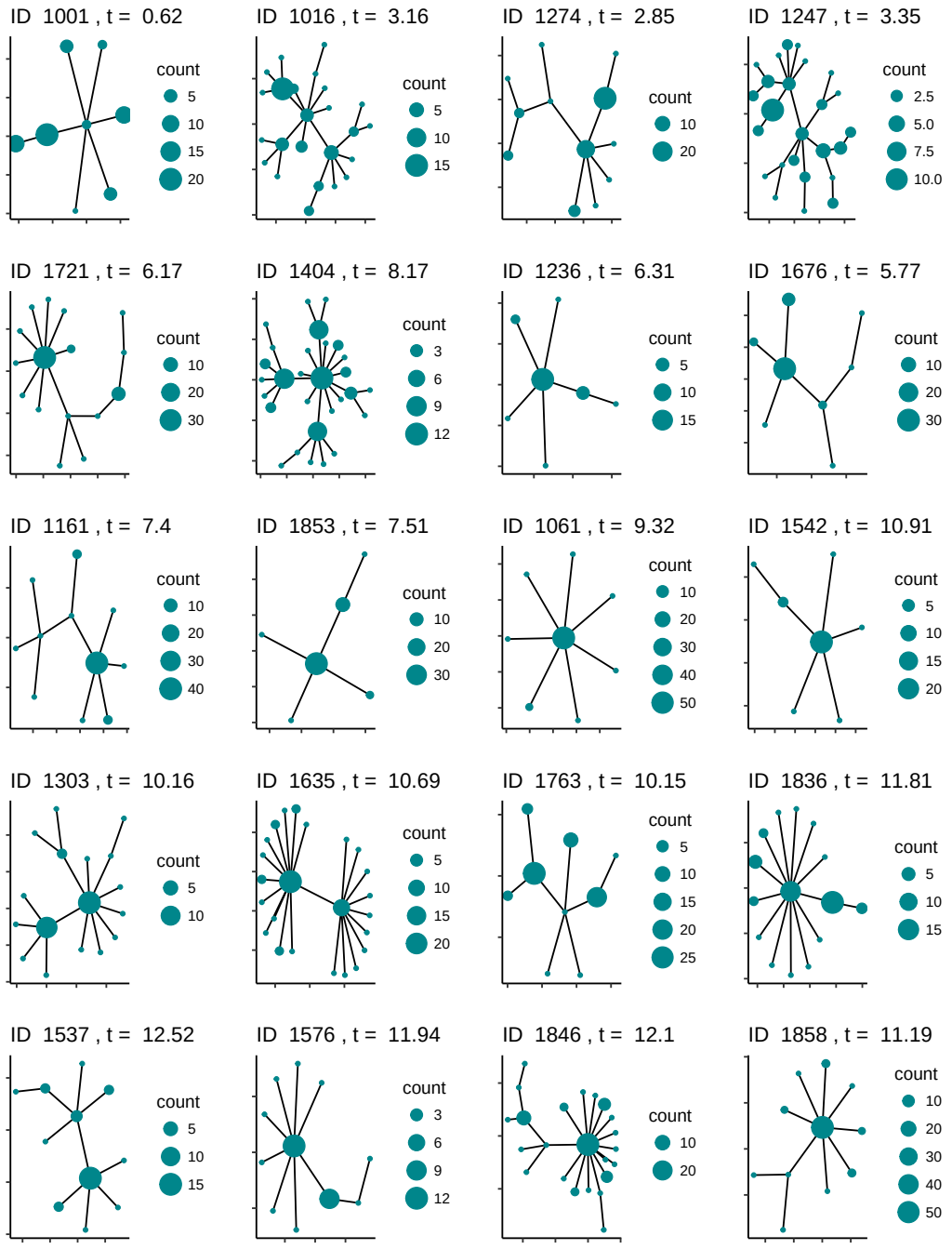


Figure 8 – Intra-host Viral Diversity with a fitness rule

With this visualisation, the results were not very different from the ones without fitness rules.

The visualisation with graphs gives us a good visual picture but it has to be confirmed by quantitative indicators.

2 Within Host Viral Diversity

In order to assess and quantify diversity, many indices and genetic distances can be used.

2.1 Assessment within one isolated host

First of all we want to study the viral genetic diversity within one host without taking into consideration the exchanges of virions between hosts, so it is an isolated host.

2.1.1 Assessment with indexes and genetic distances

Richness Estimator

The richness estimator S is the number of subtypes, so here it is the number of different genotypes or its logarithm [1]. To estimate this index on R we determine the number of non null elements in the vector of proportions.

Shannon Diversity Index

The Shannon Diversity Index is determined as following :

$$H = -\sum_{i=1}^{N} p_i \cdot ln(p_i),$$

where p_i is the proportion of individuals of the i^{th} variant subtype, with N the number of different subtypes [1]. To estimate this index on R we use the function `diversity` with the "shannon" option from the *vegan* package that takes as input a vector with the variant subtypes headcounts.

Simpson's Index

$$D = 1 - \sum_{i=1}^{N} p_i^2,$$

where p_i is the proportion of individuals of the i^{th} variant subtype, with N the number of different subtypes [1]. To estimate this index on R we also use the `diversity` function from the *vegan* package but with the "simpson" option.

Pielou's Evenness

$$J_1 = \frac{H}{ln(S)},$$

where H is the Shannon Diversity Index and S the richness estimator [1][7]. To estimate this index we use the indexes S and H.

Nucleotide Diversity or pairwised distance (p-distance)

We call π the nucleotide diversity. In one population the nucleotide diversity π can be determined as follows [5] : we consider N_i the number of individuals in the i^{th} haplotype (in a population of V individuals) and $\hat{x}_i = \frac{N_i}{V}$ is the frequency of the i^{th} haplotype in the population. We determine δ_{ij} the number of different nucleotide between the haplotypes i and j and we can determine ν the average number of different nucleotides for a set of N sequences in the population : $\hat{\nu} = \frac{N}{N-1} \sum_{i=1}^{N} \sum_{j=1}^{N} \hat{x}_i \cdot \hat{x}_j \cdot \delta_{ij}$. We define π_{ij} the probability that the haplotypes i and j are different for every nucleotide :

$$\pi = \sum_{i=1}^{N} \sum_{j=1}^{N} x_i \cdot x_j \cdot \pi_{ij} \text{ and } \hat{\pi} = \frac{\hat{\nu}}{M},$$

with M the length of the nucleotide sequence. To estimate this index we use the `nuc.div` function from the *pegas* package that takes as input a DNAbin set of sequences (an R object that gives the number of sequences, their length and their nucleotide proportion).

Jukes and Cantor Distance

This an estimated evolutionary distance between two sequences X and Y :

$$d_{JC\{XY\}} = \frac{-3}{4} ln(1 - \frac{4}{3} f_{XY}),$$

where f_{XY} is the dissimilarity between X and Y (fraction of observed differences) [2]. To estimate this index we used the `dist.dna` function from the *ape* package that also takes as input a DNAbin set of sequences. The output is a distance matrix of dimension $(N \text{x} N)$ (so to have one diversity value we take the mean on the matrix).

Nei's Distance

We consider two sequences X and Y of same length M and $x_{i\ell}$ and $y_{i\ell}$ the frequency of the i^th nucleotide at a locus $\ell = \{1, ..., M\}$. Nei distance is defined as [6] :

$$d_{Nei} = -ln \left(\frac{\sum_{\ell=1}^{M} \sum_{i=\{A,C,G,T\}} x_{i\ell} y_{i\ell}}{\sqrt{\left(\sum_{\ell=1}^{M} \sum_{i=\{A,C,G,T\}} x_{i\ell}^2 \right) \left(\sum_{\ell=1}^{M} \sum_{i=\{A,C,G,T\}} y_{i\ell}^2 \right)}} \right)$$

To estimate this index on R we use the `nei.dist` function from the *poppr* package that also takes in input a DNAbin set of sequences. This output is also a distance matrix.

2.2 Modelling

For the three genetic distances, the determination is quite long to compute. This is due to the fact that we handle huge matrices with about 20000 rows at each time step. In order to reduce this time we do not apply the diversity functions to the whole set of sequences, we sample some of them and we apply the diversity functions on these sampled sequences. The number of sampled sequence is chosen by the user, the higher it is and the better the diversity estimation is.

For a sample size of 100 sequences, a time step of 0.1 and the same parameters as in the previous simulation we get the following results for one simulation of the infection of one host :

With transformation parameters and no fitness rule

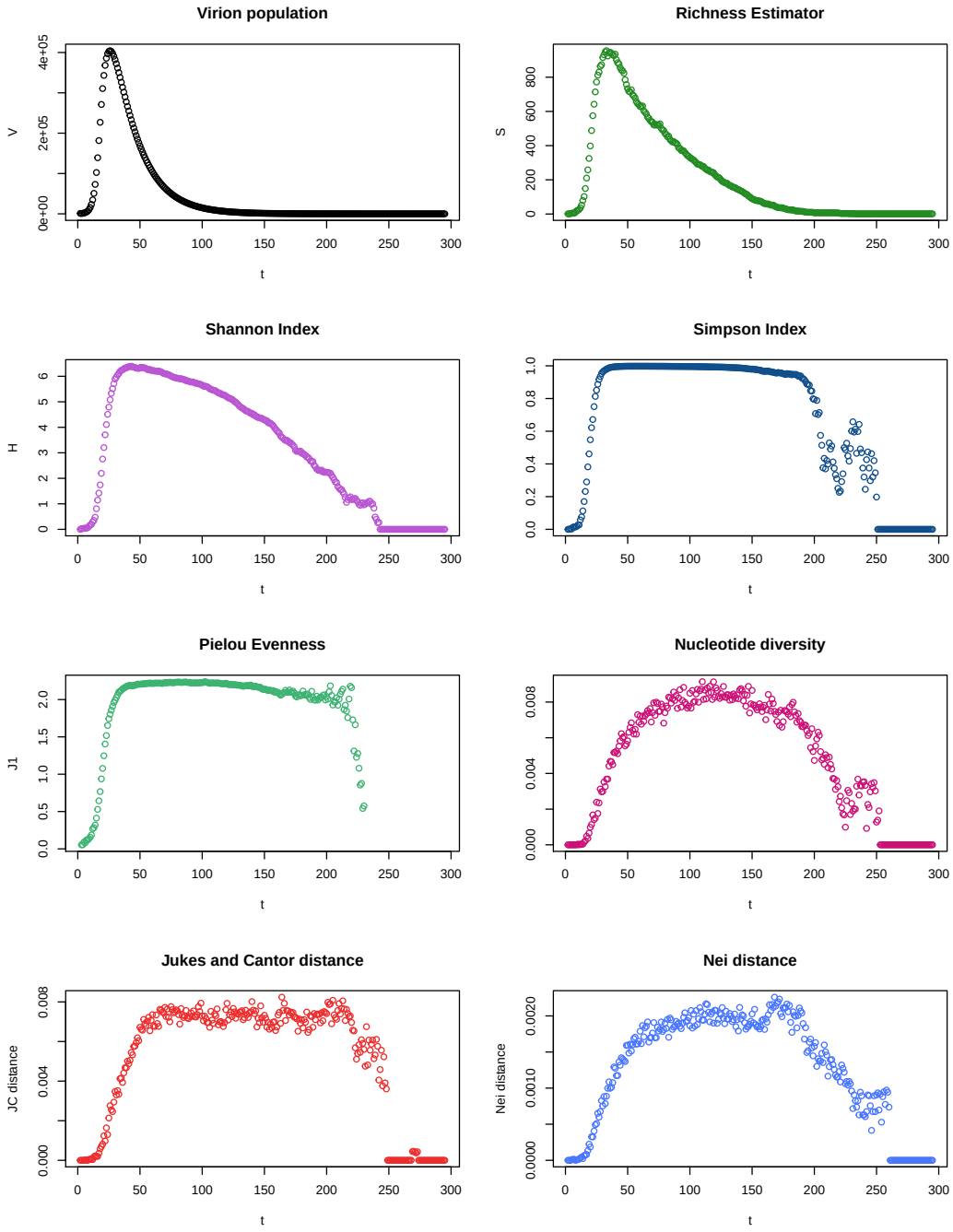


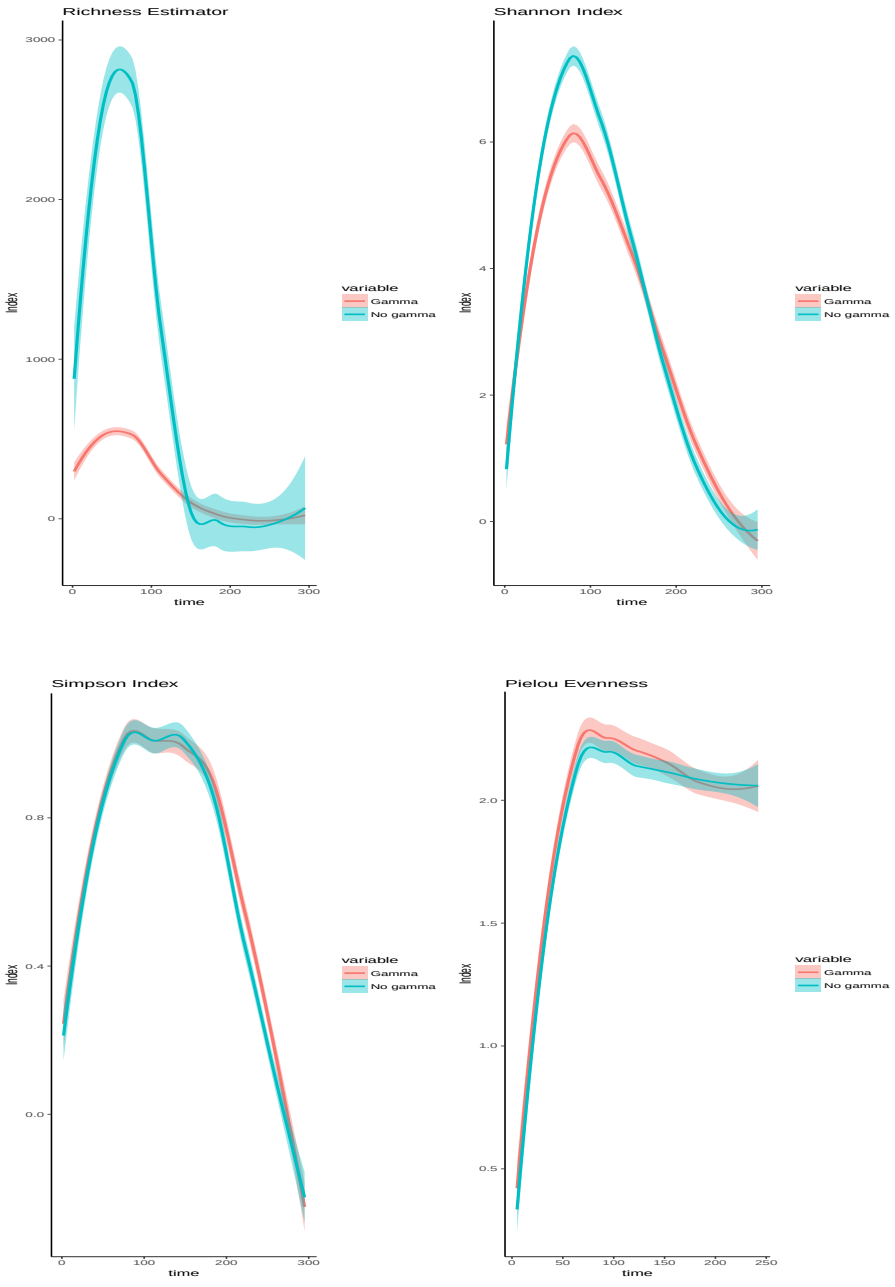
Figure 9 – Representation of several viral genetic diversity indexes in function of time during an infection with transformation parameters

We can see that the richness estimator follows the variation of the viral population. Indeed when the viral multiplication is important there are more mutations so diversity is more important, while when the population decreases and all the virions die, diversity also decreases. The decreasing is more progressive for the Shannon index but for the other indexes and genetic distances we can see that it takes a long time for diversity to decrease, even though the viral population has already drastically decreased. It means that when the virion population decreases, the surviving variants are genetically different enough so that the values of diversity indices doesn't decrease right away.

Now we would like to compare these results with the ones from a simulation without transformation parameters.

Without transformation parameters

In this case, we don't apply any gaussian noise to the multiplication probability. To compare the different results we displayed loess regressions and their confidence intervals :



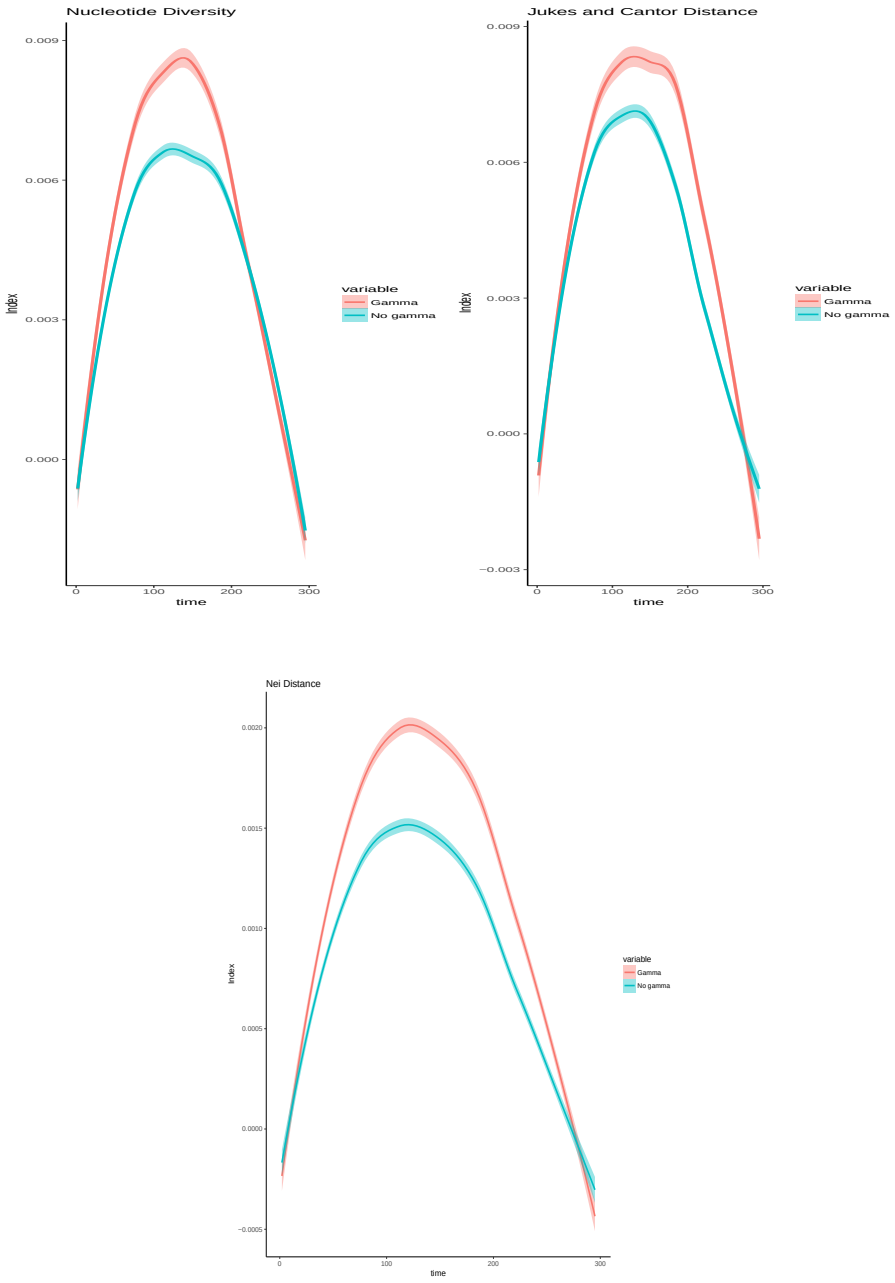


Figure 10 – Representation of several viral genetic diversity indexes in function of time during an infection with or without transformation parameters

- We obtain different results according to the index used for diversity assessment. Indeed, richness estimator, shannon, simpson and pielou are binar indices, they indicate if two sequences are identicals or not. On the contrary, the genetic distances take into consideration the degree of difference between two sequences. The use of transformation parameters γ has two effects : multiplication probabilities can either drop to 0 or there can be a diversification of variants (more different variants can multiply). According to the index we choose for diversity assessment, one of these effects is privileged : for the richness estimator and Shannon index the dropping of probabilities to 0 seems to be noticeable diversity values are higher with $\gamma_1 = 0$. For Simpson index and Pielou evenness there are no difference between the two simulations, so the diversity supposedly created by the transformation parameters is not noticeable with these indices.

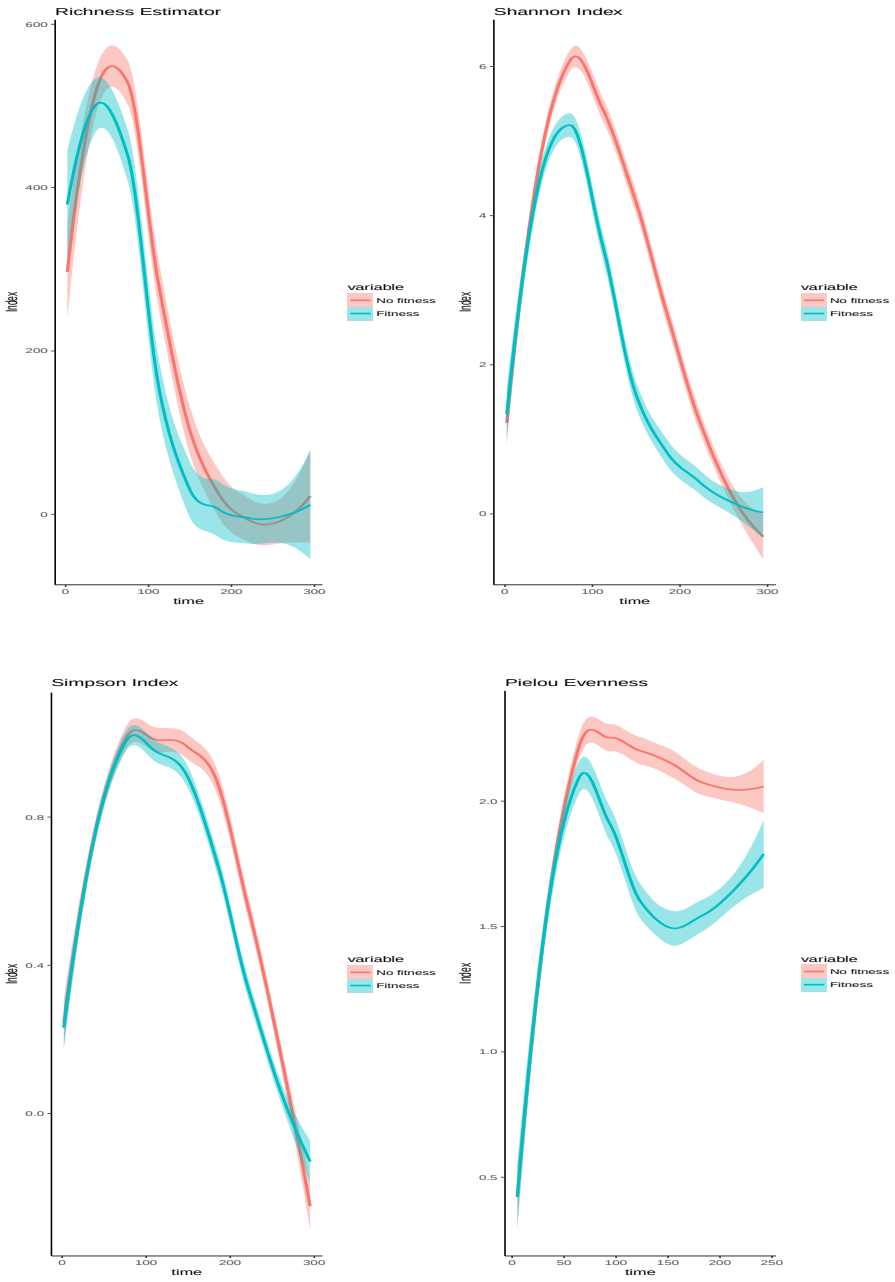
- For the genetic distances we observe that the other effect is privileged : diversity values both take the shape of bell curves but during the viral spike (i.e between days 1 and 3) diversity is more important

for the simulation with the γ transformation parameters. This is what we expected, as the addition of the gaussian noise with this parameters to the multiplication probabilities enables a more important multiplication of variants in lower proportions so the viral diversity is increased.

The viral diversification we wanted to create with the γ transformation parameters is overall noticeable with genetic distances, the other indices dont't reveal it. However these results are for one set of parameters, so maybe for different parameters we would have different results for diversity values.

With a fitness rule

We chose to use the same arbitrary fitness rule $fitness$ as in the previous simulation (we favour some variants over others). To compare the different results we displayed loess regressions and their confidence intervals :



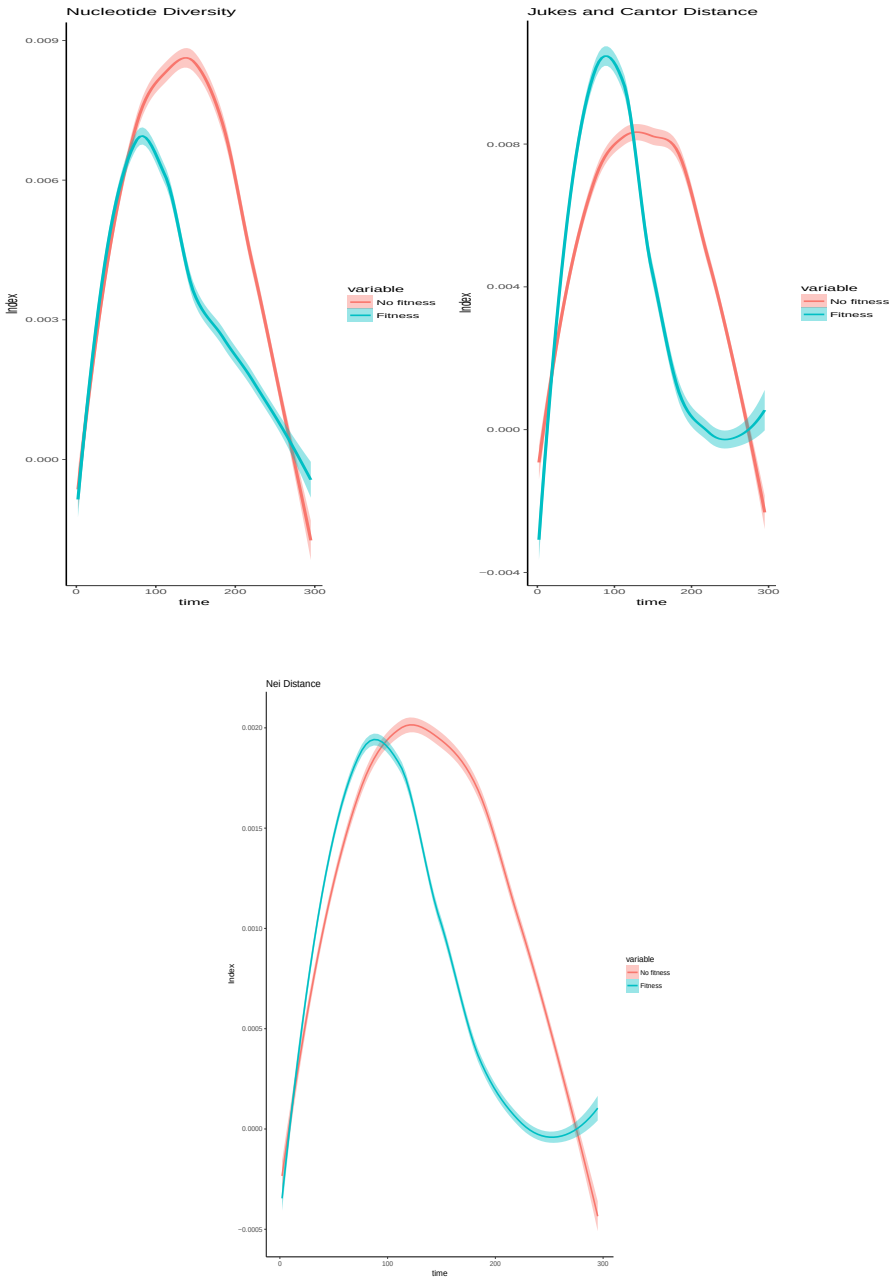


Figure 11 – Representation of several viral genetic diversity indexes in function of time during an outbreak with or without a fitness rule

- For the richness estimator, diversity values are similar between the two simulations.
- For Shannon index, at the beginning of the infection values are similar but then the ones of simulation with fitness reach a lower spike and decrease more abruptly. However at the end both simulations give similar values.
- For Simpson index, both simulations give similar diversity values until the spike. Then values decrease faster for the fitness simulation, and at the end values are similar.
- For Pielou evenness, the values of the fitness simulation are unusual : at the beginning both simulations are similar but then for the fitness one values drop faster but then increase again, while the values from the without fitness simulation still are decreasing. When there are many variants, a fitness rule decreases diversity (as it only favours some variants) but when the viral population drops, these favoured variants survive and enable the creation of a new diversity (2nd diversity increase).

- For the nucleotide diversity (p-distance), Jukes and Cantor distance and Nei distance we make the same observation about the faster decrease of the values from the fitness simulation.

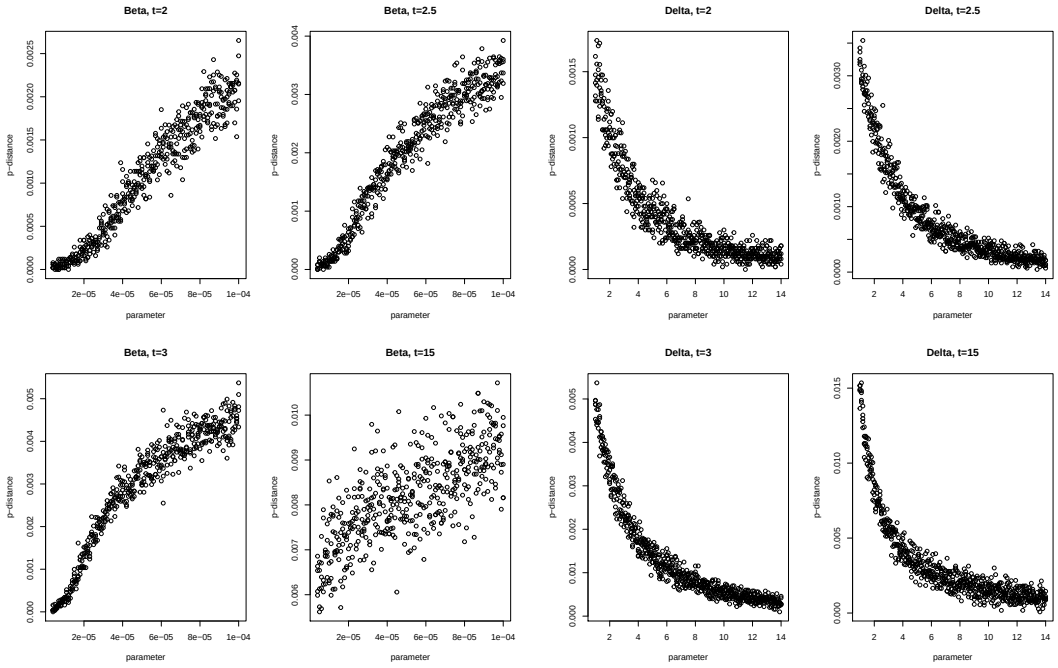
The use of a fitness function lowers diversity after the viral spike (and even before for some diversity indices). Actually the decrease starts sooner and is more progressive, while without fitness it starts later but more abruptly, so at the end of the infection diversity values are quite similar between the two simulations.

2.2.1 Sensitivity analysis

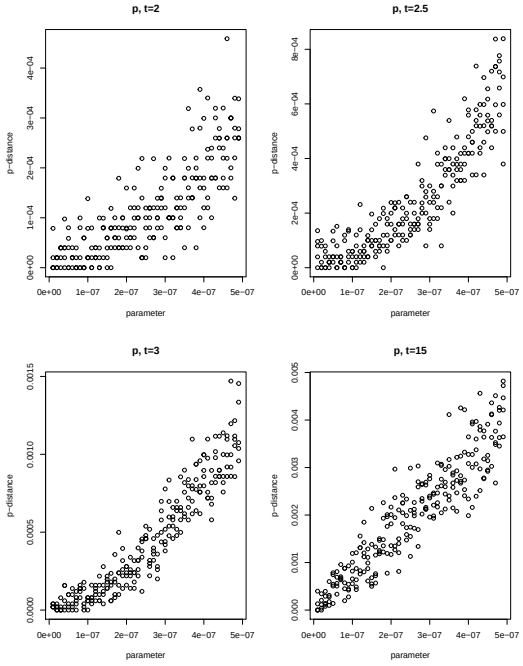
We want to determine the influence of the variation of one parameter when all the others are constant. We use the p-distance method to assess diversity for each value of the parameter of interest. We studied diversity at four different times : $t = 2$ before the viral spike, $t = 2.5$ at the viral spike, $t = 3$ juste after the viral spike and $t = 15$. We ran five diversity assessment simulations per parameter per time.

Results

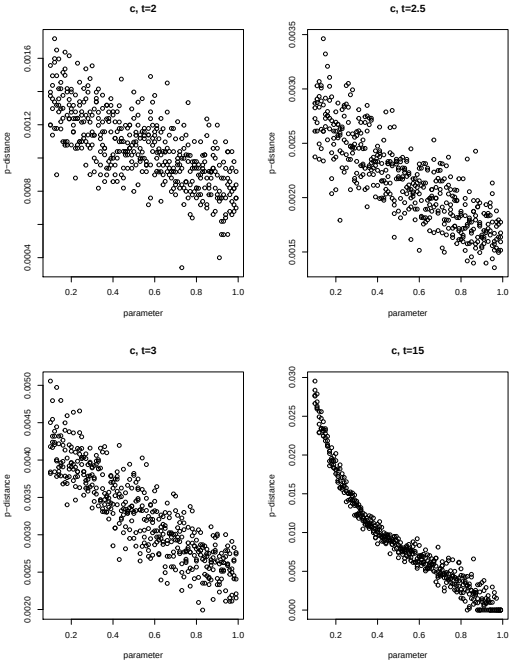
We plot the variable to explain, that is the p-distance, on the y-axis and the explicative variable, that is the studied parameter, on the x-axis. In the first instance we compare the results at the four different times.



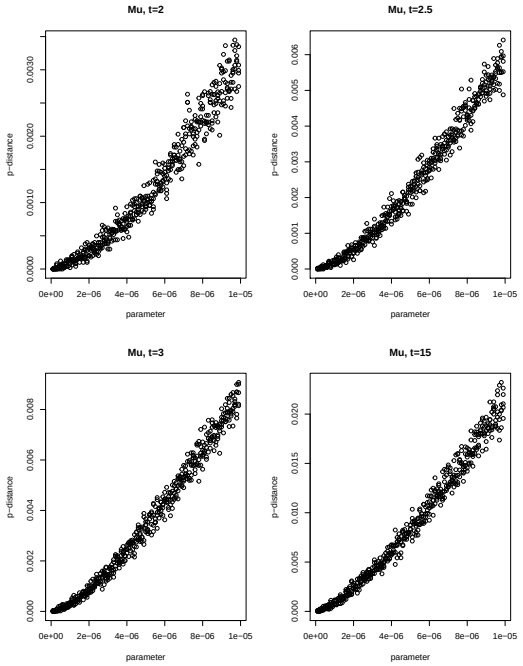
a) β parameter (infection rate) **b)** δ parameter (death rate of the infected cells)



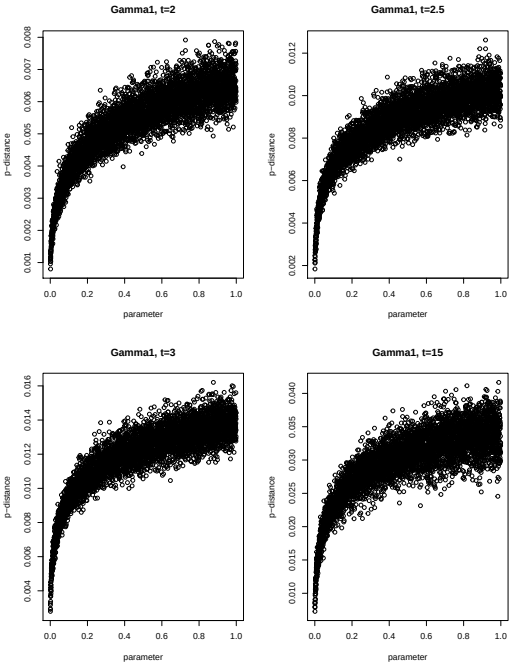
c) p parameter (virions production rate)



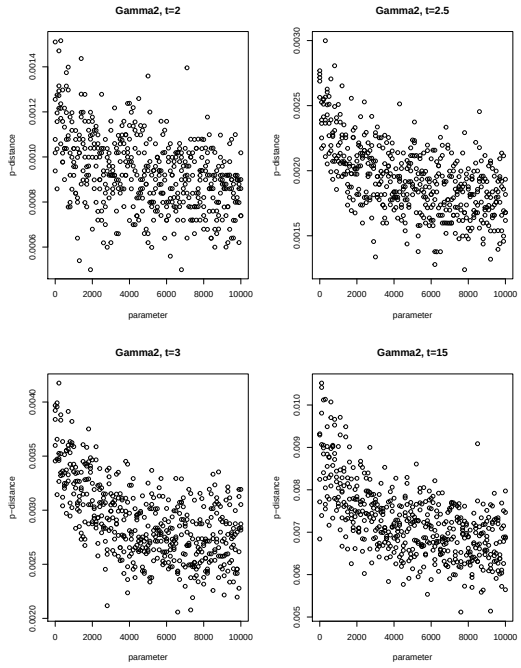
d) c parameter (virions clearance rate)



e) μ parameter (mutation rate)



f) γ_1 parameter (transformation parameter)



g) γ_2 parameter (transformation parameter)

Figure 12 – Variation of the p-distance for different parameters at different times

We can see that for some parameters the data distribution doesn't change with time, but for others there are some changes : for the β parameter (infection rate) dispersal increases with time while for p and c parameters (virions production and death rates) the dispersal decreases with time. This observation means that these three parameters don't have the same effects at every time of the infection. Some representations seem to show a linear variation : γ_2, μ, p, β (only at $t = 15$) and c (only at $t = 2$, $t = 2.5$ and $t = 3$). The δ parameter representation seems to show an inverse variation et the γ_1 representation seems to show an exponential one.

Linear regression

We first tested linear regressions on the variation of the different parameters. We obtained the following results :

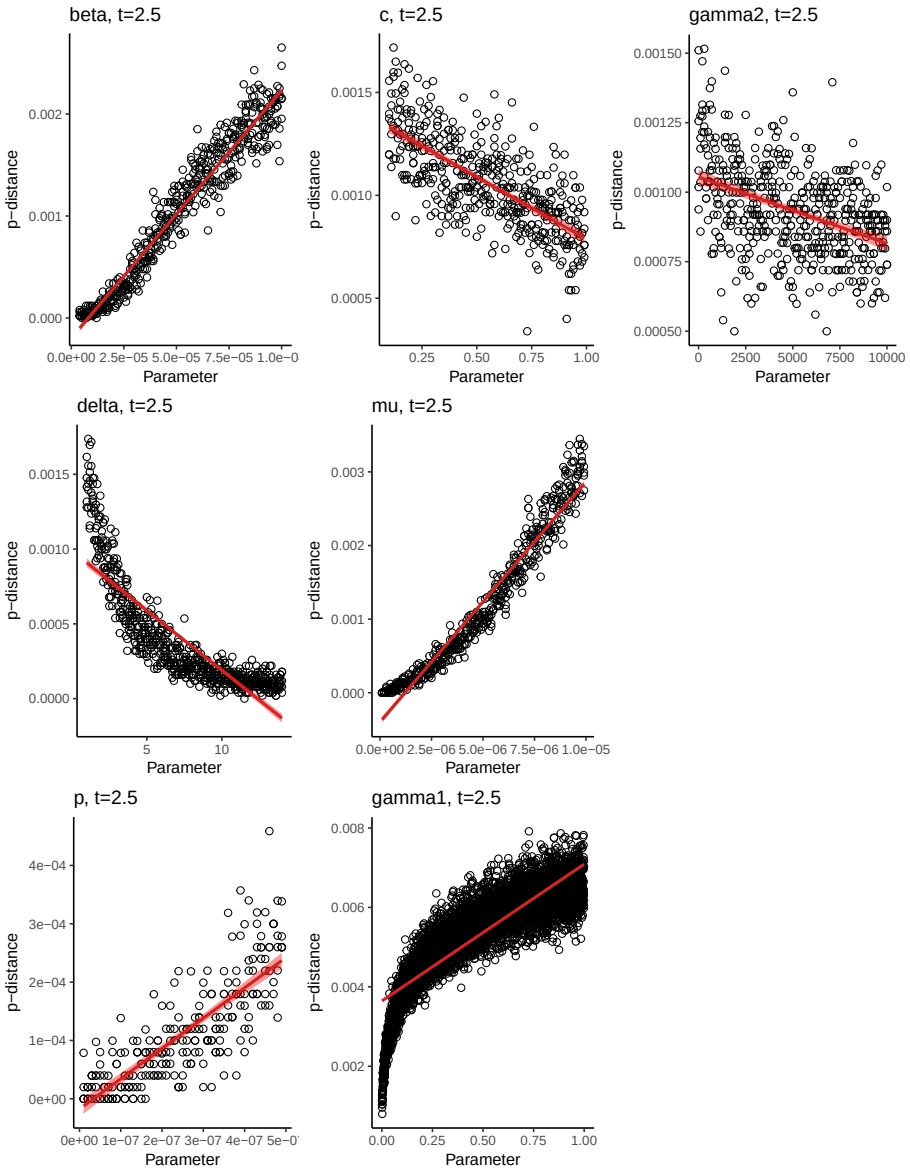


Figure 13 – Linear regression on different parameters at $t = 3$

For every regression we assessed the determination coefficient and we checked the assumptions to validate the model. We verify validity assumptions with graphic analysis : normality of the residuals (Q-Q Plot and shapiro test), null expected value and independence from x values of the residuals (Residuals Vs Fitted Plot), constant variance (Scale Location Plot) and no outlying results (Cook's Distance plot). The regressions on β, δ and γ_1 were not conclusive with low determination coefficients and unrespected validity assumptions. For the regression on β the determination rate decreases with time (as dispersal increases) but the validition assumptions are respected and we can make the opposite conclusion for the regressions on p and c. For the regression on μ the determination coefficient is high but the assumptions are never respected while for the regressions on γ_2 the coefficients are low but the assumptions are always respected. Here linear regressions are not appropriate for every parameter and for every time of the infection. We also tried linear regressions on $\frac{1}{\delta}$ and $ln\,(\gamma_1)$ but they were not conclusive either.

Nonparametric regressions [3]

The main upsides of a nonparametric regression is that it doesn't speculate any specific form for the estimator, which makes it very flexible. Here we want to do a nonparametric regression to clarify the scatter plot and the trend described by the data. The estimation of the regression function consists in finding the function that

represents at best the trend described by the data. There is a compromise between the smoothing and the flexibility of the estimator : when the smoothing is optimized the variance is lower and there is a higher bias. An estimation of y is written :

$$\hat{\mu}(x) = S_\lambda y$$

Where S_λ is the smoothing matrix or hat matrix. It depends on the smoothing type, the smoothing parameter λ and the distribution of the explicative variable.

$$df(model) = trace(S_\lambda)$$

The degree of freedom of the model (i.e the number of estimated parameters) is the trace of this matrix. This value can be used to compare different smoothing as the higher it is, the better the fitting and the flexibility of the model are.

A nonparametric regression aims at minimizing the PSE (Percent Standard Error) defined as :

$$PSE = \frac{1}{n}\sum_{i=1}^{n}\{y_i^* - \hat{\mu}(x_i)\}^2$$

With y_i^* a new observation at x_i. The best estimator of this value is the GCV (Generalized Cross Validation) defined as :

$$GCV = \frac{\frac{1}{n}\sum\limits_{i=1}^{n}\{y_i - \hat{\mu}(x_i)\}^2}{\left(1 - \frac{tr(S_\lambda)}{n}\right)^2}$$

We will determine for different regression models. We tried different methods : smoothing splines, kernel gaussian estimation, generalized additive model (gam), local weighted estimation (loess). We will specify the approach with the loess method as it is the most satisfying one.

The loess (locally-weighted running-line smoothers) method consists in a local regression. The fitting at point x_i is weighted toward the nearest data. The distance from x_i that is considered near is controlled by the smoothing parameter λ (it determines the proportion of neighbours to be taken to determine the local fitting). The weight P_{ij} of x_j at the neighbourhood of x_i is :

$$P_{ij} = \left(1 - \left(\frac{d_{ij}}{d_{max}}\right)^3\right)^3$$

When $\lambda < 1$, d_{max} is the maximum distance between two points of the considered neighbourhood and if $\lambda > 1$, $d_{max} = \lambda\, D_{max}$ with D_{max} the maximum distance between two points of the data set. The local fitting function is a polynomial of degree chosen by the user (usually 1 or 2), it is estimated by the weighted least squares method on the considered neighbourhood.

On R we implemented this regression with the `loess` function. Its default span is $\lambda = 0.75$, the default polynomial degree is 2 and the "gaussian" option for family is chosen for the least-squares fitting. The function doesn't directly return the GCV value but it returns the trace of the hat matrix with which we can determine the GCV value. We get the following results :

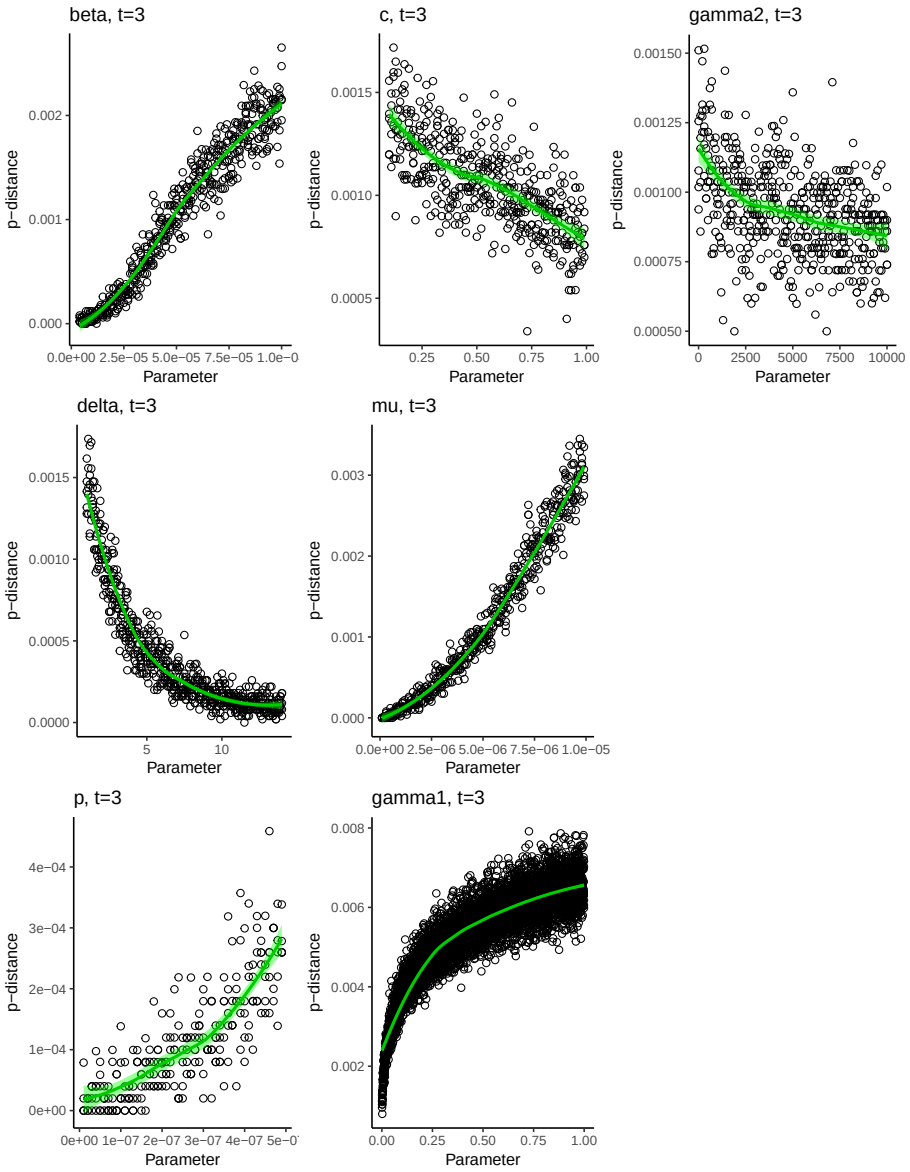
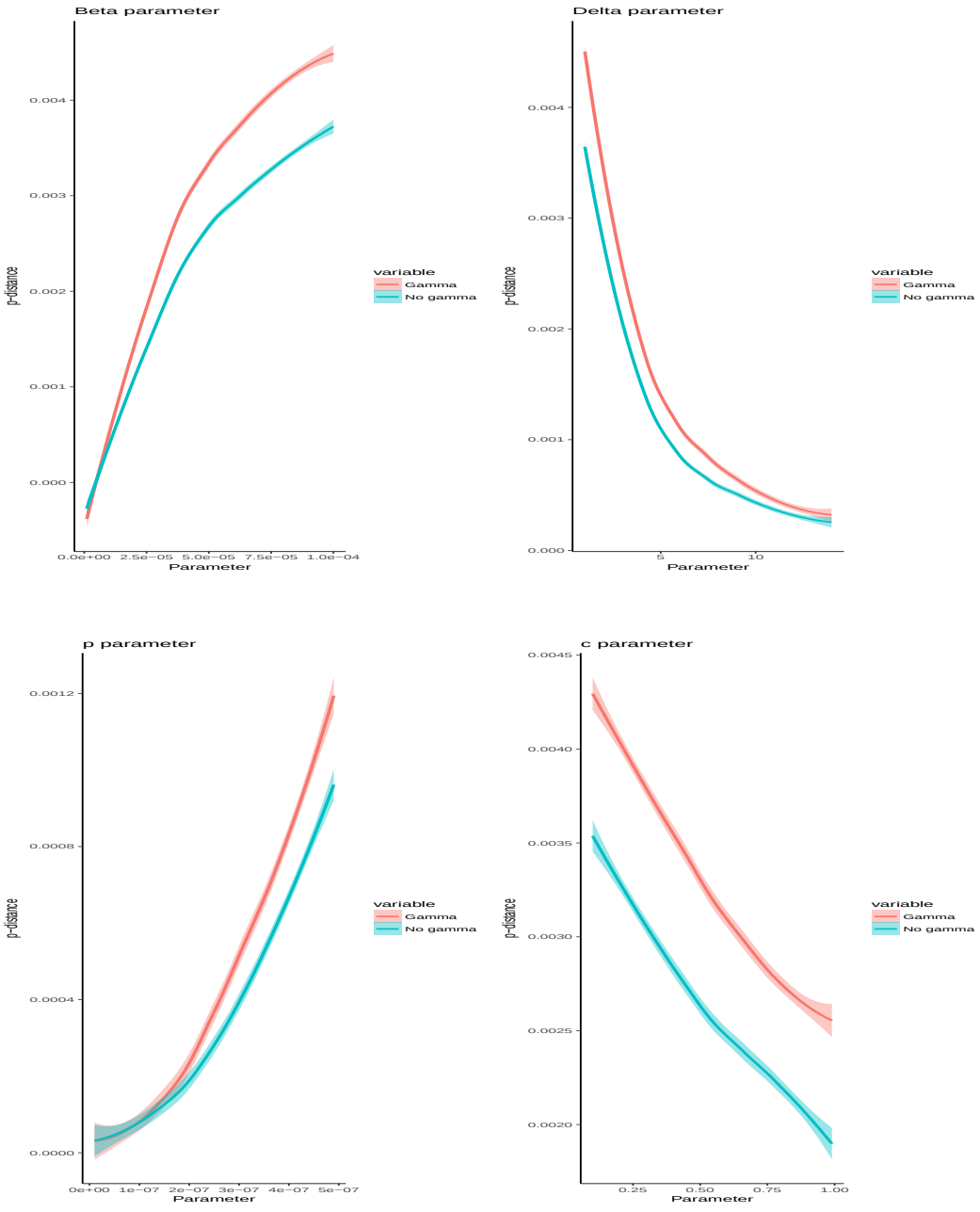


Figure 14 – Loess regression on different parameters at $t = 3$

The smoothing is very satisfactory with this regression. With the other nonparametric regressions, the variance is higher so the smoothing isn't as good and there are many oscillations. Even though we determined the GCV for each regression model, it doesn't enable us to really compare different regression methods (it is usually used to compare different parameters for one regression method). Indeed, we can't compare the smoothing parameters of the different methods as they are not equivalent. The best method to choose the best regression is the graphic visualization, even if it isn't a precise quantitative one.

2.2.2 Comparison between simulations with/without γ transformation parameters

We want to compare sensitivity analyses from simulations with and without γ parameters on every parameter (except the γ parameters) in order to identify potential interactions between the γ parameters and the other parameters of the model. We did analyses at time $t = 3$ in the same conditions for both simulations (diversity assessment with the p-distance, five diversity assessment simulations per parameter). As previously we did nonparametric loess regressions on the data retrieved from the two simulations to compare them. We obtain the following results :


Beta parameter
Delta parameter
p parameter
c parameter
p-distance
Parameter
variable
Gamma
No gamma

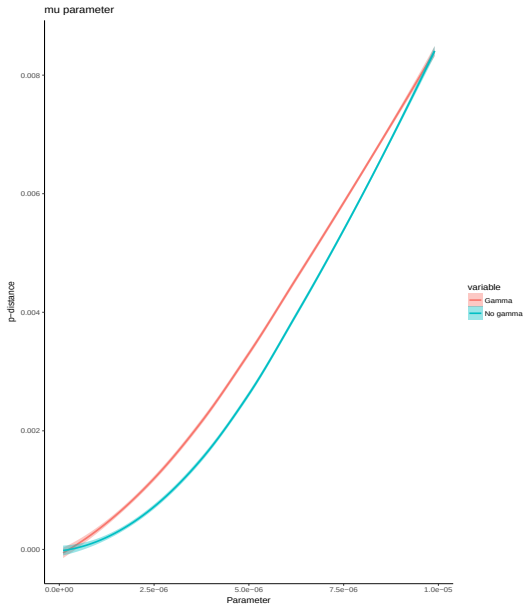
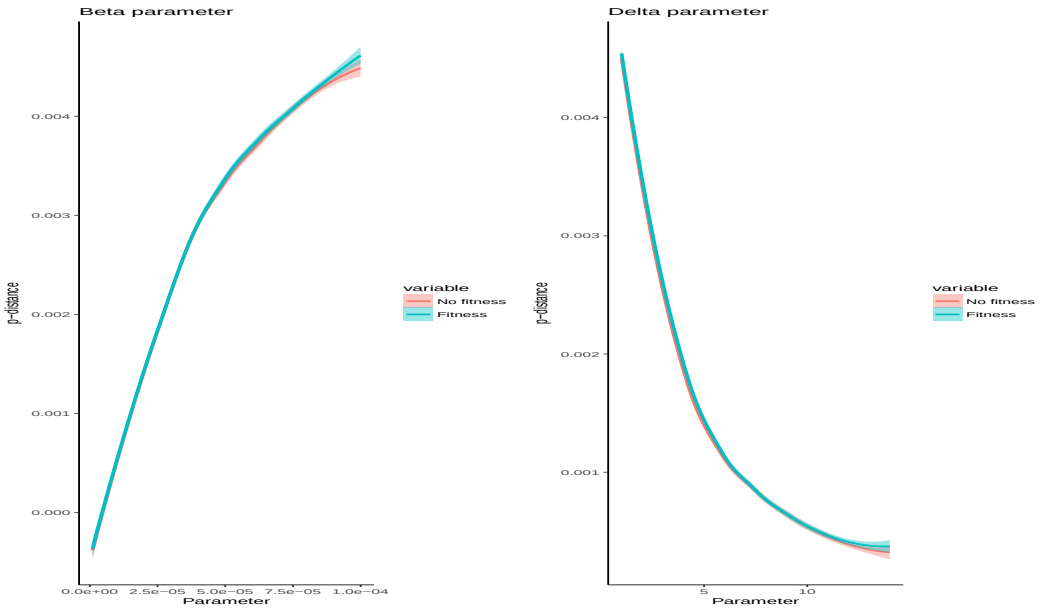


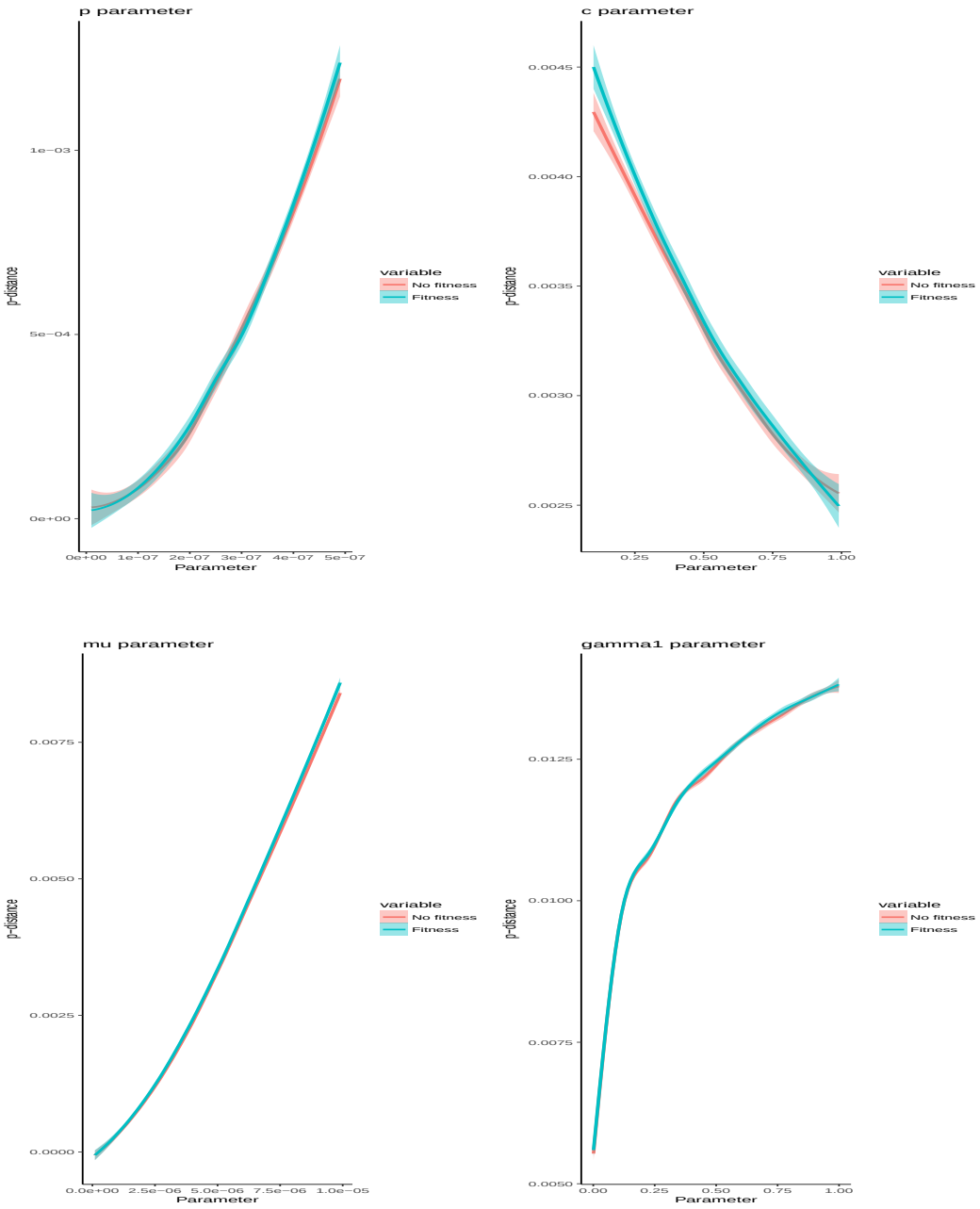
Figure 15 – Loess regressions on different parameters at $t = 3$ with or without γ parameters

- There is no interaction between the c parameter and the γ parameters as both curves are parallel. It means that the use of γ parameters doesn't impact the effect of c on diversity.
- For the others parameters there are interactions with the γ parameters. Indeed For some values of the parameters (low values for β and p, high values for δ and both for μ) there are no difference between both curves while for other values there are significant differences. It means that the values of these parameters have an effect on the way γ parameters will influence diversity.

2.2.3 Comparison between simulations with/without fitness rule

Now we want to compare sensitivity analyses from simulations with or without the fitness rule $fitness$ we previously used on every parameter of the model. We analyses at time $t = 3$ in the same conditions for both simulations (diversity assessment with the p-distance, five diversity assessment simulations per parameter). As previously we did nonparametric loess regressions on the data retrieved from the two simulations to compare them. We obtain the following results :




p parameter
c parameter
mu parameter
gamma1 parameter
p-distance
Parameter
variable
No fitness
Fitness

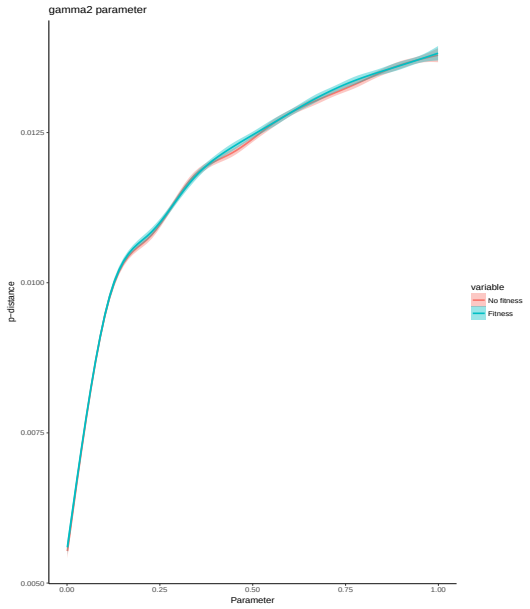


Figure 16 – Loess regressions on different parameters at $t = 3$ with or without fitness rule

For every analysed parameter, diversity values are the same for the two simulations. This fitness rule doesn't have any effect on the variation of the p-distance.

2.2.4 Limits to the sensitivity analysis

In order to have more complete results about the effects of the different parameters on diversity, we should do a multidimensional analysis with an assessment pf the significance of each parameter.

3 Conclusion

With this model, we have a better understanding of what generates genetic diversity. The quantitative definition of genetic diversity depends on the chosen diversity index, and in the framework of the SMITID project, the most interesting ones are genetics distances. There are two kinds of parameters that influence this genetic diversity : first of all, there are kinetics parameters, as when there are more variants, they are more likely to be different. However this conclusion might not be entirely true as there are less extreme situations with large populations that with smaller ones (so extreme diversification is less likely to happen within a large population of variants). There are also evolution parameters that impact genetic diversity such as the transformation parameters we introduced in this model, or the mutation rate. There might even be an interaction between these two kinds of parameters, even if it is not clear which ones influence the others.

4 Acknoledgements

I would like to thank the whole BioSP team for welcoming me in their unit, and especially Samuel who supervised me during this internship. I really enjoyed working on this project and I've learnt many valuable things in terms of statistics, mathematics and modelling, but also about the universe of research.

5 Annex

5.1 Other types of nonparametrics regressions

Smoothing splines regression [9]

A spline is a set of continuous functions that are polynomial slices. The knots are the closure points of the intervals. If we consider n knots, we consider $(g_1, ..., g_n)$ n truncated power basis functions for natural cubic splines with n knots at $(x_1, ..., x_n)$. The estimator for a smoothing spline regression has the form :

$$\hat{\mu}(x) = \sum_{j=1}^{n} \hat{\beta}_j g_j(x)$$

The β coefficients are chosen to minimize the expression $||y - G\beta||_2^2 + \lambda \beta^T \Omega \beta$, avec $G \in \mathbb{R}^{n \times n}$ the basis matrix $G_{ij} = g_j(x_i)$ for $i, j = 1, ..., n$ and $\Omega \in \mathbb{R}^{n \times n}$ the penalty matrix $\Omega_{ij} = \int g_i^{"}(t) g_j^{"}(t) dt$ for $i, j = 1, ..., n$.

The estimated β parameters are defined as :

$$\hat{\beta} = \left(G^T G + \lambda \Omega\right)^{-1} G^T y$$

So the estimator can also be written :

$$\hat{\mu}(x) = g(x)^T \hat{\beta} = g(x)^T \left(G^T G + \lambda \Omega\right)^{-1} G^T y$$

With $g(x) = (g(x_1), .., g(x_n))$.

The value to be minimized is :

$$\sum_{i=1}^{n} \left(y_i - \hat{\mu}(x_i)\right)^2 + \lambda \int \left(\hat{\mu}''(x)\right)^2 dx$$

To implement this regression on R we used the `smooth.spline` function to plot the prediction and determine the GCV (return value of the function). The smoothing parameter λ is determined by the function with by minimization of the GCV.

Kernel estimation regression

The estimation is calculated as the weighted average of the observations y_i, with the same principle as the loess method. The weight given to a point i depends of the chosen kernel. A kernel is a non-negative real-valued integrable function : gaussian, epanechnikov, uniform, triangle, tricube... Here we chose the gaussian kernel. The estimator has the form :

$$\hat{\mu}(x_i) = \frac{\sum_{j=1}^{n} d\left(\frac{x_i - x_j}{\lambda}\right) y_i}{\sum_{j=1}^{n} d\left(\frac{x_i - x_j}{\lambda}\right)}$$

With d the gaussian kernel function defined as $d(t) = \frac{1}{\sqrt{2\pi} exp\left(\frac{1}{2} t^2\right)}$. For a kernel estimation, the smoothing parameter λ is also called bandwidth. To do this regression on R we used the `ksmooth` function. We determined the bandwidth to use with the `dpill` function from the *KernSmooth* package, for a given data this function determines the best bandwidth to minimize Mallows's Cp. It is another estimator of the PSE, simpler than the GCV but biased. The `ksmooth` function only returns the predictions, so to determine the GCV we had to implement the computation of the smoothing matrix. For a kernel estimation it is defined as :

$$S_{ij} = \frac{c_i}{\lambda} d\left(\frac{y_i - y_j}{\lambda}\right)$$

With c_i a constant so that $\sum_{i,j} S_{ij} = 1$ [4].

Generalized Additive Model regression

This method consists in fitting a set of function f_j so that :

$$\mu = \beta_0 + \beta_1 f_1(x_1) + ... + f_n(x_n)$$

There is an important diversity of function f_j : they can be paramtric or not, they can be smoothin splines or locally weighted running-line smoothers, they can be determined by boosting (requires bootstrapping)... The estimator is :

$$\hat{\mu}_i = \beta_0 + \sum_{j=1}^{n} f(x_{ij})$$

With x_{ij} the value of the j^{th} predictor for the i^{th} observation.

We implemented this method on R with the `gam` function from the *mgcv* package. This function doesn't return the GCV value, but we used the `pen.edf` function on the `gam` object to determine the effective degree of freedom so then we could determine the GCV.

References

[1] M. Anderson, J. Whitlock, and V. Harwood, *Diversity and Distribution of Escherichia coli Genotypes and Antibiotic Resistance Phenotypes in Feces of Humans, Cattle, and Horses*, Applied and Environmental Microbiology, (2006).

[2] M. M. Deza and E. Deza, *Encyclopedia of Distances*, (2016), p. 456.

[3] N. Duval, *La régression non paramétrique multidimensionnelle*, Mémoire présenté à la Faculté des études supérieures de l'Université Laval dans le cadre du programme de maîtrise en statistique pour l'obtention du grade de Maître des sciences.

[4] T. Hastie and R. Tibshirani, *Generalized Additive Models*, CHAPMAN HALL/CRC.

[5] K. Holsinger and R. Mason-Game, *Hierarchical Analysis of Nucleotide Diversity in Geographically Structured Populations*, Genetics Society of America, (1996).

[6] $https://en.wikipedia.org/wiki/Genetic\_distance$, *Genetic Distance*.

[7] P. Legendre and L. Legendre, *Numerical ecology*, (2012), pp. 243–245.

[8] A. Perelson and A. Smith, *Influenza A Virus Infection Kinetics : Quantitative Data and Models*, Wiley Interdiscip Rev Syst Biol Med, 3 (2012).

[9] R. Tibshirani, *Smoothing Splines, Advanced Methods for Data Analysis*, Lectures Notes for Statistics class at Carnegie Mellon University.