Reproduction via workflow Nextflow et conteneurs Docker d'une analyse RNA-seq de cancer oculaire

---

Klaus von Grafenstein, Virginie Noël, Arnaud Maupas | UE ReproHackathon | 12/11/2021

I) Introduction

1) Reproductibilité

La reproductibilité est la capacité de pouvoir refaire et donc reproduire une expérience ou analyse pour obtenir les résultats présentés dans un article scientifique donné. La reproductibilité se place au centre de la méthodologie scientifique avec l'observation empirique et les citations.
Or, Nekrutenko et Taylor [1] montraient en 2012 que sur 50 articles publiés en 2011 sur des analyses NGS, plus de la moitié ne donnaient pas d'information sur la version des outils informatiques utilisés et leurs paramètres, ainsi que sur les détails des données génomiques. Cela avait permis de révéler un véritable problème concernant la reproductibilité dans les études bio-informatiques, mais celui-ci avait déjà été relevé dans d'autres domaines, comme par exemple la taxonomie en 2008 [2].

La reproductibilité existe sous plusieurs formes, chacune de ses formes étant définie sur des parties différentes du processus de recherche scientifique [3]. Premièrement, la reproductibilité empirique repose sur le fait de fournir des détails concrets sur les expériences non faites *in silico* et sur les observations. Deuxièmement, la reproductibilité statistique est basée sur le fait de donner les informations détaillées sur les choix des tests statistiques et méthodes ainsi que sur le choix des paramètres et hyperparamètres de ceux-ci et sur la façon dont la précision de ceux-ci est obtenue. Troisièmement, la reproductibilité informatique est centrée sur le fait de donner les informations sur les outils informatiques utilisés, comme leur version ou paramètres, ainsi que l'environnement de travail où ces outils ont été lancés.

2) Objectifs

En février 2013, l'article [4] de Harbour et Roberson montrait une présence récurrente de mutations du facteur d'épissage SF3B1 au codon 625 chez des individus atteints de cancers oculaires de type léger. Bien qu'ils en aient déduit que cette mutation affectait probablement l'épissage alternatif et avait un lien avec le développement du cancer, cela ne fut pas observé avant juillet 2013, lorsque Furney et Pedersen ont repris les données de Habour et Roberson, et montré dans leur article [5] l'impact de cette mutation sur l'épissage alternatif.

Le but de ce projet est de reproduire ces études scientifiques portant sur des données RNA-Seq d'individus avec un cancer oculaire, en regardant si les individus ayant la mutation SF3B1 ont des gènes différemment exprimés de ceux ayant également ce cancer, mais pas la mutation. Pour cela, la mise en place de principes de reproductibilité a été nécessaire, avec l'utilisation d'un outil de création de workflow (Nextflow) pour que notre analyse puisse être facilement relancée, un outil de conteneurs logiciels (Docker) pour éviter les problèmes de dépendance, et un outil de gestion de version (Git) pour travailler en groupe et avoir un historique de notre travail.

3) Outils utilisés

a) Nextflow, création de workflow

Avec l'avènement de grands jeux de données en biologie, une forte demande d'outils permettant d'améliorer la reproductibilité informatique a eu lieu, comme vu précédemment. Certains des outils populaires de création de workflow, comme Galaxy [6], présentent une interface graphique relativement simple, rendant la création et la gestion de workflows accessible à un public de chercheurs en biologie néophyte. Cependant, dans la pratique, ces outils présentent moins de flexibilité et de performance que les gestionnaires de workflow basés sur une interface par ligne de commande (comme Snakemake, Nextflow…), qui sont par conséquent souvent préférés par les bio-informaticiens plus expérimentés.

Dans le cadre de notre analyse de données RNA-Seq, nous avons choisi d'utiliser un des outils de création de workflow les plus populaires fonctionnant par ligne de commande, Nextflow [7]. Le développement de Nextflow est supervisé par Paolo Di Tommaso (Project Lead, Seqera Labs) et Evan Floden (Seqera Labs CEO). Cet outil est écrit en Groovy, un langage proche de Java. Le fonctionnement de Nextflow est axé autour de process, des processus de base qui vont exécuter des scripts. Pour chaque process, on peut définir des inputs et outputs, qui peuvent être des fichiers, variables, etc. Les process peuvent ensuite communiquer entre eux au moyen de channels (les outputs de certains process peuvent servir d'input à d'autres, et inversement). De plus, Nextflow conserve une trace de l'avancement des différents process en temps réel, et stocke les fichiers intermédiaires dans un répertoire indépendant (répertoire work), ce qui permet de pouvoir relancer les analyses à partir d'une étape donnée si nécessaire. Nextflow fonctionne avec un fichier de configuration, dans lequel il est notamment possible de renseigner si l'on souhaite utiliser des outils de conteneurisation.

b) Docker, conteneurisation

Les conteneurs fournissent à une application un environnement d'exécution qui contient tous les fichiers et dépendances nécessaires spécifiés lors de sa création. Les conteneurs sont complètement isolés de l'environnement depuis lequel ils sont lancés. En cela, ils permettent d'avoir accès à un environnement reproductible depuis n'importe quel ordinateur, cloud ou infrastructure, quel que soit le système d'exploitation. Une commande lancée dans un conteneur depuis un certain environnement donnera donc exactement les mêmes résultats dans le même conteneur depuis un autre environnement. Les conteneurs sont utilisés pour développer, partager et exécuter des applications. Ils permettent particulièrement d'avoir accès à une version donnée d'un outil, même si cette version n'est plus disponible en ligne. De plus, les conteneurs occupent moins de place qu'une machine virtuelle classique et sont plus rapides à démarrer qu'un véritable système d'exploitation.

L'une des plateformes de conteneurisation les plus répandues est Docker [8]. La conteneurisation consiste à déployer des applications par le biais de conteneurs. Il s'agit d'une pratique de plus en plus courante, car elle facilite grandement le déploiement en évitant à l'utilisateur de gérer lui-même l'installation de l'application, qui peut parfois être compliquée. Singularity est un autre exemple de plateforme de conteneurisation.

Un conteneur Docker est une instance exécutable d'une image Docker. Cette image est elle-même construite grâce aux instructions écrites dans un fichier appelé Dockerfile. Tous les utilisateurs de Docker peuvent exporter leurs images Docker vers une plateforme publique appelée Docker Hub. Des conteneurs existent déjà pour de nombreuses applications et peuvent être retirés depuis cette plateforme. Il est possible de spécifier à un process Nextflow le conteneur Docker dans lequel les commandes doivent être exécutées. C'est cette fonctionnalité que nous utilisons pour ce projet.

c) Git, versionnage

Les logiciels de gestion de version, ou version control systems (VCS), permettent de stocker des fichiers et de conserver la trace des modifications qui leur ont été apportées au cours du développement. Ils permettent ainsi un suivi des différentes versions d'un projet. De plus, il s'agit d'un bon moyen de retracer l'origine d'erreurs dans un code. Ces logiciels jouent un rôle important dans l'amélioration de la reproductibilité en permettant de stocker à un même endroit tous les fichiers relatifs à un projet, qu'il s'agisse des jeux de données, du code pour l'analyse statistique ou des figures produites, tout en limitant le nombre de fichiers produits pour les différentes versions. La plupart des VCS, comme CVS et Subversion, sont centralisés, c'est-à-dire qu'ils gardent la copie originale sur un serveur central. Une connexion à ce serveur et des droits appropriés sont nécessaires pour accéder aux fichiers et pouvoir les modifier.

Git [9], développé par Software Freedom Conservancy, est le VCS le plus utilisé. Il a pour avantages d'être décentralisé, distribué, gratuit et open source. Toute copie d'un répertoire Git peut servir de serveur ou de client, ce qui évite les points de défaillance uniques si un problème devait survenir sur l'une des copies du répertoire. Git permet à un groupe de personnes de travailler de manière asynchrone sur un même projet sans nécessiter de connexion au serveur central. Chaque personne du groupe peut alors envoyer (ou "push") ses modifications vers le serveur à tout moment. En tant qu'outil de versionnage, Git donne accès à toutes les versions des fichiers dans chaque copie du répertoire. GitHub est un service d'hébergement de Git appartenant à Microsoft. Il s'agit aussi du plus grand hébergeur de code source au monde. Dans ce projet, nous utilisons GitHub pour que chaque membre du groupe ait accès à la dernière version du workflow et aux fichiers associés, et puisse travailler dessus de manière asynchrone.

d) Cloud IFB

L'IFB [10] (Institut Français de Bio-informatique) est l'infrastructure nationale de bio-informatique en France. En tant que telle, elle offre des services dans différents domaines pour aider à la recherche et aux projets dans les sciences de la vie et la bioinformatique, dans le privé et le public. Les services proposés comportent l'accès à des données, outils et formations, ainsi que de l'aide pour les projets de recherche liés à la biologie et l'accès à une infrastructure informatique.

Ce dernier point est particulièrement intéressant, car l'IFB propose un cloud sur lequel il est possible de demander des ressources de calcul. Il s'agit plus précisément d'une fédération de clouds (Biosphere) à travers la France qui dispose au total de plus de 6 000

cœurs et 29 To de mémoire vive. Des machines virtuelles préconfigurées ou appliances sont disponibles dans le catalogue RAINBio, mais les utilisateurs peuvent aussi installer leur propre infrastructure.

e) SRA Toolkit

Le SRA Toolkit [11] est une collection d'outils et diverses librairies fournie par le NCBI pour manipuler des données SRA (Sequence Read Archives ou archive de fragment de séquences). Au sein du SRA Toolkit, nous utilisons l'outil fasterq-dump, qui permet de télécharger au format FastQ les fragments de séquences, ou reads, correspondant à un identifiant SRA donné. Cet outil est une alternative récente et supposément plus rapide à l'outil fastq-dump.

f) STAR

STAR (Spliced Transcripts Alignment to a Reference) [12] est un programme d'alignement de séquences RNA-Seq sur un génome de référence. Il permet d'obtenir des fichiers en format BAM. Le format BAM, ainsi que le format CRAM sont des versions compressées binarisées du format SAM, le format CRAM étant le moins volumineux. Ces trois formats, BAM, CRAM et SAM, sont des formats dits de cartographie d'alignement de séquence.

g) SAMtools

SAMtools [13] propose différents outils pour la manipulation d'alignements en format SAM, BAM ou CRAM. SAMtools est capable d'effectuer des conversions entre formats, de trier des fichiers, de les afficher à l'écran, de les fusionner ou encore de les indexer. Nous nous servons ici de la commande samtools index qui permet d'indexer un fichier trié sur les coordonnées, dans notre cas un fichier BAM obtenu en sortie de l'alignement réalisé par STAR.

h) Subread

FeatureCounts [14] est un des outils du package Subread, mais est aussi disponible dans d'autres programmes créés pour traiter des données NGS. FeatureCounts permet d'obtenir le nombre de reads qui s'alignent sur un ensemble de gènes de référence, dans notre cas les gènes annotés du génome humain. Pour cela, featureCounts utilise des techniques de hachage de données ainsi que des techniques statistiques dites "plan en bloc".

i) R et les packages FactoMineR, factoextra, DESeq2 et Enhanced Volcano

Nous avons choisi d'effectuer nos analyses statistiques avec R [15]. R est à la fois un langage de programmation et un logiciel libre, développé depuis 1993 par R Core Team. R peut fonctionner avec des extensions nommées packages contenant du code, des données et

de la documentation, utilisées afin de réaliser certaines analyses (manipulation de données, test statistiques, représentations graphiques...).

Dans le cadre de notre projet, nous avons utilisé R pour réaliser des analyses de type ACP (Analyse en Composante Principale) sur les valeurs d'expression génétiques pour les individus des groupes mut (présentant un cancer de type 1, avec mutations du facteur d'épissage SF3B1 au codon 625) et WT (présentant un cancer de type 1, sans mutations du facteur d'épissage SF3B1 au codon 625) afin d'observer s'il y a des clusters ou des valeurs aberrantes (outliers) dans le jeu de données. Ensuite, nous avons voulu comparer les valeurs d'expression entre les individus des groupes mut et WT, afin d'observer s'il y a des gènes différentiellement exprimés entre ces groupes.

Nous avons utilisé FactoMineR [16] et factoextra [17] pour l'ACP et DESeq2 [18] et EnhancedVolcano [19] pour l'analyse des gènes différentiellement exprimés. Le package FactoMineR permet de réaliser l'ACP à partir du jeu de données, et d'inclure le groupe de l'individu comme une variable illustrative. A partir de cela, nous obtenons des représentations graphiques à l'aide du package factoextra, dans lesquelles les individus sont placés sur un plan défini par les deux composantes principales expliquant au mieux les variations dans le jeu de données. Les individus sont colorés en fonction de leur groupe, et des ellipses sont ajoutées afin de montrer la forme globale de la dispersion des individus pour chaque groupe.

Pour l'analyse des gènes différentiellement exprimés, DESeq2 utilise une méthode basée sur la distribution binomiale négative. Les valeurs d'expression en input sont "brutes", et sont directement normalisées par les fonctions du package lors de l'analyse. Nous avons choisi comme seuil de significativité 0.05 et comme seuil de valeur absolue de log2FoldChange 1 (le fold-change est le rapport du niveau moyen d'expression d'un gène dans une condition par rapport à une autre). Avec le package EnhancedVolcano, nous avons ensuite obtenu des Volcano plot pour nos données. Il s'agit de graphiques de type "nuages de point" dans lequel le −Log10 de la p-value est représenté en fonction du Log2 fold change. Les transcrits d'intérêt (avec une p-value ajustée inférieure à 0.05 et un log2FoldChange inférieur à -1 ou supérieur à 1) sont représentés en rouge.

II) Organisation

Les premières séances de travail avaient principalement lieu lors des cours en présentiel et lors de diverses heures de pause dans notre emploi du temps. Lorsque nous ne nous réunissions pas en présentiel, les réunions avaient lieu par appel sur le logiciel Discord. Le communication liée au projet avait lieu sur des canaux de discussion Discord et via la messagerie Messenger.

Pour la phase de codage du workflow, tout d'abord, Klaus s'est chargé de l'écriture des premiers process sur Nextflow (downloadFastQ, downloadChr, downloadGtf) avec de l'aide de débogage de la part de Virginie. Ensuite, il y a eu inversion des rôles puisque Virginie a fait la suite des process (indexGenome, mapFastQ, indexBamFiles, countReads) et Klaus a apporté son aide. En parallèle, Arnaud a codé le script R pour les analyses statistiques et représentations graphiques, ainsi que le process qui lance celui-ci.
Ensuite pour la phase d'écriture du rapport et de la documentation, nous avons décidé de l'organisation suivante :

- Klaus était chargé de l'introduction sur la reproductibilité, de définir les objectifs du projet, de décrire le workflow, le jeu de données, les outils SRA Toolkit, STAR et Subread, ainsi que de résumer l'organisation du projet, d'écrire le readme, et finalement de donner nos retours sur le projet.
- Virginie a quant à elle rédigé les parties à propos de Docker, Git, le cloud utilisé et l'outil SAMtools, en plus d'avoir abordé les difficultés rencontrées, les versions des outils et l'environnement de travail. Une autre de ses missions était également d'être la correctrice orthographique du rendu.
- Enfin, Arnaud s'est consacré à la description de Nextflow, du langage de programmation R et des divers packages utilisés, ainsi qu'aux résultats des analyses statistiques et à leur interprétation, puis aux conclusions amenées par ces résultats. Il a également mis en forme les figures.

III) Résultats

1) Workflow

Au début du workflow, le process DownloadFastQ télécharge les reads au format fastQ à partir des identifiant SRA contenus dans le fichier SRAid.txt. En parallèle, les process downloadChr et downloadGtf téléchargent respectivement le génome humain de référence GRCh38 et les données d'annotation associées. Le génome est alors indexé avec STAR dans le process indexGenome. Ensuite, dans mapFastQ, les reads sont mappés sur ce génome avec l'outil STAR également. Après cela, le process indexBamFile indexe comme son nom l'indique les fichiers bam d'alignement obtenus, et le process countReads associe et compte le nombre de reads qui correspondent à des gènes de l'annotation, à l'aide de l'outils featureCounts. Avec ces données ainsi qu'un csv qui contient l'association entre les identifiants SRA et leur groupe, une analyse statistique est faite sur R et plusieurs figures et fichiers sont produits en sortie.

Le flowchart ci-dessous (**Fig.1**) résume visuellement ces différentes étapes du workflow. Il a été généré avec l'option Nextflow -with-dag flowchart.png, qui nécessite d'avoir préalablement installé l'outil graphviz.

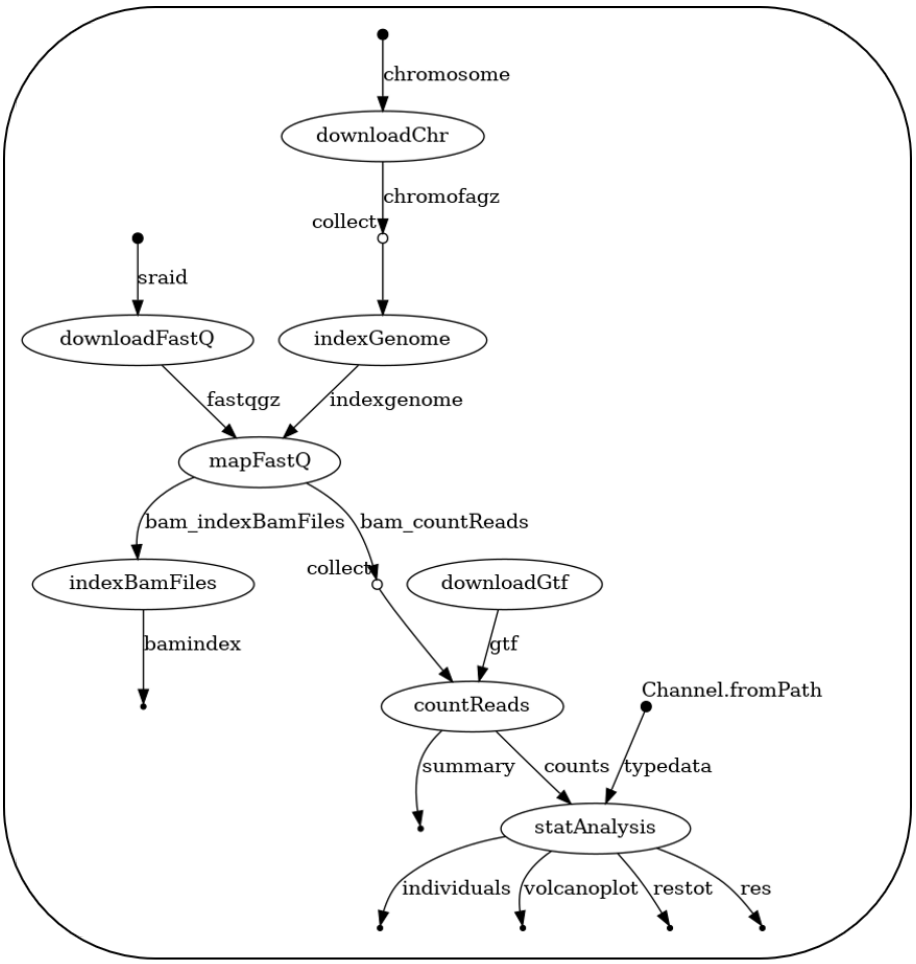

Figure 1- **Flowchart du workflow.** *Les différentes étapes du workflow sont représentées schématiquement.*

Au total, il faut 3h07min50s pour faire tourner le workflow avec tous les identifiants SRA.

2) Résultats des analyses statistiques

Pour l'ACP, on observe que les composantes principales 1 et 2 expliquent respectivement 33,2% et 19% de la variabilité observée pour les niveaux d'expression des transcrits entre les individus. A elles seules, elles expliquent donc moins de 50% de la variabilité, et il est donc délicat de tirer des conclusions trop rigides sans prendre en compte le rôle d'autres composantes.

Si l'on observe la représentation graphique obtenue avec factoextra (**Fig.2**), on observe que les individus du groupe 2 sont particulièrement groupés. Cependant, il est difficile de conclure à partir de cette PCA que ces deux groupes ont des profils d'expression clairement différents, puisque les points représentant chaque individu ne forment pas 2 clusters distincts sur le graphique. Par ailleurs, on ne repère pas d'individu « outlier » particulièrement notable dans le jeu de données.

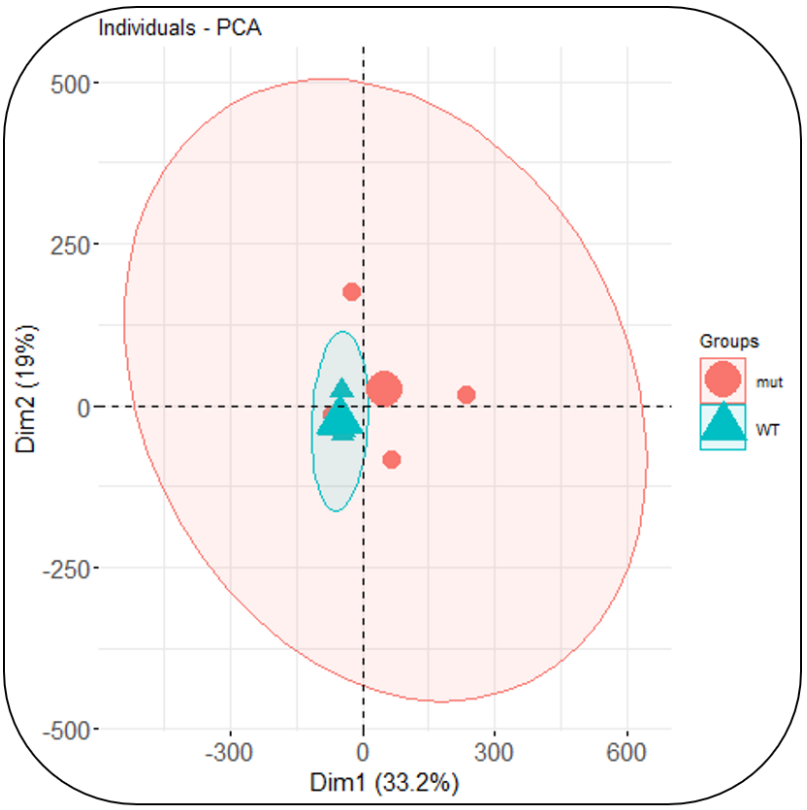

Figure 2- **Représentation graphique des résultats de l'ACP.** *Les individus des groupes mut et WT sont représentés dans le plan défini par les deux composantes principales.*

Pour l'analyse des gènes différentiellement exprimés, on observe que 4 transcrits sont considérés d'intérêt en fonction des seuils de significativité et de valeur absolue de log2FoldChange que nous avons indiqué (respectivement 0.05 et 1) : ENSG00000144824, ENSG00000198795, ENSG00000125816 et ENSG00000169548 (**Fig.3**). Les noms des genes associés suivant la nomenclature HUGO sont respectivement PHLDB2, ZNF521, NKX2-4 et ZNF280A. Les 3 premiers ont une valeur de log2FoldChange positive et ENSG00000169548 a une valeur de log2FoldChange négative. Dans notre cas, cela signifie que les 3 premiers sont significativement positivement différentiellement exprimés dans le groupe WT par rapport au groupe mut, et le dernier est significativement négativement différentiellement exprimé dans le groupe WT par rapport au groupe mut. Un volcano plot a été obtenu à l'aide du package EnhancedVolcano (**Fig.4**).

	log2FoldChange	padj	HUGO
ENSG00000144824	4.194941	0.036564611	PHLDB2
ENSG00000198795	3.595302	0.045288551	ZNF521
ENSG00000125816	7.586422	0.036564611	NKX2-4
ENSG00000169548	-8.666021	0.001707595	ZNF280A

Figure 3- **Tableau récapitulatif des gènes différentiellement exprimés.** *Le log2FoldChange, la p-value ajusté et le nom de gène suivant la nomenclature HUGO sont représentés pour chaque gène.*

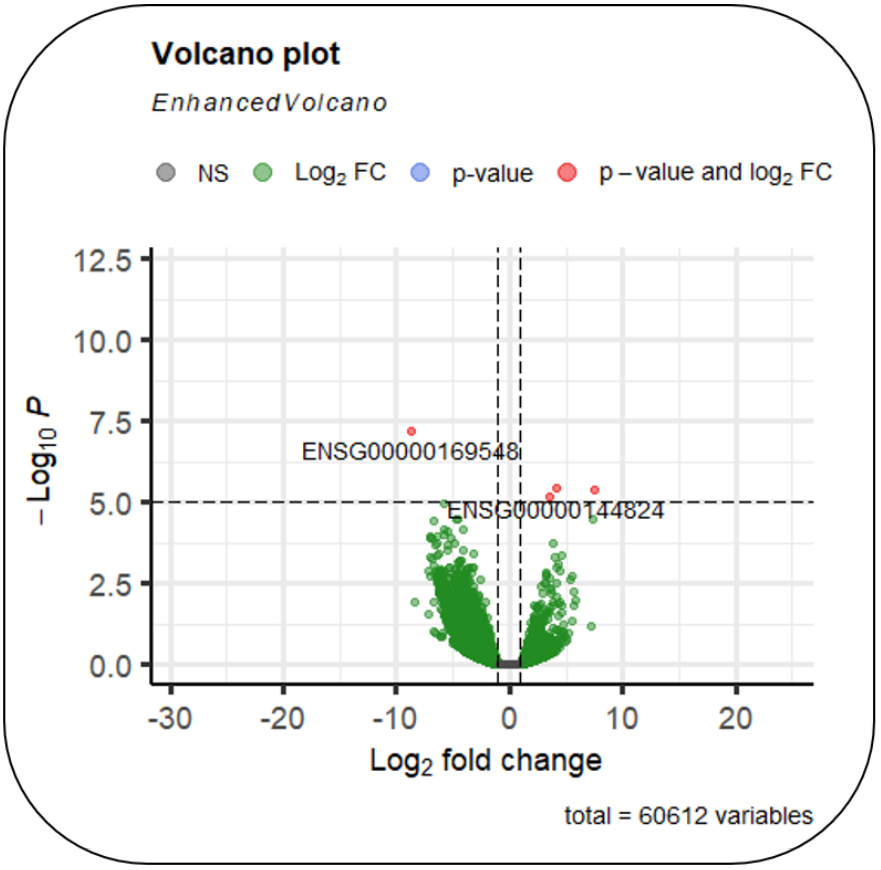

Figure 4-**Volcano plot des gènes différentiellement exprimés.** *4 gènes différentiellement exprimés (représentés en rouge) ont été identifies (4 plus exprimés et 1 moins exprimé dans le groupe WT que dans le groupe mut).*

IV) Matériel et méthodes

1) Jeu de données

Les données à l'origine des analyses réalisées dans ce projet proviennent, comme dit précédemment, de l'étude de Harbour et Roberson [4] réalisée en 2013. Les données sont les reads d'ARN des tissus cancéreux de huit patients atteints d'un cancer oculaire de type léger (classe 1), quatre des patients étant mutants sur le facteur d'épissage SF3B1 et les quatre autres de type sauvage sur ce même facteur. Les transcrits sont présentés au format "Ensembl gene ID".

Le génome humain de référence utilisé est l'assemblage datant de 2013 nommé GRCh38. En effet, il s'agit de la version/build 38, créée par le Genome Reference Consortium, un collectif international d'instituts universitaires pour la création de génomes de référence. Les données d'annotation utilisées proviennent aussi du GRCh38. On peut noter que de nouveaux assemblages du génome humain ont eu lieu depuis 2013, mais étant donné que l'objectif du projet était de reproduire les résultats des articles [4 et 5], on utilisera des données de référence disponibles à l'époque des articles.

2) Difficultés rencontrées

Au cours de ce projet, nous avons rencontré un certain nombre de difficultés en faisant tourner le script Nextflow.

Tout d'abord, nous n'avons pas réussi à faire fonctionner la commande fastq-dump sur Nextflow avec le conteneur evolbioinfo/sratoolkit:v2.5.7. A la place, nous avons choisi de nous servir de la commande fasterq-dump du conteneur pegi3s/sratoolkit. De plus, nous avons eu une erreur "failed to publish file" avec la commande publishDir en mode link. Autrement dit, Nextflow ne parvenait pas à déplacer les fichiers dans le dossier précisé. Cette erreur n'était pas présente pour tous les membres du groupe. Pour corriger cela, nous avons néanmoins choisi de changer le mode en symlink (mode par défaut), ce qui a bien fonctionné.

Nous avons également eu des problèmes de stockage et de manque de mémoire vive en prenant des machines virtuelles trop petites. Pour le process qui réalise l'indexation du génome, nous n'avions toujours pas de résultat après plus de quatre heures avec 8 coeurs. C'est pourquoi nous sommes passés à une machine à 16 coeurs.

Enfin, il nous a fallu du temps pour parvenir à faire tourner le script R avec Nextflow et pour installer tous les packages, notamment EnhancedVolcano qui s'installe un peu différemment puisqu'il vient de Bioconductor.

3) Versions des outils et options

La liste des outils utilisés pour ce projet est disponible ci-dessous, avec les versions et options utilisées.

- **Nextflow**

Version 21.10.0.5640 intégrée à la VM

- **Docker**

Version 20.10.11 intégrée à la VM

- **Git**

Version 2.25.1 intégrée à la VM

- **SRA Toolkit**

Conteneur pegi3s/sratoolkit : version 2.10.0

Options fasterq-dump :
--*threads* nombre de threads utilisés par la commande
--*split-files* à utiliser pour les reads paired-ends

- **STAR**

Conteneur evolbioinfo/star:v2.7.6a : version 2.7.6a

Options pour l'index	Options pour le mapping
--*runThreadN* nombre de threads --*runMode* genomeGenerate demande à STAR de générer un index --*genomeDir* chemin vers le répertoire dans lequel stocker l'index --*genomeFastaFiles* chemin vers les fichiers fasta du génome de référence	--*outSAMstrandField intronMotif* pour que les fichiers soient utilisables avec l'outil Cufflinks --*outFilterMismatchNmax* nombre maximum de non-appariements par paire --*outFilterMultimapNmax* nombre maximum de loci auxquels le read peut se mapper --*genomeDir* chemin vers le répertoire dans lequel stocker les résultats du mapping --*readFilesIn* chemin contenant le nom des fichiers à mapper --*runThreadN* --*outSAMunmapped None* ne renvoie pas les reads non mappés en format SAM --*outSAMtype BAM SortedByCoordinate* renvoie un fichier BAM trié --*outStd BAM_SortedByCoordinate* le fichier BAM trié est l'output de la commande --*genomeLoad NoSharedMemory* pas d'utilisation de la mémoire partagée pour charger le génome en mémoire --*limitBAMsortRAM* mémoire vive maximum en octets pour trier le fichier BAM

- **SAMtools**

Conteneur evolbioinfo/samtools:v1.11 : version 1.11

Options :
index suivi du nom du fichier BAM trié à indexer

- **Subread**

Conteneur evolbioinfo/subread:v2.0.1 : version 2.0.1

Options featureCounts :

-T nombre de threads	*-s 0* précise que les reads sont unstranded
-t gene pour avoir un comptage sur les gènes	*-a* nom du fichier d'annotation
-g gene_id identifiant à stocker dans l'output	*-o* nom du fichier en output

- **R**

Conteneur evolbioinfo/deseq2:v1.28.1 : R version 4.0.2

Packages DESeq2 version 1.28.1, FactoMineR version 2.4, factoextra version 1.0.7, EnhancedVolcano version 1.12.0

4) Environnement de travail

Pour faire tourner notre workflow, nous avions besoin de disposer de machines suffisamment puissantes pour indexer un génome humain complet et stocker toutes les données générées, c'est pourquoi nous avons utilisé des appliances Biosphere du cloud IFB. Pour pouvoir lancer une appliance Biosphere, l'utilisateur doit faire partie d'un groupe actif : dans notre cas, il s'agit du groupe ReproHack2021. Pour faire tourner notre workflow, nous utilisons le cloud de l'IFB-core, localisé à Lyon, sur des appliances BioPipes disposant notamment de Conda pour l'installation d'outils, Nextflow et Docker. Ces applicances disposent de Ubuntu 18.04. La machine virtuelle utilisée pour le workflow final dispose de 16 coeurs, 64 Go de mémoire vive et 400 Go de stockage.

V) Conclusion

1) Reproductibilité

Dans l'article [4], les analyses d'expression différentielle ont été utilisées à partir de 5 mutants SF3B1 et 6 types sauvages SF3B1, qui présentaient tous une tumeur de classe 1. Il s'agit d'une méthode de type microarray et les données sont obtenues avec Illumina Ref8 Bead Arrays. Les données ont subi une "cubic spline normalization" et une "background subtraction" à l'aide du BeadStation software, puis elles ont été analysées en utilisant la méthode (Significance Analysis of Microarrays (http://www-stat.stanford.edu/~tibs/SAM/)) et un « false discovery rate » (FDR) ≤ 5%. Cette analyse a fait sortir 10 gènes différentiellement exprimés.

Dans l'article [5], ces analyses ont été réalisées à partir de 3 mutants SF3B1 et 3 types sauvages SF3B1, aussi sur des données microarray, en utilisant l'outil GenoSplice avec un test de Student t non-apparié et en considérant les transcrits d'intérêt comme significativement différentiellement exprimés quand le fold change est ≥ 1.5 et la p-value ≤

0.05 (p-value non ajustée). Avec cette méthode et ce jeu de données, 325 gènes sont prédits comme différentiellement exprimés (46 plus exprimés, 279 sous exprimés) dans le groupe de mutants SF3B1 comparé au wild-type.

Dans notre analyse, nous avons utilisé des données RNA-seq disponibles pour 4 mutants SF3B1 et 4 types sauvages SF3B1. Nous avons utilisé le package R DESeq2 avec un seuil de significativité de 0.05 (p-value ajustée) et un seuil de valeur absolue de log2FoldChange de 1, et obtenu 4 gènes différentiellement exprimés (3 plus exprimés chez le groupe WT et 1 moins exprimé).

La grande variabilité de ces résultats nous confirme que le choix de données, le choix des méthodes et le choix des paramètres sont des enjeux clés de la reproductibilité, et qu'il est donc nécessaire de rendre ces informations accessibles pour permettre une démarche scientifique rigoureuse.

2) Retour sur le projet

Ce projet nous a permis d'une part de mettre en œuvre les principes de reproductibilité d'un projet scientifique et d'autre part, d'utiliser pour la première fois Nextflow et Docker. Le projet nous a aussi permis de nous initier à l'analyse NGS, compétence qui nous a depuis déjà servi pour l'UE NGS.

VI) Bibliographie

1- Nekrutenko, Anton, et James Taylor. « Next-Generation Sequencing Data Interpretation: Enhancing Reproducibility and Accessibility ». Nature Reviews Genetics 13, no 9 (septembre 2012): 667‑72. https://doi.org/10.1038/nrg3305.

2- Bortolus, Alejandro. « Error Cascades in the Biological Sciences: The Unwanted Consequences of Using Bad Taxonomy in Ecology ». AMBIO: A Journal of the Human Environment 37, no 2 (mars 2008): 114‑18. https://doi.org/10.1579/0044-7447(2008)37[114:ECITBS]2.0.CO;2.

3- Schillert, Arne. « Implementing Reproducible Research. V.Stodden, F.Leisch, R. D.Peng (Eds.) (2014). Boca Raton, FL: Chapman & Hall/CRC The R Series. 448 Pages, ISBN: 9781466561595.: Book Review ». Biometrical Journal 57, no 6 (novembre 2015): 1149‑50. https://doi.org/10.1002/bimj.201500120.

4- Harbour, J William, Elisha D O Roberson, Hima Anbunathan, Michael D Onken, Lori A Worley, et Anne M Bowcock. « Recurrent Mutations at Codon 625 of the Splicing Factor SF3B1 in Uveal Melanoma ». Nature Genetics 45, no 2 (février 2013): 133‑35. https://doi.org/10.1038/ng.2523.

5- Furney, Simon J., Malin Pedersen, David Gentien, Amaury G. Dumont, Audrey Rapinat, Laurence Desjardins, Samra Turajlic, et al. « SF3B1 Mutations Are Associated with Alternative Splicing in Uveal Melanoma ». Cancer Discovery 3, no 10 (octobre 2013): 1122‑29. https://doi.org/10.1158/2159-8290.CD-13-0330

6- Goecks, J., Nekrutenko, A., & Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, *11*(8), 1-13.

7- Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature biotechnology*, *35*(4), 316-319.

8- Boettiger, C. (2015). An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review, 49(1), 71-79.

9- Spinellis, D. (2012). Git. *IEEE software*, *29*(3), 100-101.

10- Blanchet, C., Collin, O., Boudet, M., Delmotte, S., Gilquin, H., Guillaume, J. F., ... & Spataro, B. (2019, December). IFB-Biosphère: Services cloud pour l'analyse des données des sciences de la vie. In *Journées RESeaux-JRES 2019*.

11-Kodama, Y., Shumway, M., & Leinonen, R. (2012). The Sequence Read Archive: explosive growth of sequencing data. *Nucleic acids research*, *40*(D1), D54-D56.

12-Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., ... & Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, *29*(1), 15-21.

13- Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., ... & Durbin, R. (2009). The sequence alignment/map format and SAMtools. Bioinformatics, 25(16), 2078-2079.

14- Liao, Y., G. K. Smyth, et W. Shi. « FeatureCounts: An Efficient General Purpose Program for Assigning Sequence Reads to Genomic Features ». Bioinformatics 30, no 7

(1 avril 2014): 923‑30. https://doi.org/10.1093/bioinformatics/btt656.

15- Gentleman, R., Carey, V. J., Huber, W., Irizarry, R. A., & Dudoit, S. (Eds.). (2005). Bioinformatics and computational biology solutions using R and Bioconductor (Vol. 1). New York: Springer.

16- Husson, F., Josse, J., Le, S., Mazet, J., & Husson, M. F. (2016). Package 'FactoMineR'. *An R package*, *96*, 698.

17- Kassambara, A., & Mundt, F. (2017). Package 'factoextra'. *Extract and visualize the results of multivariate data analyses*, *76*.

18- Love, M., Anders, S., & Huber, W. (2014). Differential analysis of count data–the DESeq2 package. *Genome Biol*, *15*(550), 10-1186.

19-Blighe K, Rana S, Lewis M (2021). *EnhancedVolcano: Publication-ready volcano plots with enhanced colouring and labeling*. R package version 1.12.0, https://github.com/kevinblighe/EnhancedVolcano.