Laboratoire de Bioinformatique pour la Génomique et la Biodiversité

Master de bioinformatique - ingénierie de plate-forme en biologie

Université de Paris

Rapport d'alternance

Gestion informatique des données de séquençage

---

21 juillet 2022

William Amory

sous la responsabilité 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 1990 et s'est terminé en 2003. Il a notament participé au séquençage du chromosome 14. Le Genoscope impliqué dans le développement de programme de génomique en France dans le cadre du projet France génomique. Aujourd'hui un des projets phare du Genoscope est le projet **Tara**, qui a pour objectifs l'étude des écosystèmes marins.

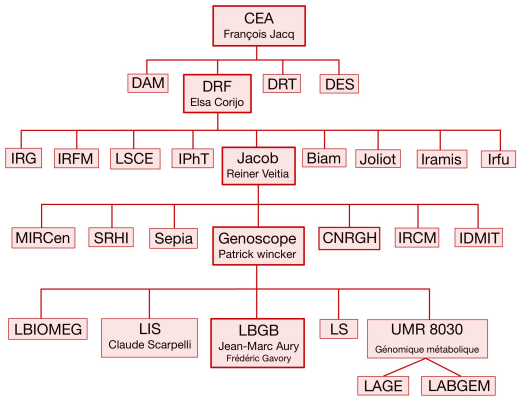

FIGURE 1 – Organigramme situant l'équipe du LBGB au sein du Genoscope et du CEA

Le Laboratoire de Bioinformatique pour la Génomique et la Biodiversité (**LBGB**) dirigé par Jean-Marc Aury, fait partie du Genoscope qui est une composante de l'institut François Jacob (**IBFJ**) de la direction de la recherche fondamentale (**DRF**) du Commissariat à l'Énergie Atomique et aux Énergies Alternatives (**CEA**), qui a été fondé le 18 octobre 1945 par Charles de Gaulle. L'intégration du genoscope au CEA a été réalisée en 2007, et en 2017 il devient une composante de l'IBFJ.

1.2 Contexte et missions du LBGB

Les missions qui sont confiées au LBGB sont : réaliser le contrôle qualité des données de séquences issues des différents séquenceurs, d'effectuer l'assemblage[2] des séquences et l'annotation[3] des génomes, dans l'objectif de mettre à disposition des laboratoires collaborateurs les données avec un premiers niveau da valorisation. Le laboratoire est divisé en plusieurs groupes de travail. Le groupe *production* (dont je fais parti), le groupe *assemblage*, le groupe *annotation* et le groupe *évaluation des technologies de séquençage*.

Les missions du groupe de *production* sont de tester des logiciels tiers, de développer et maintenir des scripts utilisant ces logiciels pour gérer efficacement la prise en charge des données en sortie de séquençeur. Cette prise en charge peut répondre à une demande de la production, mais aussi d'autres laboratoires. L'objectif principale est la mise en place et le maintient de pipelines automatisant l'ensemble. Le groupe s'appuie sur un travail de vielle et d'évaluation technologique pour chacune de ses missions.

1.3 Présentation du workflow NGS

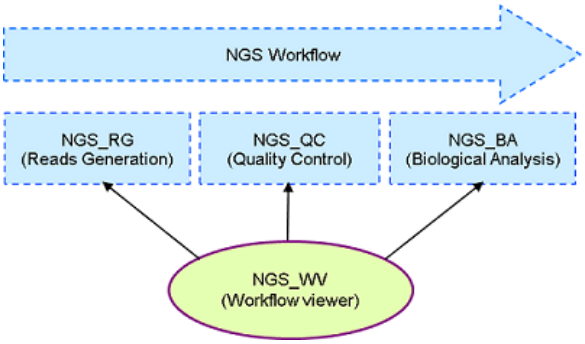

FIGURE 2 – Workflow de génération, de contrôle qualité et d'analyse biologique des fastq

Le workflow ngs[4] est composé de trois pipelines pour les technologies Illumina, Oxford Nanopore, PacBio. Le premier (ngs_rg[5]), permet la génération des reads[6]. Le second (ngs_qc[7]), permet de réaliser le contrôle qualité des fichiers de séquence. Le dernier (ngs_ba[8]), permet de faire les analyses biologiques de lot de séquence[9]. Ces trois pipelines sont automatisés dans le workflow et permmettent de réaliser la distribution des données de séquences par projet, de les trier par échantillons, runs[10] et technologies de séquençage. Ils réalisent aussi le nettoyage, l'analyse de ces fichiers et mettent à jour la base de données de référence ngl[11].

1.4 La technologie MGI

Le genoscope et le CNRGH[12] ont récement fait l'aquisition de séquenceurs MGI[13] (2 DNBSEQ-G400 et 1 DNBSEQ-T7).

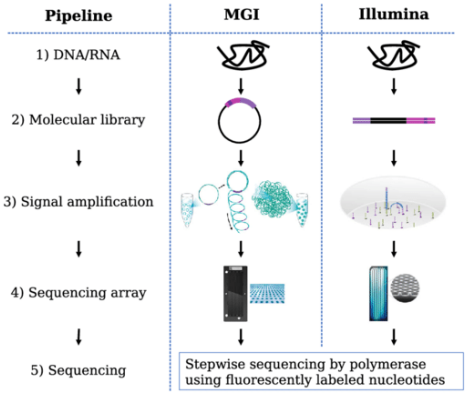

FIGURE 3 – Différences entre Illumina et MGI de technologie NGS

Il s'agit de séquenceurs à haut débit, dont les principales différences entre MGI et Illumina sont dans la création des librairies et la méthode d'amplification d'ADN. Les librairies sont double brins circulaire pour MGI, alors que pour Illumina elle est double brins linéaire. L'amplification ADN est réalisée en solution et forme des *DNA-nanoballs*[14] pour MGI puis déposée sur la Flowcell[15], alors que pour Illumina elle est réalisée après immobilisation sur les Flowcell.

	DNBSEQ-G400	DNBSEQ-T7	HiSeq 4000	NovaSeq 6000
Max Number of Flow Cells	2	4	2	2
Max Lane/Flow Cell	4	1	4	4
Run Time	∼ 14-37 h	∼ 20-30 h	∼ 24-84 h	∼ 13-44 h
Data ouput/Run	0.27-1.4 Tb	1-6 Tb	0.9-1.8 Tb	1-6 Tb
Max Reads/Run	1.8 billions	5 billions	10 billions	20 billions
Max Read Length	2 × 200 bp	2 × 150 bp	2 × 150 bp	2 × 250 bp

TABLE 1 – Spécification des séquenceurs

2 Objectifs

L'objectif principal de ma mission est la mise en place d'un workflow pour les séquenceurs de MGI. Plus précissément il s'agira de créer un pipeline de génération de reads (ngs_rg_mgi[16]) et un pour le contrôle qualité (ngs_qc_mgi[17]). Le workflow devra créer et mettre à jour l'état des runs et des *lanes*[18] dans ngl, réaliser le contrôle qualité des fichiers de séquence, au format fastq, obtenu après démultiplexage[19] des runs. Il devra mettre à jour l'avancement du traitement d'un run dans ngl, en y indiquant les statistiques obtenues lors du démultiplexage, les résultats des contrôles qualités, etc. Puisque l'objectif est d'obtenir un premier niveau de valorisation des fichier de séquences, permettant aux autres groupes (*assemblage*, *anotation*) de prendre en charge ces fichiers avant de les mettrent à disposition des laboratoires collaborateurs.

Je dois également, rechercher et réaliser des évaluations de nouveaux outils pour les différents pipelines des différentes technologies de séquençage. En vue d'un potentiel ajout d'outils ou de remplacement d'outils. Il sera donc necessaire de maintenir les pipelines des différentes technologies de séquençage en conséquence.

3 Matériels et Méthodes

3.1 Le cluster de calcul et Slurm

Le Genoscope possède 10 noeuds de calculs pour la *production* sur le nouveau cluster *inti*, ces derniers disposent de 16 cœurs et de 257 Go de RAM[20](mémoire vive). L'accès à l'utilisation des clusters est réalisé par le logiciel Slurm[21].

3.2 La base de données de référence ngl et la gestion des projets

Le Genoscope dispose de sa propre base de données de référence ngl. Celle-ci est divisée en plusieurs parties. ngl_bi[22], est la partie de la base de données utilisée par les équipes de bioinformatique. ngl_seq[23], est la partie de la base de données utilisée dès la réception des échantillons et jusqu'au séquençage de ces derniers. Il y a également les parties ngl_sub[24], ngl_reagent[25] et ngl_projects[26]. La gestion et le suivi du développement informatique sont réalisés par le système de tickets Jira[27].

3.3 Le langage de programmation Perl

L'écriture du workflow des pipelines pour les séquenceurs MGI sera réalisée dans le langage de programmation Perl. L'utilisation de ce langage est rendu necessaire pour des raisons historique du laboratoire, puisque de nombreuses librairies et modules qui seront à utiliser dans l'écriture des pipelines sont écrits en Perl[28].

C'est pour toutes ces raisons qu'il m'a été nécessaire d'apprendre à coder en Perl. j'ai donc commencé par réaliser un programme permettant de faire des analyses statistiques élémentaires 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*[29] et *paired end*[30]. Cela m'a permis de prendre en main les librairies Perl utilisées pour les différents pipelines déja en place. Ainsi que de m'habituer à l'environement de travail, l'utilisation du lancement de job sur les noeuds de calculs et l'utilisation des modules[31] pour les différents pipelines.

3.4 Évaluation de bcl2fastq et bcl-convert

Une première évaluation de deux logiciels de génération de fichiers fastq et de démultiplexage, développés et commercialisés par Illumina a également été efféctuée. Cette évaluation a été nécessaire pour déterminer les changements qu'il y aura à faire dans les pipelines en vue du remplacement de bcl2fastq (qui sera bientôt obsolète) par bcl-convert.

Dans un premier temps, il a été nécessaire de déterminer les conditions optimales de bcl2fastq (temps total, temps cpu[32], pourcentage d'utilisation cpu) en fonction des ressources disponibles sur les noeuds du cluster (*inti*) réservé à la *prodution*, avec l'objectif de pouvoir appliquer les mêmes conditions à bcl-convert. Les conditions optimales sont déterminées en fonction des paramètres suivants de bcl2fatq (l'équivalent de bcl-convert est indiqué entre crochets) :

- `r` [bcl-num-decompression-threads] : nombre de *threads*[33] accordé pour la décompréssion et la lecture des *Bases Calls*[34]
- `p` [bcl-num-conversion-threads] : conversion des *Bases Calls* en fastq
- `w` [bcl-num-compression-threads] : écriture et compréssion des fichiers fastq

Tous ces tests ont été réalisés sur le même noeud de calcul, dans l'objectif de minimiser les biais. La comparaison est effectuée sur le temps total de génération des fastq et le démultiplexage, ainsi que le temps cpu et le pourcentage d'utilisation des cpu.

4 Résultats

4.1 Résultats des évaluations de bcl2fastq et bcl-convert

4.1.1 Détermination des meilleurs paramètres pour bcl2fastq

Après avoir effectué différentes combinaisons des paramètres, il a été mis en évidence que la variation du paramètre `r` et `w` en fixant le paramètre `p`, n'apportait pas de différences significatives pour le temps total d'exécution, le temps cpu ou le pourcentage d'utilisation cpu, comme on peut l'observer sur la figure 4, pour p fixé à 12. Des resultats similaires ont été obtenus pour p égale à 4, 8 et 16.

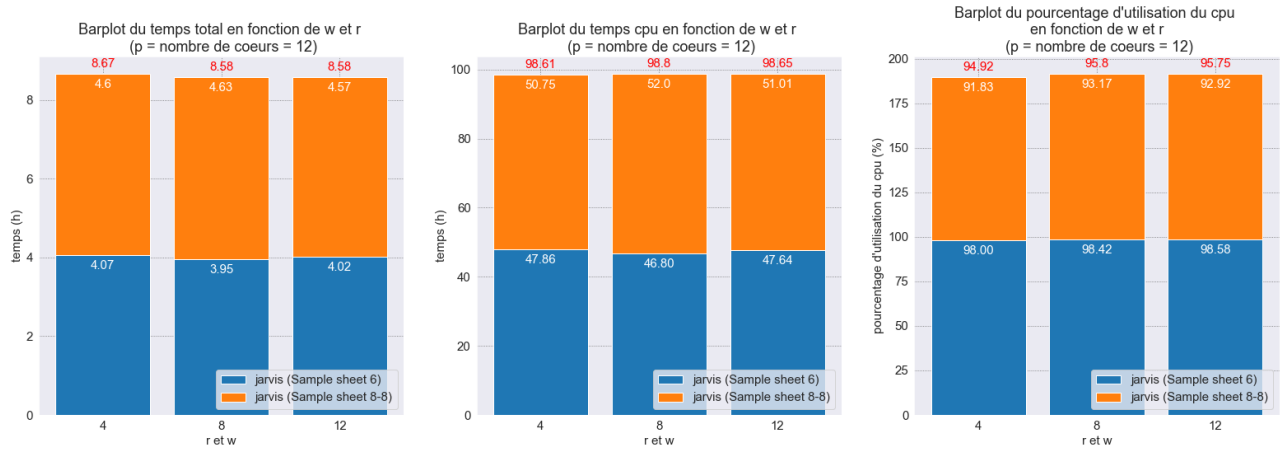

FIGURE 4 – Digrammes en bâtons du temps total d'éxécution (à gauche), temps cpu (au milieu) et du pourcentage d'utilisation des cpu (à droite) en fonction des paramètres r et w

Il y a deux *sample sheet*[35], car le nombre de bases considérés des *reads index* entre les *lanes* est différent, obligeant à réaliser deux appels différents au logiciel pour générer les fastq et le démultiplexage. Ci dessous, la figure 5, représente les résultats obtenus en faisant varier p et en fixant les paramètres r et w à 4 (ces deux paramètres sont fixés à 4 pour pouvoir comparer les 4 résultats). On observe que plus on augmente le nombre de cours et le nombre de *threads* pour p, plus l'execution est rapide. On observe que le temps cpu augmente bien avec le nombre de cœurs et que le pourcentage d'utilisation des cpu est optimal ($> 90\%$).

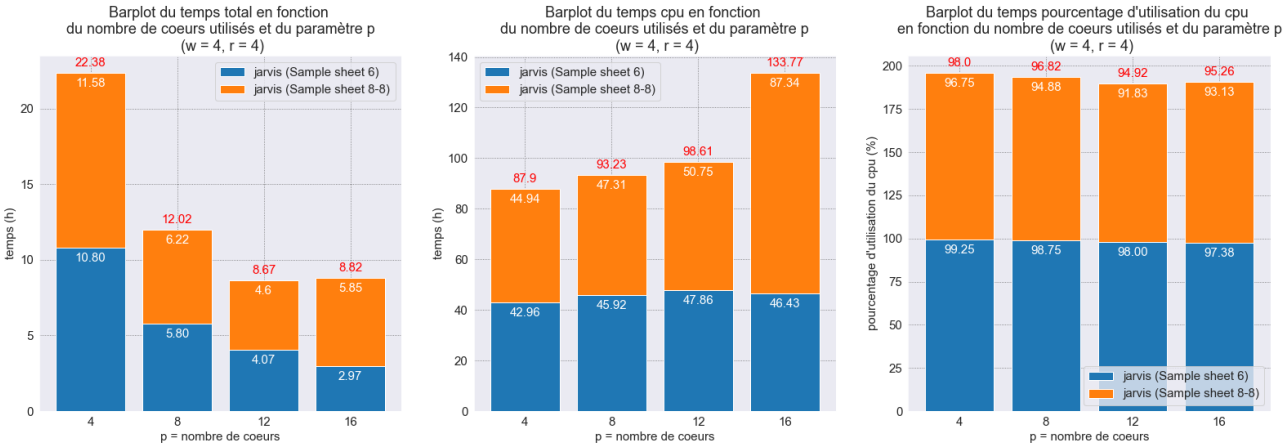

FIGURE 5 – Digrammes en bâtons du temps total d'éxécution (à gauche), temps cpu (au milieu) et du pourcentage d'utilisation des cpu (à droite) en fonction du paramètre p

Au vue des résultats obtenus nous avons décidé que les meilleurs paramètres étaient de fixer p à 12, puisque le gain apporté en augmentant à 16 est faible. Néanmoins nous le conserverons pour réaliser la comparaison avec bcl-convert, tout comme p fixé à 8, car il nous permettrait de réaliser deux générations de fastq et de démultiplexage en simultané sur un seul noeud de calcul.

4.1.2 Comparaison entre bcl2fastq et bcl-convert

J'ai donc fait varier les paramètres `p`, `r` et `w` de manière à ce que chacun des paramètre soient égale au nombre de cœurs accordés aux deux logiciels. On observe bien, sur la figure 6, que plus on augmente le nombre de cœurs pour chacun des logiciels (et donc le nombre de *threads* pour `p`, `r` et `w`) 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.

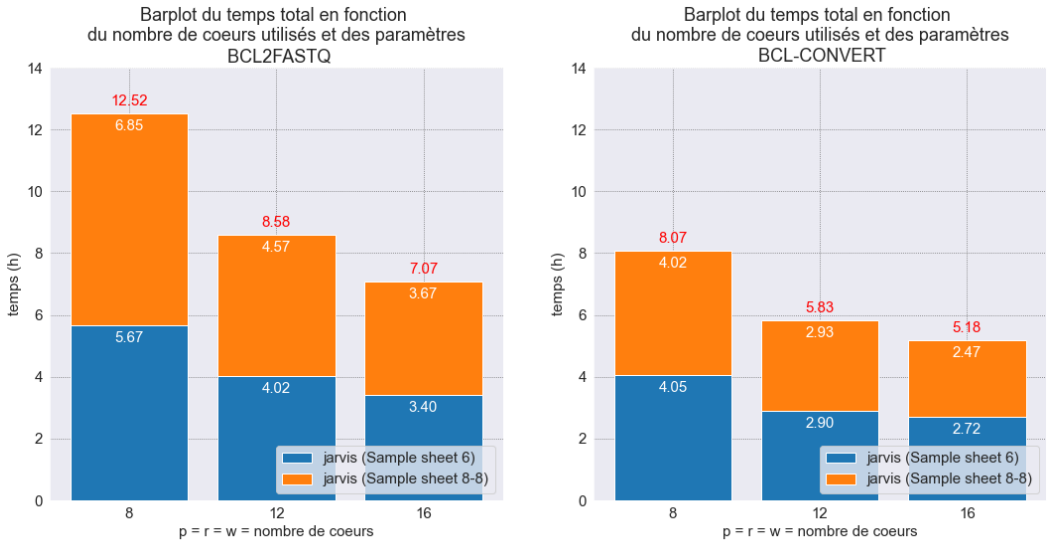

FIGURE 6 – Temps total de génération des fastq pour bcl2fastq et bcl-convert

J'ai également échangé avec le service technique d'Illumina à propos des fichiers de sortie et de l'arborescence de bcl-convert. En effet il s'avère que l'arborescence et les fichiers de sortie sont très différents entre les deux logiciels. Ces échange avaient pour objectif de savoir s'il on pouvait obtenir une arboresnce similaire à bcl2fastq, pour minimiser l'impact du changement de logiciel sur les pipelines. Le changement de bcl2fstq, qui sera bientôt obsolète, par bcl-convert va nous obliger à réaliser de gros changements dans tous les pipelines qui utilisent ces fichiers de sortie et va demander aussi au laboratoire de séquençage de s'adapter à la nouvelle *sample sheet* de bcl-convert.

5 Perspectives

5.1 bcl-convert

Concernant les perspectives de bcl-convert il reste à réaliser un cahier des charges référençant tous les changements à effectuer dans les différents pipelines pour la mise à jour de bcl2fastq vers bcl-convert. Ce cahier des charges prendra en compte le changement d'arborescence des fichiers de sortie entre les deux logiciels, ainsi que toutes les modifications à effectuer dans les différents pipelines (commande de lancement de bcl-convert, module à chargé dans l'environement, path[36] des fichiers de sortie) pour permettre le bon fonctionnement des workflows. Dû à

la pression actuelle autour de la technologie MGI, c'est un autre développeur qui se chargera de suivre ce cahier des charges et de réaliser ces modifications.

5.2 Workflow MGI

Concernant le workflow de MGI, il nous faut dans un premier temps déterminer les outils et méthodes necessaires (utilisation de ceux du workflow d'Illumina ou de nouveaux). Une fois ceci déterminé il restera à écrire les deux pipelines, celui de génération de reads (ngs_rg_mgi) et celui de contrôle qualité (ngs_rg_mgi). L'objectif sur le long terme est d'arriver à un workflow totalment automatisé, comme celui d'Illumina.

5.3 Évaluation d'outils

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

5.4 diagramme de gantt

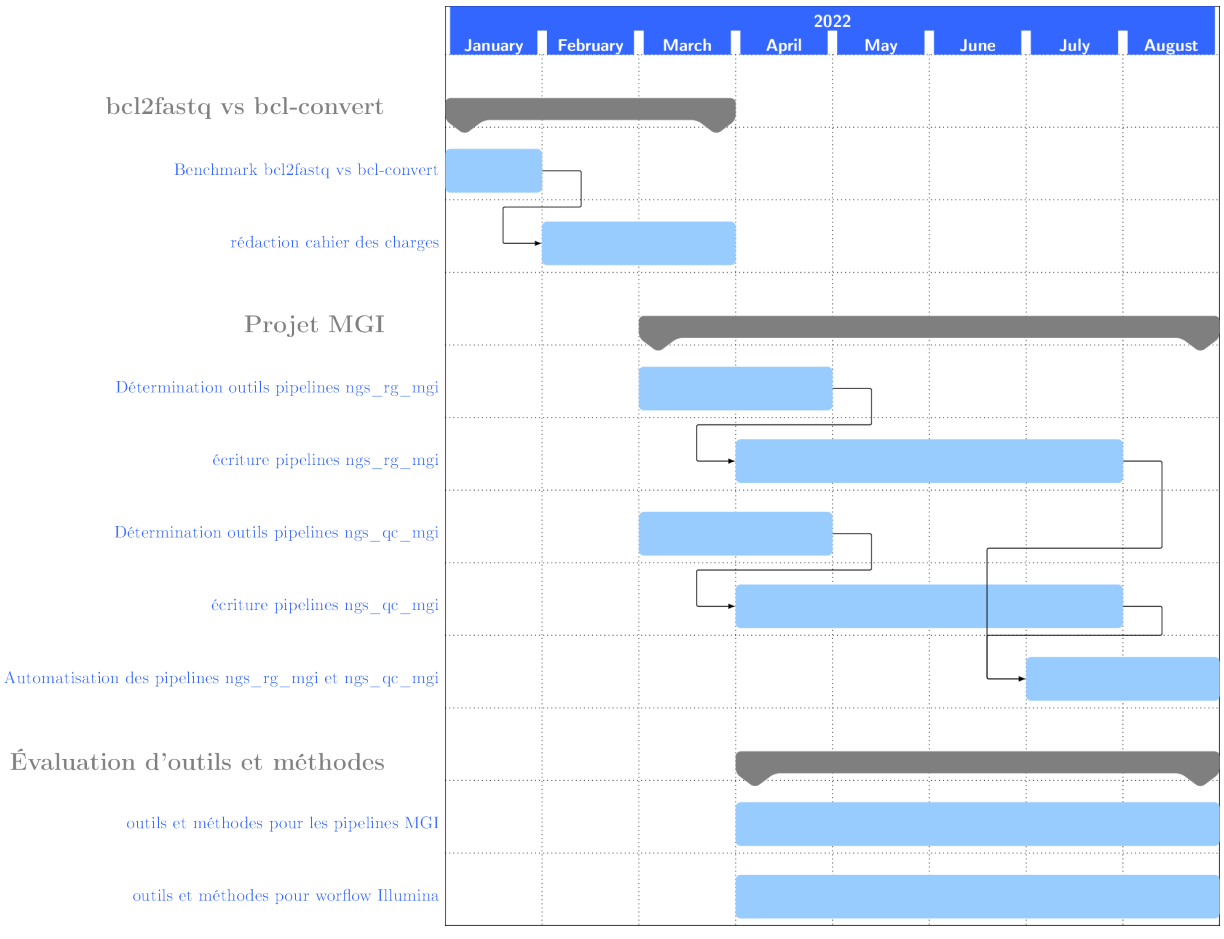

FIGURE 7 – Digramme de gantt des différents projets et missions

Notes

[1]Centre National de Séquençage

[2]Reconstruction d'un génome à partir fragment de ce dernier

[3]Documenter le plus exhaustivement possible les informations de l'assemblage permmettant de prédire la fonction d'une molécule

[4]*Next Generation Sequencing*

[5]*Next Generation Sequencing - reads generation*

[6]Lecture d'une séquence par un séquenceur d'un fragments d'ADN

[7]*Next Generation Sequencing - quality control*

[8]*Next Generation Sequencing - biological analysis*

[9]Un lot de séquences est une instance de séquences (ou reads) d'un échantillon

[10]Séquençage d'un ou plusieurs échantillons sur un séquenceur

[11]*Next Generation LIMS (Laboratory Information Management System)*

[12]Centre National de Recherche en Génétique Humaine

[13]Filiale du groupe BGI dont les missions sont : R&D, production et vente d'instruments de séquençage d'ADN, de réactifs et de produits connexes

[14]Nanobilles d'ADN générées par la réplication de l'ADN circulaire

[15]Lame d'absorbtion des fragments d'ADN et cuve réacteur du séquençage

[16]*Next Generation Sequencing - reads generation - mgi*

[17]*Next Generation Sequencing - quality control - mgi*

[18]pistes présentes sur la *flow cell*

[19]Séparation des différents *reads* d'une *lane* en fonction de l'index d'échantillon

[20]Random Access Memory

[21]Logiciel open source d'ordonnancement des tâches informatiques

[22]ngl bioinformatic

[23]ngl sequencing

[24]ngl submission (base de données des job soumis aux clusters)

[25]ngl reagent (base de données des réactifs)

[26]ngl projects (base de données des projets en cours et passé)

[27]Logiciel de gestion de projet, d'incidents et de suivi de bugs

[28]Pratical Extraction and Report Language

[29]Lecture dans un seul sens des reads par le séquenceur

[30]Lecture dans les deux sens des reads par le séquenceur

[31]Un module contient un ou plusieurs logiciels tiers ou dévellopé par les équipes du genoscope. Il est néccessaire de les charger dans notre environement de travail pour pouvoir utiliser ces logiciels.

[32]Central Processing Unit (unité centrale de traitement, en français)

[33]Processus : instructions du langage machine d'un processeur.

[34]Fichier d'attribution des bases nucléiques en fonction des pics du chromatogramme lors du séquençage

[35]Fichier contenant les informations et instructions pour la génération des fastq et le démultiplexage

[36]Chemin d'accès à un fichier ou à un répertoire dans le système de fichiers