Laboratoire de Bioinformatique pour la Génomique et la Biodiversités

Master de bioinformatique - ingénierie de plate-forme en biologie

Univesité de Paris

Rapport d'alternance

Gestion informatique des données de séquençage

---

27 janvier 2022

William Amory

sous la resposabilité de Frédérick Gavory

Table des matières

1 Introduction

1.1 LBGB au sein du Genoscope et du CEA

Le Genoscope (CNS [1]) a été créé en 1996 pour participer au projet mondial de séquençage du génome humain (*Human Genome Project*) qui à débuté en 1988 et c'est terminé en 2007, notament dans l'objectif de séquencer le chromosomes 14 humain. Lors de sa cration le Genoscope à égalment été missioné de développer des programmes de génomiques en france dans le cadre du projet France génomique. Aujourd'hui un des projet phare du Genoscope est le projet **Tara Océans**, qui a pour objectifs l'étude des écosystèmes marins planctoniques.

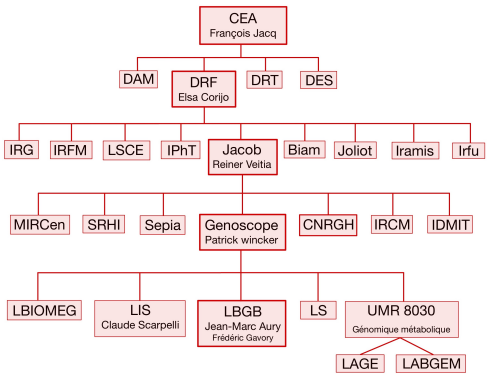

FIGURE 1 – Organigramme situant l'équipe du Laboratoire de Bioinformatique pour la Génomique et la Biodiversité (LBGB) au sein du Genoscope et du CEA

Le laboratoire de bioinformatique pour la génomique et la biodiversité (**LBGB**) dirigé par Jean-Marc Aury, fait partie du Genoscope qui est une direction de l'institut François Jacob (Jacob) de la direction de la recherche fondamental (**DRF**) du Commissariat à l'énergie atomique et aux énergies (**CEA**), comme on peut l'observé sur l'organigramme de la figure **??** qui situe la laboratoire au sein du Genoscope et de CEA. L'intégration du génoscope au CEA à été réaliser en 2007 et en 2017 il dvient une direction de l'institut François Jacob.

1.2 Contexte et missions du LBGB

Les missions qui sont confié au LBGB est de réaliser de la vielle technologique, de réaliser le contrôle qualité des fichiers de séquences issue des différents séquenceurs. Il à également la missions de réaliser l'assemblage et l'annotation des séquences et des génomes, tout en faisant de la visualisation pour chacune de ces missions. Le Laboratoire est divisé en plusieurs groupes de travails. Le groupe production (dont je fais parti), le groupe assemblage, le groupe d'annotation et le groupe d'évaluation des technologies de séquençage. Les misssions du groupe de production sont de réaliser de la vielle technologique, d'évaluer de nouveux outils, de développer, tester et maintenir les scripts dans l'objectif de répondre aux besoins des équipes de recherche et de séquençage. Notament dans la mise en place et au maintient de pipelines automatiques pour la génération des fichiers de séquences, le contrôle qualité et les analyses biologiques de ces derniers.

1. Centre National de Séquençage

2 Ojectifs

Les objectifs de ma mission est la mise en place d'un workflow pour les séquenceurs de la marque MGI dû à l'aquisition de 2 DNBSEQ-G400 et 1 DNBSEQ-T7 par le Genoscope.

FIGURE 2 – Sequenceurs DNBSEQ-G400 (à gauche) et DNBSEQ-T7 (à droite) de MGI https://en.mgi-tech.com/products/

Il s'agit de séquenceurs à haut débit équavalent à un HiSeq 4000 pour le DNBSEQ-G400 et à un NovaSeq 6000 pour le DNBSEQ-T7 de chez ILLUMINA. L'objectif sera de créer un worflow de pipelines similaire à celui présenté ci-dessous.

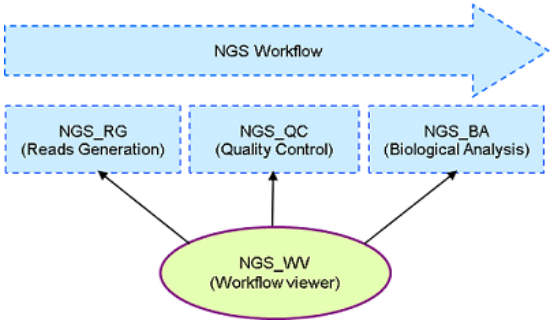

FIGURE 3 – Workflow de génération, de controle qualité et d'analyse biologique des FASTQ

Plus précissément il s'agira de créer un pipelines de génération de reads (NGS_RG-MGI) et un pour le contrôle qualité (NGS_QC-MGI). C'est deux étapes sont la condition pour que les fichiers de séquences brut soit tranféré sur bande magnétique et effacé de la base de données de références (NGL).

Ces pipelines ont pour tâches de réaliser la distribution des fichiers FASTQ par projet, de les trier par échantillons, runs et technologies. Ainsi que de réaliser le nettoyage et l'analyse des fichiers FASTQ, tout en mettant à jour la base de données NGL.

Les autres objetifs de ma mission est de réaliser des évaluations d'outils utilisé et de nouveaux outils pour les différents pipelines en place pour les différentes technologies de séquençage. Tel que l'évaluation d'outils de génération de fichier FASTQ et de démultiplexage, des outils de *flitering*, *trimming*, ect. Ainsi que de maintenir les pipelines des différentes technologies en ajoutant, remplaçant, ou stopant certains outils suite à une évaluation de ces derniers ; notament pour le workflow d'ILLUMINA et celui de MGI une fois ce dernier créé.

3 Matériels et Méthodes

L'écriture du workflow de pipelines pour les séquenceurs MGI sera réaliser dans le langage de programmation Perl. L'utilisation de ce langage est fait pour des raison historique du laboratoire, puisque de nombreuses librairies et modules qui seront à utiliser dans l'écriture des pipelines sont écrit en Perl. le workflow pour MGI s'appuirera sur le workflow d'Illumina qui est totalement implémenté en Perl. C'est pour toutes ses raisons qu'il m'a été nécessaire d'apprendre à coder en Perl.

Une première évaluation d'outils à également été effectué. Il s'agit de comparer les performance de deux logiciels de génération de fichiers FASTQ ainsi que leurs démultiplexage. Il s'agit donc de comparer les performance entre le logiciel utilisé actuelment au Genoscope, qui est bcl2fastq, avec le logiciel bcl-convert qui sont tous deux developper et commercialisé par Illumina.
Dans un premier temps il sera necessaire de déterminer la meilleur combinaison de paramètre pour bcl2fastq, pour pouvoir appliquer les mêmes paramètres pour bcl-convert.
Les paramètres en question sont :

- `r` : nombre de *threads* accordé pour la décompréssion et la lecture des *Bases Calls*
- `p` : conversion des *Bases Calls* en FASTQ
- `w` : écriture et compréssion des fichier FASTQ

Les paramètres èquivalent pour le logiciel bcl-convert sont

- `bcl-num-decompression-threads` : nombre de *threads* accordé pour la décompréssion et la lecture des *Bases Calls*
- `bcl-num-conversion-threads` : conversion des *Bases Calls* en FASTQ
- `bcl-num-compression-threads` : écriture et compréssion des fichier FASTQ

Tous ces tests ont été réalisés sur le même noeud de calcul, dans l'objectif de minimiser les biais que pourrait produire de faire une comparaison sur des resultats provenant de noeud de calcul différent. LA comparaison est effactué sur le temps total pour la génération des FASTQ et le démultiplexage, ainsi que le temps cpu et le pourcentage d'utilisation des cpu.

4 1er Résultats

Suite à l'apprentissage du Perl en réalisant un programme permettant de faire des analyses statistiques élémentaire sur des fichiers FASTQ. Tel que le taux de GC, la moyenne du score de la qualité, ainsi que plusieurs autres métriques. Le programme est capable de gérer les fichiers FASTQ issue de séquençage *single end* et *paired end*. Cela m'a permis de prendre en main les modules utilisé pour les différents pipelines déja en place,

notament pour le pipelines d'Illumina. À la suite de cette prise en main de l'environement de travail, l'utilisation du lancement de job sur les noeuds de calculs, l'apprentissage du Perl et l'utilisation des modules utilisé pour le workflow d'Illumina. On a réaliser une comparaison entre deux logiciels de génération de FASTQ et de démultiplexage.

4.1 Détermination des meilleurs paramètres pour bcl2fastq

Après avoir effectué différentes combinaisons des paramètres, nous avons mis en évidence que la variation du paramètre `r` et `w` en fixant le paramètre `p`, n'apportait pas de différences significatives. Nous avons donc fait varier les paramètres `p`, `r` et `w` de manière à ce que chacun des paramètre soit égale au nombre de coeurs accordé aux deux logiciels.

4.2 Comparaison entre bcl2fastq et bcl-convert

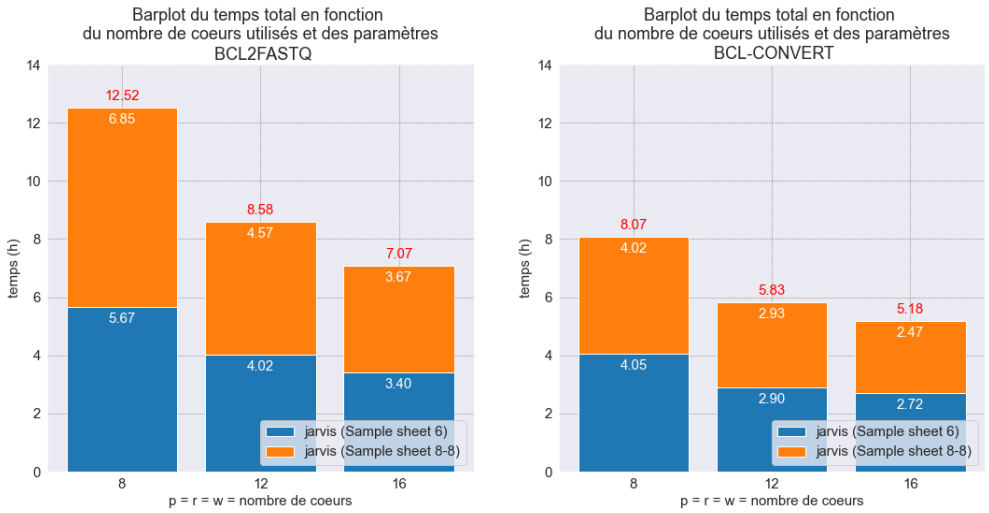

Figure 4 – Temps total de génération des FASTQ pour bcl2fastq et bcl-convert

La figure **??**, montre la différence de temps total des deux logiciels. Il y a deux *sample sheet*, car le nombre de bases considérés des *reads index* entre les *lanes* est différents, obligeant à réaliser deux appelle différents au logiciels pour générer les FASTQ et le démultiplexage. On oserve bien que plus on augmente le nombre de coeurs pour chacun des logiciels plus la génération des FASTQ et le démultiplexage est rapide. De plus on remarque que bcl-convert permet de réduire le temps d'environs 1/3 par rapport à bcl2fatq.

Il existe un autre paramètre pour bcl-convert qui permet de spécifier le nombre de têche à effectuer en parralèles qui n'existe pas pour bcl2fatq. Il reste donc à détermier si en utilisant ce paramètre et en le faisant varier, cela permetterait de réduire le temps de génération des FASTQ et le démultiplexage.

5 Perspectives

Concernant les perspective issue de cette comparaison de logiciels, il reste à déterminer les sorties que textscbcl-convert qui sont différentes de BCL2FATQ et de réaliser une adaptation du workflow d'ILLUMINA en concéquence.

Concernant le workflow de MGI, il nous faut dans un premier temps déterminer les outils et méthodes necessaires. Notament, déterminer quels sont les outils et méthodes que l'on ré-utilisera du workflow ILLUMINA, des nouveaux outils et méthodes à utiliser ou à développer. Une fois ceci déterminer il restera à écrire les deux pipeline, celui de génération de reads (NGS_RG-MGI) et celui de contrôle qualité (NGS_QC-MGI). L'objectif sur le long terme est d'arriver à un workflow totalment automatisé, comme celui ILLUMINA.

Il y a aussi l'évaluation d'autres outils utiles pour les pipelines, comme l'évaluation d'outils de *trimming*, *filtering*, d'assignation taxonomique, ect.

5.1 diagramme de gantt