CHNINAK Ishaac
WIEDERKEHR Axel
M2 Bio-informatique

Indexation et requêtage de génomes à l'aide de graphes De Bruijn colorés

1. Introduction

L'ADN est une molécule fondamentale qui porte l'information génétique de tous les êtres vivants. Sa structure, élucidée en 1953 par Watson et Crick, est composée de quatre nucléotides : l'adénine (A), la thymine (T), la cytosine (C) et la guanine (G). L'enchaînement précis de ces nucléotides détermine l'ensemble des caractéristiques génétiques d'un organisme.

La lecture de ces séquences d'ADN a connu une révolution majeure avec l'émergence des technologies de séquençage nouvelle génération (NGS). Ces technologies permettent de déterminer l'ordre exact des nucléotides dans une molécule d'ADN, produisant ainsi des données génomiques en quantité considérable. Cette masse de données génère de nouveaux défis en termes d'analyse de traitement informatique.

La comparaison de séquences génomiques est une tâche centrale en bioinformatique, notamment pour l'identification d'organismes proches, la classification de séquences inconnues ou l'analyse de similarité entre génomes. Les méthodes traditionnelles basées sur l'alignement deviennent rapidement coûteuses lorsque le nombre et la taille des génomes augmentent. Les approches *alignment-free* répondent à cette problématique en s'appuyant sur des représentations plus simples des séquences, en particulier les **k-mers**, sous-chaînes de longueur fixe extraites des génomes.

Les graphes de de Bruijn constituent une structure naturelle pour organiser les k-mers : les nœuds représentent des $(k-1)$-mers et les arêtes correspondent aux k-mers. Lorsqu'on considère plusieurs génomes simultanément, l'extension en **graphe de de Bruijn coloré** permet d'associer à chaque k-mer une information de présence dans un ou plusieurs génomes, appelée couleur. Cette structure devient alors un index efficace permettant, à partir d'une séquence requête, de mesurer sa similarité avec chaque génome de la collection via le ratio de k-mers partagés.

L'objectif de ce projet est d'implémenter un tel index selon deux stratégies : une version naïve simple mais potentiellement redondante, et une version avancée exploitant les **unitigs** afin de réduire la redondance de l'information de couleur. Les performances et les compromis de ces deux approches sont ensuite analysés expérimentalement.

2. Matériel et méthode :

2.1 Données d'entrée

Les données d'entrée consistent en une liste de fichiers FASTA, chacun contenant un génome, ainsi qu'un fichier FASTA contenant une ou plusieurs séquences requêtes.
Chaque génome est associé à un identifiant entier unique, utilisé comme couleur dans le graphe de de Bruijn. Les séquences sont lues en mémoire sous forme de chaînes de caractères, après suppression des en-têtes FASTA.

Les expériences ont été réalisées sur le jeu de données de test fourni avec l'énoncé, comprenant plusieurs génomes bactériens et un ensemble de séquences requêtes. La liste des génomes a été construite à partir des fichiers FASTA fournis, en référençant l'ensemble des génomes utilisés pour l'indexation.

2.2 Extraction des k-mers

Pour chaque génome, l'ensemble des k-mers est extrait à l'aide d'une fenêtre glissante de taille k.
Aucune canonicalisation (reverse-complement) n'est appliquée : les k-mers sont considérés tels quels, ce qui correspond directement à l'implémentation.

La valeur de k a été fixée à **k = 31** pour la validation fonctionnelle de l'indexation. Afin d'analyser l'influence de la taille des k-mers sur les performances, plusieurs valeurs supplémentaires ont été testées (**k = 15, 21, 27 et 41**).

2.3 Version naïve du graphe de de Bruijn coloré

Principe
La version naïve repose sur une structure de type dictionnaire associant à chaque k-mer l'ensemble des génomes dans lesquels il apparaît. La clé du dictionnaire correspond à un k-mer, représenté sous forme de chaîne de caractères, et la valeur associée est un ensemble d'identifiants de génomes (couleurs).

Lors de la construction, chaque k-mer extrait d'un génome entraîne l'ajout de la couleur correspondante à l'ensemble associé à ce k-mer. Cette approche ne réalise aucune factorisation entre k-mers consécutifs : si plusieurs k-mers adjacents apparaissent dans les mêmes génomes, l'information de couleur est stockée de manière redondante.

Complexité
La construction de l'index naïf est linéaire en le nombre total de k-mers extraits. En revanche, l'occupation mémoire peut devenir importante lorsque de longues régions génomiques partagent les mêmes couleurs, en raison de la redondance de l'information stockée.

2.4 Version avancée : compaction en unitigs

Construction du graphe de de Bruijn
La version avancée commence par construire un graphe de de Bruijn explicite à partir des k-mers extraits des génomes. Les nœuds du graphe représentent des $(k-1)$-mers, tandis que les arêtes correspondent aux k-mers. Les relations de successeurs et de prédécesseurs sont stockées séparément. En parallèle, une table associe chaque k-mer à l'ensemble de ses couleurs.

Identification des unitigs
Les unitigs sont construits en parcourant le graphe à partir des nœuds dont le degré entrant ou sortant est différent de 1. À partir de ces nœuds de départ, les chemins linéaires sont étendus tant que les nœuds rencontrés possèdent exactement un prédécesseur et un successeur. Chaque unitig correspond ainsi à une séquence maximale non branchante du graphe. Un dictionnaire associe ensuite chaque k-mer apparaissant dans un unitig à l'identifiant de ce unitig.

Attribution des couleurs aux unitigs
Pour chaque unitig, l'ensemble de ses couleurs est déterminé en agrégeant les couleurs des k-mers qu'il contient. Cette étape repose sur l'hypothèse clé exploitée dans ce projet : les k-mers d'un même unitig partagent une information de couleur identique ou très similaire, ce qui permet de factoriser cette information.

2.5 Sérialisation

Les structures construites sont sérialisées sur disque à l'aide du module pickle.
Dans la version naïve, seul le dictionnaire k-mer → couleurs est stocké.
Dans la version avancée, les données sérialisées incluent :

- La liste des unitigs,
- La liste des ensembles de couleurs par unitig,
- La table de correspondance k-mer → unitig,
- La valeur de *k*.

Les temps de sérialisation et de désérialisation sont mesurés séparément pour chaque valeur de k testée.

2.6 Requêtage et mesure de similarité

Pour chaque séquence requête, l'ensemble de ses k-mers est extrait.
Pour chaque k-mer présent dans l'index, les couleurs associées sont récupérées directement dans la version naïve, ou via l'identifiant du unitig correspondant dans la version avancée.

Un compteur est maintenu pour chaque génome afin de comptabiliser le nombre de k-mers de la requête présents dans ce génome. La similarité entre une requête Q et un génome Gi est définie comme le ratio de k-mers partagés, calculé comme le nombre de k-mers de Q présents dans Gi divisé par le nombre total de k-mers de Q.

Les résultats sont écrits dans un fichier texte, avec un format strictement conforme aux spécifications fournies dans l'énoncé. Les mesures de temps d'exécution et de taille des index ont été réalisées à l'aide d'un script d'automatisation permettant de lancer de manière reproductible les différentes configurations de la taille des k-mers.

3. Résultats

3.1 Validation fonctionnelle de l'indexation

Avant d'analyser les performances des deux implémentations, nous avons vérifié le bon fonctionnement de l'indexation et du requêtage. Les résultats obtenus sont conformes à l'exemple fourni dans l'énoncé du projet.

Pour des requêtes correspondant exactement à l'un des génomes indexés, le ratio de k-mers partagés atteint une valeur de 1.0000 pour le génome correspondant, et 0 pour les autres. Ce comportement valide la construction correcte du graphe de de Bruijn coloré ainsi que le calcul des similarités.

Certaines requêtes présentent toutefois des valeurs intermédiaires de similarité avec plusieurs génomes, traduisant des relations de proximité biologique, comme observé pour la souche O157:H7. Ces observations confirment la pertinence biologique des scores produits par les deux versions de l'index.

3.2 Temps de construction du graphe de de Bruijn coloré

Les temps de construction de l'index ont été mesurés pour plusieurs valeurs de la taille des k-mers (k = 15, 21, 27, 31 et 41), afin d'analyser l'influence de ce paramètre sur les performances.

Comme illustré par la Figure 1, la version naïve présente des temps de construction faibles et relativement stables lorsque k varie, de l'ordre de 0,7 seconde. Cette performance s'explique par la simplicité de la structure utilisée, reposant uniquement sur un dictionnaire associant chaque k-mer à ses couleurs.

À l'inverse, la version avancée nécessite un temps de construction plus élevé, compris entre 4,5 et 4,8 secondes. Ce surcoût est dû à la construction explicite du graphe de de Bruijn, à l'identification des nœuds de départ, puis à la compaction du graphe en unitigs. Cette étape supplémentaire est essentielle pour permettre la factorisation ultérieure de l'information de couleur. Les temps restent néanmoins peu dépendants de la valeur de k, indiquant que le coût principal est lié à la structure du graphe plutôt qu'à la taille exacte des k-mers.

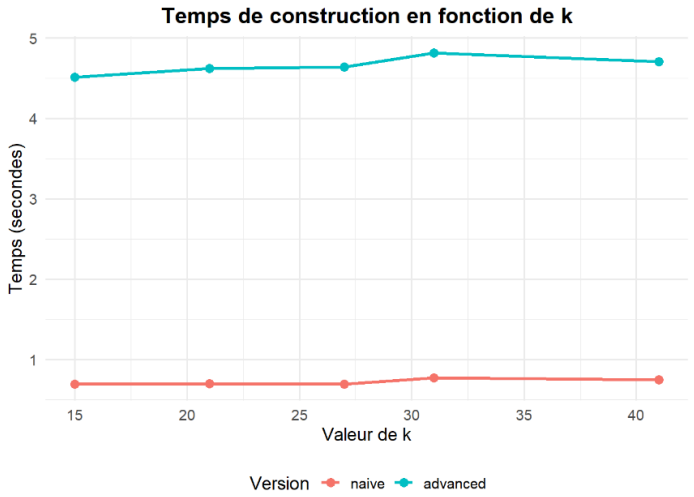

Figure 1 : Temps de construction du graphe de de Bruijn coloré en fonction de k, pour les versions naïve et avancée.

3.3 Temps de sérialisation

La Figure 2 présente l'évolution des temps de sérialisation en fonction de la valeur de k pour les deux implémentations.

La version naïve montre une augmentation progressive du temps de sérialisation lorsque k augmente, passant d'environ 0,44 seconde pour k = 15 à près de 0,68 seconde pour k = 41. Cette augmentation est directement liée au nombre croissant de k-mers distincts stockés individuellement dans la structure.

À l'inverse, la version avancée présente des temps de sérialisation nettement plus faibles et peu dépendants de k, compris entre 0,15 et 0,20 seconde. La réduction du nombre d'objets à sérialiser, permise par la compaction en unitigs, limite fortement les opérations d'écriture sur disque.

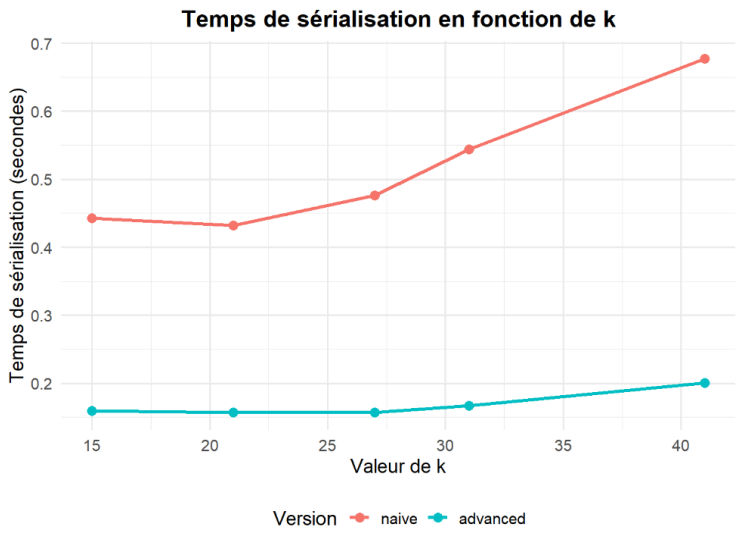

Figure 2 : Temps de sérialisation du graphe de de Bruijn coloré en fonction de k.

3.4 Taille du graphe sérialisé

La taille des fichiers d'index sérialisés constitue un indicateur central de l'efficacité de la compaction. La Figure 3 montre l'évolution de la taille de l'index sur disque en fonction de k.

Pour la version naïve, la taille du fichier augmente quasi linéairement avec k, passant d'environ 10 Mo pour k = 15 à plus de 22 Mo pour k = 41. Cette croissance reflète le stockage explicite de plusieurs centaines de milliers de k-mers distincts, chacun associé à un ensemble de couleurs.

La version avancée produit des index significativement plus compacts. Pour l'ensemble des valeurs de k testées, la taille de l'index reste comprise entre 5 et 12 Mo. La compaction du graphe en unitigs, qui regroupe des k-mers consécutifs partageant des informations de couleur similaires, permet ainsi de réduire la taille de l'index d'un facteur pouvant dépasser 2 à 3 selon la valeur de k.

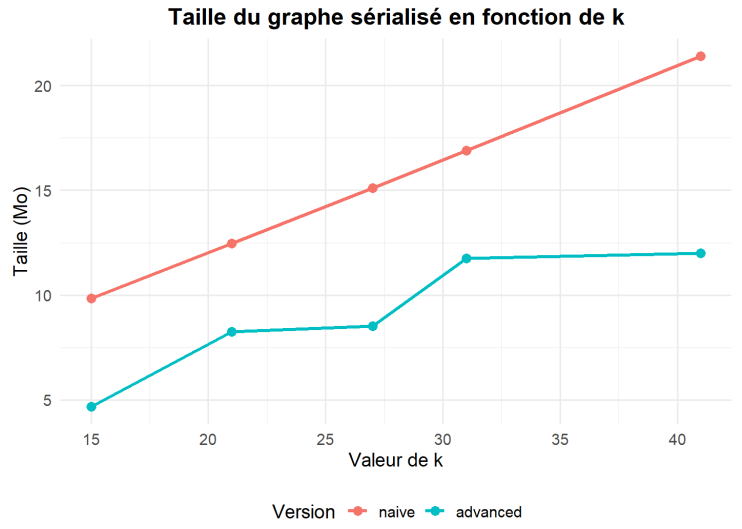

Figure 3 : Taille du graphe sérialisé en fonction de k, comparaison entre les versions naïve et avancée.

3.5 Temps de désérialisation

Les temps de désérialisation, présentés en Figure 4, suivent une tendance similaire à celle observée pour la taille des fichiers et les temps de sérialisation.

La version naïve nécessite entre 0,54 et 0,68 seconde pour recharger l'index en mémoire, avec une légère augmentation lorsque k augmente. En revanche, la version avancée permet un rechargement beaucoup plus rapide, avec des temps compris entre 0,12 et 0,22 seconde.

Cette différence s'explique par la structure plus compacte de l'index avancé, qui réduit à la fois le volume de données à lire sur disque et le coût de reconstruction des objets en mémoire.

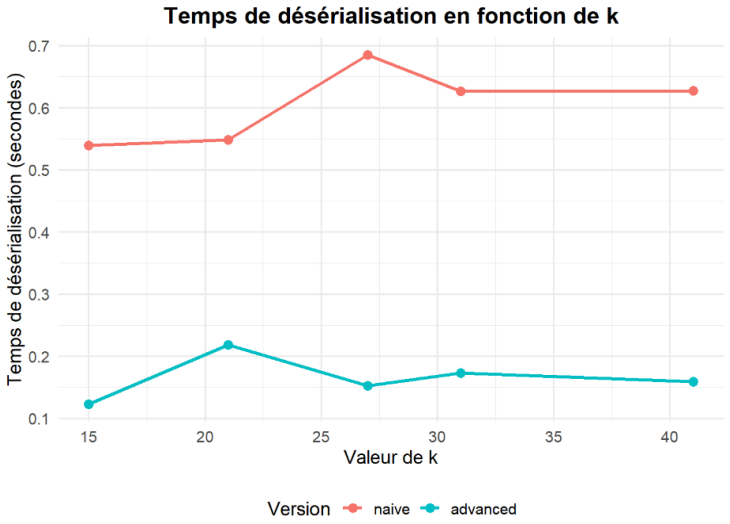

Figure 4 : Temps de désérialisation du graphe de de Bruijn coloré en fonction de k.

3.6 Temps de requêtage

Les temps de requêtage ont été mesurés pour les mêmes séquences requêtes et pour l'ensemble des valeurs de k considérées. Les résultats sont présentés en Figure 5.

Dans les deux versions, les temps de requête restent très faibles, de l'ordre de quelques dizaines de millisecondes, et ne montrent pas de dépendance marquée à la valeur de k. Les performances de la version naïve et de la version avancée sont comparables.

Cette observation indique que l'indirection supplémentaire introduite par l'utilisation des unitigs dans la version avancée n'a qu'un impact marginal sur le temps de requêtage, qui reste dominé par des accès directs à des structures de type dictionnaire.

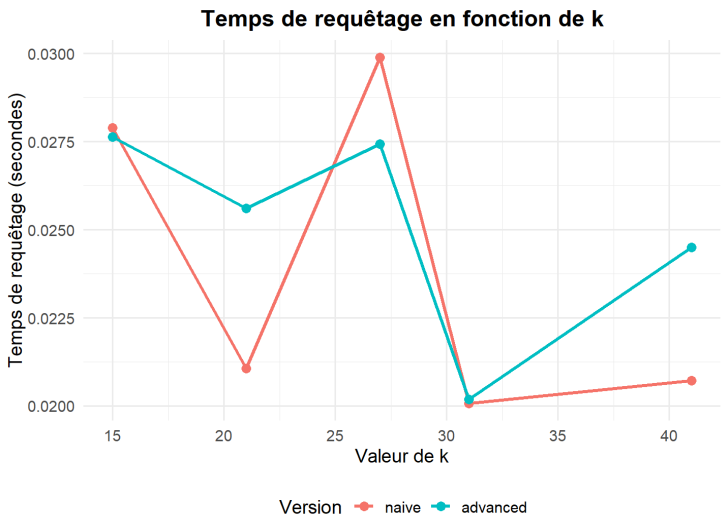

Figure 5 : Temps de requêtage en fonction de k pour les versions naïve et avancée.

3.7 Analyse des similarités entre requêtes et génomes

Les matrices de similarité obtenues mettent en évidence des profils cohérents avec les attentes biologiques. Les requêtes correspondant exactement à un génome de la collection présentent une similarité maximale avec ce génome, tandis que les requêtes issues de fragments composites montrent des ratios intermédiaires, reflétant le partage de k-mers avec plusieurs génomes.

Un extrait représentatif des résultats est présenté dans la Figure 6, illustrant à la fois des cas d'identité exacte et des relations de proximité entre génomes.

Requête	Genome 1	Genome 2	Genome 3	Genome 4	Genome 5
head_B253	1.0000	0.0000	0.0000	0.0000	0.0000
head_O157:H7	0.0000	0.0020	0.0000	0.5098	1.0000

Figure 6 : Matrice des similarités entre requêtes et génomes, basée sur le ratio de k-mers partagés (k = 31).

4. Discussion

Les résultats obtenus mettent clairement en évidence les différences structurelles et algorithmiques entre les deux implémentations du graphe de de Bruijn coloré. La version naïve, basée sur un stockage explicite des k-mers et de leurs couleurs, présente une implémentation simple et efficace en termes de temps de construction. Toutefois, cette simplicité se traduit par une redondance importante de l'information de couleur lorsque de nombreux k-mers consécutifs apparaissent dans les mêmes génomes, ce qui impacte fortement la taille de l'index et les temps de sérialisation et de désérialisation.

À l'inverse, la version avancée exploite la compaction du graphe de de Bruijn en unitigs afin de factoriser l'information de couleur. Cette stratégie entraîne un surcoût lors de la phase de construction, lié à la construction explicite du graphe et à l'identification des chemins non branchants. Néanmoins, ce coût

supplémentaire reste modéré et peu dépendant de la valeur de k, tandis que les gains obtenus en termes de taille mémoire sont significatifs. La réduction de la taille de l'index, pouvant atteindre un facteur supérieur à deux selon la valeur de k, se traduit directement par des temps de sérialisation et de désérialisation nettement plus faibles.

L'analyse de l'influence de la taille des k-mers montre que ce paramètre joue un rôle important dans la structure du graphe. Des valeurs de k plus faibles génèrent un graphe plus dense et plus branché, augmentant le nombre de k-mers distincts et la redondance de l'information. À l'inverse, des valeurs de k plus élevées conduisent à des k-mers plus spécifiques et à des unitigs plus longs, ce qui favorise la compaction du graphe et limite la croissance de la taille de l'index. Ces observations sont cohérentes avec les propriétés théoriques des graphes de de Bruijn.

En revanche, les temps de requêtage restent faibles et comparables entre les deux versions, quelle que soit la valeur de k. L'indirection supplémentaire introduite par l'utilisation des unitigs dans la version avancée n'a qu'un impact marginal sur les performances de requête. Cela montre que la compaction en unitigs permet d'améliorer l'efficacité mémoire sans dégrader la capacité de l'index à répondre rapidement aux requêtes.

Enfin, les matrices de similarité obtenues confirment la pertinence biologique de l'approche alignment-free basée sur les k-mers. Les requêtes correspondant exactement à un génome de la collection sont correctement identifiées, tandis que les requêtes issues de fragments partagés révèlent des relations de proximité entre génomes, illustrant la capacité du graphe de de Bruijn coloré à capturer des similarités génomiques fines.

5. Conclusion

Ce projet a permis d'implémenter et de comparer deux stratégies d'indexation alignment-free basées sur les graphes de de Bruijn colorés. La version naïve offre une implémentation simple et efficace en temps de calcul, mais présente une occupation mémoire importante due à la redondance de l'information de couleur stockée pour chaque k-mer.

La version avancée, fondée sur la compaction du graphe en unitigs, permet de réduire efficacement cette redondance et de produire des index nettement plus compacts, au prix d'un temps de construction plus élevé. Les résultats expérimentaux montrent toutefois que les performances de requêtage restent comparables entre les deux versions, indiquant que l'indirection introduite par l'utilisation des unitigs n'a qu'un impact marginal sur les temps de requête.

L'analyse de l'influence de la taille des k-mers met en évidence le compromis classique entre spécificité des k-mers, structure du graphe et performances de l'index. Ces observations soulignent l'intérêt des graphes de de Bruijn colorés comme structure d'indexation efficace pour la comparaison alignment-free de collections de génomes.