Rapport Hackathon - Groupe 11

Noémie JACQUET, Amine KABECHE , Nora PICAUT, Anne-Cécile TOULEMONDE

Date : 12 / 12 / 2022

L'ensemble des images dockers, dworkflows et script sont disponibles sur le repository :Github.

1 Contexte

La science s'intéresse aux phénomènes reproductibles c'est à dire qui peuvent être reproduits à volonté. La reproductibilité d'une expérience scientifique permet de s'affranchir des effets aléatoires et des erreurs de jugement ou de manipulation. Ainsi, elle permet d'inclure les observations faites au cours de l'expérience dans le processus d'amélioration des connaissances du domaine.

Dans ce projet, nous nous intéressons aux travaux réalisés sur le déterminisme génétique du mélanome uvéal. Le mélanome uvéal est le cancer primaire de l'œil le plus fréquent. Il est souvent fatal. Les travaux de [2] et [1] décrivent deux classes de mélanome uvéal :

- Les tumeurs de classe 1 sont rarement métastasées, ont peu de cellules épithéliales indifférenciées, et touchent des patients plus jeunes que ceux de la classe 2.
- Les tumeurs de classe 2 sont souvent métastasées et sont composées de cellules épithéliales indifférenciées dépourvues de copie du chromosome 3 (monosomie). Elles sont associées à des mutations entraînant une perte de fonction de *BAP1* (BRCA1-associated protein 1) située sur le chromosome 3 (à l'origine de 40% des mélanomes uvéaux, tous de classe 2)

Le but du travail de J. William Harbour et al (2013) est de trouver des mutations autres que BAP1 qui permettraient de comprendre les déterminants génétiques du mélanome uvéal. Ils ont constaté que 2 gênes présentent des mutations somatiques délétères dans 3 échantillons sur 18 : *GNAQ*(guanine nucleotide-binding protein G(q) subunit alpha) et *SF3B1* (splicing factor 3B subunit 1), qui code pour le facteur 3B. Il s'agit, d'une part, d'un composant du SnRNP U2 (petite ribonucléoprotéine nucléaire U2). Les SnRNPs sont des complexes ARN-protéines qui se combinent avec des pré-ARNm non modifiés et diverses autres protéines pour former un spliceosome. L'action des SnRNPs est essentielle pour l'élimination des introns du pré-ARNm. D'autre part, le facteur 3B est aussi un composant du U12-type du spliceosome mineur qui a pour rôle d'épisser les introns rares avec des séquences de site d'épissage différentes.

Pour expliquer l'effet de la muation *SF3B1* ils ont d'abord suggéré que les mutations de *SF3B1* conduisent à une expression différentielle globale de l'ARN. Ils ont trouvé 10 gênes exprimés de manière différentielle mais aucun ne semble expliquer l'effet des mutations *SF3B1*.

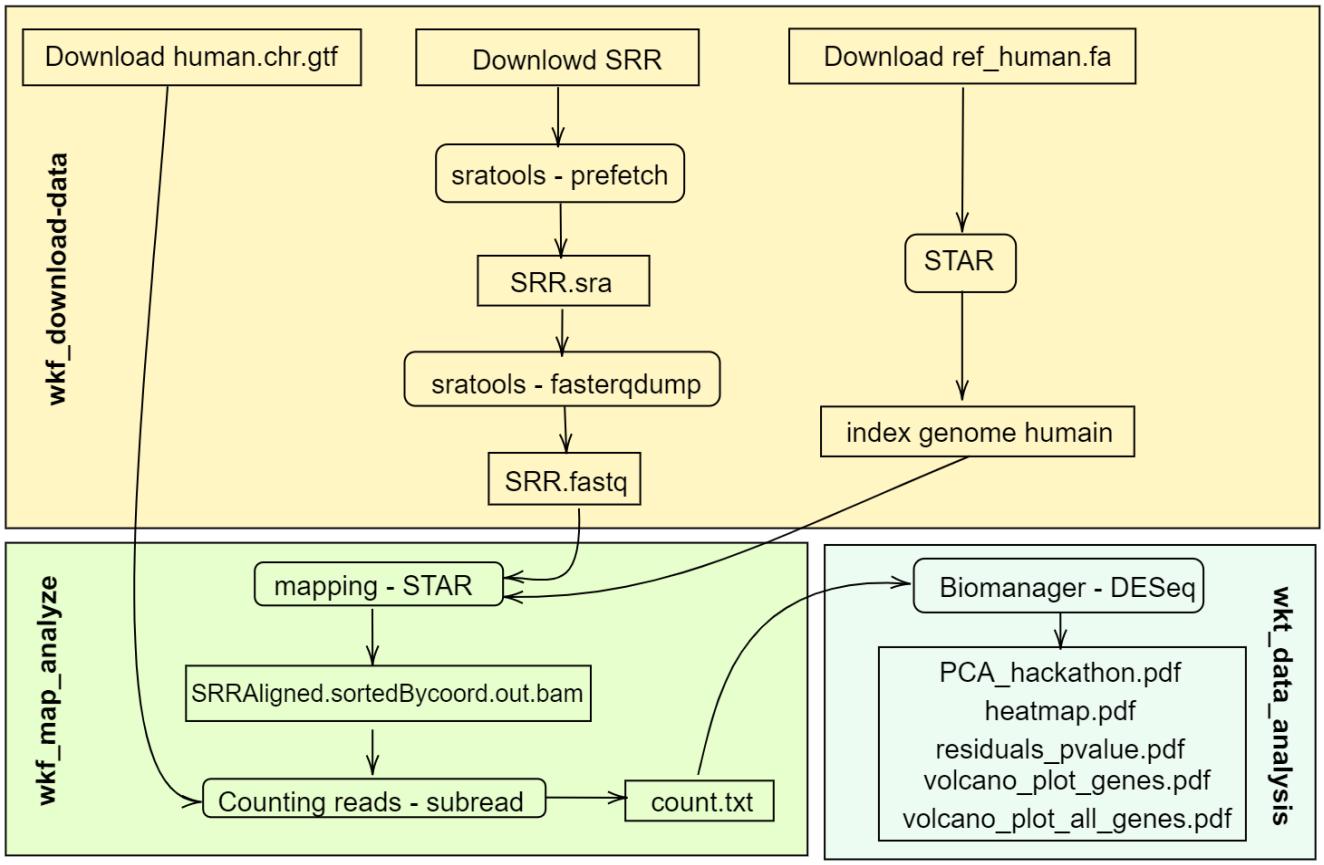
En conséquence, ils ont suggéré que les mutations *SF3B1* conduisent à la rétention des introns. Ils ont donc analysé 8 échantillons de tumeurs de classe 1 : 3 mutants *SF3B1* et 5 non-mutants *SF3B1*, pour voir s'il existe des altérations dans la rétention des donneurs (sites GU-5' en amont des introns) ou des accepteurs (sites AG-3' en aval de l'intron directionnel) des sequences ARN. Mais aucune différence entre les mutants et les non-mutants n'a été trouvée.

Simon J. Furney et al (2017) confirme les observations faites dans la publication de 2013 sur l'effet de la mutation de SF3B1 sur le bon pronostique des patients atteint du mélanome uvéale. Toutefois ils découvrent en plus que les mutations *SF3B1* sont associées à un épissage alternatif différentiel de gênes codants et non codants, notamment *ABCC5*, *UQCC* et *CRNDE* codant pour des protéines impliquées dans les déterminants causaux du mélanome uvéal.

L'objectif du projet Hackaton est donc de reproduire les analyses menées dans les deux publications et vérifier si nos résultats concordent avec ceux publiés.

2 Matériels et Méthodes

2.1 Méthode



2.1.1 Architecture du flux d'analyse

Nous avons construit trois workflows qui figurent en Annexe:

- $wkf\_download - data\_8SRR\_index$: ce premier workflow permet de télécharger les données, de réaliser l'analyse fastqc et de réaliser l'index
- $wkf\_map - analyze\_8SRR$: ce second workflow permet de faire le mapping des gênes et le comptage des reads (les fichiers de sortie de ce workflow sont c̈ountët c̈ount.summary)
- $wkf\_statistical\_anlaysis$: ce workflow qui permet d'effectuer l'analyse de l'expression différentielles des gênes n'est pas encore finalisé sous forme de snakefile, pour l'instant nous avons joint le script R et les instructions pour le lancer à la main.

Workflkow - 1:
Téléchargement des données: Cette première étape est réalisée à l'aide de l'outil sra-tools (fasterq dump) qui permet de télécharger les données des archives de séquences stockées dans la base du NBCI (National Center for Biotechnology) sous format fastq. Cela est sous forme "paired-end" (2 fichiers fastq par SRR) où les reads vont par paires, un sur chaque brin, à une distance moyenne fixe l'un de l'autre. Le paired-end est nécessaire pour bien détecter le splicing.
Analyse de la qualité des données: Les premières paires de bases d'un read sont séquencées avec beaucoup

de fiabilité, mais plus on avance dans la séquence, moins c'est précis. Le programme FastQC génére des rapport fasqc (sous format html notamment) qui permettent de contrôler la qualité.
Génération de fichiers d'index du génome humain: L'outil star, Spliced Transcripts Alignment to a Reference est conçu pour aligner directement les séquences non contiguës de l'échantillon considéré au le génome de référence. On commence donc par creer l'index surquel on va s'appuyer pour aligner les reads de l'échantillon. L'index permet à l'aglorithme de star de chercher rapidement dans le génome les reads correspondant lors de l'étape de mapping. Star prend en entrée les séquences ADN du génome humain entier au format fasta et le fichier d'annotation au format gtf. Ce fichier indique la position de tous des transcrits à l'intérieur du gnéome humain. Cela permet à star extraire les jonctions d'épissures de ce fichier et les utiliser pour améliorer considérablement la précision du mapping à l'étape suivante. Star indexe le gneome humain avec la position de tous les transcrits. Cela va nous permettre de retrouver la position des reads sur le génome. L'outil renvoie en sortie les fichier SA et SA-index qui contiennent une séquence binaire du génome entier et son indexage, puis d'autres fichiers approtant dans informations supplémentaires tels que chrLength, chrName, chrStart, chrNameLength ,exonInfo, geneInfo, transcriptInfo. La plupart de ces fichiers utilisent le format interne star et ne sont pas destinés à être utilisés par l'utilisateur final.

Workflkow - 2
Cartographie des reads dans le génome humain: La tâche de mapping consiste à comparer les séquences ARN à une séquence de référence puis construire pour chaque read des alignements. Elle prend en entrée les fichiers paired-end reads des échantillons au format fastq ainsi que les fichiers générés par la tâche d'indexage du génome. En sortie on obtient des fichiers de reads alignées sur la séquence binaire du génome humain au format BAM.
Comptage des reads: sous l'hypothèse que le nombre de reads venant d'un certain gêne est proportionnel à l'abondance de son ARN dans la cellule, on veut compter les reads venant de chaque gêne, transcrit ou exon du génome. Les fichiers de sortie sont des fichiers texte: une table de comptage et un résumé des résultats du comptage.

Workflkow - 3
À la suite du comptage, la dernière étape de ce projet vise à analyser l'expression des gênes impliqués dans l'expression du cancer uvéal. L'analyse est réalisée sur le logiciel R et s'appuie sur les packages suivants : le package "DESeq2" pour l'analyse différentielle, "fdrtool" pour la modélisation nulle empirique et les packages "ggplot2" et "RColorBrewer" pour l'affichage.

2.1.2 Analyse statistiques d'expression différentielle des gênes

Pré-traitement Après l'importation des données de comptage brute (count.txt), la première étape de l'analayse vise à construire les métadonnées à l'aide de la fonction "DESeqDataSetFromMatrix" du pakage DESeq2.
Par soucis de reproductibilité, nous avons conservé le même plan d'expérience que les deux expériences: Les 3 premiers individus sont considéres comme des individus "mutants" et les 5 autres patients sont considérés comme des individus aux génomes sauvages : "wild" pour le gêne *SF3B1*.
Cette première étape est suivie par un contrôle qualité et une normalisation des données de comptage. Le contrôle qualité permet de vérifier combien de gênes sont différents de zéro dans tous les échantillons. La normalisation a pour but de mettre à une même échelle les valeurs obtenues afin que les niveaux d'expressions ne soient plus dépendants de la profondeur de séquençage de chaque génome. L'outil utilié est la fonction DESeq() intégrée au package DESeq2, elle permet de normaliser le nombre de lectures, d'estimer leurs dispersions et d'ajuster le modèle linéaire. La dispersion des données est évaluée de manière visuelle.

Evaluation de la similarité des génomes: L'évaluation de la similarité entre les échantillons permet de s'assurer de vérifier la concordance avec la conception de l'expérience. La métrique utilisée est la distance euclidienne régularisée par une transformation logarithmique. La régularisation permet de stabiliser la variance des données (3) pour s'assurer que la distance entre les gênes ne repose pas sur une minorité de gênes très variables. Elle garantit ainsi l'homoscédasticité nécessaire à une exploration des données par analyse en composantes principales (ACP) ou par heatmap. Pour limiter le volume de données, l'ACP est projetée sur les 1000 gênes dont la variance est la plus élevée.

Analyse différentielle de l'expression des gênes L'analyse différentielle commence par une estimation visuelle de la dispersion de chaque gêne à l'aide de la fonction "estimateDispersion" du package DESeq2. Cette approche est complétée par un test statistique de l'expression différentielle: $testdeWald$.
En parallèle, il est necessaire de contrôler et corriger les valeurs residuelles des p-values pour s'appuyer sur un modèle nul correct. La validation de ce modèle repose sur l'histogramme des residus de p-value calculé à l'aide du package "fdrtool".
L'extraction des gênes différentiellement exprimés s'appuie sur un MAplot et deux volcano_plot: le premier projette l'ensemble des gênes différentiellement exprimés et le deuxième projette les gênes d'intérets présentées dans les études à l'origine de ce projet (*ADAM12, GNA11, UQCC1, ABCC5, SF3B1, GNAQ, BAP1 et CRNDE*).

2.2 Matériels

2.2.1 Données

Nous disposons comme données de départ des fichiers SRR contenant les séquences de reads de 8 patients (type sauvage versus mutant).

2.2.2 Docker

Pour créer les différentes images, nous avons choisi l'outil Docker qui nous permet d'empaqueter certaines applications et leur dépendance et de lancer ces applications dans des conteneurs isolés. L'ensemble des dockerfiles construit ainsi que les versions des outils qu'ils empaquettent sont présentés dans la liste ci-dessous:

- dockerfile.sra-tools: nous avons dû installer d'abord vdb pour la configuration comme une étape préalable avant sra-tools
- dockerfile.fastqc: La version de *fastqc* est v0.11.9.
- dockerfile_rna-star: La versio de RNA-star est **v**2.7.10b.
- dockerfile.subread-featureCounts: La version de subread est subread-2.0.3
- dockerfile.samtools: La version de samtools est la 1.16.1
- dockefile.R : La version de R installée est 4.2.2 Dans cette même image, sont aussi intégrés les packages DESeq2, ggplot2, RColorBrewer et

L'un des défis consistait à identifier les dépendances nécessaires à la compilation et à l'installation de chaque logiciel. Pour y arriver, nous sommes appuyés sur différentes recettes docker existantes. Une fois construites, les images docker ont étépoussées sur le compte Dockerhub d'un des membres du groupe afin de faciliter le référencement des images dans le workflow.

Pour l'analyse statistique, la question d'importer directement les packages dans le script s'est posée. Après plusieurs essais, cette approche a posé plusieurs problèmes quant à l'installation des packages à chaque exécution du script. C'est pourquoi, nous avons intégré les packages nécessaires directement dans l'image dockerfile.R

2.2.3 Snakemake

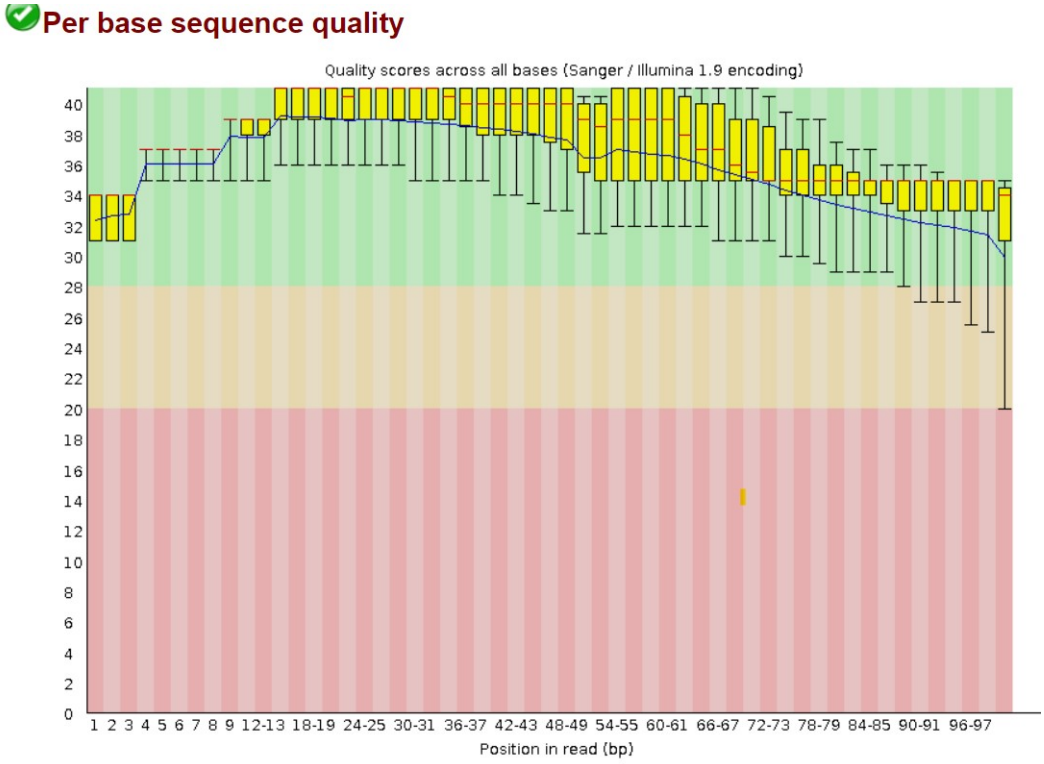
Pour construire le worflow nous avons choisi le logiciel d'edition de workflow : Snakemake. Ce système permet de créer des workflows d'analyse de données reproductibles et de paralléliser certaines tâches, comme nous avions 8 échantillons SRR. De plus, la syntaxe de snakemake est en python.

Pour le premier workflow, le téléchargement des données SRR sous format fastq (avec *sra-tools)* nous avons identifié le besoin d'utiliser *prefetch* avant *fasterqdump*. **P**our le deuxième worflow, nous n'avons finalement pas eu besoin d'utiliser *samtools*. L'execution des workflows nécessite d'utiliser des machines virtuelles (VM) avec des spécifications précises et de gérer correctement l'allocation des quotas. En effet, pour réaliser l'indexation et le mapping du génome humain avec rna-star, nous avons eu besoin d'utiliser une VM avec une RAM d'au moins 64 Go.
L'une des difficultés fut de parvenir à généraliser les règles pour les 8SRR avec le bon usage des "wildcards" Snakemake.

3 Résultats et Discussion

3.1 Qualité des données : fastqc

L'analyse des rappots FastQC indiquent une bonne qualité (cf. exemple de SRR628582_2.fastq ci-dessous). Sur l'ensemble des échantillons, nous n'observons pas de baisse de qualité en fin de read qui aurait alors nécessité un *trimming*.



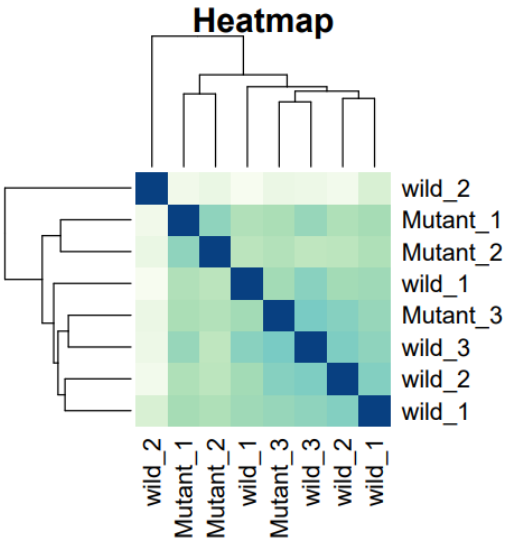
3.2 Counts and reads

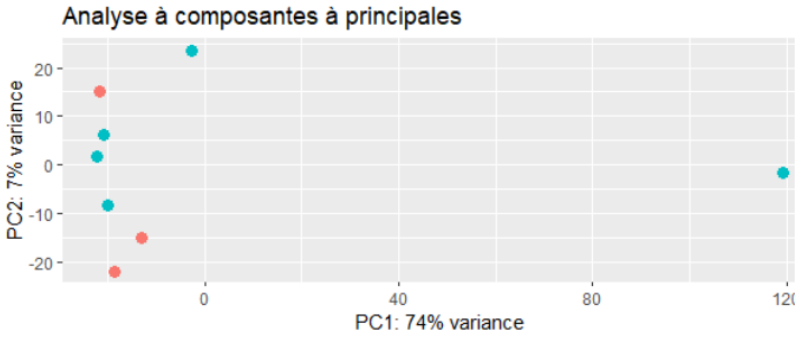
On vérifie à l'aide du fichier count.summary que l'ensemble des gènes ont été mappés: 100% des reads mappent bien sur la référence. On observe également un pourcentage important (57% en moyenne) de reads qui correspondent à des gênes annotés.

Status	SRR628582Ali	SRR628583Ali	SRR628584Ali	SRR628585Ali	SRR628586Ali	SRR628587Ali	SRR628588Ali	SRR628589Alig
Assigned	19788426	21633922	15503440	18368602	15374490	21742625	18168787	24112664
Unassigned_Unmapped	0	0	0	0	0	0	0	0
Unassigned_Read_Type	0	0	0	0	0	0	0	0
Unassigned_Singleton	0	0	0	0	0	0	0	0
Unassigned_MappingQuality	0	0	0	0	0	0	0	0
Unassigned_Chimera	0	0	0	0	0	0	0	0
Unassigned_FragmentLength	0	0	0	0	0	0	0	0
Unassigned_Duplicate	0	0	0	0	0	0	0	0
Unassigned_MultiMapping	5006374	20648970	3037598	16884332	4061642	18664268	7283850	7557156
Unassigned_Secondary	0	0	0	0	0	0	0	0
Unassigned_NonSplit	0	0	0	0	0	0	0	0
Unassigned_NoFeatures	416403	731507	224571	281604	223780	184569	314354	321891
Unassigned_Overlapping_Length	0	0	0	0	0	0	0	0
Unassigned_Ambiguity	5746705	5365613	3737143	4484260	3760188	5922046	4636575	7609895

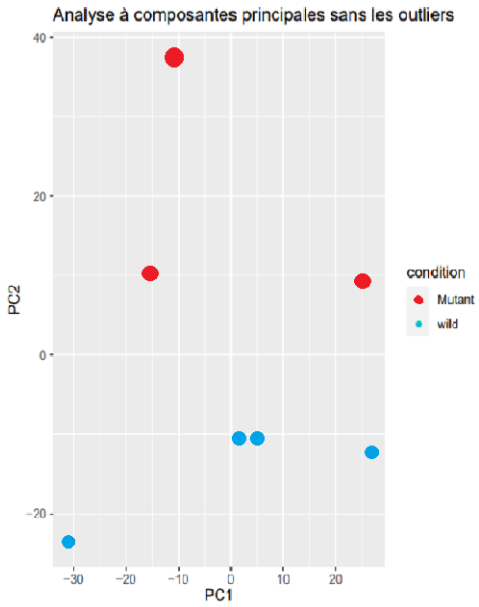
3.3 Analyse statistiques d'expression différentielle des gênes

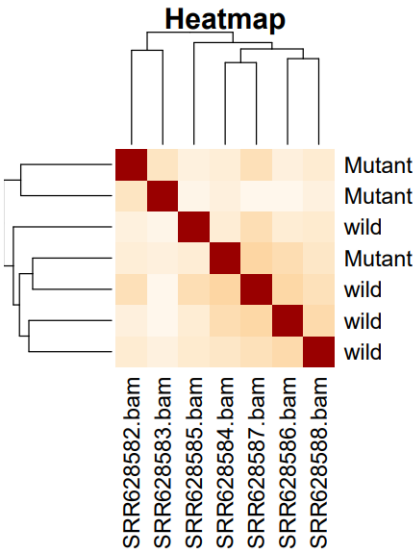
Le contrôle qualité des gênes a révélé que le nombre de gênes qui ont un nombre différent de zéro dans tous les échantillons, est de 16148 sur 60612 au total.



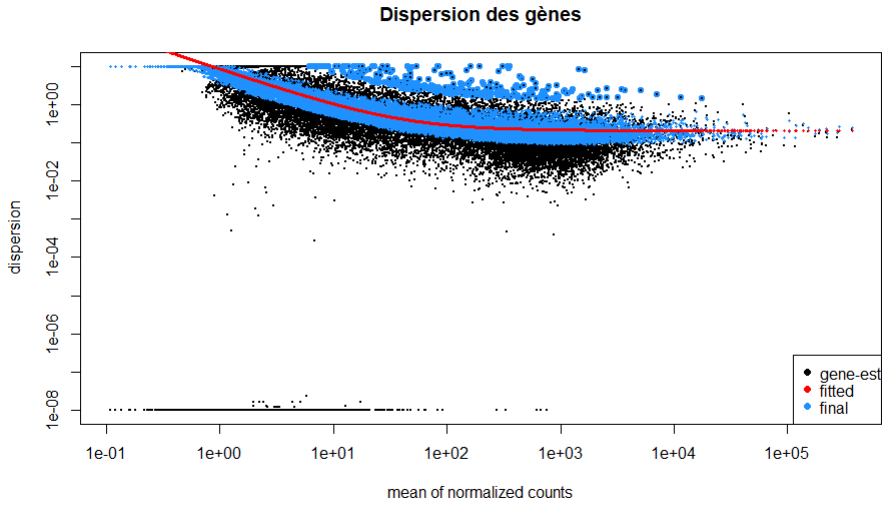


L'analyse des similarités entre les échantillons , montre qu'il y a une valeur aberrante au sein du groupe des "Mutants", l'individu *SRR n°9*. Il peut expliquer la séparation des deux groupes selon le premier axe principal PC1. Il a donc étéécarté de l'analyse, afin de réduire la variabilité totale et augmenter la puissance statistique des tests différentiels qui suivent. On obtient donc une nouvelle analyse à composantes principales où l'on peut observer deux groupes expliqués par l'axe 1 à 25,9% et 24,5 % par l'axe 2.



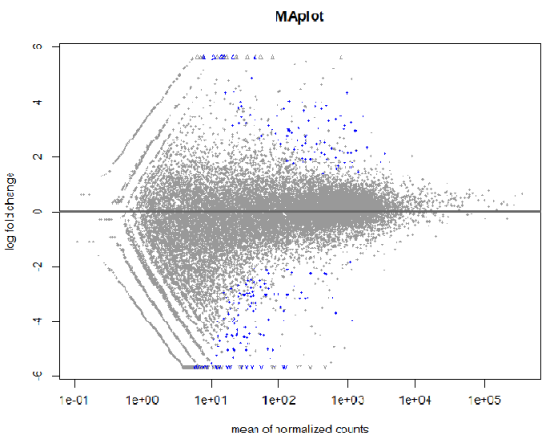


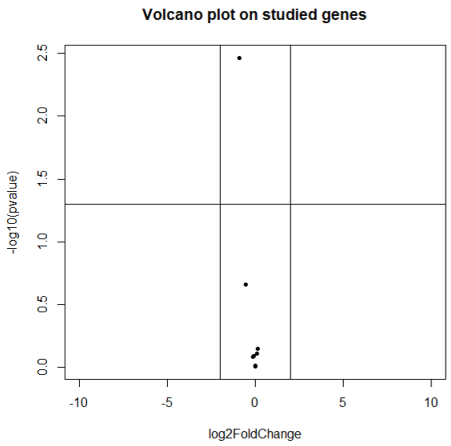
La séparation entre les deux groupes n'est pas précise, ils ne se regroupent pas dans la carte thermique et il semble que y a voir une séparation selon la deuxième composante principale. On s'attend donc à un nombre modeste de gênes exprimés de manière différentielle correspondant au petit nombre de gênes exprimés de manière différentielle ($\tilde{1}$00) rapportés par les auteurs dans la publication originale. L'estimation de la dispersion visualisée par le graphique ci-dessous révèle un nombre important de gênes avec des dispersions élevées et qui ne sont pas ajustées par le modèle. Ainsi, l'estimation de la dispersion n'est pas optimale.

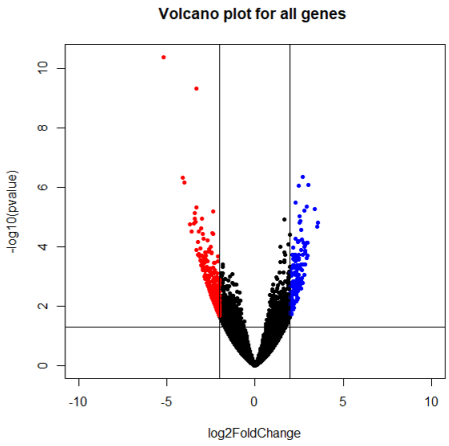


De plus, le *test de Wald* a montré qu'on obtient 394 gênes différentiellement exprimés(avec une padj < 0.1) sur les 17 624. Apres une vérification et un reajustement des p-values (cf ANNEXE:histogramme des p-values correctes):

- MAplot: montre que la médiane est à 0, les échantillons sont donc bien normalisés, cependant il révèle peu de gênes aux proches des bornes vertivales ($y = -6$ et $y = 6$) ainsi qu'une poignée de gênes située vers la borne max de l'axe des x. En somme, peu de gênes semblent expliquer la différence entre les "mutants" et les "wild".
- Volcano-plot: sur les 325 gênes nous portions notre intéret sur 6 d'entre eux. Le volcano plot montre qu'aucun de ces 6 gênes ne s'exprime différentiellement dans l'expression du cancer uvéal des 8 individus étudiés.







Conclusion

En conclusion les outils de reproductibilité tels que docker et snakemake ici testés sont un atout majeur pour garantir l'intégrité des données et la reproductibilité des résultats d'analyse des publications scientifiques. La reproduction des résultats scientifiques précédemment publiés est un processus complexe comme nous l'a montré le présent projet, A l'issue de l'analsye nous avons obtenu des résultats concordants avec la publication de [2], à savoir l'absence de mise en évidence d'épissage alternatif de gênes codants pour des protéines, un épissage alternatif qui serait lié à des mutations du gêne *SF3B1*.

Nous nous sommes heurtés à des obstacles techniques (temps de familiarisation avec les outils et leurs interactions), de ressources (gestion des VM et identification des spécifications requises en termes de RAM notamment) et scientifiques (pour la bonne compréhension des phénomènes génétiques en jeu). Le projet a été riche d'expérience pour l'ensemble du groupe. Pour le mener à bien, nous avons appliqué les méthodes de la gestion de projet et assuré la répartition du travail en équipe, avec une communication régulière sur les différentes avancées et points bloquants du projet. Nous avons identifié différents leviers qui nous permettraient d'aller plus loin:

- Pour améliorer l'organisation de groupe, le levier est de mieux maitriser la gestion des versions de nos fichiers sur Github
- Pour l'analyse différentielle, une analyse plus fine de la normalisation et une compréhension plus approfondie des résultats de l'expression différentielle aurait peut-être pu nous permettre d'identifier d'autres gènes expliquant la séparation entre les deux groupes "Mutant" et "Wild".

References

[1] Simon J Furney, Malin Pedersen, David Gentien, Amaury G Dumont, Audrey Rapinat, Laurence Desjardins, Samra Turajlic, Sophie Piperno-Neumann, Pierre de la Grange, Sergio Roman-Roman, et al. Sf3b1 mutations are associated with alternative splicing in uveal melanoma. *Cancer discovery*, 3(10):1122–1129, 2013.

[2] J William Harbour, Elisha Roberson, Hima Anbunathan, Michael D Onken, Lori A Worley, and Anne M Bowcock. Recurrent mutations at codon 625 of the splicing factor sf3b1 in uveal melanoma. *Nature genetics*, 45(2):133–135, 2013.

Annexes

workflow 1: $wkf\_download - data\_8SRR\_index$

```
SRR=['SRR628582', 'SRR628583', 'SRR628584', 'SRR628585', 'SRR628586',
'SRR628587', 'SRR628588', 'SRR628589']

rule all:
    input:
        expand ("{sample}_{RF}_fastqc.html", sample=SRR, RF=["1","2"]),
"ref_human.fa", "human.chr.gtf", directory("ref")

rule prefetch:
    output: "{sample}/{sample}.sra"
    container: "docker://nj1110/sratools"
    shell:
        "prefetch {wildcards.sample} > {output}"


rule fasterqdump:
    input: "{sample}/{sample}.sra"
    output: "{sample}_1.fastq", "{sample}_2.fastq",
    container: "docker://nj1110/sratools"
    shell:
        "fasterq-dump {input}"

rule fastqc:
    input: "{sample}_1.fastq", "{sample}_2.fastq",
    output: "{sample}_1_fastqc.html", "{sample}_2_fastqc.html",
    container: "docker://nj1110/fastqc:latest"
    shell:
        "fastqc {input}"


rule Download_ref_human:
    output:     "ref_human.fa"
    shell:
            """
        wget -O chromosome.fa.gz
ftp://ftp.ensembl.org/pub/release-101/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38
.dna.chromosome.*.fa.gz
            gunzip -c chromosome.fa > ref_human.fa
            rm chromosome.fa.gz"""

rule DownloadGFF:
    output:"human.chr.gtf"
    shell:
        """
        wget -O $PWD/Homo_sapiens.GRCh38.101.chr.gtf.gz
ftp://ftp.ensembl.org/pub/release-101/gtf/homo_sapiens/Homo_sapiens.GRCh38.101.c
hr.gtf.gz
            gunzip -c Homo_sapiens.GRCh38.101.chr.gtf >  human.chr.gtf
        rm Homo_sapiens.GRCh38.101.chr.gtf.gz"""

rule index_REFhumain:
    input: "ref_human.fa"
    output: directory("ref")
    container: "docker://nj1110/starimg2"


    shell:
        """
        mkdir ref
        chmod 777 ref
        STAR --runThreadN 6 --runMode genomeGenerate --genomeDir ref/
 --genomeFastaFiles {input}"""
```

workflow 2: $wkf\_map - analyze\_8SRR$

```
SRR=['SRR628582', 'SRR628583', 'SRR628584', 'SRR628585', 'SRR628586',
'SRR628587', 'SRR628588', 'SRR628589']


rule all:
    input:
        "count.txt"

rule Mapping:
    input: "{sample}_1.fastq", "{sample}_2.fastq",
    output: "{sample}Aligned.sortedByCoord.out.bam"
    container: "docker://nj1110/starimg2"
    shell:
        """
        STAR --outSAMstrandField intronMotif \
            --runMode alignReads \
        --outFilterMismatchNmax 4 \
        --outFilterMultimapNmax 10 \
        --genomeDir ref \
        --readFilesIn {input} \
        --runThreadN 6 \
        --outSAMunmapped None \
        --genomeLoad NoSharedMemory \
        --outSAMtype BAM SortedByCoordinate \
        --outFileNamePrefix {wildcards.sample} \
        --outTmpDir {wildcards.sample}_STARtmp """


rule Count Reads:
    input: expand("{sample}Aligned.sortedByCoord.out.bam", sample=SRR)
    output: "count.txt"
    container: "docker://nj1110/subread"
    shell: "featureCounts -T 6 -t gene -g gene_id -s 0 -a human.chr.gtf -o
count.txt {input} -p"
```

workflow 3:$wkf\_statistical\_anlaysis$

```
# Définition des fichiers d'entrée
counts = "count.txt"

# Définition des règles
rule all:
    # Définition de la liste des fichiers de sortie
    input:
        "PCA_hackathon.pdf",
        "heatmap.pdf",
        "residuals_pvalue.pdf"
        "volcano_plot_genes.pdf"
        "volcano_plot_all_genes.pdf"

rule dispersion_genes:
    # Spécification des entrées
    input:
        counts
    # Spécification des sorties
    output:
        "PCA_hackathon.pdf",
        "heatmap.pdf",
        "residuals_pvalue.pdf"
        "volcano_plot_genes.pdf"
        "volcano_plot_all_genes.pdf"
    # Utilisation de l'image Docker
    container:
        "froeuhouais/r_image"
    # Exécution du script R
    shell:
        "docker run -v /home/ubuntu/:/dossier/ -it image_r_rocker Rscript dossier/script_8SRR_v5.R"
```