TARIS Raphael
DEPIERRE David
Master 2 BioInformatique
2016/2017

Encapsulation d'un programme dans l'interface GALAXY
Projet n°7 – Assemblage de lecture en contig et identifiaction de la lecture majoritaire

Objectif : Le But de ce projet est d'intégrer à l'interface galaxy un outil permettant à partir d'un fichier d'alignement BAM de recenser les contigs et d'en determiner la lecture majoritaire.

Dans un contexte de sequencage d'ARN non codant, l'utilité d'u tel outil serait de determiner avec le plus de précisions les loci de ces portions d'ARN. En effet la lecture majoritaire dans un contig est la sequence qui a été le lue lors du sequencage, c'est donc celle qui represente le mieux et avec le plus de sensibilité la portion du genome qui a été exprimée en tant que ncRNA.

Le traitement des données se fera en deux etapes. Une premiere qui consistera à aligner les reads sur le genome de reference et de proceder à un nettoyage jusqu'a l'obtention du fichier au format BAM. La deuxième étape sera la recherche des contigs et des loci des lectures majoritaires.

Etape 1 – alignement

Données disponibles :

un fichier de lecture de sequencage au format fasta
http://genoweb.toulouse.inra.fr/~gaspin/cleaned_sequences.fa
un genome de reference au format fasta, compressé en .gz
http://genoweb.toulouse.inra.fr/~gaspin/genome.fa.gz

Nous allons utiliser l'aligner Bowtie2, et les fonctions du programme Samtools afin de filtrer et trier les lectures alignées. Les programmes cités ci-dessus seront appelés via le script shell

Le pipeline utilisé pour l'alignement est le suivant :

- Décompression du fasta du genome de reference
- Indexation du genome de reference (Bowtie2-build)
- Alignement des lectures du fastq sur le genome indexé, récuperation des reads uniquement alignées (Bowtie2-align, option -k 1)
- Filtrage des reads alignés (Samtools view) sur critère de score d'alignement
- Trie des reads alignés et filtrés (Samtools sort)

En sortie de ce pipeline nous avons un fichier d'alignement au format BAM que nous pouvons utiliser dans la deuxième etape du processus. Le script fasta2BAM.sh appelle les commandes bowtie et samtools necessaire pour cet alignement, son code est adapté pour être lancé en ligne de commande sur la plateforme genotoul, cependant il peut être facilement adapté en modifiant les chemin d'accès vers les programmes. Notons que nous n'avons pas intégrer ce script à galaxy, considérant que notre outil intégré à Galaxy prend en entrée un fichier BAM.

Etape 2 – Generation des loci et identification de la sequence majoritaire

Une fois l'alignement effectué, la tâche consiste à detecter les contigs et determiner pour chaque contig la lecture majoritaire. Un contig est un ensemble de lecture (read) qui se chevauchent lors de l'alignement et forment ensemble une sequence continue. Nous définissons la lecture majoritaire comme la plus longue sequence commune avec une occurrence maximale entre tous les reads d'un même contig. Ainsi si sur un même contig on retrouve plusieurs sequences dont le nombre d'occurence est égal et maximal, alors la plus longue sera considerée comme lecture majoritaire. Une fois les positions début et fin des loci des contigs et de leur lecture majoritaire determinée, nous les stockons sous forme de tableau dans un fichier tsv ou tabular

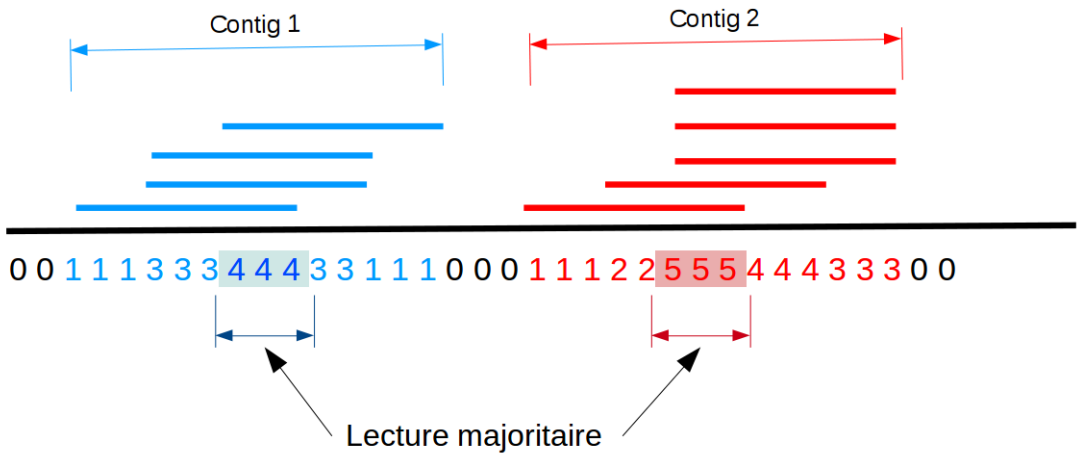


Figure 1 : Explication graphique de la définition du contig et du read.

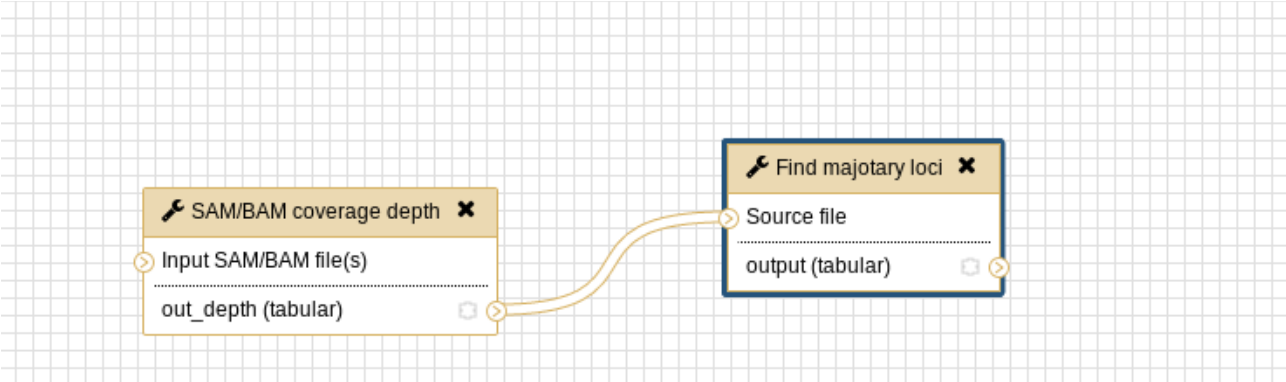
Pour ce faire, nous décidons de passer par un première étape de pré-traitement du BAM afin d'obtenir la profondeur de sequencage. Utilisation de samtools depth pour obtenir un fichier qui nous donne pour chaque position (paire de base) sur le genome de reference le nombre de reads qui ont mappé usr cette position, chaque ligne correspond a :
« ID_chromosme position nbr_reads ».

C'est à partir de ce fichier que nous allons pouvoir determiner les contigs et les lectures majoritaires selon l'algorithme suivant :

Pour chaque ligne:
Initialisation du contig pour la première ligne du fichier.
Si Position précédente +1 == Position actuelle :
    Si dans locus :
        Si Nombre de read actuel > Meilleur nombre de read :
            Définition nouveau meilleur locus.
        Si Nombre de read actuel < Meilleur nombre de read :
            Fin du locus
    Sinon:
        Si Nombre de read actuel > Meilleur nombre de read :
            Définition nouveau meilleur locus.
        Si Nombre de read actuel == Meilleur nombre de read :
            Taille du locus +1
            Si Taille du locus > Taille du meilleur locus :
            Définition nouveau meilleur locus.
Sinon :
    Ecrire dans fichier output le contig

Encapsulation de l'outil :

Pour encapsuler cet outils dans galaxy il faut avoir deux fichier, le premier est notre outil python, le deuxième est notre XML, permettant d'afficher et d'encapsuler dans une iinterface galaxy notre outil. Pour que l'outil fonctionne il faut d'abord faire passer notre fichier BAM par l'outil samtools depth (disponible sur le toolshed ainsi que dans le fichier tools associé à ce Rapport). Les deux outils sont indépendants, mais notre outil est basé sur l'architecture d'un fichier depth. Il faudra l'utiliser comme montré ci dessous :



Discussion :

En théorie cet outil devrait fonctionner sur tous les types de données issues de séquencage à haut débit. Nous pouvons très bien imaginer son application dans un contexte de ChIP-seq pour determininer avec le plus d'acuité le locus lié par une proteine ou la portion de séquence cible reconnu par un facteur trans-acting. Il devient ensuite trivial de recuperer la séquence avec l'identifiant du chromosome et les positions start et stop sur le genome de reference indexé grâce par exemple à la commande samtool faidx <refgenome.fa> <chr:start-stop>.

Nous supposons cependant que passer par le coverage n'est certainement pas la manière la plus direct d'arriver à nos fins et qu'il aurait été possible de traiter directement le BAM, grâce à des commande incluse dans bedtools ou samtools comme mpileup. Traiter directement le BAM nous aurait certainement permis de recuperer l'orientation du brin (+/-), information que nous perdons en passant par la profondeur.

Cependant nous pouvons adresser comme critique à notre programme le faite de me pas ponderer les différents candidats pour la lecture principale par leur taille et de les choisir de façon trop naïve. En effet, si 12 reads se chevauchent sur 3 paires de bases, cette séquence va devenir la lecture majoritaire, potentiellement devant une sequence de 20 nucléotides où 11 reads se chevauchent. Pour pallier à ce problème, une solution serait de pondérer les candidats à la lecture majoritaire par la taille de la séquence candidate car selon l'utilisation à venir du résultats, il serait peut être plus interressant d'avoir une lecture majoritaire d'une taille plus grande mais moins représentée plutot qu'une lecture majoritaire de 3 paires de bases.