Éléments de processus stochastiques

Méthodes de Monte Carlo par chaînes de Markov pour la détection de motifs en bioinformatique

Encadrants : Pierre Geurts, Vân Anh Huynh-Thu, Yann Claes

Année académique 2022-2023

Le travail est à réaliser par groupe de **trois** étudiants. Il sera encadré au moyen de séances de travaux dirigés qui auront lieu **les mardis de 10h30 à 12h30 au local R53 du Bâtiment B4**. Seules les questions posées lors de ces séances ou sur le forum de discussion sur Ecampus donneront lieu à des réponses de la part des encadrants. Nous encourageons vivement les étudiants à se documenter eux-mêmes sur les sujets abordés dans ce projet. Toute source utilisée devra évidemment être citée dans le rapport.

Le rapport et le code source sont à remettre via Gradescope pour **le mardi 16 mai 2023 à 22h00** au plus tard.

Contexte général et objectifs

L'objectif général du projet est de développer un algorithme permettant de résoudre le problème de détection de motifs dans des séquences d'ADN. Pour construire ce système, on se basera sur la méthode de Monte Carlo par chaînes de Markov (MCMC pour *Markov chain Monte Carlo* en anglais).

Le projet est divisé en deux parties. La première partie du projet a pour but de vous familiariser avec les chaînes de Markov et la méthode MCMC particulière qui sera utilisée. La deuxième partie, qui constitue le cœur du projet, vise à mettre en œuvre concrètement la méthode MCMC pour résoudre le problème de détection de motifs.

1 Première partie : chaînes de Markov et échantillonnage de Gibbs

La première partie du projet permet de se familiariser avec les chaînes de Markov pour la modélisation de séquences discrètes et de comprendre comment l'échantillonnage de Gibbs fonctionne en l'appliquant sur un problème illustratif simple.

Définitions et notations

Dans cette section, nous fournissons les définitions et notations minimales nécessaires pour la première partie du projet. Nous vous encourageons néanmoins à consulter d'autres références pour obtenir plus de détails.

Chaînes de Markov (d'ordre m). Soit une suite de variables aléatoires $\{X_1, X_2, \ldots, X_t, \ldots\}$. Cette suite définit un modèle (ou une chaîne) de Markov ssi, pour tout $t \geq 1$, la distribution conjointe des t premières variables peut se factoriser comme suit :

$$\mathbb{P}(X_1, X_2, \ldots, X_t) = \mathbb{P}(X_1) \prod_{l=2}^{t} \mathbb{P}(X_l | X_{l-1}).$$

Ce concept peut être généralisé en étendant le conditionnement : la même suite définit un modèle ou une chaîne de Markov d'ordre m ssi, pour tout $t \geq m$:

$$\mathbb{P}(X_t|X_{t-1}, \ldots, X_1) = \mathbb{P}(X_t|X_{t-1}, \ldots, X_{t-m}).$$

Le cas particulier d'un modèle de Markov d'ordre zéro correspond au cas où les variables sont toutes indépendantes :

$$\mathbb{P}(X_1, \ldots, X_t) = \mathbb{P}(X_1) \ldots \mathbb{P}(X_t).$$

Par défaut, un modèle de Markov, sans qualification de son ordre, sera un modèle de Markov d'ordre 1. Dans ce projet, on ne considérera que des chaînes de Markov définies sur des valeurs discrètes.

Méthodes MCMC et échantillonnage de Gibbs. Soit une distribution $\mathbb{P}(Y)$ définie sur une variable aléatoire Y prenant ses valeurs dans un espace $\mathcal{Y}$, de laquelle on souhaite échantillonner. L'idée générale des méthodes MCMC est de construire une chaîne de Markov ergodique dont la distribution invariante est égale à $\mathbb{P}$. Une réalisation de cette chaîne de Markov fournit alors des échantillons de la distribution cible $\mathbb{P}$.

Il existe plusieurs méthodes MCMC pour construire une chaîne de Markov de distribution invariante $\mathbb{P}$. Dans ce projet, on utilisera la méthode d'échantillonnage de Gibbs (voir l'algorithme 1). Cette méthode est utile lorsque la variable aléatoire Y peut se décomposer en une série de N variables aléatoires $Y = (Y_1, \ldots, Y_N)$ (prenant leurs valeurs dans des espaces $\mathcal{Y}_i$ potentiellement distincts) et qu'il est aisé d'échantillonner à partir des distributions conditionnelles $\mathbb{P}(Y_i|Y_1, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_N)$. L'idée de l'algorithme est de modifier à chaque itération une variable Y_i choisie au hasard en tirant sa nouvelle valeur selon la distribution conditionnelle de Y_i.

Plutôt que de tirer une variable aléatoirement à chaque itération, une variante de l'algorithme, dite à *balayage systématique*, consiste à choisir un ordre arbitraire entre les variables et à les parcourir chacune à leur tour de manière cyclique, toujours dans cet ordre, lors des itérations. Par rapport à l'algorithme 1, cette variante remplace le tirage $i \sim U\{1, \ldots, N\}$ par l'instruction $i = t \mod N + 1$. Les deux algorithmes ont des propriétés théoriques similaires. L'algorithme 1 étant plus facile à analyser, c'est celui-ci que nous utiliserons dans la première partie du travail. La version à balayage systématique, qui est plus populaire, pourra cependant être utilisée dans la deuxième partie du travail si vous le souhaitez.

Algorithm 1 Échantillonnage de Gibbs.

Pick a starting value $y^{(0)} = (y_1^{(0)}, \ldots, y_N^{(0)})$.
t = 0
while convergence not reached **do**
  $y^{(t+1)} = y^{(t)}$
  $i \sim U\{1, \ldots, N\}$
  $y_i^{(t+1)} \sim \mathbb{P}(Y_i|Y_1 = y_1^{(t)}, \ldots, Y_{i-1} = y_{i-1}^{(t)}, Y_{i+1} = y_{i+1}^{(t)}, \ldots, Y_N^{(t)})$
  $t = t + 1$
end while

1.1 Chaînes de Markov

L'objectif des questions ci-dessous est de vous faire réfléchir à l'utilisation de chaînes de Markov pour modéliser des séquences discrètes (questions 1 et 5) et de vérifier les propriétés théoriques de processus ergodiques (questions 2, 3, 4).

Fichiers fournis : `seq1.txt`, `seq2.txt` et `seq3.txt`, contenant des séquences d'ADN, c'est-à-dire définies sur un alphabet de 4 lettres $\{A, C, G, T\}$ (voir la section 2.1 pour plus de détails).

Questions :

1. Utilisez la méthode du maximum de vraisemblance pour construire la matrice de transition Q avec $[Q]_{i,j} = \mathbb{P}(X_{t+1} = j | X_t = i)$ (où les états A, C, G, T sont numérotés respectivement 1, 2, 3, 4) d'une chaîne de Markov modélisant la séquence dans `seq1.txt`. Représentez le diagramme d'états de la chaîne correspondante.
2. Sur base de la matrice Q estimée au point précédent, calculez les quantités suivantes pour des valeurs de t croissantes :
   — $\mathbb{P}(X_t = i)$ en supposant que la première lettre est choisie au hasard, i.e. la lettre initiale est tirée dans une loi uniforme discrète.
   — $\mathbb{P}(X_t = i)$ en supposant que la première lettre est toujours C,
   — Q^t, c'est-à-dire la t-ième puissance de la matrice de transition.
   Représentez l'évolution des deux premières grandeurs sur un graphe. Discutez et expliquez les résultats obtenus sur base de la théorie.
3. En déduire la distribution stationnaire [1] π_∞ de la chaine de Markov définie par $[\pi_\infty]_j = \lim_{t\to\infty} \mathbb{P}(X_t = j)$.
4. Générez une réalisation aléatoire de longueur T de la chaîne de Markov en démarrant d'une lettre choisie arbitrairement. Calculez pour chaque lettre le nombre de fois qu'elle apparaît dans la réalisation divisé par la longueur de la réalisation. Observez l'évolution de ces valeurs pour chaque lettre lorsque T croît. Que concluez-vous de cette expérience ? Reliez ce résultat à la théorie.
5. Calculez et comparez les fréquences d'apparition des lettres A, C, G, T dans les trois séquences fournies. Calculez ensuite les probabilités que les trois séquences aient été générées par la chaîne de Markov estimée sur base de la première. Que pouvez-vous conclure de cette expériences ?

1.2 Échantillonnage de Gibbs

L'objectif des questions ci-dessous est de vous faire étudier les propriétés théoriques de l'algorithme et ensuite de l'appliquer à un problème d'échantillonnage simple pour vérifier ces propriétés.

Questions :

1. Soit une chaîne de Markov invariante dans le temps dont la matrice de transition est Q. Montrez que si les équations suivantes (appelées les équations de balance détaillée) sont

1. On utilisera ici stationnaire et invariante de manière interchangeable.

vérifiées pour une distribution π_0 initiale et toute paire d'états $(i, j) \in \{1, \ldots, N\}^2$:

$$\pi_0(i)[Q]_{i,j} = \pi_0(j)[Q]_{j,i},$$

alors π_0 est une distribution invariante de la chaîne de Markov. Dans quel(s) cas celle-ci est-elle unique ?

2. Montrez que la matrice de transition de la chaîne de Markov définie par la méthode d'échantillonnage de Gibbs (Algorithme 1) satisfait aux équations de balance détaillée pour la distribution invariante $\mathbb{P}(Y)$. Quelles autres conditions la chaîne doit-elle respecter pour que l'algorithme fonctionne ?
3. Soit deux variables Y_1 et Y_2 prenant leurs valeurs dans $\{1, 2, 3, 4\}$.
   (a) Choisissez une distribution conjointe sur ces deux variables, sous la forme d'une matrice $Q_Y \in \mathbb{R}^{4\times4}$ telle que $[Q_Y]_{i,j} = \mathbb{P}(Y_1 = i, Y_2 = j)$, de sorte à ce que l'algorithme 1 puisse être utilisé pour échantillonner selon cette distribution. Justifiez votre choix par rapport aux conditions données au point 2.
   (b) Générez une réalisation suffisamment longue de la chaîne correspondant à l'algorithme 1. Comparez dans un graphe les fréquences d'apparition de chaque état (défini par une paire de valeurs pour Y_1 et Y_2) dans cette réalisation avec les valeurs de la matrice Q_Y.
   (c) Répétez l'expérience du point précédent en alternant un échantillonnage de Y_1 puis de Y_2, plutôt qu'en choisissant une variable aléatoirement à chaque itération. Obtenez-vous les mêmes résultats ?

2 Deuxième partie : détection de motifs dans des séquences d'ADN

L'objectif de cette seconde partie est de développer un algorithme pour détecter un motif commun à plusieurs séquences d'ADN. Ce type d'algorithme est un des outils permettant de détecter si un ensemble de gènes sont régulés par le même facteur de transcription et donc partagent une même fonction au sein d'un organisme. De manière simplifiée, une gène est une région de l'ADN qui sera transcrite en ARN messager et ensuite traduit en protéine, ces dernières étant les acteurs essentiels du fonctionnement d'un organisme vivant. Les facteurs de transcription sont des protéines particulières qui vont se coller (souvent en groupe) dans la région en amont du gène sur l'ADN (appelée la région promotrice du gène) et qui vont ainsi activer ou inhiber la transcription de ce gène et donc affecter la quantité de la protéine sous-jacente présente dans la cellule. Le fait qu'un facteur de transcription puisse se coller dans une région promotrice dépend de la structure de la protéine correspondante et de l'occurrence de certains motifs particuliers sur l'ADN. Si un groupe de gènes a été détecté comme étant particulièrement actif ou inhibé dans un phénomène particulier (cellule malade, type cellulaire particulier, étape donnée du développement de l'organisme, etc.), la détection d'un motif dans la région promotrice de ces gènes permet aux biologistes d'identifier les facteurs de transcription potentiellement impliqués dans ce phénomène.

Vu l'importance de la question pour les biologistes, le problème de détection de motifs a fait l'objet de nombreuses recherches et il existe beaucoup d'algorithmes dans la littérature [1]. On s'intéressera ici plus particulièrement à l'approche probabiliste du problème qui met en oeuvre

généralement l'échantillonnage de Gibbs. Cette approche a été proposée à la fin des années 90 et étudiée au début des années 2000 et elle fait maintenant partie des méthodes classiques de la littérature. Les idées développées ci-dessous sont basées sur les références suivantes [2, 3, 5, 4] sur le sujet. Vous êtes libres évidemment au cours du projet de consulter ces sources ou d'autres. Tout document consulté devra évidemment être cité de manière appropriée dans votre rapport.

La section 2.1 décrit les grandes idées de l'approche qu'on vous propose d'étudier et d'implémenter. Cette seconde partie du projet est plus libre que la première mais la section 2.2 établit néanmoins un plan de travail, en quatre étapes, pour atteindre l'objectif visé et donne des indications sur le contenu attendu du rapport.

2.1 Formulation du problème

Notations et contexte. Une séquence d'ADN [2] consiste en une succession de nucléotides pouvant chacun avoir l'une des $K = 4$ valeurs dans $\{A, C, G, T\}$ [3] Dans la suite, on indicera ces valeurs de 1 à 4 dans l'ordre alphabétique A, C, G, T. On développera les algorithmes ci-dessous en ne faisant pas d'hypothèse sur le nombre de "lettres" constituant nos séquences (c'est-à-dire en utilisant K plutôt que directement 4 comme borne de nos sommes par exemple) mais vous pourrez, si vous le souhaitez, vous restreindre dans vos développements et vos codes à des séquences des 4 lettres A, C, G, T.

On supposera disposer d'un ensemble noté $S = \{S_1, \ldots, S_N\}$ de N séquences de nucléotides de longueurs potentiellement différentes. Ces séquences pourront correspondre par exemple aux régions promotrices d'un ensemble de gènes. On notera $|S_k|$ la longueur de la séquence S_k. La longueur du motif sera supposée connue et notée W. Chaque séquence S sera supposée contenir une et une seule occurrence du motif (on supposera que $|S_k| \geq W$ pour tout k). On notera :

- A_k, la variable aléatoire représentant la position du premier nucléotide du motif dans la séquence S_k (avec $A_k \in \{1, \ldots, |S_k| - W + 1\}$),
- $A = \{A_1, \ldots, A_N\}$ les positions des motifs dans toutes les séquences de S,
- $A_{-k} = A \setminus A_k$, les positions des motifs dans toutes les séquences sauf A_k.

Modèles probabilistes. Deux modèles probabilistes de séquences seront considérés, un modèle pour le motif lui-même, dont les paramètres seront notés collectivement $\boldsymbol{\Theta}$, et un modèle pour les séquences en dehors du motif, dont les paramètres seront notés $\boldsymbol{\Phi}$.

Pour ce qui est du motif, on supposera que le nucléotide à la position j ($\forall j = 1, \ldots, W$) est tiré d'une distribution multinomiale de paramètres $\boldsymbol{\theta}_j = (\theta_{1,j}, \ldots, \theta_{K,j})^\top$. Le modèle comporte donc KW paramètres collectés dans la matrice $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_W) \in \mathbb{R}^{K \times W}$. Cette modélisation probabiliste permet une certaine souplesse dans la définition d'un motif, dont les nucléotides à certaines positions pourraient être non définis (dans le cas où la distribution multinomiale serait presque uniforme).

Pour le modèle des séquences hors motif, on pourra utiliser un modèle de Markov d'ordre quelconque. Un modèle d'ordre 0 supposerait que les nucléotides en dehors du motif sont indépendants les uns des autres et tirés d'une même distribution multinomiale définie par K paramètres $\boldsymbol{\Phi} = (\phi_1, \ldots, \phi_K)$. L'utilisation d'un modèle de Markov d'ordre supérieur à 0

2. Acide désoxyribonucléique.
3. A pour Adénine, C pour Cytosine, G pour Guanine, T pour Thymine.

permet de prendre en compte des dépendances spatiales entre nucléotides et permet en général d'améliorer la détection de motifs [4]. Dans ce projet, il ne sera pas utile de considérer des modèles de Markov d'ordre supérieur à 2.

Etant donné ces modèles, on suppose que chaque séquence S_k a été générée de la manière suivante :

- Une position a_k pour le motif a été tirée au hasard uniformément dans $\{1, \ldots, |S_k| - W + 1\}$.
- La séquence entre les positions 1 et $a_k - 1$ est générée selon le modèle $\mathbf{\Phi}$.
- La séquence entre les positions a_k et $a_k + W - 1$ est générée selon le modèle $\mathbf{\Theta}$.
- La séquence entre les positions $a_k + W$ et $|S_k|$ est générée selon le modèle $\mathbf{\Phi}$.

Détection de motif. Etant donné les hypothèses et notations précédentes, le problème de détection de motifs revient à estimer à partir de S l'ensemble A des positions des motifs dans les séquences et les paramètres du modèle de motif $\mathbf{\Theta}$. Les paramètres $\mathbf{\Phi}$ interviennent dans le modèle mais leur estimation n'est pas un objectif en soi.

Dans ce projet, on abordera ce problème comme un problème d'inférence bayesienne en cherchant à calculer la distribution *a posteriori* des paramètres inconnus du modèle, c'est-à-dire :

$$p(\mathbf{\Theta}, \mathbf{\Phi}, A|S).$$

Le calcul de cette distribution étant complexe, on utilisera l'échantillonnage de Gibbs pour obtenir des échantillons de cette distribution, ce qui nous permettra d'effectuer une estimation au maximum (ou en espérance) *a posteriori*.

L'application de cette idée directement serait cependant encore trop couteuse, principalement à cause de l'échantillonnage des paramètres $\mathbf{\Theta}$ et $\mathbf{\Phi}$ (qui suivront une distribution de Dirichlet, voir plus bas). L'approche qu'on vous demande d'étudier et d'implémenter consiste à restreindre l'échantillonnage de Gibbs aux valeurs de positions des motifs en marginalisant la distribution *a posteriori* de manière à en retirer les paramètres $\mathbf{\Theta}$ et $\mathbf{\Phi}$. On calculera ainsi les distributions conditionnelles ($\forall k \in \{1, \ldots, N\}$) :

$$\mathbb{P}(A_k|A_{-k}, S)$$

qui, une fois utilisée dans l'algorithme 1, nous permettront d'échantillonner selon la distribution *a posteriori* des positions $\mathbb{P}(A|S)$ et donc de déterminer la combinaison de positions la plus probable. On pourrait estimer ensuite les paramètres sur base de la distribution $p(\mathbf{\Theta}|A, S)$. Cette version de l'algorithme de Gibbs dans laquelle certains paramètres sont marginalisés s'appelle le mode *collapse* [3].

2.2 Plan de travail

Sont décrites ci-dessous les grandes étapes du travail par lesquelles nous vous suggérons de passer, avec quelques conseils sur la manière d'aborder chacune de ces étapes. Des explications supplémentaires seront données lors des séances encadrées.

2.2.1 Dérivations mathématiques

La première étape consiste à dériver les distributions conditionnelles $P(a_k|A_{-k}, S)$ servant à échantillonner les positions dans l'algorithme d'échantillonnage de Gibbs. Etant donné que

les variables a_k sont discrètes, cette dérivation servira à déterminer un score à associer à chaque position dans $\{1, \ldots, |S_k| - W + 1\}$. Ces scores, une fois normalisés, serviront de paramètres d'une distribution multinomiale pour l'échantillonnage de Gibbs.

Pour arriver à formuler cette distribution, vous pouvez procéder comme suit :

— Calculez d'abord $\mathbb{P}(S, A|\mathbf{\Theta}, \mathbf{\Phi})$ en vous basant sur le modèle de génération de séquences décrit plus haut.

— Utilisez le théorème de Bayes et la formule des probabilités totales pour exprimer $\mathbb{P}(A_k|A_{-k}, S)$ en fonction de $\mathbb{P}(S, A|\mathbf{\Theta}, \mathbf{\Phi})$. Vous aurez besoin pour cela de définir des distributions *a priori* $p(\mathbf{\Theta})$ et $p(\mathbf{\Phi})$. Les paramètres $\mathbf{\Theta}$ et $\mathbf{\Phi}$ étant les paramètres de différentes distributions multinomiales, la distribution naturelle pour ça est la distribution de Dirichlet qui est un "conjugate prior" pour la distribution multinomiale.

— Après simplification, vous devriez au final arriver à formuler $\mathbb{P}(A_k|A_{-k}, S)$ comme le rapport entre la probabilité de la sous-séquence entre A_k et $A_k + W - 1$ selon un modèle $\mathbf{\Theta}$ estimé et la probabilité de la même séquence selon un modèle $\mathbf{\Phi}$ estimé.

— A partir des développements précédents, expliquez finalement comment obtenir une estimation des paramètres $\mathbf{\Theta}$ du modèle de motif lorsque les positions A des motifs sont connues.

Un document fourni sur Ecampus reprend quelques propriétés de la distribution de Dirichlet qui vous seront utiles pour ce développement. Nous vous conseillons de faire la développement dans le cas où le modèle de markov $\mathbf{\Phi}$ est un modèle d'ordre 0 par simplicité et de discuter ensuite de comment la formule finale est modifiée dans le cas d'un modèle de Markov d'ordre supérieur.

2.2.2 Implémentation

Une fois les formules dérivées, vous pouvez implémenter concrètement l'algorithme de recherche de motifs. Cette implémentation peut suivre l'algorithme 1, en utilisant un balayage systématique ou pas.

Une limitation de l'approche cependant, connue dans la littérature, est qu'elle peut se retrouver coincée dans un optimum local correspondant à un décalage constant de toutes les positions par rapport à l'optimum global. En effet, si A est le vecteur contenant les positions réelles de début de motifs, alors $A^{shift} = A + \delta$, où δ est petit, est un optimum local, dont l'algorithme de Gibbs, qui ne modifie qu'une position à la fois, aura du mal à s'échapper. Une manière d'éviter ce problème est de vérifier périodiquement si un petit décalage δ de toutes les positions ne permettrait pas d'augmenter la probabilité $\mathbb{P}(A|S)$. Il sera probablement nécessaire que vous implémentiez cette idée pour obtenir de bonnes performances. Vous pouvez consulter la littérature pour choisir une façon de l'implémenter.

Dans le rapport, on vous demande de décrire les spécificités de votre implémentation, éventuellement en donnant un pseudo-code de l'algorithme. Précisez quelles valeurs estimées $\hat{A}$ et $\hat{\mathbf{\Theta}}$ sont renvoyées par votre algorithme et justifier votre choix.

2.2.3 Expérimentations

Une fois l'algorithme implémenté, il vous est demandé de réaliser des expériences pour étudier son fonctionnement. Vous pouvez utiliser pour cela des séquences générées par vos soin selon le modèle décrit plus haut, ainsi que différents jeux de séquences fournis sur Ecampus.

Critères de performance. Plusieurs critères peuvent être utilisés pour évaluer la qualité des prédictions fournies par votre algorithme [5]. Le premier est la vraisemblance des paramètres estimés, c'est-à-dire $P(S|A, \mathbf{\Theta}, \mathbf{\Phi})$. D'un point de vue numérique, il peut être bénéfique de calculer le logarithme de cette vraisemblance (la log-vraisemblance). Au niveau d'un motif, ce dernier est d'autant plus intéressant qu'il correspond à une séquence bien déterminée de nucléotides. On dira alors qu'il est conservé (sa nature ne change pas d'une séquence à l'autre). Une manière de mesurer la conservation d'un motif est le score de consensus suivant :

$$Score_C(\mathbf{\Theta}) = 2 - \frac{1}{W} \sum_{i=1}^{K} \sum_{j=1}^{W} \theta_{i,j} \log_2(\theta_{i,j}).$$

Ce score sera égal à 2 dans le cas où un seul nucléotide est possible à chaque position et sera égal à 0 si chaque nucléotide a la même probabilité d'apparaître à chaque position. Un critère intéressant consiste à comparer le modèle de motif au modèle de séquence hors motif, en utilisant par exemple la distance de Kullback-Leiber. Par exemple si le modèle de séquence hors motif est un modèle de Markov d'ordre 0 de paramètres $(\phi_{0,1}, \ldots, \phi_{0,K})$, cette distance peut s'écrire :

$$Score_{KL}(\mathbf{\Theta}, \mathbf{\Phi}) = \frac{1}{W} \sum_{i=1}^{K} \sum_{j=1}^{W} \theta_{i,j} \log \left(\frac{\theta_{i,j}}{\phi_{0,i}} \right).$$

Cette distance est toujours positive et elle vaut zéro quand le modèle de motif est identique au modèle de séquence hors motif. Une manière de visualiser un matrice $\mathbf{\Theta}$ est la représentation Logo [4] que vous pouvez également utiliser.

Questions suggérées. Les questions suivantes, entre autres, pourraient être abordées par le biais de vos expériences :

- Obtenez-vous les mêmes résultats quand vous relancez l'algorithme ?
- Quel impact ont les paramètres de l'algorithme (nombre d'itérations, gestion du décalage, degré du modèle de séquence hors motif, paramètres des distributions *a priori*, etc.) sur ses performances ?
- Quel est l'impact des paramètres du problème (nombre et longueur des séquences de S, longueur du motif) sur les performances ?
- Quel est l'effet de l'ordre du modèle de Markov pour les séquences hors motif ?
- Que se passe-t-il si la longueur W utilisée par l'algorithme ne correspond pas à la longueur du vrai motif ? Pouvez-vous trouver une stratégie pour déterminer la longueur du motif dans le cas où elle serait inconnue ?

2.2.4 Compétition

Lors d'une des dernières séances de cours, un ensemble de séquences vous sera fourni et vous devrez appliquer l'algorithme que vous aurez développé en direct lors de la séance pour obtenir la meilleure estimation possible des positions du motif et de sa matrice de paramètres à l'issue de la séance. Le format du fichier de séquences sera le même que pour les séquences fournies sur Ecampus. La longueur exacte du motif ne sera pas fournie mais nous vous donnerons un intervalle de valeurs possibles. La soumission se fera sur Gradescope. Les critères d'évaluation seront précisés ultérieurement mais seront dérivés des mesures présentées dans la section 2.2.3.

4. Voir `https://en.wikipedia.org/wiki/Sequence_logo` et `https://logomaker.readthedocs.io/en/latest/`.

3 Rapport et code

Vous devez nous fournir un rapport au format pdf contenant vos réponses, concises mais précises, aux questions posées ainsi que le code que vous avez utilisés pour y répondre. Le tout est à soumettre sur Gradescope. La longueur attendue du rapport est entre 15 et 30 pages (figures et bibliographie incluse, police de taille 11, pas de code dans le rapport). On s'attend à ce que la partie 2 soit développée plus longuement que la partie 1.

A la fin du rapport, expliquez dans une section séparée la contribution de chacun des membres du groupe au travail. La soumission du rapport suppose que chaque membre du groupe accepte la responsabilité de l'entièreté du contenu rapport.

Vous pouvez réaliser le projet en utilisant Python, Matlab ou C. Si vous souhaitez utiliser un autre langage, demandez au préalable l'accord des encadrants. Quel que soit le langage, l'algorithme de Gibbs doit être implémenté par vos soins. Aucun outil existant solutionnant le problème considéré ne peut être utilisé. En cas de doute, contactez toujours les encadrants.

4 Références

Toutes les références, les données, et les codes relatifs au projet seront disponibles sur Ecampus. Des conseils et réponses aux questions fréquentes seront également rassemblées sur cette page au fur et à mesure. Un forum de discussion a également été ouvert. Veillez à consulter régulièrement Ecampus tout au long du projet.

Références

[1] S. Castellana, T. Biagini, L. Parca, F. Petrizzelli, S. D. Bianco, A. L. Vescovi, M. Carella, and T. Mazza. A comparative benchmark of classic DNA motif discovery tools on synthetic data. *Briefings in Bioinformatics*, 22(6), 08 2021.

[2] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals : a gibbs sampling strategy for multiple alignment. *Science*, 262(5131) :208–214, Oct. 1993.

[3] J. S. Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427) :958–966, Sept. 1994.

[4] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau. A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling. *Bioinformatics*, 17(12) :1113–1122, 12 2001.

[5] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau. A gibbs sampling method to detect over-represented motifs in the upstream regions of co-expressed genes. In *Proceedings of the Fifth Annual International Conference on Computational Biology*, RECOMB '01, page 305–312, New York, NY, USA, 2001. Association for Computing Machinery.