Rapport du Projet de Spécialité

Aghelias NasrAllah - Badaoui Abdessamad -
Cherkaoui Ayoub - Ziri Ossama

08/02/2023 - 15/05/2023

Professeur :
Frédéric Pétrot
frederic.petrot@univ-grenoble-alpes.fr

Table des matières

1 Introduction **2**
1.1 État de l'art . . . 2
1.2 Formulation du problème d'un point de vue biologique . . . 2
1.3 Formulation du problème d'un point de vue statistique . . . 3

2 Description des ressources fournies **3**

3 Méthodes **4**
3.1 Dichotomie . . . 4
3.2 Rétrécissement d'intervalle . . . 5
3.2.1 Principe . . . 5
3.2.2 Prolongement à 2 dimensions . . . 5
3.3 Gradient descendant . . . 6
3.3.1 Principe . . . 6
3.3.2 Prolongement à 2 dimensions . . . 6
3.4 Metropolis Hastings . . . 6
3.5 Utilisation du package ABC . . . 7
3.5.1 Comment utiliser le package ABC pour l'inférence de μ ? . . . 7
3.5.2 Cross Validation . . . 8
3.5.3 Inférence de μ en utilisant le package ABC . . . 9
3.5.4 Limitations en 2 dimensions . . . 10

4 Résultats **11**
4.1 Comparaison des méthodes en une dimension . . . 11
4.2 Comparaison en deux dimensions . . . 11
4.3 Précision en une dimension . . . 12
4.4 Temps d'exécution . . . 12

5 Conclusion **13**

1 Introduction

1.1 État de l'art

L'étude du processus de génération de mutants, c'est-à-dire les modifications génétiques se produisant dans un échantillon donné sur une période donnée, est l'un des problèmes les plus anciens en biologie computationnelle.

Les approches traditionnelles pour étudier ce mécanisme reposent sur l'estimation du taux de mutations (la fréquence des modifications génétiques) à partir d'expériences simples réalisées in vitro (des expériences menées dans un environnement contrôlé en dehors d'un organisme vivant) en laboratoire. Une des mesures couramment employées à titre d'exemple est l' évaluation phénotypique : si les mutations produisent des effets visibles, ils peuvent être détectés par des mesures directes, telles que l'observation de changements dans la morphologie ou la fonction des cellules ou des organismes étudiés. Une fois on a les données l'estimation est généralement effectuée en utilisant la méthode du maximum de vraisemblance avec un modèle analytique simple.

La fonction de vraisemblance représente la loi statistique que le modélisateur estime que les nombres de mutants suivent. Par exemple, on peut utiliser comme modèle pour la fonction de vraisemblance la loi de Poisson avec un paramètre μ pour représenter le taux de mutation que l'on cherche à estimer. L'objectif est de déterminer la valeur du paramètre qui maximise cette fonction de vraisemblance, c'est-à-dire celle pour laquelle les données observées sont les plus probables à être observées.

Il existe plusieurs approches pour estimer μ en utilisant la fonction de vraisemblance. On peut dériver la fonction et trouver la valeur de μ qui annule cette dérivée (appelée estimateur du maximum de vraisemblance), ou utiliser des propriétés telles que la moyenne ou la variance de la loi pour estimer ce paramètre (appelés estimateurs des méthodes de moments).

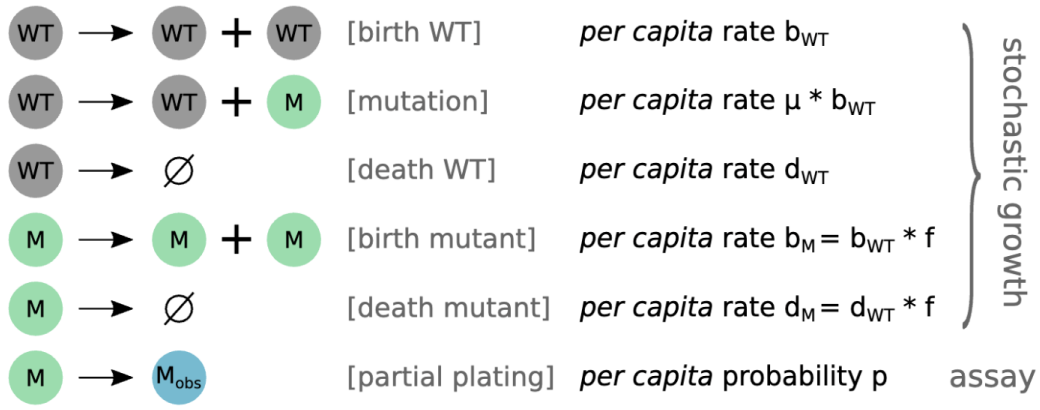
Cependant, les modèles qui décrivent le processus de génération de mutants peuvent être très complexes, avec de nombreux paramètres inconnus ou difficiles à estimer. Ces estimations peuvent devenir simplistes dans de tels cas, ce qui peut entraîner des résultats éloignés des valeurs expérimentales qui ne prennent pas en compte d'autres paramètres qui peuvent influencer le mécanisme de génération de mutants, tels que la fitness, qui représente la valeur adaptative d'une entité dans un environnement donné. De plus, de nombreuses questions de recherche actuelles nécessitent de prendre en compte des scénarios expérimentaux qui ne respectent pas les hypothèses fortes considérées pour utiliser des modèles analytiques simples.

Les limitations importantes qui ont souvent été négligées pour établir des expressions analytiques simples ont été une source de motivation pour le développement de plusieurs méthodes permettant d'estimer les paramètres souhaités (taux de mutation, fitness, etc.) sans recourir à des formules analytiques prêtes à l'emploi.

Les méthodes que nous développerons ici pour répondre à cette problématique s'inscrivent dans le cadre des techniques modernes d'inférence statistique appelées ABC (Approximate Bayesian Computing) que nous détaillerons par la suite. Le terme inférence statistique est utilisé ici comme un terme générique mettant en exergue la nature probabiliste des résultats obtenus et la possibilité d'utiliser des modèles statistiques dans un but prédictif.

1.2 Formulation du problème d'un point de vue biologique

Nous désignons la modélisation ou la simulation de ce problème sous le nom général de BDMSS (Birth-Death-Mutation-Selection-Sampling en anglais).



Ces formules peuvent être vues comme des réactions chimiques ou comme un processus permettant la génération de nouvelles entités biologiques qui n'existaient pas auparavant. Le terme "wild type" (WT) peut être considéré comme une entité biologique de référence qui est capable de générer d'autres entités biologiques, tandis que M représente un mutant. Le paramètre f (fitness) mesure la capacité d'un organisme à survivre et à se reproduire avec succès, c'est un concept clé utilisé pour évaluer la valeur adaptative d'un individu ou d'une entité biologique dans un environnement donné. Ces paramètres sont les seuls éléments que nous devons connaître et que nous manipulerons dans ce modèle biologique.

1.3 Formulation du problème d'un point de vue statistique

Le but est d'appliquer des techniques modernes d'inférence statistique, notamment l'Approximate Bayesian Computing (ABC), qui sont des approches statistiques utilisées pour estimer les paramètres inconnus d'un modèle complexe lorsque la vraisemblance exacte n'est pas calculable ou difficile à obtenir. L'objectif principal de l'ABC est d'obtenir une approximation de la distribution postérieure des paramètres en se basant sur des simulations et des comparaisons avec les données observées.

Il existe différentes variantes de l'ABC, dont certaines utilisent des méthodes de rejet (rejection sampling) tandis que d'autres utilisent des méthodes MCMC(Markov chain Monte Carlo) pour explorer l'espace des paramètres. Dans le rejet ABC, des simulations sont générées à partir de valeurs de paramètres tirées selon une distribution a priori, puis elles sont rejetées si les résumés statistiques ne correspondent pas suffisamment aux données réelles. Les simulations acceptées sont ensuite utilisées pour construire une approximation de la distribution postérieure des paramètres. Les méthodes MCMC ABC, quant à elles, utilisent des techniques d'échantillonnage pour explorer de manière itérative l'espace des paramètres et chercher les régions de haute vraisemblance compte tenu des résumés statistiques des données.

2 Description des ressources fournies

Ayant un jeu de données obtenu pour un ou deux paramètres inconnus, notre objectif est d'obtenir une bonne estimation de ces paramètres en utilisant un simulateur (ici `atreyu_forward_simulator`).

Le simulateur `atreyu_forward_simulator` peut être généré avec la commande `make`. Ensuite on peut effectuer une simulation en utilisant ce simulateur avec la commande :

```
./atreyu_forward_simulator N0 N1 D0 mu f 1 384 > resultat_simulation.txt
```

Avec :

- N_0 : la taille de population initiale
- N_1 : la taille de population finale
- D_0 : taux de mortalité
- μ : taux de mutation inconnu.
- f : effet sur la fitness (croissance relative du mutant) 0.7

— Et le 1 et 384 qui viennent juste après sont respectivement la fraction d'échantillonage (la proportion d'individus ou de données qui sont échantillonnés à partir d'une population ou d'un ensemble de données plus vaste) et le nombre de simulations.

Les résultats de la simulation seront stockés dans le fichier resultat_simulation.txt

Estimation de mu :

Pour avoir des estimations de μ, il est préférable de remplacer la commande

```
./atreyu_forward_simulator N0 N1 D0 mu f 1 384 > resultat_simulation.txt
```

par la commande

```
./atreyu_forward_simulator N0 N1 D0 mu f 1 384 | cut -d' ' -f1
 > resultat_simulation.txt
```

Ce qui va stocker les valeurs des nombres de mutants dans le fichier `resultat_simulation.txt`. À chaque ligne de ce fichier correspond le nombre de mutants associé à une simulation.

3 Méthodes

Certaines des méthodes ABC que nous avons développées utilisent le principe du rejet (rejection sampling), telles que la méthode de dichotomie, tandis que d'autres utilisent le principe des méthodes MCMC (Markov Chain Monte Carlo), comme l'algorithme Metropolis-Hastings.

Ces méthodes vont contourner le passage par les formules analytiques par les simulations. Ainsi, nous pourrons déterminer quels paramètres de simulation (les valeurs des paramètres inconnus que nous proposons) conduisent à des simulations qui sont les plus proches des données expérimentales. Toutes les méthodes que nous développerons ont un seul objectif : déterminer ces paramètres, c'est-à-dire les meilleurs paramètres qui, une fois utilisés pour générer une simulation, produisent des données qui se rapprochent le plus des données expérimentales.

Dans toutes les méthodes, le principe est le même : à partir d'une valeur initiale pour les paramètres à estimer, nous effectuons une simulation en utilisant ces paramètres. Ensuite, nous utilisons une métrique appelée le test de Kolmogorov-Smirnov pour évaluer la distance entre les données simulées et les vraies données, c'est-à-dire à quel point elles se ressemblent. Le test de Kolmogorov-Smirnov compare les distributions empiriques des données simulées et des données réelles en calculant une mesure de distance appelée la distance à la courbe.

Cette distance de la courbe représente la plus grande différence verticale entre les courbes de distribution cumulative des deux ensembles de données. En d'autres termes, elle mesure l'écart maximal entre les proportions cumulatives des valeurs observées et simulées. Plus la distance de la courbe est petite, plus les données simulées sont similaires aux données réelles.

En fonction du résultat de ce test, nous procédons à une mise à jour des valeurs des paramètres. Nous répétons ce processus d'estimation des paramètres et de calcul de la distance de la courbe jusqu'à obtenir un résultat satisfaisant, où la distance entre les données simulées et réelles est suffisamment petite pour considérer que les paramètres ont été correctement estimés.

Afin d'estimer le taux de mutation, la valeur la plus intuitive pour commencer les simulations est la médiane du nombre de mutants fournis par le jeu de données. Cette approche est préférable à la moyenne en raison de la présence de valeurs aberrantes.

3.1 Dichotomie

Nous utilisons une approche itérative pour estimer la valeur de μ. Cette méthode démarre avec une valeur initiale de μ et détermine la direction de déplacement, soit vers la gauche, soit vers la droite. Étant donné que les valeurs de μ sont extrêmement faibles, de l'ordre de 10^{-5} à 10^{-10}, nous avons adopté une échelle logarithmique pour calculer l'incrément de déplacement.

Nous effectuons ensuite des tests de Kolmogorov-Smirnov sur les jeux de données générés par μ, μ_{gauche} et μ_{droite}, en les comparant avec le jeu de données réel. À chaque itération, nous utilisons ces résultats pour sélectionner une nouvelle valeur pour à partir de l'ancienne. Cette sélection est basée sur un critère de distance entre les données simulées et les données réelles, mesurée par la distance de la courbe donnée par le test de Kolmogorov-Smirnov. Nous répétons cette étape jusqu'à ce que la distance entre les jeux de données réelles et simulées soit inférieure à une valeur

fixée, dans notre cas 0.05. Cette condition de distance minimale sert de critère d'arrêt pour sortir de la boucle itérative.

Il convient de noter que nous ajustons également le pas de déplacement à chaque itération pour obtenir une précision encore meilleure.

On accompagne cette explication par la figure .

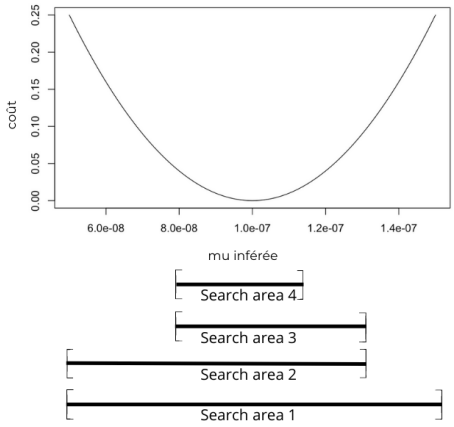


FIGURE 1 – Illustration de la méthode dichotomique appliquée à une fonction représentant la distribution idéale de la distance du test Kolmogorov-Smirnov autour de la vraie valeur de μ

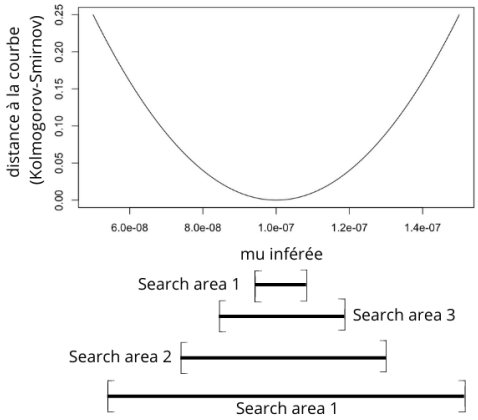


FIGURE 2 – Illustration de l'application de la méthode de rétrécissement d'intervalle sur une fonction représentant la distribution idéale de la distance test Kolmogorov-Smirnov autour de la vraie valeur de μ

3.2 Rétrécissement d'intervalle

3.2.1 Principe

Cette méthode d'inférence de μ est itérative et se déroule en plusieurs étapes. Tout d'abord, nous discrétisons un intervalle centré autour d'une valeur initiale de μ. Ensuite, nous utilisons une fonction appelée `cost_function` pour évaluer la qualité de chaque valeur discrète dans l'intervalle en comparaison avec le jeu de données original. Cette fonction retourne le résultat du test de Kolmogorov-Smirnov, qui nous permet de mesurer la similitude entre les distributions, à la fois celle des données originales et celle simulée. Ensuite, nous choisissons la valeur de μ qui a donné le meilleur résultat lors du test de Kolmogorov-Smirnov. Cette valeur est considérée comme notre nouvelle estimation de μ. Nous répétons cette étape en réduisant l'intervalle centré autour de cette nouvelle valeur de μ, ce qui nous permet de nous concentrer davantage sur la région où se trouve la vraie valeur de μ. Nous continuons cette approche itérative jusqu'à ce que la distance entre les jeux de données réelles et simulées soit inférieure à une valeur fixée, dans notre cas 0.05. Cette condition de distance minimale sert de critère d'arrêt pour sortir de la boucle itérative.

3.2.2 Prolongement à 2 dimensions

Pour étendre cette méthode en deux dimensions, nous pouvons remplacer l'intervalle de recherche unidimensionnel par un produit cartésien de deux intervalles. Cela permet de définir un espace de recherche bidimensionnel pour les deux paramètres taux de mutation μ et la fitness f. Nous commençons par une valeur initiale de μ et f et nous appliquons la fonction `cost_function` pour obtenir la mesure de la qualité de l'ajustement du modèle par rapport aux données.

Nous discrétisons les deux intervalles autour des valeurs initiales pour obtenir un ensemble discret de points bidimensionnels. Nous évaluons ensuite la fonction `cost_function` sur chacun de ces points et sélectionnons le point avec le coût le plus bas. Ce point correspond aux valeurs optimales de μ et σ qui minimisent la différence entre les données observées et celles générées par le modèle.

Nous ajustons ensuite les intervalles de recherche autour de ces valeurs optimales et répétons le processus jusqu'à ce que nous atteignons une convergence satisfaisante. Le principal défi de cette méthode en deux dimensions est la complexité de la fonction de coût et le temps de calcul nécessaire pour évaluer cette fonction pour un grand nombre de points dans l'espace de recherche.

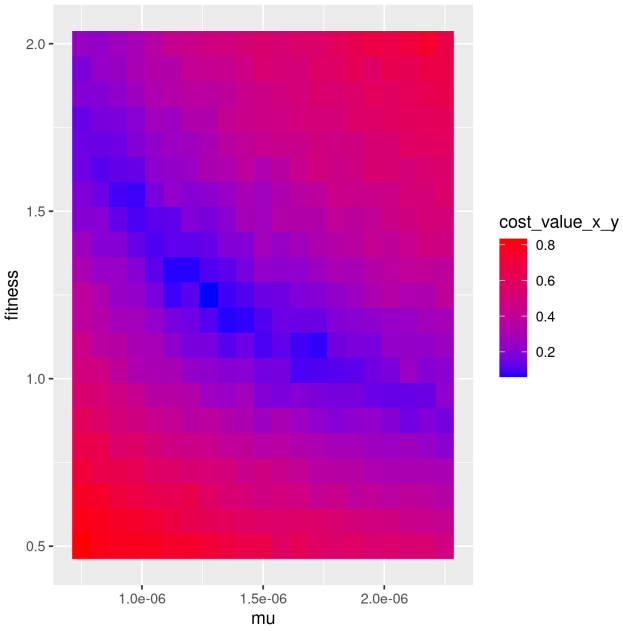


FIGURE 3 – Une carte de température des valeur de cost_ function en fonction de μ et f. On constate de faible valeurs pour le vrai tuple $\mu = 1.5 \cdot 10^{-6}$ et $f = 1.2$

3.3 Gradient descendant

3.3.1 Principe

Cette méthode d'inférence utilise une approche itérative d'optimisation pour trouver le minimum local d'une fonction convexe. Elle est spécifiquement utilisée pour inférer un paramètre à une dimension. Le processus débute par l'initialisation de la valeur du paramètre à une valeur arbitraire. Ensuite, la dérivée de la fonction de coût, qui utilise le test de Kolmogorov-Smirnov pour mesurer la similarité des distributions, est calculée par rapport au paramètre. La dérivée est calculée en utilisant le développement de Taylor d'ordre 2 adapté à l'échelle logarithmique, étant donné les valeurs de μ comme mentionné avant.Cette dérivée est utilisée pour mettre à jour la valeur du paramètre. Ce processus est répété itérativement jusqu'à ce que la distance entre les jeux de données réelles et simulées soit inférieure à une valeur fixée, dans notre cas 0.05.

Cette méthode présente certains avantages. Tout d'abord, elle permet de trouver un minimum local de la fonction convexe, ce qui est utile pour estimer le paramètre recherché. De plus, l'utilisation du test de Kolmogorov-Smirnov offre une mesure objective de la similitude des distributions, ce qui aide à guider la mise à jour du paramètre.

Cependant, cette méthode comporte également certains inconvénients. Elle est sensible aux conditions initiales, ce qui signifie que le choix initial du paramètre peut influencer le résultat final. De plus, elle peut être sujette à des minimums locaux et ne garantit pas la convergence vers le minimum global, et parfois même la convergence vers le minimum local de la fonction ce qui parfois va affecter negativement notre estimation.

3.3.2 Prolongement à 2 dimensions

Pou étendre la méthode de la descente de gradient à deux dimensions pour inférer les deux paramètres taux de mutation μ et la fitness f, on commence par initialiser les valeurs des deux paramètres à des valeurs arbitraires. Ensuite, on calcule les dérivées partielles de la fonction de coût par rapport à chaque paramètre (utilisant notamment le test de Kolmogorov Smirnov), puis on utilise ces valeurs pour mettre à jour les valeurs des paramètres. On note que les dérivées partielles sont calculées aussi comme précisé précedemment. Ce processus est répété jusqu'à ce que la distance entre les jeux de données réelles et simulées est inférieur au seuil fixé (0.05).

3.4 Metropolis Hastings

L'algorithme Metropolis-Hastings est une méthode d'échantillonnage qui fait partie de la famille plus large des méthodes MCMC. Il est couramment utilisé pour résoudre des problèmes complexes où les méthodes analytiques classiques sont difficiles ou impossibles à appliquer.

L'algorithme Metropolis-Hastings fonctionne de la manière suivante :

1/ On commence par choisir une valeur initiale pour les paramètres inconnus.

2/ À chaque itération de l'algorithme, une nouvelle valeur candidate pour les paramètres est proposée, basée sur la valeur actuelle et une distribution de proposition.

3/ On calcule la probabilité d'acceptation de cette nouvelle valeur candidate, en utilisant la comparaison entre les données observées et les prédictions du modèle statistique pour ces paramètres.

4/ La nouvelle valeur candidate est acceptée avec une probabilité déterminée par la probabilité d'acceptation calculée à l'étape précédente. Si elle est acceptée, elle devient la valeur actuelle des paramètres pour l'itération suivante. Sinon, on conserve la valeur actuelle.

Les étapes 2 à 4 sont répétées un grand nombre de fois (les itérations), permettant ainsi d'explorer l'espace des paramètres de manière itérative et d'obtenir un échantillon de valeurs des paramètres qui approxime la distribution des probabilités souhaitée.

La clé de l'algorithme Metropolis-Hastings réside dans le calcul de la probabilité d'acceptation. Cette probabilité dépend de la comparaison entre les données observées et les prédictions du modèle statistique.

Dans notre cas, nous avons choisi la loi normale comme fonction de vraisemblance (likelihood, également appelée "distribution de proposition"). Nous avons appliqué l'algorithme deux fois : la première fois pour obtenir une estimation approximative de la vraie valeur de μ, puis une deuxième fois pour améliorer la précision de l'estimation.

3.5 Utilisation du package ABC

Supposons que nous voulions calculer la distribution de probabilité postérieure d'un paramètre univarié ou multivarié, . Une valeur de paramètre θ_i , est échantillonnée à partir de sa distribution antérieure pour simuler un ensemble de données y_i , pour $i = 1, ...n$, où n est le nombre de simulations. Un ensemble de statistiques sommaires $S(y_i)$ est calculé à partir des données simulées et comparées aux statistiques sommaires obtenues à partir des données réelles $S(y_0)$ Une valeur de paramètre θ_i , est échantillonnée à partir de sa distribution antérieure pour simuler un ensemble de données y_i , pour $i = 1, ...n$, où n est le nombre de simulations. Un ensemble de statistiques sommaires $S(y_i)$ est calculé à partir des données simulées et comparées aux statistiques sommaires obtenues à partir des données réelles $S(y_0)$ à l'aide d'une mesure de distance d. à l'aide d'une mesure de distance d. Si $d(S(y_i), S(y_0))$ (c'est-à-dire la distance entre $S(y_i)$ et $S(y_0)$) est inférieure à un seuil donné, le paramètre i est acceptée. Les θ_i acceptés acceptés forment un échantillon d'une approximation de la distribution postérieure.

À cet effet, on pourra utiliser la bibliothèque ABC(Approximate Bayesian Computation) de R qui contient des fonctions prédéfinies pour les différentes algorithmes d'ABC et d'autres fonctions utiles pour bien estimer nos paramétres.

3.5.1 Comment utiliser le package ABC pour l'inférence de μ ?

On commence tout d'abord par importer la bibliothèque ABC de R. Puis on prépare les différentes variables nécessaires à notre simulateur. : $N_0 = 10$, $N_1 = 10^7$, $d_0 = 0$, $f = 0.7$. On va travailler avec un jeu de données `referenced_data` qui a été généré par un $\mu = 3.10^{-6}$.

On choisit un loi non informatif (loi uniforme) pour notre prior :

```
mu_median <- median(reference_data)/1e7
prior <- function(n) runif(n, 10^(log10(mu_median) - 1), 10^(log10(mu_median) + 1))
```

Ensuite, on choisit la moyenne comme un statistique sommaire et on simule $n_{sim} = 1000$ valeurs de μ à l'aide de notre prior, et on calcule après les statistiques sommaires associées :

```
param <- matrix(prior(n_sim * n_param), ncol = n_param)
sumstat <- matrix(apply(param, 1, modele), ncol = 1)
```

avec la fonction `model` est la fonction qui permet de calculer la statistique sommaire pour chaque simulation à l'aide de `./atreyu_forward_simulator` :

```
modele <- function(mu) {
    commande <- sprintf("./atreyu_forward_simulator %e,%e %e %e 1 384 >
                TauxMutation.txt; cut -d' ' -f1 TauxMutation.txt >
                TauxMutation2.txt", N0, N1,d0, mu, fmutant)
    system(commande)
    generated_data <- scan("TauxMutation2.txt")
    return (mean(generated_data[generated_data<filter]))
}
```

Pour plus d'informations sur les différentes fonctions du package et leurs signatures : `https://rdrr.io/cran/abc/man/`

3.5.2 Cross Validation

Avant de passer à l'étape de l'inférence, nous évaluons d'abord si l'ABC est capable d'estimer le paramètre μ. Nous utilisons la fonction `cv4abc` pour déterminer la précision de l'ABC et la sensibilité des estimations au taux de tolérance. Le code suivant évalue la précision des estimations de mu sous trois taux de tolérance en utilisant la méthode de rejet et de régression linéaire :

```
cv.res.rej <- cv4abc(param,sumstat,nval=10,tols=c(.005,.01,.05),
statistic ="mean" ,method = "rejection")
cv.res.reg <- cv4abc(param,sumstat,nval=10,tols=c(.005,.01,.05),
statistic = "mean" ,method = "loclinear")
```

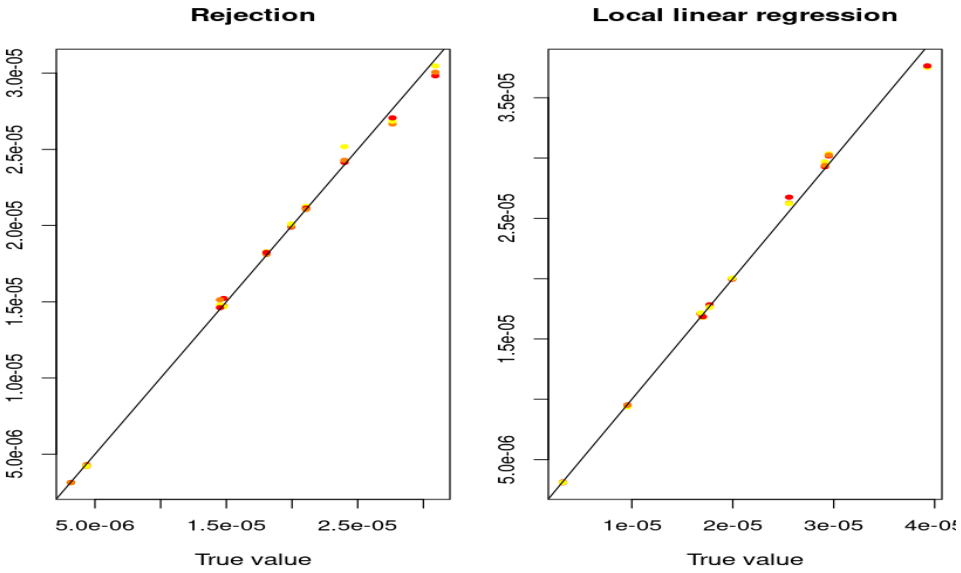


FIGURE 4 – Cross validation pour l'inférence du paramètre μ avec deux différentes méthodes : Rejection et Local linear regression. Les couleurs correspondent aux différents niveaux de taux de tolérance dans un ordre croissant du rouge au jaune.

les graphiques montrent les moyennes de la distribution a posteriori de μ pour chaque échantillon de cross validation. Les points du graphique de cross validation sont dispersés autour de la ligne d'identité, ce qui indique que peut être bien estimé à l'aide des notre statistiques sommaires (le mean dans notre cas). En outre, les estimations n'étaient pas seulement précises pour μ, mais aussi insensibles au taux de tolérance. En conséquence, l'erreur de prédiction est relativement faible et indépendante du taux de tolérance.

3.5.3 Inférence de μ en utilisant le package ABC

On utilise la fonction `abc`, avec la méthode loclinear :

```
abc_out <- abc(target = mean(reference_data[reference_data<filter]),
param = param, sumstat = sumstat, method = "loclinear", tol = tol)
```

Les résultats obtenus sont résumés dans les graphiques suivants :

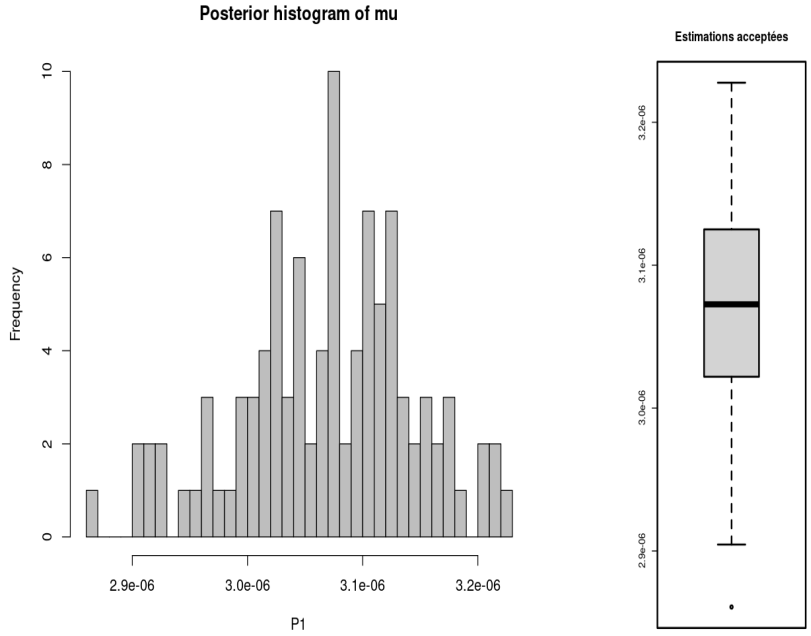


FIGURE 5 – Un histogramme représentant la distribution à posteriori de μ et un box-plot des estimations de μ acceptées par l'algorithme ABC.

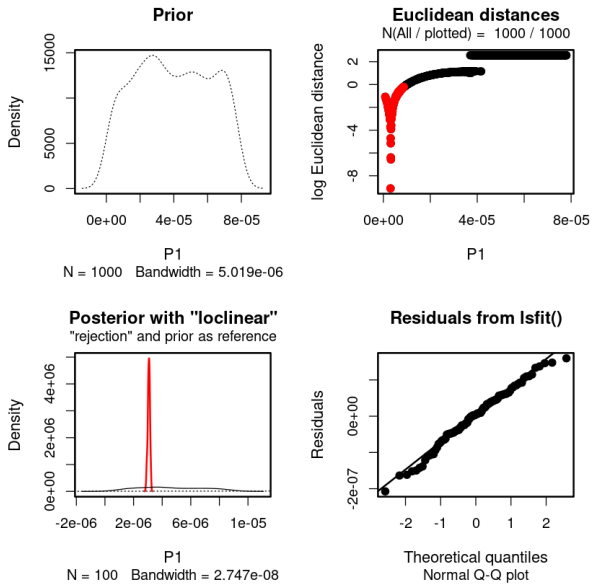


FIGURE 6 – **Diagnostics d'inférence et de régression des paramètres** : les tracés montrent (dans le sens des aiguilles d'une montre) la distribution a priori, les distances entre les statistiques récapitulatives observées et simulées en fonction des valeurs des paramètres (où les points rouges indiquent les valeurs acceptées), le tracé Q–Q normal des résidus des régression, et la distribution a posteriori obtenue (et la distribution a priori, pour référence).

Le box-plot montre que la moitié des estimations acceptées du paramètre μ était entre $3.02.10^{-06}$ et $3.12.10^{-06}$ avec un médian égal à peu près à $3.07.10^{-06}$.En plus, On remarque qu'il n'y a pas une déviation remarquable de la ligne médiane du centre du box-plot par rapport à sa longueur, ce qui montre la symétrie des estimations acceptées.

Les autres figures montrent que la distribution a posteriori est très différente de la distribution a priori, confirmant que la statistique sommaire fournit bien des informations sur le taux de mutation (Figure inférieure gauchee). La figure supérieure droite montre la distance entre les statistiques sommaires simulées et observées en fonction des valeurs à prior.Encore une fois, ce tracé confirme que le statistique sommaire transmet des informations sur μ, car les distances correspondant aux valeurs acceptées sont regroupées et non réparties autour de la plage précédente de μ . La figure inférieure droit affiche un graphique Q-Q standard des résidus de la régression. Ce graphique sert de diagnostic de régression linéaire ou non linéaire lorsque la méthode est "loclinéaire". On remarque que les points tombent sur la ligne de référence à 45 degrés ce qui montrent que les données sont distribuées normalement (symmetrically distributed with no skew).

L'espérance conditionnelle $E[\theta|y]$ est le prédicteur optimal du paramètre μ au sens des moindres carrés : `mean(abc_out$adj.values)` $= 3.028377.10^{-06}$. On retrouve donc une valeur qui est très proche du vrai valeur.

3.5.4 Limitations en 2 dimensions

En passant en 2 dimensions, c'est-à-dire l'inférence de deux paramètres (μ et f), le problème devient de plus en plus difficile, car il faut bien choisir les bons statistiques qui résument bien notre distribution. Essayons par exemple de travailler avec le même jeu de donnée qu'avant (un jeu de données `referenced_data` généré par un $\mu = 3.10^{-06}$) et avec les deux statistiques suivantes : la moyenne et la variance, on obtient les figures suivantes pour la cross-validation avec la méthode local linear regression :

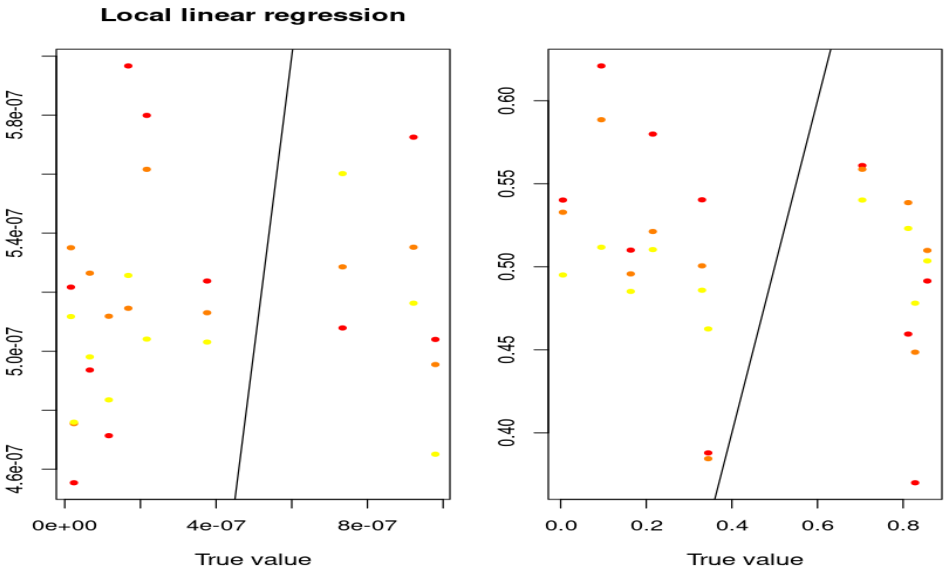


FIGURE 7 – Cross validation pour l'inférence du paramètre μ et f avec la méthode local linear regression. Les couleurs correspondent aux différents niveaux de taux de tolérance dans un ordre croissant du rouge au jaune.

Les points du graphique de cross validation ne sont pas dispersés autour de la ligne d'identité ni pour le paramètre μ(figure à gauche) ni pour le paramètre f(figure à droite), ce qui indique que les deux paramètres ne peuvent pas être bien estimés à l'aide des nos statistiques sommaires (la moyenne et la variance dans notre cas). En fait on avait essayé de faire plusieurs types de statistiques (moyenne, variance, quantiles, etc ...) mais on n'a pas obtenu des bons résultats. Le problème devient un peu plus difficile puisqu'il faut choisir les bons statistiques informatifs qui tendent vers l'exhaustivité(c'est-à-dire qui tend vers la propriété $p(\theta|S(y))=p(\theta|y)$ avec θ nos paramètres à estimer, y nos données observées et $S(y)$ la statistique sommaire).

4 Résultats

4.1 Comparaison des méthodes en une dimension

Soit le jeu de données générées à partir des paramètres suivants `N0 = 10, N1 = 1e-7, d0 = 0, mu = 3e-6, fmutant = 0.7 et p = 1` avec un nombre de réplicats de `384`. On applique nos cinq méthodes sur ce jeu de données `500` fois pour inférer le taux de mutation μ. On trace la densité des résultats correspondant et on en déduit la méilleur valeur de μ inférée par la moyenne. De surcroît, on estime l'erreur relative par rapport à la vrai valeur par la formule suivante `abs((mu_inférée - mu_vraie)/mu_vraie)`.

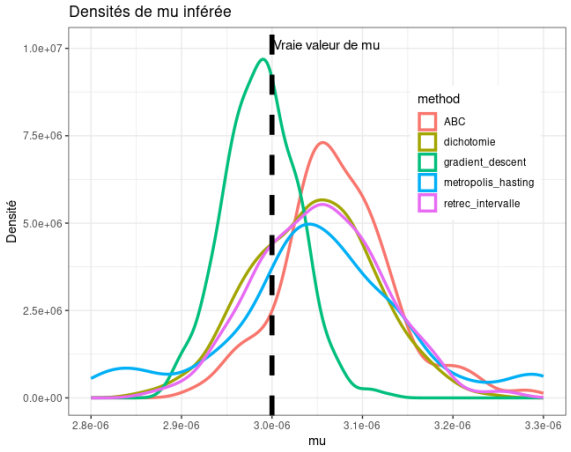


FIGURE 8 – Distribution de μ inférée par les différents cinq méthodes

Méthode	Moyenne	Erreur relative
Dichotomie	3.045101e-06	0.015033368
Retrec Intervalle	3.052895e-06	0.01763151
Gradient Descent	2.987599e-06	0.004133724
Métropolis Hasting	3.127877e-06	0.04262557
ABC	3.079647e-06	0.02654916

FIGURE 9 – Estimations de μ par la moyenne et les erreurs relatives

4.2 Comparaison en deux dimensions

Soit le jeu de données générée précédemment. On cherche maintenant à inférer (prédire) le taux de mutation μ ainsi que la fitness f avec nos deux méthodes : rétrécissement d'intervalle et gradient descent. Vu la lenteur des méthodes développés, on ne répète notre inférence que 50 fois pour obtenier les moyennes et les erreurs relatives suivants :

Rétrécissement d'intervalle : Moyenne μ : 3.484132e-06, Erreur relative μ : 0.161377, Moyenne f : 0.630957, Erreur relative f : 0.098932.

Gradient Descent : Moyenne μ : 3.399519e-06, Erreur relative μ : 0.133173, Moyenne f : 0.749894, Erreur relative f : 0.071277.

Dans le contexte de l'inférence en deux dimensions, il semble y avoir une précision inférieure par rapport aux résultats obtenus lorsqu'on utilise seulement une dimension. La raison de cette baisse de précision peut s'expliquer par le fait que les deux paramètres utilisés ne sont pas nécessairement indépendants l'un de l'autre. Cette dépendance peut être visualisée par une diagonale contenant des valeurs faibles dans la figure 3.

4.3 Précision en une dimension

Un violin plot est un type de graphique qui combine un diagramme en boîte (boîte à moustaches) avec un histogramme, permettant ainsi de visualiser la distribution des données. Le corps principal du graphique est formé par deux courbes symétriques, l'une pour chaque sens de l'axe des abscisses, représentant la densité de probabilité des données. Les boîtes, quant à elles, indiquent les quartiles de la distribution des données, avec la ligne médiane représentant la médiane de la distribution. Les points représentent les valeurs aberrantes qui se trouvent en dehors de l'intervalle interquartile.

Dans notre cas, les violon plot permettent de visualiser la distribution de l'erreur relative du taux de mutation qu'on cherche à inférer, dont l'expression a été donnée dans la section précédente, pour chaque méthode d'inférence sur 50 jeux de données différents. Ces jeux de données ont été crées en faisant variés principalement le taux de mutation en gardant les autres paramètres fixes à savoir `N0 = 10, N1 = 1e-7, d0 = 0, fmutant = 0.5 et p = 1` avec un nombre de réplicats de `384`.On peut ainsi comparer les performances des méthodes et évaluer leur précision et leur fiabilité.

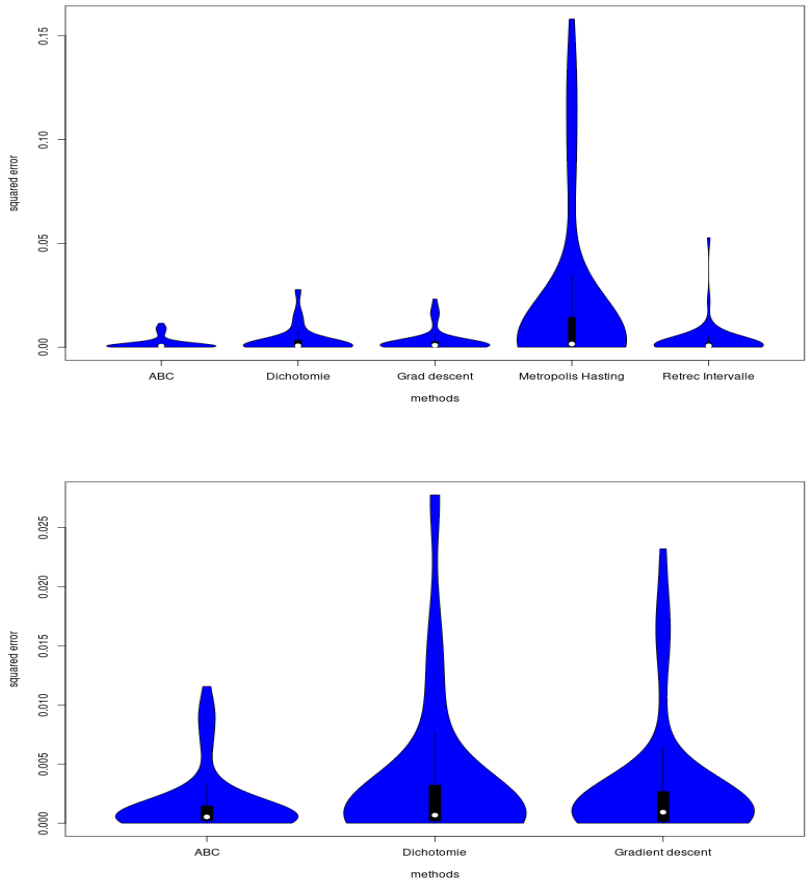


FIGURE 10 – Erreur relative des différentes méthodes

NB : On remarque que la méthode ABC a la meilleure précision et aussi que sa loi a la plus petite variance, c'est-à-dire que sa convergence est plus probable que les autres. La méthode de Metropolis Hastings a beaucoup de valeurs aberrantes parce qu'on a diminué son nombre d'itérations, pour accélerer son temps d'execution, mais cela a affecté considérablement sa précision. Les autres méthodes se situent au milieu.

4.4 Temps d'exécution

En reprenant les mêmes données que précédemment et en resuivant les mêmes étapes, on obtient les graphes suivants représentant le temps d'éxécution pour obtenir l'inférence du taux de mutation.

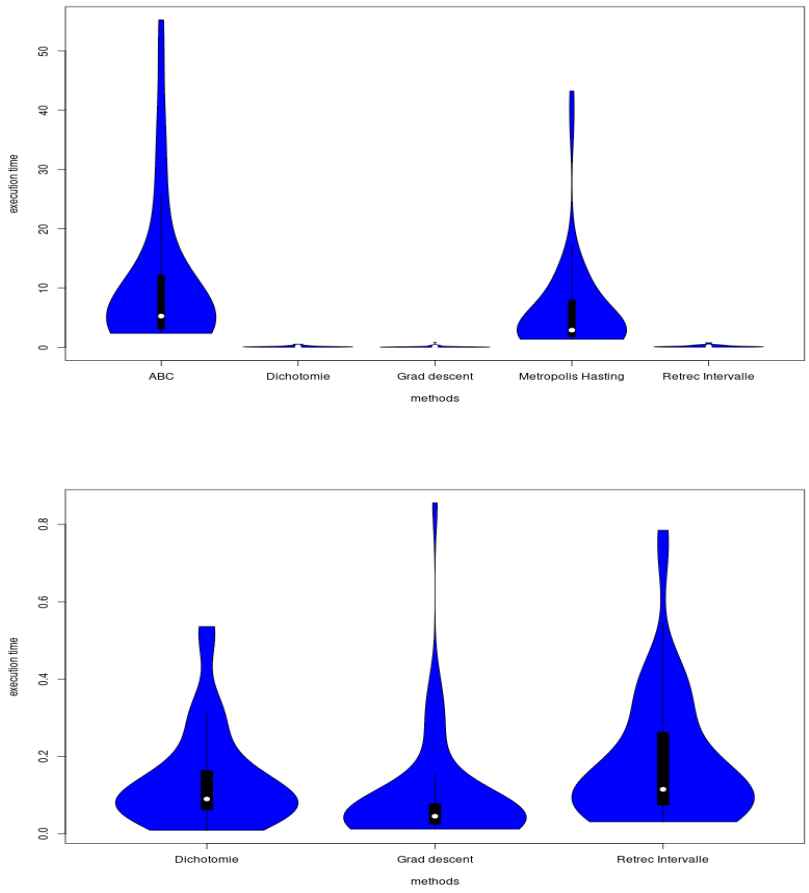


FIGURE 11 – Temps d'exécution en secondes

NB : On remarque que la méthode de gradient descendant est la plus rapide. La méthode ABC bien qu'elle soit très précise, elle est moins rapide que le gradient descendant. Les autres méthodes se situent encore une fois au milieu.

5 Conclusion

En conclusion, ce rapport a examiné le développement de plusieurs méthodes d'ABC (calcul bayésien approché) pour estimer le taux de mutation lorsque les autres paramètres sont connus à partir de données expérimentales, en s'appuyant sur un simulateur déjà développé. Les méthodes présentées incluent la dichotomie, le rétrécissement d'intervalle, le gradient descendant, le Metropolis Hastings, et l'utilisation du package ABC. Les résultats montrent que ces méthodes ont des avantages et des limitations en termes de précision et de temps d'exécution. Bien que la méthode ABC ait montré une précision élevée, elle peut être coûteuse en termes de temps de calcul, tandis que les autres méthodes présentent des temps d'exécution plus courts mais une précision inférieure, d'où la difficulté de trouver un compromis entre la complexité du modéle, la précision et le temps d'exécution. Le choix de la méthode dépend donc des priorités en matière de précision et de temps d'exécution. L'utilisation de plusieurs méthodes pour estimer les paramètres d'intérêt et la validation des résultats obtenus est recommandée. En fin de compte, les outils présentés dans ce rapport sont utiles pour l'inférence de paramètres dans les processus biologiques et peuvent aider les chercheurs à sélectionner la méthode la plus adaptée à leurs besoins pour mieux comprendre et modéliser les mécanismes de mutation dans des contextes écologiques complexes.