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

---

2 septembre 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)

DNB : *DNA-nanoballs* (Nano « billes » d'ADN générés lors de l'amplification ADN pour les séquenceurs de la technologie MGI)

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 logiciels et services d'analyses 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 développe notamment la base de données de génomes GenBank et la base de données des publications PubMed

NGL : *Next Generation LIMS* (bases de données du Genoscope et du CNRGH)

NGL_BI : *NGL Bioinformatic* (base de données des analyses et traitements bio-informatique)

NGL_PROJECT : *NGL projects* (base de données des projets en cours et passé)

NGL_REAGENT : *NGL reagent* (base de données des réactifs)

NGL_SEQ : *NGL Sequencing* (base de données de suivi des échantillons)

NGL_SUB : *NGL submission* (base de données des soumissions de projet ou d'articles (exemple : la soumission d'un projet au NCBI))

NGS : *Next Generation Sequencing*

NGS_BA : *Next Generation Sequencing - biological analysis*

NGS_QC *Next Generation Sequencing - quality control*

NGS_RG : *Next Generation Sequencing - reads generation*

Oxford Nanopore : Entreprise Anglaise de biotechnologie fondée en 2005, qui développe 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 développe 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 notamment 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.

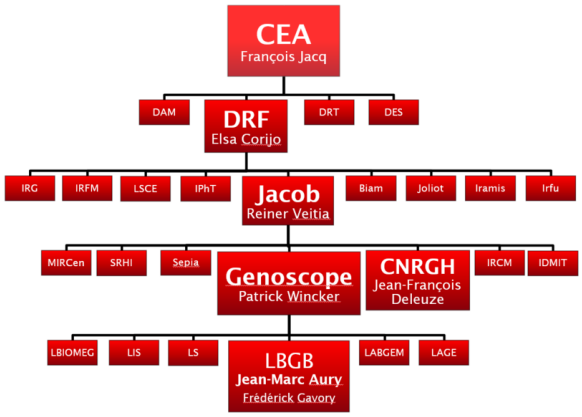


FIGURE 1 – Organigramme situant l'équipe du LBGB au sein du Genoscope et du CEA

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 différentes technologies de séquençage, d'effectuer l'assemblage[1] des séquences et l'annotation[2] des génomes, dans l'objectif de mettre à disposition des laboratoires collaborateurs internes ou externes 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 « d'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 séquenceur. 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

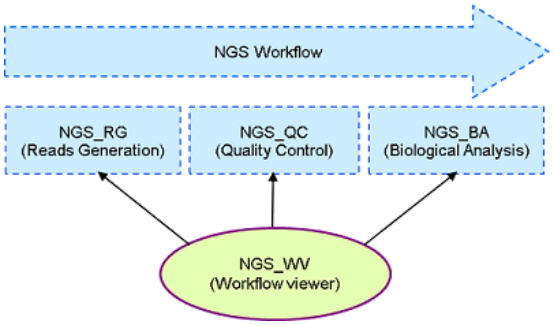


FIGURE 2 – Workflow de génération, de contrôle qualité et d'analyse biologique des fastq

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

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[5] 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 récemment fait l'acquisition de séquenceurs MGI (2 DNBSEQ-G400 et 1 DNBSEQ-T7).

FIGURE 3 – Séquenceurs DNBSEQ-G400 (en haut) et DNBSEQT7 (en bas) de MGI https://en.mgi-tech.com/products/

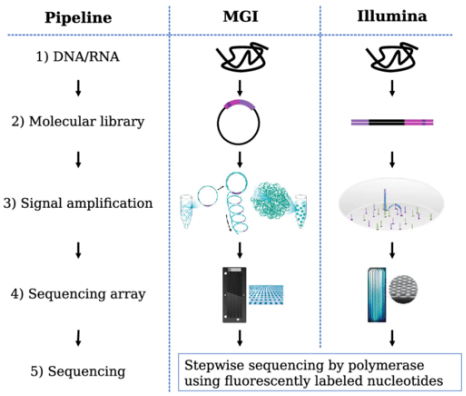


FIGURE 4 – Différences entre Illumina et MGI de technologie NGS

Il s'agit de séquenceurs à haut débit (DNBSEQ-G400) et très haut débit (DNBSEQ-T7), dont les principales différences entre MGI et Illumina sont dans la création des librairies[6] 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*[7]), puis déposée sur la flowcell[8] pour MGI, alors que pour Illumina elle est réalisée après immobilisation sur les flowcells.

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 mission

L'objectif principal de ma mission est la mise en place d'un workflow NGS pour les séquenceurs MGI. Dans un premier temps il s'agira de créer un pipeline de génération de fichiers de séquences (NGS_RG_MGI[9]) puis un pour le contrôle qualité de ces fichiers (NGS_QC_MGI[10]). Le workflow devra créer et mettre à jour l'état des runs, des pistes ou (*lane*) et de readset[11] dans NGL, réaliser le contrôle qualité des fichiers de séquences, au format FASTQ, obtenus en fin de séquençage. Il devra mettre à jour l'avancement du traitement d'un run dans NGL, en y insérant les métriques et statistiques obtenues lors du démultipléxage[12], 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 », « annotation ») de prendre en charge ces fichiers avant de les mettre 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 nécessaire 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 l'outil fastx_clean

de l'extension fastxend de la suite FASTX Toolkit[13] 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 composée 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 qui demande peu de mémoire Ram. Les partitions « xlarge » et « xxlarge » ont chacun deux noeuds composés de 48 coeurs et 2To de Ram, de 56 coeurs et 6To de Ram respectivement. Ces deux partitions sont utilisées pour les processus demandant plusieurs jours ou semaines de calculs. La partition « production » du cluster *inti* est composée de 12 noeuds qui disposent de 16 coeurs et de 257 Go de Ram. Les différents pipeline du workflow NGS utilisent cette partition. 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, est la partie de la base de données utilisée par les équipes de bioinformatique. NGL_SEQ, 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, NGL_REAGENT et NGL_PROJECTS. La gestion et le suivi des développements informatiques 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 sera réalisée dans le langage de programmation Perl. L'utilisation de ce langage est rendu nécessaire pour des raisons historiques du laboratoire, puisque de nombreuses librairies et modules qui seront utilisés dans le développement 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*[14] et *paired end*[15]. Cela m'a permis de prendre en main les librairies Perl utilisées pour les différents pipelines déjà en place. Ainsi que de m'habituer à l'environnement de travail, l'utilisation du lancement de job sur les noeuds de calculs et l'utilisation des modules[16] pour les différents pipelines.

3.4 Librairie et API Perl permettant d'interagir avec la base de données NGL

L'interaction entre les pipelines du workflow NGS et la base de données NGL s'effectue par des fichiers JSON[17]. Cette interaction est possible grâce à une API[18] développé en Perl par l'équipe de « production » du LBGB, elle permet d'ajouter, modifier, récupérer, supprimer des données des fichiers JSON. Une librairie Perl (*DBFactory*) permet d'interagir avec cette API directement depuis une autre librairie ou script Perl, c'est cette dernière qui sera utilisé dans le développement des pipelines du workflow NGS pour la technologie MGI.

3.5 Logiciels de démultipléxage et génération de fichiers de séquences (bcl2fastq - bcl-convert)

Ces deux logiciels de génération de fichiers de séquences et de démultipléxage (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 la technologie 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*[19]), temps CPU (*CPU time*[20]), pourcentage d'utilisation CPU (*%CPU*[21])) en fonction des ressources disponibles sur les noeuds du cluster (*inti*) réservé à la *prodution*, afin de pouvoir comparer les performances des 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*[22] accordé pour la décompression et la lecture des *Bases Calls*[23]
- `p` [bcl-num-conversion-threads] : nombre de *threads* accordé pour la conversion des *Bases Calls* en fastq
- `w` [bcl-num-compression-threads] : nombre de *threads* accordé l'écriture et la compression des fichiers fastq

J'ai réalisés tous ces tests sur le même noeud de calcul, dans l'objectif de minimiser les biais. La comparaison est effectuée sur le temps total du démultipléxage, ainsi que sur le temps CPU et le pourcentage d'utilisation des CPU.

3.6 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 démultipléxage permettant la création des fichiers de séquences correspondant 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 métriques, graphiques et fichiers permettant leurs évaluations.

Concernant le pipeline de génération de fichiers de séquences pour la technologie MGI, j'ai de développé un pipeline dont l'objectif final est le même que celui d'Illumina en prenant en compte que le démultipléxage est directement réalisé par les séquenceurs. Les métriques, graphiques et fichiers de statistiques sont également différents d'Illumina. Il sera donc nécessaire de trouver comment obtenir les métriques, graphiques et fichiers, ou de les calculer, à partir des données générées par le séquenceur, pour permettre de les insérer dans NGL_BI

3.7 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. Ils réalisent le contrôle qualité et l'estimation des duplicats de séquence des fichiers avant et après nettoyage (*trimming*), ils retirent le *PhiX*[24] (pour les technologies Illumina), réalisent l'assignation taxonomique des séquences, réalisent un alignement des séquences si un génome de référence existe, réalisent le calcul du pourcentage de séquences qui ont leurs reads *forward* (brin sens) et *reverse* (brin anti-sens) qui se chevauchent et réalisent 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, il s'agira de développer un pipeline dont l'objectif est le même que celui d'Illumina en prenant en compte qu'avec cette technologie il n'y auras pas de *PhiX* à enlever dans les fichiers de séquences.

4 Résultats

4.1 Etude comparative des logiciels bcl2fastq et bcl-convert

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 résultats similaires ont été obtenus pour p égale à 4, 8 et 16.

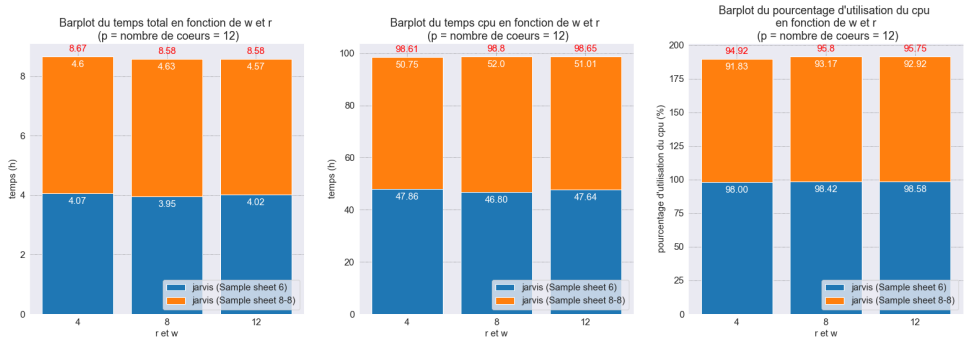


FIGURE 5 – Digrammes en bâtons du temps total d'exécution (à gauche), temps cpu (au milieu) et du pourcentage d'utilisation des cpu (à droite) en fonction des paramètres r et w

Il y a deux *sample sheet*[25], car le nombre de bases considérées des index[26] entre les pistes est différent, obligeant à réaliser deux appels différents au logiciel pour générer les FASTQ et le démultipléxage. 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 résultats). On observe que plus on augmente le nombre de coeurs pour p, plus l'exécution 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 ($> 90\%$).

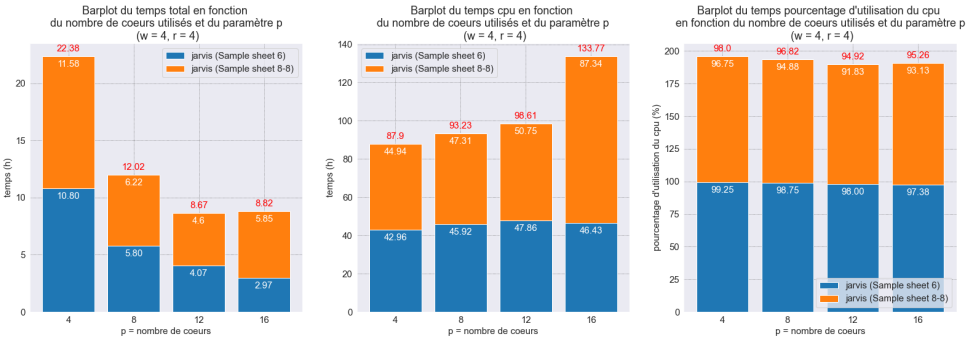


FIGURE 6 – Digrammes en bâtons du temps total d'exécution (à gauche), temps cpu (au milieu) et du pourcentage d'utilisation des cpu (à droite) en fonction du paramètre p

Néanmoins, on remarque qu'il n'y a pas d'amélioration du temps total d'exécution entre p fixé à 12 et à 16 coeurs. on observe même une augmentation du temps cpu.

Au vue des résultats obtenus j'ai décidé que les meilleurs paramètres étaient de fixer p à 12, puisque le gain apporté en augmentant à 16 est faible. Néanmoins j'ai décidé de le conserver 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 démultipléxage en simultané sur un seul noeud de calcul de la partition « production » puisqu'ils font 16 coeurs.

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, plus la génération des FASTQ et le démultipléxage est rapide. De plus on remarque que bcl-convert permet de réduire le temps d'environs 1/3 par rapport à bcl2fastq.

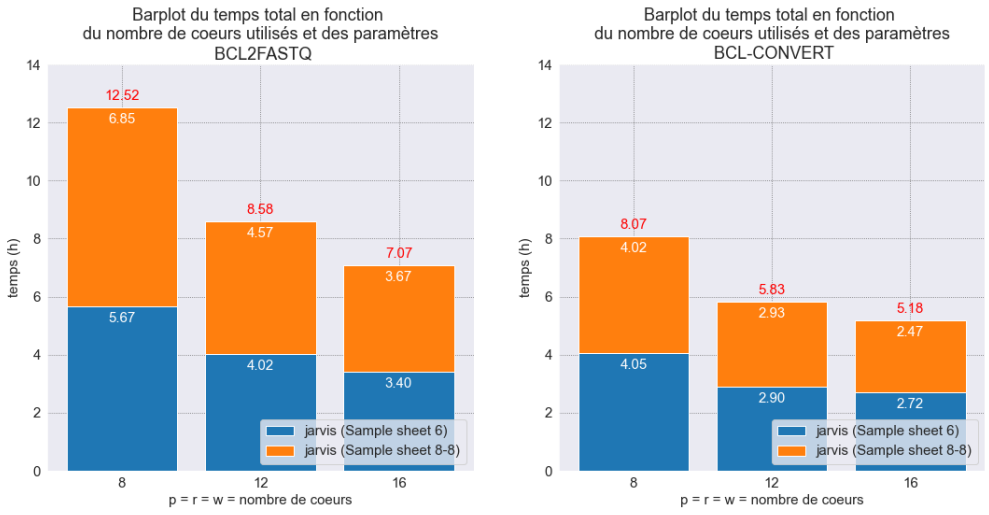


FIGURE 7 – Temps total de génération des FASTQ pour bcl2fastq et bcl-convert

J'ai également échangé avec le service technique d'Illumina à propos des fichiers de sortie et de l'arborescence de ces 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 si l'on pouvait obtenir une arborescence similaire à bcl2fastq, pour minimiser l'impact du changement de logiciel sur les pipelines. Le changement de bcl2fastq, qui sera bientôt obsolète, par bcl-convert va donc nous obliger à réaliser de gros changements dans tous les pipelines qui utilisent ces fichiers de sortie.

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 et de favoriser le temps d'exécution.

J'ai consigné l'intégralité des changements entre les deux logiciels dans un cahier des charges. Il contient, la nouvelle commande à lancer, les modules à charger dans l'environnement, le chemin relatif des fichiers de sorties et leurs descriptions, ainsi qu'un exemple d'arborescence des fichiers de sorties. Ce qui permettra 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 de l'équipe, qui sera en charge de réaliser cette migration.

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

Le pipeline NGS_RG_MGI que j'ai développé à pour objectif 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 dans NGL_BI, à l'aide de la librairie Perl permettant d'interagir avec celle-ci (cf. 3.5 page 7). Les différentes étapes du pipeline suivent le schéma (figure 8) que l'on a définit ci-dessous.

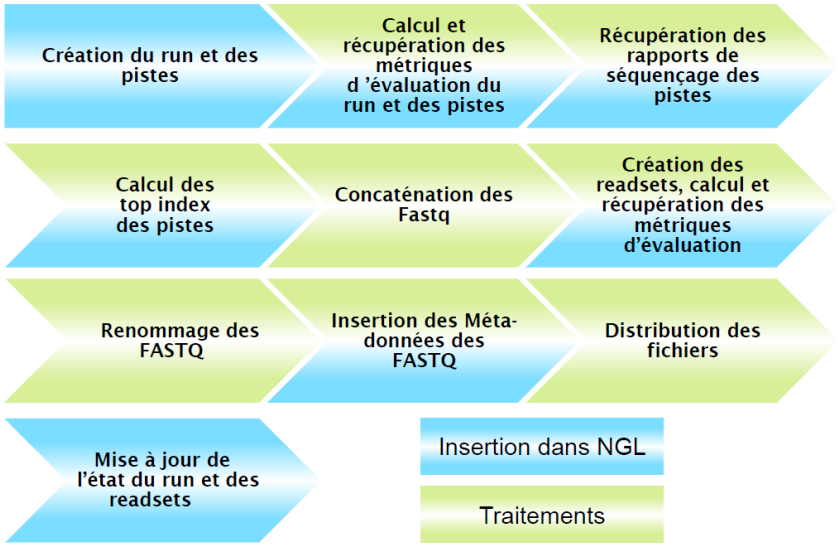
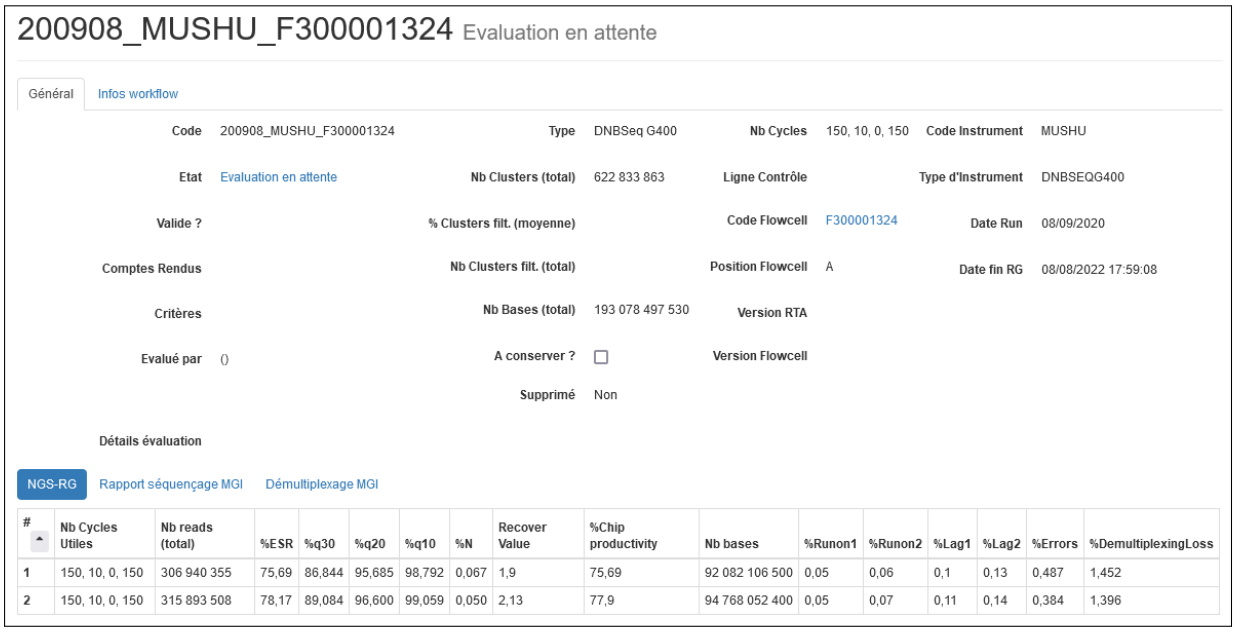


FIGURE 8 – Schéma des différentes étapes du pipeline NGS_RG_MGI. Les étapes qui interagissent avec NGL sont en bleu et les étapes demandant un traitement informatique des données de séquençage sont en vert

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

La première étape du pipeline, que je développé, consiste à créer le run et ses pistes dans la base de données NGL, en y insérant les métriques permettant d'évaluer le run et les pistes (figure 9). Le nom du run est constitué de la date de séquençage, du nom du séquenceur et de l'identifiant de la flowcell du run ce qui le rend unique.

Les différentes métriques sont insérées à l'aide des librairies Perl permettant d'interagir avec ngl en postant ses métrique dans le fichier JSON du run de la base de données, ce qui permet l'affichage de ces dernières dans l'interface web de NGL_BI. Toutes ses métriques sont détaillées plus précisément en annexes (page 25).



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 9 – 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

J'ajoute ensuite, en seconde étape, les rapports de séquençage des pistes que le séquenceur génère en fin de séquençage. Il s'agit de rapports html qui contiennent plusieurs tableaux de métriques et de graphiques permettant d'évaluer les pistes du run. Il y a notamment les graphiques de la distribution de la qualité moyenne en fonction des cycles (figure 10.A), de la distribution des bases nucléiques en fonction des cycles (figure 10.B), de la distribution du pourcentage de Guanine/Cytosine en fonction des cycles (figure 10.C), de la distribution de l'intensité brut au cours des cycles (figure 10.D). Les tableaux et graphiques de ces rapports de séquençage permettent de faciliter l'évaluation du run et de ses pistes.

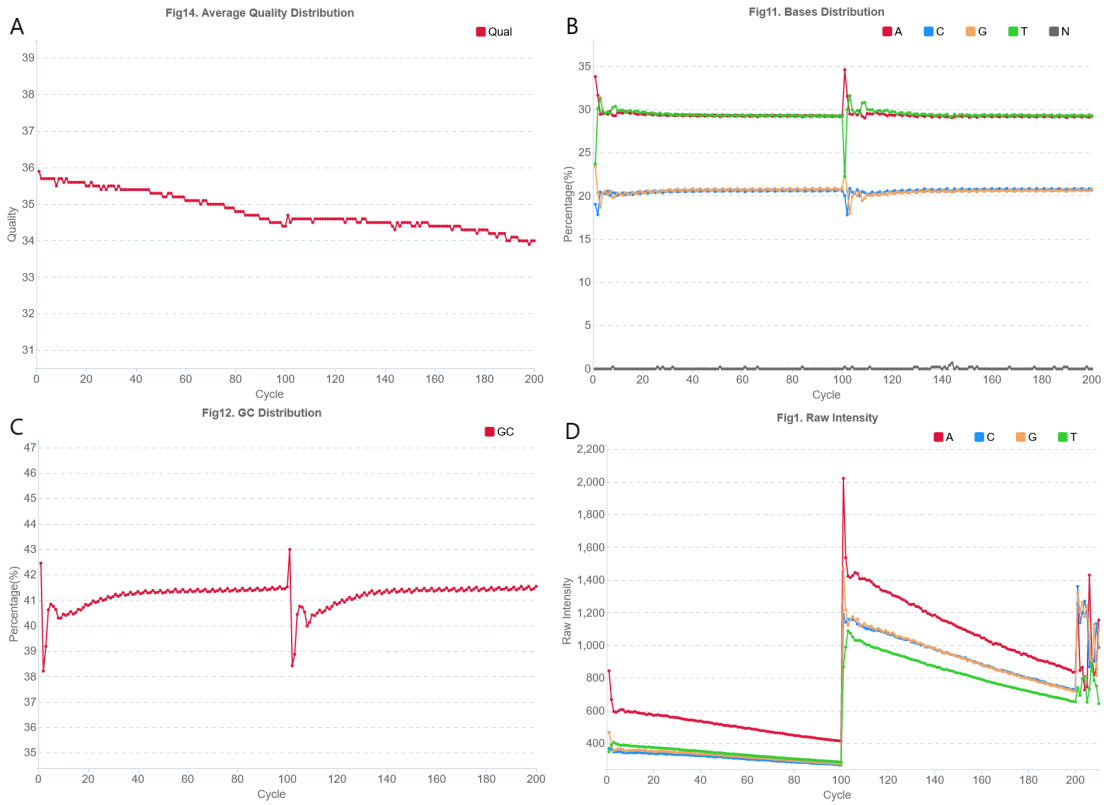


FIGURE 10 – Graphiques des distributions de la qualité moyenne (A), des bases nucléiques (B), du pourcentage de GC (C) et de l'intensité brut (D) au cours des cycles de séquençage

Toujours dans l'optique de facilité l'évaluation du run et de ces pistes, l'étape suivante du pipeline, à été d'ajouter la liste des index représentées à plus de 0.01% de la piste, ainsi que les index attendus. Ces index sont triés et affichés par ordre décroissant dans NGL (figure 11). 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 11 – Capture d'écran de la page du run 200908_MUSHU_F300001324 de NGL en cours de génération de fichiers de séquences (onglet « démultipléxage MGI »)

Concaténation des fichiers FASTQ d'un même readset

Ensuite la quatrième étape du pipeline que j'ai développé à pour objectif d'obtenir un seul fichier FASTQ par readset. En effet la technologie MGI requiert une homogénéité en composition en base nucléiques (A, T, C, G) au niveau de chaque cycle des index, un déséquilibre étant susceptible d'entraver la récupération du signal pour les bases de ces cycles et donc de fausser leur *base calling* puis le démultipléxage des séquences. Ainsi il est recommandé de « barcoder » les échantillons avec 4 barcodes (index), dont les séquences assurent une composition égale en A, T, C et G à chaque position des barcodes. Dans ce cas, nous obtenons donc plusieurs fichiers par échantillons qu'il faut donc fusionner pour en obtenir qu'un seul par échantillon lors du démultipléxage effectué par le séquenceur.

Si le readset est associé à un seul readset alors on réalise une décompression du fichier FASTQ, à l'inverse si il est associé à plusieurs index on réalise une décompression et une concaténation des fichiers FASTQ (cf. figure 12).

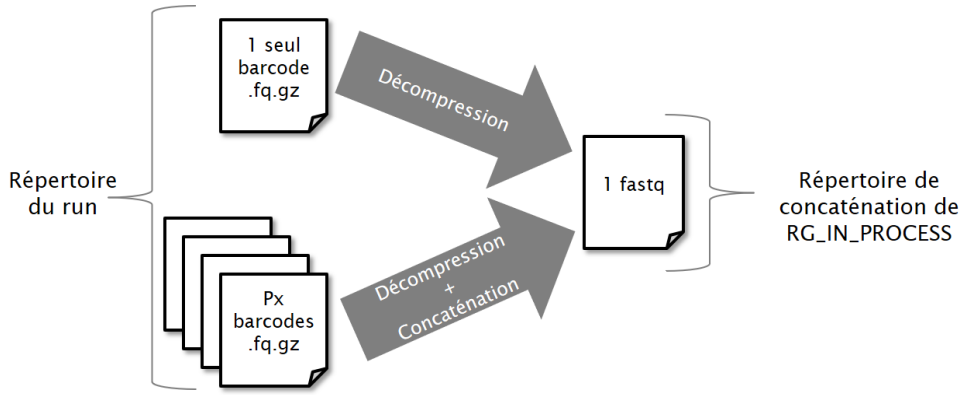


FIGURE 12 – Schéma de l'étape de « concaténation » des fichiers FASTQ d'un readset

Création et insertion des métriques des readset du run dans NGL

La cinquième étape a pour objectif de permettre l'évaluation des readsets, en les créants et en insérant les métriques d'évaluation de ces derniers, récupérer en parssant des fichiers ou/et en les calculant, dans NGL (figure 13). On y retrouve notamment 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. J'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, etc. Toutes ces métriques sont décrites en annexes (page 26)

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 ce qui le rend unique également.

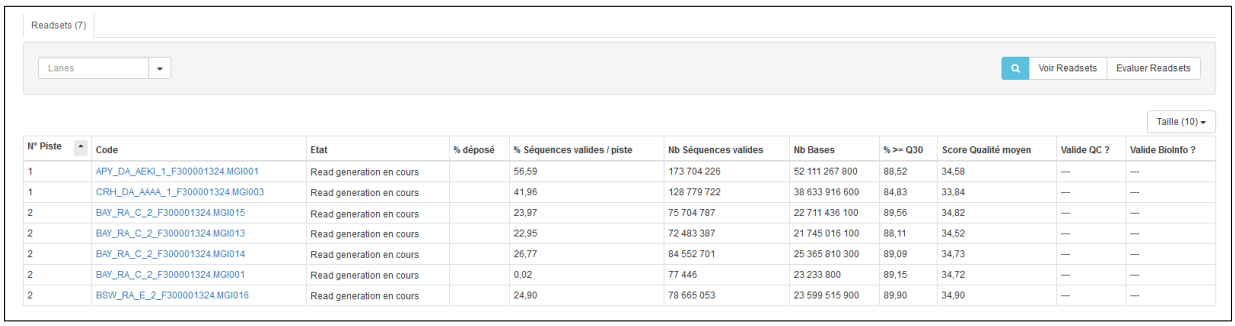
FIGURE 13 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de reads (étapes de création du readset et d'insertion de ces métriques d'évaluation)

L'étape suivante du pipeline est d'ajouter la répartition des index au sein d'un readset (figure 14), ce qui permet de vérifier la composition en index du readset et de vérifier l'homogénéité de ces index au sein du readset.

Index	Nb occurrences	% de cet index dans le readset
barcode2	89 597 340	51,580
barcode1	84 106 886	48,420

FIGURE 14 – 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 à partir des métriques que j'ai ajouté dans le fichier JSON du readset. (figure 15).



N° Piste	Code	Etat	% déposé	% Séquences valides / piste	Nb Séquences valides	Nb Bases	% >= Q30	Score Qualité moyen	Valide QC ?	Valide BioInfo ?
1	APY_DA_AEKI_1_F300001324.MGI001	Read generation en cours		56,59	173 704 226	52 111 267 800	88,52	34,58	---	---
1	CRH_DA_AAAA_1_F300001324.MGI003	Read generation en cours		41,96	128 779 722	38 633 916 600	84,83	33,84	---	---
2	BAY_RA_C_2_F300001324.MGI015	Read generation en cours		23,97	75 704 787	22 711 436 100	89,56	34,82	---	---
2	BAY_RA_C_2_F300001324.MGI013	Read generation en cours		22,95	72 483 387	21 745 016 100	88,11	34,52	---	---
2	BAY_RA_C_2_F300001324.MGI014	Read generation en cours		26,77	84 552 701	25 365 810 300	89,09	34,73	---	---
2	BAY_RA_C_2_F300001324.MGI001	Read generation en cours		0,02	77 446	23 233 800	89,15	34,72	---	---
2	BSW_RA_E_2_F300001324.MGI016	Read generation en cours		24,90	78 665 053	23 599 515 900	89,90	34,90	---	---

FIGURE 15 – 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 septième étape, consiste à renommer les fichiers de séquences des readsets et d'insérer les méta-données de ces derniers dans NGL via le fichier JSON du readset (figure 17). Le renommage des fichiers est nécessaire pour que chaque fichiers de séquences aient un nom unique et « parlant ».

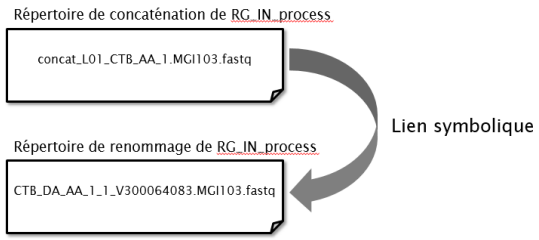


FIGURE 16 – Schéma de l'étape de renommage des fichiers FASTQ d'un readset

Le nom doit permettre d'identifier rapidement et simplement de quel projet, échantillon, flowcell, ect. appartiennent les fichiers. Le renommage des fichiers est effectués en créant un lien symbolique des fichiers obtenus à l'étape de « concaténation » dans un répertoire temporaire (cf figure 16).

Les méta-données des fichiers de séquences du readset que j'insère 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 (« raw », « clean ») et s'ils sont « utilisable », c'est à dire si ils représentent le niveau de qualité maximale disponible pour les échantillons en question. On y retrouve donc le chemin vers le répertoire 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 des valeur de la qualité (Pour les séquenceurs MGI l'encodage est en Phred +33).

L'encodage Phred +33, signifie que l'encodage des valeurs de la qualité sont encodées à partir du caractère « ! », qui vaut 33 en base 10 dans la table ASCII[27] pour une valeurs de qualité de 0. Il y a 42 niveaux de valeurs de qualité, 41 étant le niveau maximale de qualité d'une base nucléique. L'encodage de la qualité en Phred +33, est donc encodé à partir du caractère « ! »jusqu'au caractère « J », qui vaut 74 en base 10 dans la table ASCII et représente la valeur maximale de la qualité d'une base.

Un score qualité de 10 indique qu'il y a un probabilité de 90% que le *base call* soit correct (1 risque sur 10 que la base soit incorrect) et un score de 40 indique une probabilité de 99,99% que le *base call* soit correct (1 risque sur 10000 que la base soit incorrect).

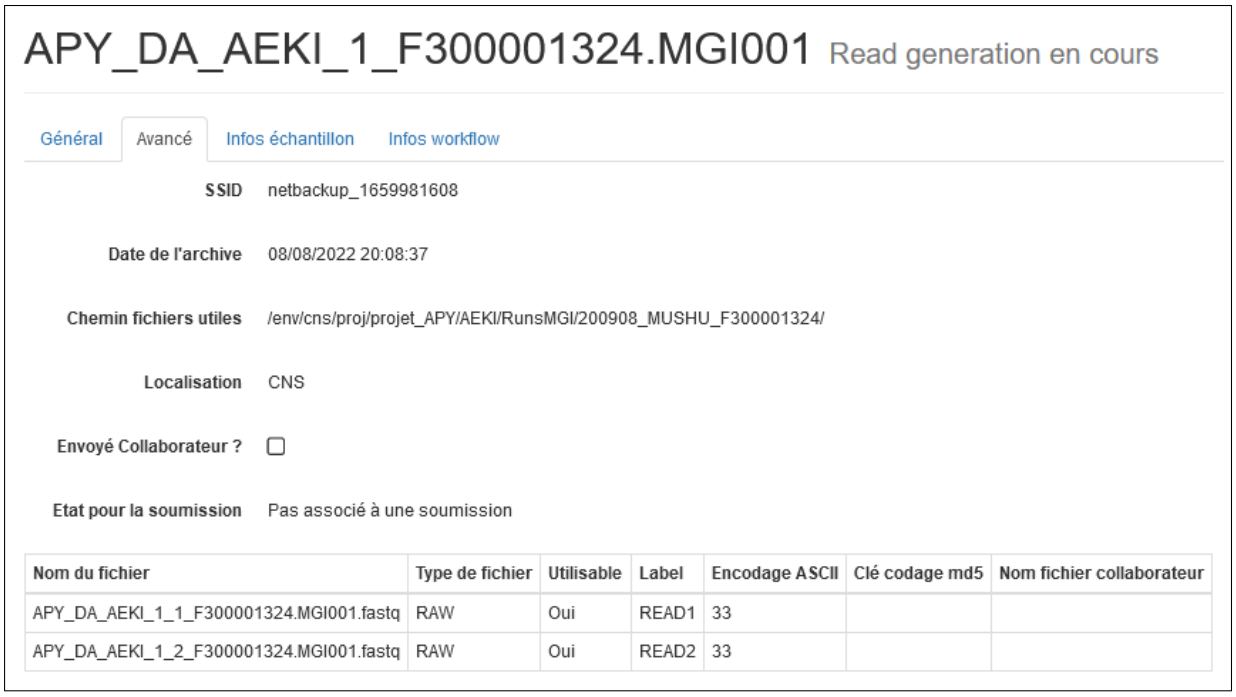


FIGURE 17 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de fichiers de séquences (Onglet « Avancé »)

Distribution des Fichiers séquences et des fichiers de statistiques

La huitième étape que j'ai développé, permet de distribuer des fichiers de séquences « attendus », les fichiers de statistiques du run et les fichiers de séquences « non attendus » dans leurs répertoires dédiés.

Les fichiers de séquences « attendus » sont copiés vers leur répertoire final en fonction du centre dans lequel le séquençage a eu lieu (Genoscope, CNRGH) et leurs droits d'accès sont modifiés pour que les utilisateurs aient le droit de lecture, mais n'aient pas les droits d'écriture et d'exécution.

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

Concernant les fichiers de séquencer « non attendus », il s'agit des fichiers de séquences des index ne faisant pas partie d'un readset. Puisque lors du démultipléxage par les séquenceurs on obtient un fichier FASTQ par index. Ces fichier sont renommés, archivés, compressés et leurs droits sont changés pour les mêmes raisons que les fichiers de séquences « attendus », avant d'être distribués vers leur répertoire dédiés. Ces fichiers de séquences sont conservés dans l'éventualité d'une mauvaise déclaration d'index par les équipes de séquençage, 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 readsets dans l'état de « F-RG », correspondant à la fin du pipeline NGS_RG. De plus, les runs n'étant plus dans un état "IW-RG" ou "IP-RG", correspondant à l'attente de prise en charge par NGS_RG une fois le séquençage terminé ou en cours de génération de reads, ils ne seront plus éligibles à une prise en charge par NGS_RG. Cela entraine donc une mise à jour automatique du run à l'état « IW-V », correspondant à l'attente de validation, ce qui permet d'indiquer aux utilisateurs que le run peut être évalué. Les readsets sont aussi automatiquement mis à jour vers l'état « IW-QC », correspondant à l'attente de prise en charge par le pipeline NGS_QC, ce qui permettre d'indiquer au pipeline de contrôle qualité qu'il peut effectuer le contrôle qualité des readsets 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

Les pipelines de génération de fichiers de séquences est très spécifique à l'environnement de gestion des projets de séquençage du Genoscope et du CNRGH, ainsi qu'à la base de données NGL. Celui pour la technologie MGI est inspiré du pipeline de l'autre technologie de séquençage *short reads* (Illumina) déjà en place, dont la finalité du pipeline est le même, c'est à dire la prise en charge des données de séquençage en fin de séquençage et la mise à disposition des fichiers de séquences et des métriques d'évaluation.

La future amélioration du pipeline NGS_RG_MGI, consistera à la mise en place d'une étape supplémentaire pour les runs qui comporterons des *mids*[28]. Un *mid* est une séquence d'une dizaines de nucléotides ajoutés en amont du primer du read *forward* (figure 18). Il s'agit d'un index supplémentaire qui permet lors du séquençage de déposer un nombres plus important d'échantillons différents sur une même piste d'une flowcell.

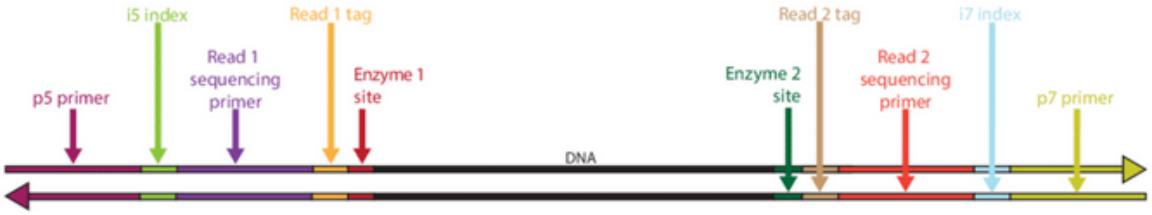


FIGURE 18 – Schéma représentant le vecteur à séquencer qui contient les index (i5 index et i7 index), les mids (Read 1 tag et Read 2 tag), les primers des index (p5 primer et p7 primer) et des reads (Read 1 sequencing primer et Read 2 sequencing primer), ainsi que l'ADN d'intérêt (en noir) et les sites enzymatiques permettant l'insertion de l'ADN d'intérêt dans le vecteur (Enzyme 1 site et Enzyme 2 site).

L'étape supplémentaire sera d'ajouter est un second démultipléxage en fonction de ces mids, qu'on appelle le démidage pour la création des readset et des 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, sera constitué de deux grande phases. La première consistera à réaliser un contrôle qualité des fichiers « raw » (fichiers de séquences bruts), puis dans un second temps de réaliser un contrôle qualité sur les fichiers « clean » (fichiers de séquences nettoyés) ainsi que d'autres traitements sur les fichiers « clean ».

Le pipeline devra prendre en charge automatiquement les fichiers dont les readset sont dans l'état « IW-QC » dans NGL, qui représente les readset en attente de contrôle qualité. Celui-ci devra suivre le schéma de traitements et d'analyse (figure 19) déjà en place pour l'autre technologie de séquençage *short reads* (Illumina). L'étape de nettoyage du *PhiX* ne sera pas nécessaire pour la technologie MGI, puisque pour cette dernière il n'est pas utile d'ajouter ces séquences de contrôle.

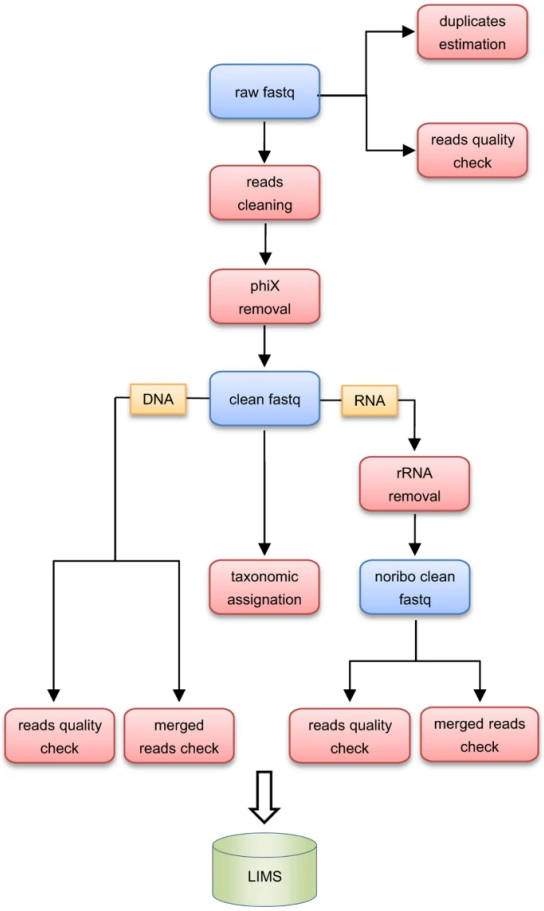


FIGURE 19 – Schéma des étapes du pipeline NGS_RG_Illumina

Lors de la première phase on réalisera dans un premier temps un échantillonnage de 20000 séquences par fichiers « raw ». Cela permettra d'améliorer le temps d'exécution du contrôle qualité, tout en ayant une grande représentativité de la qualité des séquences des fichiers « raw ».

Ensuite on réalisera le « Trimming » (nettoyage des séquences) des fichiers « raw », avant de passer à la seconde phase du pipeline pour obtenir les fichiers « clean ».

Cette seconde phase commencera par un échantillonnage de 20000 séquences par fichiers « clean », toujours dans l'optique d'améliorer le temps d'exécution du contrôle qualité et des autres traitements sur les fichiers « clean ».

Le Trimming consistera à retirer les séquences qui ont une qualité moyenne faible (inférieur à 20), les dernières bases des séquences qui ont une faible qualité (inférieur à 20) seront retirer de la séquence. Si il y a trop de bases inconnus dans la séquence celle-ci sera retirée, de même si dans la séquence ont retrouve 3 ou plus de bases inconnus successivement dans la séquence celle-ci sera coupé jusqu'à la fin de la séquence.

Les différentes étapes qui seront à réaliser pour la première phase sont :

- Réaliser le contrôle qualité des séquences des fichiers brut, avec l'outil fastx_clean

de l'extention fastxend de la suite FASTX Toolkit développé par le Genoscope.
- Réaliser l'estimation des duplicats des séquences des fichiers brut, avec l'outil fastx_estimate_duplicate de l'extension fastxend de la suite FASTX Toolkit développé par le Genoscope.

Les différentes étapes qui seront à réaliser pour la seconde phase, une fois le Trimming effectué sont :

- Réaliser le contrôle qualité des séquences des fichiers clean, avec l'outil fastx_clean.
- Réaliser l'estimation des duplicats de séquences des fichier clean, avec l'outil fastx_estmate_duplica
- Réaliser l'assignation taxonomique des séquences des fichiers clean, à l'aide du logiciel Centrifuge.
- Réaliser l'alignement des séquences des fichiers clean sur un génome de référence si celui-ci est disponible, à l'aide de l'outil BWA.
- Réaliser le « *merging* » des séquences des reads *foward* et *reverse* pour les run *pair end* (calcule du pourcentage de reads qui ont le read *forward* et *reverse* qui se chevauchent, calcule de la moyenne et médiane du nombre de bases qui ce chevauchent entre les 2 reads, ...), à l'aide de l'outil fastx_mergepairs de l'extention fastxend de la suite FASTX Toolkit développé par le Genoscope.

Une fois les étapes de la seconde phase réalisées, il sera nécessaire de réaliser la distribution des fichiers de séquences nettoyés dans leur répertoire final. Les fichiers raw seront alors effacés si ces derniers ont été archivé sur bande magnétique, et rendu indisponible pour les utilisateurs. L'objectif étant de permettre aux utilisateurs d'avoir les fichiers de séquences avec le meilleurs niveau de qualité possible.

Durant toutes les étapes du pipeline, on insert les métriques et graphiques obtenus au cours des différentes étapes du pipeline dans NGL_BI. Ce qui permettra de réaliser la validation des readset ou non. L'interaction entre le pipeline et la base de données est réalisée, comme pour le pipeline NGS_RG_MGI, par la librairie Perl (DBFactory) qui permet d'interagir avec NGL (cf. 3.5 page 7).

L'objectif est d'obtenir un pipeline de contrôle qualité opérationnel le plus rapidement possible, c'est pour cela que les outils utilisés seront les même que ceux utilisés pour le pipeline NGS_QC_Illumina. Néanmoins, les outils utilisés évoluerons au fil du temps, avec les évaluations d'outils pour les pipelines de contrôle qualité que je réaliserais au cours de l'année à suivre.

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, néanmoins 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 également 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 taxinomique 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 permettant de prédire la fonction d'un gène, d'une molécule, d'une région de l'ADN, ...

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

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

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

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

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

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

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

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

[11]Lot de séquences

[12]Séparation des séquences en plusieurs fichiers en fonction de leurs index (séquence d'une dizaines de nucléotides en amont du primer de la séquence)

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

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

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

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

[17]*JavaScript Object Notation* est un format de données textuelles structurées et organisées

[18]*application programming interface* est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités

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

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

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

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

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

[24]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.

[25]Fichier contenant les informations et instructions pour la génération des FASTQ et le démultipléxage

[26]Séquence d'une dizaines de nucléotides en amont du primer de la séquence d'ADN à séquencer, permettant de séparer les séquences de plusieurs échantillons sur une même piste de la flowcell (démultipléxage)

[27]*American Standard Code for Information Interchange* est une norme de l'encodage des caractères

[28]un mid (*molecular identifer*) est une séquence d'une dizaine de nucléotides ajoutés en aval du *primer* du read *forward* permettant de réaliser un second démultipléxage

Références

[1] BCL Convert.

[2] BCL Convert Software Guide v3.7.5 (1000000163594). page 22.

[3] bcl2fastq2 Conversion Software v2.20 Software Guide (15051736). page 27.

[4] The Comprehensive Perl Archive Network - www.cpan.org.

[5] perl - The Perl 5 language interpreter - Perldoc Browser.

[6] BGI goes head-to-head with Illumina. *Nature Biotechnology*, 33(8) :792–792, Aug. 2015.

[7] S. Anslan, V. Mikryukov, K. Armolaitis, J. Ankuda, D. Lazdina, K. Makovskis, L. Vesterdal, I. K. Schmidt, and L. Tedersoo. Highly comparable metabarcoding results from MGI-Tech and Illumina sequencing platforms. *PeerJ*, 9 :e12254, Sept. 2021.

[8] C. A. Austin-Tse, V. Jobanputra, D. L. Perry, D. Bick, R. J. Taft, E. Venner, R. A. Gibbs, T. Young, S. Barnett, J. W. Belmont, N. Boczek, S. Chowdhury, K. A. Ellsworth, S. Guha, S. Kulkarni, C. Marcou, L. Meng, D. R. Murdock, A. U. Rehman, E. Spiteri, A. Thomas-Wilson, H. M. Kearney, H. L. Rehm, and Medical Genome Initiative*. Best practices for the interpretation and reporting of clinical whole genome sequencing. *npj Genomic Medicine*, 7(1) :27, Dec. 2022.

[9] S. Drmanac, M. Callow, L. Chen, P. Zhou, L. Eckhardt, C. Xu, M. Gong, S. Gablenz, J. Rajagopal, Q. Yang, C. Villarosa, A. Au, K. Davis, A. Jorjorian, J. Wang, A. Chen, X. Zhang, A. Borcherding, X. Wei, M. Zhang, Y. Xie, N. Barua, J. Shafto, Y. Dong, Y. Zheng, L. Wang, L. Zhai, J. Li, S. Liao, W. Zhang, J. Liu, H. Jiang, J. Wang, H. Li, X. Xu, and R. Drmanac. CoolMPS ™ : Advanced massively parallel sequencing using antibodies specific to each natural nucleobase. preprint, Genomics, Feb. 2020.

[10] S. A. Jeon, J. L. Park, S.-J. Park, J. H. Kim, S.-H. Goh, J.-Y. Han, and S.-Y. Kim. Comparison between MGI and Illumina sequencing platforms for whole genome sequencing. *Genes & Genomics*, 43(7) :713–724, July 2021.

[11] H.-M. Kim, S. Jeon, O. Chung, J. H. Jun, H.-S. Kim, A. Blazyte, H.-Y. Lee, Y. Yu, Y. S. Cho, D. M. Bolser, and J. Bhak. Comparative analysis of 7 short-read sequencing platforms using the Korean Reference Genome : MGI and Illumina sequencing benchmark for whole-genome sequencing. *GigaScience*, 10(3) :giab014, Mar. 2021.

[12] J. Patterson, E. J. Carpenter, Z. Zhu, D. An, X. Liang, C. Geng, R. Drmanac, and G. K.-S. Wong. Impact of sequencing depth and technology on de novo RNA-Seq assembly. *BMC Genomics*, 20(1) :604, July 2019.

6 Annexes

Description des métriques d'évaluation 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 9 page 12) :

Nb Cycles Utiles : Nombre de cycles des reads et des index (nombre de cyles pour le read *forward*, nombre de cycles 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 : *Effective spot rate* ((nombre de reads / nombre total de DNB) $\times$ 100)

%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 : Rapport d'intensité entre le read *forward* et *reverse*

%Chip productivity : Pourcentage de productivité de la piste (nombre reads qui passent un pré-filtre MGI / nombre total de DNB)

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 écartés lors du démultipléxage

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 13 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 15 page 16) :

%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 20).

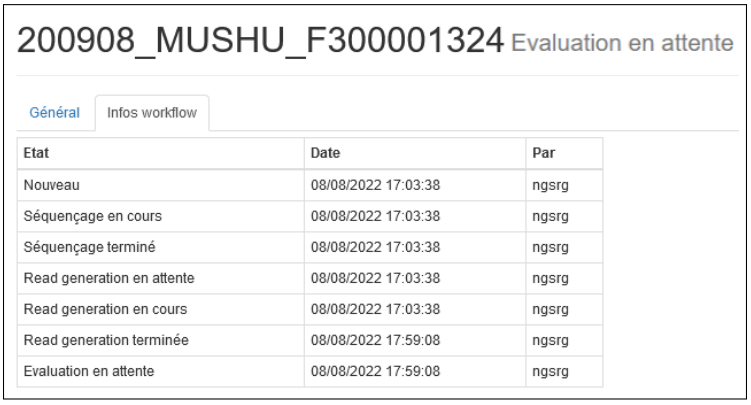


FIGURE 20 – Capture d'écran de la page du run 200908_MUSHU_F300001324 de NGL en cours de génération de fichiers de séquences (onglet « infos workflow »)

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 21) et de l'onglet « Infos workflow » (figure 22). 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

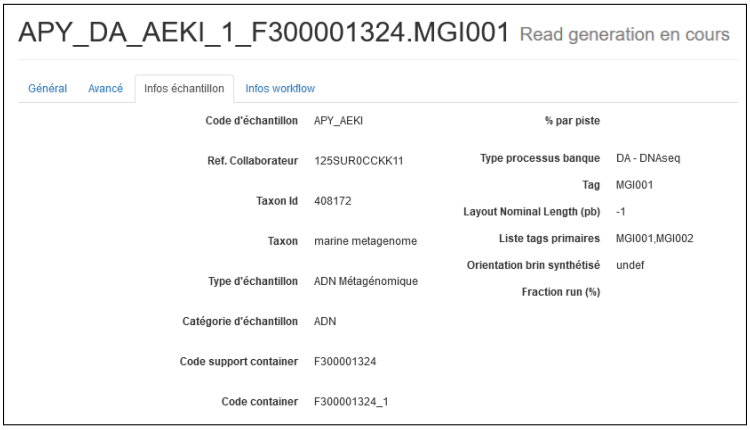


FIGURE 21 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de reads (onglet « Infos échantillon »)

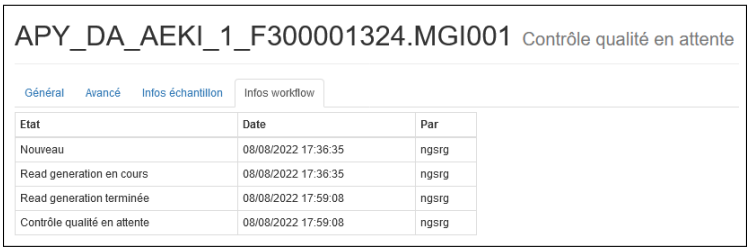


FIGURE 22 – Capture d'écran de la page du readset APY_DA_AEKI_1_F300001324.MGI001 de NGL en cours de génération de reads (onglet « Infos workflow »)