Compte-rendu TP1 AeA

Claire Hunter Antoine Canda

12 février 2017

1 Introduction

Ce projet a pour objectif l'introduction à des méthodes de recherche de motifs répétés dans un texte. Ici nous nous plaçons dans le contexte de la bio-informatique, où le texte est un génome représenté par un fragment de chromosome d'une taille de 100 000 nucléotides. L'objectif du TP est de produire le DotPlot comparant cette chaîne à elle-même, en représentant pour chaque mot de taille N la position de chacune de ses occurrences dans le texte, soit directement sous la forme du mot, soit sous celle de son reverse et/ou de son complémentaire et/ou de son reverse-complémentaire, puisque nous considérons que les mots sont des séquences d'ADN avec ses propriétés propres.

2 Usage

L'archive contient un script shell qui permet de lancer l'exécution de notre travail. Le script produit tous les fichiers de données contenant les points du DotPlot correspondant aux couples de positions des occurrences. Des images PNG sont également générées afin de visualiser les résultats. Nos cas d'étude se trouvent dans le fichier de script et seront détaillés plus loin, dans la partie "Analyse des résultats". Nous avons joint dans l'archive les sources commentées, le jar exécutable, ainsi que quelques images de DotPlot obtenues.

3 Réalisation

Nous commençons par lire le fichier FASTA pour créer la séquence d'ADN grâce à la classe Sequence de notre projet. Nous avons dans la classe DotPlot la méthode qui permet de créer le fichier contenant la position des points pour le DotPlot. Nous avons décidé de représenter la séquence d'ADN (ou ARN) recherchée dans la classe Mot pour tenir compte directement de ses autres formes possibles dans une certaine abstraction de la classe String. Nous avons trois implémentations d'algorithmes de recherche de fait.

La première est une implémentation de l'algorithme Knuth-Morris-Pratt qui n'est pas entièrement fonctionnelle. Nous avons malgré tout pu mieux comprendre son fonctionnement basé sur un pré-traitement du motif qui permet en cas d'erreur de faire un saut intelligent dans la recherche. De plus, après réflexion, cet algorithme n'est pas adapté pour une recherche de plusieurs mots, car il faudrait comparer chaque caractère à une certaine position avec les caractères à cette position de tous les mots de taille N pouvant exister, ce qui est inefficace. Un algorithme de KMP équivalent pour la recherche de plusieurs mots existe (l'algorithme d'Aho-Corasick) mais nous ne l'implémentons pas dans ce TP.

Ensuite nous avons une implémentation naïve de l'algorithme de Karp-Rabin. L'implémentation est naïve dans le sens où l'on doit commencer avec un parcours de la dite séquence pour trouver tous les mots de taille N uniques et que pour chaque mot que l'on considère on parcourt toute la séquence d'ADN présente. Néanmoins le code est fonctionnel et on obtient un fichier .dat pour $n = 10$ en 16 minutes.

Enfin nous avons implémenté une version plus optimisée de l'algorithme de Karp-Rabin où on ne fait qu'un seul parcours de la séquence d'ADN en considérant les mots au fur et à mesure et en utilisant une HashMap basé sur une clé qui est un double représentant la valeur du hash calculé par Karp-Rabin avec en valeur associé un couple (Mot, List<Integer>) qui pour le mot associé contient la position des occurrences. En procédant ainsi, nous obtenons un fichier .dat pour des mots de taille N en 2,5 secondes environ.

4 Analyse des résultats

Nous avons testé notre algorithme sur 3 séquences d'ADN différentes, en utilisant à chaque fois soit le mot obtenu uniquement, soit avec ses 3 autres formes décrites précédemment, et ce pour différentes tailles de n. Les séquences sont le fragment du chromosome 3, arnmessager-1 et un pré-ARN. Dans la suite, nous allons détailler l'analyse avec

le mot unique et le mot avec ses 3 autres formes, mais l'analyse en prenant d'autres formes serait identique.

Pour le chromosome 3, nous avons essayé une petite taille de $n = 5$, car nous avons uniquement 1024 mots différents en les considérant uniques ou bien moins si nous tenons compte des autres formes. Notre hypothèse s'est avérée exacte car nous avons obtenu plus de 50 millions de points en considérant les autres formes ce qui fait un fichier de plus de 600 Mo. Il n'y a quasiment aucune information pertinente à en retirer mais c'était prévisible. Pour une taille $n = 8$ (fig. 1), nous avons toujours des DotPlots assez chargés, mais nous pouvons déjà remarquer clairement une croix blanche centrale avec au niveau de l'intersection une forte densité de points.

De la même façon nous avons souhaité tester avec des tailles de n plus importantes comme 15 (fig. 3). 15 n'est pas si important en soi mais il y a déjà plus de mots de taille 15 possibles (4^{15} uniques) et qu'une séquence ADN peut être particulièrement longue (des centaines de millions de bases) fait qu'il est encore possible d'avoir beaucoup de doublons. Le résultat a été très parlant : nous n'obtenons quasiment que la diagonale dans le DotPlot, soit pratiquement que des mots uniques. Nous avons donc dans notre cas une taille trop élevée pour pouvoir en tirer des conclusions biologiques, si ce n'est qu'il n'y a pas eu de fragments d'ADN de taille 15 qui ont par exemple lors d'une mutation été dupliqués et fusionnés à d'autres endroits causant des motifs répétés sous différentes formes.

Quand nous regardons les DotPlots pour n=10, notamment celui où nous considérons les différentes formes de chaque mot (fig. 2), nous observons une zone plus dense des nucléotides aux positions entre 10 000 et 20 000, entre 30 000 et 40 000 et aux alentours des positions 80 000. Le centre de cette "croix" forme aussi une concentration extrêmement importante de points. Nous la retrouvons sur plusieurs autres DotPlots. Une hypothèse que nous posons est que l'on est autour d'une zone codant une fonction essentielle, voire vitale et que toute mutation aux alentours de cette zone entraîne la mort ce qui expliquerait le fait qu'elle soit préservée, on ne la retrouve d'ailleurs peu en dehors de cette zone.

Pour la séquence de l'arnmessager-1 nous observons deux zones pour n=5 qui ressortent distinctement du DotPlot : celle des 750 premiers nucléotides et celle des 2 500 derniers. D'ailleurs dans l'arnmessager-1 dans le DotPlot arn1-n5-7.png, vers les nucléotides 3 500, nous retrouvons ce motif en croix très clairement. Il a donc une signification biologique particulière. Pour $n = 10$, nous n'avons quasiment plus aucune information sauf cette petite concentration au niveau du centre de la croix vers les nucléotides 3 500 : nous avons donc un motif de taille au moins 10 répétée dans cette zone.

Enfin, quand nous regardons le pré-ARN, nous ne trouvons que peu d'informations car avec une taille faible de mot (n=5) nous n'avons quasiment que des mots uniques. De plus, si nous prenons plus petit nous avons très peu de mots possibles.

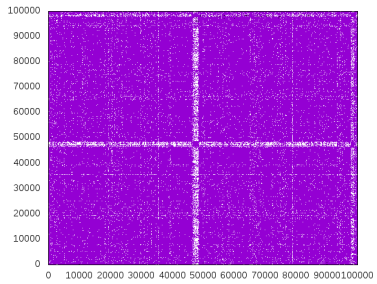

FIGURE 1 – DotPlot chrom 3, n=8

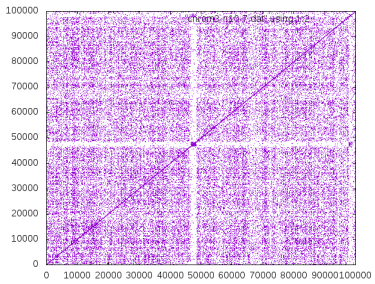

FIGURE 2 – DotPlot chrom 3, n=10

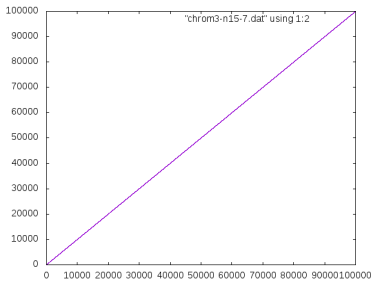

FIGURE 3 – DotPlot chrom 3, n=15

5 Conclusion

Ce projet a été l'occasion de découvrir de nouvelles méthodes de recherche de motif ainsi que la découverte des DotPlots avec gnuplot, qui est un outil particulièrement intéressant pour visualiser des résultats. Nous avons pu durant ce projet réfléchir à des solutions techniques pour optimiser un minimum notre code pour avoir des résultats plus rapides. Nous avons compris ce besoin au vu de la taille importante des données manipulées dans le domaine de la bio-informatique et dans d'autres domaines de l'algorithmique du texte.

Nous avons trouvé que des mots de petites tailles ou au contraire de grandes tailles n'étaient pas pertinents pour faire des comparaisons car la structure même de l'ADN composé de 4 nucléotides différents (Adénine, Thymine ou Uracile, Cytosine et Guanine) fait que nous avons jusqu'à 4^n mots différents possibles pour un n donné. Toutefois, ce n sera différent en fonction de la taille de la séquence que l'on étudie. En effet, sur une séquence relativement courte comme le fragment de taille 100 000 que nous avons étudié, pour n valant 10 nous avons pu avoir des résultats intéressants à analyser alors que pour n valant 15 ce n'était plus aussi vrai. Sur des séquences plus importantes, peut-être que considérer des mots de taille 20 ou 30 sera pertinent.