École nationale
de la statistique
et de l'analyse
de l'information

Étude de la distribution du nombre de transcrits dans des gouttelettes

Groupe :
Louis Allain
Léonard Gousset
Julien Heurtin

Encadrant :
Emmanuel Curis

Projet statistique de deuxième année

2023

Remerciements

Tout d'abord nous tenons à remerier notre tuteur, Emmanuel Curis pour l'attention qu'il aura porté sur notre étude, ses conseils et ses enseignements. Son expertise médicale aura été d'une grande aide sur l'ensemble du projet.

Nous remercions également Sébastien Da Veiga, enseignant-chercheur à l'ENSAI pour sa bienveillance mais aussi pour son aide, tant sur l'aspect mathématique que sur l'implémentation du code R.

Enfin, nous remercions Mathieu Marbac Lourdelle, enseignant-chercheur à l'ENSAI pour ses explications sur le modèle de mélange au début de notre projet.

Abstract

A transcriptome represents the set of RNAs (Ribonucleic Acid) present in a cell at a given time. They are responsible for protein production but also for other important functions such as variant detection, identification of alternative transcripts, etc. To study the transcriptome of a cell, individual cells can be isolated and analyzed separately using a technique called "single cell RNA-Seq". This technique involves dividing the cells in suspension into individual droplets which are then analyzed one by one. The RNA from each droplet is converted into DNA and amplified, then sequenced to identify the specific transcripts of each cell. However, not all droplets contain an individual cell and some may contain cellular debris or even two different cells, which can skew the results. Therefore, before analyzing the results, it is necessary to filter the droplets to keep only those containing a single, intact cell. This filtering is typically done using arbitrary thresholds based on the percentage of transcripts from mitochondrial genes and the total number of transcripts detected.

This project aims to use a mixture model to select the droplets to analyze during data collection. The mixture model assumes that the total number of transcripts per droplet follows a distribution that is the result of different sources : the number of transcripts in empty droplets, the number of transcripts in "type" cells, and the number of transcripts in droplets containing two cells. These sources can be modeled by different probability distributions such as the Poisson or normal distributions.

Now that we have built the mixture model, it should be possible to distinguish different populations of droplets in terms of their RNA content. However, the lack of reference data to evaluate the model's performance on real data is a major drawback for analyzing the results. Despite this lack of results, we created samples following the hypothesis to test our model. The results on the samples are very encouraging for the success of our model, indeed, as we have shown in our project, the sample data are correctly classified. Thus, the model should be able to distinguish droplets containing a single cell from those containing more than one cell, as well as droplets containing cellular debris. Nevertheless, this would assume that the hypothesized laws represent the reality of our data.

However, we have seen by analyzing the results that our model does not allow to classify the droplets properly. The problem is therefore that the laws do not represent the data well enough. We can consider that the model is too simplistic to create our three groups. We think that it might be interesting to represent the cell debris by two groups. We would keep the Poisson distribution but adding another normal distribution to represent the hump present after the Poisson distribution. Moreover, it is possible to explore other approaches to define relevant thresholds for droplet selection. These approaches can be based on biological or physiological criteria to select appropriate droplets.

In conclusion, our mixing model is a promising method to select droplets containing a single cell. However, it is still too far from reality to be used.

Résumé

L'objectif de cette étude est d'évaluer la régénération des populations de cellules après une greffe, chez les patients atteints de leucémie. Pour ce faire, on va analyser le transcriptôme de chaques nouvelles cellules souches du patient. Afin d'éviter les coûts de recherche pour vérifier si chaque gouttelette contient bien une cellule unique, nous avons fait un modèle statistique afin d'identifier ces gouttelettes. Cette modélisation s'est faite par une loi de mélange en fonction des hypothèses que l'on nous a données et de nos observations personnelles.

L'analyse exploratoire nous a permis de visualiser les hypothèses selon lesquelles il y aurait 3 groupes. Le premier représente les gouttelettes ne contenant que des débris de cellule, on peut le modéliser par une loi de Poisson. Le deuxième groupe quant à lui représente les gouttelettes contenant une seule cellule, c'est le groupe qui nous intéresse et peut être modélisé par loi normale. Enfin le troisème groupe représente les gouttelettes contenant 2 cellules, ce dernier peut lui aussi être modélisé par une loi normale.

Une fois le modèle implémenté sur nos données, nous obtenons les probabilités d'inclusion de chaque groupe, ainsi que les paramètres de la loi de Poisson et des deux lois normales. Ces paramètres ne correspondent pas exactement aux données mais sont encourageants pour la suite de l'étude. Nous avons déjà plusieurs pistes pour l'amélioration du modèle, notamment par un changment de la loi de Poisson en une loi log-normale, la mise en place d'une nouvelle loi normale ou encore l'introduction de lois binomiales-négatives.

Table des matières

Table des figures

Liste des tableaux

Contexte

Décrite en 1847 par le pathologiste allemand Rudolf Vichrow, la leucémie est un cancer des cellules de la moelle osseuse, responsables de la production de cellules sanguines, d'où le nom de cancer du sang.

La leucémie correspond donc à une production anormale de cellules sanguines et entraine une perturbation du fonctionnement normal du sang. Les symptômes de la leucémie peuvent varier en fonction du type de leucémie et du stade de la maladie, mais ils peuvent inclure la fatigue, la pâleur, les infections fréquentes, les saignements anormaux et les douleurs osseuses. Le diagnostic de la leucémie est généralement confirmé par des tests sanguins, des examens de moelle osseuse et d'autres investigations médicales.[2]

Le traitement de la leucémie a considérablement évolué au fil des années grâce aux avancées médicales. Les approches courantes comprennent la chimiothérapie, la radiothérapie, la thérapie ciblée et la greffe de cellules souches. La greffe de cellules souches, également appelée greffe de moelle osseuse, peut être utilisée pour remplacer la moelle osseuse malade par une moelle osseuse saine provenant d'un donneur compatible.

L'évolution de la leucémie au fil des années a été marquée par des avancées significatives. Au cours des dernières décennies, les taux de survie des patients atteints de leucémie ont augmenté grâce à une meilleure compréhension de la biologie de la maladie, à la découverte de nouveaux médicaments et à des approches de traitement plus efficaces.

Cependant, malgré ces progrès, la leucémie reste une maladie grave qui peut avoir des conséquences dévastatrices. Selon les dernières statistiques, la leucémie est responsable d'un nombre important de décès dans le monde entier. Selon l'Organisation Mondiale de la Santé (OMS), on estime qu'environ 437 000 nouveaux cas de leucémie ont été diagnostiqués en 2020, et que plus de 309 000 personnes en sont décédées cette année-là. Ces chiffres mettent en évidence l'impact considérable de cette maladie sur la santé publique et soulignent la nécessité continue de recherches et de développements dans ce domaine.

Les facteurs de risque de la leucémie comprennent souvent des antécédents familiaux, une exposition à des radiations ou à des produits chimiques toxiques, des infections virales et d'autres facteurs environnementaux, bien que dans de nombreux cas, la cause exacte de la leucémie reste inconnue.

La leucémie se manifeste à cause de la modification du génome[10] [1] des cellules leucémiques, notamment par des mutations de leur ADN pendant la transformation de la cellule. Les fonctions codées dans le transcriptome diffèrent selon les besoins d'une cellule, par exemple un neurone n'a pas besoin des même fonctionnalités qu'une cellule sanguine. Une sélection de gènes pour chaque cellule est donc nécessaire, c'est ce qu'on appelle « l'expression génétique ». Une fois tous les gènes nécessaires sélectionnés, un intermédiaire, appelé ARNm, est produit à partir de chacun d'entre eux, ce processus est appelé « transcription ». On dit que l'ensemble des ARN messagers (aussi appelés « transcrits ») d'une cellule constitue le transcriptome de cette dernière.[9]

1. Ensemble du matériel génétique codé sous forme d'ADN

Introduction

Un transcriptome représente l'ensemble des ARN (Acide RiboNucléique) présents dans une cellule à un moment donné. Pour étudier le transcriptome d'une cellule, on peut isoler des cellules individuelles et analyser chaque cellule séparément en utilisant une technique appelée "single cell RNA-Seq"[1]. Cette technique consiste à diviser les cellules en suspension en gouttelettes individuelles qui sont ensuite analysées une par une. L'ARN de chaque gouttelette est converti en ADN et amplifié, puis séquencé pour identifier les transcrits spécifiques de chaque cellule.

Cependant, toutes les gouttelettes ne contiennent pas une cellule individuelle et certaines peuvent contenir des débris cellulaires ou même deux cellules différentes, ce qui peut fausser les résultats. Par conséquent, avant d'analyser les résultats, il est nécessaire de filtrer les gouttelettes pour ne garder que celles contenant une seule cellule en bon état. Ce filtrage se fait généralement en utilisant des seuils arbitraires basés sur le pourcentage de transcrits provenant de gènes mitochondriaux et le nombre total de transcrits détectés.

Ce projet vise ainsi à utiliser un modèle de mélange pour sélectionner les gouttelettes à analyser lors de la collecte de données. Le modèle de mélange suppose que le nombre total de transcrits par gouttelette suit une distribution qui est le résultat de différentes sources : le nombre de transcrits dans les gouttelettes vides, le nombre de transcrits dans les cellules "type", et le nombre de transcrits dans les gouttelettes contenant deux cellules. Ces sources peuvent être modélisées par des lois de probabilité différentes, comme la loi de Poisson ou la loi Normale.

Pour répondre à ce sujet, nous allons donc commencer par étudier les données à notre disposition et en faire une approche exploratoire. Cela nous permettra de poser un cadre plus clair pour la construction du modèle de mélange. Enfin, une fois le modèle de mélange construit, il nous sera possible de vérifier que les hypothèses du modèle sur les lois de probabilité permettent de décrire la réalité de nos données.

1 Présentation des données et approche exploratoire

1.1 Origine et structure des données

Pour traiter les patients contre la leucémie, une greffe de moelle est souvent indiquée, et pour cela, on recourt donc à des dons de moelle osseuse. Les données expérimentales récupérées proviennent ainsi d'un patient à qui on a effectué une greffe de moelle osseuse. On souhaite donc analyser la moelle osseuse du patient pour voir comment se régénèrent les populations de cellules.

De cette moelle osseuse, on vient en extraire des gouttelettes pour l'analyse de cellules individuelles. On analyse le transcriptome des gouttelettes et obtenons donc l'ensemble des ARN ("transcrits") pour chaque gouttelette. Nous avons décidé de supprimer les gouttelettes ne contenant aucun transcrit car sans ces derniers, nous sommes certains que la gouttelette ne contiendra aucune cellule ni débris de cellule.
Pour chaque gouttelette observée, les chercheurs ont enregistré le nombre total de transcrits et le pourcentage d'ARN mitochondrial. On se retrouve avec des données de comptage pour chaque goutellette. Dans la suite, nous allons supprimer le pourcentage d'ARN mitochondrial des données car non utile pour la construction de notre modèle.
Nous avons alors 33 538 transcrits présent sous la forme de code et leur nombre d'occurence pour chacunes des 327 395 gouttelettes. On retrouve alors un tableau de données sous la forme suivante :

	Transcrits n°1	Transcrits n°2	...	Transcrits n°33 538
Gouttelette n°1	1	0	...	4
Gouttelette n°2	0	0	...	5
...	...	...	...	...
Gouttelette n°327 395	12	58	...	0

Table 1 – Représentation de la table de nos données

L'objectif premier n'étant pas de déterminer quels sont les transcrits que l'on retrouve nécessairement dans une cellule, nous avons décidé d'aggréger toutes les variables de comptage en une seule : le nombre de gène présents dans une gouttelette. On se retrouve avec un tableau de données de la forme suivante :

	Somme des transcrits
Gouttelette n°1	5
Gouttelette n°2	24
...	...
Gouttelette n°327 395	5008

Table 2 – Représentation de la table de nos données comme support de travail

Au final, nous avons une seule variable, quantitative, à valeurs dans $\mathbb{N}$ qui représente le nombre de transcrits au sein d'une gouttelette.
On va alors observer la distribution des données. On retrouve les quantiles importants, le minimum-maximum et la moyenne dans le tableau suivant. La moyenne est suivi de son écart type, on précise que nous sommes dans $\mathbb{N}$ donc le nombre de transcrits est forcément positif.

	Somme des transcrits
Minimum	1
1er Quartile	1
Moyenne	169,32 $(\pm 1572, 48)$
Médiane	1
3ème Quartile	3
Maximum	72639

Table 3 – Distribution de nos transcrits

On observe grâce à ces résultats qu'il y au moins 75% des gouttelettes qui ont un nombre de transcrits inférieur ou égal à 3. Pour représenter nos données, nous sommes alors obligé de passer en échelle logarithmique au risque de ne rien pouvoir observer sinon.

On peut alors représenter nos données graphiquement. On obtient le graphique suivant :

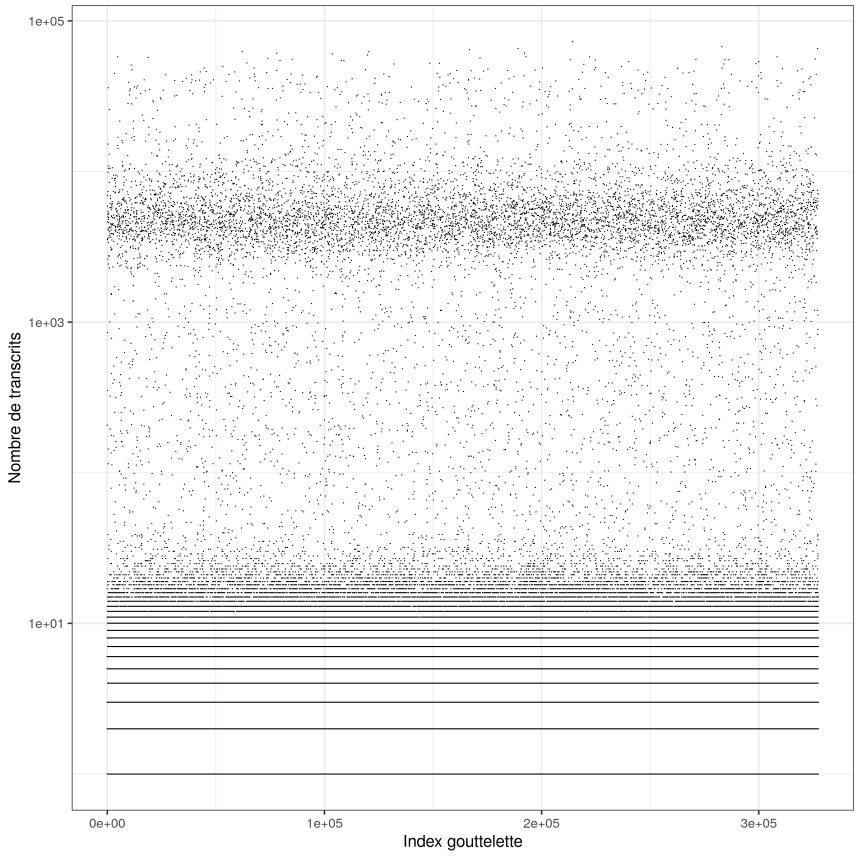


Figure 1 – Nombre de transcrits par gouttelettes

Note de lecture : Chaque point représente le nombre de transcrits en ordonnée pour une gouttelette.

L'idée avec ce graphique est de se faire une première idée des classes que l'on souhaite obtenir en observant des concentrations de points. Nous pouvons remarquer des zones de concentration

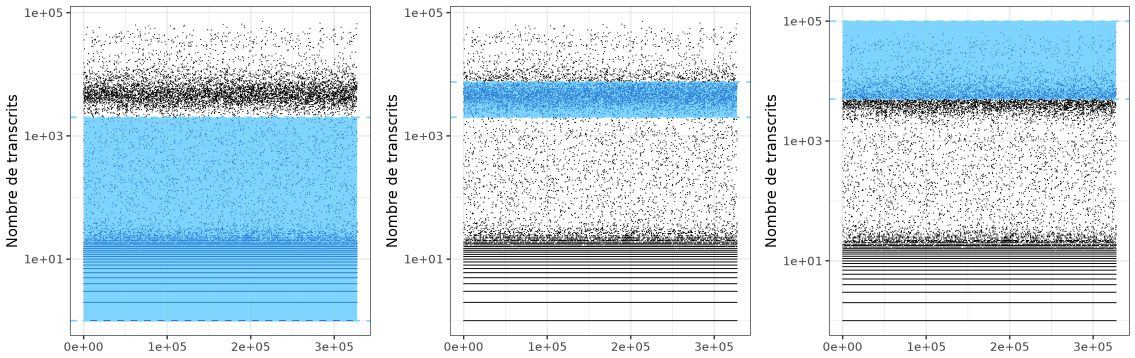
de points à certains endroits que nous définirons pas la suite. On précise que les lignes observables sur le graphique, jusqu'à environ 10 transcrits, sont bien des données discrètes.

Dans la partie suivante, nous mènerons une première analyse visuelle sur le graphique obtenu pour tenter d'approcher les classes que l'on souhaite obtenir.

1.2 Analyse exploratoire des transcrits par gouttelettes

1.2.1 Analyse graphique des classes

L'objectif est de construire 3 classes avec ces données. Nous avons repéré 3 concentration de points que nous avons mis en couleur sur le graphique suivant :



(a) Groupe 1 (b) Groupe 2 (c) Groupe 3

FIGURE 2 – Nombre de transcrits par gouttelettes selon les groupes

Note : A noter que les groupes 2 et 3 peuvent se chevaucher, nous expliquons cela dans la suite

En ordonnée, on peut considérer une première concentration de gouttelletes avec un nombre de transcrits compris entre 1 et 2000 environ. Cette concentration peut représenter la classe avec uniquement des débris de cellules.

On observe une autre concentration de point, mais cette fois-ci sur un intervalle bien plus grand du fait de l'échelle logarithmique. On peut imaginer que cette concentration de points représente alors les gouttelettes à une et deux cellules. Nous savons qu'en moyenne, le nombre de transcrits des gouttelettes contenant deux cellules est le double des gouttelettes contenant une seule cellule. Cependant l'échelle logarithmique ne permet pas de voir ce lien donc on suppose les deux groupes proches et c'est pour cela que les zones en bleus se chevauchent sur le graphique.

On estime visuellement le nombre de transcrits entre 2000 et 7000 pour une unique cellule et au-delà pour deux cellules présentes dans la gouttelette.

Dans la suite, nous allons utiliser les intervalles précédemment définis pour représenter le nombre de transcrits en abcisse par gouttelettes pour chaque groupe. Cela nous permettra d'observer la forme de la distribution du nombre de transcrits.

1.2.2 Etude de la distribution du nombre de transcits au sein des classes

Dans un premier temps, nous allons donc observer les données comprenant de 1 à 2000 transcrits. Nous précisons que le graphique va jusqu'à 25 transcrits car au-delà le nombre de goutte-

lettes étant trop faible, cela ne permet pas de bien observer la forme de la distribution. Sur cet intervalle, les gouttlettes devraient alors comprendre seulement des débris de cellule. Les débris de cellule dans une gouttelette se réfèrent à des particules ou des fragments cellulaires qui se sont détachés de la cellule lors de la collecte des gouttelettes. Nous retrouvons alors le graphique suivant :

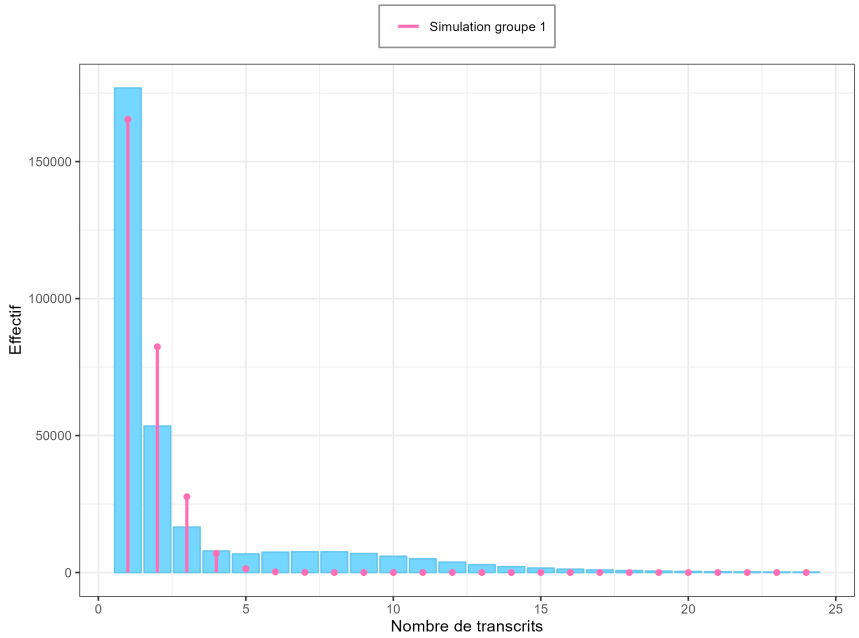


FIGURE 3 – Distribution des gouttelettes comprenant des débris de cellule et fonction de masse d'une loi de Poisson de paramètre 1

Pour les gouttelettes ayant jusqu'à 5 transcrits, on peut reconnaître ici une distribution similaire à celle d'une loi de Poisson, avec un paramètre proche de 1. Cette loi de Poisson serait donc tronquée en 0 car nous avons supprimé ces données. Cependant, on observe une hausse du nombre de gouttelettes ayant plus de 5 transcrits en forme de bosse. Cela risque d'impacter le modèle de mélange avec l'hypothèse faite qu'on représente les débris de cellule par une loi de Poisson.

De plus on observe bien qu'au moins 75% des gouttelettes ont un nombre de transcrits égal ou inférieur à 3.

Pour le deuxième groupe défini comme ayant un nombre de transcrits compris entre environ 2000 et 7000, nous obtenons le graphique suivant :

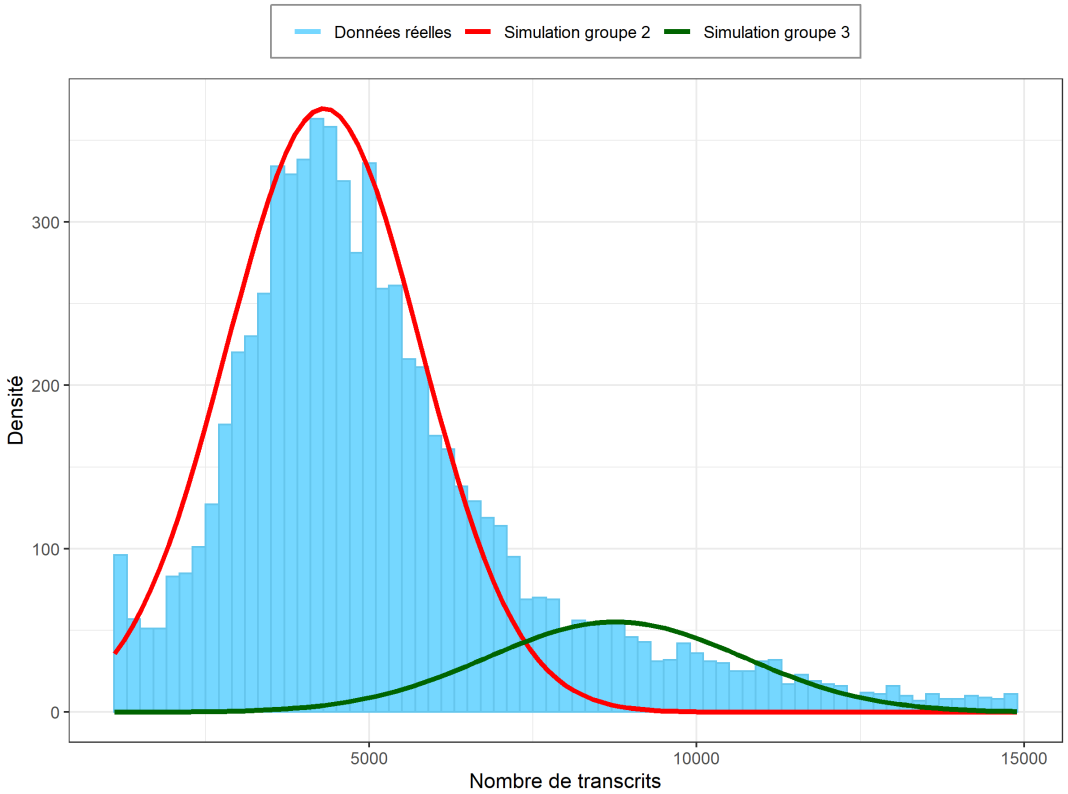


FIGURE 4 – Distribution des groupes 2 et 3 et superposition avec des lois normales
Note de lecture : Chaque point représente le nombre de transcrits en ordonnée pour une gouttelette.

L'histogramme ci-dessus représente la distribution du nombre de transcrits par gouttelettes pour les groupes 2 et 3. À l'aide des informations contenues dans l'énoncé et d'une analyse visuelle, nous voyons que nos données peuvent a priori être approchées par deux lois normales. La distribution des gouttelettes modélisée par la loi normale rouge $\mathcal{N}(4300, 1500^2)$ est censée représenter l'ensemble des gouttelettes contenant une seule cellule. La distribution modélisée par la loi normale verte $\mathcal{N}(8600, 2121^2)$ représente quand à elle l'ensemble des gouttelettes contenant 2 cellules.
On remarque un chevauchement des deux lois pour un nombre de transcrits contenu entre 4000 et 8000. Cela signifie qu'on peut retrouver deux gouttelettes contenant chacunes 7500 transcrits mais pour autant ne pas contenir le même nombre de cellules. L'une peut contenir une cellule avec 7500 transcrits tandis que l'autre peut contenir deux cellules avec 4000 et 3500 transcrits chacune.
Nos données peuvent donc être approchées par un mélange de trois lois, une loi de Poisson et deux lois normales. Pour effectuer la modélisation des ces données, on utilisera un modèle de mélange. Il permettra de déterminer le groupe d'appartenance d'une gouttelette en fonction du nombre de transcrits qu'elle possède, selon les probabilités de chaque groupe. Il y a néanmoins quelques problèmes avec l'approche de nos données par ce mélange de lois.

La loi de Poisson présente un inconvénient car elle ne possède qu'un seul paramètre, λ, qui correspond à son espérance mais aussi sa variance. Cela rend difficile la recherche du paramètre

approprié pour nos données. En effet, au vu de la répartition du nombre de transcrits avec 50% des gouttelettes qui ont 1 transcrit, on se doute que λ ne doit pas être très éloigné de 1. Cependant cela implique aussi une variance proche de 1, donc très faible. Cela risque alors de pas inclure des gouttelettes dans le groupe représenté par la loi de Poisson alors que ces gouttelettes contiennent bien des débris de cellule.

Pour les lois normales, le problème ici réside dans le fait que, dans le sujet, il est suggéré que les données de gouttelettes contenant 1 ou 2 cellules pourraient être modélisées à l'aide de ces dernières, qui sont des lois de densité avec des valeurs dans $\mathbf{R}$. Cependant, nos données sont des nombres entiers positifs. Ainsi, pour modéliser ces données à l'aide de lois normales, qui sont également des lois de densité avec des valeurs dans $\mathbf{R}$, nous devons effectuer une correction de continuité, qui consiste à considérer que chaque valeur entière correspond en réalité à un intervalle continu d'une valeur (par exemple, 1 correspondrait à l'intervalle entre 0,5 et 1,5). Cela nous permet d'approcher nos données par des lois normales et de les utiliser pour effectuer des analyses statistiques appropriées.

Avec une approximation de nos données par un mélange de 3 lois connues, on va pouvoir modéliser nos données grâce à un modèle de mélange.

2 Construction du modèle de mélange

Dans cette partie, nous allons présenter toutes les étapes pour la construction de notre modèle de mélange [4]. Dans un premier temps, nous rappellerons la définition d'un modèle de mélange ainsi que présenterons les différentes notations que nous utiliserons par la suite. Dans un second temps, nous viendrons mettre en place un algorithme d'espérance-maximisation pour estimer les paramètres de notre modèle. Nous pourrons ensuite implémenter le modèle sur **R** pour le tester avec des échantillons. Enfin, nous aborderons les difficultés numériques dans la mise en place de ce modèle sur **R**.

2.1 Présentation et notation

Premièrement, nous allons définir les paramètres de nos lois normales pour la suite de notre modèle. Par la suite, nous pourrons alors expliquer la mise en place de notre modèle de mélange.

2.1.1 Rappels sur la loi normale

Nous rappelons ici que pour une loi normale $\mathcal{N}(0,1)$ on note la densité :

$$\phi(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$$

Et la fonction de répartition :

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt$$

Pour une variable aléatoire $X \sim \mathcal{N}(\mu, \sigma^2)$ la densité s'écrit :

$$f_X(x|\mu,\sigma^2) = \frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)$$

Et la fonction de répartition :

$$F_X(x|\mu,\sigma^2) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

Enfin, on rappelle que pour deux variables aléatoires indépendantes de même loi normale $X_1 \sim \mathcal{N}(\mu,\sigma^2)$ et $X_2 \sim \mathcal{N}(\mu,\sigma^2)$, on a :

$$X_1 + X_2 \sim \mathcal{N}(2\mu, 2\sigma^2)$$

Nous allons par la suite pouvoir utiliser ce résultat pour la construction de notre modèle.

2.1.2 Notre modèle de mélange

On note $x = (x_1, ..., x_n)$ les données et $\mathcal{X} \subset \mathbb{N}^*$ leur support.
Nous avons p la loi de mélange à $K = 3$ composantes de paramètre θ : $p(x_i, \theta)$.
Enfin, $p_1, ..., p_K$ représentent les lois de probabilités du mélange et $\pi_1, ..., \pi_K$ leurs coefficients tels que $\forall k \in [1, K],\ \pi_k > 0$ et $\sum_{k=1}^{K} \pi_k = 1$.

La loi de mélange s'écrit $\forall x_i \in \mathcal{X}$:

$$p(x_i, \theta) = \sum_{k=1}^{K} \pi_k p_k(x_i, \alpha_k)$$

Avec $\theta = \{\{\pi_k, \alpha_k\} : k = 1, ..., K\}$ l'ensemble des paramètres du modèle et α_k les paramètres de la loi p_k.

On note X_i la variable aléatoire qui représente le nombre de transcrits dans une cellule i. Les X_i sont des variables indépendantes et identiquement distribuées. On suppose que cette distribution suit une loi normale $\mathcal{N}(\mu, \sigma^2)$. On note également Y_j la variable aléatoire qui représente le nombre de transcrits dans une gouttelettes j, qui sont également indépendantes et identiquement distribuées.

Nous allons modéliser les gouttelettes du premier groupe, contenant peu de transcrits, par une loi de Poisson de paramètre λ tronquée en zéro. En effet, nous n'avons plus de gouttelettes contenant zéro transcrit comme expliqué précedemment :

$$\forall k \in \mathcal{X} : \mathbb{P}(Y = k | Y \geq 1) = \frac{\mathbb{P}(Y = k)}{\mathbb{P}(Y \geq 1)} = \frac{\lambda^k e^{-\lambda}}{k!} \left(\frac{1}{1 - e^{-\lambda}} \right) = \frac{\lambda^k}{k! (e^{\lambda} - 1)}$$

Pour modéliser les gouttelettes du deuxième groupe, c'est à dire les gouttelettes contenant une unique cellule, nous allons utiliser une loi normale. En effet, nous allons supposer que tous les transcrits contenu dans la gouttelettes proviennent de la cellule : $Y = X \sim \mathcal{N}(\mu, \sigma^2)$. Cependant nos données réelles sont discrètes. Pour les modéliser à l'aide d'une loi continue on peut approcher la fonction de masse par une différence de fonction de répartition [2] :

$$\begin{aligned} &\forall k \in \mathcal{X} \backslash \{1\} : \mathbb{P}(Y = k) = F_Y(k + 1/2) - F_Y(k - 1/2) \\ &\text{et pour } k = 1 : \mathbb{P}(Y = 1) = F_Y(k + 1/2) \end{aligned}$$

Enfin, pour modéliser le troisième groupe, comprenant les gouttelettes ayant deux cellules, nous allons encore utiliser une loi normale. En effet, le nombre de transcrits que l'on retrouve dans la gouttelette est entièrement issu de ces deux cellules, qui sont indépendantes. On a donc $Y = X_1 + X_2 \sim \mathcal{N}(2\mu, 2\sigma^2)$. Comme pour le deuxième groupe nous allons approcher la fonction de masse par une différence de fonction de répartition :

$$\begin{aligned} &\forall k \in \mathcal{X} \backslash \{1\} : \mathbb{P}(Y = k) = F_Y(k + 1/2) - F_Y(k - 1/2) \\ &\text{et pour } k = 1 : \mathbb{P}(Y = 1) = F_Y(k + 1/2) \end{aligned}$$

Ainsi, pour notre modèle de mélange nous avons trois lois telles que $\forall x_i \in \mathcal{X}$:

$$p_1(x_i, \alpha_1) = p_1(x_i, \lambda) = \frac{\lambda^{x_i}}{x_i! (e^{\lambda} - 1)} \tag{1}$$

2. Complément sur la correction de continuité en annexe A

$$p_2(x_i, \alpha_2) = p_2(x_i, \mu, \sigma) = \begin{cases} \Phi\left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma}\right) - \Phi\left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma}\right), & \text{si } x_i \neq 1 \\ \Phi\left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma}\right), & \text{sinon} \end{cases} \tag{2}$$

$$p_3(x_i, \alpha_3) = p_3(x_i, \mu, \sigma) = \begin{cases} \Phi\left(\frac{\left(x_i + \frac{1}{2}\right) - 2\mu}{\sqrt{2}\sigma}\right) - \Phi\left(\frac{(x_i - \frac{1}{2}) - 2\mu}{\sqrt{2}\sigma}\right), & \text{si } x_i \neq 1 \\ \Phi\left(\frac{\left(x_i + \frac{1}{2}\right) - 2\mu}{\sqrt{2}\sigma}\right), & \text{sinon} \end{cases} \tag{3}$$

2.2 Vraisemblance et Algorithme `EM`

Nous allons estimer θ par maximum de vraisemblance. La log-vraisemblance de notre modèle s'écrit :

$$\mathcal{L}(x, \theta) = \sum_{i=1}^{n} \ln \left(\sum_{k=1}^{K} \pi_k p_k \left(x_i, \alpha_k\right) \right)$$

La maximisation de cette quantité est trop compliquée. Définissons Z un variable aléatoire telle que $Z_{ik} = 1$ si l'individu i appartient à la classe k. Cette variable aléatoire représente de l'information supplémentaire sur les données. Nous allons calculer la log-vraisemblance sur cet ensemble de données (x, z), c'est la log-vraisemblance compétée :

$$\mathcal{L}(x, \theta, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \ln \left(\pi_k p_k \left(x_i, \alpha_k\right) \right)$$

Pour maximiser la log-vraisemblance complétée nous allons utiliser l'algorithme `EM`. Cet algorithme itère les étapes `E` et `M` jusqu'à ce qu'un critère d'arrêt soit atteint. Ce critère d'arrêt peut être de différente nature. Une nombre d'itération fixé, une différence entre les paramètres estimés inférieure à une certaine quantité ou encore une différence de log-vraisemblance complétée inférieur à une autre quantité. À l'itération r on a :

— Tout d'abord l'étape `E` permet d'estimer les z_{ik}. En effet, on calcule l'espérance de la log-vraisemblance complétée sachant les données x et les paramètres à l'itération précédente $\theta^{[r-1]}$:

$$\begin{aligned} t_{ik}(\theta^{[r-1]}) &:= \mathbb{E}[Z_{ik} | x, \theta^{[r-1]}] \\ &= \frac{\pi_k p_k(x_i, \alpha_k^{[r-1]})}{p(x_i, \theta^{[r-1]})} \end{aligned}$$

Les $t_{ik}(\theta^{[r-1]})$ s'interprètent également comme la probabilité conditionnelle que x_i ait été généré par la composante k. C'est cette deuxième écriture qui nous permet de les évaluer.

— L'étape `M` va maximiser l'espérance de la log-vraisemblance complétée $\mathcal{L}(x, \theta, t^{[r]})$ en fonction de θ.

$$
\begin{aligned}
\theta^{[r]} &:= \arg\max_{\theta\in\Theta} \mathcal{L}(x,\theta,t^{[r]}) \\
&= \arg\max_{\theta\in\Theta} \sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}(\theta^{[r-1]}) \ln(\pi_k p_k(x_i,\alpha_k)) \\
&= \arg\max_{\theta\in\Theta} \sum_{i=1}^{n}\Bigg[t_{i1}(\theta^{[r-1]}) \ln(\pi_1 p_1(x_i,\lambda)) + t_{i2}(\theta^{[r-1]}) \ln(\pi_2 p_2(x_i,\mu,\sigma)) \\
&\qquad\qquad + t_{i3}(\theta^{[r-1]}) \ln(\pi_3 p_3(x_i,\mu,\sigma))\Bigg] \\
&= \arg\max_{\theta\in\Theta} \sum_{i=1}^{n}\Bigg[t_{i1}(\theta^{[r-1]}) \ln\left(\pi_1 \frac{\lambda^{x_i}}{x_i!\,(e^{\lambda}-1)}\right) \\
&\qquad\qquad + t_{i2}(\theta^{[r-1]}) \ln\left(\pi_2\left[\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-\mu}{\sigma}\right) - \Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-\mu}{\sigma}\right)\right]\right) \\
&\qquad\qquad + t_{i3}(\theta^{[r-1]}) \ln\left(\pi_3\left[\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right) - \Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)\right]\right)\Bigg]
\end{aligned}
$$

2.3 Mise en place de l'algorithme `EM`

Pour réaliser l'étape `M`, nous essayerons de trouver $\theta^{[r]}$ en dérivant l'espérance de la log-vraisemblance complétée $\mathcal{L}(x,\theta,t^{[r]})$ par rapport à chacun des paramètres (λ,μ,σ). On utilisera, ensuite, la méthode du multiplicateur de Lagrange pour prendre en compte la contrainte sur les π_k : $\sum_{k=1}^{K}\pi_k = 1$ et trouver les (π_1,π_2,π_3) optimaux.

2.3.1 Estimation des paramètres (λ,μ,σ)

— On calcul [3] pour λ :

$$
\frac{\partial\mathcal{L}}{\partial\lambda}(x,\theta,t^{[r]}) = \sum_{i=1}^{n} t_{i1}\left(\theta^{[r-1]}\right)\left(\frac{x_i}{\lambda} - \frac{e^{\lambda}}{e^{\lambda-1}}\right) \tag{4}
$$

3. Démonstration en annexe B

— On obtient [4] la formule suivante pour μ :

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \mu}(x, \theta, t^{[r]}) \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \frac{-\frac{1}{\sigma}\phi\left(\frac{(x_i+\frac{1}{2})-\mu}{\sigma}\right) + \frac{1}{\sigma}\phi\left(\frac{(x_i-\frac{1}{2})-\mu}{\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-\mu}{\sigma}\right) - \Phi\left(\frac{(x_i-\frac{1}{2})-\mu}{\sigma}\right)} \\
&+ \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]}) \frac{-\frac{\sqrt{2}}{\sigma}\phi\left(\frac{(x_i+\frac{1}{2})-2\mu}{2\sigma}\right) + \frac{\sqrt{2}}{\sigma}\phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{2\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right) - \Phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{\sqrt{2}\sigma}\right)}
\end{aligned} \tag{5}$$

— Enfin, on a [5] la formule suivante pour σ :

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \sigma}(x, \theta, t^{[r]}) \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \frac{-\left(\frac{\left(x+\frac{1}{2}\right)-\mu}{\sigma^2}\right)\phi\left(\frac{\left(x+\frac{1}{2}\right)-\mu}{\sigma}\right) + \left(\frac{\left(x-\frac{1}{2}\right)-\mu}{\sigma^2}\right)\phi\left(\frac{\left(x-\frac{1}{2}\right)-\mu}{\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-\mu}{\sigma}\right) - \Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-\mu}{\sigma}\right)} \\
&+ \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]}) \frac{-\left(\frac{\left(x+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma^2}\right)\phi\left(\frac{\left(x+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right) + \left(\frac{\left(x-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma^2}\right)\phi\left(\frac{\left(x-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right) - \Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)}
\end{aligned} \tag{6}$$

Nous ne disposons d'aucune solution analytique pour aucun des trois paramètres. Nous sommes donc contraints d'utiliser un algorithme d'optimisation numérique pour estimer (λ, μ, σ). Qu'en est-il des proportions de chaque groupe ?

2.3.2 Estimation des proportions (π_1, π_2, π_3)

Pour calculer $(\pi_1^{[r]}, \pi_2^{[r]}, \pi_3^{[r]})$, nous utiliserons la méthode du multiplicateur de Lagrange car nous disposons d'une contrainte sur les π_k, $\sum_{k=1}^{K} \pi_k = 1$. Nous noterons ici γ le multiplicateur de Lagrange puisque λ est déjà utiliser pour le paramètre de la loi de Poisson. Notre fonction à optimiser devient :

$$\mathcal{L}(x, \theta, t^{[r]}) - \gamma\left(\sum_{k=1}^{K} \pi_k - 1\right)$$

On calcule les dérivées par rapport à chaque $\pi_{k,\, k\in[1,K]}$ et par rapport à γ :

4. Démonstration en annexe C
5. Démonstration en annexe D

— $\forall k \in [1, K]$, on a :

$$\frac{\partial}{\partial \pi_k}\left[\mathcal{L}(x, \theta, t^{[r]}) - \gamma\left(\sum_{k=1}^{K}\pi_k - 1\right)\right] = \sum_{i=1}^{n} t_{ik}(\theta^{[r-1]})\frac{1}{\pi_k} - \gamma$$

Ainsi :

$$\frac{\partial}{\partial \pi_k}\left[\mathcal{L}(x, \theta, t^{[r]}) - \gamma\left(\sum_{k=1}^{K}\pi_k - 1\right)\right] = 0 \quad \Leftrightarrow \quad \pi_k^{[r]} = \frac{1}{\gamma}\sum_{i=1}^{n} t_{ik}(\theta^{[r-1]})$$

— On dérive maintenant par rapport à γ

$$\frac{\partial}{\partial \gamma}\left[\mathcal{L}(x, \theta, t^{[r]}) - \gamma\left(\sum_{k=1}^{K}\pi_k - 1\right)\right] = \sum_{k=1}^{K}\pi_k - 1$$

En utilisant le fait que t_{ik} représente la probabilité que l'individu i appartienne à la classe k, on comprend bien que pour chaque individu, ses probabilités d'appartenir à chacune des classes vont sommer à 1. Ainsi, $\forall i \in [1, n], \sum_{k=1}^{K} t_{ik}(\theta^{[r-1]}) = 1$. On a donc :

$$\begin{aligned}
\frac{\partial}{\partial \gamma}\left[\mathcal{L}(x, \theta, t^{[r]}) - \gamma\left(\sum_{k=1}^{K}\pi_k - 1\right)\right] = 0 \quad &\Leftrightarrow \quad \sum_{k=1}^{K}\pi_k^{[r]} = 1 \\
&\Leftrightarrow \quad \frac{1}{\gamma}\sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}(\theta^{[r-1]}) = 1 \\
&\Leftrightarrow \quad \hat{\gamma} = \frac{1}{n}
\end{aligned}$$

On obtient finalement une forme analytique pour les proportions de chaque groupe :

$$\forall k \in [1, K], \quad \pi_k^{[r]} = \frac{1}{n}\sum_{i=1}^{n} t_{ik}(\theta^{[r-1]}) \tag{7}$$

2.3.3 Optimisation numérique pour μ, σ et λ

Pour trouver numériquement μ, σ et λ nous utilisons la fonction `nloptr` [5] du *package* du même nom. C'est un *package open source* qui implémente plusieurs algorithmes d'optimisation non linéaire. Nous utilisons [6] ici l'algorithme `BOBYQA` [7] [7]. C'est un algorithme très récent, développé par Michael J. D. Powell en 2009, qui présente deux avantages fondamentaux dans notre cas, par rapport aux autres algortihmes d'optimisation.

Le premier est que c'est un algortihme dit *Derivative free*. C'est à dire qu'il n'utilise pas le gradient de la fonction objective pour trouver son minimum. L'algorithme va approcher la fonction objective, par régression polynomiale, et c'est cette fonction approchée qu'il va minimiser. Cela permet de ne pas avoir à calculer le gradient, ni à l'approcher [8].

Le second avantage est que `BOBYQA` est un algorithme de degré 2. La régression polynomiale pour approcher la fonction objective est de degré 2. Cette approche est bien plus précise qu'une approche de degré 1 et permet de grandement accélerer les calculs.

6. Justification de ce choix en 2.5.3
7. **B**ound **O**ptimization **BY** **Q**uadratic **A**pproximation
8. Plus de détails en 2.5.2

2.4 Vérification du modèle avec des échantillons aléatoires

2.4.1 Une bonne estimation des paramètres

Pour vérifier la bonne convergence de notre modèle nous avons simulé un jeu de données issues d'une loi de Poisson et de deux lois normales. Nous connaissons donc le paramètre θ réel de ce jeu de donnée. Nous avons ensuite vérifié que notre modèle de mélange retrouvait bien les paramètres initiaux : $\hat{\theta} \simeq \theta$.

θ		$\hat{\theta}$
θ_1	λ = 3	3,00 ($\pm$ 0,02)
	μ = 5050	5080,84 ($\pm$ 9,28)
	σ = 1010	953,00 ($\pm$ 3,17)
	π_1 = 0,5	0,50 ($\pm$ 0,002)
	π_2 = 0,25	0,25 ($\pm$ 0,001)
	π_3 = 0,25	0,24 ($\pm$ 0,01)

θ		$\hat{\theta}$
θ_3	λ = 250	250,06 ($\pm$ 0,30)
	μ = 8000	8000,90 ($\pm$ 1,99)
	σ = 100	220,42 ($\pm$ 0,80)
	π_1 = 0,2	0,19 ($\pm$ 0,01)
	π_2 = 0,6	0,60 ($\pm$ 0,001)
	π_3 = 0,2	0,20 ($\pm$ 0,003)

θ		$\hat{\theta}$
θ_2	λ = 50	49,98 ($\pm$ 0,08)
	μ = 100	100,04 ($\pm$ 0,11)
	σ = 10	9,87 ($\pm$ 0,09)
	π_1 = 0,6	0,59 ($\pm$ 0,001)
	π_2 = 0,1	0,09 ($\pm$ 0,001)
	π_3 = 0,3	0,29 ($\pm$ 0,002)

θ		$\hat{\theta}$
θ_4	λ = 2	2,00 ($\pm$ 0,03)
	μ = 50	49,99 ($\pm$ 0,03)
	σ = 5	5,01 ($\pm$ 0,04)
	π_1 = 0,2	0,21 ($\pm$ 0,02)
	π_2 = 0,4	0,40 ($\pm$ 0,001)
	π_3 = 0,4	0,39 ($\pm$ 0,01)

TABLE 4 – Comparasion des paramètres réels avec les paramètres estimés

2.4.2 Une classification encourageante

Après avoir vérifié que le modèle de mélange renvoie les mêmes paramètres que ceux donnés en entré, nous voulons vérifier si la classification du modèle correspond réellement à la classification des lois générées. Notre modèle donne pour chaque individu, la probabilité d'appartenir aux groupes 1, 2 ou 3. Nous attribuons ainsi à chaque individu la classe dont sa probabilité d'appartenance est la plus élevée.

Pour ce faire, nous commencons par générer, pour chaque θ ci-dessus, des matrices de confusions afin d'évaluer la qualité des prédictions issues du modèle. Depuis ces matrices, il est possible de calculer le F1-score associé. C'est une mesure qui permet d'évaluer la performance de modèles de classification. Un F1-score proche de 1 signifie que la qualité de classification du modèle est bonne, à l'inverse, plus il est proche de 0, plus la précision du classifieur est faible.

Paramètre \ Groupe	Groupe 1	Groupe 2	Groupe 3
θ_1	1	$0,98$	$0,97$
θ_2	$0,90$	$0,56$	$0,90$
θ_3	$0,87$	$0,89$	$0,88$
θ_4	$0,78$	$0,80$	$0,86$

TABLE 5 – F1-score par classe pour chaque paramètre θ

On retrouve en moyenne un F1-score plutôt élevé, notamment pour θ_1 où il est proche de 1. Le seul désagrément que nous retrouvons correspond au F1-score de θ_2 pour le groupe 2, qui est égal à 0.56. Ce score plutôt faible peut s'expliquer par la trop faible proportion de ce groupe dans cet jeu de données.

Au vu de la variété des lois de mélange utilisées et de la précision des résultats obtenus, nous pouvons estimer que notre modèle de mélange converge correctement.

2.5 Diffculté d'implémentation

Nous présentons ici les principales difficultés rencontrées durant l'implémentation de notre modèle de mélange, ainsi que les solutions que nous avons mises en place pour pallier ces problèmes.

2.5.1 Manque de précision

Le principale problème que nous avons rencontré est le manque de précision des calculs dans **R**. En effet, pour calculer notre vraisemblance nous devons calculer, par exemple, la différence suivante :

$$\Phi\left(\frac{\left(1+\frac{1}{2}\right)-4000}{100}\right)-\Phi\left(\frac{(1-\frac{1}{2})-4000}{100}\right) \tag{8}$$

Chacune de ces évaluations de différence de fonction de répartition sont très faible. De l'ordre de 10^{-350}, pour cet exemple. Notre première solution a été d'augmenter la précision des calculs de **R**. Grâce au package `Rmpfr`, pour *Multiple Precision Floating-Point Reliable*, nous étions en mesure d'augmenter le nombre de bits sur lesquels étaient codées nos données. Cependant, le temps de calculs de la moindre opération prenait alors plusieurs minutes. Cela est dû au fait que pour coder sur plusieurs bits le package `Rmpfr` stock les décimales dans des listes de listes. Cette solution n'est pas viable pour supporter tous les calculs que notre modèle requiert.

La seconde solution que nous avons trouvée, et que nous avons gardée, est de passer par une astuce calculatoire qui ne perd pas de précision numérique. En effet, dans notre calcul de la log-vraisemblance complétée apparaît des calculs du type :

$$\log\left(\Phi(A)-\Phi(B)\right)$$

Calculé de cette manière, **R** renvoie souvent **Inf**. L'astuce est d'écrire le calcul sous la forme :

$$\log\left(e^{\log(\Phi(A))}-e^{\log(\Phi(B))}\right)$$

Le package *DPQ* implémente une fonction qui permet de réaliser notre calcul de manière numériquement stable :

$$logspace.sub\,(A, B) = \log\left(e^A - e^B\right)$$

Ainsi, sous cette forme, la majorité de nos calculs sont dépourvus de **Inf**, **-Inf** et 0 numériques. Malgré cela, lors de l'optimisation numérique, il se peut que $\mathbf{R}$ approxime certains calculs à **Inf**. POur gérer ces cas la nous avons utilisé la fonction `try` de $\mathbf{R}$ qui permet de ne garder que les cas où nous ne rencontrons pas de problèmes numériques.

2.5.2 Optimisation difficile

La mise en place de l'optimisation numérique des paramètres μ, σ et λ a été l'un des problèmes les plus chronophages de notre travail. Notre première idée a été d'utiliser la fonction `optim`, qui permet d'implémenter l'algorithme d'optimisation `L-BFGS-B` [9]. Cela s'est avéré infructueux et nous avons plusieurs hypothèses pour expliquer cela.

Tout d'abord `L-BFGS-B`[3] est un algorithme d'optimisation qui utilise une approximation de la fonction objective par son développement de Taylor. Pour effectuer cette approximation l'algorithme a besoin du gradient. Soit l'utilisateur le lui donne, soit l'algorithme l'approche par différence finie. À cause de calculs similaires à la différence des fonctions de répartitions (8), nous n'avons pas réussi à calculer le gradient de la log-vraisemblance complétée dans $\mathbf{R}$. L'algorithme a donc utilisé une approximation du gradient. De plus, au vu des erreurs sorties par la fonction `optim`, nous pouvons emmettre l'hypothèse que notre fonction objective, la log-vraisemblance complétée, est trop plate pour que l'algorithme trouve un maximum.

Nous nous sommes donc tournés vers le *package* `nloptr` qui offre une plus grande variété d'algorithmes d'optimisation. Notamment des algorithmes qui n'ont pas besoin du gradient pour trouver le maximum de la fonction objective. Il existe deux grandes classes d'algorithmes de ce type. Les algorithmes *Pattern Search*, comme la méthode du `Simplex`[6] (Nelder-Mead) ou le `Golden-section Search`. On trouve également les algorithmes dit *modèles locaux*, qui approchent la fonction objective par régression polynomiale ou par interpolation et maximisent cette approximation. Les algorithmes `COBYLA`[8] et `BOBYQA` sont de tels algorithmes. Il a donc fallu choisir un algorithme dans cette classe et déterminer celui qui a la convergence la plus rapide.

2.5.3 Temps de calculs

L'optimisation sur 10^5 données prend beaucoup de temps, pour n'importe quel algorithme d'optimisation. C'est pourquoi la première chose à faire a été de choisir le « meilleur » algorithme pour notre problème. Nous en avons donc, arbitrairement, retenu trois :

- `NELDERMEAD`
- `COBYLA`
- `BOBYQA`

Pour faire notre choix, nous avons d'abord vérifié qu'ils convergeaient tous, en moyenne, vers le même maximum de vraisemblance.

9. **L**imited-memory **B**royden–**F**letcher–**G**oldfarb–**S**hanno **B**ounded

Statistique \ Algorithme	NELDERMEAD	COBYLA	BOBYQA
Moyenne	$-732418(\pm 138738)$	$-786151(\pm 158637)$	$-744238(\pm 138288)$
Maximum	-622598	-622598	-622598
Minimum	-959582	-960101	-959440

TABLE 6 – Comparaison de la log-vraisemblance complétée obtenue par chacun des algorithmes, sur plusieurs échantillons aléatoires issues de la même loi

Puis, nous les avons testés pour différentes tailles d'échantillons, et relevé le temps mis par la foncton `nloptr` pour optimiser μ et σ.

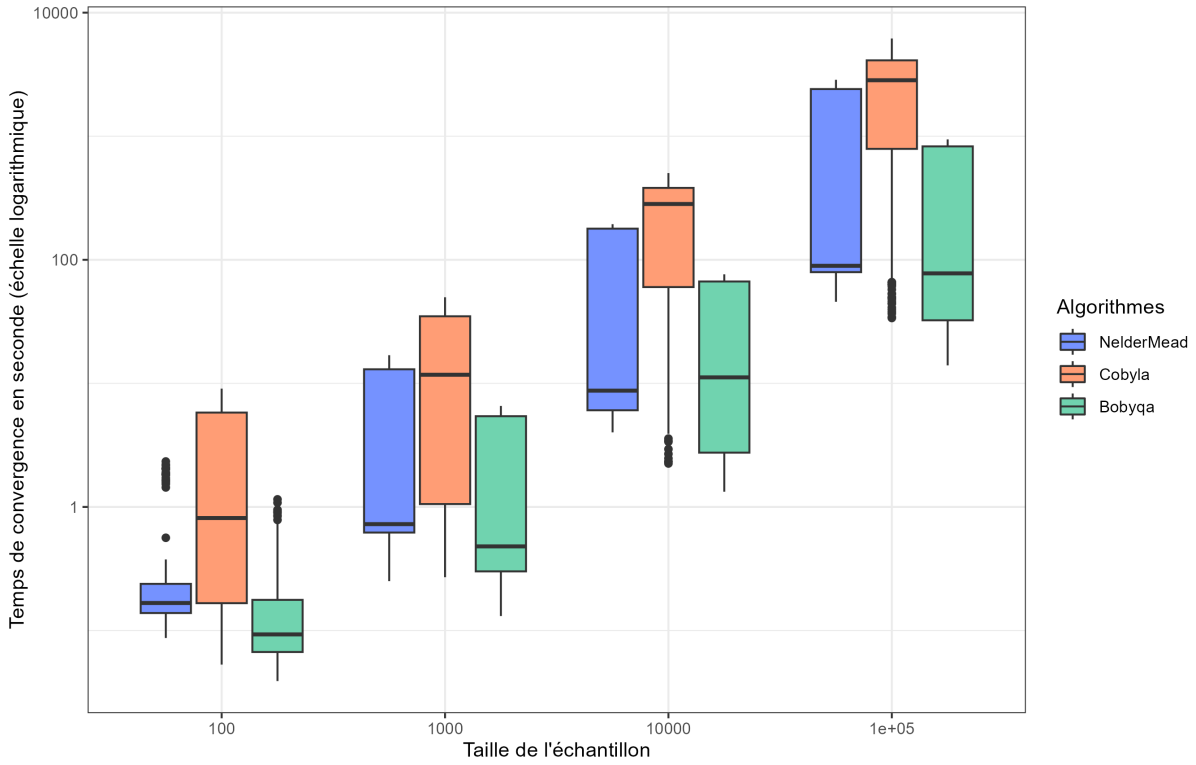


FIGURE 5 – Etude de l'évolution du temps de convergence, par algorithme, en fonction de la taille de l'échantillon

On voit bien que l'algorithme `BOBYQA` est plus rapide pour chaque taille d'échantillon. Cette tendance est d'autant plus importante que l'échelle du temps de convergence est logarithmique.

La deuxième solution pour accélérer les calculs a été de transférer notre code sur les *clusters* de l'ENSAI. Cela a eu deux bénéfices. Le premier est que les processeurs du *cluster* sont plus rapides que ceux dont nous disposons sur nos machines personnelles. Le second avantage a été la libération de nos ordinateurs. En effet, faire tourner notre programme sur nos machines nous empêchait de les utiliser librement. Le *cluster* a également cet avantage de pouvoir être accessible depuis n'importe quel ordinateur.

Enfin, nous avons utilisé la parallélisation sur plusieurs cœurs, rendue possible grâce aux packages `doParallel` et `foreach`. Aujourd'hui les processeurs ont plusieurs cœurs et sont donc capables d'effectuer plusieurs opérations en parallèle. Les *clusters* de l'école en disposent d'un grand nombre, 32. C'est notamment utile lorsque l'on utilise de nombreuse boucle `for`. En utilisant tous les cœurs du cluster nous pouvons réaliser 32 passages simultanés dans la boucle. Ce n'est pas négligeable. Cette parallélisation a été très utile notamment lors de la vérification de notre modèle sur des données générées et lors de la comparaison des algorithmes d'optimisation.

Parallélisation / Usage	Avec	Sans
Vérification	2h	$\simeq$40h
Comparaison	7h	$\simeq$210h
Application	3h	$\simeq$90h

TABLE 7 – Comparaison des temps de calcul avec et sans parallélisation pour chacun de nos cas d'usage

Les temps de calculs sans parallélisation sont des estimations. Nous avons utilsé seulement 30 coeurs simultanément, donc le temps de calcul sans parallélisation est environ trente fois plus élevé.

3 Une première classification

Dans cette section, nous allons discuter de notre modèle de mélange, comprendre ce qu'il nous apprend et ses limites.

3.1 Sorties de l'algorithme EM

Sur 300 initialisations aléatoires de l'algorithme EM, nous avons gardé le résultat convergeant vers la log-vraisemblance complétée la plus élevée. Nous avons implémenté comme critère d'arrêt, une différence entre deux log-vraisemblances complétées successive inférieure à 0.1.

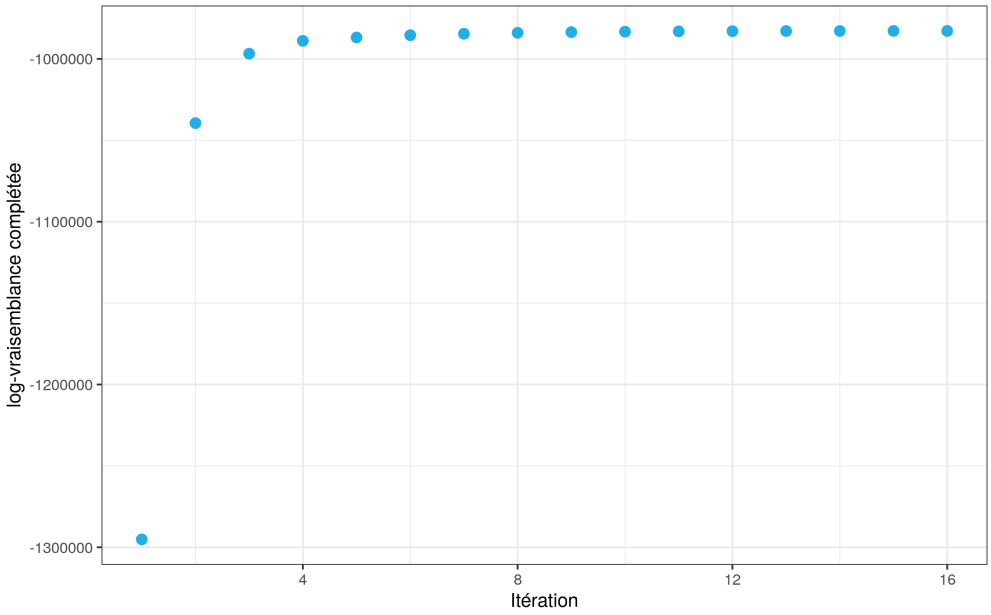
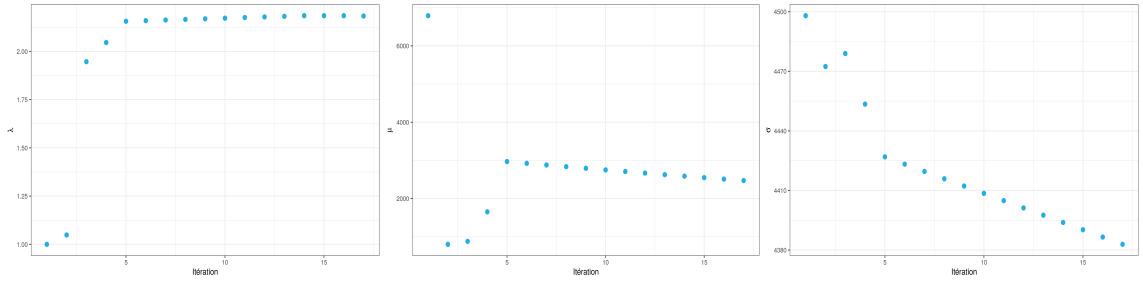


FIGURE 6 – Évolution de la log-vraisemblance complétée au fil des itérations de l'algorithme EM

Nous vérifions bien une log-vraisemblance complétée strictement croissante. Nous arrivons finalement à une log-vraisemblance complétée de -982834. Malheureusement ce résultat brut n'est pas interprétable. Il faudrait le comparer à la log-vraisemblance complétée d'un autre modèle. Nous allons discuter d'autres méthodes pour rendre compte de la fiabilité du modèle. Commençons par étudier les paramètres estimés du modèle.



(a) Évolution de λ (b) Évolution de μ (c) Évolution de σ

FIGURE 7 – Évolution de l'estimation des paramètres au cours des itérations de l'algortihme EM

Nous obtenons pour λ une valeur de $2,18$. Ce qui est cohérent avec ce que nous avions estimé à l'aide de notre analyse exploratoire. Le paramètre μ converge lui vers 2471. Ce qui est un peu en deçà de ce à quoi nous nous attendions, sans que cette valeur ne semble aberrante non plus. Enfin, σ converge vers 4382. Ce qui peut sembler élevé et peut augmenter la difficulté à définir les groupes. Cette variance, supérieur à la moyenne, indique qu'un nombre de gouttelette non négligeable pourraient avoir un nombre de transcrits négatifs. Cela mets en garde contre la qualité et la fidelité de notre modèle. Regardons maintenant la convergence des proportions de chaque groupe.

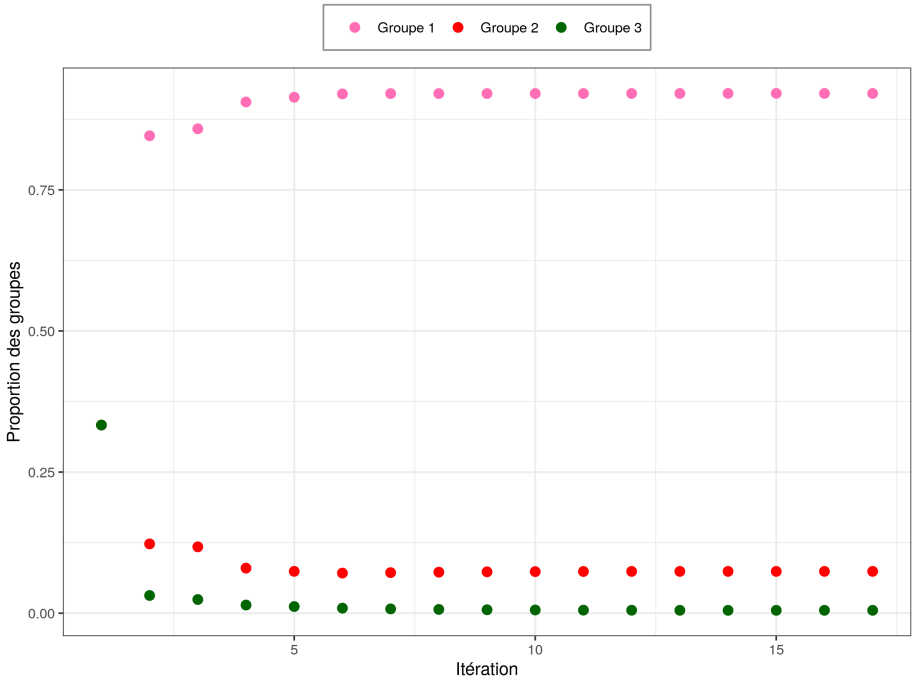
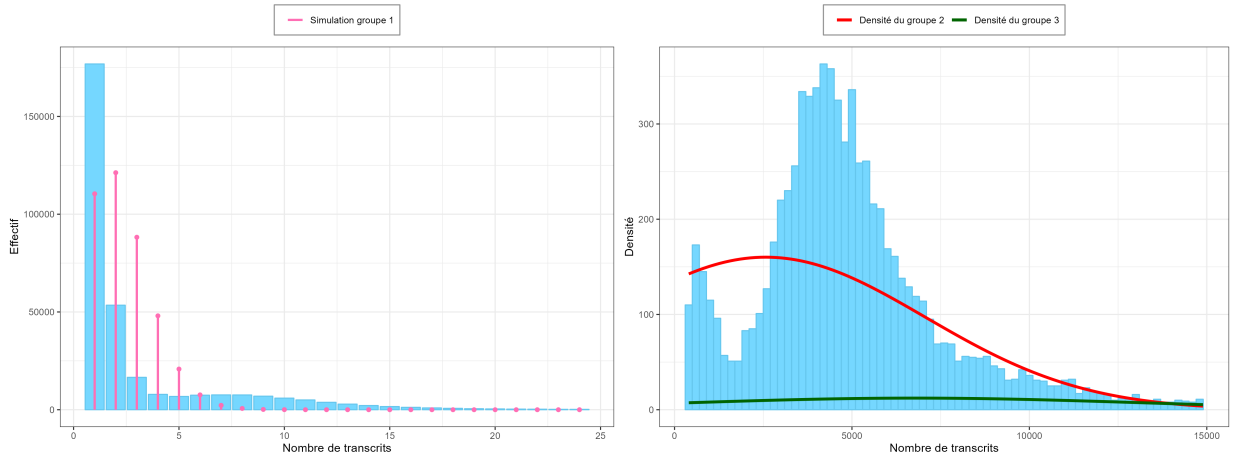


FIGURE 8 – Évolution des proportions de chaque groupes au cours de l'algorithme `EM`

Le groupe 1 représente 92% des gouttelettes, comme ce que nous envisagions puisque les gouttelettes contenant entre 1 et 6 transcrits représentent environ 80% de toutes les gouttelettes. Le groupe 2 contient 7% des gouttelettes et le groupe 3 seulement 1%. Ce dernier est extrêmement rare, en effet, il est difficile d'obtenir deux cellules intactes dans une seule gouttelette avec la méthode utilisée.

3.2 Analyse des résultats

Maintenant que nous avons eu les résultats des différents paramètres pour les lois, ainsi que les pourcentages des groupes, nous pouvons construire un graphique avec nos données et les distributions des lois. Nous obtenons les graphiques suivants :



(a) Pour le groupe 1

(b) Pour les groupe 2 et 3

FIGURE 9 – Comparaison de la distribution des données et des lois de mélange

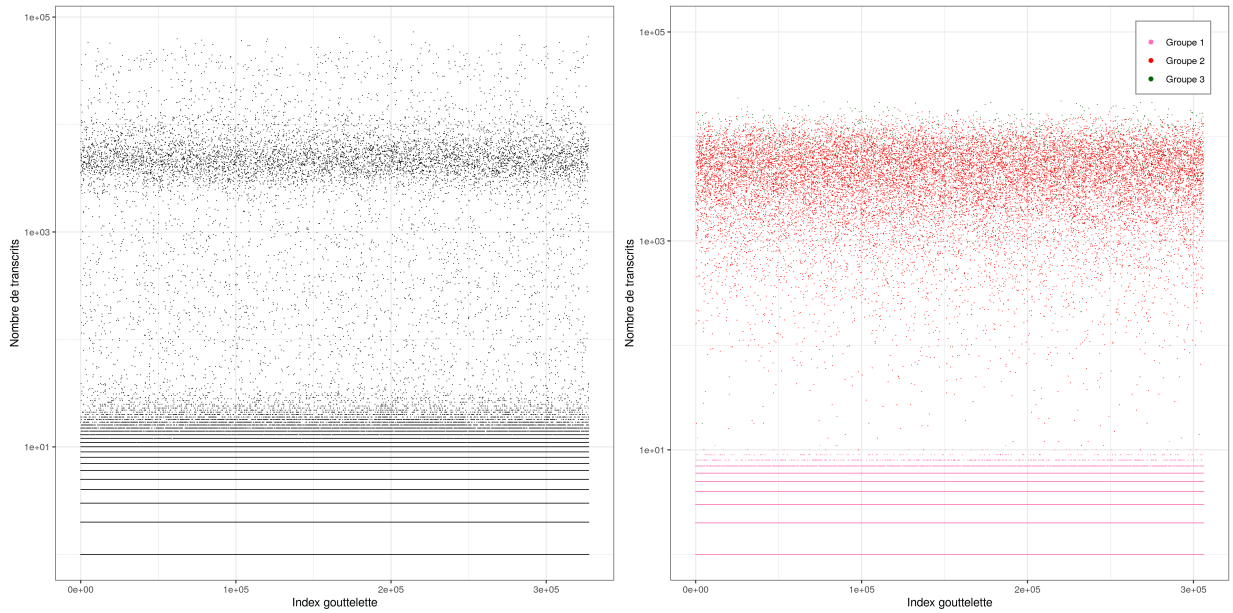
On constate que la loi de Poisson modèlise incorrectement les données. Notre modèle sous-estime le nombre de gouttelettes à un seul transcrit mais sur-estime les gouttelettes contenant 2 à 5 transcrits. On remarque aussi qu'elle modélise mal les gouttelettes formant un replat entre 6 et 15 transcrits. Le graphique 9b mets en évidence les défauts de modélisation des lois normales. La loi normale associée au groupe 2 (rouge), possède un écart-type trop élevé par rapport à la distribution réelle. La densité se trouve alors aplatie et modélise un nombre trop important d'individu différents. Notamment les gouttelettes contenant entre 6 et 1000 transcrits qui sont mal modélisées par la loi de Poisson.

Le modèle de mélange nous permet d'attribuer un groupe à chacune des gouttelettes. Étudions cette classification.

Groupes / Statistiques	Groupe 1	Groupe 2	Groupe 3
Minimum	1	12	15558
Moyenne	$2,4\ (\pm 2,5)$	$1550\ (\pm 2714)$	$31313\ (\pm 12039)$
Maximum	11	15696	72639
Population	301921	24963	511
Proportion	$92,2\%$	$7,6\%$	$0,2\%$

On peut voir que le modèle de mélange sépare distinctement les classes 1 et 2. Le groupe 1 n'inclut que des gouttelettes avec 11 transcrits ou moins. Ce n'est pas correct puisque les gouttelettes contenant une centaine de transcrits devraient aussi appartenir au groupe 1. C'est en étudiant le groupe 2 qu'on se rend compte que ces gouttelettes sont associées au groupe contenant une cellule. Comme anticipé, la loi de Poisson ne parvient pas à modéliser à la fois un très grand nombre de gouttelettes contenant un transcrit et des gouttelettes contenant une centaine de transcrits. En effet, la moyenne et la variance d'une loi de Poisson sont confondues dans le paramètre λ. Par conséquent, les gouttelettes contenant entre 11 et 1000 transcrits sont, à tord, modélisées par la loi normale du groupe 2. Ceci a pour effet de diminuer la moyenne de ce groupe et d'en augmenter considérablement la variance. Le groupe 3 est lui, indirectement impacté par ce problème de modélisation.

Pour vérifier que notre loi estimée par modèle de mélange est fidèle à la loi réelle des données, nous avons générer un échantillon aléatoire suivant notre loi de mélange, de 300000 individus et comparé la répartition du nombre de transcrits par gouttelettes de notre loi de mélange à celle des données réelles.



(a) Pour les données réelles (b) Pour un échantillon issu de notre loi de mélange

FIGURE 10 – Nombre de transcrits par gouttelettes

Nous pouvons voir que notre loi modélise mal les gouttelettes du groupe 3 et donc un très grand nombre de transcrits. Ces gouttelettes sont en réalité moins rares que ce que prévoit notre modèle. De plus, on constate bien sur la figure 10 que la composante de Poisson ne peut modéliser les gouttelettes contenant beaucoup de débris. En particulier, à partir d'une dizaine de transcrits les gouttelettes sont modélisées par la seconde composante. Cette composante normale, qui modélise les gouttelettes à une cellule doit en plus modéliser les gouttelettes contenant des débris.

Pour pallier à ce problème nous avons pensé à deux solutions. La première est d'isoler les gouttelettes qui ont entre une dizaine et un millier de transcrits à l'aide d'une quatrième composante dans le modèle de mélange. La seconde idée, est de modéliser le groupe 1 par une autre distribution que la loi de Poisson comme une loi binomiale négative. En effet, cette distribution ne couple pas la moyenne et la variance. Elle pourrait mieux correspondre au groupe 1 que la loi de Poisson.

Conclusion

Avec ce modèle de mélange, on devrait pouvoir distinguer les différentes populations de gouttelettes en termes de leur contenu en ARN. Cependant, le manque de données de référence pour évaluer la performance du modèle sur des données réelles contraint l'analyse des résultats.

Malgré cela, nous avons créer des échantillons suivant les lois en hypothèse pour tester notre modèle. Les résultats sur les échantillons sont très encourageant pour la réussite de notre modèle, en effet, comme nous l'avons montré précédemment, les données d'échantillons sont correctement classées. Ainsi, le modèle devrait être capable de distinguer les gouttelettes contenant une unique cellule de celles qui en contiennent plus d'une, ainsi que les gouttelettes contenant des débris cellulaires. Néanmoins, cela supposerait que les lois en hypothèse représentent bien la réalité de nos données. Or, nous avons vu en analysant les résultats que notre modèle ne permet pas de bien classer les gouttelettes.

Le problème est donc que les lois ne représentent pas bien les données. On peut considérer que le modèle est trop simpliste pour créer nos trois groupes. On pense qu'il pourrait être intéressant de représenter les débris de cellulles par deux groupes. Nous garderions la loi de Poisson mais en ajoutant une autre loi normale pour représenter la bosse présente après la distribution de la loi de Poisson. De plus, il est possible d'explorer des modèles de mélange avec d'autres lois de probabilités. Enfin, il est possible explorer d'autres approches pour définir des seuils pertinents pour la sélection des gouttelettes. Ces approches peuvent être basées sur des critères biologiques ou physiologiques pour sélectionner les gouttelettes appropriées.

En conclusion, notre modèle de mélange est une méthode prometteuse pour sélectionner les gouttelettes contenant une unique cellule. Cependant, il est encore trop loin de la réalité pour être utilisé.

Références

[1] "Analyses données RNAseq single cell". In : (). URL : https://indico.math.cnrs.fr/event/3780/contributions/3241/attachments/2196/2551/01_scRNAseq_pres_Delphine_Bioinfo_2018-10-18.pdf.

[2] Fondation ARC. "Les leucémies de l'adulte". In : (). URL : https://www.fondation-arc.org/sites/default/files/2017-03/brochure_leucemies_adulte.pdf.

[3] Charles George Broyden et al. "The convergence of a class of double-rank minimization algorithms, A New Approach to Variable Metric Algorithms, A Family of Variable Metric Updates Derived by Variational Means, Conditioning of quasi-Newton methods for function minimization". In : (1970).

[4] Camille BRUNET-SAUMARD. "Modèles de mélange". In : *Polycopié de cours, 1ère année, ENSAI* (2022).

[5] Steven G. Johnson. *The NLopt nonlinear-optimization package*. URL : http://github.com/stevengj/nlopt.

[6] J. A. Nelder et R. Mead. "A simplex method for function minimization". In : *The Computer Journal 7* (1965).

[7] M. J. D. Powell. "The BOBYQA algorithm for bound constrained optimization without derivatives". In : *Department of Applied Mathematics and Theoretical Physics, Cambridge England, technical report NA2009/06* (2009). DOI : http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf.

[8] M. J. D. Powell, S. Gomez et J.-P. Hennart. "A direct search optimization method that models the objective and constraint functions by linear interpolation". In : *Advances in Optimization and Numerical Analysis, eds.* (1994).

[9] Grégory Ségala. "Corrélation génome transciptome". In : (). URL : https://www.rts.ch/decouverte/sante-et-medecine/corps-humain/9022318-quelle-est-la-correlation-entre-genome-transcriptome-proteome-et-metabolome-regulation-de-lexpression-des-genes.html.

[10] Wikipédia. "Le génome". In : (). URL : https://fr.wikipedia.org/wiki/G%C3%A9nome.

A Correction de continuité

Pour approcher une variable aléatoire discrète X par une loi de probabilié continue, on utilise ce que l'on appelle la correction de continuité. On réécrit la probabilité de la fonction de masse $\forall k \in \mathbb{N}$:

$$\mathbb{P}(X = k)$$

comme la probabilité que X soit dans l'intervalle $[k - 1/2, k + 1/2]$:

$$\mathbb{P}(k - 1/2 \leq X \leq k + 1/2)$$

Cette probabilité se réécrit ensuite comme une différence de fonction de répartition :

$$\begin{aligned}
\mathbb{P}(k - 1/2 \leq X \leq k + 1/2) &= 1 - \mathbb{P}(k - 1/2 \leq X) - \mathbb{P}(X \geq k + 1/2) \\
&= 1 - \mathbb{P}(X \geq k + 1/2) - \mathbb{P}(k - 1/2 \leq X) \\
&= F_X(k + 1/2) - F_X(k - 1/2)
\end{aligned}$$

Il faut faire attention aux bornes du support de X :

$$\mathbb{P}(X = 0) \Rightarrow \mathbb{P}(X \leq 0 + 1/2) \text{ et } \mathbb{P}(X = n_{max}) \Rightarrow \mathbb{P}(X \geq n_{max} - 1/2)$$

Dans notre cas, nous ne considererons que la borne inférieur, puisque nous pouvons supposer un nombre maximum de transcrits dans une gouttelettes tendant vers l'infini.

B Dérivée partielle par rapport à λ

On résout $\frac{\partial \mathcal{L}}{\partial \lambda}(x, \theta, t^{[r]}) = 0$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \lambda}(x, \theta, t^{[r]}) &= \frac{\partial}{\partial \lambda} \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik}(\theta^{[r-1]}) \ln(\pi_k p_k(x_i, \alpha_k)) \\
&= \frac{\partial}{\partial \lambda} \sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) \ln(\pi_1 p_1(x_i, \lambda)) \\
&= \sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) \frac{\partial}{\partial \lambda} \left[\ln(\frac{\lambda^{x_i} e^{-\lambda}}{x_i!}) \right] \\
&= \sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) \frac{\partial}{\partial \lambda} \left[x_i \ln(\lambda) - \lambda - \ln(x_i!) \right] \\
&= \sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) (\frac{x_i}{\lambda} - 1)
\end{aligned}$$

Ainsi, on a :

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \lambda}(x, \theta, t^{[r]}) = 0 &\Leftrightarrow \sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) (\frac{x_i}{\lambda} - 1) = 0 \\
&\Leftrightarrow \frac{1}{\lambda} \sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) x_i = \sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) \\
&\Leftrightarrow \lambda^{[r]} = \frac{\sum_{i=1}^{n} t_{i1}(\theta^{[r-1]}) x_i}{\sum_{i=1}^{n} t_{i1}(\theta^{[r-1]})}
\end{aligned}$$

C Dérivée partielle par rapport à μ

Calculons $\frac{\partial \mathcal{L}}{\partial \mu}(x, \theta, t^{[r]})$:

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \mu}(x, \theta, t^{[r]}) \\
&= \frac{\partial}{\partial \mu} \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik}(\theta^{[r-1]}) \ln(\pi_k p_k(x_i, \alpha_k)) \\
&= \frac{\partial}{\partial \mu} \left[\sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln(p_2(x_i, \mu, \sigma)) + t_{i3}(\theta^{[r-1]}) \ln(p_3(x_i, \mu, \sigma)) \right] \\
&= \frac{\partial}{\partial \mu} \left[\sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right) \right) \right. \\
&\qquad \left. + t_{i3}(\theta^{[r-1]}) \ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) 2\mu}{\sqrt{2}\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - 2\mu}{\sqrt{2}\sigma} \right) \right) \right]
\end{aligned}
$$

Pour une meilleure lisibilité, nous séparons le calcul en deux et nous nous appuyerons sur la Propriété 1 [10] :

$$
\begin{aligned}
\frac{\partial \Phi}{\partial \mu} \left(\frac{(x_i \pm \frac{1}{2}) - \mu}{\sigma} \right) &= -\frac{1}{\sigma} \phi \left(\frac{(x_i \pm \frac{1}{2}) - \mu}{\sigma} \right) \\
\frac{\partial \Phi}{\partial \mu} \left(\frac{(x_i \pm \frac{1}{2}) - 2\mu}{\sqrt{2}\sigma} \right) &= -\frac{\sqrt{2}}{\sigma} \phi \left(\frac{(x_i \pm \frac{1}{2}) - 2\mu}{\sqrt{2}\sigma} \right)
\end{aligned}
$$

— Commençons par calculer :

$$
\begin{aligned}
&\frac{\partial}{\partial \mu} \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln(p_2(x_i, \mu, \sigma)) \\
&= \frac{\partial}{\partial \mu} \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right) \right) \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \frac{\partial}{\partial \mu} \left[\ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right) \right) \right] \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \frac{-\frac{1}{\sigma} \phi \left(\frac{(x_i + \frac{1}{2}) - \mu}{\sigma} \right) + \frac{1}{\sigma} \phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right)}{\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right)}
\end{aligned}
$$

10. Démonstration en annexe E

— On a aussi :

$$\begin{aligned}
&\frac{\partial}{\partial\mu}\sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\ln(p_3(x_i,\mu,\sigma)) \\
&= \frac{\partial}{\partial\mu}\sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\ln\left(\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{\sqrt{2}\sigma}\right)\right) \\
&= \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\frac{\partial}{\partial\mu}\left[\ln\left(\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{\sqrt{2}\sigma}\right)\right)\right] \\
&= \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\frac{-\frac{\sqrt{2}}{\sigma}\phi\left(\frac{(x_i+\frac{1}{2})-2\mu}{\sqrt{2}\sigma}\right)+\frac{\sqrt{2}}{\sigma}\phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{\sqrt{2}\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{\sqrt{2}\sigma}\right)}
\end{aligned}$$

Ainsi,

$$\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial\mu}(x,\theta,t^{[r]}) \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]})\frac{-\frac{1}{\sigma}\phi\left(\frac{(x_i+\frac{1}{2})-\mu}{\sigma}\right)+\frac{1}{\sigma}\phi\left(\frac{(x_i-\frac{1}{2})-\mu}{\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-\mu}{\sigma}\right)-\Phi\left(\frac{(x_i-\frac{1}{2})-\mu}{\sigma}\right)} \\
&+ \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\frac{-\frac{\sqrt{2}}{\sigma}\phi\left(\frac{(x_i+\frac{1}{2})-2\mu}{2\sigma}\right)+\frac{\sqrt{2}}{\sigma}\phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{2\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{(x_i-\frac{1}{2})-2\mu}{\sqrt{2}\sigma}\right)}
\end{aligned}$$

D Dérivée partielle par rapport à σ

Calculons $\frac{\partial \mathcal{L}}{\partial \sigma}(x, \theta, t^{[r]})$:

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \sigma}(x, \theta, t^{[r]}) \\
&= \frac{\partial}{\partial \sigma} \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik}(\theta^{[r-1]}) \ln(\pi_k p_k(x_i, \alpha_k)) \\
&= \frac{\partial}{\partial \sigma} \left[\sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln(p_2(x_i, \mu, \sigma)) + t_{i3}(\theta^{[r-1]}) \ln(p_3(x_i, \mu, \sigma)) \right] \\
&= \frac{\partial}{\partial \sigma} \left[\sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right) \right) \right. \\
&\qquad \left. + t_{i3}(\theta^{[r-1]}) \ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) 2\mu}{\sqrt{2}\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - 2\mu}{\sqrt{2}\sigma} \right) \right) \right]
\end{aligned}
$$

À nouveau, faisons deux étapes de calculs, en s'aidant de la Propriété 2 [11] :

$$
\begin{aligned}
\frac{\partial \Phi}{\partial \sigma} \left(\frac{(x_i \pm \frac{1}{2}) - \mu}{\sigma} \right) &= - \left(\frac{(x_i \pm \frac{1}{2}) - \mu}{\sigma^2} \right) \phi \left(\frac{(x_i \pm \frac{1}{2}) - \mu}{\sigma} \right) \\
\frac{\partial \Phi}{\partial \sigma} \left(\frac{(x_i \pm \frac{1}{2}) - 2\mu}{\sqrt{2}\sigma} \right) &= - \left(\frac{(x_i \pm \frac{1}{2}) - \mu}{\sqrt{2}\sigma^2} \right) \phi \left(\frac{(x_i \pm \frac{1}{2}) - 2\mu}{\sqrt{2}\sigma} \right)
\end{aligned}
$$

— Dans un premier temps, on a :

$$
\begin{aligned}
&\frac{\partial}{\partial \sigma} \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln(p_2(x_i, \mu, \sigma)) \\
&= \frac{\partial}{\partial \sigma} \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right) \right) \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \frac{\partial}{\partial \sigma} \ln \left(\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{(x_i - \frac{1}{2}) - \mu}{\sigma} \right) \right) \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]}) \frac{- \left(\frac{\left(x + \frac{1}{2}\right) - \mu}{\sigma^2} \right) \phi \left(\frac{\left(x + \frac{1}{2}\right) - \mu}{\sigma} \right) + \left(\frac{\left(x - \frac{1}{2}\right) - \mu}{\sigma^2} \right) \phi \left(\frac{\left(x - \frac{1}{2}\right) - \mu}{\sigma} \right)}{\Phi \left(\frac{\left(x_i + \frac{1}{2}\right) - \mu}{\sigma} \right) - \Phi \left(\frac{\left(x_i - \frac{1}{2}\right) - \mu}{\sigma} \right)}
\end{aligned}
$$

11. Démonstration en annexe E

— Dans un second temps, on trouve :

$$
\begin{aligned}
&\frac{\partial}{\partial\sigma}\sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\ln(p_3(x_i,\mu,\sigma)) \\
&= \frac{\partial}{\partial\sigma}\sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\ln\left(\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)\right) \\
&= \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\frac{\partial}{\partial\sigma}\ln\left(\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)\right) \\
&= \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\frac{-\left(\frac{\left(x+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma^2}\right)\phi\left(\frac{\left(x+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)+\left(\frac{\left(x-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma^2}\right)\phi\left(\frac{\left(x-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)}
\end{aligned}
$$

Ainsi,

$$
\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial\sigma}(x,\theta,t^{[r]}) \\
&= \sum_{i=1}^{n} t_{i2}(\theta^{[r-1]})\frac{-\left(\frac{\left(x+\frac{1}{2}\right)-\mu}{\sigma^2}\right)\phi\left(\frac{\left(x+\frac{1}{2}\right)-\mu}{\sigma}\right)+\left(\frac{\left(x-\frac{1}{2}\right)-\mu}{\sigma^2}\right)\phi\left(\frac{\left(x-\frac{1}{2}\right)-\mu}{\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-\mu}{\sigma}\right)-\Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-\mu}{\sigma}\right)} \\
&+ \sum_{i=1}^{n} t_{i3}(\theta^{[r-1]})\frac{-\left(\frac{\left(x+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma^2}\right)\phi\left(\frac{\left(x+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)+\left(\frac{\left(x-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma^2}\right)\phi\left(\frac{\left(x-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)}{\Phi\left(\frac{\left(x_i+\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)-\Phi\left(\frac{\left(x_i-\frac{1}{2}\right)-2\mu}{\sqrt{2}\sigma}\right)}
\end{aligned}
$$

E Démonstrations

Propriété 1. *Pour une variable aléatoire $X \sim \mathcal{N}(\mu, \sigma^2)$ de fonction de répartition $F_X(x; \mu, \sigma^2)$, on a :*

$$\frac{\partial F_X}{\partial \mu}\left(x; \mu, \sigma^2\right) = -\frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)$$

Démonstration. Comme $X \sim \mathcal{N}(\mu, \sigma^2)$, la fonction de répartition de X vérifie :

$$F_X(x; \mu, \sigma^2) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

La règle de la chaîne permet d'écrire :

$$\begin{aligned}
\frac{\partial F_X}{\partial \mu}\left(x; \mu, \sigma^2\right) &= \frac{\partial}{\partial \mu}\Phi\left(\frac{x-\mu}{\sigma}\right) \\
&= \Phi'\left(\frac{x-\mu}{\sigma}\right) \times \frac{\partial}{\partial \mu}\left(\frac{x-\mu}{\sigma}\right) \\
&= \phi\left(\frac{x-\mu}{\sigma}\right) \times \left(-\frac{1}{\sigma}\right) \\
&= -\frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)
\end{aligned}$$

■

Propriété 2. *Pour une variable aléatoire $X \sim \mathcal{N}(\mu, \sigma^2)$ de fonction de répartition $F_X(x; \mu, \sigma^2)$, on a :*

$$\frac{\partial F_X}{\partial \sigma}\left(x; \mu, \sigma^2\right) = -\left(\frac{x-\mu}{\sigma^2}\right)\phi\left(\frac{x-\mu}{\sigma}\right)$$

Démonstration. Comme $X \sim \mathcal{N}(\mu, \sigma^2)$, la fonction de répartition de X vérifie :

$$F_X(x; \mu, \sigma^2) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

La règle de la chaîne permet d'écrire :

$$\begin{aligned}
\frac{\partial F_X}{\partial \sigma}\left(x; \mu, \sigma^2\right) &= \frac{\partial}{\partial \sigma}\Phi\left(\frac{x-\mu}{\sigma}\right) \\
&= \Phi'\left(\frac{x-\mu}{\sigma}\right) \times \frac{\partial}{\partial \sigma}\left(\frac{x-\mu}{\sigma}\right) \\
&= \phi\left(\frac{x-\mu}{\sigma}\right) \times \left(-\frac{x-\mu}{\sigma^2}\right) \\
&= -\left(\frac{x-\mu}{\sigma^2}\right)\phi\left(\frac{x-\mu}{\sigma}\right)
\end{aligned}$$

■