Laboratoire de Bioinformatique pour la Génomique et la Biodiversité

Master de bioinformatique - ingénierie de plate-forme en biologie

Université Paris Cité

Rapport d'alternance

Gestion informatique des données de séquençage

---

31 août 2022

William Amory

sous la responsabilité de Frédérick Gavory

Table des matières

Glossaire

BGI : *Beijing Genomics Institute*, est une entreprise Chinoise de biotechnologie fondé en 1999.

CEA : Commissariat à l'Énergie Atomique et aux Énergies Alternatives

CNRGH : Centre National de Recherche en Génomique Humaine

CNS : Centre National de Séquençage (Genoscope)

CPU : *Central Processing Unit* (Unité Central de Traitement)

DRF : Direction de la Recherche Fondamentale

ERGA *European Reference Genome Atlas*

IBFJ : Institut de Biologie François Jacob

Illumina : Entreprise Californienne de biotechnologie fondée en 1998, qui réalise : R&D, production et vente d'instruments de séquençage d'ADN à haut débit et très haut débit, ainsi que des logicels et services d'anlyses bio-informatique des données de séquençage.

Jira : Logiciel de gestion de projet, de suivi d'incidents et de bugs développé par l'entreprise Atlassian

LBGB : Laboratoire de Bioinformatique pour la Génomique et la Biodiversité

Lims : *Laboratory Information Management System*

MGI : Filiale du groupe BGI fondée en 2016 dont les missions sont : R&D, production et vente d'instruments de séquençage d'ADN, de réactifs et de produits connexes

NCBI : *National Center for Biotechnologiy Information*, est un institut national des Etats Unis d'Amériques pour l'information biologique moléculaire. Il dévellope notament la base de données de génomes GenBank et la base de données des publications PubMed

NGL : *Next Generation LIMS*

NGS : *Next Generation Sequencing*

Oxford Nanopore : Entreprise Anglaise de biotechnologie fondée en 2005, qui dévellope et produit des systèmes de séquençage, basé sur les propriété diélectrique de ces dernières.

PacBio : *Pacific Biosciences of California* est une entreprise Californienne fondée en 2004, qui dévellope et produit des systèmes de séquençage en temps réel à molécule unique (SMRT) d'ADN

Path : Chemin d'accès à un fichier ou à un répertoire dans le système de fichier

Perl : *Pratical Extraction and Report Language*

Ram : *Random Access Memory* (Accès Mémoire Aléatoire, aussi appelé mémoire vive)

Slurm : *Simple Linux Utility for Resource Management* qui est un logiciel open source d'ordonnancement des tâches informatiques

1 Introduction

1.1 Le LBGB au sein du Genoscope et du CEA

Le Genoscope (CNS) 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 est impliqué dans le développement de programme de génomique en France dans le cadre du projet France génomique. Aujourd'hui les projets phares du Genoscope sont les projets **Tara** (*Pacific*, Océans, *Artic* ...), qui ont pour objectifs l'étude des écosystèmes marins ; Le projet **ERGA**, dont l'objectif est de créer une base de données de références de haute qualité des génomes d'espèces européennes.

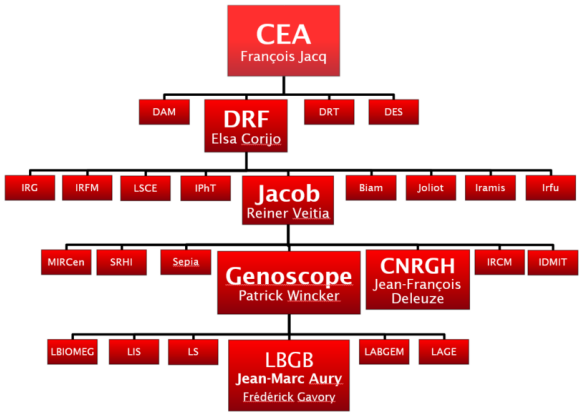


FIGURE 1 – Organigramme situant l'équipe du LBGB au sein du Genoscope et du CEA

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 de biologie 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 de réaliser le contrôle qualité des données de séquences issues des différents séquenceurs, d'effectuer l'assemblage[1] des séquences et l'annotation[2] des génomes, dans l'objectif de mettre à disposition des laboratoires collaborateurs les données avec un premier niveau da valorisation. Le laboratoire est divisé en plusieurs groupes de travail. Le groupe « production » (dont je fais partie), 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, ainsi que développer et maintenir des scripts utilisant ces logiciels pour automatiser la prise en charge des données en sortie de séquençeur. Cette prise en charge peut répondre à une demande de la production et des laboratoires du Genoscope et du CNRGH, mais aussi pour des laboratoires extérieurs. L'objectif principal est la mise en place et le main-

tient de pipelines automatisant l'ensemble. Le groupe s'appuie sur un travail de veille et d'évaluation technologique pour chacune de ses missions.

1.3 Présentation du workflow NGS

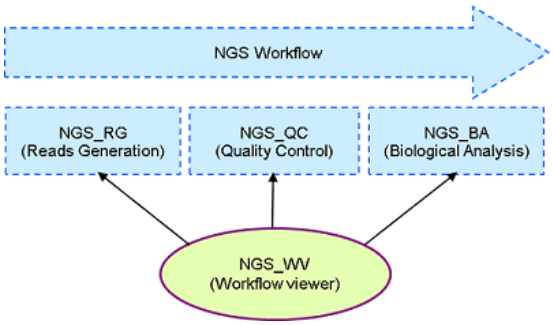


FIGURE 2 – Workflow de génération, de contrôle qualité et d'analyse biologique des fastq

Le workflow NGS est composé de trois pipelines pour les technologies Illumina et Oxford Nanopore. Le premier (ngs_rg[3]), permet la génération des reads[4] et des fichiers de séquences correspondants aux échantillons. Le second (ngs_qc[5]), permet de réaliser leur contrôle qualité. Le dernier (ngs_ba[6]), permet de faire les analyses biologiques inter-échantillons (*readset*)[7].

Ces trois pipelines sont automatisés dans le workflow et permettent de réaliser la distribution des données de séquençage dans des répertoires dédiés, triées par projet, échantillon, runs[8] et technologie de séquençage. Ils réalisent aussi le nettoyage, l'analyse de ces fichiers et mettent à jour la base de données de référence NGL. Les trois pipelines du workflow NGS sont monitorés par NGS *Workflow Viewer* (NGS_WV), qui est une application web permettant de surveiller l'avancement des pipelines pour les runs pris en charge par le NGS-workflow.

1.4 La technologie MGI

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

FIGURE 3 – Sequenceurs DNBSEQ-G400 (en haut) et DNBT7 (en bas) de MGI https://en.mgi-tech.com/products/

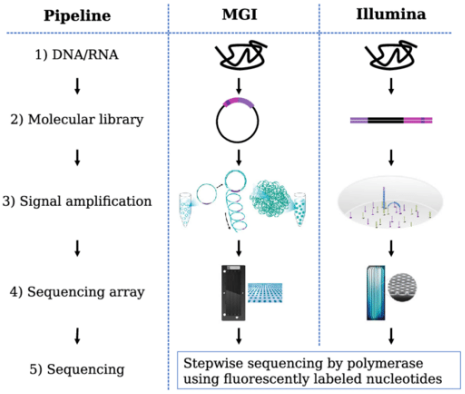


FIGURE 4 – Différences entre Illumina et MGI de technologie NGS

Il s'agit de séquenceurs à haut débit et très haut débit, dont les principales différences entre MGI et Illumina sont dans la création des librairies[9] 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 DNB (*DNA-nanoballs*[10]), puis déposée sur la Flowcell[11] pour MGI, alors que pour Illumina elle est réalisée après immobilisation sur les Flowcell.

Sequencers specifications				
	MGI		Illumina	
	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 de ma misssion

L'objectif principal de ma mission est la mise en place d'un workflow NGS pour les séquenceurs de MGI. Plus précissément il s'agira de créer un pipeline de génération de fichiers de séquences (ngs_rg_mgi[12]) et un pour le contrôle qualité de ces fichiers (ngs_qc_mgi[13]). Le workflow devra créer et mettre à jour l'état des runs, des *lanes*[14] et de *readset*[15] dans NGL, réaliser le contrôle qualité des fichiers de séquences, au format fastq, obtenus après démultiplexage[16] des runs. Il devra mettre à jour l'avancement du traitement d'un run dans NGL, en y insérant les statistiques obtenues lors du démultiplexage, les résultats des contrôles qualités, etc. Puisque l'objectif est d'obtenir un premier niveau de valorisation des fichiers 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 ou de remplacement d'outils. Il sera donc necessaire de maintenir les pipelines des différentes technologies de séquençage en conséquence. Par exemple l'évaluation de logiciels de trimming (Cutadapt, Trimmomatic) en vue d'un remplacement du logiciel fastx_clean de

la suite FASTX Toolkit[17] qui est un outil mono-coeur pour un outil multi-coeurs. Ou bien trouver et évaluer un logiciel d'assignation taxonomique plus performant que le logiciel Centrifuge utilisé actuellement.

3 Matériels et Méthodes

3.1 Le cluster de calcul et Slurm

Le Genoscope possède un cluster (*inti*) de calcul de 71 noeuds répartis sur 5 partitions. La partition « normal » est composé de 47 noeuds qui disposent entre 12 et 36 coeurs et entre 96 et 386 Go de Ram. La partition « small » est composée de 8 noeuds dont 4 qui possèdent 8 coeurs et 64 Go de Ram, et 4 autres qui disposent de 16 coeurs et 128 Go de Ram. Cette partition est utilisée pour les processus courts et/ou peu de mémoire. Les partitions « xlarge » et « xxlarge » ont deux noeuds composés de 48 et 2To de Ram, et de 56 coeurs et 6To de Ram respectivement. Ces deux partitions sont utilisées pour les processus demandant plusieurs jours ou semaines de caluls. La partition « production » du cluster *inti* est composé 12 noeuds qui disposent de 16 coeurs et de 257 Go de Ram. L'accès à l'utilisation du cluster et de ses noeuds est réalisé par le logiciel Slurm.

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[18], est la partie de la base de données utilisée par les équipes de bioinformatique. NGL_SEQ[19], 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[20], NGL_REAGENT[21] et NGL_PROJECTS[22]. La gestion et le suivi du développement informatique sont réalisés par le système de tickets Jira.

3.3 Le langage de programmation Perl

L'écriture du workflow des pipelines pour les séquenceurs MGI est réalisée dans le langage de programmation Perl. L'utilisation de ce langage est rendu necessaire pour des raisons historiques du laboratoire, puisque de nombreuses librairies et modules qui ont été utilisés dans l'écriture des pipelines sont écrits en Perl.

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*[23] et *paired end*[24]. 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[25] pour les différents pipelines.

3.4 Logiciels de démultiplexage et génération de fichiers de séquences (bcl2fastq - bcl-convert)

Ces deux logiciels de *Base Calling* (bcl2fastq et bcl-convert), sont tous deux développés et commercialisés par Illumina. Cette évaluation entre ces deux logiciels est nécessaire pour déterminer les changements qu'il y aura à faire dans les pipelines de génération de fichiers de séquences pour les technologies Illumina, en vue du remplacement de bcl2fastq (qui sera bientôt obsolète) par bcl-convert.

Dans un premier temps, il est nécessaire de déterminer les conditions optimales de bcl2fastq (temps total (*Elapsed time*[26]), temps CPU (*CPU time*[27]), pourcentage d'utilisation CPU (*%CPU*[28])) en fonction des ressources disponibles sur les noeuds du cluster (*inti*) réservé à la *prodution*, pour pouvoir comparer les performancesdes 2 logiciels. Les conditions optimales sont déterminées en fonction des paramètres suivants de bcl2fastq (l'équivalent de bcl-convert est indiqué entre crochets) :

- `r` [bcl-num-decompression-threads] : nombre de *threads*[29] accordé pour la décompréssion et la lecture des *Bases Calls*[30]
- `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 sont réalisés sur le même noeud de calcul, dans l'objectif de minimiser les biais. La comparaison est effectuée sur le temps total du démultiplexage, ainsi que sur le temps CPU et le pourcentage d'utilisation des CPU.

3.5 Les pipelines de génération de fichiers de séquences pour les technologies Illumina et Nanopore

Les pipelines de générations de fichiers de séquences pour les technologies Illumina et Nanopore réalisent dans un premier temps le démultiplexage permettant la création des fichiers de séquences corespondant aux échantillons et des fichiers de statistiques de ces derniers. Ils créent les runs, les pistes, et les readset dans NGL_BI en y insérant les metriques, graphiques et fichiers permettant leurs évaluations.

Concernant le pipeline de génération de fichiers de séquences pour la technologie MGI, il s'agira de dévelloper un pipeline simillaire à celui d'Illumina en prenant en compte que

le démultiplexage est directement réalisé par les séquenceurs. Les métriques, graphiques et fichiers de statistiques sont également différents d'Illumina. Il sera donc necessaire de trouver comment obtenir les métriques, graphiques et fichiers, ou de les calculer, à partir des données générées lors du le séquenceur permettant de les insérer dans NGL_BI

3.6 Les pipelines de contrôle qualité des lots de séquences pour les technologies Illumina et Nanopore

Les pipelines de contrôle qualité des lots de séquences réalisent différentes étapes de contrôle qualité et de nettoyage des lots de séquences. Il réalise le contrôle qualité et l'estimation de duplicat des séquences des fichiers avant et après nettoyage (*trimming*), il retire le *PhiX*[31] (pour les technologies Illumina), réalise l'assignation taxonomique des séquences, réalise un allignement des séquences si un génome de référence existe, réalise le calcul du pourcenatage de séquences qui ont leurs reads *forward* (brin sens) et *reverse* (brin anti-sens) qui se chevauchent et réalise la distribution des fichiers de séquences nettoyés dans leurs répertoires de projet, d'échantillon, de type de technologie et de run.

Concernant le pipeline de contrôle qualité des fichiers de séquences pour la technologie MGI, qui est en cours de développement. Il s'agit de développer un pipeline similaire à celui d'Illumina en prenant en compte qu'avec cette technologie il n'y a pas de *PhiX* à enlever dans les fichiers de séquences.

4 Résultats

4.1 Etude comparative des logiciels 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 5, pour p fixé à 12. Des resultats similaires ont été obtenus pour p égale à 4, 8 et 16.

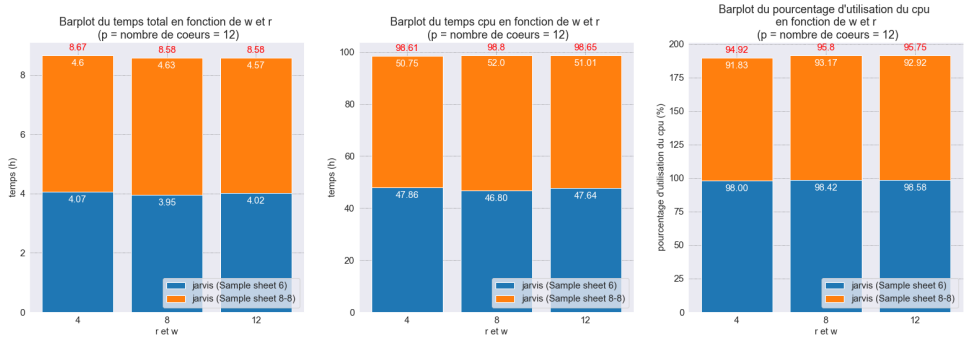


FIGURE 5 – Digrammes en bâtons du temps total d'éxécution (à gauche), temps cpu (au milieu) et du pourcentage d'utilisation des cpu (à droite) en fonction des paramètres r et w

Il y a deux *sample sheet*[32], car le nombre de bases considérées 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 6, 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 coeurs pour p, plus l'execution est rapide. On observe que le temps cpu augmente bien avec le nombre de coeurs et que le pourcentage d'utilisation des cpu est optimal ($> 90\%$).

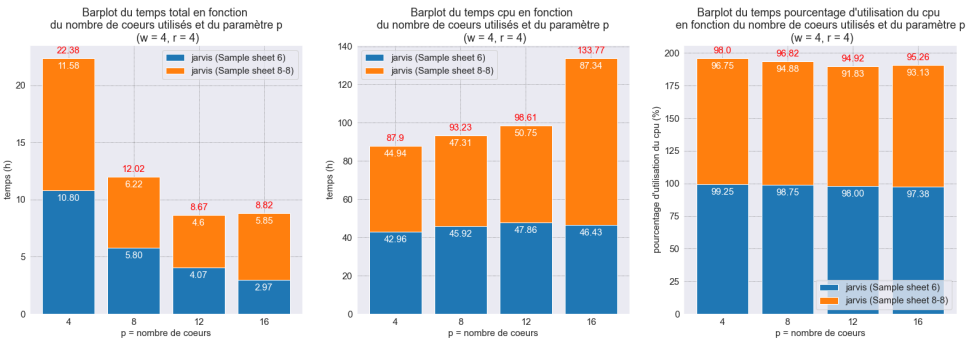


FIGURE 6 – Digrammes en bâtons du temps total d'éxécution (à gauche), temps cpu (au milieu) et du pourcentage d'utilisation des cpu (à droite) en fonction du paramètre p

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 coeurs accordés aux deux logiciels. On observe bien, sur la figure 7, 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.

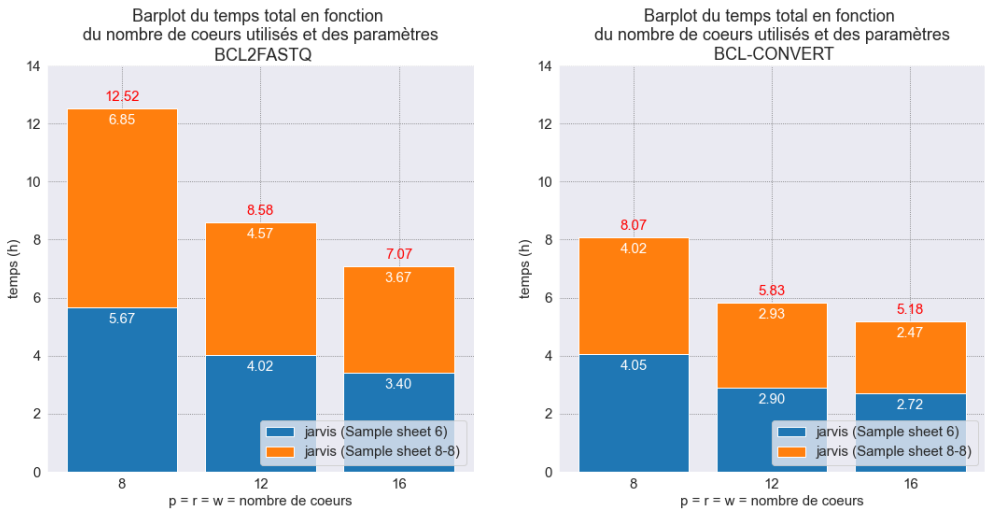


FIGURE 7 – Temps total de génération des fastq pour bcl2fastq et bcl-convert

J'ai également échangé avec le service technique d'Illumina à propos des fichiers de sortie et de l'arborescence de des derniers en utilisant bcl-convert. En effet il s'avère que l'arborescence et les fichiers de sortie sont très différents entre les deux logiciels. Ces échanges 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.

4.1.3 Préparation de la migration de bcl2fastq vers bcl-convert

Le logiciel bcl-convert est plus rapide d'environ 1/3 par rapport à bcl2fastq. Sachant également que ce dernier sera bientôt obsolète et que le nombre de coeurs disponibles par noeuds pour la partition « *production* »du cluster de calcul est de 16 coeurs. Nous avons décidé t'attribuer l'intégralité des coeurs d'un noeud de « *production* », c'est à dire 16 coeurs. L'intégralité des changement entre les deux logiciels a été consignés dans un cahier des charges. Il contient, la commande à lancer pour réaliser le *Base Calling*, les modules à charger dans l'environement, le chemin relatif des fichiers de sorties et leur description, ainsi qu'un exemple d'arborescence des fichiers de sorties. Ce qui permettera au développeur qui ce chargera de cette migration de suivre ce cahier des charges et ainsi faciliter cette migration. Dû à la pression actuelle autour de la technologie MGI, c'est un autre développeur qui sera en charge de réaliser cette migration.

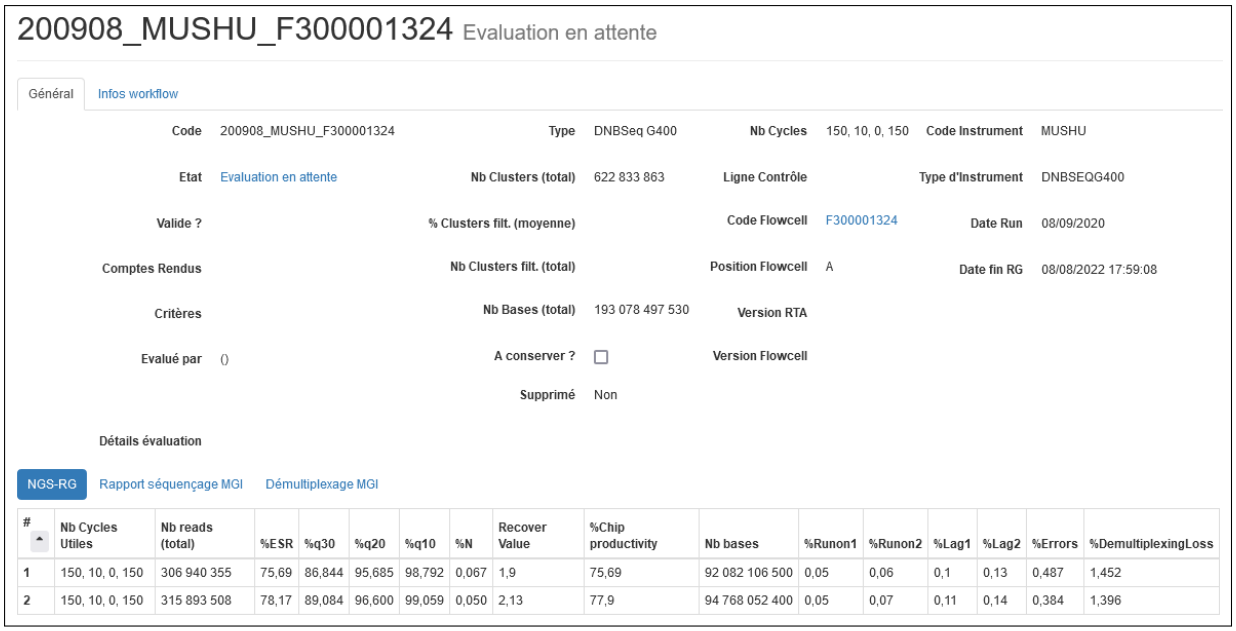
4.2 Le pipeline de génération de fichiers de séquences pour la technologie MGI

L'objectif du pipeline NGS_RG_MGI est de générer et distribuer les fichiers de séquences dans le bon répertoire de projet, d'échantillon, de type de séquençage et de run. Tout en créant et mettant à jour les runs, pistes et readsets. Notament concernant les métriques d'évaluations des ces derniers. Le pipeline est composé de plusieurs grandes étapes.

Création et insertion des métriques du run et des pistes dans NGL

La première étape consite à créer le run et ses pistes dans la base de données NGL, en y intégrant les métriques permettant d'évaluer le run et les pistes (figure 8). Le nom du run est constitué de la date de séquençage, du nom du séquenceur et de l'identifiant de la flowcell du run.

On y retrouve notament, le nombre total de reads (Nb Cluster (total)), le nombre de bases totales (Nb Bases (total)) générées par le run, la taille des reads et des index (Nb Cycles). Concernant les pistes on retrouve le nombre total de bases et de reads générés sur la piste, le pourcentage de bases qui ont une qualité supérieur ou égale à Q30, Q20 et Q10. On a également le pourcentage de bases inconnus (%N), ainsi que d'autres métriques qui permettent d'avaluer le run et les pistes. Celles-ci sont détaillées plus précisement en anexes (page 21).



200908_MUSHU_F300001324 Evaluation en attente

Général | Infos workflow

Code	200908_MUSHU_F300001324	Type	DNBSeq G400	Nb Cycles	150, 10, 0, 150	Code Instrument	MUSHU
Etat	Evaluation en attente	Nb Clusters (total)	622 833 863	Ligne Contrôle		Type d'Instrument	DNBSEQG400
Valide ?		% Clusters filt. (moyenne)		Code Flowcell	F300001324	Date Run	08/09/2020
Comptes Rendus		Nb Clusters filt. (total)		Position Flowcell	A	Date fin RG	08/08/2022 17:59:08
Critères		Nb Bases (total)	193 078 497 530	Version RTA			
Evalué par	()	A conserver ?	☐	Version Flowcell			
		Supprimé	Non				

Détails évaluation

NGS-RG | Rapport séquençage MGI | Démultiplexage MGI

#	Nb Cycles Utiles	Nb reads (total)	%ESR	%q30	%q20	%q10	%N	Recover Value	%Chip productivity	Nb bases	%Runon1	%Runon2	%Lag1	%Lag2	%Errors	%DemultiplexingLoss
1	150, 10, 0, 150	306 940 355	75,69	86,844	95,685	98,792	0,067	1,9	75,69	92 082 106 500	0,05	0,06	0,1	0,13	0,487	1,452
2	150, 10, 0, 150	315 893 508	78,17	89,084	96,600	99,059	0,050	2,13	77,9	94 768 052 400	0,05	0,07	0,11	0,14	0,384	1,396

FIGURE 8 – Capture d'écran de la page du run 200908_MUSHU_F300001324 de NGL en cours de génération de fichiers de séquences (étapes d'ajout des métriques d'évaluation du run et des pistes).

Insertion des rapports de séquençage des pistes et de la listes des index dans NGL

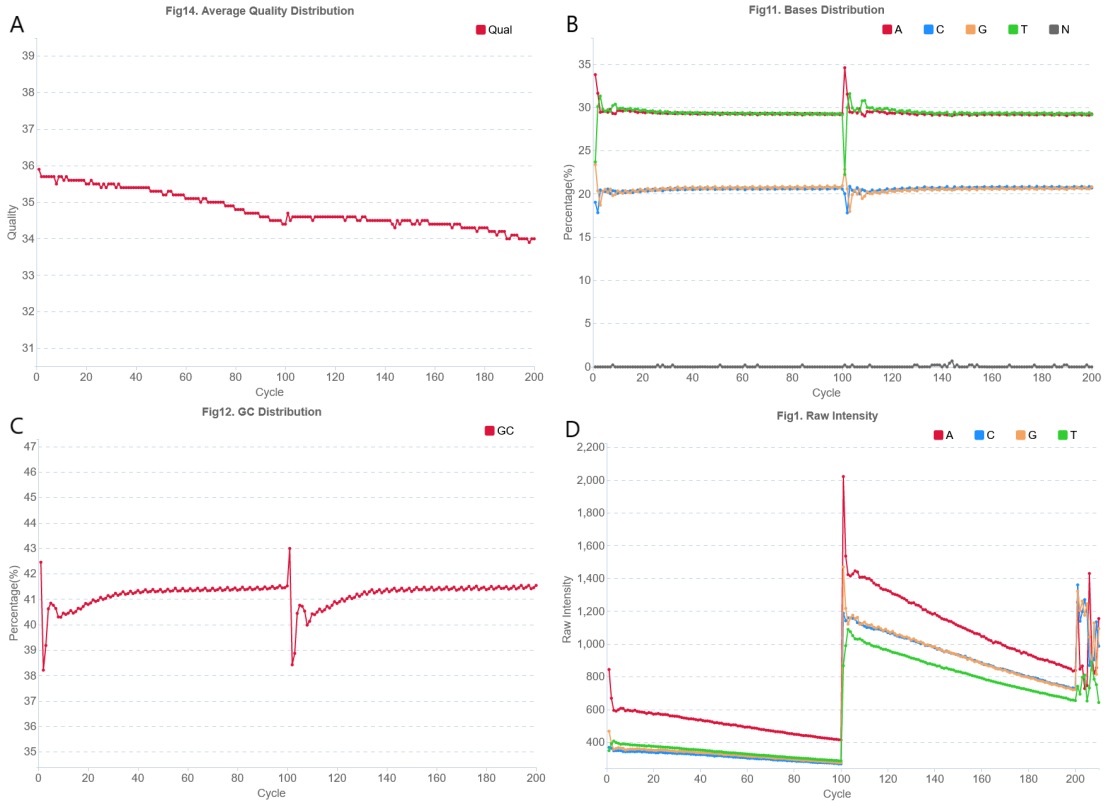


FIGURE 9 – Graphiques des distributions de la qualité moyenne (A), des bases nucléiques (B), du pourcentage de GC (C) et de l'intensité brut (D) au cours des cycles de séquençage

La seconde étape ajoute les rapports de séquençages des pistes que le séquenceur génére en fin de séquençage. Il s'agit de rapport html qui contiennent plusieurs tableaux de métriques et de graphiques permettant d'évaluer les pistes du run. Il y a notament les graphiques de la distribution de la qualité moyenne en fonction des cycles (figure 9.A), de la distribution des bases nucléiques en fonction des cycles (figure 9.B), de la distribution du pourcentage de Guanine/Cytosine en fonction des cycles (figure 9.C), de la distribution de l'intensité brut au cours des cycles (figure 9.D). Les tableaux et graphiques de ces rapports de séquençage permettent de faciliter l'évaluation du run et de ses pistes.

Toujours dans l'optique de facilité l'évaluation du run et de ces pistes on ajoute, la liste des index représentées à plus de 0.01% de la pistes, ainsi que les index attendus. Ces index sont triés et affichés par ordre croissant dans NGL (figure 10). Les index attendus sont colorés en vert et les index non-attendus ou inconnus sont colorés en rouge, ce qui permet de vérifier que les index attendus sont bien majoritairement représentés sur les pistes de la flowcell du run.

NGS-RG | Rapport séquençage MGI | Démultiplexage MGI

Lane 1

barcode	count	percent
barcode2	89 597 340	29,190
barcode1	84 106 886	27,402
barcode3	74 172 719	24,165
barcode4	54 607 003	17,791
GATTCGTCCT	206 151	0,067
ATCGGACTAT	181 509	0,059
GATCCGTCCT	156 796	0,051
ATTCCGTCCT	156 103	0,051
CGCAGTAAGT	148 841	0,048
ATCGACCTAT	119 597	0,039
TCAATAGGTT	114 220	0,037
CGGAGTAAGT	99 851	0,033
GGCAGTAAGT	85 114	0,028
ATGGACCTAT	83 324	0,027
ACGGACCTAT	75 840	0,025
CAATTAGGTT	71 106	0,023
CGGCATAAGT	70 917	0,023
GATTCCTCCT	59 842	0,019
CGGCAGAAGT	53 283	0,017
barcode29	48 716	0,016
barcode124	37 751	0,012
CGGCGTAAGT	36 893	0,012

FIGURE 10 – Capture d'écran de la page du run 200908_MUSHU_F300001324 de NGL en cours de génération de fichiers de séquences (onglet « Démultiplexage MGI »)

Concaténation des fichiers FASTQ d'un même readset

Ensuite la troisième étape à pour objectif d'obtenir un seul fichier FASTQ par readset. En effet la technologie MGI requiert une homogénéité de dépôt entre les différents index (aussi appelé« barcode ») d'une piste. Ce qui implique qu'il est possible d'avoir plusieurs index associés à un même index, donc qu'un échantillon peut être divisé en plusieurs fractions. Le démultiplexage est directement réalisé par le séquenceur et le réalise à partir des index connus (listes d'index fournis par MGI), on obtient donc un fichier FASTQ par index. Il est impossible de préciser au séquenceur quels index sont associés à un même readset pour le démulitiplexage. Cette étape est donc essentielle pour obtenir un seul fichier FASTQ par readset. Si le readset est associé à un seul readset alors on réalise une décompréssion du fichier FASTQ, alors que s'il est associé à plusieurs index on réalise une décompréssion et une concaténation des fichiers FASTQ (cf. figure 11).

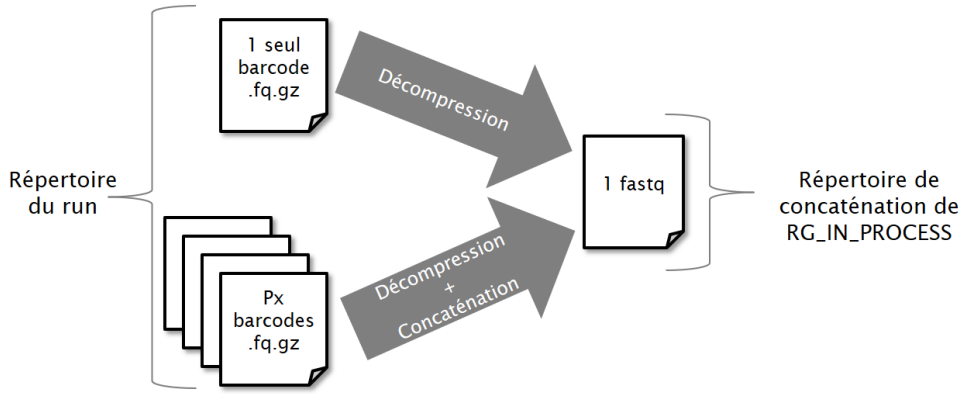


FIGURE 11 – Schéma de l'étape de « concaténation » des fichiers FASTQ d'un readset

Création et insertion des métriques des readset du run dans NGL

La quatrième étape a pour objectif de permettre l'évaluation des readsets, en les créant et en insérant les métriques d'évaluation de ces derniers dans NGL (figure 12). On y retrouve notamament le nombre de bases nucléiques et de reads du readset, ainsi que le pourcentage d'échantillon déposé sur la piste et le pourcentage de séquences valides par rapport au nombre total de séquences de la piste. On y insère également certaines métriques du run dont le readset fait partie, comme le nombre de cycles des reads et des index, la date de run, ect. Toutes ces métriques sont décrites en annexes (page 22)

Le nom du readset est constitué de l'identifiant de projet, de l'identifiant du type de banque utilisée (ADN, ARN ...), de l'identifiant d'échantillon, de l'indice de la piste, de l'identifiant de la flowcell et de l'identifiant du premier barcode.

FIGURE 12 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de reads (étapes de création du readset et d'insertion de ces métriques d'évaluation)

On ajoute également la répartition des index au sein d'un readset (figure 13), ce qui permet de vérifier la composition en index du readset et de vérifier l'homogénéité de ces index au sein du readset. Ce nom de readset est unique, ce qui permet de déterminer rapidement et simplement à quel projet, échantillon, ect appartiennent les fichiers séquences de ce readset.

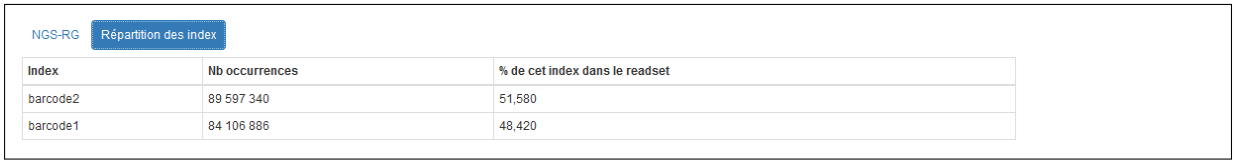


FIGURE 13 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de reads (onglet « Répartition des index »)

Au niveaux du run un tableau référençant les readsets et leurs métriques d'évaluation est également ajouté (figure 14).

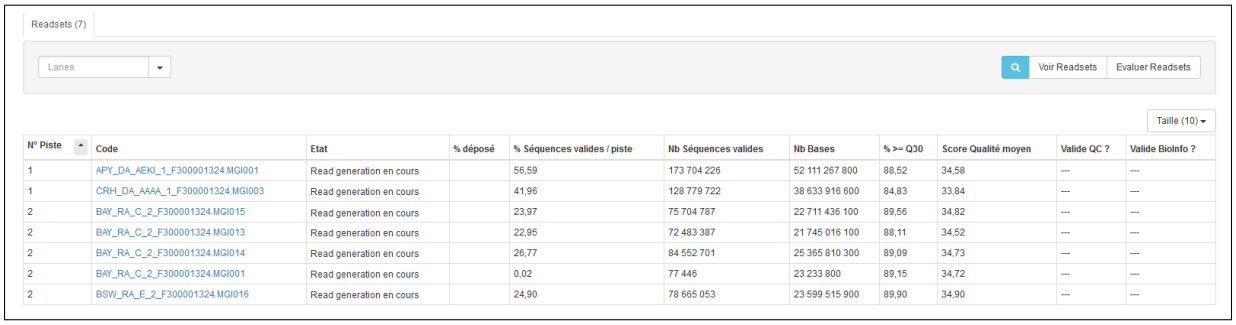


FIGURE 14 – Capture d'écran de la page du run 200908_MUSHU_F300001324 de NGL en cours de génération de fichiers de séquences (Tableau des readset du run)

Renommage des fichiers séquences et insertion des méta-données dans NGL

La cinquième étape consiste à renomer les fichiers de séquences des readsets et d'insérer les méta-données de ces derniers dans NGL (figure 16). Le renommage des fichiers est nécessaire pour que chaque fichiers de séquences aient un nom unique et « parlant ».

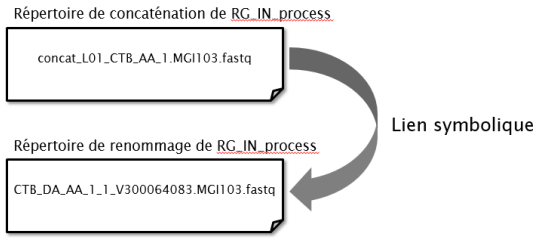


FIGURE 15 – Schéma de l'étape de renommage des fichiers FASTQ d'un readset

Le nom doit permettre d'indentifier rapidement et simplement de quel projet, échantillon, flowcell, ect. appartienent les fichiers. Le renomage des fichiers est effectués en créant un lien symbolique des fichiers obtenus à l'étape de « concaténation » dans un répertoire temporaire (cf figure 15).

Les méta-données, des fichiers de séquences du readset, qui sont insérées dans NGL permet aux utilisateurs de trouver rapidement l'emplacement de ces derniers sur le système de fichier, le type de fichier qui est disponible (brut, nettoyer) et s'ils sont utilisable. On y retrouve donc le chemin vers le repertoire de ces fichiers, leurs noms, leurs types, s'ils sont utilisable, s'il s'agit du read *forward* ou *reverse*, et le type d'encodage de la quality (Pour les séquenceurs MGI l'encodage est en ASCII 33).

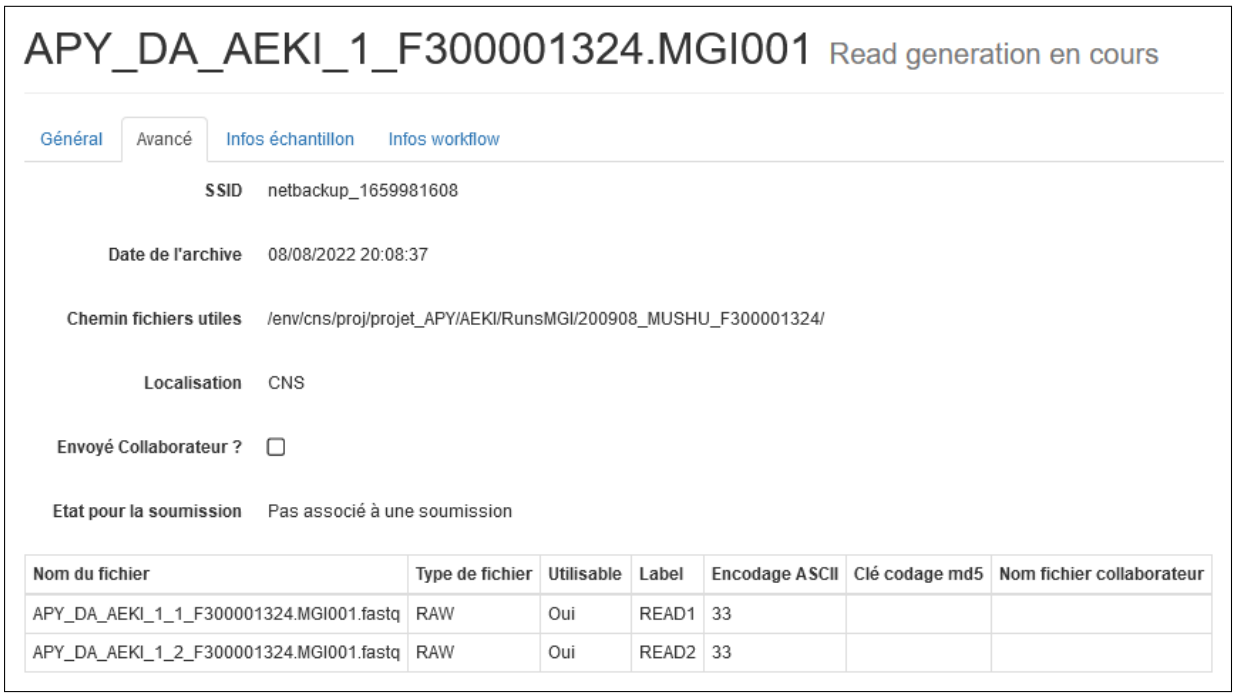


FIGURE 16 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de fichiers de séquences (Onglet « Avancé »)

Distribution des Fichiers séquences et des fichiers de statistiques

La sixième étape est de distribuer des fichiers de séquences « attendus », les fichiers de statistiques du run et les fichiers de séquences « non attandus » dans leurs répertoires dédiés.

Les fichiers attendus sont copiés vers leurs répertoires final si le séquençage a été effectué au genoscope. Sinon, si ce dernier a été effectué au CNRGH, alors les fichiers sont copiés et compressés. Il y a cette différence entre les 2 centres, car le pipline de contrôle qualité prend en charge de fichiers compressés pour le CNRGH, contrairement à celui utilisé pour le Genoscope.

Les fichiers de statistiques du run sont archivés par pistes et par types (.html, .fq.stat) avant d'être copiés vers leurs répertoires final. Ces fichiers sont conservés dans le cas où une métrique désirés ne fait pas partie de celles insérées dans NGL ou pour tout autres problémes qui nécéssiteraient de récupérer les fichiers de statistiques du run.

Concernant les fichiers « non attendus », il s'agit des fichiers de séquences des index ne faisant pas partie d'un readset. Puisque lors du demultiplexage par les séquenceurs on obtient un fichier FASTQ par index. Ces fichier sont renomés et archivés, avant d'être distribués vers leur répertoire dédiés. Ces fichers de séquences sont conservés dans l'éventualité d'une mauvaise déclaration d'index par les équpies de séquençages, pour pouvoir récupérer les fichiers fastq appartenant à cet index ou si l'on souhaite étudier les séquences des fichiers « non-attendus ».

Mise à jour de fin de génération de fichiers de séquence dans NGL

L'étape finale du pipeline de génération de fichiers de séquences pour la technologie MGI, est de mettre à jour le run et les readset dans l'état de « fin de génération de reads ». Cela entraine une mise à jour automatique du run à l'état « d'évaluation en attente », ce qui permet d'indiquer aux utilisateurs que le run peut être évalué. Les readset sont aussi automatiquement mis à jour vers l'état « d'attente de contrôle qualité », permettant d'indiquer au pipeline de contrôle qualité qu'il peut effectuer le contrôle qualité des readset de ce run.

5 Discussions et perspectives

5.1 perspectives du workflow NGS pour la technologie MGI

5.1.1 Améliorations futures du pipeline NGS_RG pour la technologie MGI

Le pipeline de génération de fichiers de séquences pour la technologie MGI est similaire au pipeline de la technologie Illumina. Néainmoins il n'est pas possible de comparer ces deux derniers au niveau de leurs performances du fait de leurs différences. En effet le pipeline de génération des fichiers de séquences pour la technologie Illumina, contient les étapes de *Base Calling* et de démultiplexage (Conversion des fichiers *Base Calls* en fichiers FASTQ par èchantillon) qui est réalisé par le pipeline NGS_RG_ILLUMINA. À contrario, pour la technologie MGI, cette étape est directement réalisée par le séquenceur. De plus il n'est pas possible de comparer le pipeline de génération de fichiers de séquences avec des pipelines d'autres laboratoire ou outils de génération de fichiers de séquences dû fait de la spécificité du pipeline pour le Genoscope et le CNRGH. En effet l'objectif de celui-ci est de mettre à jour la base de données de référence interne au Genoscope et au CNRGH (NGL), à l'architecture de stockage des fichiers de séquences et aux noms finaux donnés à ces fichiers pour qu'ils soient uniques.

La future amélioration du pipeline NGS_RG_MGI, consistera à la mise en place d'une étape suplémentaire pour les runs qui comporterons de *mids*[33]. Cette étape suplémantaire sera donc le démidage, il s'agit d'un second démultiplaxage en fonction des mids pour la création des readsets et fichiers de séquences.

5.1.2 Développement du pipeline de contrôle qualité pour le technologie MGI

Le pipeline de contrôle qualité des fichiers de séquences pour la technologie MGI qui est en cours de développement, est constitué de deux grande étapes. La première consiste à réaliser un contrôle qualité des fichiers de séquences brut, puis dans un second temps de réaliser un contôle qualité sur les fichiers de séquences nettoyés ainsi que d'autres traitements sur les fichiers nettoyés. Il prend en charge automatiquement les fichiers dont les readset sont dans l'état « en attente de contrôle qualité » dans NGL.

Avant de réaliser le trimming su les fichiers de séquences brut, on réalise un échantilonage de 20000 séquences par fichiers pour réaliser le contrôle qualité des fichiers. Cela permet d'améliorer le temps d'execution du contrôle qualité, tout en ayant une grande représentativité de la qualité des séquences des fichiers. On réalise donc le contôle qualité et l'estimation de dupliquas sur les échantillon des fichiers brut.

Ensuite on réalise le trimming des fichiers brut, comme il n'y a pas de *PhiX* à retirer, il s'agit du seul traitement de netoyage des fichiers brut. On réalise l'échantilonnage de 20000 séquences par fichiers nettoyés pour réaliser leurs contrôle qualité.

Le second contrôle qualité réalisé sur les échantillons des fichiers brut, est composé d'un contrôle de la qualité des séquences, d'une assignation taxonomique des séquences des fichiers, l'estimation des dupliquas (séquences retrouvés plusieurs fois dans un fichier de séquences), on réalise aussi un alignement des séquences sur un génome de références si ce derniers existe et est disponible et on calcule le pourcentage de reads qui ont le read *forward* et *reverse* qui se chevauchent.

Une fois le deuxième contrôle qualité effectué, la dernière étape du pipeline est de distribuer les fichiers de séquences nettoyés dans leur répertoire final, de rendre indisponible les fichiers de séquences brut avant de les effacer une fois celle-ci copié sur bande magnétique. Durant toutes les etapes du pipeline, on insert les métriques et graphiques qui permettront de réaliser la validation des readset ou non.

5.2 Evaluation d'outils de contrôle qualité

Les premiers outils à être évalués sont cutadapt et trimmomatic en vue d'un remplacement de fastx_clean de FASTX Toolkit. Ce dernier est un logiciels mono-coeur contrairement à cutadapt et trimmomatic qui sont multi-coeurs. Le temps d'exécution entre ces logiciels sera le critère d'évaluation le plus important, néaimoins on prendra également en compte les différents fichiers de sortie (fichiers de statistiques, fichiers de séquences qui ne passe pas les filtres données ...) pour l'évaluation et le remplacement de fastx_clean.

Un potentiel successeur au logiciels d'assignation taxonomique Centrifuge devra egalement être effectuer, dans l'optique d'améliorer les pipelines de contrôle qualité. L'objectif est de trouver un logiciels dont les performance son équivalentes ou meilleurs, surtout au niveau du temps d'exécution, mais également au niveau de l'assignation taxonimique des séquences et des fichiers de sortie.

Notes

[1]Reconstruction d'un génome à partir de fragments de ce dernier

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

[3]*Next Generation Sequencing - reads generation*

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

[5]*Next Generation Sequencing - quality control*

[6]*Next Generation Sequencing - biological analysis*

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

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

[9]Collection de fragment d'ADN issue du génome complet d'un organisme ou plusieurs organismes (méta-génomique) et clonés dans un vecteur (le plus souvanet dans des plasmides)

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

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

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

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

[14]pistes présentes sur la *flowcell*

[15]Lot de séquences

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

[17]Collection de commandes pour le traitement et l'évaluation de lot de séquences au format FASTA ou FASTQ

[18]NGL Bioinformatic

[19]NGL Sequencing

[20]NGL submission (base de données des soumissions de projet (example la soumission d'un projet au NCBI))

[21]NGL reagent (base de données des réactifs)

[22]NGL projects (base de données des projets en cours et passé)

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

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

[25]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.

[26]Temps écoulé entre le début du programme et le fin de celui-ci

[27]Temps d'utilisation des cpu par le programme

[28]((*CPU time* + temps utilisé par les appels système) / *Elapsed time*) / nombres de CPU utilisé par le programme

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

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

[31]Parties du génome du phage *Lambda* qui sont ajoutés sur les pistes des flowcell avant le séquençage, permettant de contrôler le bon déroulé du séquençage.

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

[33]séquence d'une dizaine de nucléotide ajouté en aval du *primer* du read *forward* permettant de réaliser un second démultiplexage

6 Annexes

Description des métriques d'éavaluation d'un run et des pistes d'un run MGI dans NGL-BI

Liste et les description des métriques d'évaluation du run et des pistes (cf. figure 8 page 12) :

Nb Cycles Utiles : Nombre de cycles des reads et des index (nombre de cyles pour le read *forward*, nombre de cyles pour le premier index *forward*, nombre de cycles pour le read *reverse*, nombre de cycles pour le second index)

Nb reads (total) : Nombre de reads Total générer par la piste (S'il s'agit d'un run *pair-end* il s'agit du nombre de cluster de reads (read *forward* + read *reverse*))

%ESR : ???

%q30 : Pourcentage de bases qui ont une qualité supérieur ou égale à 30 (pour un encodage de la qualite en ASCII 33)

%q20 : Pourcentage de bases qui ont une qualité supérieur ou égale à 20 (pour un encodage de la qualite en ASCII 33)

%q10 : Pourcentage de bases qui ont une qualité supérieur ou égale à 10 (pour un encodage de la qualite en ASCII 33)

%N : Pourcentage de bases inconus

Recover value : ???

%Chip productivity : Pourcentage de productivité de la piste (nombre de puits actif de la piste de la flowcell / nombre total de puit de la piste)

Nb bases : Nombre total de bases générés par la piste

%Runon1 : Pourcentage de read *forward* qui ont une incorporation de nucléotide d'avance par rapport au cycles en cours

%Runon2 : Pourcentage de read *reverse* qui ont une incorporation de nucléotide d'avance par rapport au cycles en cours

%Lag1 : Pourcentage de read *forward* qui ont une incorporation de nucléotide de retard par rapport au cycles en cours

%Lag2 : Pourcentage de read *reverse* qui ont une incorporation de nucléotide de retard par rapport au cycles en cours

%Errors : Pourcentage d'erreur d'incorporation de nucléotide

%DemultiplexingLoss : Pourcentage de read ecartés lors du démultiplexage

Description des métriques d'un readset d'un run MGI dans NGL-BI

Liste des métriques d'évaluation des readset dans NGL_BI (cf. figure 12 page 15) :

Nb reads : Nombre de reads avant nettoyage des fichiers du readset
%déposé : Pourcentage d'échantillon déposé sur la piste de la flowcell
Nb bases : Nombre de bases avant nettoyage des fichiers séquences du readset
% séquences valides/piste : Pourcentage de séquences de la piste appartenant à ce readset (nombre total de reads du readset / nombre total de reads de la piste)

Liste des métriques d'évaluation des readsets dans le tableau qui référence tous les readsets d'un run (cf. figure 14 page 15) :

%déposé : Pourcentage d'échantillon déposé sur la piste de la flowcell
% séquences valides/piste : Pourcentage de séquences de la piste appartenant à ce readset (nombre total de reads du readset / nombre total de reads de la piste)
Nb Séquences valides : Nombre de reads du readset
Nb Bases : Nombre de bases du readset
% >= Q30 : Pourcentage de bases qui ont une qualité supérieur ou égale à 30 (pour un encodage de la qualité en ASCII 33)
Score Qualité moyen : Moyenne de la qualité des bases du readset

Autres informations à propos d'un run MGI dans NGL-BI

On retouve également les informations permettant de suivre l'avancement du workflow NGS au niveau de l'onglet « infos workflow » (figure 17).

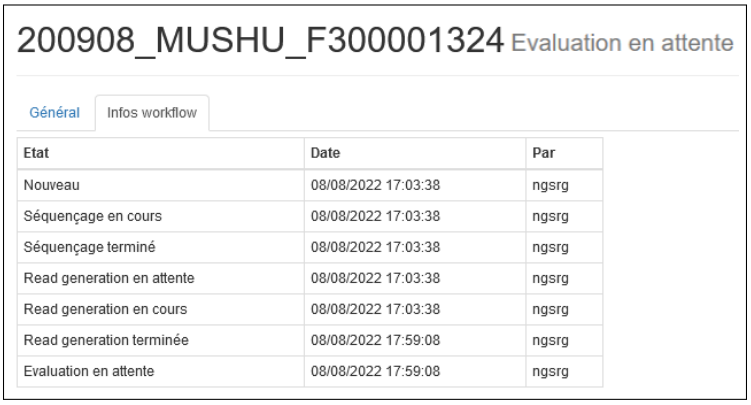


FIGURE 17 – Capture d'écran de la page du run 200908_MUSHU_F300001324 de NGL en cours de génération de fichiers de séquences (onglet « infos workflow »)

Autres informations à propos d'un readset MGI dans NGL-BI

Il y a deux autres onglets en plus de l'onglet « Général » et « Avancé ». Il s'agit de l'onglet « Infos échantillon » (figure 18) et de l'onglet « Infos workflow » (figure 19). Tout comme pour le run, l'onglet « Infos workflow » permet de suivre l'avancement du workflow NGS pour le readset. Concernant l'onglet « Infos échantillon », référence toutes les informations à propos de l'échantillon du readset. On y retrouve son code, le taxon dont il fait partie et son ID, la catégorie d'échantillon (ADN, ARN ...), la listes des barcodes utilisés et d'autres informations

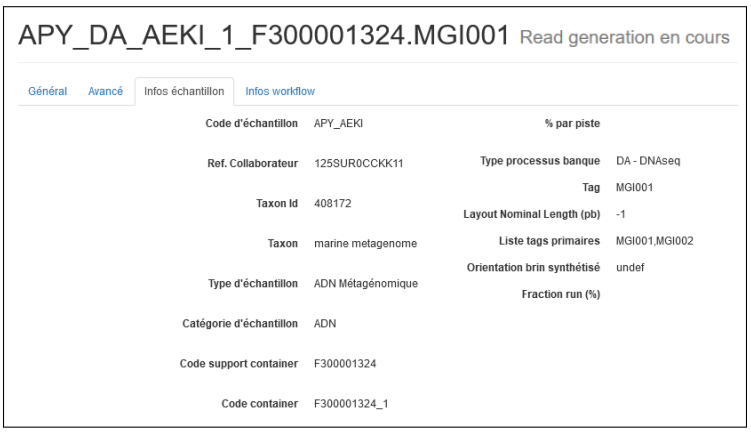


FIGURE 18 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de reads (onglet « Infos échantillon »)

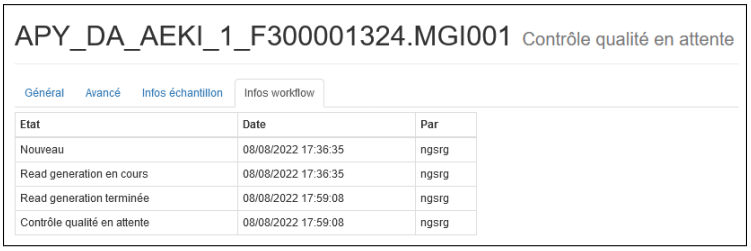


FIGURE 19 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de reads (onglet « Infos workflow »)