1 Introduction

Nous étudions l'article de recherche *An Adaptive Sequential Monte Carlo Method for Approximate Bayesian Computation*, écrit par P. Del Moral, A. Doucet et A. Jasra. Les idées développées dans cet article s'inscrivent dans le contexte de l'*Approximate Bayesian Computation* (ABC), une méthode permettant de réaliser de l'inférence dans le cas où la vraisemblance ne peut être calculée, que ce soit pour des raisons théoriques ou computationnelles. Pour ce faire, il est proposé une nouvelle approche combinant des méthodes de *Sequential Monte-Carlo* (SMC) aux méthodes ABC. L'une des principales contributions de ce papier est que l'algorithme présenté est de complexité linéaire (en la taille d'échantillon Monte-Carlo), là où dans la littérature, les algorithmes existants avaient une complexité quadratique.

2 Problème

2.1 Contexte

On se place dans le contexte du *Sequential Monte Carlo* (SMC) pour l'approximation bayésienne : étant donné un ensemble de paramètres Θ et un ensemble d'observations $\mathcal{D}$, on cherche pour $\theta \in \Theta$, $y \in \mathcal{D}$ à calculer le posterior $\pi(\theta|y)$ à partir du prior $\pi(\theta)$.

Dans le cas où il est irréalisable de calculer le terme de vraisemblance $f(y|\theta)$, mais que l'on sait générer des "pseudo-observations" x à partir de $f(.|\theta)$, on peut utiliser une approximation de la relation de Bayes pour approcher le posterior :

$$\pi_\epsilon(\theta, x|y) = \frac{\pi(\theta) f(x|\theta) 1_{A_{\epsilon,y}}(x)}{\int_{x' \in A_{\epsilon,y}, \theta' \in \Theta} \pi(\theta') f(x'|\theta') dx' d\theta'}$$

où $A_{\epsilon,y} = \{z \in \mathcal{D} : \rho(\eta(z), \eta(y)) < \epsilon\} : \epsilon > 0$ correspond à un seuil de tolérance au-delà duquel on rejette les pseudo-observations, où $\eta : \mathcal{D} \mapsto \mathcal{S}$ est une statistique et $\rho : \mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}^+$ une distance.

2.2 La méthode ABC

Dans cette partie, nous allons expliquer les idées principales motivant les algorithmes type ABC.
En notant θ les paramètres du modèle, notons qu'il n'existe pas nécessairement de formule explicite pour calculer $\pi(y|\theta)$, et quand bien même une telle formule existerait, elle s'avère parfois algorithmiquement inexploitable. Nous supposons que savons simuler des données à partir de cette loi pour toute valeur de θ. Ainsi, en notant y_{obs} les données observées, nous pouvons définir une approximation de la vraisemblance par la formule suivante :

$$p(\theta|S(y_{obs})) = \int \pi(y|\theta) K(\{S(y) - S(y_{obs})\}/h) dy$$

avec $S(y)$ une statistique sur y, K un noyau et $h > 0$.

À partir de cette vraisemblance approximée, nous pouvons définir l'ABC-posterior par $\pi_{ABC}(\theta|S(y_{obs}))$. Cette quantité est une approximation de la posterior théorique de θ, qui pourra être utilisé pour réaliser de l'inférence

statistique. Le résultat de cette loi dépend donc du noyau choisi K, de la statistique S et du choix de la constante h.
L'intérêt de l'ABC est que l'on peut appliquer des méthodes Monte Carlo pour approximer l'ABC-posterior sous réserve de pouvoir simuler $\pi(y|\theta)$. En outre, plusieurs algorithmes permettant d'approximer l'ABC-posterior, comme l'importance sampling ou le MCMC. Un algorithme MCMC est présenté ci-dessous à titre informatif.

Entrée : Données y_{obs}, une fonction $S(\cdot)$, un noyau $K(\cdot)$ tel que max K(x)=1, un noyau de transition $g(\cdot|\cdot)$, un réel $h > 0$ et un entier $N > 0$.

Initialisation : On définit $s_{obs} = S(y_{obs})$ et on choisit/simule θ_0, s_0

Itérer : Pour i=1,...,N.

- Simuler θ à partir de $g(\theta|\theta_{i-1})$
- Simuler y_{sim} à partir de $\pi(y|\theta)$, puis calculer s=$S(y_{sim})$
- Avec une probabilité
$$\min\{1, \frac{K((s - s_{obs})/h)}{K((s_{i-1} - s_{obs})/h)} \frac{\pi(\theta)g(\theta_{i-1}|\theta)}{\pi(\theta_{i-1})g(\theta|\theta_{i-1})}\}$$
accepter θ et fixer $\theta_i = \theta$, $s_i = s$, sinon fixer $\theta_i = \theta_{i-1}$ et $s_i = s_{i-1}$

Sortie : Un ensemble de paramètres $\{\theta_i\}_{i=1}^N$

---

3 La nouvelle approche SMC-ABC

Comme son nom le laisse deviner, cette nouvelle approche est une fusion entre le SMC et l'ABC.

- L'objectif de l'échantillonnage SMC est d'approximer une suite de probabilités $\{\pi_n\}_{0 \leq n \leq T}$ par un ensemble de N variables aléatoires $\{Z_n^{(i)}\}_{i=1}^N$. Au temps $n = 0$, la distribution π_0 est choisie de sorte qu'elle soit facile à approximer, puis la particule $\{Z_{n-1}^{(i)}\}_{i=1}^N$ est déplacée en utilisant un noyau markovien $K_n(z_{n-1}, z_n)$.

- Dans le contexte de l'ABC, nous voulons échantillonner à partir de la suite de distributions $\{\pi_{\epsilon_n}(\theta|y)\}$ telle que $\epsilon_0 > ... > \epsilon_T$, où $\pi_{\epsilon_n}(\theta|y)$ est la marginale de $\pi_{\epsilon_n}(\theta, x|y)$.
Dans cet algorithme, les niveaux de tolérance ϵ_n ne sont pas fixés, mais calculés à partir des poids $\{W_n^{(i)}\}$. Pour ce faire, les auteurs définissent la proportion de particules vivantes $PA(\{W_n^{(i)}\}, \epsilon_n) = \frac{\sum_{i=1}^N 1_{(0,+\infty)}(W_n^{(i)})}{N}$. Puis, ϵ_n est calculé à partir de l'égalité suivante :

$$PA(\{W_n^{(i)}\}, \epsilon_n) = \alpha PA(\{W_{n-1}^{(i)}\}, \epsilon_{n-1})$$

L'algorithme proposé par l'article est le suivant :

Etape 0 - Initialisation : A $n = 0$, on pose $\epsilon_0 = \infty$, et on pose comme poids initiaux $W_0^{(i)} = \frac{1}{N}$. Ainsi, nous avons $PA(\{W_0^{(i)}\}, \epsilon_0) = \frac{1}{N}$. Pour $1 \leq i \leq N$, on échantillonne $\theta_0^{(i)}$ selon la prior $\pi(\cdot)$, puis $X_0^{(i)}$ selon la vraisemblance $f(\cdot|\theta_0^{(i)})$.

Récurrence : On répète les étapes suivantes :

· **Étape 1 - Calcul du seuil :** Si $\epsilon_n \leq \epsilon$ ou $n = T + 1$, on s'arrête. Sinon, on incrémente n, et on calcule ϵ_n en résolvant

$$PA(\{W_n^{(i)}\}, \epsilon_n) = \alpha PA(\{W_{n-1}^{(i)}\}, \epsilon_{n-1})$$

où les nouveaux poids à cette étape n sont donnés par

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{1_{A_{\epsilon_n, y}}(X_{n-1}^{(i)})}{1_{A_{\epsilon_{n-1}, y}}(X_{n-1}^{(i)})}$$

· **Étape 2 - Resampling :** Si $PA(\{W_n^{(i)}\}, \epsilon_n) < \frac{1}{2}$ on rééchantillonne les particules, et on impose des poids uniformes $W_n^{(i)} = \frac{1}{N}$.

· **Étape 3 - Sampling :**
Pour $i \in \{1, ..., N\}$, si $W_n^{(i)} > 0$ on échantillonne $(\theta_n^{(i)}, X_n^{(i)}) \sim K_n((\theta_{n-1}^{(i)}, X_{n-1}^{(i)}), \cdot)$, puis on retourne à l'étape du calcul du seuil.

3.1 Contexte d'application

Nous avons appliqué cet algorithme à un cas concret d'épidémiologie. Nous considérons le modèle classique de naissance-mort-mutation (Tanaka et al. 2006) permettant de simuler la dynamique de l'évolution du nombre de maladies infectieuses au sein d'une population. On note φ, τ, ξ le taux de naissance, de mort et de mutation. Le nombre de porteur du génotype i à l'instant n est $X_n^{(i)}$ et le nombre de génotypes distincts est G_n.

Les données observées proviennent d'un épisode tuberculeux à San Francisco. Ils sont constitués de 326 génotypes différents pour $n = 473$ personnes. Les données peuvent-être résumées comme suit :

$$30^1\ 23^1\ 15^1\ 10^1\ 8^1\ 5^2\ 4^4\ 3^{13}\ 2^{20}\ 1^{282}$$

où m^k signifie qu'il y a k clusters de taille m.

Avant de commencer l'algorithme, il est nécessaire de préciser la définition pour les termes suivants.
- $A_{\epsilon_n, y} = \{z \in D : \rho(\eta(z), \eta(y)) < \epsilon\}$,
- η est une statistique définie par $\eta = (\eta_1, \eta_2) = (g, 1 - \sum_{i=1}^{g}(n_i/n)^2)$ avec g le nombre de clusters distincts, n_i le nombre d'individus du génotype i, n le nombre d'individus total.
- ρ est une métrique de distance tel que $\rho(\eta, \overline{\eta}) = \frac{1}{n}|\eta_1 - \overline{\eta}_1| + |\eta_2 - \overline{\eta}_2|$.

3.2 Initialisation

La première étape de l'algorithme est l'initialisation $n = 0$. On initialise les priors de nos paramètres comme suit : $\varphi \sim \mathcal{G}a(1, 0.1), \xi \sim \mathcal{TN}(0, 198, 0.06735^2), \tau \sim 1_{[0,\varphi)}$.

On fixe les poids $W_0^{(i)} = N$ et on calcule $PA(\{W_0^{(i)}\}, \epsilon_0) = \frac{\sum_{i=1}^{N} 1_{(0,+\infty)}(W_n^{(i)})}{N} = 1$. L'algorithme décrit dans le papier utilise la notion d'*Effective Sample Size* définie par $ESS(\{W_n^{(i)}\}) = (\sum_{i=1}^{N}(W_n^{(i)})^2)^{-1}$, qui est une mesure de la qualité de l'estimateur. Nous avons pris l'initiative de remplacer l'ESS par le PA, car comme suggéré dans le papier, l'ESS est une fonction croissante en ϵ_n contrairement au PA ce qui peut poser des problèmes de convergence.
De plus, l'article propose de choisir le niveau de tolérance ϵ_n en résolvant $ESS(\{W_n^{(i)}\}, \epsilon_n) = \alpha ESS(\{W_{n-1}^{(i)}\}, \epsilon_{n-1})$.

Cependant, pour des valeurs de α proche de 1, le SMC converge lentement, mais aboutit à une bonne approximation finale, tandis que pour α proche de 0 la convergence est rapide, mais peu fiable.

Enfin, comme indiqué dans le papier, on initialise nos populations avec un génotype et un individu de ce type, autrement dit $X_0^{(1)} = 1, G_0 = 1$

3.3 Calcul du seuil

En entrant dans la boucle, on calcule ϵ_n en résolvant $PA(\{W_n^{(i)}\}, \epsilon_n) = \alpha PA(\{W_{n-1}^{(i)}\}, \epsilon_{n-1})$. Pour ce faire, nous calculons une liste des distances entre nos données simulées X et nos données observées Y, puis nous déterminons ϵ_n en sélectionnant le α-quantile de la liste des distances.
Comme dans le papier, nous avons choisi la valeur $\alpha = 0.9$ pour notre implémentation. Ensuite, étant donné cette nouvelle valeur de ϵ_n, nous déterminons les nouveaux poids comme décrit dans l'algorithme ci-dessus.

3.4 Resampling

Dans le papier étudié, la condition pour réaliser le réechantillonnage est $ESS(W_n^{(i)}) < N_T$ avec N_T fixé. Cependant, puisque nous avons décidé de travailler avec la PA plutôt que l'ESS, nous devons adapter cette condition. Expérimentalement, nous avons donc remplacé cette condition par $PA(\{W_n^{(i)}\}, \epsilon_n) < N_T/N = 1/2$, autrement dit, cela signifie que l'on rééchantillonne nos populations lorsque la moitié des particules ont été tuées.

Le rééchantillonage des N particules se fait en remplaçant chaque particule dans X_n par une des particules qui a survécu au rétrécissement de ϵ_n, une particule pouvant ainsi être représentée plusieurs fois dans X_n.

$$\hat{\pi}_{\epsilon_n}(d(\theta, x)|y) = \sum_{i=1}^{N} W_n^{(i)} \delta_{(\theta_{n-1}^{(i)}, X_{n-1}^{(i)})}(d(\theta, x))$$

Cela revient à relancer des simulations uniquement à partir des particules suivant une trajectoire se rapprochant de l'échantillon de données.

3.5 Sampling

La dernière étape est le tirage des paramètres $\theta_n = (\varphi_n, \xi_n, \tau_n)$ ainsi que des X_n. La méthode implémentée, qui est aussi celle proposée par le papier, est un algorithme de Metropolis-Hasting. On tire les paramètres (φ, τ, ξ) suivant des lois normales

$$\varphi_n \sim \mathcal{N}(\varphi_{n-1}, 0.05), \tau_n \sim \mathcal{N}(\tau_{n-1}, 0.05), \xi_n \sim \mathcal{N}(\xi_{n-1}, 0.7)$$

Sous condition que $\varphi > 0, \tau > 0, \xi > 0$ et $\tau < \varphi$.
Ensuite, on met à jour nos populations de la façon suivante :

- On tire un événement parmi la naissance, la mort et la mutation avec probabilité $(\varphi/(\varphi+\tau+\xi), \tau/(\varphi+\tau+\xi), \xi/(\varphi+\tau+\xi)$
- On tire un génotype avec comme probabilité de tirer le génotype i : $X_{n-1}^{(i)} / \sum_{j=1}^{G_{n-1}} X_{n-1}^{(j)}$
- Si l'on tire le génotype i :
  - et l'événement est une naissance, alors le génotype i choisi évolue comme suit : $X_n^{(i)} = X_{n-1}^{(i)} + 1$
  - et l'événement est une mort alors le génotype i choisi évolue comme suit : $X_n^{(i)} = X_{n-1}^{(i)} - 1$
  - et l'événement est une mutation alors le génotype i choisi évolue comme suit : $X_n^{(i)} = X_{n-1}^{(i)} - 1$ et un nouveau génotype contenant 1 seul individu est créé.

On accepte cette simulation avec une probabilité d'acceptation

$$\min\left\{1, 1_{\epsilon_n,Y}(X_n) \times \frac{f_{\varphi_n}(\varphi_n) \times f_{\xi_n}(\xi_n) \times f_{\tau_n}(\tau_n)}{f_{\varphi_{n-1}}(\varphi_{n-1}) \times f_{\xi_{n-1}}(\xi_{n-1}) \times f_{\tau_{n-1}}(\tau_{n-1})}\right\}$$

La loi d'évolution de notre population naissance-mort-mutation décrit ci-dessous correspond à l'évolution de X par un noyau markovien.

4 Implémentation numérique

Nous avons implémenté la méthode décrite ci-dessus. Nous avons réalisé 10 simulations de 10 000 itérations chacune. Le temps de calcul pour une simulation est d'environ 10 heures. Pour chaque simulation, nous avons 1 000 populations (ou encore scénarios) différentes. À titre d'exemple, dans l'article, il est indiqué qu'une simulation comportait 50 000 itérations réalisé en approximativement 40 heures.
Au bout de 10 000 itérations, nous atteignons des valeurs $\epsilon \approx 10^{-1}$ et la courbe commence à s'aplanir, ce qui explique l'important temps de convergence.

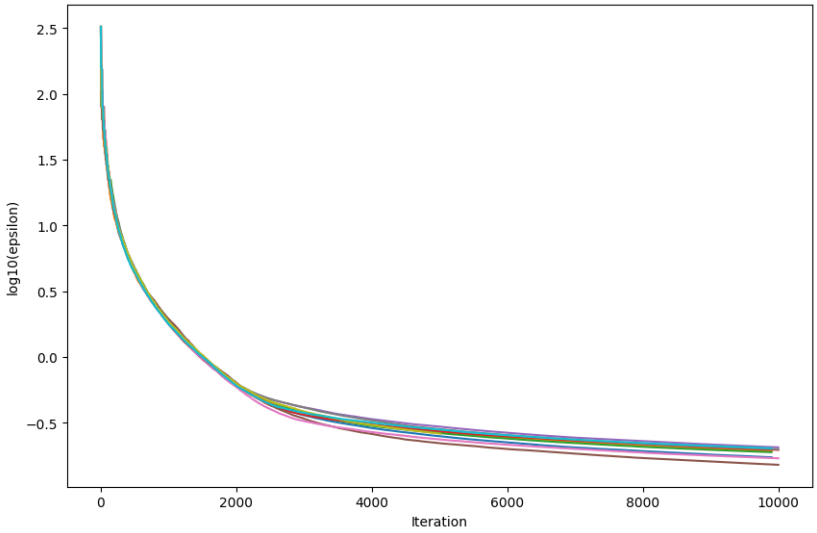

Figure 1: Seuil de tolérance ϵ en fonction du nombre d'itérations

- La FIGURE 2 ci-dessous représente l'histogramme du nombre d'individus présents dans chaque population pour chacune de nos 10 simulations.
On observe un pic en 0, ce qui correspond aux populations qui seraient rééchantillonnées si l'on avait réalisé une itération supplémentaire. Outre cette valeur particulière, l'histogramme a l'allure d'une loi normale asymétrique centré en 4000, excepté pour la simulation 9.

- La FIGURE 3 ci-dessous représente l'histogramme du nombre de génotypes distincts pour chaque population. La majorité des populations ont un faible nombre de génotypes distincts et il y a un nombre significatif de populations ayant un nombre de génotypes distincts importants. Ces résultats peuvent sembler assez éloignés des données **y** que nous souhaitons imiter (326 génotypes différents). Cela peut s'expliquer par le choix de la métrique $\rho(\eta, \bar{\eta}) = \frac{1}{n}|g - \bar{g}| + |\sum_{i=1}^{\bar{g}}(\bar{n}_i/\bar{n})^2 - \sum_{i=1}^{g}(n_i/n)^2|$. Comme on peut le voir en comparant l'histogramme de la FIGURE 2 et celui de la FIGURE 3, on peut remarquer que $g << n$. Cela entraîne que la métrique ne pénalise pas beaucoup le fait que le nombre de génotypes distincts soient différents.

- La FIGURE 4 ci-dessous représente l'histogramme de la proportion du génotype majoritaire au sein de la population, autrement dit $\frac{\text{Nombre d'individus du génotype majoritaire}}{\text{Nombre d'individus total}}$.
Deux pics principaux sont présents dans ces histogrammes. Le premier pic, à 0.5 correspond aux populations à 2 génotypes et 1 individu par génotype. Le second pic à 1 correspond aux populations à 1 génotype. Ce sont des populations qui sont très certainement vouées à disparaitre prochainement, puisque la maladie n'a pas pu se propager correctement.

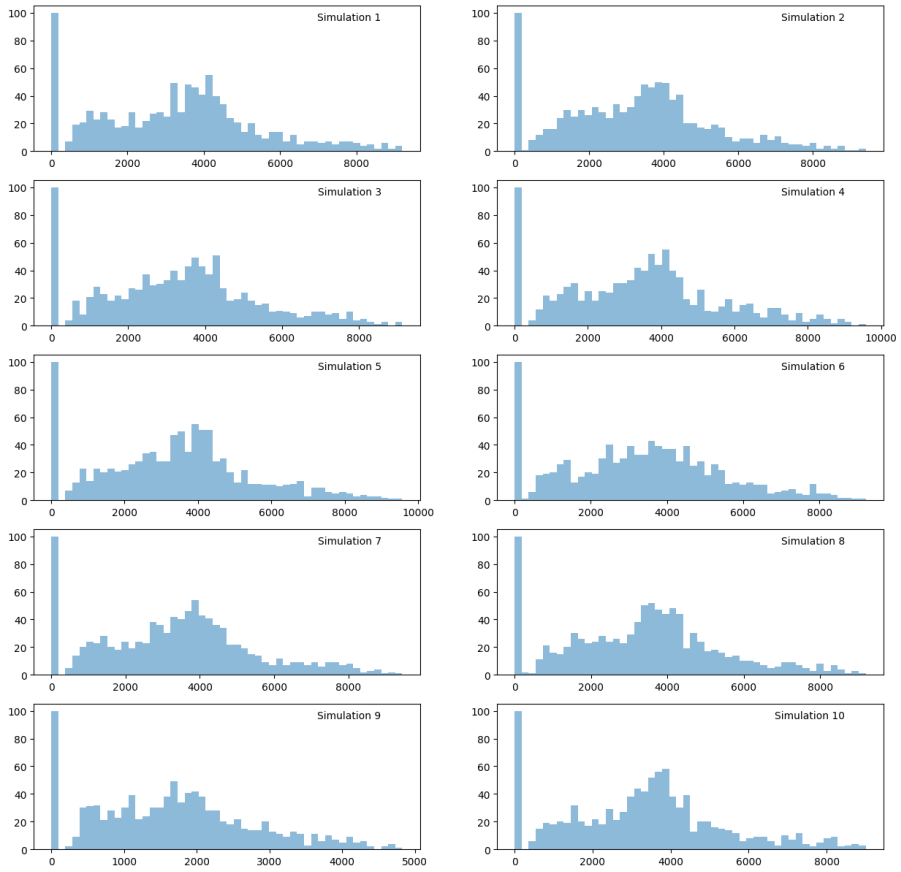

Figure 2: Histogramme du nombre d'individus par scénario

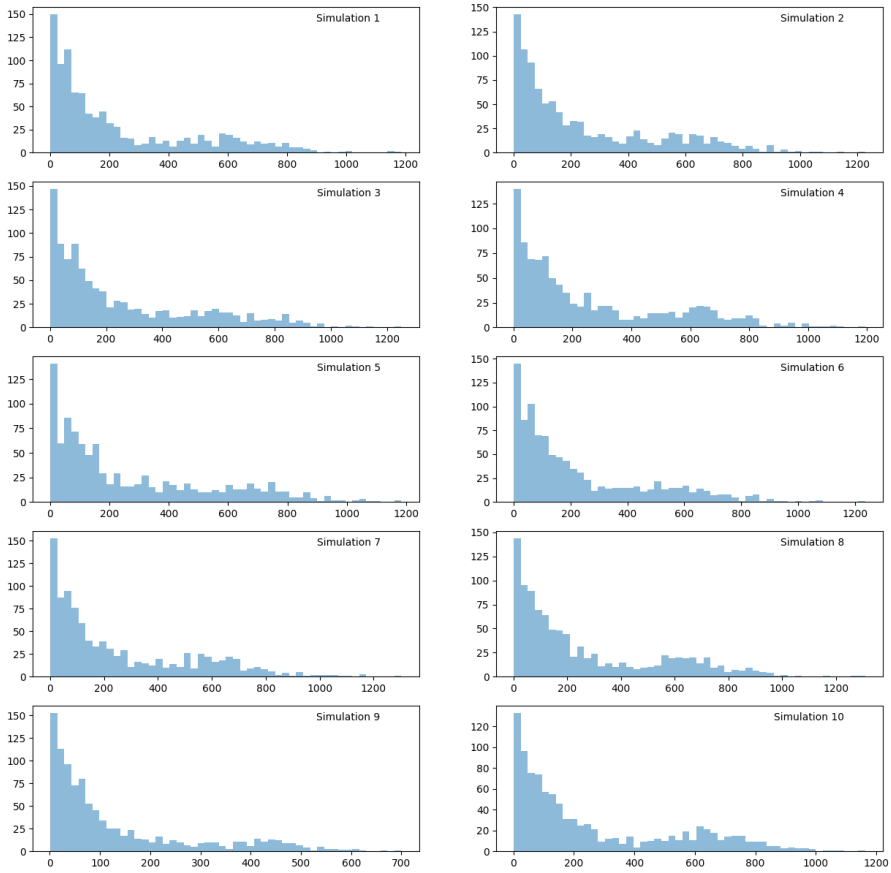

Figure 3: Histogramme du nombre de génotypes distincts

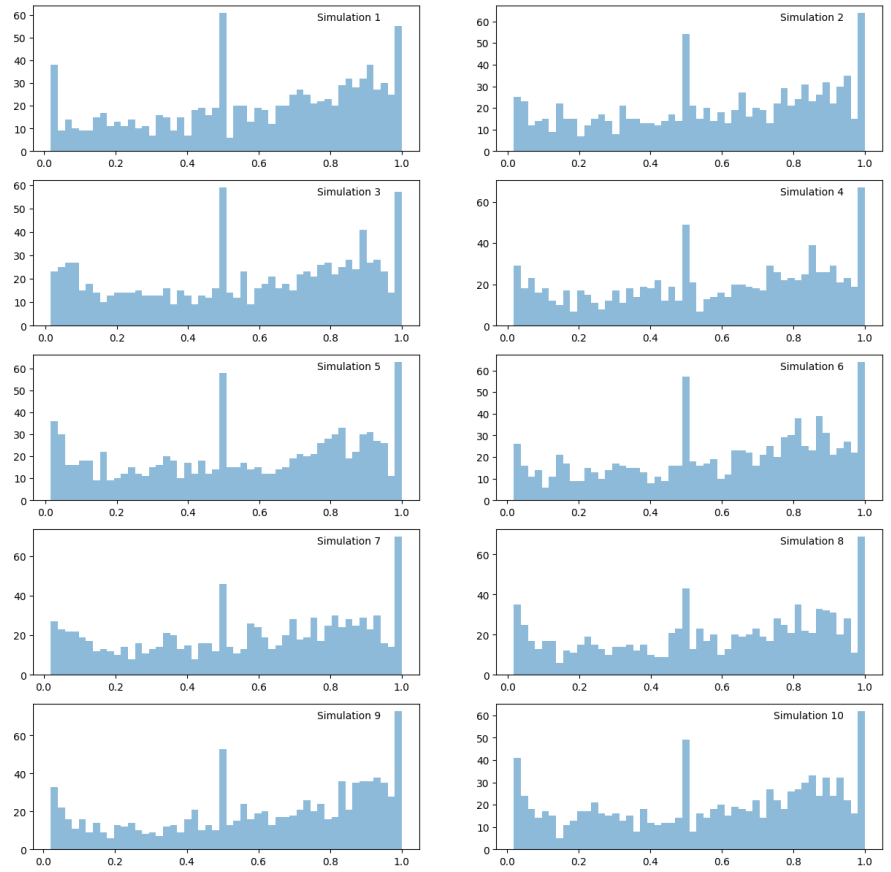

Figure 4: Histogramme de la proportion du génotype majoritaire

Par la suite, nous nous sommes intéressés aux paramètres φ, τ, ξ, correspondants aux taux de natalité, mortalité et de mutation. Les paramètres prennent des valeurs très différentes : en effet, le taux de natalité est élevé, ce qui est cohérent avec le grand nombre de génotypes que nous avons pu observer FIGURE 3. Le taux de mortalité est plus étalé, tandis que le taux de mutation est plutôt faible. Les histogrammes de chacun des trois paramètres pour chacune des simulations sont disponibles en Annexe.

Simulation	Moyenne			Ecart-type		
	φ	τ	ξ	φ	τ	ξ
Sim.1	10.88	5.39	0.26	10.31	6.72	0.04
Sim.2	10.53	5.37	0.26	10.70	7.10	0.04
Sim.3	9.80	4.82	0.26	9.65	5.93	0.04
Sim.4	9.92	4.77	0.26	9.86	6.07	0.04
Sim.5	9.53	4.77	0.26	9.90	6.23	0.04
Sim.6	10.16	5.15	0.26	9.95	6.34	0.04
Sim.7	9.81	4.93	0.26	9.73	6.04	0.04
Sim.8	9.86	4.94	0.26	9.84	6.08	0.04
Sim.9	10.44	5.25	0.26	10.56	6.73	0.04
Sim.10	9.74	4.82	0.26	9.48	5.79	0.04

Figure 5: Moyenne et écart-type des paramètres par simulation

Les écarts-types obtenus sont plutôt importants, excepté pour ξ. Cela peut s'expliquer par le fait que nous avons réalisé uniquement 10 000 itérations au lieu des 50 000 du papier pour des raisons de complexité temporelle. Par exemple, nous atteignons des valeurs de $\epsilon \approx 10^{-1}$ là où l'article atteint 10^{-4}. Puisque nous avons réduit le nombre d'itérations, nos simulations n'ont pas eu le temps de converger correctement, ce qui explique que certains de nos résultats peuvent paraitre surprenants au premier abord. Plus particulièrement, ici, nos paramètres ne sont pas encore assez stables, ce qui explique cette forte variance.
Cependant, si on calcule la moyenne des espérances des simulations, ainsi que la variance des moyennes, on obtient les résultats suivants

Variable	Moyenne	Ecart-type
φ	10.12	0.40
τ	5.02	0.25
ξ	0.26	0

Figure 6: Moyenne et écart-type des paramètres sur les 10 simulations

Les écarts-types entre les moyennes des 10 simulations sont plutôt faibles, ce qui signifie que bien que les résultats des méthodes de Monte-Carlo soient aléatoires, les résultats restent proches.

5 Conclusion

Cet article présente une nouvelle approche combinant les concepts SMC et ABC. L'objectif est de pallier AU problème de l'inférence dans le cas où la vraisemblance n'est pas calculable. Pour ce faire, la méthode proposée réalise des tirages tant que nos données échantillonnées sont à une distance ϵ des données étudiées. Le papier propose de choisir adaptativement les seuils de tolérance ϵ_n, puis de réaliser un tirage des données simulées et des paramètres en utilisant un algorithme de Metropolis-Hasting. Nous avons reproduit un exemple de l'article consistant en l'étude d'un modèle épidémiologique de natalité/mortalité/mutation. Nous avons obtenu une certaine stabilité de nos résultats entre les différentes simulations. Cependant, puisque l'algorithme est assez coûteux temporellement (10 000 itérations en environ 10h), nous n'avons pas pu atteindre des valeurs de ϵ suffisamment basses pour obtenir une convergence satisfaisante.

6 Annexe

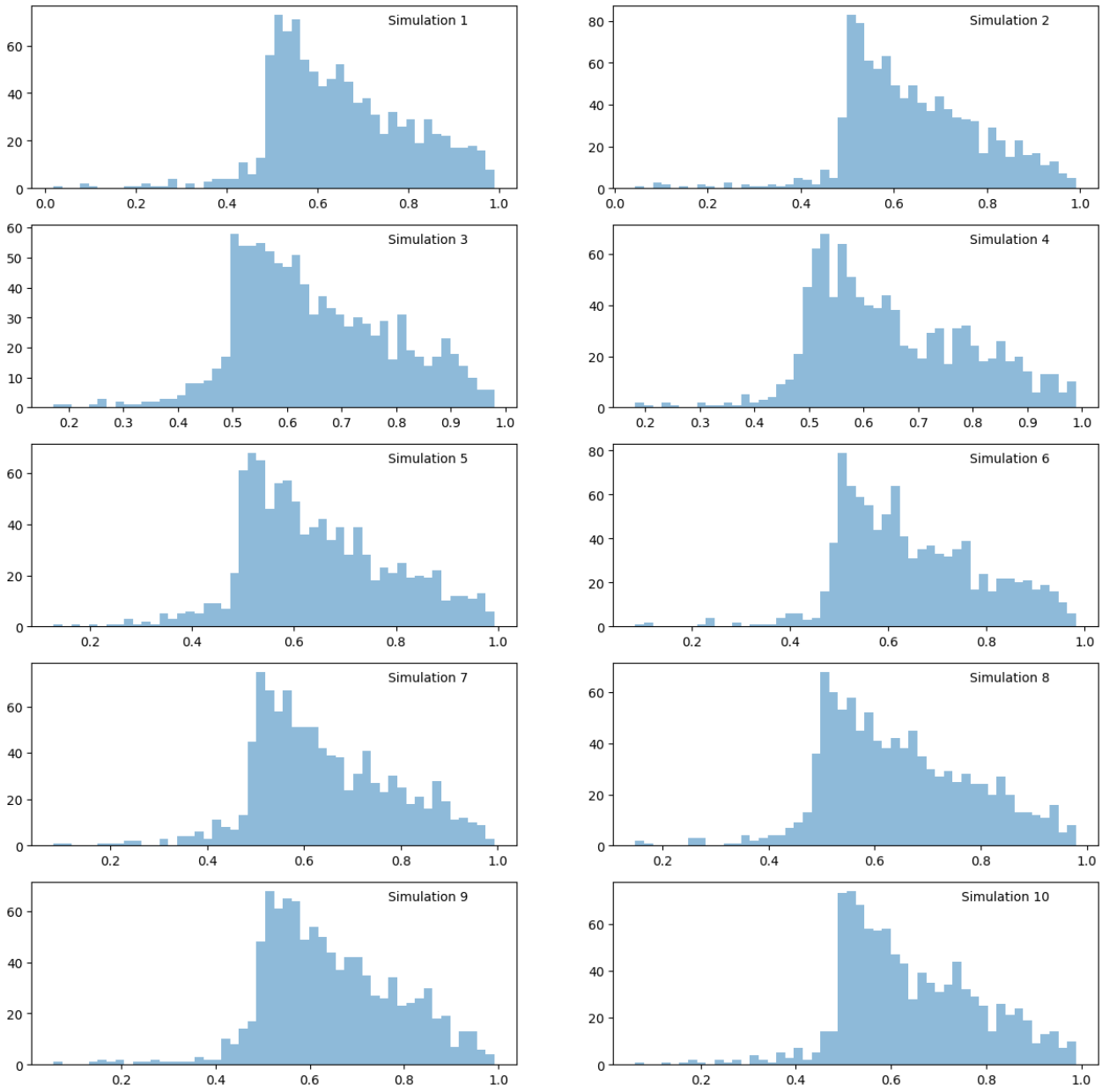

Figure 7: Histogramme du taux de natalité φ

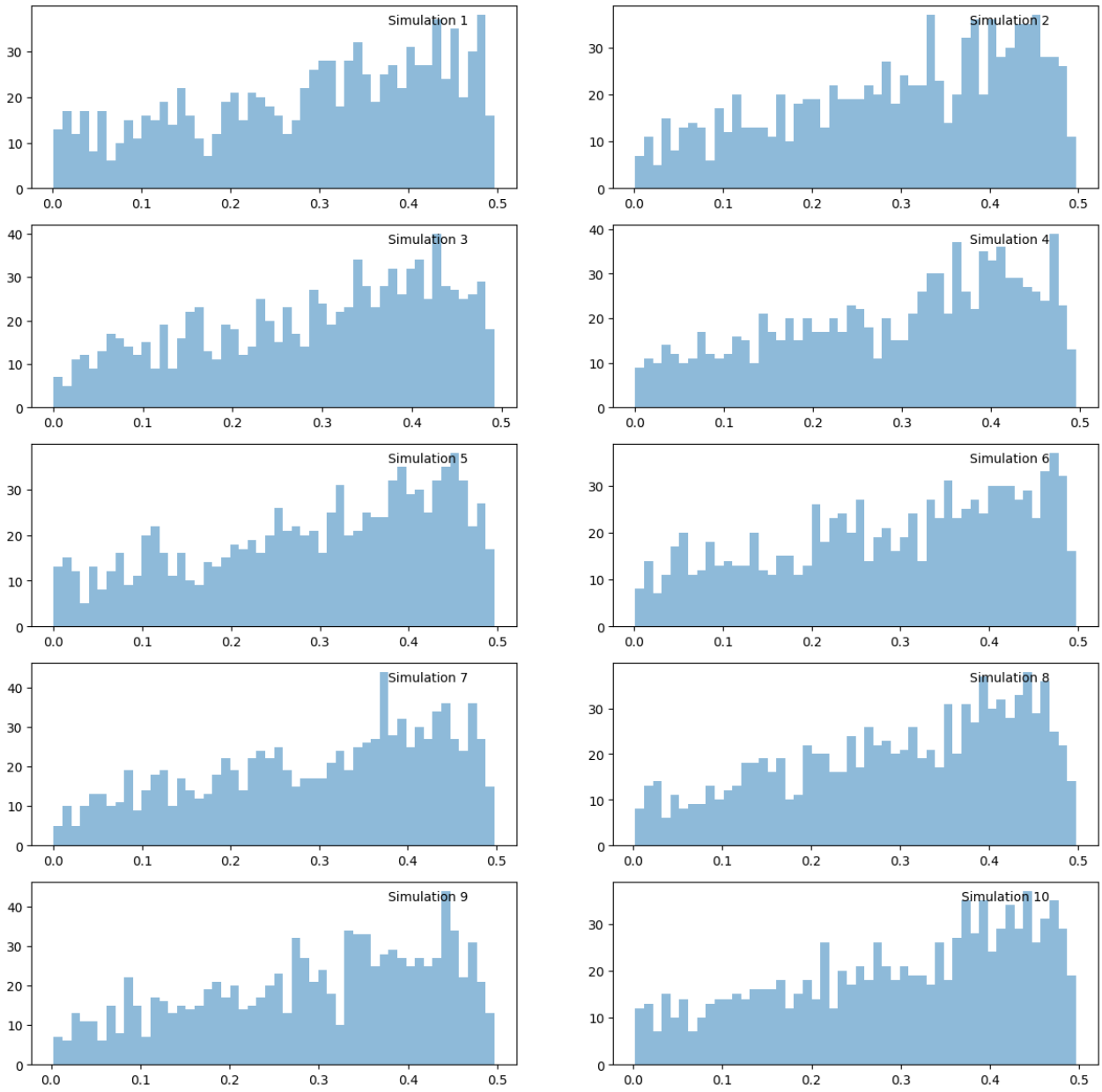

Figure 8: Histogramme du taux de mortalité τ

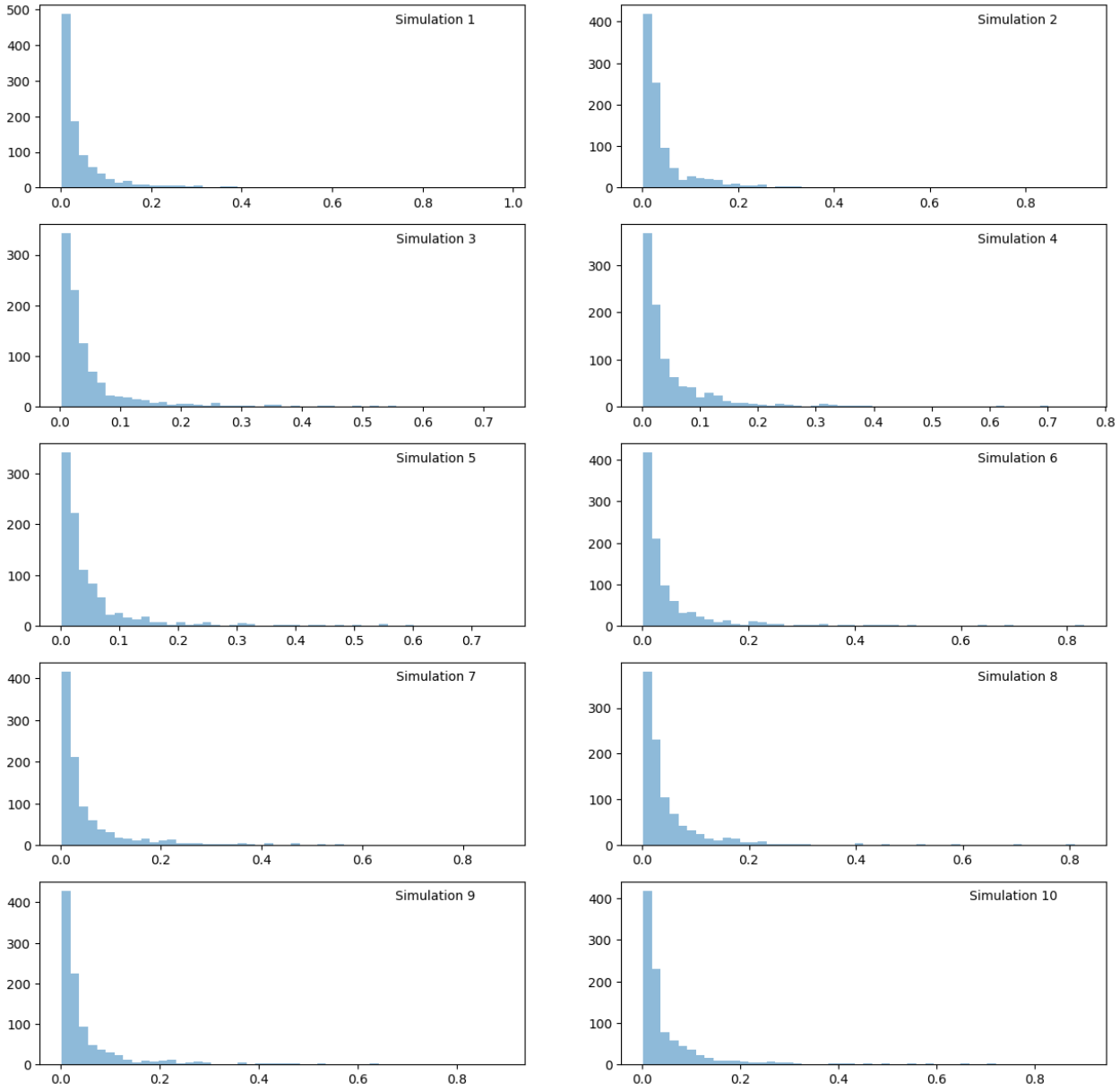

Figure 9: Histogramme du taux de mutation ξ